



Marvell[®] 88MC200 Microcontroller

Datasheet



Document Conventions

	Note: Provides related information or information of special importance.
	Caution: Indicates potential damage to hardware or software, or loss of data.
	Warning: Indicates a risk of personal injury.

Document Status

Doc Status: Not applicable	Technical Publication: 0.xx

For more information, visit our website at: www.marvell.com

Disclaimer

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of Marvell. Marvell retains the right to make changes to this document at any time, without notice. Marvell makes no warranty of any kind, expressed or implied, with regard to any information contained in this document, including, but not limited to, the implied warranties of merchantability or fitness for any particular purpose. Further, Marvell does not warrant the accuracy or completeness of the information, text, graphics, or other items contained within this document.

Marvell products are not designed for use in life-support equipment or applications that would cause a life-threatening situation if any such products failed. Do not use Marvell products in these types of equipment or applications.

With respect to the products described herein, the user or recipient, in the absence of appropriate U.S. government authorization, agrees:

- 1) Not to re-export or release any such information consisting of technology, software or source code controlled for national security reasons by the U.S. Export Control Regulations ("EAR"), to a national of EAR Country Groups D:1 or E:2;
- 2) Not to export the direct product of such technology or such software, to EAR Country Groups D:1 or E:2, if such technology or software and direct products thereof are controlled for national security reasons by the EAR; and,
- 3) In the case of technology controlled for national security reasons under the EAR where the direct product of the technology is a complete plant or component of a plant, not to export to EAR Country Groups D:1 or E:2 the direct product of the plant or major component thereof, if such direct product is controlled for national security reasons by the EAR, or is subject to controls under the U.S. Munitions List ("USML").

At all times hereunder, the recipient of any such information agrees that they shall be deemed to have manually signed this document in connection with their receipt of any such information.

Copyright © 2013, Marvell International Ltd. All rights reserved. Marvell, the Marvell logo, Moving Forward Faster, Alaska, Fastwriter, Datacom Systems on Silicon, Libertas, Link Street, NetGX, PHYAdvantage, Prestera, Raising The Technology Bar, The Technology Within, Virtual Cable Tester, and Yukon are registered trademarks of Marvell. Ants, AnyVoltage, Discovery, DSP Switcher, Feroceon, GalNet, GalTis, Horizon, Marvell Makes It All Possible, RADLAN, UniMAC, and VCT are trademarks of Marvell. All other trademarks are the property of their respective owners.

Table of Contents

1	Product Overview	23
1.1	Introduction	23
1.2	Features	24
1.3	Pin Descriptions	26
1.3.1	Pinout.....	26
1.4	Feature Descriptions	39
1.4.1	ARM Cortex-M3 CPU Core	39
1.4.2	Embedded SRAM	39
1.4.3	In-Package Flash.....	39
1.4.4	Boot ROM	39
1.4.5	AHB Bus Matrix	39
1.4.6	Power, Reset and Clock Control.....	39
1.4.7	Direct Memory Access (DMA).....	39
1.4.8	General Purpose IO (GPIO)	40
1.4.9	Watchdog Timer (WDT)	40
1.4.10	Real Time Clock (RTC).....	41
1.4.11	General Purpose Timers.....	41
1.4.12	Advanced Encryption Standard (AES) Engine	41
1.4.13	Cyclic Redundancy Check (CRC).....	42
1.4.14	General Purpose ADC	42
1.4.15	Analog Comparators	42
1.4.16	DAC	43
1.4.17	UART	43
1.4.18	I2C	43
1.4.19	QSPI Interface	44
1.4.20	SSP	44
1.4.21	SDIO	45
1.4.22	USB	45
1.5	Part Ordering.....	45
2	Processor Overview	49
2.1	Overview	49
2.1.1	Cortex M3 Features	49
2.1.2	Memory Protection Unit (MPU).....	49
2.1.3	Nested Vectored Interrupt Controller (NVIC)	49
2.1.4	SysTick Timer	50

3	I/O Configuration	51
3.1	Pinmux Alternate Functions	51
3.1.1	GPIO_0 (Offset=0x0)	52
3.1.2	GPIO_1 (Offset=0x4)	52
3.1.3	GPIO_2 (Offset=0x8)	53
3.1.4	GPIO_3 (Offset=0xC)	53
3.1.5	GPIO_4 (Offset=0x10)	54
3.1.6	GPIO_5 (Offset=0x14)	54
3.1.7	GPIO_6 (Offset=0x18)	55
3.1.8	GPIO_7 (Offset=0x1C)	55
3.1.9	GPIO_8 (Offset=0x20)	56
3.1.10	GPIO_9 (Offset=0x24)	56
3.1.11	GPIO_10 (Offset=0x28)	57
3.1.12	GPIO_11 (Offset=0x2C)	57
3.1.13	GPIO_16 (Offset=0x40)	58
3.1.14	GPIO_17 (Offset=0x44)	58
3.1.15	GPIO_18 (Offset=0x48)	59
3.1.16	GPIO_19 (Offset=0x4C)	59
3.1.17	GPIO_20 (Offset=0x50)	60
3.1.18	GPIO_21 (Offset=0x54)	60
3.1.19	GPIO_22 (Offset=0x58)	60
3.1.20	GPIO_23 (Offset=0x5C)	61
3.1.21	GPIO_24 (Offset=0x60)	61
3.1.22	GPIO_25 (Offset=0x64)	62
3.1.23	GPIO_26 (Offset=0x68)	63
3.1.24	GPIO_27 (Offset=0x6C)	63
3.1.25	GPIO_28 (Offset=0x70)	64
3.1.26	GPIO_29 (Offset=0x74)	64
3.1.27	GPIO_30 (Offset=0x78)	65
3.1.28	GPIO_32 (Offset=0x80)	65
3.1.29	GPIO_33 (Offset=0x84)	66
3.1.30	GPIO_34 (Offset=0x88)	66
3.1.31	GPIO_35 (Offset=0x8C)	67
3.1.32	GPIO_40 (Offset=0xA0)	67
3.1.33	GPIO_41 (Offset=0xA4)	68
3.1.34	GPIO_42 (Offset=0xA8)	69
3.1.35	GPIO_43 (Offset=0xAC)	69
3.1.36	GPIO_44 (Offset=0xB0)	70
3.1.37	GPIO_45 (Offset=0xB4)	70
3.1.38	GPIO_50 (Offset=0xC8)	71
3.1.39	GPIO_51 (Offset=0xCC)	71
3.1.40	GPIO_52 (Offset=0xD0)	72
3.1.41	GPIO_53 (Offset=0xD4)	72
3.1.42	GPIO_54 (Offset=0xD8)	73
3.1.43	GPIO_55 (Offset=0xDC)	73
3.1.44	GPIO_56 (Offset=0xE0)	74
3.1.45	GPIO_57 (Offset=0xE4)	74
3.1.46	GPIO_58 (Offset=0xE8)	75
3.1.47	GPIO_59 (Offset=0xEC)	75
3.1.48	GPIO_60 (Offset=0xF0)	76
3.1.49	GPIO_61 (Offset=0xF4)	76
3.1.50	GPIO_62 (Offset=0xF8)	77
3.1.51	GPIO_63 (Offset=0xFC)	77
3.1.52	GPIO_64 (Offset=0x100)	78
3.1.53	GPIO_65 (Offset=0x104)	78

3.1.54	GPIO_66 (Offset=0x108)	79
3.1.55	GPIO_68 (Offset=0x110)	79
3.1.56	GPIO_72 (Offset=0x120)	80
3.1.57	GPIO_73 (Offset=0x124)	80
3.1.58	GPIO_74 (Offset=0x128)	81
3.1.59	GPIO_75 (Offset=0x12C)	81
3.1.60	GPIO_76 (Offset=0x130)	82
3.1.61	GPIO_77 (Offset=0x134)	82
3.1.62	GPIO_78 (Offset=0x138)	83
3.1.63	GPIO_79 (Offset=0x13C)	83
3.1.64	I/O Padding	84
4	System Control	87
4.1	Overview	87
4.2	Features	87
4.3	Register Description	87
5	Power, Reset, and Clock Control	89
5.1	Overview	89
5.2	Power Supply	89
5.2.1	Power Pins	90
5.2.2	I/O Power Configuration	91
5.2.3	AON Domain	92
5.2.3.1	Ultra Low-Power Comparator	92
5.2.3.2	Brownout Detection	93
5.3	Power Modes	93
5.4	Power Mode Transitions	96
5.5	Wake-up Sources	97
5.5.1	Wake-up from PM1 Mode	97
5.5.2	Wake-up from PM2/3/4 Modes	97
5.5.3	Reset Controller	98
5.6	Clock Controller	98
5.6.1	Overview	98
5.6.2	Clock Sources	99
5.6.3	SFLL	101
5.6.4	Cortex-M3 Core Clock and Bus Clock	101
5.6.5	UART Clocks	102
5.6.6	AUPLL for Audio Clock and USB Clock	102
5.6.7	CAU Clock	103
5.6.8	GPT Clock	104
5.6.8.1	GPT Sampling Clock	104
5.6.9	Clock Output	104
5.7	Register Description	105
6	Memory Map, Interrupts and AHB Bus Fabric	107
6.1	Overview	107
6.2	Memory Map	107
6.3	Interrupts	110
6.4	AHB Bus Fabric	113

7	Direct Memory Access Controller (DMA)	117
7.1	Overview	117
7.2	Features	117
7.2.1	DMA Operation	117
7.2.2	DMA Block Diagram	117
7.2.3	Basic Definitions	118
7.2.4	Peripheral Burst Transaction Requests	119
7.2.4.1	Watermark Level and Transmit FIFO Underflow	121
7.2.4.2	Choosing the Transmit Watermark Level	121
7.2.4.3	Selecting DEST_MSIZ and Transmit FIFO Overflow	123
7.2.4.4	Receive Watermark Level and Receive FIFO Overflow	123
7.2.4.5	Choosing the Receive Watermark Level	123
7.2.4.6	Selecting SRC_MSIZ and Receive FIFO Underflow	124
7.2.5	Interrupt	124
7.2.6	DMA Channel Mapping	125
7.2.7	Operation Mode	125
7.3	Register Descriptions	126
8	Real Time Clock (RTC)	127
8.1	Overview	127
8.2	Functional Description	127
8.2.1	Counter Clock	127
8.2.2	Counting Mode	128
8.2.3	Counter Update Mode	128
8.2.4	Interrupt	128
8.3	Programming Notes	128
8.3.1	Initialization	128
8.3.2	UPP_VAL	129
8.4	Register Description	129
9	General Purpose Timers (GPT)	131
9.1	Overview	131
9.2	Functional Description	131
9.2.1	Counter	133
9.2.1.1	Counter Clock	133
9.2.1.2	Counting Mode	133
9.2.1.3	Counter Update Mode	134
9.2.2	Interrupt	134
9.2.3	Channel Operation Modes	135
9.2.3.1	Counter Match Register 0 and 1 (CMR0 and CMR1)	135
9.2.3.2	No Function Mode	135
9.2.3.3	Input Capture Mode	135
9.2.3.4	One-Shot Pulse Mode	136
9.2.3.5	One-Shot Edge Mode	137
9.2.3.6	Pulse-Width Modulation (PWM) Edge-Aligned Mode	138
9.2.3.7	Pulse-Width Modulation (PWM) Center-Aligned Mode	139
9.2.4	ADC Trigger	141
9.2.5	DAC Trigger	142

9.3	Programming Notes	143
9.3.1	Initialization	143
9.3.2	UPP_VAL	144
9.3.3	User Request Register	144
9.4	Register Description	144
10	Secure Digital Input/Output (SDIO Controller).....	145
10.1	Overview	145
10.2	Signal Descriptions	145
10.3	Controller Operation	146
10.3.1	Operation	148
10.3.1.1	Data Transfers	148
10.4	Commands and Operations	149
10.4.1	Overview	149
10.4.1.1	Read/Write Commands	149
10.4.2	Controller Functional Description	150
10.5	Interrupts	152
10.6	Clock Control	153
10.7	Data FIFOs	153
10.7.1	Command Response Register	153
10.7.2	Receive Data FIFO Configuration	154
10.7.3	Transmit Data FIFO Configuration	154
10.7.4	DMA and Programmed I/O	154
10.8	Low-Power Mode Operation	155
10.9	Card Communication Protocol	155
10.9.1	PIO Operation	155
10.9.2	DMA Operation	156
10.9.3	Abort Transaction	157
10.9.3.1	Synchronous Abort	157
10.10	Register Descriptions	158
11	USB OTG Interface Controller	159
11.1	Features	159
11.2	Internal Bus Interface	160
11.2.1	DMA Engine	160
11.2.2	Dual Port RAM Controller	161
11.2.3	Protocol Engine	161
11.2.4	Port Controller	161
11.3	Signal Descriptions	161
11.4	Functional Description	162
11.4.1	Host Data Structure	162
11.5	USB Controller Operation	163
11.5.1	FIFO Operation in Device Mode	163
11.5.1.1	Streaming Mode	163
11.5.1.2	Additional Notes on TX FIFO Buffering – IN Endpoints	164
11.5.1.3	Non-Streaming Mode	165
11.5.1.4	FIFO Operation in Host Mode	165
11.5.2	Clock Control and Enables	168
11.5.3	Programming Guidelines	168

11.6	Register Descriptions	168
12	WatchDog Timer (WDT)	169
12.1	Functional Description	169
12.1.1	Counter Operation	169
12.1.2	Interrupt	169
12.1.3	System Reset	170
12.1.4	Reset Pulse Length	170
12.2	Initialization Sequence	170
12.3	Register Description	171
13	Quad Serial Peripheral Interface (QSPI) Controller	173
13.1	Overview	173
13.2	Features List	173
13.3	Block Diagram	173
13.4	IO Description	174
13.5	Functional Description	174
13.5.1	Basic Operation	174
13.5.2	Serial Flash Data Format	175
13.6	Usage Models and Programming Notes	182
13.7	QSPI0 Interface to In-package Serial Flash	182
13.7.1	Basic Read to Serial Flash Without DMA and Using Polling	182
13.7.1.1	Page Program to Serial Flash Without DMA and Using Polling	182
13.8	Register Description	183
14	In-Package Flash	185
14.1	Overview	185
14.2	Features	185
14.3	Block Diagram	186
14.4	Functional Description	187
14.4.1	QSPI0 Interface	187
14.4.1.1	Standard SPI Operation	187
14.4.1.2	Dual SPI Operation	187
14.4.1.3	Quad SPI Operation	187
14.4.2	Write Protection	187
14.4.2.1	Write Protect Features	187

14.5	Control and Status Registers	188
14.5.1	Status Register	188
14.5.1.1	BUSY	188
14.5.1.2	Write Enable Latch (WEL)	188
14.5.1.3	Block Protect Bits (BP2, BP1, BP0)	188
14.5.1.4	Top/Bottom Block Protect (TB)	188
14.5.1.5	Sector/Block Protect (SEC)	188
14.5.1.6	Complement Protect (CMP)	188
14.5.1.7	Status Register Protect (SRP1, SRP0)	189
14.5.1.8	Erase/Program Suspend Status (SUS)	189
14.5.1.9	Security Register Lock Bits (LB3, LB2, LB1)	189
14.5.1.10	Quad Enable (QE)	189
14.5.2	Instructions	192
14.5.2.1	Instruction Set Table 1 (Erase, Program Instructions) 1	194
14.5.2.2	Instruction Set Table 2 (Read Instructions)	195
14.5.2.3	Instruction Set Table 3 (ID, Security Instructions)	196
15	General Purpose Input Output (GPIO)	197
15.1	Overview	197
15.2	GPIO Block Diagram	197
15.3	GPIO Function Description	197
15.3.1	GPIO Ports	197
15.3.2	I/O Control	198
15.3.3	GPIO Interrupt	198
15.3.4	External Interrupts	198
15.4	GPIO Register	199
16	Advanced Encryption Standard (AES)	201
16.1	Features	201
16.2	Functional Description	201
16.2.1	AES Operational Flow	201
16.2.2	AES Configuration	202
16.2.3	Data Access Method	203
16.2.4	Starting the AES Engine	203
16.2.5	Interrupt Request	203
16.2.6	Partial Code Support	204
16.2.7	Error Status Check	204
16.2.8	Output Vector	204
16.2.9	AES Operation Pseudo Code	205
16.3	References for AES Standard	206
16.4	Register Description	206
17	Cyclic Redundancy Check (CRC)	207
17.1	Overview	207
17.2	Features	207
17.3	CRC Operation Flow	207
17.4	Register Descriptions	208
18	Universal Asynchronous Receiver Transmitter (UART)	209
18.1	Overview	209

18.2	Block Diagram	209
18.3	Function Description	210
18.3.1	External Signal Descriptions	210
18.3.2	Protocol	211
18.3.3	SIR Protocol	211
18.3.4	FIFO Access	212
18.3.5	Calculating Baud Rates	213
18.3.6	Interrupts	213
18.3.7	DMA Support	213
18.3.8	Auto Flow Control	214
18.4	Register Descriptions	216
19	Inter-Integrated Circuit (I2C)	217
19.1	Overview	217
19.2	Features	217
19.3	Signal Descriptions	217
19.4	Operation	218
19.4.1	I2C Block Diagram	218
19.4.2	I2C Bus Terminology	218
19.5	I2C Behavior	219
19.5.1	START and STOP Generation	220
19.5.1.1	Combined Formats	221
19.5.2	I2C Protocols	221
19.5.2.1	START and STOP Conditions	221
19.5.2.2	Addressing Slave Protocol	221
19.5.2.3	Transmitting and Receiving Protocol	222
19.5.3	Multiple Master Arbitration	224
19.5.4	Clock Synchronization	224
19.5.5	Operation Modes	225
19.5.5.1	Slave Mode Operation	225
19.5.5.2	Master Mode Operation	228
19.5.6	I2C.CLK Frequency Configuration	229
19.5.6.1	Calculating High and Low Counts	229
19.5.7	DMA Controller Interface	230
19.6	Register Descriptions	230
20	Synchronous Serial Protocol (SSP)	231
20.1	Overview	231
20.2	Features	231
20.3	External Signal Descriptions	231

20.4	Operation	232
20.4.1	FIFO Operation	232
20.4.1.1	Parallel Data Formats for FIFO Storage	233
20.4.1.2	FIFO Operation in Packed Mode	233
20.4.1.3	Trailing Bytes in RXFIFO	233
20.4.2	Using Programmed I/O Data Transfers	234
20.4.3	Using DMA Data Transfers	234
20.4.4	Data Formats	234
20.4.4.1	Serial Data Formats for Transfer to/from Peripherals	234
20.4.4.2	TI-SSP Format Details	235
20.4.4.3	Motorola SPI Format Details	237
20.4.5	Programmable Serial Protocol (PSP) Format	240
20.4.5.1	High Impedance on SSPx_TXD	244
20.4.6	Network Mode	247
20.4.6.1	Network Mode Registers	248
20.4.7	I2S Emulation Using SSP	249
20.5	Register Descriptions	251
21	Analog Digital Converter (ADC)	253
21.1	Overview	253
21.2	Features	253
21.3	External Signal Description	253
21.4	ADC Functional Description	254
21.4.1	ADC Block Diagram	254
21.4.2	ADC On-Off Control and Conversion Trigger	255
21.4.3	ADC Input	255
21.4.4	Input Range	257
21.4.5	Temperature Measurement	257
21.4.6	ADC Reference Voltage	258
21.4.7	ADC Throughput and Resolution	259
21.4.8	ADC Conversion Results	259
21.4.9	ADC Interrupts	261
21.4.10	ADC Calibration	261
21.4.11	DMA Request	262
21.4.12	Battery Monitor	262
21.4.13	External Trigger from GPT	262
21.5	Register Description	262
22	Digital Analog Converter (DAC)	263
22.1	Overview	263
22.2	Features	263
22.3	External Signal Description	263

22.4	DAC Configuration	263
22.4.1	Synchronous Mode	264
22.4.2	Asynchronous Mode	265
22.4.3	Sinusoidal Waveform Generation	265
22.4.4	Triangle Waveform Generation.....	266
22.4.4.1	Up and Down Mode.....	266
22.4.4.2	Up Mode	266
22.4.5	Noise Generation	267
22.4.6	DMA Request	267
22.4.7	Event Trigger from GPT or GPIO	267
22.5	Registers Description	267
23	Analog Comparator (ACOMP)	269
23.1	Overview	269
23.1.1	Features.....	269
23.2	External Signal Description	269
23.3	Functional Description.....	270
23.3.1	ACOMP0/1 Control Signals	270
23.3.1.1	Warmup Time.....	270
23.3.1.2	Response Time	270
23.3.1.3	Hysteresis.....	270
23.3.2	Comparator Output	271
23.3.2.1	Asynchronous Comparison Output at Register	271
23.3.2.2	Synchronous/Asynchronous Comparison Output at GPIO	271
23.3.2.3	Comparison Output Inversion.....	272
23.3.3	Comparator Output Edge Detection	272
23.3.4	Interrupt	274
23.4	Register Description	276
24	Boot ROM	277
24.1	Overview	277
24.2	Boot ROM Flow Charts	277
24.2.1	Loading Code Through UART	281
24.2.2	Loading Code Directly from Flash	281
24.2.3	PM3 Wakeup	281
24.3	Flash Image Format.....	281
24.3.1	BootInfo/Section Header.....	282
24.3.2	Code Image	284
24.3.3	Retention Data Format	285
24.4	UART Download Protocol	285
25	Electrical, Mechanical and Thermal Specifications	287
25.1	Package Information	287
25.2	Maximum Ratings and Operating Conditions.....	291
25.2.1	Absolute Maximum Ratings	291
25.2.2	Operating Conditions	292

25.3	Electrical Characteristics	293
25.3.1	DC Voltage and Current Characteristics	293
25.3.2	I/O Port Characteristics	293
25.3.3	Clock Characteristics	294
25.3.4	Power and Brownout Detection	295
25.3.5	ADC Electrical Characteristics	297
25.3.6	Analog Temperature Sensor Characteristics	300
25.3.7	ACOMP Electrical Characteristics	301
25.3.8	DAC Electrical Characteristics	302
25.3.9	AC Electrical Characteristics	304
	25.3.9.1 SSP Timing Diagram and Specifications	305
	25.3.9.2 QSPI Timing Diagram and Specifications	305
	25.3.9.3 SDIO Timing Diagram and Specifications	306
	25.3.9.4 RESETn Pin Specification	307
Appendix A: 88MC200 Microcontroller Register Tables		A-1



88MC200 Microcontroller Datasheet

List of Figures

1	Product Overview	23
	Figure 1: 88MC200 Block Diagram	24
	Figure 2: 88MC200 QFN88 Pinouts.....	27
	Figure 3: 88MC200 QFN68 Pinout	28
	Figure 4: 88MC200 Microcontroller Package Markings for 88-Pin Part.....	46
	Figure 5: Part Ordering Number	46
2	Processor Overview	49
3	I/O Configuration	51
	Figure 6: I/O Padding Structure	85
4	System Control	87
5	Power, Reset, and Clock Control	89
	Figure 7: 88MC200 MCU Power Supply Overview	89
	Figure 8: Power Mode Transitions	97
	Figure 9: High-Level Clocking Diagram	100
6	Memory Map, Interrupts and AHB Bus Fabric	107
	Figure 10: System Memory Map Diagram	108
	Figure 11: Bus Matrix Interconnection	114
7	Direct Memory Access Controller (DMA)	117
	Figure 12: DMA Block Diagram	118
	Figure 13: Breakdown of DMA Transfer into Burst Transactions.....	120
	Figure 14: Breakdown of DMA Transfer into Single and Burst Transactions.....	121
	Figure 15: Case 1 Watermark Levels where IC_DMA_TDLR = 2.....	122
	Figure 16: Case 2 Watermark Levels where IC_DMA_TDLR = 6.....	122
	Figure 17: I2C Receive FIFO	124
8	Real Time Clock (RTC)	127
	Figure 18: RTC Block Diagram	127
	Figure 19: Count-up Mode	128
9	General Purpose Timers (GPT)	131
	Figure 20: GPT Block Diagram	132
	Figure 21: Clock Source Selection.....	133
	Figure 22: Count Up Mode.....	134
	Figure 23: Input Capture	136
	Figure 24: One-Shot Pulse	137
	Figure 25: One-Shot Edge	138

Figure 26:	PWM Edge-Aligned	139
Figure 27:	PWM Center-Aligned	141
Figure 28:	ADC Trigger for (a) PWM Edge-Aligned and (b) PWM Center-Aligned	142
Figure 29:	DAC Trigger for (a) PWM Edge-Aligned and (b) PWM Center-Aligned	143
10	Secure Digital Input/Output (SDIO Controller)	145
Figure 30:	SDIO Controller Block Diagram	146
Figure 31:	Interaction of Typical SDIO System	147
11	USB OTG Interface Controller	159
Figure 32:	USB Controller Block Diagram	160
Figure 33:	End Point Queue Head Organization	162
Figure 34:	Periodic Schedule Organization	163
12	WatchDog Timer (WDT)	169
Figure 35:	Interrupt Generation	170
Figure 36:	Counter Restart and System Reset	170
13	Quad Serial Peripheral Interface (QSPI) Controller	173
Figure 37:	Block Diagram of the QSPI Controller	174
Figure 38:	Frame of Data Format for Serial Flash Access	175
Figure 39:	Non-DMA Mode Read Flow	178
Figure 40:	Non-DMA Mode Write Flow	179
Figure 41:	DMA Mode Write Flow	180
Figure 42:	DMA Read Flow	181
14	In-Package Flash	185
Figure 43:	Serial Flash Memory Block Diagram	186
Figure 44:	Status Register (1)	190
Figure 45:	Status Register (2)	190
15	General Purpose Input Output (GPIO)	197
Figure 46:	General Purpose I/O Block Diagram	197
16	Advanced Encryption Standard (AES)	201
Figure 47:	AES Operational Flow	202
17	Cyclic Redundancy Check (CRC)	207
18	Universal Asynchronous Receiver Transmitter (UART)	209
Figure 48:	Block Diagram	210
Figure 49:	Serial Data Structure	211
Figure 50:	SIR Data Format	212
Figure 51:	Auto Flow Control Block Diagram	214
Figure 52:	Auto RTS Timing	215
Figure 53:	Auto CTS Timing	215
19	Inter-Integrated Circuit (I2C)	217

Figure 54:	I2C Block Diagram	218
Figure 55:	Master/Slave and Transmitter/Receiver Relationship	219
Figure 56:	Data Transfer on the I2C Bus	220
Figure 57:	START and STOP Condition	221
Figure 58:	7-Bit Address Format	222
Figure 59:	10-Bit Address Format	222
Figure 60:	Master-Transmitter Protocol	223
Figure 61:	Master-Receive Protocol	223
Figure 62:	Multiple Master Arbitration	224
Figure 63:	Multi-Master Clock Synchronization	225
20	Synchronous Serial Protocol (SSP)	231
Figure 64:	Texas Instruments Synchronous Serial Frame Protocol (Single Transfers)	236
Figure 65:	Texas Instruments Synchronous Serial Frame Protocol (Multiple Transfers)	237
Figure 66:	Motorola SPI Frame Protocol (Single Transfers)	238
Figure 67:	Motorola SPI Frame Protocol (Multiple Transfers)	239
Figure 68:	Motorola SPI Frame Protocols for SPO and SPH Programming (SPH Set)	239
Figure 69:	Motorola SPI Frame Protocols for SPO and SPH Programming (SPH Cleared)	240
Figure 70:	Programmable Serial Protocol Format	243
Figure 71:	Programmable Protocol Format (Consecutive Transfers)	244
Figure 72:	TI SSP with SSP_SSCR1[TTE] = 1 and SSP_SSCR1[TTELP] = 0	244
Figure 73:	TI SSP with SSP_SSCR1[TTE] = 1 and SSP_SSCR1[TTELP] = 1	245
Figure 74:	Motorola* SPI with <TXD Tri-State Enable> = 1 and <TXD Tri-State Enable On Last Phase> = 0	245
Figure 75:	PSP Format with SSP_SSCR1[TTE] = 1, SSP_SSCR1[TTELP] = 0, and SP_SSCR1[SFRMDIR] = 1	246
Figure 76:	PSP Format with SSP_SSCR1[TTE] = 1, and either SSP_SSCR1[TTELP] = 1, or SSP_SSCR1[SFRMDIR] = 0247	248
Figure 77:	Network Mode (Example Using 4 Time Slots)	249
Figure 78:	Network Mode and PSP Frame Format	250
Figure 79:	Normal I2S Format	251
Figure 80:	MSB-Justified I2S Format	251
21	Analog Digital Converter (ADC)	253
Figure 81:	ADC Block Diagram	255
Figure 82:	ADC Temperature Sensor Mode with External Diode	258
22	Digital Analog Converter (DAC)	263
Figure 83:	Synchronous Mode	264
Figure 84:	Sinusoidal Waveform Generation	265
Figure 85:	Full Triangle Generation Mode	266
Figure 86:	Half Triangle Generation Mode	267
23	Analog Comparator (ACOMP)	269
Figure 87:	Comparator Hysteresis	271
Figure 88:	Comparator Output Edge Detection	273

Figure 89: Interrupt	275
24 Boot ROM	277
Figure 90: Boot ROM Flow Chart 1.....	278
Figure 91: Boot ROM Flow Chart 2.....	279
Figure 92: Boot ROM Flow Chart 3.....	280
Figure 93: Flash Image Memory Mapping	282
25 Electrical, Mechanical and Thermal Specifications.....	287
Figure 94: Mechanical Drawing for 68-pin Package	288
Figure 95: Mechanical Drawing for 88-pin Package	290
Figure 96: BOD Sequence Trigger Level.....	297
Figure 97: Digital Filter Frequency Response.....	300
Figure 98: SSP Serial Frame Format.....	305
Figure 99: QSPI Timing Diagram.....	306
Figure 100: SDIO DC Parameter: Bus Timing (PAD)	307

List of Tables

1	Product Overview	23
	Table 1: QFN68 and QFN88 Microcontroller Packages	24
	Table 2: Pin Descriptions	28
	Table 3: 88MC200 Microcontroller Part Ordering Options	47
2	Processor Overview	49
3	I/O Configuration	51
	Table 4: I/O Pin Mode Configuration	84
4	System Control	87
5	Power, Reset, and Clock Control	89
	Table 5: VDD_MCU Address Memory	90
	Table 6: 88MC200 Power Pins	90
	Table 7: I/O Power Configuration	91
	Table 8: CORTEX-M3 Core Subsystem Power Modes	93
	Table 9: SRAM Memory Power Modes	93
	Table 10: Flash Memory Power Modes	93
	Table 11: 88MC200 MCU System Power Modes	93
	Table 12: Address of Memories Available in State Retention Mode	94
	Table 13: Low Power Mode Wake Up Sources	97
	Table 14: 88MC200 Clock Sources	99
	Table 15: MCU Clock Frequency Table	101
	Table 16: APB0 Bus Clock Divider Ratio	102
	Table 17: APB1 Bus Clock Divider Ratio	102
	Table 18: UART Slow and Fast Clock Programming	102
	Table 19: USB Clock Programming	103
	Table 20: Audio Clock Programming	103
	Table 21: XTAL32K Output Bits in Register PAD_CTRL1_REG of PMU	104
6	Memory Map, Interrupts and AHB Bus Fabric	107
	Table 22: System Address Memory Map	109
	Table 23: Memory Map for On-chip SRAM	110
	Table 24: External Interrupts	110
	Table 25: GPIO Mapping to External Interrupts	111
7	Direct Memory Access Controller (DMA)	117
8	Real Time Clock (RTC)	127
	Table 26: Counter Update Mode	128
9	General Purpose Timers (GPT)	131
	Table 27: Counter Update Mode	134
	Table 28: Available Interrupt Events	134
	Table 29: One-Shot Pulse Control Registers	137

Table 30:	PWM Edge-Aligned Control Registers.....	138
Table 31:	PWM Center-Aligned Control Registers	139
10	Secure Digital Input/Output (SDIO Controller).....	145
Table 32:	SDIO Card I/O Signal Summary	145
Table 33:	Response and Data Errors	152
Table 34:	Controller-Generated Interrupts.....	152
11	USB OTG Interface Controller	159
Table 35:	USB OTG Controller Signal Descriptions	161
Table 36:	USB Host Controller Signal Descriptions.....	162
12	WatchDog Timer (WDT).....	169
13	Quad Serial Peripheral Interface (QSPI) Controller	173
Table 37:	External Interface (QSPI1).....	174
14	In-Package Flash	185
Table 38:	Status Register Protect (SRP1, SR0) Bits	189
Table 39:	Status Register Memory Protection (CMP = 0).....	191
Table 40:	Status Register Memory Protection (CMP=1).....	192
Table 41:	Instruction Set Table	194
Table 42:	Instruction Set Table 2 (Read Instructions)	195
Table 43:	Instruction Set Table 3 (ID, Security Instructions)	196
15	General Purpose Input Output (GPIO)	197
16	Advanced Encryption Standard (AES)	201
Table 44:	Padding Scheme	204
Table 45:	Error Status for Different AES Block Cipher Modes.....	204
Table 46:	AES Output Vector	205
17	Cyclic Redundancy Check (CRC).....	207
18	Universal Asynchronous Receiver Transmitter (UART)	209
Table 47:	Serial Interface Signal Descriptions.....	211
19	Inter-Integrated Circuit (I2C)	217
Table 48:	I2C Signal Descriptions	217
Table 49:	I2C Bus Terminologies	218
20	Synchronous Serial Protocol (SSP).....	231
Table 50:	SSP Interface Signal Descriptions.....	231
Table 51:	Programmable Protocol Parameters	242
21	Analog Digital Converter (ADC)	253
Table 52:	ADC Module External Interface Signals	254
Table 53:	ADC Input Configurations	256
Table 54:	Detailed Input Range	257
Table 55:	ADC Conversion Time and Throughput Rate Lookup Table	259

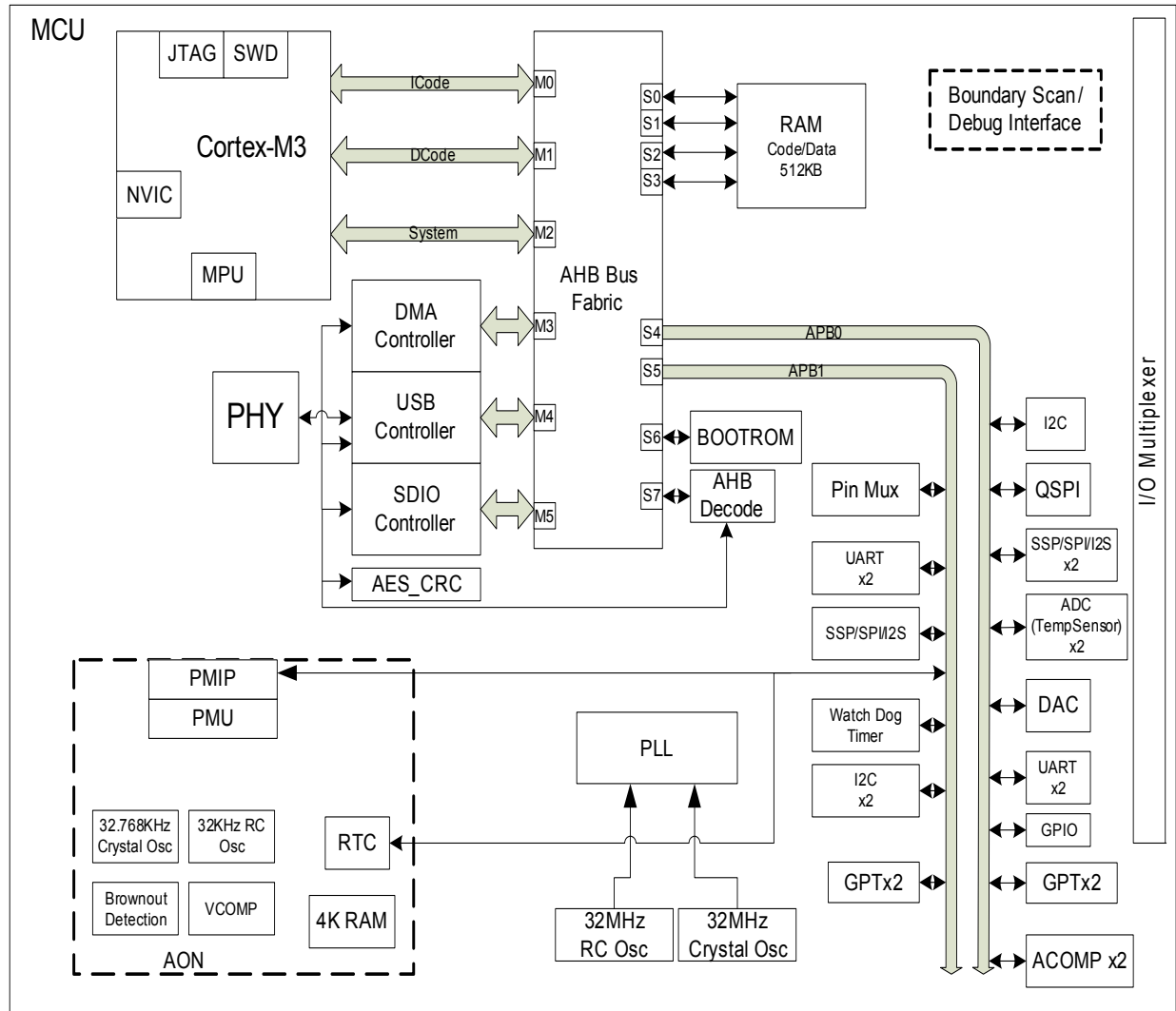
Table 56:	ADC Conversion Result Format (PGA=1 and OSR[1:0]=2'b11).....	259
Table 57:	ADC Conversion Result Format (PGA=1 and OSR[1:0]=2'b10).....	260
Table 58:	ADC Conversion Result Format (PGA=1 and OSR[1:0]=2'b01).....	260
Table 59:	ADC Conversion Result Format (PGA=1 and OSR[1:0]=2'b00).....	260
Table 60:	Equations for Gain and Offset Correction	261
22	Digital Analog Converter (DAC)	263
Table 61:	DAC External Signal Description	263
Table 62:	Output Voltage Calculation Formula	264
23	Analog Comparator (ACOMP)	269
Table 63:	ACOMP Module Interface Signals	270
24	Boot ROM	277
Table 64:	BootInfo Layout.....	282
Table 65:	Sub-Field in common Dfg0	283
Table 66:	Sub-Field in common Cfg1	284
Table 67:	Sub-Field in bootCfg0	284
Table 68:	Sub-Field for bootCfg1.....	284
Table 69:	Retention Data Structure	285
25	Electrical, Mechanical and Thermal Specifications.....	287
Table 70:	Package Information for 68-pin Package.....	288
Table 71:	Package Information for 88-pin Package (See Note under Table 70)	291
Table 72:	Absolute Maximum Ratings	291
Table 73:	Voltage, Temperature, and Frequency Electrical Specification	292
Table 74:	I/O Static Characteristics on 3.3V VDD_IO	293
Table 75:	I/O Static Characteristics on 1.8V VDD_IO	293
Table 76:	Current Consumption.....	294
Table 77:	MAINXTAL Oscillator	294
Table 78:	RC32K Electrical Characteristics	295
Table 79:	Power-on Reset	295
Table 80:	VBAT Brownout Detection (BOD) Electrical Characteristics.....	295
Table 81:	ADC Electrical Characteristics	297
Table 82:	ADC Digital Filter Electrical Characteristics	300
Table 83:	Analog Temperature Sensor Characteristics	301
Table 84:	ACOMP Electrical Characteristics	301
Table 85:	DAC Electrical Characteristics	302
Table 86:	Ultra Low-Power Comparator Electrical Characteristics	304
Table 87:	SSP Timing	305
Table 88:	QSPI Characteristics	305
Table 89:	SDIO Characteristics in Full Speed Mode	306
Table 90:	SDIO Timing Specification High-Speed Mode	307

1 Product Overview

1.1 Introduction

The 88MC200 device from Marvell is a highly integrated system-on-chip (SoC) microcontroller that features a 32-bit ARM Cortex-M3 high-performance processor with a software-programmable clock rate as high as 200 MHz, 512 KB of CODE/SRAM memory, on-chip DC-DC converter, and in-package serial flash with 8Mbits. In addition, the 88MC200 microcontroller offers a rich array of peripherals that enable a broad class of applications as shown in the block diagram ([Figure 1](#)).

Figure 1: 88MC200 Block Diagram



1.2 Features

Table 1 describes the two packages available for the 88MC200 microcontroller: QFN68 and QFN88.

Table 1: QFN68 and QFN88 Microcontroller Packages

Feature List		QFN68	QFN88
Integrated Core	Core Type	ARM Cortex M3	ARM Cortex M3
	Core Clock Maximum Freq	200 MHz	200 MHz

Table 1: QFN68 and QFN88 Microcontroller Packages (Continued)

Feature List		QFN68	QFN88
Memory	In-package Flash	8Mbits	8Mbits
	SRAM	512 KB	512 KB
	ROM	4KB	4KB
Peripherals	JTAG/SWD	Yes	Yes
	SSP/SPI/I2S	3	3
	QSPI	2	2
	I2C master/slave	2	3
	UART	3	4
	USB OTG FS	1	1
	SDIO	1	1
	External Pin IRQ	18	23
	GPIO	45	63
	ADC	Number of ADCs	2
		Number of ext channels for ADC0	4 single-ended Or 2 differential
		Number of ext channels for ADC1	4 single-ended Or 2 differential
	TempSensor	Number of Internal Sensors	2
		Number of External Sensors	2
	DAC	Number of DACs	1
		Number of ext channels per DAC	2 single-ended or 1 differential with 2 DACs combined
	ACOMP	Number of ACOMPs	2
		Number of ext channels per ACOMP	8 single-ended Or 4 differential
	General Purpose Timers		4-15 PWM channels
	Watchdog Timer		1
	RTC		1
	CRC		1
	AES		1

Table 1: QFN68 and QFN88 Microcontroller Packages (Continued)

Feature List		QFN68	QFN88
Wake-up mechanism	Wake-up thru dedicated GPIO	Yes	Yes
	Wake-up thru IRQ	PM1 mode only	PM1 mode only
	Wake-up thru RTC	Yes	Yes
Voltage Rails	Power Supply Voltage	1.8 - 3.6V	1.8 - 3.6V
	I/O Supply Voltage	1.8 - 3.6V	1.8 - 3.6V
	USB Supply Voltage	3.3V	3.3V
Power Modes	PM0 (Active)	Yes	Yes
	PM1 (Idle)	Yes	Yes
	PM2 (Standby)	Yes	Yes
	PM3 (Sleep)	Yes	Yes
	PM4 (Shut off)	Yes	Yes
Package	Packaged (trays and tape-in-reel)	QFN68 (8x8mm ²)	QFN88 (10x10 mm ²)
Temperature	Ambient Temperature	-40 to +85° C (Industrial grade) 0 to +85° C (Commercial grade)	
	Storage Temperature	-55 to +125 °C	

1.3 Pin Descriptions

1.3.1 Pinout

[Figure 2](#) shows the 88MC200 QFN88 and QFN68 pinouts. [Table 2](#) provides pin descriptions.

Figure 2: 88MC200 QFN88 Pinouts

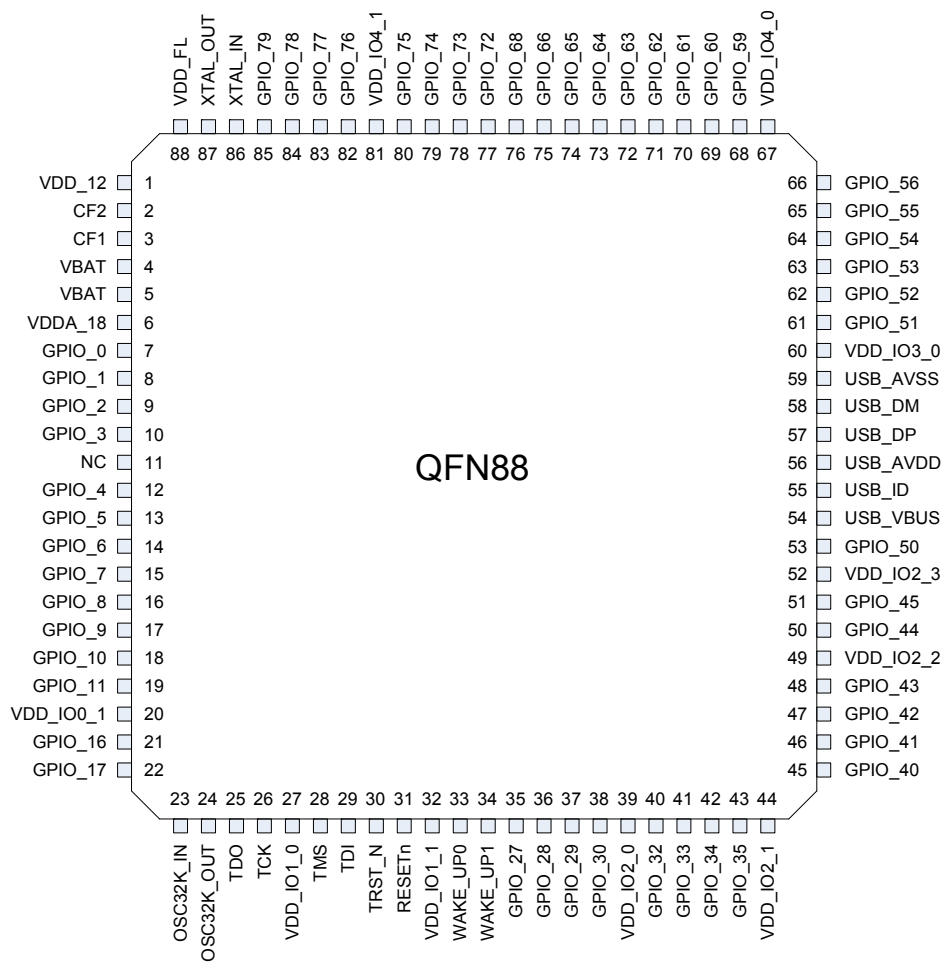


Figure 3: 88MC200 QFN68 Pinout

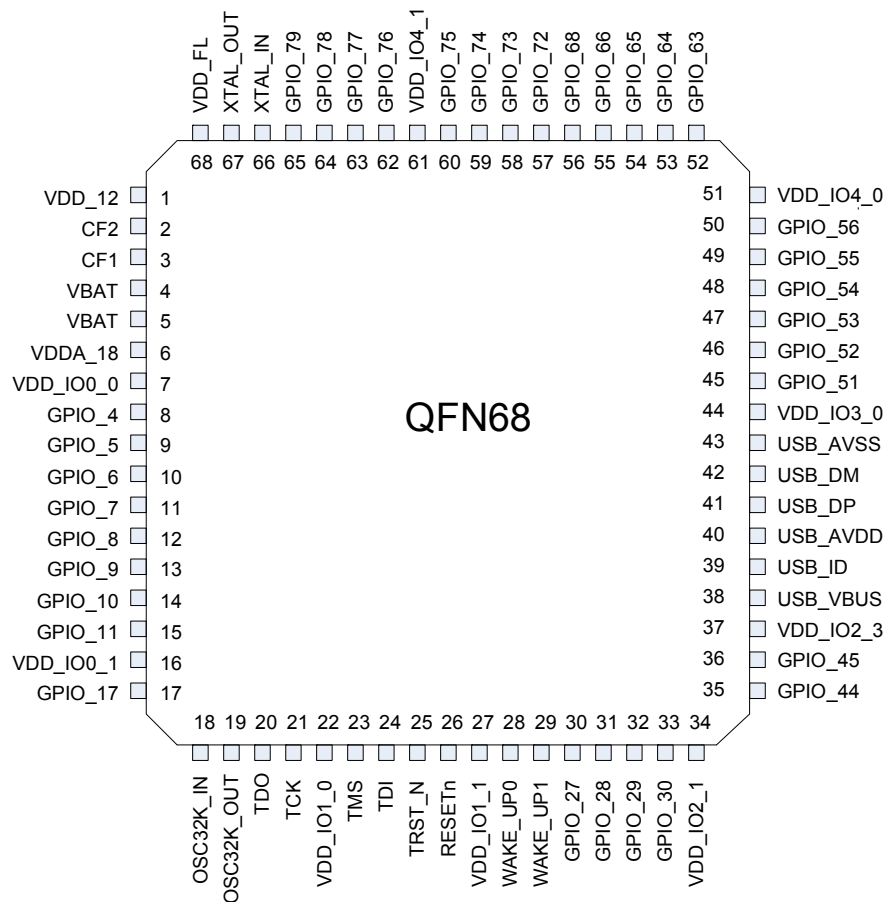


Table 2: Pin Descriptions

QFN88	QFN68	Signal	Direction	Description
1	1	VDD_12	Flycap	1.2V Fly Cap
2	2	CF2	Flycap	Capacitor connection
3	3	CF1	Flycap	Capacitor connection
4	4	VBAT	Power	1.8V~3.6V power supply connection
5	5	VBAT	Power	1.8V~3.6V power supply connection
6	6	VDDA_18	Flycap	1.8V Fly Cap

Table 2: Pin Descriptions (Continued)

QFN88	QFN68	Signal	Direction	Description
7		GPIO_0	I/O	Digital IO #0
		ADC0_IN7/ACOMP0_IN7 /ACOMP1_IN7	AI/O	ADC0 Channel 7 or ACOMP0/1 Channel 7
		GPT0_CH0	I/O	GPT0 Channel 0
8		GPIO_1	I/O	Digital IO #1
		ADC0_IN6/ACOMP0_IN6 /ACOMP1_IN6	AI/O	ADC0 Channel 6 or ACOMP0/1 Channel 6
		GPT0_CH1	I/O	GPT0 Channel 1
9		GPIO_2	I/O	Digital IO #2
		ADC0_IN5/ACOMP0_IN5 /ACOMP1_IN5	AI/O	ADC0 Channel 5 or ACOMP0/1 Channel 5
		GPT0_CH2	I/O	GPT0 Channel 2
10		GPIO_3	I/O	Digital IO #3
		ADC0_IN4/ACOMP0_IN4 /ACOMP1_IN4	AI	ADC0 Channel 4 or ACOMP0/1 Channel 4
		GPT0_CH3	I/O	GPT0 Channel 3
	7	VDD_IO0_0	power	IO supply
12	8	GPIO_4	I/O	Digital IO #4
		ADC0_IN3/ACOMP0_IN3 /ACOMP1_IN3/DACA/ADC0_REF	AI/O	ADC0 Channel 3 or ACOMP0/1 Channel 3 or DAC Channel A or ADC0 Vref
		GPT0_CH4	I/O	GPT0 Channel 4
		I2C1_SDA	I/O	SDA for I2C1
		GPT1_CLKIN	I	GPT1 clock in
13	9	GPIO_5	I/O	Digital IO #5
		ADC0_IN2/ACOMP0_IN2 /ACOMP1_IN2	AI/O	ADC0 Channel 2 or ACOMP0/1 Channel 2
		GPT0_CH5	I/O	GPT0 Channel 5
		I2C1_SCL	I/O	SCL for I2C1
		GPT3_CLKIN	I	GPT3 clock in
14	10	GPIO_6	I/O	Digital IO #6
		ADC0_IN1/ACOMP0_IN1 /ACOMP1_IN1	AI/O	ADC0 Channel 1 or ACOMP0/1 Channel 1
		GPT1_CH0	I/O	GPT1 Channel 0
		GPT0_CLKIN	I	GPT0 clock in
		GPT3_CH0	I/O	GPT3 Channel 0

Table 2: Pin Descriptions (Continued)

QFN88	QFN68	Signal	Direction	Description
15	11	GPIO_7	I/O	Digital IO #7
		ADC0_IN0/ACOMP0_IN0 /ACOMP1_IN0	AI/O	ADC0 Channel 0 or ACOMP0/1 Channel 0
		GPT1_CH1	I/O	GPT1 Channel 1
		GPT2_CLKIN	I	GPT2 clock in
		GPT3_CH1	I/O	GPT3 Channel 1
16	12	GPIO_8	I/O	Digital IO #8
		ADC1_IN0	AI/O	ADC1 Channel 0
		GPT1_CH2	I/O	GPT1 Channel 2
		I2C1_SDA	I	SDA for I2C1
		GPT3_CH2	I/O	GPT3 Channel 2
17	13	GPIO_9	I/O	Digital IO #9
		ADC1_IN1	AI	ADC1 Channel 1
		GPT1_CH3	I/O	GPT1 Channel 3
		I2C1_SCL	I/O	SCL for I2C1
		GPT3_CH3	I/O	GPT3 Channel 3
18	14	GPIO_10	I/O	Digital IO #10
		ADC1_IN2 /DAC_REF	AI/O	ADC1 Channel 2 or DAC Vref
		GPT1_CH4	I/O	GPT1 Channel 4
		I2C2_SDA	I/O	SDA for I2C2
		GPT3_CH4	I/O	GPT3 Channel 4
19	15	GPIO_11	I/O	Digital IO #11
		ADC1_IN3/DACB/ADC1_VREF	AI/O	ADC1 Channel 3 or DAC Channel B or ADC1 Vref
		GPT1_CH5	I/O	GPT1 Channel 5
		I2C2_SCL	I/O	SCL for I2C2
		GPT3_CH5	I/O	GPT3 Channel 5
20	16	VDD_IO0_1	PWR	IO POWER
		GPIO_12	I/O	Digital IO #12
21		GPIO_16	I/O	Digital IO #16
		GPT2_CH4	I/O	GPT2 Channel 4
		GPT3_CH0	I/O	GPT3 Channel 0
		GPT0_CH4	I/O	GPT0 Channel 4

Table 2: Pin Descriptions (Continued)

QFN88	QFN68	Signal	Direction	Description
22	17	GPIO_17	I/O	Digital IO #17
		GPT1_CH0	I/O	GPT1 Channel 0
		GPT2_CH5	I/O	GPT2 Channel 5
		GPT3_CH1	I/O	GPT3 Channel 1
		GPT0_CH5	I/O	GPT0 Channel 5
23	18	OSC32K_IN	AI	32.768 MHz crystal oscillator input
		GPIO_18	I/O	Digital IO #18
		GPT3_CH0	I/O	GPT3 Channel 0
		UART1_SIR_OUT	O	SIR_OUT fro UART0
		I2C0_SDA	I/O	SDA for I2C0
24	19	OSC32K_OUT	AO	32.768 MHz crystal oscillator output
		GPIO_19	I/O	Digital IO #19
		GPT3_CH1	I/O	GPT3 Channel 1
		UART1_SIR_IN	I	SIR_IN for UART1
		I2C0_SCL	I/O	SCL for I2C0
25	20	TDO	O	TDO fro JTAG
		GPIO_20	I/O	Digital IO #20
26	21	TCK	O	TCK for JTAG
		GPIO_21	I/O	Digital IO #21
27	22	VDD_IO1_0	PWR	IO POWER
28	23	TMS	I/O	TMS for JTAG
		GPIO_22	I/O	Digital IO #22
29	24	TDI	I	TDI for JTAG
		GPIO_23	I/O	Digital IO #23
30	25	TRST_N	I	TRSTn for JTAG
		GPIO_24	I/O	Digital IO #24
31	26	RESETn	I	Active low chip reset
32	27	VDD_IO1_1	PWR	IO POWER

Table 2: Pin Descriptions (Continued)

QFN88	QFN68	Signal	Direction	Description
33	28	WAKE_UP0	I	Wake up signal
		GPIO25	I/O	Digital IO #25
		ACOMP0_GPIO_OUT	O	ACOMP0 output synchronous or asynchronous level signals
		ACOMP1_GPIO_OUT	O	ACOMP1 output synchronous or asynchronous level signals
		UART_SIR_IN	I	SIR_IN for UART0
		32K_CLKOUT	O	32 kHz clock out
34	29	WAKE_UP1	I	Wake up signal
		GPIO_26	I/O	Digital IO #26
		ACOMP0_EDGE_PULSE	O	Output pulse aligned with synchronized comparison result
		ACOMP1_EDGE_PULSE	O	Output pulse aligned with synchronized comparison result
		UART0_SIR_OUT	O	SIR_OUT for UART0
		COMP_IN_N	AI	Negative input to AON domain comparator
		32K_CLKOUT	O	32 kHz clock out
35	30	GPIO_27	I/O	Digital IO #27
		ACOMP0_GPIO_OUT	O	ACOMP0 output synchronous or asynchronous level signals
		GPT3_CH2	I/O	Timer3 Channel 2
		UART0_DSRn	I	DSRn for UART0
		BOOT	AI	Boot pin
		COMP_IN_P	AI	Positive input to AON domain comparator
		32K_CLKOUT	O	32 kHz clock out
36	31	GPIO_28	I/O	Digital IO #28
		ACOMP0_EDGE_PULSE	O	ACOMP0 output synchronous or asynchronous level signals
		GPT3_CH3	I/O	GPT3 Channel 3
		UART0_DCDn	I	DCDn for UART0
		SDIO_LED	O	LED for SDIO

Table 2: Pin Descriptions (Continued)

QFN88	QFN68	Signal	Direction	Description
37	32	GPIO_29	I/O	Digital IO #29
		ACOMP1_GPIO_OUT	O	ACOMP1 output synchronous or asynchronous level signals
		GPT3_CH4	I/O	GPT3 Channel 4
		ACOMP0_GPIO_OUT	O	ACOMP0 output synchronous or asynchronous level signals
		UART0_Rin	I	Rin for UART0
		SDIO_CDn	I	SDIO card detect signal
38	33	GPIO_30	I/O	Digital IO #30
		ACOMP1_EDGE_PULSE	O	Output pulse aligned with Synchronized comparison result
		GPT3_CH5	I/O	GPT3 Channel 5
		ACOMP0_EDGE_PULSE	O	Output pulse aligned with Synchronized comparison result
		UART0_DTRn	O	DTRn for UART0
		SDIO_WP	I	Write protect for SDIO
39		VDD_IO2_0	PWR	IO POWER
40		GPIO_32	I/O	Digital IO #32
		SSP0_CLK	I/O	Clock for SSP0
		UART2_CTSn	I	CTS _n for UART2
		GPT2_CH0	I/O	GPT2 Channel 0
		GPT0_CH0	I/O	GPT0 Channel 0
41		GPIO_33	I/O	Digital IO #33
		SSP0_FRM	I/O	Frame for SSP0
		UART2_RTSn	O	RTSn for UART2
		GPT2_CH1	I/O	GPT2 Channel 1
		GPT0_CH1	I/O	GPT0 Channel 1
42		GPIO_34	I/O	Digital IO #34
		SSP0_RXD	I	RXD for SSP0
		UART2_TXD	O	TXD for UART2
		GPT2_CH2	I/O	GPT2 Channel 2
		GPT0_CH2	I/O	GPT0 Channel 2

Table 2: Pin Descriptions (Continued)

QFN88	QFN68	Signal	Direction	Description
43		GPIO_35	I/O	Digital IO #35
		SSP0_TXD	O	TXD for SSP0
		UART2_RXD	I	RXD for UART2
		GPT2_CH3	I/O	GPT2 Channel 3
		GPT0_CH3	I/O	GPT0 Channel 3
44	34	VDD_IO2_1	power	IO supply
45		GPIO_40	I/O	Digital IO #40
		UART3_CTSn	I	CTS _n for UART3
		SSP2_CLK	I/O	Clock for SSP2
		GPT1_CH2	I/O	GPT1 Channel 2
46		GPIO_41	I/O	Digital IO #41
		UART3_RTSn	O	RTS _n for UART3
		SSP2_FRM	I/O	Frame for SSP2
		GPT1_CH3	I/O	GPT1 Channel 3
47		GPIO_42	I/O	Digital IO #42
		UART3_TXD	O	TXD for UART3
		SSP2_RXD	I	RXD for SSP2
		GPT1_CH4	I/O	GPT1 Channel 4
48		GPIO_43	I/O	Digital IO #43
		UART3_RXD	I	RXD for UART3
		SSP2_TXD	O	TXD for SSP2
		GPT1_CH5	I/O	GPT1 Channel 5
49		VDD_IO2_2	PWR	IO POWER
50	35	GPIO_44	I/O	Digital IO #44
		I2C0_SDA	I/O	SDA for I2C0
		GPT0_CLKIN	I	GPT0 clock in
		GPT3_CH0	I/O	GPT3 Channel 0
		ADC_DAC_TRIGGER	I/O	ADC/DAC external trigger
		SDIO_CDn	I	SDIO card detect signal
51	36	GPIO_45	I/O	Digital IO #45
		I2C0_SCL	I/O	SCL fro I2C0
		GPT1_CLKIN	I	GPT1 clock in
		ADC_DAC_TRIGGER	I	ADC/DAC external trigger
		USB2_DRWBUS	O	Drive VBUS to 5 V
		SDIO_WP	I	SDIO write protect

Table 2: Pin Descriptions (Continued)

QFN88	QFN68	Signal	Direction	Description
52	37	VDD_IO2_3	PWR	IO POWER
53		GPIO_50	I/O	Digital IO #50
		GPT1_CH5	I/O	GPT1 Channel 5
		SDIO_LED	O	LED for SDIO
54	38	USB_VBUS	AI/O	VBUS selection input in device mode; unused in host mode; input/output for OTG mode to supply +5V@10mA during session negotiation
55	39	USB_ID	AI	USB2 OTG IDPIN pad
56	40	USB_AVDD	AI	3.3V voltage source for analog
57	41	USB_DP	AI/O	USB2 D+ pad
		GPIO_57	I/O	Digital IO #57
		GPT0_CLKIN	I	GPT0 clock in
		UART3_SIR_OUT	O	SIR_OUT for UART3
58	42	USB_DM	AI/O	USB2 D- pad
		GPIO_58	I/O	Digital IO #58
		GPT1_CLKIN	I	GPT1 clock in
		UART3_SIR_IN	I	SIR_IN for UART3
59	43	USB_AVSS	AI	Analog ground pad
60	44	VDD_IO3_0	PWR	IO POWER
61	45	GPIO_51	I/O	Digital IO #51
		SDIO_CLK	O	CLK for SDIO
		SSP2_CLK	O	CLK for SSP2
		GPT0_CH0	I/O	GPT0 Channel 0
		UART2_DSRRn	I/O	UART2 DSTn
62	46	GPIO_52	I/O	Digital IO #52
		SDIO_3	I/O	Data 3 for SDIO
		SSP2_FRM	I/O	Frame for SSP2
		GPT0_CH1	I/O	GPT0 Channel 1
		UART2_DCDn	I	DCDn for UART2
63	47	GPIO_53	I/O	Digital IO #53
		SDIO_2	I/O	Data2 for SDIO
		SSP2_RXD	I	RXD for SSP2
		GPT0_CH2	I/O	GPT0 Channel 2
		UART2_Rin	I	Rin for UART2

Table 2: Pin Descriptions (Continued)

QFN88	QFN68	Signal	Direction	Description
64	48	GPIO_54	I/O	Digital IO #54
		SDIO_1	I/O	Data1 for SDIO
		SSP2_TXD	O	TXD for SSP2
		GPT0_CH3	I/O	GPT0 Channel 3
		UART2_DTRn	O	DTRn for UART2
65	49	GPIO_55	I/O	Digital IO #55
		SDIO_0	I/O	Data0 from SDIO
		GPT2_CLKIN	I	GPT2 clock in
		GPT0_CH4	I/O	GPT0 Channel 4
		UART2_SIR_OUT	O	SIR_OUT for UART2
66	50	GPIO_56	I/O	Digital IO #56
		SDIO_CMD	I/O	CMD for SDIO
		GPT3_CLKIN	I	GPT3 clock in
		GPT0_CH5	I/O	Timer0 Channel 5
		UART2_SIR_IN	I	SIR_IN for UART2
67	51	VDD_IO4_0	PWR	IO POWER
68		GPIO_59	I/O	Digital IO #59
		UART1_CTSn	I	CTS _n for UART1
		GPT3_CH2	I/O	GPT3 Channel 2
		UART3_DSRn	I	DSR _n for UART3
69		GPIO_60	I/O	Digital IO #60
		UART1_RTSn	O	RTS _n for UART1
		GPT3_CH3	I/O	GPT3 Channel 3
		UART3_DCDn	I	DCD _n for UART3
70		GPIO_61	I/O	Digital IO #61
		UART1_TXD	O	TXD for UART1
		GPT3_CH4	I/O	GPT3 Channel 4
		UART3_Rin	I	Rin for UART3
71		GPIO_62	I/O	Digital IO #62
		UART1_RXD	I	RXD for UART1
		GPT3_CH5	I/O	GPT3 Channel 5
		UART3_DTRn	O	DTR _n for UART3

Table 2: Pin Descriptions (Continued)

QFN88	QFN68	Signal	Direction	Description
72	52	GPIO_63	I/O	Digital IO #63
		UART1_CTSn	I	CTSn for UART1
		SSP1_CLK	I/O	Clock for SSP1
		GPT3_CH2	I/O	GPT3 Channel 2
		UART1_DSRn	I	DSRn for UART1
73	53	GPIO_64	I/O	Digital IO #64
		UART1_RTSn	O	RTSn for UART1
		SSP1_FRM	I/O	Frame for SSP0
		GPT3_CH3	I/O	GPT3 Channel 3
		UART1_DCDn	I	DCDn for UART1
74	54	GPIO_65	I/O	Digital IO #65
		UART1_TXD	O	TXD for UART1
		SSP1_RXD	I	RXD for SSP1
		GPT3_CH4	I/O	GPT3 Channel 4
		UART1_Rin	I	Rin for UART1
75	55	GPIO_66	I/O	Digital IO #66
		UART1_RXD	I	RXD for UART1
		SSP1_TXD	O	TXD for SSP0
		GPT3_CH5	I/O	GPT3 Channel 5
		UART1_DTRn	O	DTRn for UART1
76	56	GPIO_68	I/O	Digital IO #68
		GPT2_CH2	I/O	GPT2 Channel 2
		GPT1_CLKIN	I	GPT1 clock in
77	57	GPIO_72	I/O	Digital IO #72
		UART0_CTSn	I	CTSn for UART0
		GPT2_CLKIN	I	GPT2 clock in
		GPT1_CH2	I/O	GPT1 Channel 2
		QSPI1_SSn	O	Frame for QSPI1
78	58	GPIO_73	I/O	Digital IO #73
		UART0_RTSn	O	RTSn for UART0
		GPT3_CLKIN	I	GPT3 clock in
		GPT1_CH3	I/O	GPT1 Channel 3
		QSPI1_CLK	O	Clock for QSPI1

Table 2: Pin Descriptions (Continued)

QFN88	QFN68	Signal	Direction	Description
79	59	GPIO_74	I/O	Digital IO #74
		UART0_TXD	O	TXD for UART0
		GPT1_CH4	I/O	GPT1 Channel 4
		RC32M_CLKOUT	O	RC32M clock out
80	60	GPIO_75	I/O	Digital IO #75
		UART0_RXD	I	RXD for UART0
		GPT1_CH5	I/O	GPT1 Channel 5
81	61	VDD_IO4_1	PWR	IO supply
82	62	GPIO_76	I/O	Digital IO #76
		UART2_CTSn	I	CTS _n for UART2
		SSP0_CLK	I/O	Clock for SSP0
		I2C0_SDA	I/O	SDA for I2C0
		QSPI1_D0	I/O	Data0 for QSPI1
83	63	GPIO_77	I/O	Digital IO #77
		UART2_RTSn	O	RST _n for UART2
		SSP0_FRM	I/O	Frame for SSP0
		I2C0_SCL	I/O	SCL for I2C0
		QSPI1_D1	I/O	Data1 for QSPI1
84	64	GPIO_78	I/O	Digital IO #78
		UART2_TXD	O	Output data from UART2
		SSP0_RXD	I	Input data to SSP0
		GPT1_CH0	I/O	GPT1 Channel 0
		QSPI1_D2	I/O	Data2 for QSPI1
85	65	GPIO_79	I/O	Digital IO #79
		UART2_RXD	I	Input data to UART2
		SSP0_TXD	O	Output data from SSP0
		GPT1_CH1	I/O	GPT1 Channel 1
		QSPI1_D3	I/O	Data3 for QSPI1
86	66	XTAL_IN	AI	Crystal oscillator input
87	67	XTAL_OUT	AO	Crystal oscillator output
88	68	VDD_FL	Flycap	Pin for adding decoupling caps for the internally generated power of the in-package flash

1.4 Feature Descriptions

1.4.1 ARM Cortex-M3 CPU Core

The 88MC200 device integrates a full-feature ARM Cortex-M3 CPU core that can operate as high as 200 MHz of clock frequency. The NVIC module accepts as many as 64 external interrupts with 16 priority levels. Full debug support comes through DWJ-DP.

1.4.2 Embedded SRAM

The 88MC200 device embeds 512 KB of CODE/DATA SRAM memory, which consists of four segments: RAM0/1/2/3. In addition, the 88MC200 microcontroller supports 4 KB SRAM in an AON domain. The 4 KB SRAM memory is retained even in shut-off power mode.

1.4.3 In-Package Flash

The 88MC200 device is integrated with 8 Mbits of in-package serial flash memory. The features include:

- Total 1 Mbyte flash memory
- 200 Mbps maximum serial data rate in Quad mode with 50 MHz functional clock
- Write protect all or portions of Flash memory
- Sector erase (4KB) and Block erase (32 or 64 KB)
- Page program up to 256 bytes
- QSPI0 interface is dedicated for access to serial Flash

1.4.4 Boot ROM

The internal ROM memory is used to store the boot code. After a reset, the ARM processor begins code execution from this ROM. Features include:

- 4KB size Boot ROM
- Supports boot from in-package flash, UART, and JTAG
- Supports CRC check
- Supports secure code update

1.4.5 AHB Bus Matrix

The low-latency 88MC200 AHB Bus Matrix enables parallel access a number of shared AHB slaves from many AHB masters.

The AHB Bus Matrix supports six masters (ICODE, DCODE and SYSTEM bus from Cortex-M3, DMA, USB and SDIO). The slaves could be BOOTROM, RAM0, RAM1, RAM2, RAM3, APB0, APB1 etc.

1.4.6 Power, Reset and Clock Control

In the 88MC200 microcontroller, the power supply, power mode, on-chip DC-DC converter, clocking, reset and wake-up signals are managed by the Power Management Unit (PMU), which is in the Always-ON (AON) power domain.

1.4.7 Direct Memory Access (DMA)

The 88MC200 microcontroller includes a Direct Memory Access (DMA) module, which provides the data transfer without the interference of the CPU, keeping CPU resources free for other operations. The DMA module has eight channels and can perform memory-to-memory, peripheral-to-memory, memory-to-peripheral, and peripheral-to-peripheral transactions.

- Eight independent DMA channels with programmable channel priority
- Peripheral devices can request a DMA transfer through hardware or software handshaking interface
- One FIFO per channel for source and destination. FIFO depth is 4 x 32 bits
- Programmable source and destination addresses, with increment, decrement, or no change mode
- Maximum burst transaction size is 16, and maximum block size in source transfer is 1023
- Five interrupt sources with flags:
 - Block Transfer Complete Interrupt
 - Destination Transaction Complete Interrupt
 - Error Interrupt
 - Source Transaction Complete Interrupt
 - DMA Transfer Complete Interrupt

1.4.8 General Purpose IO (GPIO)

The 88MC200 device can be configured to support as many as 63 multi-purpose GPIO pins. Each GPIO pin can be configured by software as an input or output.

Each GPIO function is muxed with other on-chip peripherals/modules through the pin-mux module, which allows only one alternate function connected to an I/O pin.

The features include:

- As many as 63 GPIO pins
- Programmable control for GPIO pad configuration
 - Pullup, pulldown or tri-state
 - 4mA current sink/source capability for 3.3V I/O supply and 2mA for 1.8V I/O supply
- Programmable control for GPIO interrupt
 - Interrupt generation masking
 - Edge-triggered on rising, falling, or both
- External Pin Interrupt (IRQ)
- Highly flexible pin muxing allows use as GPIO or one of several peripheral functions:
 - JTAG for Cortex-M3
 - GPT
 - UART
 - I2C
 - QSPI
 - I2C
 - SSP
 - USB
 - SDIO
 - ADC/DAC/Analog comparator

1.4.9 Watchdog Timer (WDT)

The 88MC200 Watchdog Timer increases application reliability by regaining control of the system when a failure occurs because of software errors. The WDT can generate a reset or an interrupt when the counter reaches a given time-out value.

The 88MC200 WDT supports the following features:

- WDT module receives clock from APB clock
- 32-bit down counter with the minimal time-out value of 65536
- Configurable reset or interrupt generation with the given time-out value
- Supports eight types of reset pulse length

1.4.10 Real Time Clock (RTC)

The 88MC200 RTC is an independent hardware timer, providing a continuously running counter that can be supplied to a customer clock-calendar and timer interrupt. The RTC is optimized in the AON domain and can wake up the CPU core automatically from low-power modes. The main features include:

- 32-bit up counter with a programmable upper overflow boundary
- Interrupt is generated when it reaches the upper boundary
- Selectable clock source
 - Internal RC32K clock
 - External crystal oscillator 32.768 kHz

1.4.11 General Purpose Timers

The 88MC200 microcontroller includes four identical 32-bit general-purpose timers (GPT). Each GPT consists of a 32-bit up counter with a programmable prescaler. Each GPT can implement the functions of input capture, output compare, and PWM waveforms, which are widely used for a variety of purposes.

GPT main features include:

- Selectable clock source
- A 32-bit up counter with programmable clock divider and pre-scalar
- Six channels per timer with multiple modes
 - Input capture for external inputs
 - “One-shot” mode to trigger a one-time output change and interrupt
 - Edge-aligned and Center-aligned pulse-width modulation (PWM)
- Support auto-trigger ADC/DAC module for PWM mode
- Support DMA transfer for input capture
- Interrupt generation on counter and channel events

1.4.12 Advanced Encryption Standard (AES) Engine

The AES engine provides fast and energy-efficient hardware encryption and decryption services. The AES engine supports ECB, CBC, CTR, CCM, and MMO mode. The key length can be as many as 256 bits.

To save system power and bus bandwidth, the AES supports DMA transfer. The main features of the AES engine are as follows:

- Supports as many as six block cipher modes: ECB, CBC, CTR, CCM*, MMO, and Bypass
- 128, 192, and 256 bits Key Size
- Partial Code Supports
 - CCM*: Automatic padding 0 for both A string and M string
 - MMO: Automatic padding “100...00”+2 byte length information
 - CBC: Cipher stealing is performed to partial codeword
 - CTR: Partial code word does not affect the operation
 - ECB: Check partial case, assert error when partial cases detected

- Data IO includes Register Interface and DMA
- Interrupt on completed AES operation (Output FIFO is empty and Input FIFO is full)
- Error indications for each block cipher mode
- Separate 4*32-bit input and output FIFOs
- Special feature for Security Mode
 - CTR: Supports counter modular from 16 to 128
 - CCM*: Supports 0 to $(2^{32}-1)$ bytes associated string and message string
 - Supports 11-13 bytes nonce and supports L from 2 to 4

1.4.13 Cyclic Redundancy Check (CRC)

A Cyclic Redundancy Check (CRC) or polynomial code checksum is a hash function designed to detect accidental changes to raw computer data, and is used to verify data transmission or storage integrity. The CRC hardware module supports several CRC standards commonly used and generates up to 32-bit CRC code for error detection.

The main features of the CRC module include:

- 32-bit parallel bit stream input, and up to 32-bit CRC output
- Supports up to 2^{32} (4292967296) byte length for CRC calculation
- CRC standard polynomials:
 - CRC-16-CCITT($x^{16}+x^{12}+x^5+1$)
 - CRC-16-IBM($x^{16}+x^{15}+x^2+1$)
 - CRC-16-T10-DIF($x^{16}+x^{15}+x^{11}+x^9+x^8+x^7+x^5+x^4+x^2+x+1$)
 - CRC-32-IEEE 802.3 ($x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$)

1.4.14 General Purpose ADC

Two ADCs (ADC0 and ADC1) are integrated in the 88MC200 device. Each ADC is a second-order sigma-delta converter with up to 16-bit resolution. ADC conversion results are stored to RAM through DMA. Each ADC features:

- Selectable decimation rates with corresponding effective resolution of 10 to 16 bits
- Highest throughput rate with 10-bit resolution is 250k sample/s
- As many as four differential channels or eight single-ended channels
- Flexible gain buffer setting of 2x, 1x and 0.5x
- Reference voltage can be set from internal, external, or Vref_1.8

1.4.15 Analog Comparators

Two analog comparators, COMP1 and COMP0, are designed to have true rail-to-rail inputs and operate over the full voltage range of Vbat (2.0V to 3.6V). Each comparator compares two analog signals and returns a digital value indicating which input voltage is higher. Features include:

- Eight selectable external positive inputs
- Eight selectable external negative inputs
- Two selectable internal positive inputs
 - DACA output
 - DACB output
- Five selectable internal negative inputs
 - DACA output
 - DACB output

- VBAT scaled by four selectable factors
- Internal 1.2V
- VSSA
- Selectable hysteresis, eight levels between 0 and $\pm 70\text{mV}$
- Selectable response time as fast as 110ns
- Synchronous interrupt generation on rising edge and/or falling edge
- Can operate in “Sleep” mode with extremely low power consumption
- Comparator output on GPIOs through alternate functionality, output inversion available

1.4.16 DAC

The DAC module has 10-bit resolution. It can be configured to two single-ended channels or one differential channel. DMA mode is supported. The main features also include:

- Throughput rate as high as 500k sample/s
- Capable of directly driving piezo speaker with 1,000-ohm load
- Built-in waveform generators - sinusoidal, triangle, noise, etc, at various frequencies
- Reference voltage selectable from internal or external

1.4.17 UART

Four UART devices are integrated in the 88MC200 microcontroller to communicate with an external host or devices, with features that include:

- Programmable FIFO access mode for 16 x 8 bits Transmit and Receive FIFO
- DMA support
- Auto flow control
- Programmable data format:
 - 5 to 8 data bits plus parity
 - Odd, even, no parity
 - One, one-and-a-half, or 2 stop bits
- Six interrupt types with flags:
 - Receiver line status
 - Receiver Data Available
 - Character Timeout (in FIFO mode only)
 - Transmitter Holding Register Empty or FIFO at/below threshold (Programmable interrupt mode enable)
 - Modem Status
 - Busy Detect Indication
- 7 additional shadow registers to be used to reduce the software overhead

1.4.18 I²C

The I²C bus interface complies with the common I2C (I²C) protocol and can operate in standard mode (with data rates up to 100 Kb/s), fast mode (with data rates up to 400 Kb/s). Additionally, fast mode devices are downward compatible. It also supports DMA capability.

- Three I²C serial interfaces – consists of a serial data line (SDA) and a serial clock (SCL)
- Three speeds:
 - Standard mode (100 Kb/s)
 - Fast mode (400 Kb/s)

- High-Speed mode (2Mb/s)
- Clock synchronization
- Master or slave I²C operation, multi-master, multi-slave operation, and arbitration support
- 7- or 10-bit addressing
- 7- or 10-bit combined format transfers
- Bulk transmit mode in slave
- 16*32 bits deep transmit and receive buffers, respectively
- Interrupt operation
- DMA function support

1.4.19 QSPI Interface

The 88MC200 device integrates two QSPI interfaces. The QSPI interface is a synchronous controller that is connected to a serial interface complied with SPI protocol. The slave supports a serial bit rate as fast as 200 Mbps with a functional clock of 50 MHz. Two FIFOs are both eight samples deep*32 bits.

The main features of the QSPI are:

- Directly supports SPI protocol
- One FIFO for Transmit data (TXFIFO) and a second, independent, FIFO for Receive data (RXFIFO); both eight locations deep* 32 bits wide
- Single, Dual and Quad mode supported
- 200 Mbps maximum serial data rate in Quad mode as provided 50 MHz functional clock
- Master mode operation supported

1.4.20 SSP

The SSP port is a synchronous serial controller that can be connected to a variety of external Analog-to-Digital converters (ADC), audio and telecommunication CODECs, and many other devices that use serial protocols for data transfer. The SSP ports are configurable to operate in Master mode (the attached peripheral functions as a slave) or Slave mode (the attached peripheral functions as a master). The SSP ports support serial bit rates from 6.3 Kbps (minimum recommended speed) up to 25 Mbps. Serial data sample size can be set to 8, 16, 18, or 32 bits in length. A FIFO is provided for Transmit data and a second independent FIFO is provided for Receive data. The two FIFOs are both 16 samples deep x 32 bits wide or both 32 samples deep x 16 bits wide. The FIFOs can be loaded or emptied by the Cortex M3 Processor using programmed I/O (PIO) or by DMA burst transfers.

- Directly supports Texas Instruments* Synchronous Serial Protocol (SSP), and Motorola* Serial Peripheral Interface (SPI). The I²S protocol is supported by programming the PSP; data sample sizes can be set to 8, 16, 18, or 32 bits
- One FIFO for Transmit data (TXFIFO) and a 2nd, independent, FIFO for Receive data (RXFIFO); for non-packed data mode, the two FIFOs are each 16 rows deep x 32 bits wide for a total of 16 samples FIFO packed mode allows double depth FIFOs if the samples are 8 bits or 16 bits wide; for packed data mode, both FIFOs are 32 locations deep x 16 bits wide for a total of 32 samples
- 25 Mbps maximum serial bit-rate
- Master mode and Slave mode operation supported
- Receive-without-Transmit operation
- Network mode with up to eight time slots for PSP formats, and independent Transmit/Receive in any/all/none of the time slots

1.4.21 SDIO

The SDIO Controller supports the Secure Digital I/O communication protocol. The Host Controller handles SDIO Protocol at transmission level, acknowledging data, adding cyclic redundancy check (CRC), start/end bit, and checking for transaction format correctness. The SDIO module in the controller supports one SDIO card based on the standards outlined in the SDIO Card Specification Version 2.0.

- Meets SDIO card specification Version 2.0
- Card Detection (Insertion / Removal)
- Password protection of cards
- Host clock rate variable up to 50 MHz
- Supports 1-bit, 4-bit SDIO modes
- Allows card to interrupt host in 1-bit, 4-bit SDIO modes
- Up to 100 Mbps read and write rates using 4 parallel data lines (SD 4-bit mode)
- CRC7 for command and CRC16 for data integrity
- Designed to work with I/O cards, read-only cards and read/write cards
- Supports Read Wait Control, Suspend/Resume operation
- Supports FIFO Overrun and Underrun condition by stopping functional clock

1.4.22 USB

The USB OTG-capable dual-role host/device controller is compliant with the USB 2.0 specification.

- Full USB OTG functionality with integrated transceiver, allowing support for an Enhanced Host Controller Interface (EHCI) host or a device
- Support Full-Speed/Low-Speed USB 2.0 Host/Device/OTG modes
- As many as 16 configurable bi-directional endpoints for Device mode
 - Transfer types supported: Control, Interrupt, Bulk, Isochronous
 - Endpoint 0: Dedicated for the control of endpoint
- Control signals for external power supply and detection of voltages for OTG signalling
- Capability to respond as self- or bus-powered device and control to allow charging from bus
- Full 1 KB Transmit FIFOs for each endpoint
- 2 KB shared Receive buffer for all incoming data

1.5 Part Ordering

The 88MC200 microcontroller is offered in several QFN packages. [Figure 4](#) shows the laser marking on the 88MC200 package. [Table 3](#) shows part ordering options.

Figure 4: 88MC200 Microcontroller Package Markings for 88-Pin Part

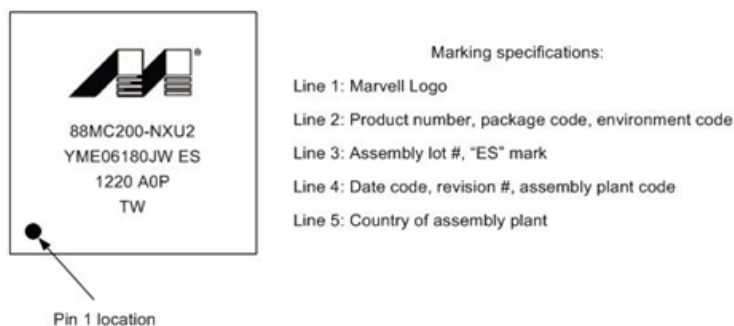
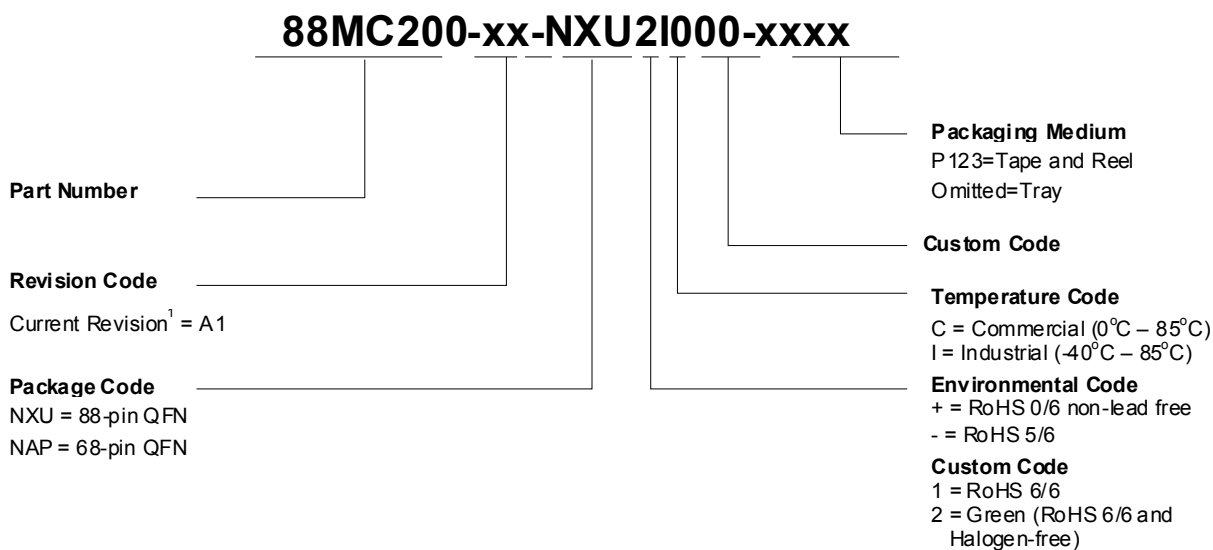


Figure 5: Part Ordering Number



Note: ¹ Contact your local sales representative for the latest version information when ordering.

Table 3: 88MC200 Microcontroller Part Ordering Options

Part Number	Description
88MC200-A1-NXU2C000-P123	"C" grade, 88-pin QFN in tape and reel
88MC200-A1-NAP2C000-P123	"C" grade, 68-pin QFN in tape and reel
88MC200-A1-NXU2I000-P123	"I" grade, 88-pin QFN in tape and reel
88MC200-A1-NAP2I000-P123	"I" grade, 68-pin QFN in tape and reel
88MC200-A1-NXU2C000	"C" grade, 88-pin QFN in tray
88MC200-A1-NAP2C000	"C" grade, 68-pin in QFN in tray
88MC200-A1-NXU2I000	"I" grade, 88-pin QFN in tray
88MC200-A1-NAP2I000	"I" grade, 68-pin in QFN in tray
NOTE: MOQ (Minimum order quantity) For tape and reel: - NXU (88-pin QFN) - 1000 pcs - NAP (68-pin QFN) - 2000 pcs For tray: - NXU (88-pin QFN) - 1680 pcs - NAP (68-pin QFN) - 2600 pcs - All small quantity non-production samples will be shipped in tray.	



88MC200 Microcontroller Datasheet

2 Processor Overview

2.1 Overview

The Marvell 88MC200 microcontroller integrates the full featured ARM Cortex-M3 processor in its SoC subsystem. The ARM Cortex-M3 processor provides high performance and low-cost platform. It offers many new features, including a Thumb-2 instruction set, low interrupt latency, hardware division, memory protection unit, etc.

Details of the ARM Cortex-M3 core are available in the ARM Cortex-M3 r2p1 technical reference manual.

2.1.1 Cortex M3 Features

- 32-bit ARM Cortex-M3 architecture optimized for embedded applications
- Cortex-M3 core can operate at up to 200 MHz
- Thumb-2 mixed 16/32-bit instruction set
- Hardware division and fast multiplier
- Little-endian memory space
- Memory protection unit (MPU) for protected operating system functionality
- Includes Nested Vectored Interrupt Controller (NVIC)
- SysTick Timer provided by Cortex-M3 core
- Wakeup Interrupt Controller (WIC) for waking up the CPU from reduced power modes
- Standard JTAG debug interface
- Serial Wire JTAG debug port (SWJ-DP)
- Enhanced system debug with extensive breakpoint

2.1.2 Memory Protection Unit (MPU)

The Memory Protection Unit (MPU) is used to improve the reliability of an embedded system by protecting critical data in the applications.

The MPU separates the memory into distinct regions and implements protection by preventing disallowed accesses. The MPU can manage as many as eight protection regions. The protection area sizes are between 32 bytes and all 4 gigabytes of addressable memory.

The MPU is optional and can be bypassed for applications that do not need it.

2.1.3 Nested Vectored Interrupt Controller (NVIC)

The NVIC is integrated in the Cortex-M3 core. The tight coupling to the CPU allows for low interrupt latency and efficient processing of interrupts. Features include:

- Supports up to 64 interrupts
- Supports 16 interrupt priority levels, Level 0 is the highest interrupt priority
- Control system exceptions and peripheral interrupts
- Supports interrupt tail chaining
- Non-maskable interrupt

Chapter 6: Memory Map, Interrupts, and AHB Bus Fabric provides a detailed description of the interrupts.

2.1.4 SysTick Timer

The ARM Cortex-M3 includes a system tick timer (SysTick). This timer is dedicated to real-time operating systems, but could also be used as a standard downcounter. It features:

- A 24-bit downcounter
- Auto reload capability
- Maskable system interrupt generation when the counter reaches 0
- Programmable clock source

3

I/O Configuration

This chapter describes the pin-multiplexing scheme and I/O padding for the Marvell 88MC200 system. Many of the package pins are multiplexed so that they can be configured as general-purpose I/Os or any one of the alternate functions using the Multi-Function Pin Alternate-Function Select registers. Some functions can be configured to appear on one of several different pins using alternate function controls. The I/O pins can be individually configured to support the following functions listed below:

- External Interrupt
- JTAG
- GPT
- UART
- I2C
- SSP
- SDIO
- USB
- RTC
- QSPI
- AES
- ADC/DAC/Analog Comparator
- WAKEUP event/interrupt input for power mode switch
- GPIO

The I/O pad can support pullup, pulldown, or tri-state configurations. Detailed information regarding the I/O pins on each package is listed in *Chapter 1: Overview*.

3.1 Pinmux Alternate Functions

Each I/O or package pin has a dedicated register to control its functionality. The function number is specified from 0 to 7 by the least significant 3 bits of each register. The default function is Function 0 after reset. Function 0 for all I/O pins except for TDO/TCK/TMS/TDI/TRST_N(GPIO_20~24), WAKE_UP0/WAKE_UP1(GPIO_25/26) and USB_DP/USB_DM(GPIO_57/58) is the GPIO function, and the default value of the I/O function is a pullup. A detailed description of the Pinmux register is located in Appendix Section 1.

The alternate functions for each GPIO are described in the following sections.

3.1.1 GPIO_0 (Offset=0x0)

Function #	Name	Input/output	Description
7	N/A	N/A	N/A
6	ADC0_IN7/ACOMP0_IN7/ACOMP1_IN7	AI/O	ADC0 Channel 7 or ACOMP0/1 Channel 7
5	N/A	N/A	N/A
4	N/A	N/A	N/A
3	N/A	O	N/A
2	GPT0_CH0	I/O	GPT0 Channel 0
1	N/A	N/A	N/A
0	GPIO_0	I/O	GPIO 0

3.1.2 GPIO_1 (Offset=0x4)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	ADC0_IN6/ACOMP0_IN6/ACOMP1_IN6	AI/O	ADC0 Channel 6 or ACOMP0/1 Channel 6
5	N/A	N/A	N/A
4	N/A	N/A	N/A
3	N/A	N/A	N/A
2	GPT0_CH1	I/O	GPT0 Channel 1
1	N/A	N/A	N/A
0	GPIO_1	I/O	GPIO 1

3.1.3 GPIO_2 (Offset=0x8)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	ADC0_IN5/ACOMP0_IN5/ACOMP1_IN5	AI/O	ADC0 Channel 5 or ACOMP0/1 Channel 5
5	N/A	N/A	N/A
4	N/A	N/A	N/A
3	N/A	N/A	N/A
2	GPT0_CH2	I/O	GPT0 Channel 2
1	N/A	N/A	N/A
0	GPIO_2	I/O	GPIO 2

3.1.4 GPIO_3 (Offset=0xC)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	ADC0_IN4/ACOMP0_IN4/ACOMP1_IN4	AI/O	ADC0 Channel 4 or ACOMP0/1 Channel 4
5	N/A	N/A	N/A
4	N/A	N/A	N/A
3	N/A	N/A	N/A
2	GPT0_CH3	I/O	GPT0 Channel 3
1	N/A	N/A	N/A
0	GPIO_3	I/O	GPIO 3

3.1.5 GPIO_4 (Offset=0x10)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	ADC0_IN3/ACOMP0_IN3/ACOMP1_IN3/DACA/ADC0_REF	A/I/O	ADC0 Channel 3 or ACOMP0/1 Channel 3 or DAC Channel A or ADC0 Vref
5	N/A	N/A	N/A
4	GPT1_CLKIN	I	GPT1 clock in
3	I2C1_SDA	I/O	SDA for I2C1
2	GPT0_CH4	I/O	GPT0 Channel 4
1	N/A	N/A	N/A
0	GPIO_4	I/O	GPIO 4

3.1.6 GPIO_5 (Offset=0x14)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	ADC0_IN2/ACOMP0_IN2/ACOMP1_IN2	A/I/O	ADC0 Channel 2 or ACOMP0/1 Channel 2
5	N/A	N/A	N/A
4	GPT3_CLKIN	I	GPT3 clock in
3	I2C1_SCL	I/O	SCL for I2C1
2	GPT0_CH5	I/O	GPT0 Channel 5
1	N/A	N/A	N/A
0	GPIO_5	I/O	GPIO 5

3.1.7 GPIO_6 (Offset=0x18)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	ADC0_IN1/ACOMP0_IN1/ ACOMP1_IN1	AI/O	ADC0 Channel 1 or ACOMP0/1 Channel 1
5	N/A	N/A	N/A
4	GPT3_CH0	I/O	GPT3 Channel 0
3	GPT0_CLKIN	I	GPT0 clock in
2	GPT1_CH0	I/O	GPT1 Channel 0
1	N/A	N/A	N/A
0	GPIO_6	I/O	GPIO 6

3.1.8 GPIO_7 (Offset=0x1C)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	ADC0_IN0/ACOMP0_IN0/ ACOMP0_IN0	AI/O	ADC0 Channel 0 or ACOMP0/1 Channel 0
5	N/A	N/A	N/A
4	GPT3_CH1	I/O	GPT3 Channel 1
3	GPT2_CLKIN	I	GPT2 clock in
2	GPT1_CH1	I/O	GPT1 Channel 1
1	N/A	N/A	N/A
0	GPIO_7	I/O	GPIO7

3.1.9 GPIO_8 (Offset=0x20)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	ADC1_IN0	AI/O	ADC1 Channel 0
5	N/A	N/A	N/A
4	GPT3_CH2	I/O	GPT3 Channel 2
3	I2C1_SDA	I/O	SDA for I2C1
2	GPT1_CH2	I/O	GPT1 Channel 2
1	N/A	N/A	N/A
0	GPIO_8	I/O	GPIO 8

3.1.10 GPIO_9 (Offset=0x24)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	ADC1_IN1	AI/O	AD1 Channel 1
5	N/A	N/A	N/A
4	GPT3_CH3	I/O	GPT3 Channel 3
3	I2C1_SCL	I/O	SCL for I2C1
2	GPT1_CH3	I/O	GPT1 Channel 3
1	N/A	N/A	N/A
0	GPIO_9	I/O	GPIO 9

3.1.11 GPIO_10 (Offset=0x28)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	ADC1_IN2/ DAC_REF	AI/O	ADC1 Channel 2 or DAC Vref
5	N/A	N/A	N/A
4	GPT3_CH4	I/O	GPT3 Channel 4
3	I2C2_SDA	I/O	SDA for I2C2
2	GPT1_CH4	I/O	GPT1 Channel 4
1	N/A	N/A	N/A
0	GPIO_10	I/O	GPIO 10

3.1.12 GPIO_11 (Offset=0x2C)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	ADC1_IN3/DACB/ ADC1_REF	AI/O	ADC1 Channel 3 or DAC Channel B or ADC1 Vref
5	N/A	N/A	N/A
4	GPT3_CH5	I/O	GPT3 Channel 5
3	I2C2_SCL	I/O	SCL for I2C2
2	GPT1_CH5	I/O	GPT1 Channel 5
1	N/A	N/A	N/A
0	GPIO_11	I/O	GPIO 11

3.1.13 GPIO_16 (Offset=0x40)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	GPT0_CH4	I/O	GPT0 Channel 4
3	GPT3_CH0	I/O	GPT3 Channel 0
2	GPT2_CH4	I/O	GPT2 Channel 4
1	N/A	N/A	N/A
0	GPIO_16	I/O	GPIO 16

3.1.14 GPIO_17 (Offset=0x44)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	GPT0_CH5	I/O	GPT0 Channel 5
3	GPT3_CH1	I/O	GPT3 Channel 1
2	GPT2_CH5	I/O	GPT2 Channel 5
1	GPT1_CH0	I/O	GPT1 Channel 0
0	GPIO_17	I/O	GPIO 17

3.1.15 GPIO_18 (Offset=0x48)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	OSC32K_IN	AI	32.768 kHz crystal oscillator input
5	N/A	N/A	N/A
4	UART1_SIR_OUT	O	SIR_OUT fro UART0
3	I2C0_SDA	I/O	SDA for I2C0
2	GPT3_CH0	I/O	GPT3 Channel 0
1	N/A	N/A	N/A
0	GPIO_18	I/O	GPIO 18

3.1.16 GPIO_19 (Offset=0x4C)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	OSC32K_OUT	AO	32.768 kHz crystal oscillator output
5	N/A	N/A	N/A
4	UART1_SIR_IN	I	SIR_IN for UART1
3	I2C0_SCL	I/O	SCL for I2C0
2	GPT3_CH1	I/O	GPT3 Channel 1
1	N/A	N/A	N/A
0	GPIO_19	I/O	GPIO 19

3.1.17 GPIO_20 (Offset=0x50)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6-2	N/A	N/A	N/A
1	GPIO_20	I/O	GPIO 20
0	TDO	O	TDO for JTAG

3.1.18 GPIO_21 (Offset=0x54)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6-2	N/A	N/A	N/A
1	GPIO_21	I	GPIO 21
0	TCK	I/O	TCK for JTAG

3.1.19 GPIO_22 (Offset=0x58)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6-2	N/A	N/A	N/A
1	GPIO_22	I/O	GPIO 22
0	TMS	I	TMS for JTAG

3.1.20 GPIO_23 (Offset=0x5C)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6-2	N/A	N/A	N/A
1	GPIO_23	I/O	GPIO 23
0	TDI	I	TDI for JTAG

3.1.21 GPIO_24 (Offset=0x60)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6-2	N/A	N/A	N/A
1	GPIO_24	I/O	GPIO 24
0	TRST_n	I	TRST_n for JTAG

3.1.22 GPIO_25 (Offset=0x64)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART0_SIR_IN	I	SIR_IN for UART0
3	ACOMP1_GPIO_OUT	O	ACOMP1 output synchronous or asynchronous level signals
2	ACOMP0_GPIO_OUT	O	ACOMP0 output synchronous or asynchronous level signals
1	GPIO_25	I/O	GPIO 25
0	WAKE_UP0	I	Wake up 0 signal

3.1.23 GPIO_26 (Offset=0x68)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	COMP_IN_N	AI	Negative input to AON domain comparator
5	N/A	N/A	N/A
4	UART0_SIR_OUT	O	SIR_OUT for UART0
3	ACOMP1_EDGE_PULSE	O	Output pulse aligned with Synchronized comparison result
2	ACOMP0_EDGE_PULSE	O	Output pulse aligned with Synchronized comparison result
1	GPIO_26	I/O	GPIO 26
0	WAKE_UP1	I	Wake up 1 signal

3.1.24 GPIO_27 (Offset=0x6C)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	COMP_IN_P	AI	Positive input to AON domain comparator
5	BOOT	I	Boot pin
4	UART0_DSRn	I	DSRn for UART0
3	N/A	N/A	N/A
2	GPT3_CH2	I/O	GPT Channel 2
1	ACOMP0_GPIO_OUT	O	ACOMP0 output synchronous or asynchronous level signals
0	GPIO_27	I/O	GPIO 27

3.1.25 GPIO_28 (Offset=0x70)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	SDIO_LED	O	LED for SDIO
4	UART0_DCDn	I	DCDn for UART0
3	N/A	N/A	N/A
2	GPT3_CH3	I/O	GPT3 Channel 3
1	ACOMP0_EDGE_PULSE	O	ACOMP0 output synchronous or asynchronous level signals
0	GPIO_28	I/O	GPIO 21

3.1.26 GPIO_29 (Offset=0x74)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	SDIO_CDn	I	Card Detect signal
4	UART0_Rin	I	Rin for UART0
3	ACOMP0_GPIO_OUT	O	ACOMP0 output synchronous or asynchronous level signals
2	GPT3_CH4	I/O	GPT3 Channel 4
1	ACOMP1_GPIO_OUT	O	ACOMP1 output synchronous or asynchronous level signals
0	GPIO_29	I/O	GPIO 29

3.1.27 GPIO_30 (Offset=0x78)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	SDIO_WP	I	Write Protect for SDIO
4	UART0_DTRn	O	DTRn for UART0
3	ACOMP0_EDGE_PULSE	O	Output pulse aligned with Synchronized comparison result
2	GPT3_CH5	I/O	GPT3 Channel 5
1	ACOMP1_EDGE_PULSE	O	Output pulse aligned with Synchronized comparison result
0	GPIO_30	I/O	GPIO 30

3.1.28 GPIO_32 (Offset=0x80)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	GPT0_CH0	I/O	GPT0 Channel 0
3	GPT2_CH0	I/O	GPT2 Channel 0
2	UART2_CTSn	I	CTS _n for UART2
1	SSP0_CLK	I/O	Clock for SSP0
0	GPIO_32	I/O	GPIO 32

3.1.29 GPIO_33 (Offset=0x84)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	GPT0_CH1	I/O	GPT0 Channel 1
3	GPT2_CH1	I/O	GPT2 Channel 1
2	UART2_RTSn	O	RTSn for UART2
1	SSP0_FRM	I/O	Frame for SSP0
0	GPIO_33	I/O	GPIO 33

3.1.30 GPIO_34(Offset=0x88)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	GPT0_CH2	I/O	GPT0 Channel 2
3	GPT2_CH2	I/O	GPT2 Channel 2
2	UART2_TXD	O	TXD for UART2
1	SSP0_RXD	I	RXD for SSP0
0	GPIO_34	I/O	GPIO 34

3.1.31 GPIO_35(Offset=0x8C)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	GPT0_CH3	I/O	GPT0 Channel 3
3	GPT2_CH3	I/O	GPT2 Channel 3
2	UART2_RXD	I	RXD for UART2
1	SSP0_TXD	O	TXD for SSP0
0	GPIO_35	I/O	GPIO 35

3.1.32 GPIO_40(Offset=0xA0)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	N/A	N/A	N/A
3	GPT1_CH2	I/O	GPT1 Channel 2
2	SSP2_CLK	I/O	Clock for SSP2
1	UART3_CTSn	I	CTS _n for UART3
0	GPIO_40	I/O	GPIO 40

3.1.33 GPIO_41(Offset=0xA4)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	N/A	N/A	N/A
3	GPT1_CH3	I/O	GPT1 Channel 3
2	SSP2_FRM	I/O	Frame for SSP2
1	UART3_RTSn	O	RTSn for UART3
0	GPIO_41	I/O	GPIO 41

3.1.34 GPIO_42(Offset=0xA8)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	N/A	N/A	N/A
3	GPT1_CH4	I/O	GPT1 Channel 4
2	SSP2_RXD	I	RXD for SSP2
1	UART3_TXD	O	TXD for UART3
0	GPIO_42	I/O	GPIO 42

3.1.35 GPIO_43 (Offset=0xAC)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	N/A	N/A	N/A
3	GPT1_CH5	I/O	GPT1 Channel 5
2	SSP2_TXD	O	TXD for SSP2
1	UART3_RXD	I	RXD for UART3
0	GPIO_43	I/O	GPIO 43

3.1.36 GPIO_44 (Offset=0xB0)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	SDIO_CDn	I	SDIO card signal
4	ADC_DAC_TRIGGER	I	External trigger for ADC/DAC
3	GPT3_CH0	I/O	GPT3 Channel 0
2	GPT0_CLKIN	I	GPT0 clock in
1	I2C0_SDA	I/O	SDA for I2C0
0	GPIO_44	I/O	GPIO 44

3.1.37 GPIO_45 (Offset=0xB4)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	SDIO_WP	I	SDIO write protect
4	ADC_DAC_TRIGGER	I	External trigger for ADC/DAC
3	GPT3_CH1	I/O	GPT3 Channel 1
2	USB_DRVVBUS	O	Drive VBUS to 5V
1	I2C0_SCL	I/O	SCL fro I2C0
0	GPIO_45	I/O	GPIO 45

3.1.38 GPIO_50 (Offset=0xC8)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	N/A	N/A	N/A
3	N/A	N/A	N/A
2	SDIO_LED	O	LED for SDIO
1	GPT1_CH5	I/O	GPT1 Channel 5
0	GPIO_50	I/O	GPIO 50

3.1.39 GPIO_51 (Offset=0xCC)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART2_DSRn	I/O	DSTn for UART2
3	GPT0_CH0	I/O	GPT0 Channel0
2	SSP2_CLK	O	CLK for SSP2
1	SDIO_CLK	O	CLK for SDIO
0	GPIO_51	I/O	GPIO 51

3.1.40 GPIO_52 (Offset=0xD0)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART2_DCDn	I	DCDn for UART2
3	GPT0_CH1	I/O	GPT0 Channel 1
2	SSP2_FRM	I/O	Frame for SSP2
1	SDIO_3	I/O	Data 3 for SDIO
0	GPIO_52	I/O	GPIO 52

3.1.41 GPIO_53 (Offset=0xD4)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART2_RIN	I	Rin for UART2
3	GPT0_CH2	I/O	GPT0 Channel 2
2	SSP2_RXD	I	RXD for SSP2
1	SDIO_2	I/O	Data2 for SDIO
0	GPIO_53	I/O	GPIO 53

3.1.42 GPIO_54 (Offset=0xD8)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART2_DTRn	O	DTRn for UART2
3	GPT0_CH3	I/O	GPT0 Channel 3
2	SSP2_TXD	O	TXD for SSP2
1	SDIO_1	I/O	Data1 for SDIO
0	GPIO_54	I/O	GPIO 54

3.1.43 GPIO_55 (Offset=0xDC)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART2_SIR_OUT	O	SIR_OUT for UART2
3	GPT0_CH4	I/O	GPT0 Channel 4
2	GPT2_CLKIN	I	GPT2 clock in
1	SDIO_0	I/O	Data0 fro SDIO
0	GPIO_55	I/O	GPIO 55

3.1.44 GPIO_56 (Offset=0xE0)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART2_SIR_IN	I	SIR_IN for UART2
3	GPT0_CH5	I/O	GPT0 Channel 5
2	GPT3_CLKIN	I	GPT3 clock in
1	SDIO_CMD	I/O	CMD for SDIO
0	GPIO_56	I/O	GPIO 56

3.1.45 GPIO_57(Offset=0xE4)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART3_SIR_OUT	O	SIR_OUT for UART3
3	N/A	N/A	N/A
2	GPT0_CLKIN	I	GPT0 clock in
1	GPIO_57	I/O	GPIO 57
0	USB_DP	A/I/O	USB2 D+ pad

3.1.46 GPIO_58 (Offset=0xE8)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART3_SIR_IN	I	SIR_IN for UART3
3	N/A	N/A	N/A
2	GPT1_CLKIN	I	GPT1 clock in
1	GPIO_58	I/O	GPIO 58
0	USB_DM	AI/O	USB2 D- pad

3.1.47 GPIO_59 (Offset=0xEC)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART3_DSRn	I	DSRn for UART3
3	GPT3_CH2	I/O	GPT3 Channel 2
2	N/A	N/A	N/A
1	UART1_CTSn	I	CTSn for UART1
0	GPIO_59	I/O	GPIO 59

3.1.48 GPIO_60 (Offset=0xF0)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART3_DCDn	I	DCDn for UART3
3	GPT3_CH3	I/O	GPT3 Channel 3
2	N/A	N/A	N/A
1	UART1_RTSn	O	RTSn for UART1
0	GPIO_60	I/O	GPIO 60

3.1.49 GPIO_61 (Offset=0xF4)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART3_RIN	I	Rin for UART3
3	GPT3_CH4	I/O	GPT3 Channel 4
2	N/A	N/A	N/A
1	UART1_TXD	O	TXD for UART1
0	GPIO_61	I/O	GPIO 61

3.1.50 GPIO_62 (Offset=0xF8)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART3_DTRn	O	DTRn for UART3
3	GPT3_CH5	I/O	GPT3 Channel 5
2	N/A	N/A	N/A
1	UART1_RXD	I	RXD for UART1
0	GPIO_62	I/O	GPIO 62

3.1.51 GPIO_63 (Offset=0xFC)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART1_DSRn	I	DSRn for UART1
3	GPT3_CH2	I/O	GPT3 Channel 2
2	SSP1_CLK	I/O	Clock for SSP1
1	UART1_CTSn	I	CTSn for UART1
0	GPIO_63	I/O	GPIO 63

3.1.52 GPIO_64 (Offset=0x100)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART1_DCDn	I	DCDn for UART1
3	GPT3_CH3	I/O	GPT3 Channel 3
2	SSP1_FRM	I/O	Frame for SSP0
1	UART1_RTSn	O	RTSn for UART1
0	GPIO_64	I/O	GPIO 64

3.1.53 GPIO_65 (Offset=0x104)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART1_Rin	I	Rin for UART1
3	GPT3_CH4	I/O	GPT3 channel 4
2	SSP1_RXD	I	RXD for SSP1
1	UART1_TXD	O	TXD for UART1
0	GPIO_65	I/O	GPIO 65

3.1.54 GPIO_66 (Offset=0x108)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	UART1_DTRn	O	DTRn for UART1
3	GPT3_CH5	I/O	GPT3 Channel 5
2	SSP1_TXD	O	TXD for SSP1
1	UART1_RXD	I	RXD for UART1
0	GPIO_66	I/O	GPIO 66

3.1.55 GPIO_68 (Offset=0x110)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	N/A	N/A	N/A
3	N/A	N/A	N/A
2	GPT1_CLKIN	I	GPT1 clock in
1	GPT2_CH2	I/O	GPT2 channel 2
0	GPIO_68	I/O	GPIO 68

3.1.56 GPIO_72 (Offset=0x120)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	QSPI1_SS _n	O	SS for QSPI1
3	GPT1_CH2	I/O	GPT1 Channel 2
2	GPT2_CLKIN	I	GPT2 clock in
1	UART0_CTS _n	I	CTS _n for UART0
0	GPIO_72	I/O	GPIO 72

3.1.57 GPIO_73 (Offset=0x124)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	QSPI1_CLK	O	Clock for QSPI1
3	GPT1_CH3	I/O	GPT1 Channel 3
2	GPT3_CLKIN	I	GPT3 clock in
1	UART0_RTS _n	O	RTS _n for UART0
0	GPIO_73	I/O	GPIO 73

3.1.58 GPIO_74 (Offset=0x128)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	N/A	N/A	N/A
3	GPT1_CH4	I/O	GPT1 Channel 4
2	RC32M_CLKOUT	O	RC32M clock out
1	UART0_TXD	O	TXD for UART0
0	GPIO_74	I/O	GPIO 74

3.1.59 GPIO_75 (Offset=0x12C)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	N/A	N/A	N/A
3	GPT1_CH5	I/O	GPT1 Channel 5
2	N/A	N/A	N/A
1	UART0_RXD	I	RXD for UART0
0	GPIO_75	I/O	GPIO 75

3.1.60 GPIO_76 (Offset=0x130)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	QSPI1_D0	I/O	Data0 for QSPI1
3	I2C0_SDA	I/O	SDA for I2C0
2	SSP0_CLK	I/O	Clock for SSP0
1	UART2_CTSn	I	CTS _n for UART2
0	GPIO_76	I/O	GPIO 76

3.1.61 GPIO_77 (Offset=0x134)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	QSPI1_D1	I/O	Data1 for QSPI1
3	I2C0_SCL	I/O	SCL for I2C0
2	SSP0_FRM	I/O	Frame for SSP0
1	UART2_RTSn	O	RST _n for UART2
0	GPIO_77	I/O	GPIO 77

3.1.62 GPIO_78 (Offset=0x138)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	QSPI1_D2	I/O	Data2 for QSPI1
3	GPT1_CH0	I/O	GPT1 Channel 0
2	SSP0_RXD	I	Input data to SSP0
1	UART2_TXD	O	Output data from UART2
0	GPIO_78	I/O	GPIO 78

3.1.63 GPIO_79 (Offset=0x13C)

Function #	Name	Input/Output	Description
7	N/A	N/A	N/A
6	N/A	N/A	N/A
5	N/A	N/A	N/A
4	QSPI1_D3	I/O	Data3 for QSPI1
3	GPT1_CH1	I/O	GPT1 Channel 1
2	SSP0_TXD	O	Output data from SSP0
1	UART2_RXD	I	input data to UART2
0	GPIO_79	I/O	GPIO 79

3.1.64 I/O Padding

The I/O padding can be configured to pullup, pulldown, or tri-state mode. The I/O pins are configured to the default mode when selecting one pinmux alternate function. When one I/O pin is set to a GPIO function and the data transfer direction is input, users can reconfigure the I/O pin to pullup, pulldown, or tri-state mode by setting bits [15:13] and bit [3] of the corresponding I/O Pinmux Configuration register. See [Table 4](#).

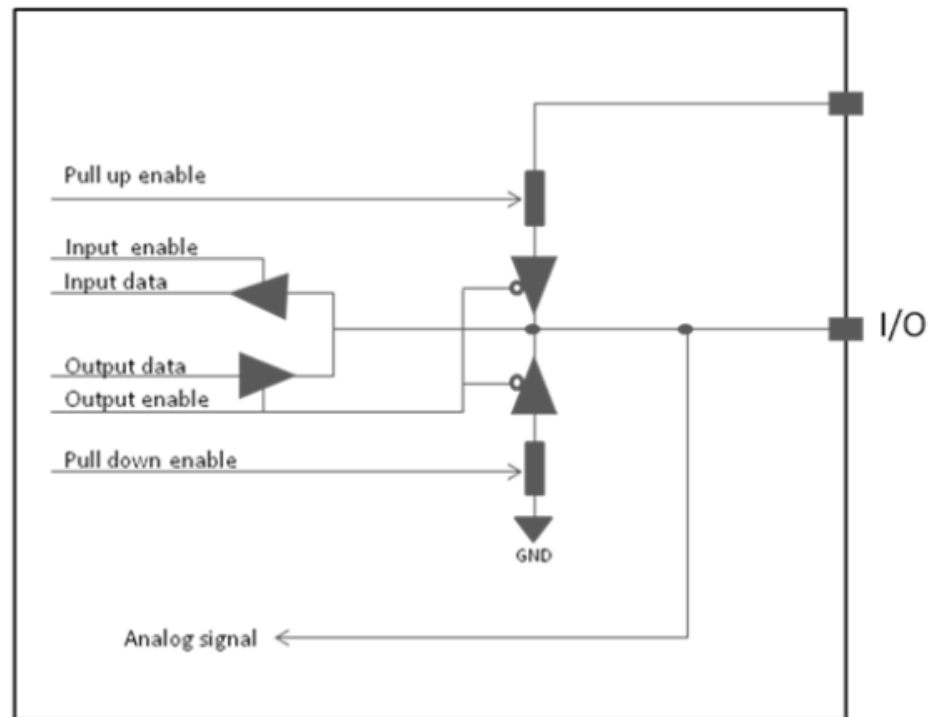
Table 4: I/O Pin Mode Configuration

Bit Field of I/O PinMux Configuration Register				Description
[15]	[14]	[13]	[3]	
0	X	X	1	Pullup and pulldown from pinmux alternate function
1	1	0	1	Pullup enabled
1	0	1	1	Pulldown enabled
1	1	1	1	Not allowed
1	0	0	0	Tri-state

Note: X = "don't care"

GPIO_57/58 does not support pullup mode. [Figure 6](#) shows the I/O padding structure.

Figure 6: I/O Padding Structure





88MC200 Microcontroller Datasheet

4 System Control

4.1 Overview

The System Control unit provides several system features and control registers for memory space configuration, the DMA handshake interface mapping, peripheral software reset and USB control.

These registers must be configured correctly to ensure correct functionality of the memories, DMA, USB and other peripherals on-chip.

4.2 Features

The following are some of the features of System Control:

- Memory Space Configuration: The MEM register helps to re-configure RAM1 and RAM2 to CODE or SRAM space available on-chip, based on the requirements.
- DMA handshake interface mapping: The DMA_HS register enables mapping the DMA handshaking interface to the required DMA channel in order to perform DMA transfers for different peripherals on-chip.
- Peripheral software reset: The PERI_SW_RST register is used to program reset for various peripherals on the chip. Writing 0 to certain bits resets the corresponding module. It resets only the function clock domain.

4.3 Register Description

A detailed description of the registers along with the register memory map table is located in Appendix Section 2.



88MC200 Microcontroller Datasheet

5 Power, Reset, and Clock Control

5.1 Overview

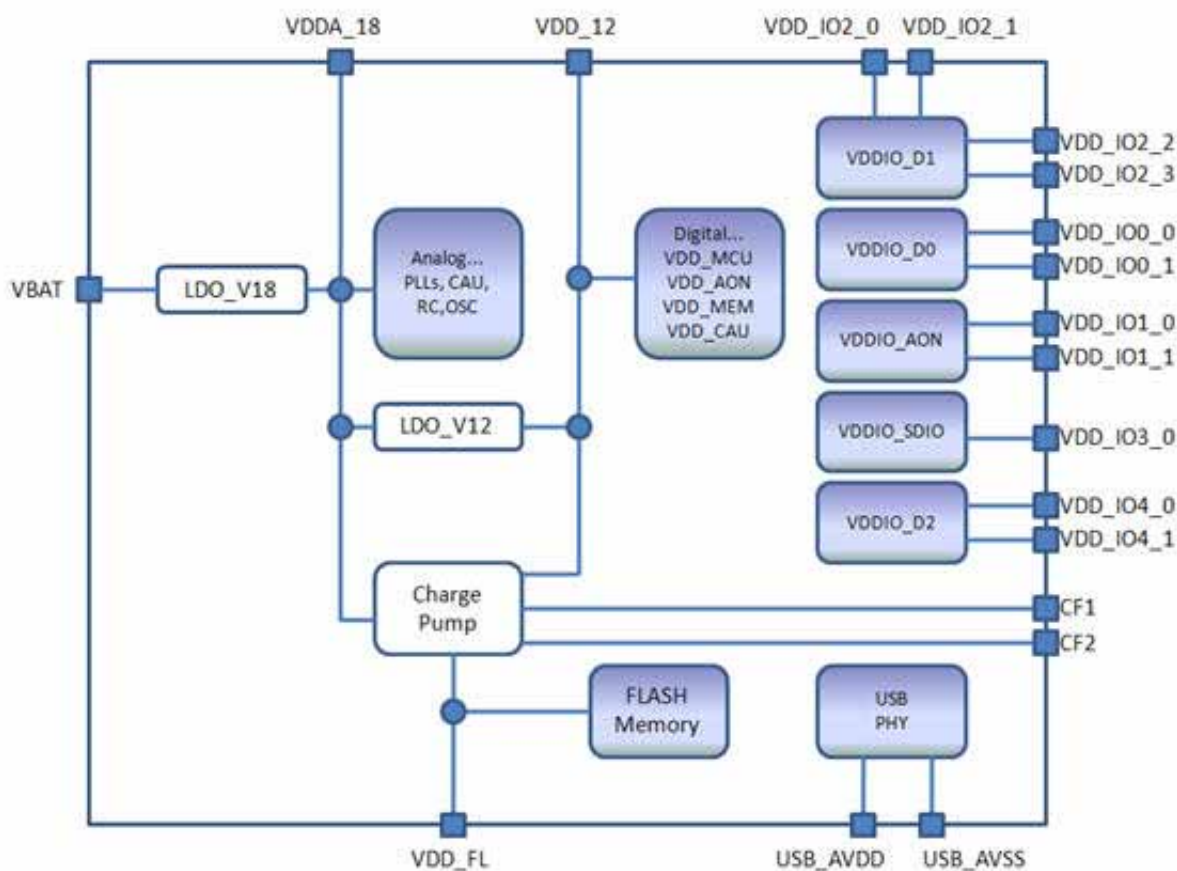
This chapter describes the power, reset, and clock control functions of the 88MC200 microcontroller.

The power supply, power mode, and on-chip DC-DC converter, clocking, reset, and wake-up signals are managed by the Power Management Unit (PMU), which is in the Always ON (AON) power domain.

5.2 Power Supply

The 88MC200 MCU power supplies are shown in [Figure 5](#).

Figure 7: 88MC200 MCU Power Supply Overview



The 88MC200 microcontroller provides several independent power domains. There are four digital power domains in 88MC200.

- VDD_AON: PMU, RTC, low power comparator, and 4K memory brownout detection logic are in VDD_AON power domain. They are operational in all power modes.
- VDD_MEM: 192 kB of SRAM sits on this power domain. It is on in PM0, PM1, PM2 and PM3.
- VDD_MCU: Cortex-M3, the remainder of the 320 KB SRAM (refer to Table 5: VDD_MCU Address Memory), all AHB and APB peripherals and pin mux are in this power domain. It is on in PM0, PM1, and PM2 power modes.
- VDD_CAU: RC32M digital and ADC/DAC/ACOMP digital control logic are in this power domain. It is on in PM0, PM1, and PM2 power modes.

Table 5: VDD_MCU Address Memory

SYS_CRTL.MEM.CFG	CODE	SRAM
2'b00	0x128000-0x12FFFF (32 kB) 0x130000-0x15FFFF (192 kB)	0x20008000-0x2000FFFF (32 kB) 0x20010000-0x2001FFFF (64 kB)
2'b01	0x128000-0x12FFFF (32 kB)	0x20008000-0x2000FFFF (32 kB) 0x20010000-0x2001FFFF (64 kB) 0x20020000-0x2004FFFF (192 kB)
2'b10	0x128000-0x12FFFF (32 kB) 0x130000-0x15FFFF (192 kB) 0x168000-0x16FFFF (32 kB)	0x20010000-0x2001FFFF (64 kB)

5.2.1 Power Pins

Table 6: 88MC200 Power Pins

QFN88 Pin Number	QFN68 Pin Number	Pin Name	Description
1	1	VDD_12	1.2V Flycap
2	2	CF2	External Cap connection
3	3	CF1	External Cap connection
4	4	VBAT	VBAT power supply
5	5	VBAT	VBAT power supply
6	6	VDDA_18	1.8V Flycap
	7	VDD_IO0_0	GPIO_D0 domain power
20	16	VDD_IO0_1	
27	22	VDD_IO1_0	AON domain power
32	27	VDD_IO1_1	

Table 6: 88MC200 Power Pins (Continued)

QFN88 Pin Number	QFN68 Pin Number	Pin Name	Description
39		VDD_IO2_0	GPIO_D1 domain power
44	34	VDD_IO2_1	
49		VDD_IO2_2	
52	37	VDD_IO2_3	
56	40	USB_AVDD	USB domain power
60	44	VDD_IO3_0	SDIO domain power
67	51	VDD_IO4_0	GPIO_D2 domain power
81	61	VDD_IO4_1	
88	68	VDD_FL	Flycap for in-package flash

5.2.2 I/O Power Configuration

The 88MC200 device has six configurable I/O domains, which are defined as GPIO_D0 domain, GPIO_D1 domain, GPIO_D2 domain, AON domain, SDIO domain, FL domain. The FL domain supports 3.3V only; the others support both 1.8V and 3.3V independently. They also support independent power-off functionality.

- GPIO_D0 domain: GPIO_0~GPIO_17
- AON domain: GPIO_18~GPIO_27
- GPIO_D1 domain: GPIO_28~GPIO_50
- SDIO domain: GPIO_51~GPIO_56
- GPIO_D2 domain: GPIO_59~GPIO_79
- FL domain: FLASH pads

The required I/O power can be chosen by configuring the PMU register, IO_PAD_PWR_CFG as shown in [Table 7](#). A detailed description of the IO_PAD_PWR_CFG is located in the PMU Register description in Appendix Section 3.

Table 7: I/O Power Configuration

IO Power domain	Corresponding GPIOs	Register Field	Value	Description
GPIO_D0 I/O domain power	GPIO_0~GPIO_17	PMU.IO_PAD_PWR_CFG.V18EN_LVL_GPIO0_V18EN_CORE	0 (default)	3.3V
			1	1.8V
AON IO domain power	GPIO_18~GPIO_27	PMU.IO_PAD_PWR_CFG.V18EN_LVL_AON_V18EN_CORE	0 (default)	3.3V
			1	1.8V
GPIO_D1 I/O domain power	GPIO_28~GPIO_50	PMU.IO_PAD_PWR_CFG.V18EN_LVL_GPIO1_V18EN_CORE	0 (default)	3.3V
			1	1.8V
SDIO IO domain power	GPIO_51~GPIO_56	PMU.IO_PAD_PWR_CFG.V18EN_LVL_SDIO_V18EN_CORE	0 (default)	3.3V
			1	1.8V

**Note**

The real power supply to VDD_IOx_y power pin should match the corresponding configuration of the IO_PAD_PWR_CFG register.

There are two bits: VDD_IO4_REG_PDB_CORE and VDD_IO6_REG_PDB_CORE in GPIO_D1 domain. Since the GPIO_D1 domain has more pads than others, it requires additional I/O power supply. For configuration, software must set VDD_IO4_REG_PDB_CORE the same as VDD_IO6_REG_PDB_CORE.

After the 88MC200 microcontroller powers on, all I/O domains are off except GPIO_18~GPIO_27, AON domain, which is controlled by the IO_PAD_PWR_CFG.POR_LVL_[domain]_LOW_VDDB_CORE. The default I/O is applied to 3.3V, which is controlled by bit IO_PAD_PWR_CFG.V18EN_LVL_[domain]_V18EN_CORE. The default pad regulator works in normal mode, which is controlled by IO_PAD_PWR_CFG.VDD_[domain]_REG_PDB_CORE.

Firmware could configure the power voltage of the corresponding I/O domain at any time to apply to different devices. Also firmware could configure the corresponding domain, where the pad regulator is located into powerdown mode to save power consumption at any time.

Firmware must power on the I/O domain (IO_PAD_PWR_CFG.POR_LVL_[domain]_LOW_VDDB_CORE) first before the I/O data transfer starts. The pad value is tri-stated before the I/O is powered on. In Sleep mode or for power consumption savings, firmware must also power off the I/O domain (IO_PAD_PWR_CFG.POR_LVL_[domain]_LOW_VDDB_CORE) before entering Sleep mode; otherwise the pad value is unknown.

**Note**

No matter the I/O function, the input level of I/O pins should not exceed the corresponding I/O domain power supply.

5.2.3

AON Domain

The PMU, RTC, ultra low-power comparator, brown detect logic, and 4K_MEM are in the AON domain. These modules can be powered in all power modes.

The PMU module manages the different power modes, power mode transition, and wake-up from low-power mode. The 4K_MEM is 4KB-size SRAM and located from 0x480C_0000 memory space. Even in the lowest power mode, the content of 4K_MEM can be retained so it can be used to store critical application data. A 32-bit RTC is included in the AON domain. Refer to the RTC chapter for a detailed description.

5.2.3.1

Ultra Low-Power Comparator

The low-power comparator can operate in differential mode and in single-ended mode.

In differential mode, comp_in_n (which is connected to GPIO_26) and comp_in_p (which is connected to GPIO_27) are compared to each other by the comparator. In single-ended mode, the comparator compares reference voltage GPIO_27 input. The reference voltage is controlled by the PMIP_CMP_CTRL.COMP_REF_SEL bit registers. When using the comparator in differential mode, both GPIO_26 and GPIO_27 have to be in hi-Z state. For single-ended mode, GPIO_27 must be in hi-Z state. It can generate an interrupt or wakeup, which is controlled by the PMIP_CMP_CTRL register.

5.2.3.2 Brownout Detection

The 88MC200 device contains VBAT brownout detection circuits. It can generate a reset when a voltage supply is below a pre-set threshold. The Cortex-M3 core and all peripherals except PMU and low-power comparator are reset by this event. The brownout reset event is disabled by default. To enable this reset event, first program the PMIP_BRNDET_VBAT register to set the brownout threshold and enable brownout circuits, then set the brownout PMU reset enable register, PMIP_BRN_CFG.

5.3 Power Modes

The 88MC200 microcontroller supports several power modes: PM0, PM1, PM2, PM3 and PM4. PM0 is the active mode. PM1/2/3/4 is the different low power mode. The 88MC200 microcontroller can be set to one of the low power modes by the PMU to optimize the power consumption.

In different power modes, the 88MC200 subsystem may be in different power states. [Table 8](#), [Table 9](#), and [Table 10](#) shows the power mode definitions of different subsystems. [Table 11](#) shows the 88MC200 system power modes.

Table 8: CORTEX-M3 Core Subsystem Power Modes

CORTEX-M3 state	RUN (C0)	IDLE (C1)	STDBY (C2)	OFF (C3)
	HCLK on, FCLK on	HCLK off, FCLK on	HCLK off, FCLK off	Power is removed
Notes			State retentive mode	

Table 9: SRAM Memory Power Modes

SRAM state	RUN (M0)	STDBY (M2)	OFF (M3)
	CLK on	PWDN	Power is removed
Notes		State retentive mode	

Table 10: Flash Memory Power Modes

FLASH state	active	STDBY	PWDN	OFF
	Ready for access	Chip select is de-asserted	Leakage reduction mode	Power is removed
Notes				

Table 11: 88MC200 MCU System Power Modes

	PM0	PM1	PM2	PM3	PM4
CORTEX-M3	C0	C1	C2	C3	C3
SRAM	M0	M0	M2	M2 ^[1]	M3
FLASH	active stdby	active stdby	PWDN	off	off
RTC	on	on	on	on	on

Table 11: 88MC200 MCU System Power Modes (Continued)

	PM0	PM1	PM2	PM3	PM4
Peripherals	on ^[2]	on ^[2]	state retentive	off	off
XTAL	on/off	on/off	on/off	off	off
SFLL	on/off	on /off	on/off	off	off
AUPLL	on/off	on /off	on/off	off	off

NOTE:

1. Only 192 KB out of 512 KB of SRAM will be in state retention mode. Refer to Table 12 for addresses of the memories in state retention mode.
2. When in PM0 and PM1 modes, users can shut off functional clocks for the peripherals that are not required for their applications via programming PMU.PERI_CLK_EN registers.

Table 12: Address of Memories Available in State Retention Mode

SYS_CTRL.CFG	CODE	SRAM
2'b00 and 2'b01 (Default : 2'b00)	0x100000-0x127FFF (160 kB)	0x10000000 – 0x20007FFF (32 kB)
2'b10	0x100000-0x127FFF (160 kB) 0x160000 – 0x167FFF (32 kB)	N/A

The PMU can set the 88MC200 device to one of the five power modes to optimize power consumption in different system configurations.

PM0 - Active Mode

The MCU enters PM0 state upon the completion of the POR reset sequence and a wakeup from PM2, PM3, and PM4 states. When in PM0 state, all internal power domains and external power supplies may be fully powered and functional. Each peripheral function clocks can be gated off via programming the PMU clock enable registers.

PM1 - Idle Mode

In PM1 mode, the clock to the Cortex-M3 core is stopped. All other on-chip functions may continue operation in idle mode. The core can be quickly reactivated and resume execution by a generated interrupt.

Entering into the PM1 mode is performed by the CORTEX-M3 core executing the WFI instruction with clearing the Cortex-M3 System Control Register SLEEPDEEP bit. The Cortex-M3 NVIC continues monitoring interrupts and wakes up the Cortex-M3 when an interrupt is detected.

PM2 – Standby Mode

PM2 state offers lower power consumption by placing the Cortex-M3 core, most of the MCU peripherals, and SRAM arrays in a low-power mode. In this mode, the core state and registers, peripheral registers, and internal SRAM values are preserved and the state of I/Os is kept. Flash memory is in PWDN mode. Marvell recommends disabling BOD before entering PM2.

Entering into PM2 state is performed by writing PMU PWR_MODE registers to the PM2 state (01). The sequence for entering PM2 state is as follows:

1. Set Cortex-M3 System Control Register SLEEPDEEP bit
2. Program PMU_CLK_SRC register to switch source clock to RC32M if system source clock is not RC32M
3. Write PMU_PERI_CLK_DIV register to change PMU clock to lowest frequency (1MHz)
4. Program PMU_SFLL_CTRL0 and ANA_GRP_CTRL0 registers to disable XTAL and PLLs

5. Program PMU WAKEUP_PUPD_CTRL register to pull up EXT_PIN PAD if choosing EXT_PIN active-low wake-up, pull down EXT_PIN PAD if choosing EXT_PIN active-high wake-up
6. Set PMUPWR_MODE registers to PM2 state
7. Execute WFI instruction

Exiting the PM2 state occurs when the PMU detects any wakeup that is enabled before entering the PM2 state. The sequence is as follows:

1. PMU detects wakeup event.
2. Cortex-M3 wakes up via interrupt
3. If corresponding bit in NVIC is set, the ISR subroutine is executed
4. Continue the instruction that follows WFI

PM3 - Sleep Mode

In PM3 mode, all the power supplies except the power supplies for AON domain and Memory modules are turned off. RTC can be still running with a 32 kHz clock and 4K_MEM in AON domain is in retention mode. 192 KB SRAM is in Retention mode (refer to table 5-5). Flash is powered off. During this mode, all functional clocks except the RTC clock are gated off. Wake-up from PM3 mode occurs when external wake-up pins or enabled RTC wakeup. Marvell recommends disabling BOD and ULP Comparator before entering PM3.

The system can be programmed to enter PM3 mode by programming the PMU.PWR_MODE.pwr_mode register to 2'b10. The sequence for entering PM3 state is as follows:

1. Set Cortex-M3 System Control Register SLEEPDEEP bit
2. Program PMU_CLK_SRC register to switch source clock to RC32M if system source clock is not RC32M
3. Program PMU_SFLL_CTRL0 and ANA_GRP_CTRL0 registers to disable XTAL and PLLs
4. Power off I/O domain (IO_PAD_PWR_CFG.POR_LVL_[domain]_LOW_VDDDB_CORE) except AON domain to save power consumption and prevent IO pads entering unknown state, and power off the regulators of all IO domains (IO_PAD_PWR_CFG.VDD_[domain]_REG_PDB_CORE) to save power consumption as well.
5. Program PMU WAKEUP_PUPD_CTRL register to pull up EXT_PIN PAD if choosing EXT_PIN active-low wake-up, pull down EXT_PIN PAD if choosing EXT_PIN active-high wake-up.
6. Program PMU PAD_CTRL0_REG to set XTAL32K_IN, XTAL32K_OUT, TDO PAD to power saving mode.¹
7. Software enables RC32 or XTAL32K if it is not enabled
8. Set PMUPWR_MODE registers to PM3 state
9. Execute WFI instruction

Note:

¹ After waking up from PM3, configure XTAL32K_IN, XTAL32K_OUT, TDO PAD to normal mode for other PINMUX function use.

Exiting the PM3 state occurs when the PMU detects any wakeups that are enabled before entering the PM3 state. The sequence is as follows:

1. PMU detects wakeup event
2. CORTEX-M3 wakes up and Fast Boot occurs

PM4 - Shutoff Mode

In PM4 mode, the power is shut off to the entire chip with the exception of the AON domain. RTC can still be running with a 32 kHz clock and 4K_MEM in retention mode. Wakeup from PM4 mode occurs when external wake-up pins or enabled RTC wakeup. Marvell suggests disabling BOD and ULP Comparator before entering PM4.

Entering into the PM4 state is performed by writing the power mode registers to the PM4 state (2'b11). The sequence to enter the PM4 state is as follows:

1. Set Cortex-M3 System Control Register DEEPSLEEP bit
2. Enable RC32K or XTAL32K clock if not already enabled
3. Program PMU_CLK_SRC register to switch source clock to RC32M if system source clock is not RC32M
4. Program PMU_SFLL_CTRL and ANA_GRP_CTRL0 registers to disable XTAL and PLLs
5. Power off all I/O domains (IO_PAD_PWR_CFG.POR_LVL_[domain]_LOW_VDDCORE) except AON domain to save power and prevent IO pads entering unknown state, and power off the regulators of all I/O domains (IO_PAD_PWR_CFG.VDD_[domain]_REG_PDB_CORE) to save power as well.
6. Program PMU_WAKEUP_PUPD_CTRL register to pull up EXT_PIN PAD if choosing EXT+PIN active-low wakeup, pulldown EXT_PIN PAD if choosing EXT_PIN active-high wakeup
7. Program PMU_PAD_CTRL0_REG to set XTAL32K_IN, XTAL32K_OUT, TDO PAD to power-saving mode¹
8. Set PMUPWR_MODE registers to the PM4 state
9. Execute WFI instruction

Exiting the PM4 state occurs when the PMU detects any wakeups that are enabled before entering PM4 state. The sequence is as follows:

1. PMU detects wakeup event.
2. Cortex-M3 wakes up and resets all except AON domain.

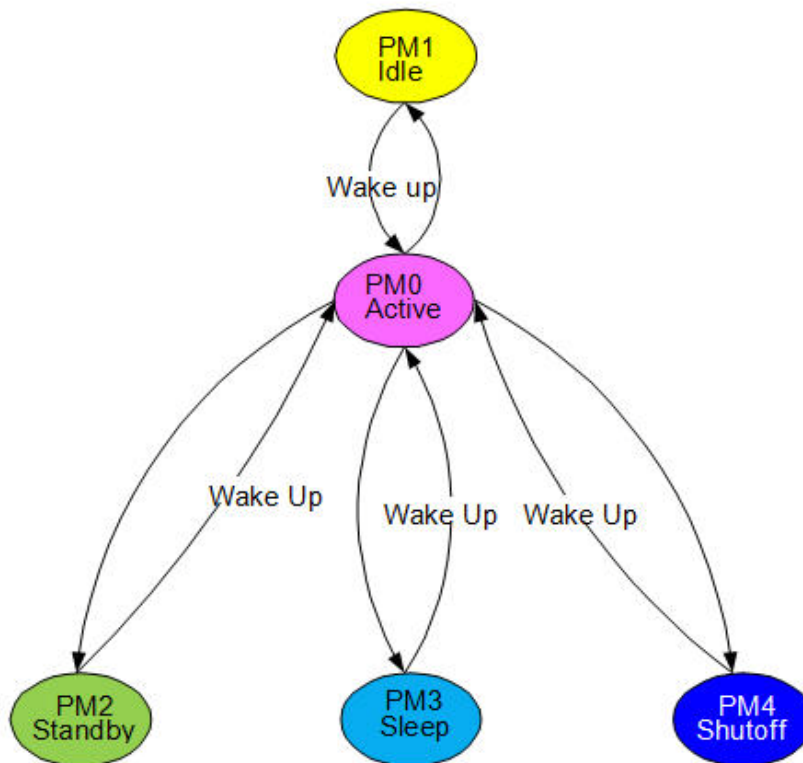
Note:

¹ After waking up from PM4, configure XTAL32K_IN, XTAL32K_OUT, TDO PAD to normal mode for other PINMUX function use.

5.4 Power Mode Transitions

The state machine for the power mode state transitions is shown in [Figure 8](#).

Figure 8: Power Mode Transitions



5.5 Wake-up Sources

5.5.1 Wake-up from PM1 Mode

Any enabled interrupt can wake up the core from Idle mode. A detailed list of all 88MC200 interrupts is located in Section 6.3 of Chapter 6.

5.5.2 Wake-up from PM2/3/4 Modes

Exiting from Standby or Sleep mode, the 88MC200 microcontroller supports the internal and external wake-up sources shown in Table 13.

The PMU supports both active-high and active-low wake-up from EXT_PIN0 and EXT_PIN1. It is programmed via the PMU WAKEUP_EDGE_DETEC registers.

Table 13: Low Power Mode Wake Up Sources

	PM2	PM3	PM4
RTC	yes	yes	yes

Table 13: Low Power Mode Wake Up Sources

EXT_PIN0	yes	yes	yes
EXT_PIN1	yes	yes	yes
ULP_COMP	yes	no	no

5.5.3

Reset Controller

The 88MC200 MCU has the following reset sources:

- POR (power on reset) – PMU detects power supply ramping from 0 volt to V_{BAT} voltage level. Entire SoC is reset in this event.
- One dedicated PIN reset – PMU detects a PIN reset. Entire SoC is reset in this event.
- Low power mode exit reset – When PMU detects wakeup while in PM3 and PM4 modes, it resets Cortex-M3 core and all peripherals except for AON domain.
- Warm reset - Cortex-M3 core and all peripherals are reset except PMU, low-power comparator, and core debug logic:
 - Triggered when LOCKUP.
 - PMIP brownout reset – The reset is generated when the voltage level of VBAT is detected as dropping below a pre-set threshold level. Brownout source must be enabled for it to function.
 - Cortex-M3-triggered soft reset.
 - WDT timeout.

5.6

Clock Controller

5.6.1

Overview

The clock controller unit controls system source clock, clock frequencies for Cortex-M3, AHB and APB bus clocks, and all peripheral function clocks. The 88MC200 microcontroller includes 4 different clock sources and 2 PLLs:

- MAINXTAL - External crystal oscillator (4-50 MHz)
- XTAL32K - External crystal oscillator 32.768 kHz
- RC32M - Internal RC32M
- RC32K - Internal RC32K
- SFLL - System PLL
- AUPLL – Audio/USB PLL

Two clock sources are external: a 4-50 MHz crystal oscillator and a 32.768 kHz crystal oscillator. The other two are internal RCs: 32 MHz (approximate) clock and 32 kHz clock.

There are two PLLs: SFLL and AUPLL. The SFLL generates a maximum of 200 MHz clock to support the Cortex-M3 core, AHB bus clocks, and most of peripherals. Programmable dividers divide the 200 MHz clock to support all peripheral function clocks as well as APB bus clocks. AUPLL is used to generate clock for audio and USB modules.

Users can switch the clock source to internal the RC32M clock or the 2-50 MHz oscillator via programming the PMU clock source select registers for low performance applications. Dynamic source clock change is supported between RC32M and MAINXTAL or RC32M and SFLL. Switching source clocks between MAINXTAL and SFLL is not allowed, and vice versa directly. Users should always switch to RC32M first. All peripheral function clocks can be shut off via the peripheral clock-enable registers if they are not used for the application. [Table 14](#) has a list of all the 88MC200 clock sources.

5.6.2 Clock Sources

Table 14: 88MC200 Clock Sources

Clock Name	Frequency (MHz)	Description
MAINTXTAL	4 - 50	External Xtal
RC32M	32 (+/- 50%)	onchip RC OSC, before calibration
	32 (+/- 2%)	onchip RC OSC, after calibration
RC32K	32k (+/- 50%)	onchip RC OSC, before calibration
	32k (+/- 2%)	onchip RC OSC, after calibration
XTAL32K	32.768k	External Xtal
SFLL	200	Output frequency is programmable
AUPLL	60	For USB
	Audio bit clock	programmable

Figure 9: High-Level Clocking Diagram

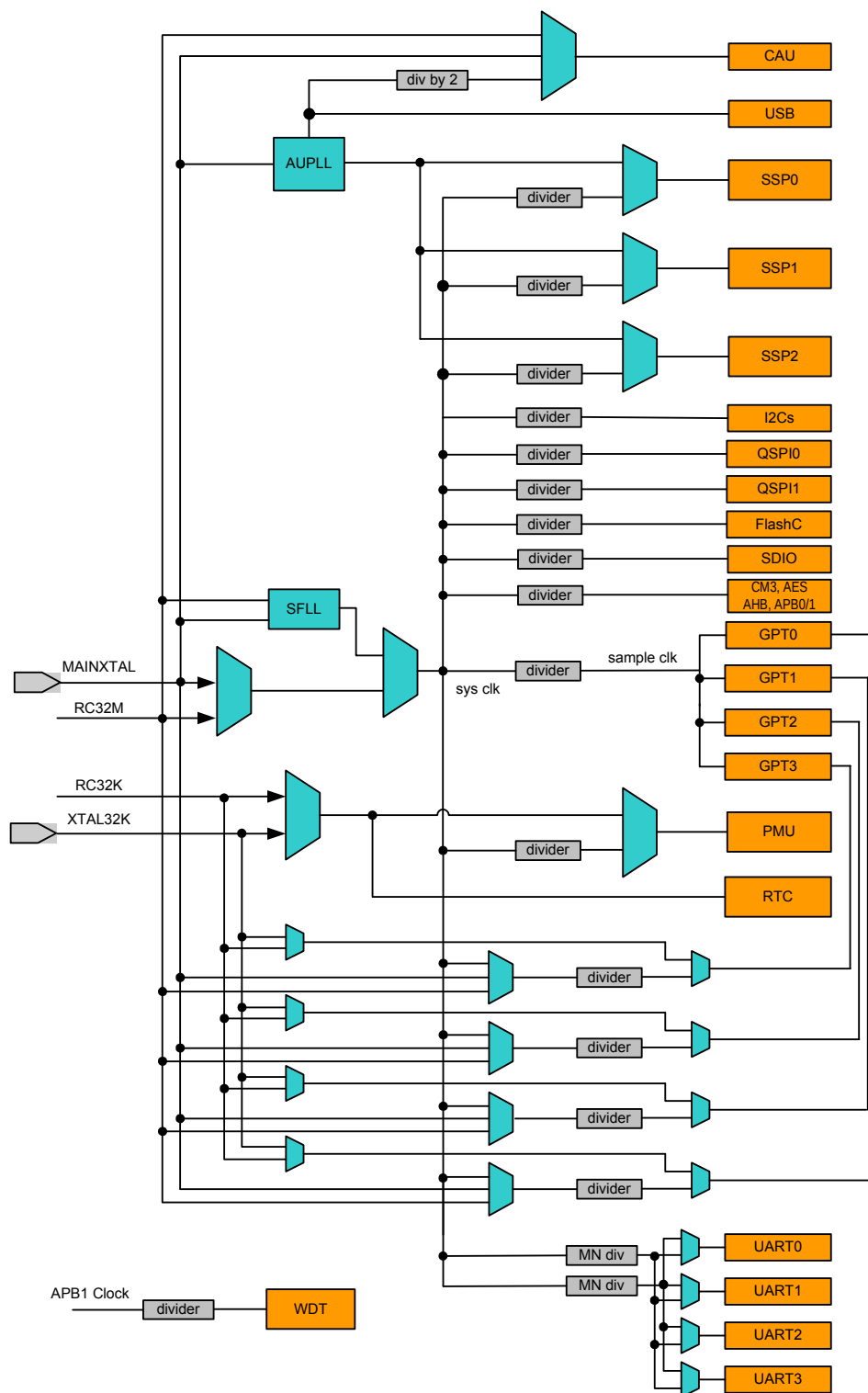


Table 15: MCU Clock Frequency Table

Module	Max Frequency (MHz)	Clock source	Programmable divider
CORTEX-M3 HCLK	200	SFLL	YES
CORTEX-M3 FCLK	200	SFLL	YES
AHB BUS	200	SFLL	YES
APB1 BUS	50	SFLL	YES
APB0 BUS	50	SFLL	YES
Memory	200	SFLL	YES
AES/CRC	200	SFLL	YES
USB	60	AUPLL	NO
SSP	25	SFLL/AUPLL	YES
SSP Audio	24.587	AUPLL	YES
UART	58.9	SFLL	YES
SDIO	50	SFLL	YES
GPT	50	SFLL	YES
RTC	32 KHz	OSC / RC32K	NO
QSPI	50	SFLL	YES
I2C	100	SFLL	YES
CAU	32	AUPLL/MAINXTAL/RC32M	YES

5.6.3 SFLL

SFLL is the main source clock for 88MC200 fast system clock. The output frequency can be programmed via PMU SFLL_CTRL0 and SFLL_CTRL1 registers.

$$SFLL \text{ output frequency} = (\text{reference clock frequency} * FBDIV) / REFDIV$$

To guarantee SFLL works properly, program PLL_READY_DET_LOW and PLL_READY_DET_HIGH fields in the ANA_GRP_CTRL0 registers, and KVCO fields in the PMU SFLL_CTRL0 registers. Refer to the PMU register description for details.

5.6.4 Cortex-M3 Core Clock and Bus Clock

The PMU clock control unit generates clocks for Cortex-M3 core, as well as AHB and APB bus clocks for the 88MC200 MCU. All those clocks are always from the same clock source and core HCLK, FCLK, and AHB bus clocks are always running at the same frequency. APB bus clock supports 1:1, 2:1, 4:1, and 8:1 divider ratio. Program the PMU.CLK_SRC register to select the source clock from RC32M, SFLL, and MAINXTAL. Program the PMU.MCU_CORE_CLK_DIV register to divide the frequency from the source clock. APB bus clock divider ratio is controlled by the PMU.PER11_CLK_DIV register. [Table 16](#) and [Table 17](#) contain a list of the APB clock divider ratios for different values of the PMU.PER11_CLK_DIV register bits.

Table 16: APB0 Bus Clock Divider Ratio

PER1_CLK_DIV[17:16]	APB0 CLK Divider Ratio
00	1:1
01	2:1
10	4:1
11	8:1

Table 17: APB1 Bus Clock Divider Ratio

PER1_CLK_DIV[19:17]	APB1 CLK Divider Ratio
00	1:1
01	2:1
10	4:1
11	8:1

5.6.5 UART Clocks

Select the UART frequency via the PMU.UART_CLK_SEL register. Two programmable fractional dividers generate the preferred UART clock frequencies. Change fractional divisors via programming the nominator and denominator fields in the PMU.UART_FAST_CLK_DIV and PMU.UART_SLOW_CLK_DIV registers based on source clock frequency, which is selected by the PMU.CLK_SRC register to obtain the preferred UART clock frequency. See [Table 18](#).

Table 18: UART Slow and Fast Clock Programming

UART_FAST_CLK_DIV, UART_SLOW_CLK_DIV	
Bit [10:0]	denominator
Bit [23:11]	numerator

The relation between source clock and output clock frequencies is as follows:

$$\frac{\text{Nominator}}{\text{Denominator}} = \frac{\text{Source\_clock}}{\text{output clock}}$$

5.6.6 AUPLL for Audio Clock and USB Clock

The 88MC200 MCU has a dedicated AUPLL to generate the audio bit clock for the SSP module and 60 MHz USB clock. When SSP works in I2S mode, the I2S audio clock can be from AUPLL only. Program the REFDIV and FBDIV fields in PMU.AUPLL_CTRL0 to obtain the necessary PLL VCO frequency. This PLL supports only 540 MHz and 600 MHz VCO frequencies.

$$\text{FVCO} = (\text{REFCLK} / \text{REFDIV}) * \text{FBDIV}$$

Program the POSTDIV_USB field in the PMU.AUPLL_CTRL2 register based on the VCO output frequency to obtain a 60 MHz output clock for USB.

Table 19: USB Clock Programming

FVCO	POSTDIV_USB	DIV
540 MHz	0	9
600 MHz	1	10

Audio bit clock output is controlled by the PMU.AUPLL_CTRL2 register.

Table 20: Audio Clock Programming

FVCO (MHz)	FREQ_OFFSET [15:0]	POSTDIV_AUDIO [6:0]	CLKOUT_AUDIO
540	d'1839	96	1,411,200
		64	2,116,800
		48	2,822,400
		32	4,233,600
		24	5,644,800
		16	8,467,200
		12	11,289,600
		8	16,934,400
		6	22,579,200
		4	33,868,800
		3	45,158,400
600	d'9045	36	4,096,000
		24	6,144,000
		18	8,192,000
		12	12,288,000
		9	16,384,000
		8	18,432,000
		6	24,576,000
		4	36,864,000
		3	49,152,000

AUPLL is disabled after power on reset. It is important to set the preferred REFDIV, FBDIV, POSTDIV, and OFFSET before setting the AUPLL power up bit in PMU.AUPLL_CTRL0 register.

5.6.7 CAU Clock

The PMU provides the CAU main clock which is used for ADC, DAC, and ACOMP. This clock can be programmed to select the following clocks as source clocks via the PMU.CAU_CLK_SEL register.

- RC32M

- MAINXTAL
- 30 MHz clock generated from 60 MHz AUPLL output

Maximum CAU clock frequency is 32 MHz. MAINXTAL cannot be selected as the CAU clock when its frequency exceeds 32 MHz. Users can turn off this clock via the PMU CAU Clock Gate register.

5.6.8 GPT Clock

The PMU provides the clock source for the GPT. When the CLK_SRC bit in the CLK_CNTL register of GPT is set to 0, the GPT selects the clock source from the PMU. To enable the GPTx clock, set to 0 the GPTx_CLK_EN bit in the PERI_CLK_EN register. The GPT clock can be programmed to select from the following clocks via GPTx_CLK_SEL0 (x = 0, 1, 2, 3) and GPTx_CLK_SEL1 bits in the GPTx_CTRL register of PMU module:

- System clock
- RC32M
- MAINXTAL
- XTAL32K
- RC32K

The clock can be divided if the system clock/RC32M/MAINXTAL is selected. For GPT0, GPT1 and GPT2, the clock can be divided through the GPTx_CLK_DIV bits in the GPTx_CTRL register of PMU module (x = 0, 1, 2). For GPT3, the clock can be divided through GPT3_CLK_DIV_2_0 and GPT3_CLK_DIV_5_3 in the PERI2_CLK_DIV register.

5.6.8.1 GPT Sampling Clock

When GPT is in the input function, the PMU provides the sampling clock to GPT to sample input signals. The sampling clock source is the system clock and it can be divided through the GPT_SAMPLE_CLK_DIV bits in the PERI2_CLK_DIV register.

5.6.9 Clock Output

RC32M and XTAL32K can be output through the corresponding GPIO pins. GPIO_74 outputs RC32M when it is set to its Function 2 of PINMUX. Refer to Section 3.1 Pinmux Alternate Functions for details.

GPIO_25, GPIO_26, and GPIO_27 output XTAL32K when setting the corresponding bits in register PAD_CTRL1_REG to 1 as shown in Table 21. At that time, the GPIOs cease their default PINMUX function. Instead, they are set as XTAL32K clock output.

Table 21: XTAL32K Output Bits in Register PAD_CTRL1_REG of PMU

Pin	Bit Name	Value	Description
GPIO_25	WAKEUP0_CTR	0	Function configured in PINMUX
		1	XTAL32K output
GPIO_26	WAKEUP1_CTR	0	Function configured in PINMUX
		1	XTAL32K output
GPIO_27	GPIO_27_CTRL	0	Function configured in PINMUX
		1	XTAL32K output

5.7 Register Description

A detailed description of the PMU registers is located in Appendix A.



88MC200 Microcontroller Datasheet

6

Memory Map, Interrupts and AHB Bus Fabric

6.1 Overview

This chapter provides a detailed description of the system memory map, interrupts and the AHB bus fabric of the Marvell 88MC200 system.

6.2 Memory Map

The Marvell 88MC200 microcontroller includes 4 kB Boot ROM, 4 kB SRAM in AON domain, and 512 kB of SRAM on-chip. It also contains an in-package 8Mbit serial flash memory. The ROM and serial flash are allocated on the CODE space. The 512 kB SRAM is allocated to the CODE and SRAM space.

The Cortex M3 CPU accesses the CODE memory space using the ICODE or DCODE AHB bus interface, and accesses the SRAM space with the SYS AHB bus interface.

The 512 kB SRAM memory consists of four segments: RAM0, RAM1, RAM2 and RAM3. RAM0 and RAM1 are 192 kB each, and RAM2 and RAM3 are 64 kB each. In the default configuration, RAM0 and RAM1 are part of the CODE space and RAM2 and RAM3 are part of the SRAM space. 192 kB of SRAM memory can be in Retention mode even in PM3 low-power mode. With this 192 kB retention SRAM, the chip can implement the fast wakeup from PM3 mode.

The 4kB SRAM in the AON domain is mapped to the peripheral address space and begins at address 0x480C_0000. The on-chip peripherals are mapped to the peripheral address space. Accessing reserved portions of the peripheral address space does not cause a data abort or error response but does provide undetermined data.

[Figure 10](#) and [Table 22](#) show the system memory map with SRAM in default configuration.

Figure 10: System Memory Map Diagram

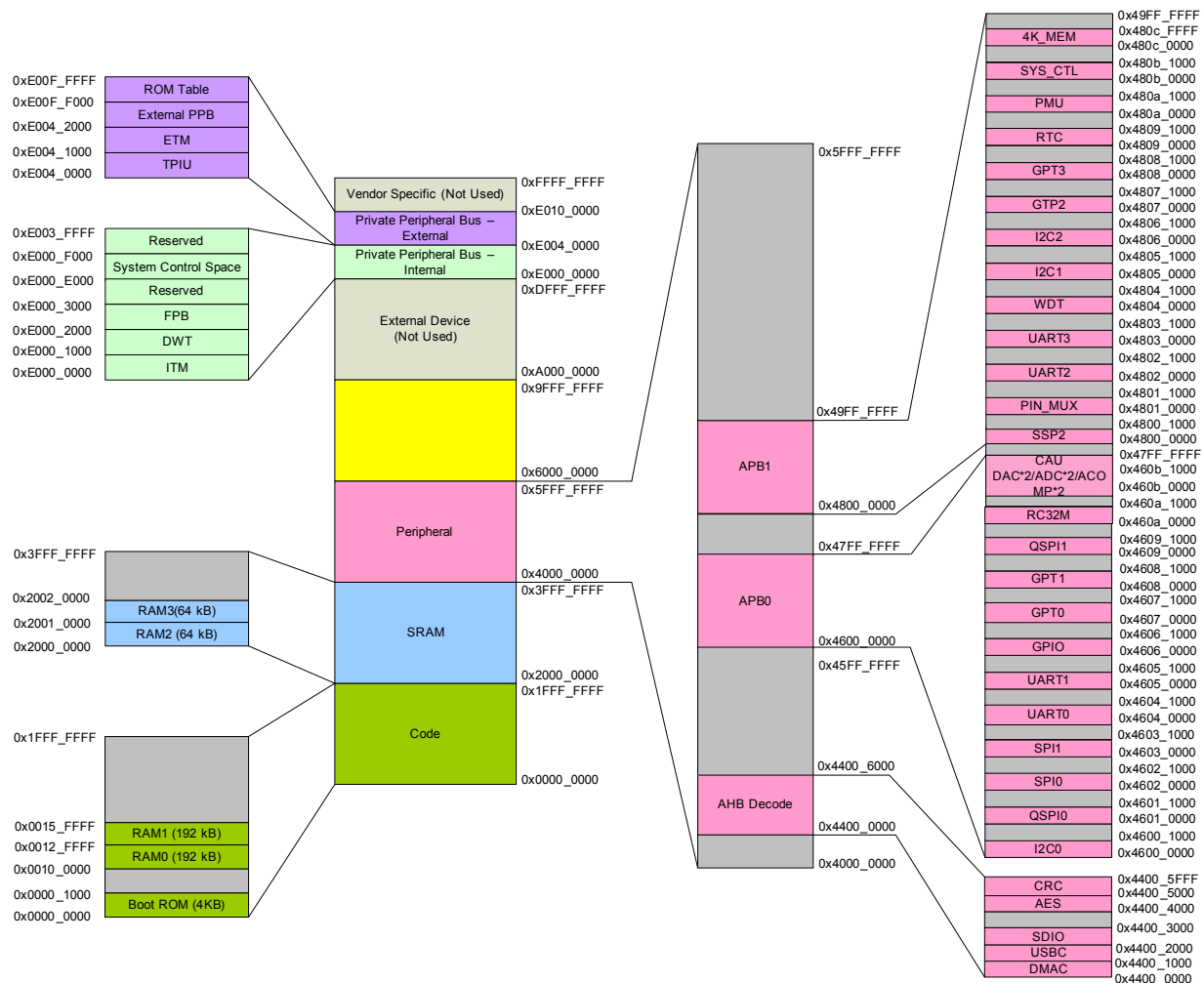


Table 22: System Address Memory Map

Module	Start Address	End Address
ROM	0x0000_0000	0x0000_0FFF
CODE RAM (configurable; default: 384kB)	0x0010_0000	0x0015_FFFF
SRAM RAM (configurable; default: 128 kB)	0x2000_0000	0x2001_FFFF
DMAC	0x4400_0000	0x4400_0FFF
USBC	0x4400_1000	0x4400_1FFF
SDIO	0x4400_2000	0x4400_2FFF
FLASHC	0x4400_3000	0x4400_3FFF
AES	0x4400_4000	0x4400_4FFF
CRC	0x4400_5000	0x4400_5FFF
I2C0	0x4600_0000	0x4600_0FFF
QSPI0	0x4601_0000	0x4601_0FFF
SSP0	0x4602_0000	0x4602_0FFF
SSP1	0x4603_0000	0x4603_0FFF
UART0	0x4604_0000	0x4604_0FFF
UART1	0x4605_0000	0x4605_0FFF
GPIO	0x4606_0000	0x4606_0FFF
GPT0	0x4607_0000	0x4607_0FFF
GPT1	0x4608_0000	0x4608_0FFF
QSPI1	0x4609_0000	0x4609_0FFF
RC32M	0x460A_0000	0x460A_0FFF
ADC0	0x460B_0000	0x460B_0FFF
ADC1	0x460B_1000	0x460B_01FF
DAC	0x460B_0200	0x460B_02FF
ACOMP	0x460B_0300	0x460B_03FF
SSP2	0x4800_0000	0x4800_0FFF
Pin Mux	0x4801_0000	0x4801_0FFF
UART2	0x4802_0000	0x4802_0FFF
UART3	0x4803_0000	0x4803_0FFF
Watchdog Timer	0x4804_0000	0x4804_0FFF
I2C1	0x4805_0000	0x4805_0FFF
I2C2	0x4806_0000	0x4806_0FFF
GPT2	0x4807_0000	0x4807_0FFF
GPT3	0x4808_0000	0x4808_0FFF
RTC	0x4809_0000	0x4809_0FFF
PMU	0x480A_0000	0x480A_0FFF
SYS_CTL	0x480B_0000	0x480B_0FFF
4k_MEM	0x480C_0000	0x480C_0FFF

In the 88MC200 system, RAM1 and RAM2 memories are reconfigurable to the SRAM or CODE space respectively by the SYS_CTRL.CFG register. As default, RAM1 is mapped to the CODE space and RAM2 is mapped to the SRAM space. The SYS_CTRL.CFG register can be programmed to increase the accessible CODE space or SRAM space based on the system requirement by remapping RAM2 to the CODE space or RAM1 to the SRAM space, respectively.

A detailed description of the memory configuration register is located in Appendix Section 2.

Accesses to unmapped addresses of RAM1 and RAM2 are provided with an error response. Writes to the ROM space, if any, also yield an error response.

Table 23 shows the memory map for on-chip SRAM in different configurations. 192 kB of the 512KB SRAM can be in Retention mode in PM3 low-power mode. 160 kB retention SRAM is located in the CODE space, starting from address 0x0010_0000 regardless of the memory configuration. The location of the other 32 kB retention SRAM changes according to the different memory configuration. When SYS_CTRL.CFG is set to 2'b00 or 2'b01, the 32 kB retention SRAM is located in the SRAM space, starting from address 0x2000_0000. When SYS_CTRL.CFG is set to 2'b10, the 32 kB retention SRAM is relocated in the CODE space, starting from address 0x0016_0000.

Table 23: Memory Map for On-chip SRAM

SYS_CTRL.MEM.CFG	CODE	SRAM
2'b00	RAM0: 0x100000-0x12FFFF (192 kB) RAM1: 0x130000-0x15FFFF (192 kB)	RAM2: 0x20000000 – 0x2000FFFF (64kB) RAM3: 0x20010000 – 0x2001FFFF (64kB)
2'b01	RAM0: 0x100000-0x12FFFF (192 kB)	RAM2: 0x20000000 – 0x2000FFFF (64kB) RAM3: 0x20010000 – 0x2001FFFF (64kB) RAM1: 0x20020000 - 0x2004FFFF (192 kB)
2'b10	RAM0: 0x100000-0x12FFFF (192 kB) RAM1: 0x130000-0x15FFFF (192 kB) RAM2: 0x160000 – 0x16FFFF (64kB)	RAM3: 0x20010000 – 0x2001FFFF (64kB)

The Marvell 88MC200 system has an 8Mbit in-package flash. All accesses to addresses outside 8Mbit will be provided with undetermined data.

6.3 Interrupts

The Marvell 88MC200 device can accept 64 external interrupts through the NVIC module in the Cortex M3 processor. The interrupts are listed in Table 24.

Table 24: External Interrupts

Name	Source	Type	Polarity	Map
WD Timeout	WDT	Level	Active High	INTNMI
LOCKUP	Cortex-M3			
Ext. Pin 0	External	Configurable	Active High	INTIRQ[0]
Ext. Pin 1	External	Configurable	Active High	INTIRQ[1]
RTC INT	RTC	Level	Active High	INTIRQ[2]
CRC INT	CRC	Level	Active High	INTIRQ[3]
AES INT	AES	Level	Active High	INTIRQ[4]
I2C0 INT	I2C 0	Level	Active High	INTIRQ[5]
I2C1 INT	I2C 1	Level	Active High	INTIRQ[6]
I2C2 INT	I2C 2	Level	Active High	INTIRQ[7]
DMAC INT	DMAC	Level	Active High	INTIRQ[8]
GPIO INT	GPIO	Level	Active High	INTIRQ[9]
SSP0 INT	SSP 0	Level	Active High	INTIRQ[10]
SSP1 INT	SSP 1	Level	Active High	INTIRQ[11]
SSP2 INT	SSP 2	Level	Active High	INTIRQ[12]
QSPI0 INT	QSPI0	Level	Active High	INTIRQ[13]
GPT0 INT	GPT 0	Level	Active High	INTIRQ[14]

Table 24: External Interrupts (Continued)

Name	Source	Type	Polarity	Map
GPT1 INT	GPT 1	Level	Active High	INTIRQ[15]
GPT2 INT	GPT 2	Level	Active High	INTIRQ[16]
GPT3 INT	GPT 3	Level	Active High	INTIRQ[17]
UART0 INT	UART 0	Level	Active High	INTIRQ[18]
UART1 INT	UART 1	Level	Active High	INTIRQ[19]
UART2 INT	UART 2	Level	Active High	INTIRQ[20]
UART3 INT	UART 3	Level	Active High	INTIRQ[21]
WDT INT	WDT	Level	Active High	INTIRQ[22]
ADC1 INT	CAU	Level	Active High	INTIRQ[23]
ADC0 INT	CAU	Level	Active High	INTIRQ[24]
DAC INT	CAU	Level	Active High	INTIRQ[25]
ACOMP WKUP INT	CAU	Level	Active High	INTIRQ[26]
ACOMP INT	CAU	Level	Active High	INTIRQ[27]
SDIO INT	SDIO	Level	Active High	INTIRQ[28]
USB INT	USB	Level	Active High	INTIRQ[29]
Reserved	Reserved	Reserved	Reserved	INTIRQ[30]
PLL INT	PMU	Level	Active High	INTIRQ[31]
QSPI1 INT	QSPI1	Level	Active High	INTIRQ[32]
RC32M INT FUNC	RC32M	Level	Active High	INTIRQ[33]
Ext. Pin int	PMU	Level	Active High	INTIRQ[59:34]
ULP_COMP	PMU	Level	Active High	INTIRQ[60]

External pin interrupts connected to INTIRQ [59:34] are generated using GPIOs in the design by programming the PMU.EXT_SEL_REGx register bits to the required value. The mapping of the GPIOs to generate external interrupts is shown in [Table 25](#). Two bits from the PMU.EXT_SEL_REG0 or PMU.EXT_SEL_REG1 are used to select the GPIO connected to the external interrupt on Cortex-M3. A detailed description of the ext_sel_reg value is located in Appendix Section 3.

Table 25: GPIO Mapping to External Interrupts

External Interrupt Bit	GPIO Connected	Ext Sel Reg Value
34	GPIO[0]	ext_sel_reg0[1:0] = 00
	GPIO[1]	ext_sel_reg0[1:0] = 01
	GPIO[2]	ext_sel_reg0[1:0] = 10 or 11
35	GPIO[3]	ext_sel_reg0[3:2] = 00
	GPIO[4]	ext_sel_reg0[3:2] = 01
	GPIO[5]	ext_sel_reg0[3:2] = 10 or 11
36	GPIO[6]	ext_sel_reg0[5:4] = 00
	GPIO[7]	ext_sel_reg0[5:4] = 01
	GPIO[8]	ext_sel_reg0[5:4] = 10 or 11

Table 25: GPIO Mapping to External Interrupts (Continued)

External Interrupt Bit	GPIO Connected	Ext Sel Reg Value
37	GPIO[9]	ext_sel_reg0[7:6] = 00
	GPIO[10]	ext_sel_reg0[7:6] = 01
	GPIO[11]	ext_sel_reg0[7:6] = 10 or 11
38	Reserved	Reserved
39	Reserved	ext_sel_reg0[11:10] = 00
	GPIO[16]	ext_sel_reg0[11:10] = 01
	GPIO[17]	ext_sel_reg0[11:10] = 10 or 11
40	GPIO[18]	ext_sel_reg0[13:12] = 00
	GPIO[19]	ext_sel_reg0[13:12] = 01
	GPIO[20]	ext_sel_reg0[13:12] = 10 or 11
41	GPIO[21]	ext_sel_reg0[15:14] = 00
	GPIO[22]	ext_sel_reg0[15:14] = 01
	GPIO[23]	ext_sel_reg0[15:14] = 10 or 11
42	GPIO[24]	ext_sel_reg0[17:16] = 00
	GPIO[28]	ext_sel_reg0[17:16] = 01
	GPIO[29]	ext_sel_reg0[17:16] = 10 or 11
43	GPIO[30]	ext_sel_reg0[19:18] = 00
	Reserved	ext_sel_reg0[19:18] = 01
	GPIO[32]	ext_sel_reg0[19:18] = 10 or 11
44	GPIO[33]	ext_sel_reg0[21:20] = 00
	GPIO[34]	ext_sel_reg0[21:20] = 01
	GPIO[35]	ext_sel_reg0[21:20] = 10 or 11
45	Reserved	Reserved
46	Reserved	ext_sel_reg0[25:24] = 00
	GPIO[40]	ext_sel_reg0[25:24] = 01
	GPIO[41]	ext_sel_reg0[25:24] = 10 or 11
47	GPIO[42]	ext_sel_reg0[27:26] = 00
	GPIO[43]	ext_sel_reg0[27:26] = 01
	GPIO[44]	ext_sel_reg0[27:26] = 10 or 11
48	GPIO[45]	ext_sel_reg0[29:28] = 00
	Reserved	ext_sel_reg0[29:28] = 01
	Reserved	ext_sel_reg0[29:28] = 10 or 11
49	Reserved	ext_sel_reg0[31:30] = 00
	Reserved	ext_sel_reg0[31:30] = 01
	GPIO[50]	ext_sel_reg0[31:30] = 10 or 11

Table 25: GPIO Mapping to External Interrupts (Continued)

External Interrupt Bit	GPIO Connected	Ext Sel Reg Value
50	GPIO[51]	ext_sel_reg1[1:0] = 00
	GPIO[52]	ext_sel_reg1[1:0] = 01
	GPIO[53]	ext_sel_reg1[1:0] = 10 or 11
51	GPIO[54]	ext_sel_reg1[3:2] = 00
	GPIO[55]	ext_sel_reg1[3:2] = 01
	GPIO[56]	ext_sel_reg1[3:2] = 10 or 11
52	GPIO[57]	ext_sel_reg1[5:4] = 00
	GPIO[58]	ext_sel_reg1[5:4] = 01
	GPIO[59]	ext_sel_reg1[5:4] = 10 or 11
53	GPIO[60]	ext_sel_reg1[7:6] = 00
	GPIO[61]	ext_sel_reg1[7:6] = 01
	GPIO[62]	ext_sel_reg1[7:6] = 10 or 11
54	GPIO[63]	ext_sel_reg1[9:8] = 00
	GPIO[64]	ext_sel_reg1[9:8] = 01
	GPIO[65]	ext_sel_reg1[9:8] = 10 or 11
55	GPIO[66]	ext_sel_reg1[11:10] = 00
	Reserved	ext_sel_reg1[11:10] = 01
	GPIO[68]	ext_sel_reg1[11:10] = 10 or 11
56	Reserved	Reserved
57	GPIO[72]	ext_sel_reg1[15:14] = 00
	GPIO[73]	ext_sel_reg1[15:14] = 01
	GPIO[74]	ext_sel_reg1[15:14] = 10 or 11
58	GPIO[75]	ext_sel_reg1[17:16] = 00
	GPIO[76]	ext_sel_reg1[17:16] = 01
	GPIO[77]	ext_sel_reg1[17:16] = 10 or 11
59	GPIO[78]	ext_sel_reg1[19:18] = 00
	GPIO[79]	ext_sel_reg1[19:18] = 01
	GPIO[79]	ext_sel_reg1[19:18] = 10 or 11

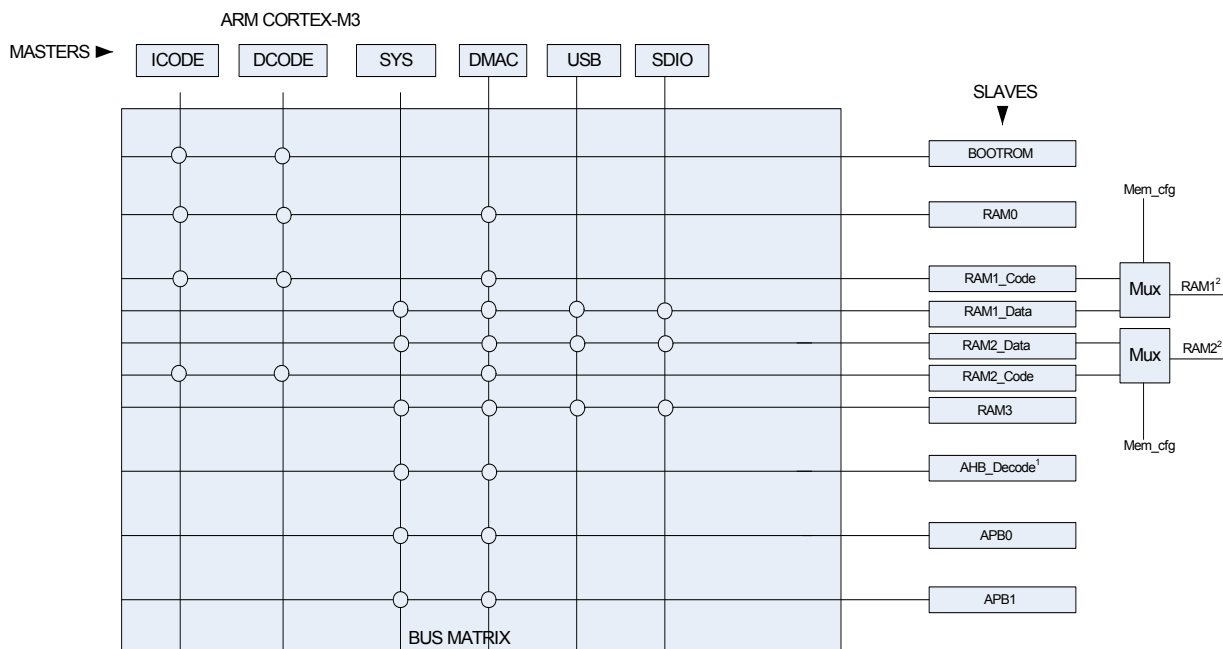
6.4 AHB Bus Fabric

The Marvell 88MC200 AHB Bus Matrix is a low latency bus matrix which enables parallel access to a number of shared AHB slaves from a number of different AHB masters. The bus matrix routes the control data signals between the masters and slaves based on the address map specified in Section 6.2 Memory Map. The sparse connectivity feature allows for the configuration of only the necessary master – slave connections, thus providing reduced area and multiplexer delays.

The AHB Bus Fabric in the 88MC200 system connects six masters and nine slaves. The master ports connected to the bus matrix include ICODE, DCODE and SYSTEM bus from CORTEX M3, DMA Controller, USB Controller and SDIO Controller. The slave ports connected to the bus matrix include the BOOTROM, Flash memory, RAM0, RAM1, RAM2, RAM3, APB0, APB1 (APB0 and

APB1 contain the instances of the APB peripherals in the system) and AHB Decode (decodes addresses to the various AHB peripherals in the system). RAM1 and RAM2 can be configured to be part of the code or data memory. Masters connected to these memories vary based on whether the memory is in the CODE or SRAM space. Therefore, there are separate slave ports for each on the AHB Bus Fabric.

Figure 11: Bus Matrix Interconnection



Note:

¹AHB Decode maps to registers in DMAC, USBC, SDIO, AES-CRC.

² RAM1 and RAM2 are based on memory configuration. They can be either part of the code space or data space. By default, RAM1 is in the code space and RAM2 is in the data space. For a detailed information of configuration, please refer to Section 6.2, *Memory Map*.

The interconnection diagram in [Figure 9](#) shows the connection between the various masters and slaves in the system. APB0 and APB1 are top level blocks that contain the APB peripherals. The AHB Decode block maps to registers in the DMA Controller, USB Controller, SDIO Controller, AES and CRC blocks. RAM1 and RAM2 are selected to be in the CODE space or SRAM space using the SYS_CTRL.CFG register (0x480B0004).

A detailed description of the SYS_CTRL.CFG register is located in Appendix A.



88MC200 Microcontroller Datasheet

7 Direct Memory Access Controller (DMA)

7.1 Overview

Direct memory access (DMA) is used to transfer data between peripherals and memory as well as memory to memory without CPU actions.

This DMA module has eight channels to manage the data transfer between memory and peripherals. Only DMA can serve as a flow control device.

7.2 Features

- Eight independently dedicated channels
- Non-memory peripheral devices can request a DMA transfer through hardware or software handshaking interface
- Programmable channel priority
- Single FIFO per channel for source and destination; each channel FIFO depth is 16x32 bits
- Maximum burst transaction size is 16, and maximum block size in source transfer width is 1023
- Programmable source and destination addresses; address increment, decrement, or no change
- Five interrupt sources with flags:
 - Block Transfer Complete Interrupt
 - Destination Transaction Complete Interrupt
 - Error Interrupt
 - Source Transaction Complete Interrupt
 - DMA Transfer Complete Interrupt

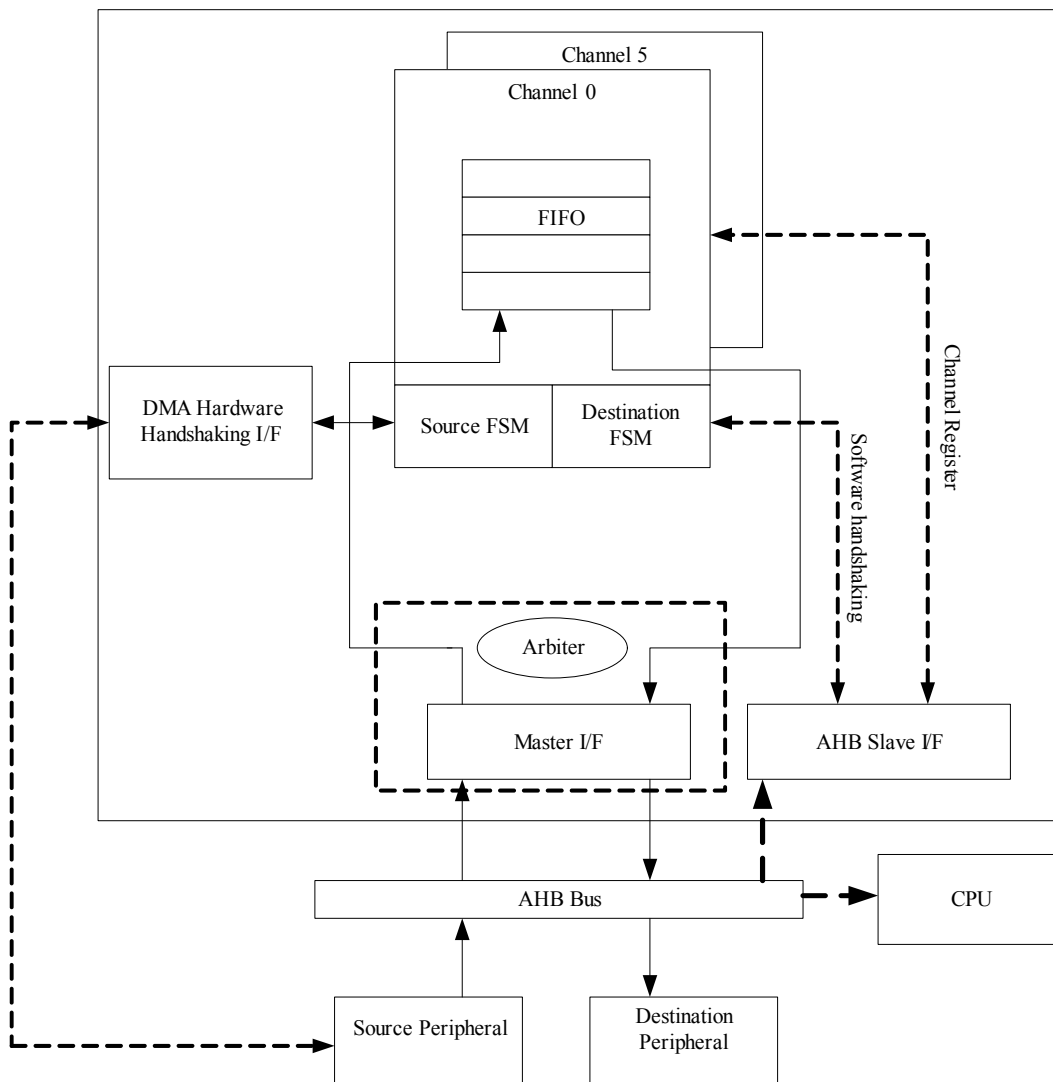
7.2.1 DMA Operation

7.2.2 DMA Block Diagram

One channel of the DMA is required for each source/destination pair. In the most basic configurations, the DMA has one master interface and one channel. The master interface reads the data from a source peripheral and writes it to a destination peripheral. Two AHB transfers are required for each DMA data transfer; this is also known as a *dual-access transfer*.

Figure 12 shows the functional groupings of the main interfaces to the DMAC block.

Figure 12: DMA Block Diagram



7.2.3 Basic Definitions

The following terms are concise definitions of the DMA concepts used throughout this chapter:

- **Source peripheral** – Device from which the DMA reads data. The DMA then stores the data in the Channel FIFO. The source peripheral teams up with a destination peripheral to form a channel.
- **Destination peripheral** – Device to which the DMA writes the stored data from the FIFO (previously read from the source peripheral).
- **Channel** – Read/write data path between a source peripheral and a destination peripheral

- **Block** – Block of DMA data, the amount of which is the block length. For transfers between the DMA and memory, a block is broken directly into a sequence of bursts and single transfers. For transfers between the DMA and a non-memory peripheral, a block is broken into a sequence of DMA transactions (single and bursts).
- **Transaction** – Basic unit of a DMA transfer. A transaction is relevant only for transfers between the DMA and a source or destination peripheral if the peripheral is a non-memory device. There are two types of transactions:
 - Single transaction – Length of a single transaction is always 1.
 - Burst transaction – Length of a burst transaction is programmed into the DMA. The burst transaction is converted into a sequence of bursts.
- **FIFO mode** – Special mode to improve bandwidth. When enabled, the channel waits until the FIFO is less than half full to fetch the data from the source peripheral, and waits until the FIFO is greater than or equal to half full in order to send data to the destination peripheral. Because of this mode, the channel can transfer the data using bursts, which eliminates the need to arbitrate in each single AHB transfer. When this mode is not enabled, the channel waits only until the FIFO can transmit or accept a single AHB transfer before it requests.

7.2.4 Peripheral Burst Transaction Requests

For a source FIFO, an active edge is triggered on a DMA request when the source FIFO exceeds some watermark level. For a destination FIFO, an active edge is triggered on a DMA request when the destination FIFO drops below some watermark level. This section investigates the optimal settings of these watermark levels on the source and destination peripherals and their relationship to, respectively:

- Source transaction length, DMA.CTLx.SRC_MSIZEx
- Destination transaction length, DMA.CTLx.DEST_MSIZEx

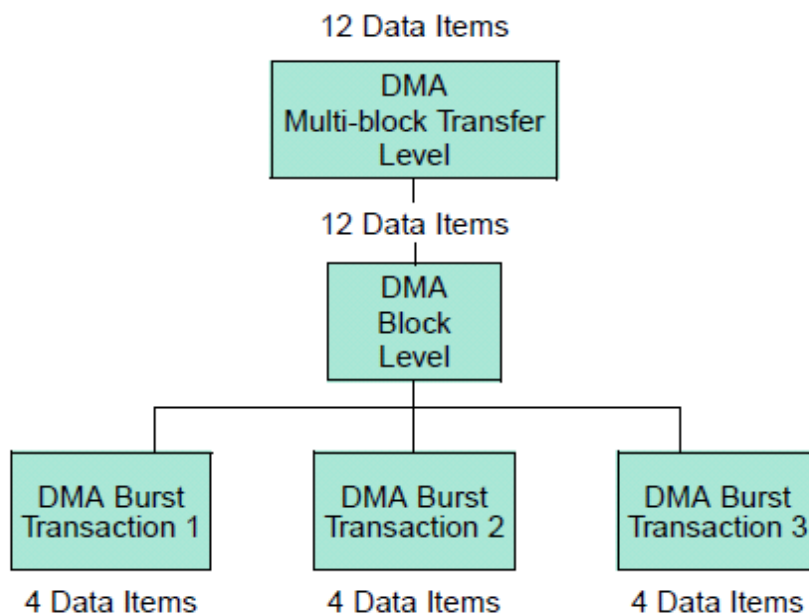
For demonstration purposes, a Receive I²C is used as a source peripheral, and a Transmit I²C is used as a destination peripheral.

As a block flow-control device, the DMA Controller is programmed by the processor with the number of data items (block size) that are to be transmitted or received by the I²C; this is programmed into the BLOCK_TS field of the dmac CTLx register.

The block is broken into a number of transactions, each initiated by a request from the I²C. The DMA Controller must also be programmed with the number of data items (in this case, I²C FIFO entries) to be transferred for each DMA request. This is also known as the *burst transaction length* and is programmed into the SRC_MSIZEx/DEST_MSIZEx fields of the Dmac CTLx register for source and destination, respectively.

Figure 13 shows a single block transfer, where the block size programmed into the DMA Controller is 12 and the burst transaction length is set to 4. In this case, the block size is a multiple of the burst transaction length. Therefore, the DMA block transfer consists of a series of burst transactions. If the I²C generates a transmit request to this channel, four data items are written to the I²C TX FIFO. Similarly, if the I²C generates a receive request to this channel, four data items are read from the I²C RX FIFO. Three separate requests must be made to this DMA channel before all 12 data items are written or read.

Figure 13: Breakdown of DMA Transfer into Burst Transactions



Block Size: DMA.CTLx.BLOCK_TS = 12

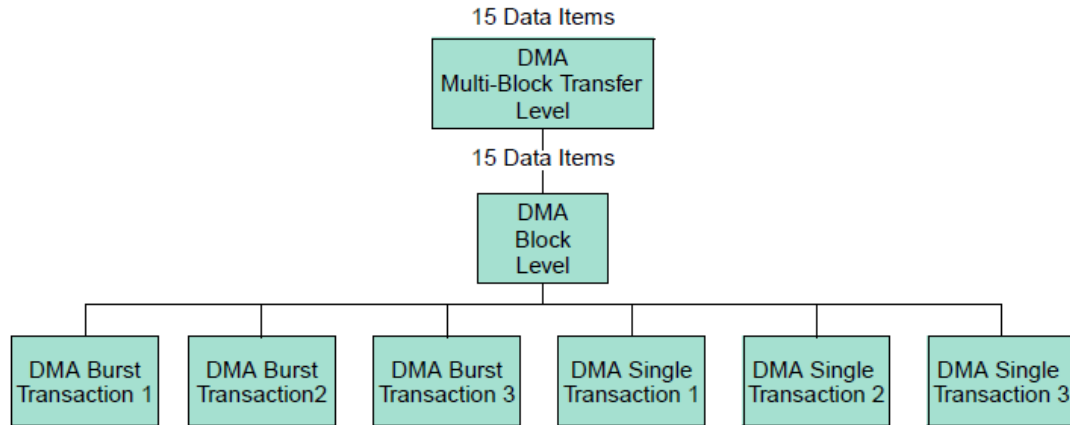
Number of data items per source burst transaction: DMA.CTLx.SRC_MSIZ = 4

I²C receive FIFO watermark level:

I2C.IIC_DMA_RDLR + 1 = DMA.CTLx.SRC_MSIZ = 4

When the block size programmed into the DMA Controller is not a multiple of the burst transaction length, as shown in [Figure 14](#), a series of burst transactions followed by single transactions are needed to complete the block transfer.

Figure 14: Breakdown of DMA Transfer into Single and Burst Transactions



Block Size: DMA.CTLx.BLOCK_TS = 15

Number of data items per burst transaction: DMA.CTLx.DEST_MSIZ = 4

I²C transmit FIFO watermark level:

I2C.IIC_DMA_TDLR = DMA.CTLx.DEST_MSIZ = 4

7.2.4.1 Watermark Level and Transmit FIFO Underflow

During I²C serial transfers, transmit FIFO requests are made to the DMAC whenever the number of entries in the transmit FIFO is less than or equal to the DMA Transmit Data Level Register(IC_DMA_TDLR) value; this is known as the *watermark level*. The DMAC responds by writing a burst of data to the transmit FIFO buffer, of length CTLx.DEST_MSIZ. Data should be fetched from the DMA often enough for the transmit FIFO to perform serial transfers continuously; that is, when the FIFO begins to empty, another DMA request should be triggered. Otherwise, the FIFO can run out of data causing a STOP to be inserted on the I²C bus. Set the watermark level correctly to avoid this incident.

7.2.4.2 Choosing the Transmit Watermark Level

Consider an example where it is assumed that:

$$\text{DMA.CTLx.DEST\_MSIZ} = \text{FIFO\_DEPTH} - \text{I2C.IC\_DMA\_TDLR}$$

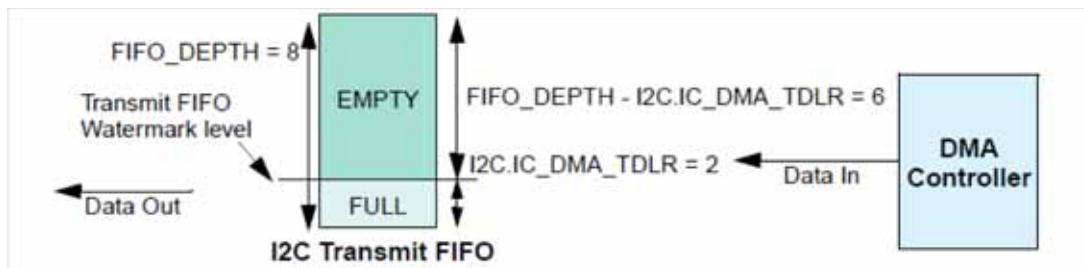
Here, the number of data items to be transferred in a DMA burst is equal to the empty space in the Transmit FIFO. Consider two different watermark level settings.

Case 1: IC_DMA_TDLR = 2

- Transmit FIFO watermark level = I2C.IC_DMA_TDLR = 2
- DMA.CTLx.DEST_MSIZ = FIFO_DEPTH - I2C.IC_DMA_TDLR = 6
- I²C transmit FIFO_DEPTH = 8
- DMA.CTLx.BLOCK_TS = 30

See [Figure 15](#) for a graphic representation of this example.

Figure 15: Case 1 Watermark Levels where IC_DMA_TDLR = 2



Therefore, the number of burst transactions needed equals the block size divided by the number of data items per burst:

$$\text{DMA.CTLx.BLOCK\_TS} / \text{DMA.CTLx.DEST\_MSIZE} = 30 / 6 = 5$$

The number of burst transactions in the DMA block transfer is 5. But the watermark level, I2C.IC_DMA_TDLR, is quite low. Therefore, the probability of an I²C underflow is high where the I²C serial transmit line must transmit data, but where there is no data remaining in the transmit FIFO. This situation occurs because the DMA has not had time to service the DMA request before the transmit FIFO becomes empty.

Case 2: IC_DMA_TDLR = 6

Transmit FIFO watermark level = I2C.IC_DMA_TDLR = 6

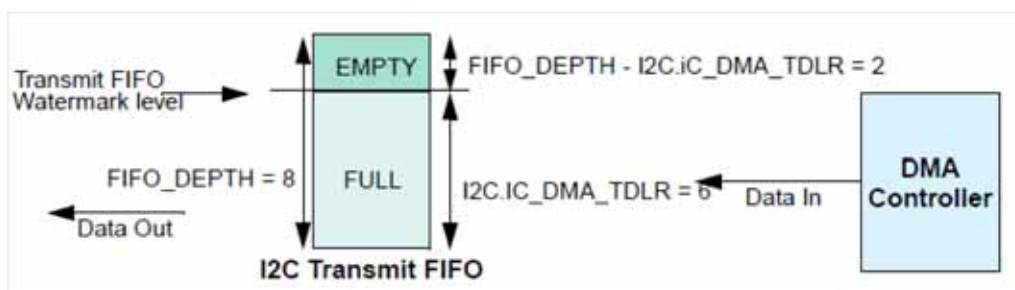
DMA.CTLx.DEST_MSIZE = FIFO_DEPTH - I2C.IC_DMA_TDLR = 2

I²C transmit FIFO_DEPTH = 8

DMA.CTLx.BLOCK_TS = 30

See [Figure 16](#) for a graphic representation of this example.

Figure 16: Case 2 Watermark Levels where IC_DMA_TDLR = 6



Number of burst transactions in Block:

$$\text{DMA.CTLx.BLOCK\_TS} / \text{DMA.CTLx.DEST\_MSIZE} = 30 / 2 = 15$$

In this block transfer, there are 15 destination burst transactions in a DMA block transfer. But the watermark level, I2C.IC_DMA_TDLR, is high. Therefore, the probability of an I²C underflow is low

because the DMA controller has plenty of time to service the destination burst transaction request before the I²C transmit FIFO becomes empty.

Thus, the second case has a lower probability of underflow at the expense of more burst transactions per block. This situation provides a potentially greater amount of AMBA bursts per block and worse bus utilization than the former case.

Therefore, the goal in choosing a watermark level is to minimize the number of transactions per block, while at the same time keeping the probability of an underflow condition to an acceptable level. In practice, this is a function of the ratio of the rate at which the I²C transmits data to the rate at which the DMA can respond to destination burst requests.

For example, promoting the channel to the highest priority channel in the DMA, and promoting the DMA master interface to the highest priority master in the AMBA layer, increases the rate at which the DMA controller can respond to burst transaction requests. This in turn allows the user to decrease the watermark level, which improves bus utilization without compromising the probability of an underflow occurring.

7.2.4.3 Selecting DEST_MSIZ and Transmit FIFO Overflow

As can be seen from [Figure 7-3](#), programming DMA.CTLx.DEST_MSIZ to a value greater than the watermark level that triggers the DMA request may cause overflow when there is not enough space in the I²C transmit FIFO to service the destination burst request. Therefore, the following equation must be adhered to in order to avoid overflow:

$$\text{DMA.CTLx.DEST\_MSIZ} \leq \text{I2C.FIFO\_DEPTH} - \text{I2C.IC\_DMA\_TDLR} \quad (1)$$

In Case 2: IC_DMA_TDLR = 6, the amount of space in the transmit FIFO at the time the burst request is made equals the destination burst length: DMA.CTLx.DEST_MSIZ. Thus, the transmit FIFO may be full, but not overflowed, at the completion of the burst transaction.

Therefore, for optimal operation, set DMA.CTLx.DEST_MSIZ at the FIFO level that triggers a transmit DMA request; that is:

$$\text{DMA.CTLx.DEST\_MSIZ} = \text{I2C.FIFO\_DEPTH} - \text{I2C.IC\_DMA\_TDLR} \quad (2)$$

Adhering to equation (2) reduces the number of DMA bursts needed for a block transfer, and this in turn improves AMBA bus utilization.



Note

The transmit FIFO will not be full at the end of a DMA burst transfer if the I²C has successfully transmitted one data item or more on the I²C serial transmit line during the transfer.

7.2.4.4 Receive Watermark Level and Receive FIFO Overflow

During I²C serial transfers, receive FIFO requests are made to the DMAC whenever the number of entries in the receive FIFO is at or above the DMA Receive Data Level Register; that is, IC_DMA_RDLR+1 (again, known as the watermark level). The DMAC responds by writing a burst of data to the transmit FIFO buffer of length CTLx.SRC_MSIZ.

Data should be fetched by the DMA often enough for the receive FIFO to accept serial transfers continuously; that is, when the FIFO begins to fill, another DMA transfer is requested. Otherwise, the FIFO can fill with data (overflow). Set the watermark level correctly to prevent this condition from occurring.

7.2.4.5 Choosing the Receive Watermark Level

Similar to choosing the transmit watermark level described earlier, the receive watermark level, set IC_DMA_RDLR+1 to minimize the probability of overflow, as shown in [Figure 17](#). It is a trade-off

between the number of DMA burst transactions required per block versus the probability of an overflow occurring.

7.2.4.6 Selecting SRC_MSIZ and Receive FIFO Underflow

As can be seen in [Figure 17](#), programming a source burst transaction length greater than the watermark level may cause underflow when there is not enough data to service the source burst request. Therefore, Equation 3 must be adhered to avoid underflow.

If the number of data items in the receive FIFO is equal to the source burst length at the time the burst request is made – DMA.CTLx.SRC_MSIZ – the receive FIFO may be emptied, but not underflowed, at the completion of the burst transaction. For optimal operation, set DMA.CTLx.SRC_MSIZ at the watermark level; that is:

$$\text{DMA.CTLx.SRC\_MSIZ} = \text{I2C.IC\_DMA\_RDLR} + 1 \quad (3)$$

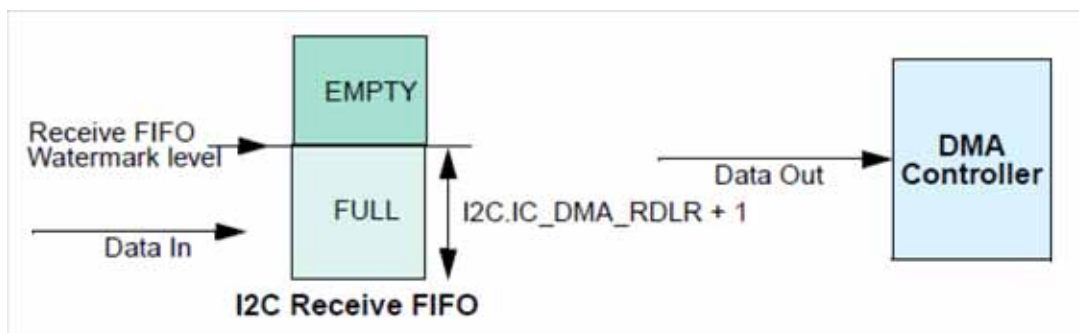
Adhering to Equation (3) reduces the number of DMA bursts in a block transfer, which in turn can avoid underflow and improve AMBA bus utilization.



Note

The receive FIFO will not be empty at the end of the source burst transaction if the I²C has successfully received one data item or more on the I²C serial receive line during the burst.

Figure 17: I²C Receive FIFO



7.2.5 Interrupt

For each channel, DMA has five types of interrupt sources

- IntBlock – Block Transfer Complete Interrupt. This interrupt is generated on DMA block transfer completion to the destination peripheral.
- IntDstTran – Destination Transaction Complete Interrupt. This interrupt is generated after completion of the last AHB transfer of the requested single/burst transaction from the handshaking interface (either the hardware or software handshaking interface) on the destination side.



Note

If the destination for a channel is memory, then that channel never generates the IntDstTran interrupt. Therefore, the corresponding bit in this field is not set.

- IntErr – Error Interrupt. This interrupt is generated when an ERROR response is received from an AHB slave on the HRESP bus during a DMA transfer. In addition, the DMA transfer is cancelled and the channel is disabled.
- IntSrcTran – Source Transaction Complete Interrupt. This interrupt is generated after completion of the last AHB transfer of the requested single/burst transaction from the handshaking interface (either the hardware or software handshaking interface) on the source side.
- IntTfr – DMA Transfer Complete Interrupt. This interrupt is generated on DMA transfer completion to the destination peripheral.

7.2.6 DMA Channel Mapping

DMA handshake mapping can refer to the DMA handshake mapping register in the Cortex-M3 Core and System Control chapter.

7.2.7 Operation Mode

A typical software flow for DMA configuration for transfer is outlined as follows:

1. Read the Channel Enable register to choose a free (disabled) channel.
2. Clear any pending interrupts on the channel from the previous DMA transfer by writing to the Interrupt Clear registers: DMA.CLEARTRF, DMA.CLEARBLOCK, DMA.CLEARSRCTRAN, DMA.CLEARSTTRAN, and DMA.CLEARERR. Reading the Interrupt Raw Status and Interrupt Status Registers confirms that all interrupts have been cleared.
3. Program the following channel registers:
 - a) Write the starting source address in the DMA.SARx register for channel x
 - b) Write the starting destination address in the DMA.DARx register for channel x
 - c) Program DMA.CTLx and DMA.CFGx
 - d) Write the control information for the DMA transfer in the DMA.CTLx register for channel x

For example, in the register, you can program the following:

 - i. Set up the transfer type (memory or non-memory peripheral for source and destination) by programming the TT_FC of the DMA.CTLx register.
 - ii. Set up the transfer characteristics, such as:
 - Transfer width for the source in the SRC_TR_WIDTH field.
 - Transfer width for the destination in the DST_TR_WIDTH field.
 - Incrementing/decrementing or fixed address for the source in the SINC field.
 - Incrementing/decrementing or fixed address for the destination in the DINC field.
- e) Write the channel configuration information into the DMA.CFGx register for channel x
 - i. Designate the handshaking interface type (hardware or software) for the source and destination peripherals; this is not required for memory. This step requires programming the HS_SEL_SRC/HS_SEL_DST bits, respectively. Writing a 0 activates the hardware handshaking interface to handle source/destination requests. Writing a 1 activates the software handshaking interface to handle source and destination requests.
 - ii. If the hardware handshaking interface is activated for the source or destination peripheral, assign a handshaking interface to the source and destination peripheral; this requires programming the SRC_PER and DEST_PER bits, respectively.

4. After the DMA-selected channel has been programmed, enable the channel by writing a 1 to the DMA.CHENREG.CH_EN bit. Ensure that bit 0 of the DMA.DMACFGREG register is enabled.
5. Source and destination request single and burst DMA transactions in order to transfer the block of data (assuming non-memory peripherals). The DMA acknowledges at the completion of every transaction (burst and single) in the block and carries out the block transfer.
6. Once the transfer completes, hardware sets the interrupts and disables the channel. At this time, you can respond to either the Block Complete or Transfer Complete interrupts, or poll for the transfer complete raw interrupt status register (DMA.RAWTFR[n], n = channel number) until it is set by hardware, in order to detect when the transfer is complete. Note that if this polling is used, the software must ensure that the transfer complete interrupt is cleared by writing to the Interrupt Clear register, DMA.CLEARTFR[n], before the channel is enabled.

7.3 Register Descriptions

A detailed description of the DMA registers is located in Appendix Section 4.

8

Real Time Clock (RTC)

8.1 Overview

This chapter describes the Real Time Clock (RTC). Registers are controlled via the APB bus.

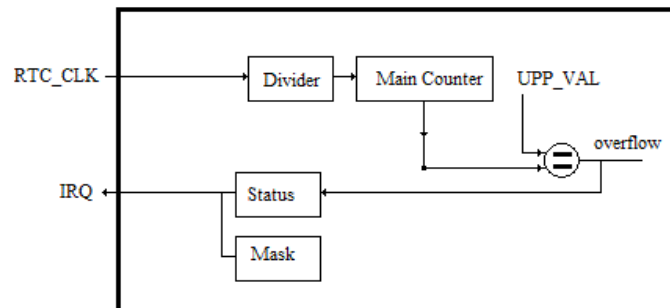
Real Time Clock is optimized for a counter in the always-on (AON) domain . It supports the following functions:

- Selectable clock source
- Programmable clock divider
- 32-bit Up counter with a programmable upper overflow boundary
- Interrupt is generated on the counter clock when it reaches the upper boundary

8.2 Functional Description

This section describes the supported RTC functions. [Figure 18](#) is the RTC block diagram.

Figure 18: RTC Block Diagram



8.2.1 Counter Clock

The clock source of the RTC comes from the PMU. It can be set to XTAL32K or RC32K through **RTC_INT_SEL** bits in **PERI_CLK_SRC** register of PMU module. To avoid any potential issues, stopping the counter is required before changing the clock source. Reset the counter after changing the clock source.

The RTC can divide the clock simultaneously. **CLK_DIV** stores the clock division factor. The clock division formula is:

$$\text{counter\_clock\_divide} = \text{counter\_clock} / (2^{\text{CLK\_DIV}})$$

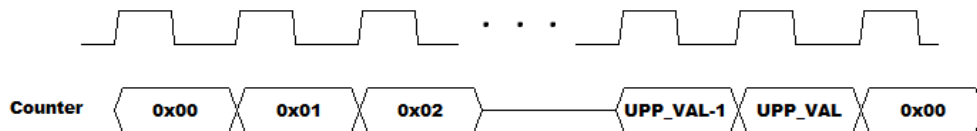
For example, if a timer clock divider register is set to 2, then the timer gets one tick every 4 clock ticks. The bit width of a clock divider register is 4, which makes the maximum value of **CLK_DIV** as 15 and the maximum division ratio as 32768:1.

8.2.2 Counting Mode

The RTC counter works in a counting-up mode. UPP_VAL defines the upper boundary of the counter; default value is 0xFFFFFFFF, the maximum counter value. The lower boundary is always zero.

The RTC counter value increments until reaching the upper boundary defined by UPP_VAL (event counter-reach-upper). In the next tick, the counter resets to zero and begins counting up again. Upon a counter reset (write 1 to CNT_RESET), the counter resets to zero. A full cycle from 0 to UPP_VAL consists of UPP_VAL+1 counter ticks. Count-up mode is illustrated in [Figure 19](#).

Figure 19: Count-up Mode



8.2.3 Counter Update Mode

The counter value can be read from the APB bus through the register CNT_VAL. The update mode of CNT_VAL can be configured using CNT_UP_MOD as shown in [Table 26](#).

Table 26: Counter Update Mode

CNT_UPDT_MOD	Description
3	Reserved
2	Auto-update CNT_VAL is updated on every counter clock tick
1	Reserved
0	Update off

8.2.4 Interrupt

When the counter reaches the UPP_VAL, the CNT_UPP_INT bit in the INT_RAW register is set to 1. Interrupt status bits are always enabled to be set in the INT_RAW register. The interrupt status bit can be cleared by writing 1 to the corresponding bit in the INT_RAW register. Each interrupt status has a corresponding mask in the INT_MSK register. If the corresponding mask is set to 1, the interrupt status does not assert the interrupt. By default, all bits are masked. The INT register is the masked result of INT_RAW register. The interrupt is asserted if any of the bits in INT register is 1.

8.3 Programming Notes

8.3.1 Initialization

1. Before using the RTC, poll STS_RESETN bit to be set.
2. Program various parameters

- a) Select clock source with RTC_INT_SEL bit in PERI_CLK_SRC register of PMU module.
- b) Set counter upper value in UPP_VAL.
- c) If the counter value needs to be read out, program CNT_UPDT_MOD to 0x2. Otherwise, leave it at 0x0
3. Write 1 to CNT_RESET to reset the counter. Poll CNT_RST_DONE bit to be set to determine when the counter finishes resetting. Do not access any other registers until CNT_RST_DONE is 1.
4. Write 1 to CNT_START to start the counter. Poll CNT_RUN bit to be set to determine when the counter begins to count.

8.3.2 UPP_VAL

The value written to UPP_VAL is not valid immediately. It is not effective until the counter overflows. To make the value valid immediately, write 1 to CNT_RESET.

8.4 Register Description

A detailed description of the RTC registers is located in Appendix A.



88MC200 Microcontroller Datasheet

9

General Purpose Timers (GPT)

9.1 Overview

This chapter describes the General Purpose Timers (GPT). The 88MC200 microcontroller includes four 32-bit GPTs. Registers are controlled via the APB bus.

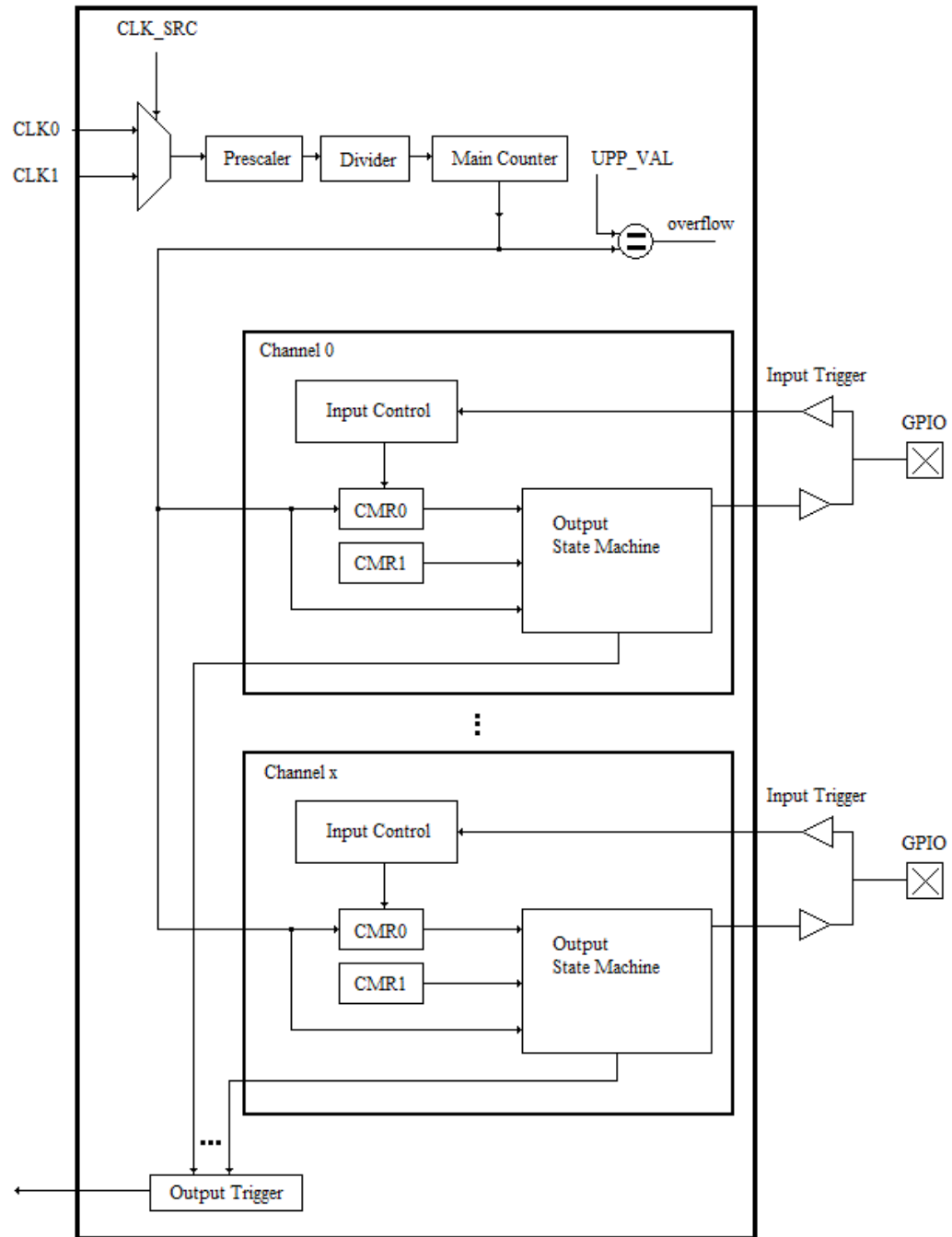
Each GPT is a multi-purpose counter that supports the following functions:

- Selectable clock source
- Programmable clock divider and pre-scalar
- 32-bit Up counter
- Six independent channels with multiple modes
- Input capture for external inputs
- Edge-aligned and Center-aligned pulse-width modulation (PWM)
- “One-shot” mode to trigger a one-time output change and interrupt
- Auto-trigger ADC/DAC module for PWM mode
- DMA transfer for input capture
- Interrupt generation on counter and channel events

9.2 Functional Description

Each timer supports as many as six channels. Each channel shares the same clock source but has a separate set of registers for configuration. In this way, each channel can serve different applications independently. The register prefix CHx_ represents that the register is for the Channel x. The structure of the GPT is shown in [Figure 20](#).

Figure 20: GPT Block Diagram



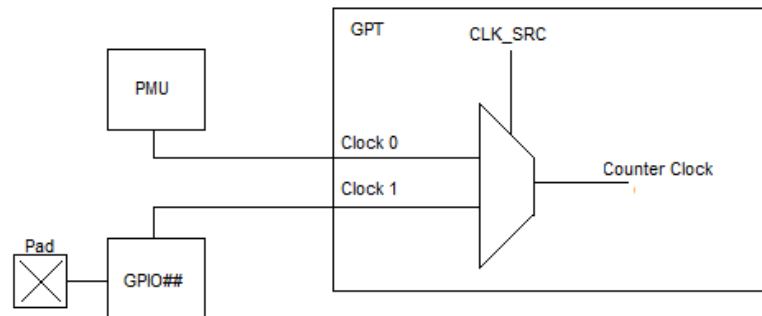
9.2.1 Counter

9.2.1.1 Counter Clock

Counter Clock Source

The clock source of the timer can be selected with CLK_SRC in CLK_CNTL register in the GPT. Two choices are available: Clock 0 (default) from PMU, and Clock 1 from the GPIO. Clock 0 can be chosen from multiple sources. Details regarding the sources of Clock 0 are in the PMU and Clocking registers description. When using Clock 1, the corresponding GPIO function must be programmed to the appropriate value, and the pad must be connected to a clean external clock. To avoid any potential issues, stopping the counter is necessary before changing the clock source. Reset the counter after changing the clock source. Figure 21 shows the clock source selection.

Figure 21: Clock Source Selection



Clock Pre-Scaling and Division

The GPT can divide and pre-scale the clock simultaneously. The combination of the divider and pre-scalar allows for many possible integer ratios within the range. CLK_PRE can linearly pre-scale the counter clock using the formula:

$$\text{counter\_clock\_prescale} = \text{counter\_clock} / (\text{CLK\_PRE} + 1)$$

Each pre-scalar has 8 bits, allowing a scaling factor from 1 to 256.

After the clock pre-scaling, the resulting clock can be further divided down. CLK_DIV stores the clock division factor. The clock division formula is:

$$\text{counter\_clock\_divide} = \text{counter\_clock\_prescale} / (2^{\text{CLK\_DIV}})$$

For example, if a timer clock divider register is set to 2, then the timer gets one tick every 4 clock ticks. The bit width of a clock divider register is 4, which makes the maximum value of CLK_DIV as 15 and the maximum division ratio as 32768:1.

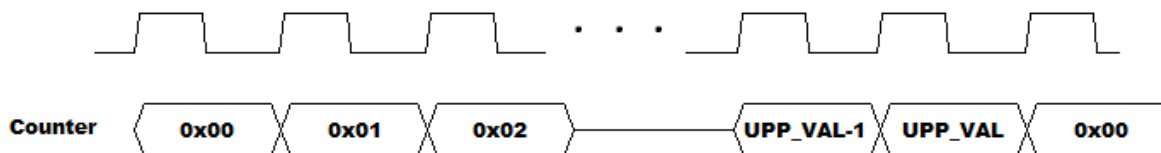
9.2.1.2 Counting Mode

The GPT always counts up. UPP_VAL defines the upper boundary of the counter. The main counter counts from 0 to UPP_VAL, overflows to 0 and continues counting. A full cycle from 0 to UPP_VAL consists of UPP_VAL+1 counter ticks. The CNT_UPP_STS status bit is set upon an overflow. Upon a count reset (write 1 to CNT_RESET), the counter resets to zero.

The value written to UPP_VAL is not valid immediately. It is not effective until the counter overflows. To make the value valid immediately, write 1 to CNT_RESET.

Count-up mode is illustrated in [Figure 22](#).

Figure 22: Count Up Mode



9.2.1.3 Counter Update Mode

The counter can be read from the APB bus through the register CNT_VAL. Updates to CNT_VAL are determined by CNT_UPDT_MOD. See [Table 27](#).

Table 27: Counter Update Mode

CNT_UPDT_MOD	Description
3	Update off. If CNT_VAL does not need to be read, CNT_UPDT_MOD can be set to off to save power.
2	Reserved
1	Auto-update fast. Used when counter clock is at least 5 times slower than the APB clock. CNT_VAL is updated on every counter clock tick.
0	Auto-update normal. Can be used for any clock relationship between the counter clock and the APB clock. Only every 3-4 counter ticks are updated to CNT_VAL.

9.2.2 Interrupt

[Table 28](#) shows the type of events that can generate interrupts.

Table 28: Available Interrupt Events

Event	Available?
Channel status	Yes
Channel error status	Yes
Reach UPP_VAL	Yes
DMA overflow	Yes

Three registers are used to control the interrupt: STS, INT, and INT_MSK. They all have corresponding bits in the same location. The status bits are in the STS register. Various events in the

timer set the status bits automatically. The status bit can be cleared by writing 1 to the corresponding bit in STS.

Each status bit has a corresponding mask in INT_MSK register. If the mask bit is set to 1, the status bit is masked and does not generate an interrupt. If the mask bit is 0, then the status bit can generate an interrupt. By default, all bits are masked.

The INT register is the masked result of the STS register. If the mask bit is 1, then the corresponding bit in the INT register is 0. If the mask bit is 0, then the corresponding bit in the INT register is the same value as that in STS register.

The interrupt is asserted if any of the bits in INT register is 1.

9.2.3 Channel Operation Modes

9.2.3.1 Counter Match Register 0 and 1 (CMR0 and CMR1)

CMR0 and CMR1 are a pair of multipurpose registers for each channel. In input-capture mode, CMR0 is used to store the captured values. In all other modes, CMR0 and CMR1 are used to determine counter parameters.

The flow of updating the values of CMR0 and CMR1 is as follows:

1. Write new values to CMR0 and CMR1
2. Write 1 to CHx_CMR_UPDT in the USER_REQ register
3. Check the value of CHx_ERR_STS, if it is 0, it means CMR0 and CMR1 are updated successfully; if it is 1, clear CHx_ERR_STS and repeat Steps 2-3

9.2.3.2 No Function Mode

Set CHx_IO to 0 to configure the channel to no function. The channel does nothing and does not set the status bit. Set unused channels to this mode to save power and avoid unpredictable behaviors.

9.2.3.3 Input Capture Mode

Set CHx_IO to 1 to configure the channel to input-capture mode.

In input-capture mode, the channel waits for one of two trigger events to occur:

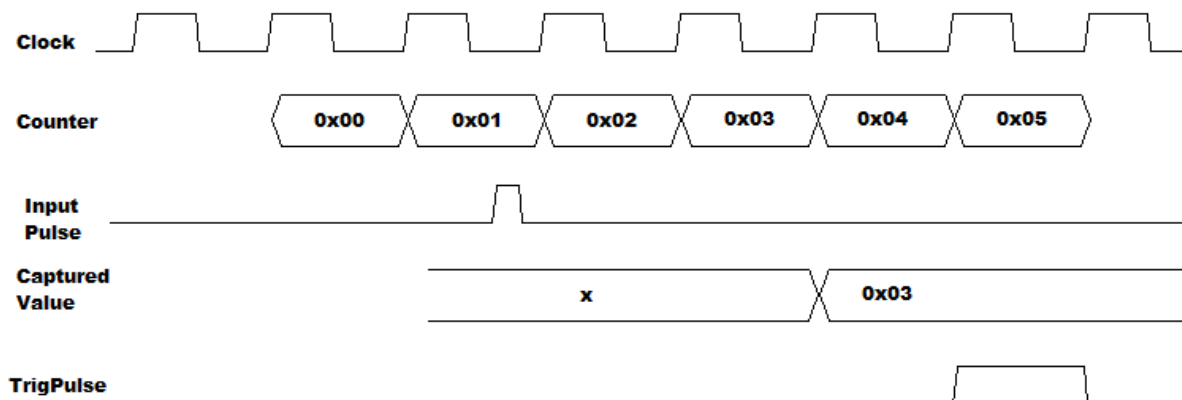
- An external trigger can come from a GPIO. The timer samples the edge transition using a fast sampling clock.
- Write to CHx_USER_ITRIG (x = 1, 2, 3, 4, 5, or 5) to generate a software trigger.

CMR0 is the capture register.

The external trigger event can be a rising or falling edge. An external trigger event is considered valid after being filtered with the settings programmed in the IC_CNTL and CHx_CNTL registers. Space external triggers sufficiently apart relative to the sampling parameters to allow sufficient time to read out the value before the next capture. Small glitches can be filtered using the input capture registers, but in general the triggers should be clean.

Each valid trigger event sets the channel status bit CHx_STS.

A valid trigger event can be generated manually by writing 1 to CHx_USER_ITRIG. This Write bypasses any sampling filters in IC_CNTL register. In this mode, during the tick where a trigger event occurs, the counter value is copied to the capture register. [Figure 23](#) illustrates an input-capture event.

Figure 23: Input Capture


DMA

In the input-capture mode, CHx_CMRO is shared as a capture register. If the captured value is required to be stored in memory by DMA, the general-purpose timer provides hardware handshake signals to automate this process. The DMA signals follow the protocol of the DMA Controller.

To enable the DMA function:

1. Set DMAz_EN (z=0,1) in the DMA_CNTL_EN register to 0
2. Select GPT channel x as the source by programming DMAz_CH = x
3. Program CHx_CNTL to set channel x to input capture
4. Set DMAz_EN to 1 to enable the DMA channel

On the DMA Controller:

1. Write to the DMA_HS register in system control module to set DMA handshake mapping
2. Set SAR to the address of the capture register (CHx_CMRO)
3. Set DAR to the memory address
4. In the CTL register
 - a) Write to SRC_TR_WIDTH and DST_TR_WIDTH to set the transfer width to 32 bits
 - b) Write to SRC_MSIZ and DEST_MSIZ to set the burst transfer length to one item
 - c) Write to TT_FC to set the transfer type to peripheral-to-memory
 - d) Write to BLOCK_TS to configure the transfer length
 - e) Set SINC to maintain source address
 - f) Set DINC to make destination address increase
5. In CFG register, set HS_SEL_SRC and HS_SEL_DST to select hardware handshaking; set SRC_PER and DST_PER to assign hardware handshaking interfaces.

9.2.3.4 One-Shot Pulse Mode

Set CH_IO to 4 to configure the channel to one-shot pulse mode. See [Table 29](#) and [Figure 24](#).

Table 29: One-Shot Pulse Control Registers

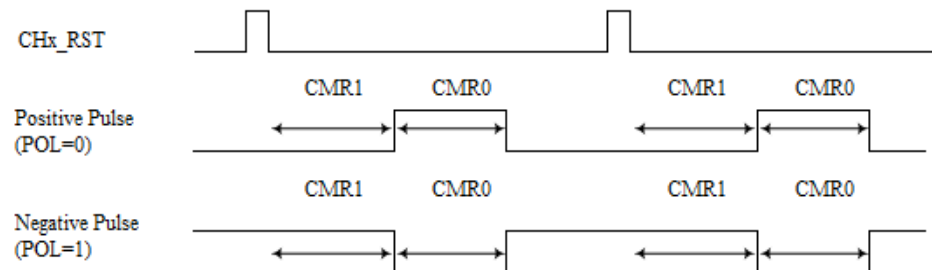
Positive Polarity (POL = 0)	Positive pulse
Negative Polarity (POL = 1)	Negative pulse
Duty Cycle	CMR0
Period	CMR0 + CMR1

- This mode generates a single pulse.
- Setting CMR1 to 0 results in an instant pulse generation.
- Setting CMR0 to 0 results in no pulse, but the status bit remains set at the end of period.

Write 1 to CHx_RST to generate one pulse:

1. After the channel reset, the output state resets to POL.
2. Wait CMR1 cycles, then change output state to the reverse value of POL.
3. Wait CMR0 cycles, then change output state to POL and set the channel status bit.

Figure 24: One-Shot Pulse



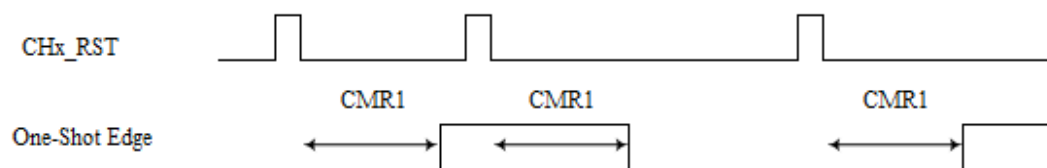
9.2.3.5 One-Shot Edge Mode

Set CH_IO to 5 to configure the channel to one-shot edge mode. This mode generates a single edge transition. Setting CMR1 to 0 results in an instant edge transition. Refer to [Figure 25](#).

Write 1 to CHx_RST to generate one edge transition:

1. Channel reset
2. Wait CMR1 cycles, then invert the current output state and set the channel status bit

Figure 25: One-Shot Edge



9.2.3.6

Pulse-Width Modulation (PWM) Edge-Aligned Mode

Set CH_IO to 6 to configure the channel to PWM edge-aligned mode. See [Table 30](#) and [Figure 26](#).

Table 30: PWM Edge-Aligned Control Registers

Positive Polarity (POL = 0)	High -> Low
Negative Polarity (POL = 1)	Low -> High
Duty Cycle	CMR0
Period	CMR0 + CMR1

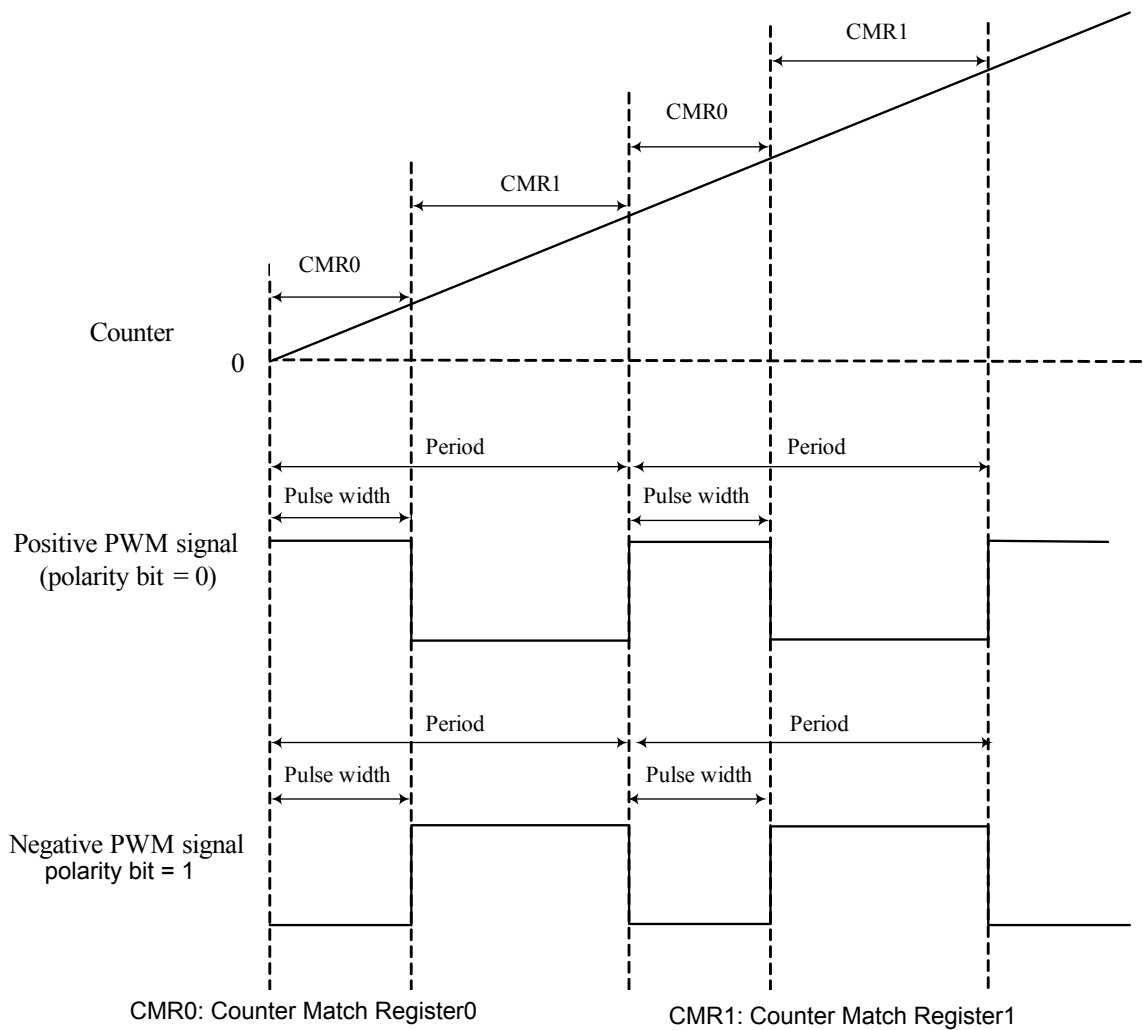
PWM edge-aligned is a periodic square waveform aligned to the starting edge of the period. To adjust the duty cycle, subtract a number from either CMR0 or CMR1 and add it to the other, thereby keeping the period the same.

Setting CMR0 to 0 results in a 0% duty cycle, and setting CMR1 to 0 results in a 100% duty cycle. Setting both CMR0 and CMR1 to 0 pauses the PWM. The output remains at the previous state and no additional interrupts are generated. To restart the PWM, set at least one CMR0 or CMR1 to a non-zero value, then write 1 to CHx_CMUR_UPDT.

The behavior of the PWM Edge-Aligned mode is as follows:

1. Change CH_IO to 6
2. Channel reset: Output state is first reset to POL
3. On the next counter tick, output state changes to the reverse value of POL
4. Wait CMR0 cycles, then set the output state to POL
5. Wait CMR1 cycles, then set the output state to the reverse value of POL and set the channel status bit
6. Repeat 4-5

Figure 26: PWM Edge-Aligned



9.2.3.7 Pulse-Width Modulation (PWM) Center-Aligned Mode

Set CH_IO to 7 to configure the channel to PWM center-aligned mode. See [Table 31](#) and [Figure 27](#).

Table 31: PWM Center-Aligned Control Registers

Positive Polarity (POL = 0)	Low -> High -> High -> Low
Negative Polarity (POL = 1)	High -> Low -> Low -> High
Duty Cycle	2 x CMR0
Period	2 x CMR0 + 2 x CMR1

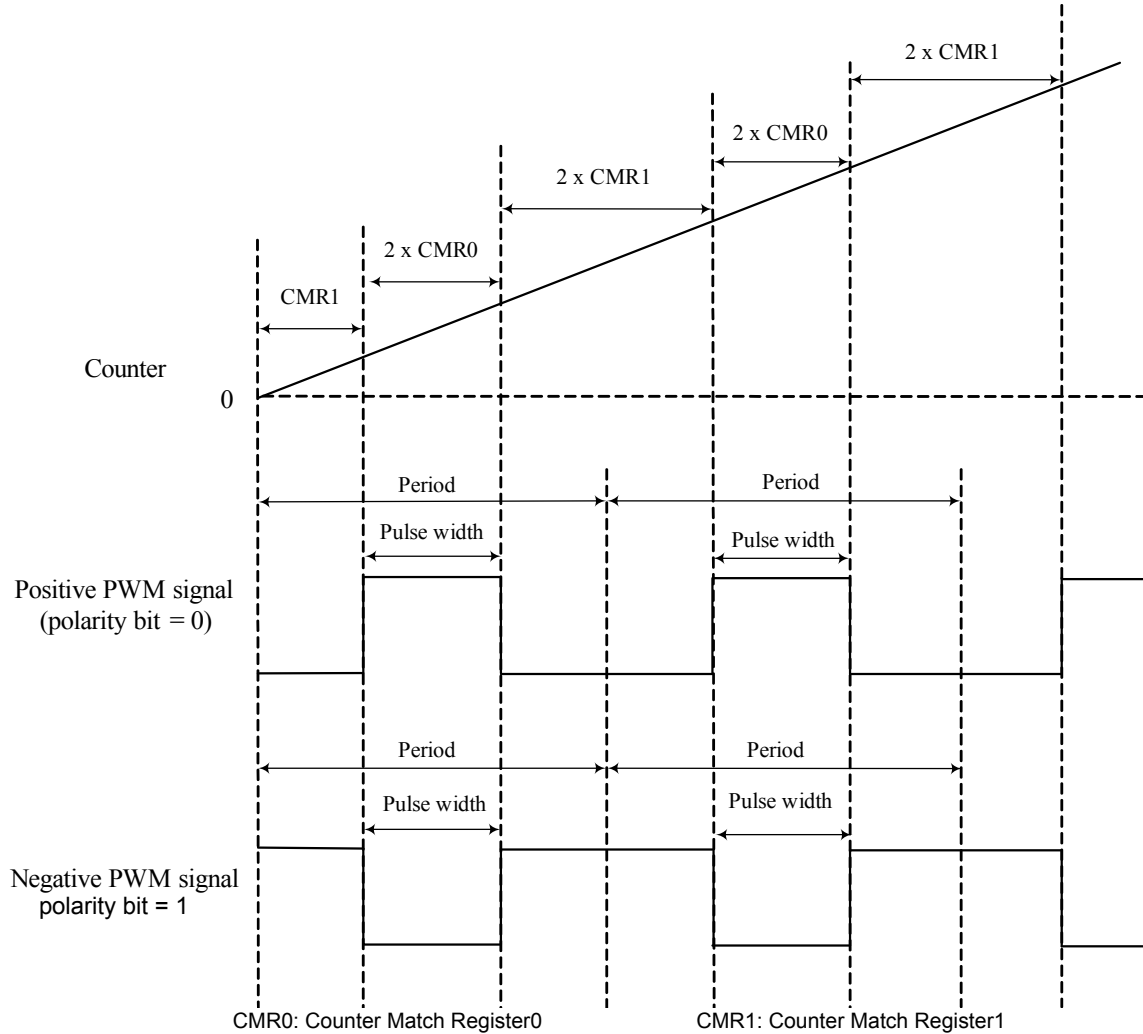
PWM center-aligned is a periodic square waveform aligned to the center of the period. To adjust the duty cycle, subtract a number from either CMR0 or CMR1 and add it to the other, thereby keeping the period the same.

Setting CMR0 to 0 results in a 0% duty cycle, and setting CMR1 to 0 results in a 100% duty cycle. Setting both CMR0 and CMR1 to 0 pauses the PWM. The output remains at the previous state and no additional interrupts are generated. To restart the PWM, set at least one CMR0 or CMR1 to a non-zero value.

The behavior of the PWM Center-Aligned mode is as follows (see [Figure 27](#)):

1. Change CH_IO to 7
2. Write 1 to CHx_RST: Output state is first reset to POL
3. Wait CMR1 cycles, then set the output state to the reverse value of POL
4. Wait 2x CMR0 cycles, then set the output state to POL
5. Wait CMR1 cycles, then set the channel status bit
6. Repeat steps 3, 4, and 5

Figure 27: PWM Center-Aligned



9.2.4 ADC Trigger

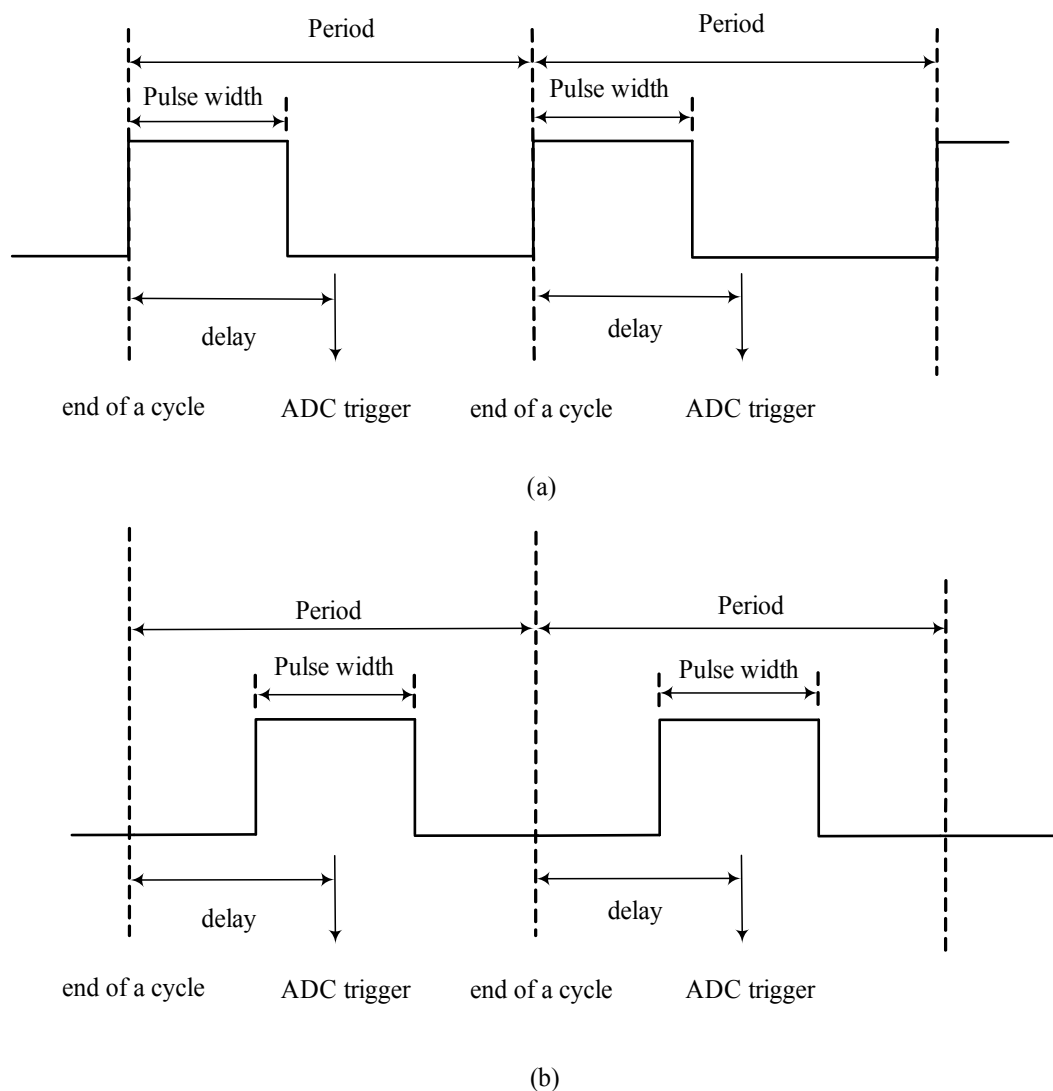
The ADC trigger is available only in GPT0 and GPT1.

The ADC trigger is a hardware handshake signal that periodically signals the Analog-Digital Converter (ADC) to begin a data conversion. The ADC trigger source can be selected from the six GPT channels using TRIG_CHSEL bits in the TCR register. The selected GPT channel must be in a PWM mode for the ADC trigger to assert. The ADC trigger can be delayed from the end of the PWM period by programming TRIG_DLY bits in the TDR register. When TRIG_EN equals 1, the ADC trigger is enabled. Refer to [Figure 28](#).

Note the following:

- TRIG_DLY has a four-cycle resolution which allows the delay to cover the maximum period for the PWM Center-Aligned mode
- The effect of ADC delay must be shorter than the PWM period
- The PWM period must be longer than the ADC conversion time

Figure 28: ADC Trigger for (a) PWM Edge-Aligned and (b) PWM Center-Aligned



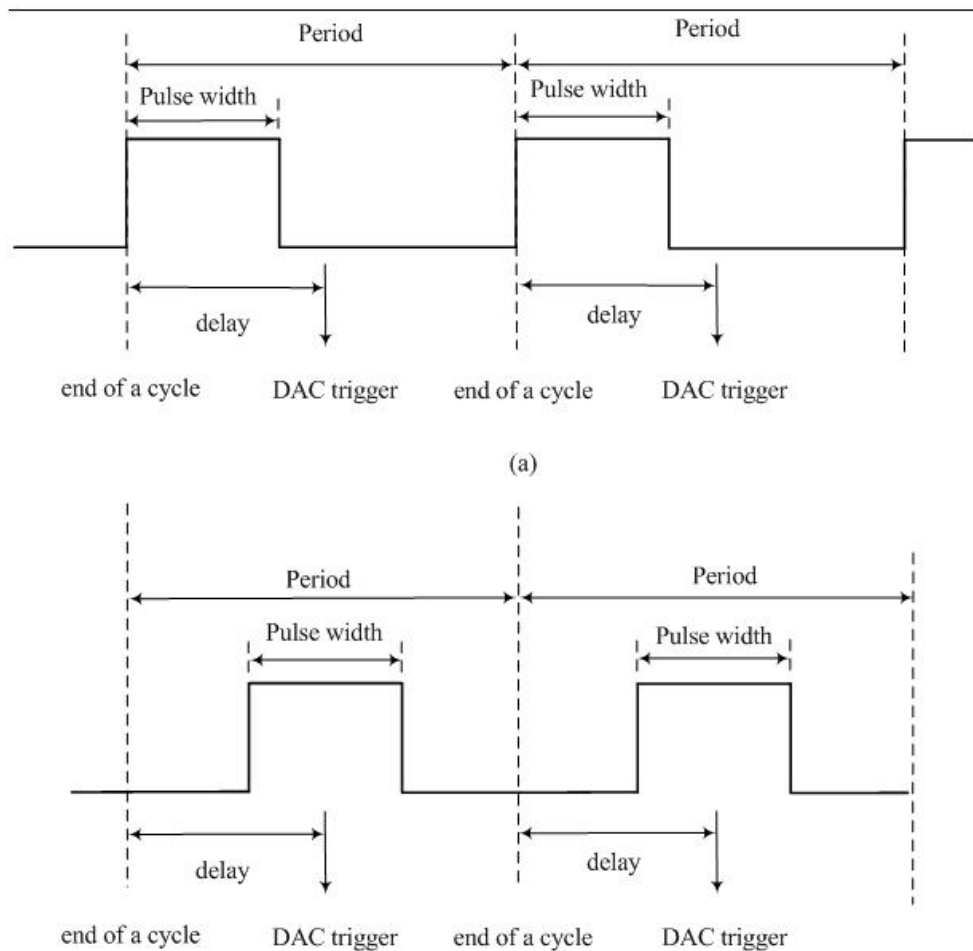
9.2.5 DAC Trigger

The DAC trigger--available only in GPT2 and GPT3--is a hardware handshake that signals the DAC to begin a conversion. The DAC trigger source can be selected from the six GPT channels using TRIG_CHSEL bit in the TCR register. The selected GPT channel must be in a PWM mode for the DAC trigger to assert. The DAC trigger can be delayed from the end of PWM period by programming TRIG_DLY bits in the TDR register. The DAC trigger is enabled when TRIG_EN = 1.

Note the following:

- TRIG_DLY has a four-cycle resolution which allows the delay to cover the maximum period for the PWM Center-Aligned mode.
- The TRIG_DLY delay time must be shorter than the PWM period.
- The PWM period must be longer than the DAC conversion time

Figure 29: DAC Trigger for (a) PWM Edge-Aligned and (b) PWM Center-Aligned



9.3 Programming Notes

9.3.1 Initialization

1. Before using the GPT, poll STS_RESETN bit to be set.
2. Program various parameters
 - a) Select clock source with CLK_SRC. If CLK_SRC is set to 0, corresponding registers in PMU module should be configured for further clock source selection.

- b) Select clock prescaler and divider values in CLK_PRE & CLK_DIV.
 - c) Set counter upper value in UPP_VAL.
 - d) If the counter value needs to be read out, program CNT_UPDT_MOD. Otherwise, leave it at 0x0.
 - e) Set CH_IO to select channel mode and configure related parameters for the channels to be used.
3. Write 1 to CNT_RESET to reset the counter. Poll CNT_RST_DONE for a 1 to determine when the counter finishes resetting. Do not access any other registers until CNT_RST_DONE is 1.
 4. Write 1 to CNT_START to start the counter. Poll CNT_RUN for a 1 to determine when the counter begins to count.
 5. If GPT channels are used, write 1 to CHx_RST to enable the corresponding channel.

9.3.2 UPP_VAL

The value written to UPP_VAL is not valid immediately. It is not effective until the counter overflows. To make the value valid immediately, write 1 to CNT_RESET.

9.3.3 User Request Register

The User Request Register (USER_REQ) performs various operations on each channel. Operations are not pipelined and must be synchronized to the proper clock domain, so operations in the USER_REQ register must not be performed in quick succession. The definition of quick succession in this case is five counter clock cycles.

For example, if a Write is to perform a channel reset, wait at least five counter clock cycles before performing the next channel reset. When the counter-clock frequency is close to the APB clock frequency, the quick succession delay can be easily waited out with just a few extra register accesses. When the counter clock frequency is much slower than the APB clock, then the safest way is to read CNT_VAL and wait five counter clock cycles to pass.

9.4 Register Description

A detailed description of the GPT registers is located in Appendix A.

10 Secure Digital Input/Output (SDIO Controller)

10.1 Overview

The 88MC200 microcontroller has one SDIO Host Controller which supports the Secure Digital I/O communication protocol. The Host Controller handles SDIO Protocol at the transmission level, adding cyclic redundancy check (CRC), start/end bit, and checking for transaction format correctness.

The SDIO module in the controller supports one SDIO card based on the standards outlined in the SDIO Card Specification Version 2.0.

The controller has the following features:

- Meets SDIO card specification version 2.0.
- Card detection (Insertion/Removal)
- Password protection of cards
- Host clock rate variable up to 50 MHz
- Supports 1-bit and 4-bit data transmitting modes
- Allows card to interrupt host in 1-bit and 4-bit modes
- Up to 100 Mbps read and write rates using four parallel data lines (4-bit mode) in the full-speed mode and up to 200 Mbps in high-speed mode
- CRC7 for command and CRC16 for data integrity
- Designed to work with I/O cards, read-only cards and read/write cards
- Supports Read wait Control, Suspend/Resume operation
- Supports FIFO overrun and underrun condition by stopping functional clock

10.2 Signal Descriptions

The controller signal pins are SDIO_CLK, SDIO_CMD, SDIO_0, SDIO_1, SDIO_2, SDIO_3, SDIO_CDn, SDIO_WP, and SDIO_LED. [Table 32](#) describes function of each signal pin.

Table 32: SDIO Card I/O Signal Summary

Signal Pin	Direction	Description
SDIO_CLK	Output	Bus clock
SDIO_CMD	Bidirectional	Bidirectional pin for command and responses
SDIO_0	Bidirectional	Bidirectional pin for read and write data. Also used for the device to signal busy to the controller during write operations.
SDIO_1	Bidirectional	Used for 4-bit data transfers and to signal SDIO interrupt conditions to the controller (optional)
SDIO_2	Bidirectional	Used for 4-bit data transfer and to signal SDIO read wait to the controller (optional)
SDIO_3	Bidirectional	Used for 4-bit data transfer (optional)
SDIO_CDn	Input	Card detection input (active low)

Table 32: SDIO Card I/O Signal Summary (Continued)

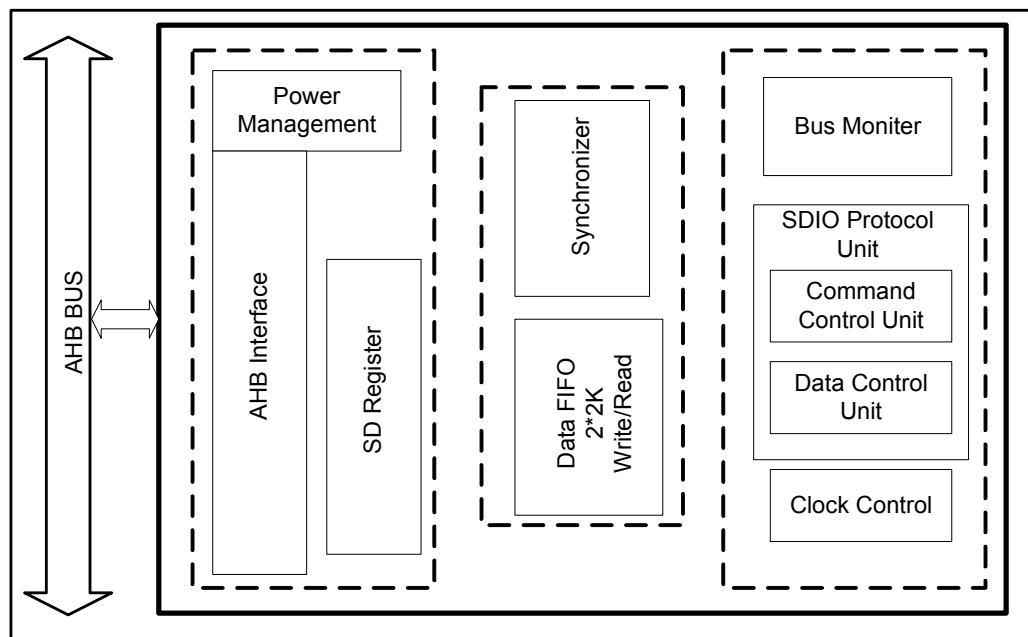
Signal Pin	Direction	Description
SDIO_WP	Input	Write protect input from card
SDIO_LED	Output	LED-on output to warn user that the card is being accessed

10.3 Controller Operation

- Provides all card-specific functions
- Serves as the bus master for the SDIO system
- Implements the standard interface to the card device
- Provides card initialization
- Provides CRC generation and validation
- Handles command, response, and data transactions
- Provides DMA operation for data transfers from system memory to SDIO FIFOs and from SDIO FIFOs to system memory

Figure 30 is the block diagram of the SDIO Controller with its various internal blocks.

Figure 30: SDIO Controller Block Diagram



The controller interfaces to the 88MC200 system through the AHB bus. The controller contains an AHB interface block, which provides system access to the internal registers and data FIFOs of the controller. The AHB interface also implements the DMA function of the controller. Two 2Kbyte data FIFOs are located within the controller to provide data buffering for both transmit and receive data with respect to the SDIO card.

Synchronization is provided within the controller to interface logic signals between the AHB and SDIO_CLK clock domains. The SDIO controller has an internal clock control unit which is responsible for generating SDIO_CLK based on frequency divider settings within the controller. The internal bus monitor is responsible for monitoring the bus for timeout conditions and protocol violations. The controller also contains all of the logic necessary to generate the SDIO protocol.

The controller can operate in DMA mode or non-DMA (PIO) mode. The controller consists of command and control registers and data FIFOs. The software has accessed these registers and FIFOs, and generates commands, interprets responses, and controls subsequent actions. Either the software or the internal DMA can be used to transfer data from system memory to the data FIFOs or from the data FIFOs to system memory.

Figure 31: Interaction of Typical SDIO System

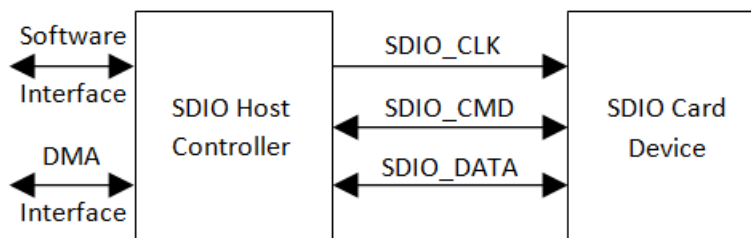


Figure 31 shows the interaction between a typical SDIO system consisting of the SDIO card device and the SDIO Host Controller. The SDIO bus connects the card/storage device to the controller. Software or the controller can turn SDIO_CLK on or off. The card and the controller communicate through the command and data lines and implement a message-based protocol. The messages consist of the following tokens:

- **Command:** A command is a six-byte token that starts an operation. The command set includes card initialization, card register reads and writes, and data transfers. The controller sends the command serially on the SDIO_CMD pin.
- **Response:** A response is a token that is an answer to a command token. Each command has either a specific response type or no response type. The format for a response varies according to the command sent and the card mode.
- **Data:** Data may be transmitted in serial, 4-bit wide depending on the negotiated bus width for data tokens between the host and card or storage device. The format for the data depends on the card mode.

For the 88MC200 microcontroller, all operations contain a command and most commands have an associated response. Read and write commands also have an associated data transfer. Command and response are sent and received on the bidirectional SDIO_CMD pin and data is sent and received on the bidirectional SDIO_x¹ pin(s). Refer to the SDIO Card Specification Version 1.0 for timing diagrams of commands and responses, with and without data transfer.

The SDIO controller can interface to cards with SDIO protocol. All protocols are serial command interfaces and either serial or parallel data interfaces to the cards or storage devices. The SDIO protocol supports block and multiple-block data transfers.

1. SDIO_x represents the four SDIO data pins, x ranges from 0 to 3

10.3.1 Operation

For SDIO operation, the SDIO_CMD and SDIO_x pins are bidirectional and require external pull-up resistors. The command and response are sent on the SDIO_CMD pin. Multiple byte data is sent on the SDIO_x pins.

In SDIO protocol, card addressing is implemented by point-to-point SDIO_CMD and SDIO_x pins. Although point-to-point communication is used, a card address is provided in the commands. Refer to the SDIO Card Specification Version 1.0 for further details.

The command is protected with a suffixed seven-bit CRC. The response has five types of coding schemes, including the no-response. The response length is 48 or 136 bits, and it may be protected with a suffixed 7-bit CRC, depending on the response type.

A read or write-data transfer is protected with a suffixed 16-bit CRC. For write-data transfers, after the data and the 16-bit CRC have been transmitted, the card sends a 5-bit CRC status token, which indicates whether the data transmission was erroneous. After the CRC status token, the card indicates that it is busy programming the data by pulling the SDIO_0 data line low.

10.3.1.1 Data Transfers

The SDIO mode supports these data-transfer modes:

- Single block read/write
- Multiple block read/write
 - Open-ended multiple block read/write
 - Multiple block-read/write with pre-defined block count
- IO_RW_DIRECT command (CMD52)
- IO_RW_EXTENDED command (CMD53)

All data transfers can be stopped at any time by the application with an SDIO abort command (with CMD52 and ASx bits set). (Refer to the SDIO Card Specification Version 1.0 for a description of the SDIO abort command and the ASx bits within the CCCR registers.)

Single Block Data Transfers

In single block-data transfers, a single block of data is transmitted. The starting address is specified in the read/write command.

The application must provide the block size to the controller. The block size is the number of bytes to be transferred. The data block is protected with a 16-bit CRC that is generated by the transmit unit, and is checked by the receiving unit. The CRC is appended after transfer of the last data bit.

Multiple Block Data Transfers

Multiple block-data transfers are similar to the single block-data transfers, except multiple blocks of data are transferred sequentially. Each block has the same length. Each block is stored to or retrieved from contiguous-memory addresses, starting at the address specified in the command.

Two types of multiple block-data transfers are defined:

- **Open-ended multiple block-read/write:** The number of blocks to be transferred is not defined in the card. The card continuously transfers data blocks until it receives an Abort command (CMD52 with ASx bits set). Each data block is protected with a 16-bit CRC.
- **Multiple block-read/write with pre-defined block count:** The number of blocks to be transferred is defined in the card. The card transfers only the number of data blocks specified. An Abort command (CMD52 with ASx bits set) is not required at the end of the data transfer (in this case) unless the data transmission terminated with an error.

The card stops data transmission if the card detects an error during a multiple block-read operation of either type. The application must then stop the operation via the Abort command (CMD52 with ASx bits set).

If the card detects an error during a multiple block-write operation of either type, the card ignores any additional incoming data. The application must then stop the operation via the Abort command (CMD52 with ASx bits set).

The application can stop a data transmission at any time. An SDIO Abort command (with CMD52 with ASx bits set) terminates multiple-block data transfers, regardless of the type. No CMD52 is necessary to stop transmission at the end of a pre-defined multiple block-data transfer.

10.4 Commands and Operations

10.4.1 Overview

SDIO cards are based on and are compatible with SD cards. The SDIO card provides high-speed data I/O with low power consumption for mobile electronic devices.

Some features of SDIO include:

- Plug-and-play (PnP) support
- Multi-function support, including multiple I/O and combined I/O and memory
- Up to seven I/O functions plus one memory supported on one card
- Allows cards to interrupt application
- Read_Wait operation
- Suspend/Resume operation

10.4.1.1 Read/Write Commands

SDIO includes two main data transfer commands, IO_RW_DIRECT (CMD52) and IO_RW_EXTENDED (CMD53).

IO_RW_DIRECT Command (CMD52)

The IO_RW_DIRECT command (CMD52) allows the simplest access to a single register within the 128K register space in any I/O function.

For SDIO, CMD52 may be used to abort a data transfer by writing to the SDIO card CCCR Abort register.

Consult the SDIO Card Specification for a description of the IO_RW_DIRECT command.

IO_RW_EXTENDED Command (CMD53)

The IO_RW_EXTENDED command (CMD53) allows the read/write of multiple I/O registers with a single command. CMD53 supports multi-byte transfer modes and block mode. Multi-byte mode performs a read or write of multiple bytes of data to/from a single I/O register. Block mode performs a read or write of multiple bytes of data to/from an I/O register address that is increased by 1 after each operation/block.

Multi-byte mode or block mode is specified in the command argument. In the block mode, the number of blocks to be transferred is specified in the command argument. Therefore, the application does not need to stop the data transmission because the number of blocks of data transferred is known by the card and the controller. In multi-byte mode, if the byte/block count field in the CMD53 argument is 0, 512 bytes will be read or written. Also, the SDIO.BLKLEN register should be written with the value of 512.

If the byte/block count field in the CMD53 argument is 0 in block mode, the data transfer is identical to the memory mode open-ended multiple block-data transfer. In this case, the data transmission must be completed by writing the I/O abort function bits.

Consult the SDIO Card Specification for a description of the IO_RW_EXTENDED command.

SDIO Data Transfer Aborts

The application may issue an I/O abort command at any time during an I/O extended read or write data transfer by sending a CMD52 to the SDIO card CCCR register. The abort command stops the data transmission. On data writes, the abort occurs between data blocks. Also, after an I/O card receives an abort on a data write, the card may respond as busy after sending the CMD52 response.

Consult the SDIO Card Specification for a description of how to use CMD52 command to abort a data transmission.

SDIO Interrupts

An SDIO card is allowed to generate an interrupt request to the CPU by asserting the SDIO_1 data pin low. The card continues to keep the SDIO_1 pin low until the interrupt request is either recognized and acted on by the CPU or the interrupt request is de-asserted due to the end of the SDIO interrupt period.

Consult the SDIO Card Specification for a description of SDIO interrupts.

SDIO Suspend/Resume

For SDIO, the application may temporarily halt (suspend) a data transfer to one function or to memory to free the SDIO bus for a higher priority data transfer to a different function or memory. Once the higher priority data transfer has completed, the application may resume the suspended data transfer from the point where it was halted.



Note

The application can suspend multiple transactions and resume them in any preferred order. The suspend/resume operation works for SDIO 1-bit and 4-bit modes.

Consult the SDIO Card Specification for a description of SDIO suspend/resume operation.

SDIO Read Wait

SDIO uses a read-wait mechanism to enable the host to send a CMD52. With read-wait, the host uses the SDIO_2 pin to signal the card to temporarily halt its sending of read data. This signal allows a CMD52 to be sent while the data is halted.

The read-wait operation is only supported for multiple-block read-data transfers in SDIO 1-bit and 4-bit modes.

Consult the SDIO Card Specification for a description of SDIO read-wait operation.

10.4.2

Controller Functional Description

Software must read and write the controller registers and FIFOs to initiate communication to a card or mass storage device. The FIFOs may also be read and written by the internal DMA controller, when enabled.

The controller provides the interface between software and the bus. It is responsible for the timing and protocol between software and the bus. The controller consists of Control and Status registers and two 8-bit FIFOs that are each 2048 entries deep – for both read and write operations.

The registers and FIFOs are accessible by software.

The controller also supports minimal data latency by buffering data and generating and checking CRCs.

Controller Reset

The controller can be reset by asserting the MSWRST bit in the CNTL2 register or by a hard or soft reset of the 88MC200 processor. All registers and FIFO controls are set to their default values after any reset.

SDIO Card Initialization Sequence

The sequence for SDIO card initialization is detailed in this section. (The following assumes that interrupts have been enabled in the SDIO.I_STAT_EN and SDIO.I_SIG_EN registers.) Refer to the SDIO Card Specification Version 1.0 for details of the commands and responses referenced below.

1. Wait for CDINS interrupt in the SDIO.I_STAT register.
2. Write 1 to the CDINS bit in the SDIO.I_STAT register to clear this bit.
3. Write 1 to the CLKEN bit in the SDIO.CNTL2 register to enable the external SDIO_CLK.
4. Write to the SDIO.CNTL1 register to set the VLTGSEL field according to the supported voltage in the SDIO.CAP0 register. Also set the BUSPWR bit in this register to enable the bus power.
5. Write all 0s to the SDIO.BLK_CNTL register to disable data transfer. Also, write all 0s to the SDIO.ARG register. Write 32'h05020000 to the SDIO.CMD_XFRMD register. This value sends the CMD5 (IO_SEND_OP_COND) command to the card and expects a 48-bit response. (Refer to the SDIO Card Specification Version 1.0 for a description of commands and arguments for each command.)
6. Wait for CMDCOMP interrupt in the SDIO.I_STAT register.
7. Write 1 to the CMDCOMP bit in the
8. SDIO.I_STAT register to clear the bit.
9. Read the RESP0 register to determine the card response. The response contains the contents of the card OCR register, and the card voltage profile should match the controller voltage range specified in the SDIO.CAP0 register. The response should also contain the number of IO functions that the card supports. The number of IO functions should be greater than 0. (Refer to the SDIO Card Specification Version 1.0 for a description of the OCR register.)
10. If the card voltage profile does not match, then the card cannot be supported. If the card voltage profile matches, repeat Step 5, but set the SDIO.ARG register to specify the appropriate operating voltage for the card.
11. Wait for CMDCOMP interrupt in the SDIO.I_STAT register.
12. Write 1 to the CMDCOMP bit in the SDIO.I_STAT register to clear the bit.
13. Read the SDIO.RESP0 register to determine the card response. The response contains the contents of the card OCR register, and the card power-up ready/busy bit (bit 31 of the response) should be set to 1 when the card is ready. If this bit is 0, return to Step 9. (Refer to the SDIO Card Specification Version 1.0 for a description of the OCR register.)
14. Write all 0s to the SDIO.BLK_CNTL register to disable data transfer. Write all 0s to the SDIO.ARG register. Write 32'h031A0000 to the SDIO.CMD_XFRMD register. This value sends the CMD3 (SEND_RELATIVE_ADDR) command to the card and expects a 48 bit response. Also, this value enables the command index and CRC checking. (Refer to the SDIO Card Specification Version 1.0 for a description of commands and arguments for each command.)
15. Wait for CMDCOMP interrupt in the SDIO.I_STAT register.
16. Write 1 to the CMDCOMP bit in the SDIO.I_STAT register to clear the bit.

17. Read the SDIO.RESP0 register to determine the card's response. The response contains the new, published relative card address (RCA) of the card and the card's status. The card's newly published RCA is located in bits 31:16 of the SDIO.RESP0 register, while the card's status is located in bits 15:0 of the SDIO.RESP0 register. (Refer to the SDIO Card Specification Version 1.0 for a description of commands and responses for each command.)
18. Further commands to the SDIO card can now be performed by issuing a CMD7 (SELECT/DESELECT_CARD) command.

Response and Data Error Detection

The controller detects response and data errors on the SDIO bus, and reports them in the SDIO.I_STAT status register, if the appropriate bit is set in the SDIO.I_STAT_EN register. Responses are also recorded in the SDIO.RESPx registers. If a response or data error occurs and the appropriate enable bit is set in the SDIO.I_SIG_EN register, an interrupt request is generated to the Interrupt Controller. Software may either respond to an interrupt request or poll the SDIO.I_STAT_EN register. See [Table 33](#).

Table 33: Response and Data Errors

Error	Description
CDINT	Interrupt request from card occurred
DENDERR	End bit error detected in read data or CRC status
DCRCERR	A CRC error was detected from the read data or by the card from the write data
DTOERR	Read data timeout or busy timeout
CIDXERR	Command index error occurred in the command response
CENDERR	End bit error detected in the command response
CCRCERR	A CRC error was detected from command response
CTOERR	A response timeout occurred

10.5 Interrupts

The controller generates interrupts to signal the status of a command sequence. The software is responsible for:

- Enabling the interrupts appropriately by programming the SDIO.I_STAT_EN register
- Determining the source of any received interrupt

See [Table 34](#).

Table 34: Controller-Generated Interrupts

Interrupt	Description
Data End Bit Error	Asserted when a 0 is detected at the end bit position of read data which uses the DAT line or the end bit position of the CRC status.
Data CRC Error	A data CRC error condition has occurred.
Data Timeout Error	A data timeout error condition has occurred.
Command Index Error	Asserted when a command index error occurs in the command response
Command End Bit Error	Asserted when the end bit of a command response is 0.
Command CRC Error	A command CRC error condition has occurred.
Command Timeout Error	A command timeout error condition has occurred.
Card Interrupt	An SDIO card interrupt condition has occurred.

Table 34: Controller-Generated Interrupts (Continued)

Interrupt	Description
Card Removal	Asserted when a card removal event is detected.
Card Insertion	Asserted when a card insertion event is detected.
Buffer Read Ready	Asserted in non-DMA mode when valid read data exists in the buffer.
Buffer Write Ready	Asserted in non-DMA mode when space is available in the buffer for write data.
DMA Interrupt	Asserted when the controller detects the system memory buffer boundary specified in the Block Size register.
Block Gap Event	If the BGREQSTP bit in the CNTL1 register is set, this interrupt is asserted at a Block Gap.
Transfer Complete	Asserted when a read/write transaction is completed.
Command Complete	Asserted when the end bit of the command response is received.

10.6 Clock Control

Both the controller and software can control the bus clock (SDIO_CLK) by turning it on and off. This capability helps control the data flow to prevent underruns and overflows, and also conserves power. Software can also change the SDIO_CLK frequency to achieve the maximum data-transfer rate specified for a card-identification frequency.

The controller can turn off SDIO_CLK automatically to prevent data overflows and underruns if any of the following events occur:

- Both SDIO FIFOs become full during data reads
- One SDIO FIFO is being read by the software and the other SDIO FIFO becomes full
- Both SDIO FIFOs become empty during data writes
- One SDIO FIFO is being written by the software and the other SDIO FIFO becomes empty.
- For read data transfers, the controller turns on SDIO_CLK after the SDIO FIFO has been emptied. For write data transfers, the controller turns on SDIO_CLK after the SDIO FIFO is no longer empty.

10.7 Data FIFOs

The controller FIFOs for received data and transmitted data are physically the same FIFOs. There are two FIFOs which are automatically configured to receive data from or transmit data to the bus, depending on the preferred command. These FIFOs are accessible by software or by internal DMA operation and are described in the following paragraphs. The data FIFOs are switched from the Receive mode to the Transmit mode of operation based on the state of the DXFRDIR bit in the CMD_XFRMD register. Software must ensure that all data transfers from a transaction are complete (based on the XFRCOMP bit in the I_STAT register) before modifying the DXFRDIR bit to change the transfer direction. The command responses from the card are captured in the RESPx registers.

10.7.1 Command Response Register

The SDIO.RESPx registers contain the response received from a card after a command is sent from the controller. In total the SDIO.RESPx register is a read only, 128-bit register, which are read as four 32-bit words starting at address offset 0x010.

The SDIO.RESPx registers hold all possible response lengths. The locations of the command response fields within the SDIO.RESPx registers are specified in Appendix Section A7.

The SDIO.RESPx registers do not contain the response CRC. The status of the CRC check is recorded in the CCRCERR bit of the SDIO.I_STAT register.

10.7.2 Receive Data FIFO Configuration

In the receive data configuration, the FIFOs are read-only through software or the internal DMA and are readable on 1-, 2-, or 4- byte boundaries. Each receiving data FIFO is 2048 entries deep and 1 byte wide. Access to the receive data FIFO is handled by the controller and depends on the status of the controller.

The receive data FIFOs are used in a double-buffering fashion to capture data from the bus. At any time during receive mode, the software or internal DMA has read access to one of the receive data FIFOs, the bus has write access to the other receive data FIFO. The controller handles the swapping of the read and write controls to each of the FIFOs to implement the double-buffering scheme. This swapping process continues throughout the data transfer and is transparent to both software and the SDIO card.

If at any time both FIFOs become full and the data transmission is not complete, the controller turns SDIO_CLK off (or issues a read-wait if the card supports the read-wait mechanism) to prevent any overflows. When the SDIO_CLK is off, data transmission from the card stops until SDIO_CLK is turned back on. After software or the internal DMA has emptied the FIFO that it is connected to, the controller turns SDIO_CLK on to continue data transmission.

10.7.3 Transmit Data FIFO Configuration

In the transmit data configuration, the FIFOs are write only through software or the internal DMA and are writable on 1-, 2-, or 4- byte boundaries. Each transmit data FIFO is 2048 entries deep and 1 byte wide. Access to the transmit-data FIFO is handled by the controller and depends on the status of the controller.

The transmit-data FIFOs are used in a double-buffering fashion to transmit data to the bus. At any time during Transmit mode, the software or internal DMA has write access to one of the transmit-data FIFOs, the bus has read access to the other transmit-data FIFO. The controller handles the swapping of the read and write controls to each of the FIFOs to implement the double-buffering scheme. This swapping process continues throughout the data transfer and is transparent to both software and the SDIO card.

If at any time both FIFOs become empty and the data transmission is not complete, the controller turns off SDIO_CLK to prevent any underruns. When the SDIO_CLK is off, data transmission to the card stops until SDIO_CLK is turned back on. After data has been transmitted to the FIFO and it is no longer empty, the controller turns on SDIO_CLK to continue data transmission.

10.7.4 DMA and Programmed I/O

Data transfers with the controller FIFOs are performed via the internal DMA or programmed I/O.

To access the controller FIFOs by internal DMA, software must set the DMA_EN bit in the SDIO.CMD_XFRMD register. Software must also program the SDIO.SYSADDR register with the appropriate system source or target address for data transfers. (The DXFRDIR bit in the SDIO.CMD_XFRMD register determines if the SDIO.SYSADDR register contains a source or target address.) Also, software must program the DMA_BUFSZ field of the SDIO.BLK_CNTL register to the appropriate system memory buffer size for the DMA transfers. DMA operations begin when software writes the upper byte of the SDIO.CMD_XFMD register. The system memory buffer size can be configured for a range of 4KB to 512 KB buffers, and the controller generates a DMA interrupt in the SDIO.I_STAT register when the system memory buffer boundary is detected. At this point, the DMA transfers have stopped, and the software must reprogram the SDIO.SYSADDR register to resume the DMA operation. During DMA transfer operation, the controller performs the necessary read or write operations on the AHB bus, and internally generates the write or read accesses to the internal FIFOs.

With programmed I/O, software waits for SDIO.I_STAT register BUFWRDY or BUFDRDY bits to generate interrupt requests, if enabled, before reading or writing the DP register to access the internal FIFOs. Alternatively, the application may poll the SDIO.I_STAT register for the FIFO request

by disabling the interrupt request generation in the controller SDIO.I_SIG_EN register and enabling the interrupt status in the SDIO.I_STAT_EN register.

10.8 Low-Power Mode Operation

Software must configure the controller appropriately for entering or exiting the System-Idle or low-power modes. For both Idle and low-power modes, all data transfers must be completed either through issuing an Abort command (CMD52) or through the software waiting for the command-complete interrupt to signal the end of a card transaction. To enter an Idle or low-power mode, the SDIO card must be placed into the Idle state (if the card is in Card ID Operation mode) or the Stand-by state (if the card is in Data Transfer Operation mode). The controller DMA must be disabled by clearing the DMA_EN bit in the SDIO.CMD_XFRMD register. For entering Idle mode, the software can then shut down the clock to SDIO by writing to the appropriate CCU register. While in Idle mode, a card insertion or removal event can be programmed to wake up the system to Active mode without an active SDIO_CLK running.

In low-power modes, the controller retains state, and any data remaining in the FIFOs (either Transmit or Receive) is retained for access after returning to Run mode. If the non-data retention low-power mode is entered, the controller and data FIFOs lose all state information, and the system software must reprogram the controller upon returning to Run mode.

The system requires the card interface be placed in an Idle mode while the system remains in Run-Power mode. The same procedure as just described can be used to shut down the SDIO_CLK for the interface to conserve power. In this mode, a card insertion or removal event can generate a system interrupt if the software has programmed it to do so by writing the appropriate bits in the SDIO.I_STAT_EN and SDIO.I_SIG_EN registers. The system software can also place the SDIO card into the Wait-IRQ state when the card is in Interrupt mode. In this mode, the SDIO_CLK must remain active, and the card generates a response on the SDIO interface to produce the interrupt to the host. This mode allows the system to eliminate the polling of status from the SDIO card to conserve power.

10.9 Card Communication Protocol

Block transfers are basically classified into three types according to how the number of blocks is specified:

- **Single Block Transfer:** The number of blocks is specified to the controller before the transfer. The number of blocks specified is always one.
- **Multiple Block Transfer:** The number of blocks is specified to the controller before the transfer. The number of blocks specified is one or more.
- **Infinite Block Transfer:** The number of blocks is not specified to the controller before the transfer. This transfer is continued until an abort transaction command is executed. This abort transaction is performed by CMD52.

10.9.1 PIO Operation

The sequence for PIO operation follows, which assumes that interrupts have been enabled in the SDIO.I_STAT_EN and SDIO.I_SIG_EN registers.

1. Set the value corresponding to the data byte length of one block and the value corresponding to the data block count in the SDIO.BLK_CNTL register.
2. Set the value corresponding to the issued command in the SDIO.ARG register.
3. Set the preferred value of the control bits as well as the value of the issued command to the SDIO.CMD_XFRMD register. **Note:** When writing the upper byte of the Command register, the command is issued. (Refer to Table for a description of the SDIO.CMD_XFRMD register.)
4. Wait for the CMDCOMP interrupt in the SDIO.I_STAT register.
5. Write 1 to the CMDCOMP bit in the SDIO.I_STAT register to clear this bit.

6. Read SDIO.RESPx registers and get necessary information in accordance with the issued command.
7. If this sequence is for a write to a card, go to Step 8. If this sequence is for a read from a card, go to Step 13.
8. Wait for BUFWRDY interrupt in the SDIO.I_STAT register.
9. **Write Transfer using PIO.** On receiving the BUFWRDY interrupt, the system processor acts as a master and starts transferring the data via the buffer data port register (DP). Transmitter begins sending the data on the bus when a block of data is ready in the first FIFO. While the transmitting the data on the bus, the BUFWRDY interrupt is sent to the ARM processor for the second block of data. The ARM processor acts as a master and starts sending the second block of data via SDIO.DP register to the second FIFO. The BUFWRDY interrupt is asserted only when a FIFO is empty to receive a block of data.
10. Write 1 to the BUFWRDY bit in the SDIO.I_STAT register to clear this bit.
11. Write block data (according to the number of bytes specified in Step 1) to the SDIO.DP register.
12. Repeat until all blocks are sent and then go to Step 18.
13. **Read Transfer using PIO.** The BUFDRDY interrupt is asserted whenever a block of data is ready in one of the FIFOs. On receiving the BUFDRDY interrupt, the ARM processor acts as a master and starts reading the data via SDIO.DP. The receiver in the controller starts reading the data from bus only when a FIFO is empty to receive a block of data. When both the FIFOs are full the controller stops the data coming from the card through read/wait mechanism (if card supports read/wait) or through stopping the clock.
14. Wait for BUFDRDY interrupt in the SDIO.I_STAT register.
15. Write 1 to the BUFDRDY in the SDIO.I_STAT register to clear this bit.
16. Read block data (in according to the number of bytes specified in step 1) from the SDIO.DP register.
17. Repeat until all blocks are received and then go to Step 18.
18. If this sequence is for Single or Multiple Block Transfer, go to Step 19. For an infinite block transfer, go to Step 21.
19. Wait for XFRCOMP interrupt in the SDIO.I_STAT register.
20. Write 1 to the XFRCOMP bit in the SDIO.I_STAT register to clear this bit.
21. Perform the sequence for an abort transaction.

10.9.2 DMA Operation

The sequence for data transfer with DMA operation is as follows, which assumes that interrupts have been enabled in the SDIO.I_STAT_EN and SDIO.I_SIG_EN registers.

1. Set the system address for DMA in the SDIO.SYSADDR register.
2. Set the value corresponding to the data byte length of one block and the value corresponding to the data block count in the SDIO.BLK_CNTL register.
3. Set the value corresponding to the issued command in the SDIO.ARG register.
4. Set the preferred value of the control bits as well as the value of the issued command to the SDIO.CMD_XFRMD register. Note: When writing the upper byte of Command register, the command is issued.
5. Wait for the CMDCOMP interrupt in the SDIO.I_STAT register.
6. Write 1 to the CMDCOMP bit in the SDIO.I_STAT register to clear this bit.
7. Read SDIO.RESPx registers and get necessary information in accordance with the issued command.
8. **Write Transfer using the DMA.** On receiving the response end bit from the card for the Write command (data flowing from host to card) the controller serves as the master and requests the AHB bus. After receiving the grant, the controller begins reading a block of data from the system memory and it fills the first FIFO. Whenever a block of data is ready, the transmitter starts

sending the data on the bus. While transmitting the data on the bus, the controller requests the AHB bus to fill the second block in the second FIFO. Similarly, the controller reads a block of data from the system memory whenever a FIFO is empty. This process continues until all the blocks are read from the system memory. The XFRCOMP interrupt is set only after transferring all the blocks of data to the card.

9. Read Transfer using the DMA. The block of data received from the card (data flowing from card to host) is stored in the first FIFO. Whenever a block of data is ready, the controller acts as the master and requests the AHB bus. After receiving the grant, the controller starts writing a block of data into the system memory from the first FIFO. While transmitting the data into system memory, the controller receives the second block of data and stores it in the second FIFO. Similarly, the controller writes a block of data into the system memory whenever data is ready. This process continues until all the blocks are transferred to the system memory. The XFRCOMP interrupt occurs only after transferring all the blocks of data to the system memory.



Note

The controller receives a block of data from the card only when it has room to store a block of data in a FIFO. When both of the FIFOs are full, the controller stops the data coming from the card through a read/wait mechanism (if the card supports read/wait) or through stopping the clock.

10. Wait for the XFRCOMP and DMAINT interrupts in the SDIO.I_STAT register.
11. If XFRCOMP is set, go to Step 14, else if DMAINT is set, go to Step 12. XFRCOMP has higher priority than DMAINT.
12. Write 1 to the DMAINT bit in the SDIO.I_STAT register to clear this bit.
13. Set the address of the next continuous system memory buffer in the SDIO.SYSADDR register and go to Step 10.
14. Write 1 to the XFRCOMP and DMAINT bits in the SDIO.I_STAT register to clear these bits.

10.9.3 Abort Transaction

An abort transaction is performed using CMD52. There are two cases where the software must perform an abort transaction: (1) When the software stops infinite block transfers, and (2) when software stops transfers while a multiple block transfer is executing.

There are two ways to issue an abort command. The first is an asynchronous abort. The second is a synchronous abort. In an asynchronous abort sequence, the software can issue an abort command at anytime unless the CCMDINHBT bit in the SDIO.STATE register is set to 1. In a synchronous abort, the software issues an abort command after the data transfer is stopped by using the BGREQSTP bit in the SDIO.CNTL1 register.

10.9.3.1 Synchronous Abort

The sequence for a synchronous abort is as follows:

1. Set the BGREQSTP bit in the SDIO.CNTL1 register.
2. Wait for the XFRCOMP interrupt in the SDIO.I_STAT register.
3. Write 1 to the XFRCOMP bit in the
4. SDIO.I_STAT register to clear this bit.
5. Issue the abort command.
6. Set both the DATSWRST and CMDSWRST bits in the SDIO.CNTL2 register.
7. Poll both the DATSWRST and CMDSWRST bits in the SDIO.CNTL2 register until both bits are cleared. When both the DATSWRST and CMDSWRST bits in the SDIO.CNTL2 register are cleared, the abort sequence is complete.

10.10 Register Descriptions

Refer to Appendix Section T for a detailed description of the SDIO registers.

11

USB OTG Interface Controller

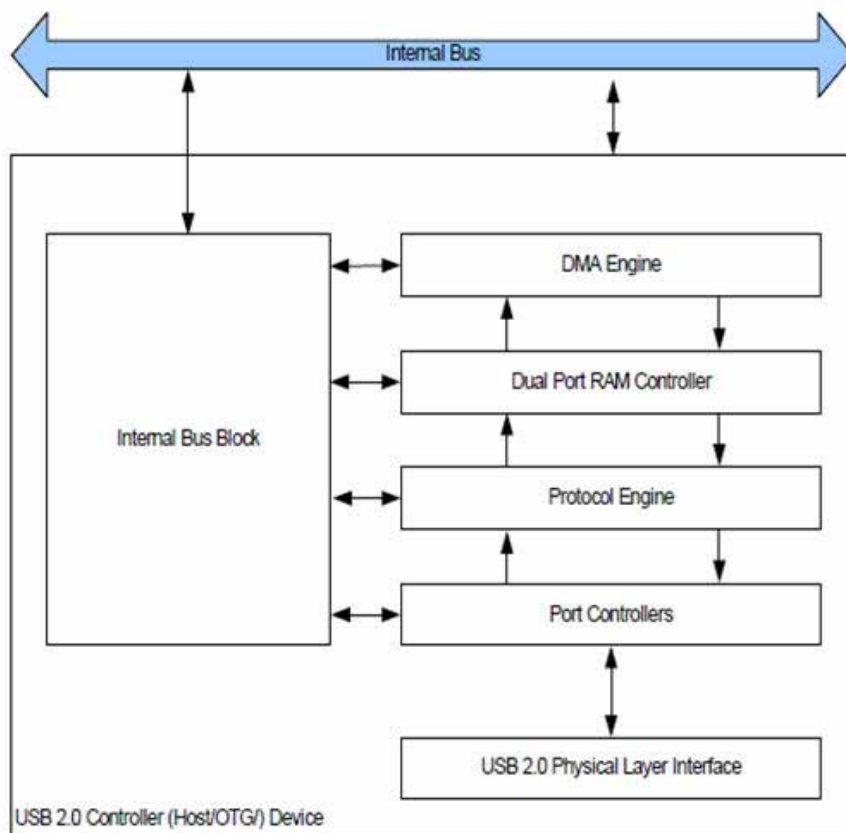
The 88MC200 USB Interface includes one USB OTG-capable dual-role host/device controller that is compliant with the USB 2.0 specification.

11.1 Features

- Full USB OTG functionality with integrated transceiver, allowing support for an Enhanced Host
- Controller Interface (EHCI) host or a device
- Supports Full-Speed/Low-Speed USB 2.0 Host/Device/OTG modes
- Up to 16 configurable bi-directional endpoints for device mode
 - Transfer types support: Control, Interrupt, Bulk or Isochronous
 - Endpoint 0 - dedicated for control endpoint
- Control signals for external power supply and detection of voltages for OTG signaling
- Capability to respond as self- or bus-powered device and control to allow charging from bus
- Full 1 KB TxFIFOs for each endpoint, which can hold the largest USB2 packet.
- 2 KB shared Rx buffer for all incoming data

Each of the major blocks shown in [Figure 32](#) and briefly described in the following sections.

Figure 32: USB Controller Block Diagram



11.2 Internal Bus Interface

The internal bus contains all the control and status registers that allow a processor to interface to the USB core. These registers allow a microprocessor to control the configuration of the core, ascertain the capabilities of the core, and control the core in operation for both host and device modes.

11.2.1 DMA Engine

The DMA Engine Block presents a bus initiator (master) interface to the internal bus. It is responsible for moving all of the data to be transferred over the USB between the USB core and buffers in the system memory.

The DMA controller must access both the control information and packet data from the system memory. The control information is contained in the link list-based queue structures. The DMA controller has state machines that can parse all of the data structures defined in this controller specification.

11.2.2 Dual Port RAM Controller

The Dual Port RAM controller, which is used for context information, builds configurable FIFOs between the Protocol Engine block and the DMA controller. These FIFOs decouple the system processor memory bus request from the extremely tight timing required by the USB itself. The use of the FIFO buffers differs between host and device mode operation. In Host mode, a single data channel is maintained in each direction through the dual-port memory. In Device mode, multiple FIFO channels are maintained for each of the active endpoints in the system.

11.2.3 Protocol Engine

The Protocol Engine parses all the USB tokens and generates the response packets. It is responsible for checking for errors, checking field generation, formatting all handshaking, ping, and data response packets on the bus, and for generating any signals based on a USB-based time frame. In Host mode, the Protocol Engine also generates all of the token packets that are required by the USB protocol. The Protocol Engine contains several sub-functions:

- The token state machines track all of the tokens on the bus and filter the traffic based on the address and endpoint information in the token. In Host mode, these state machines also generate the tokens required for data transfer and bus control.
- The CRC5 and CRC16 CRC generator/checker circuits check and generate the CRC check fields for the token and data packets.
- The data and handshake state machines generate any responses required on the USB and move the packet data through the dual-memory FIFOs to the DMA controller block.
- The Interval timers provide timing strobes that identify important bus timing events: bus timeout interval, microframe interval, start of frame interval, and bus reset, resume, and suspend intervals.
- Reports all transfer status to the DMA engine.

11.2.4 Port Controller

The Port controller block interfaces to the UTMI/UTMI+ compatible transceiver macrocell. The primary function of the Port controller block is to isolate the remainder of the USB core from the transceiver and to move all of the transceiver signaling into the primary clock domain of the USB core. This process allows the core to run synchronously with the system processor and its associated resources.

11.3 Signal Descriptions

[Table 35](#) describes the USB OTG signals and [Table 36](#) details the Host Controller signals.

Table 35: USB OTG Controller Signal Descriptions

Name	Type	Description
USBOTG_P	Bidirectional	USB D+
USBOTG_N	Bidirectional	USB D-
USBVBUS	Input	VBUS selection – Input in device mode, unused in host mode, Input/Output for OTG mode to supply +5V@10mA during session negotiation
ID_PIN	Input	0: A-device 1: B-device

Table 36: USB Host Controller Signal Descriptions

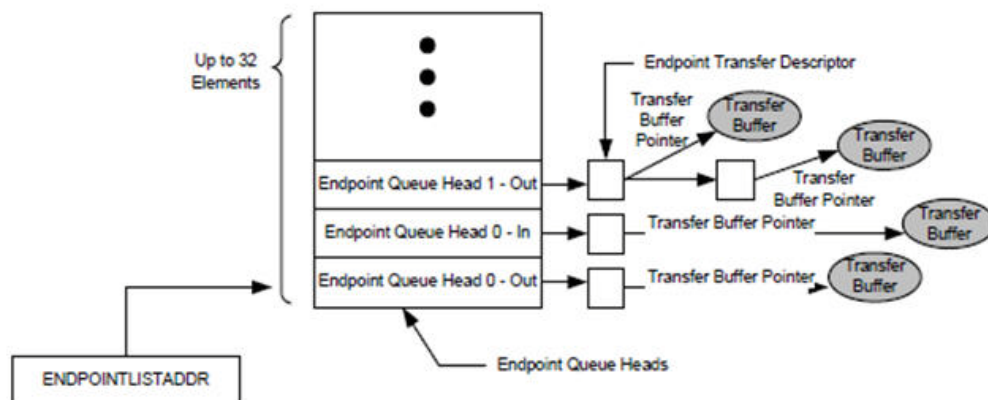
Name	Type	Description
USBH_P	Bidirectional	USB D+
USBH_N	Bidirectional	USB D-

11.4 Functional Description

The USB OTG Controller is a fully-compliant USB peripheral device that can also assume the role of a USB host. The OTG state machines determine the role of the device based on the connector signals and then initializes the device in the appropriate mode of operation (host or peripheral) based upon its method of connection. After connecting, the devices can negotiate using the OTG protocols to assume the role of host or peripheral depending on the task to be accomplished. The attached peripherals share USB bandwidth through a host-scheduled, token-based protocol.

Figure 33 displays the endpoint queue head data structure.

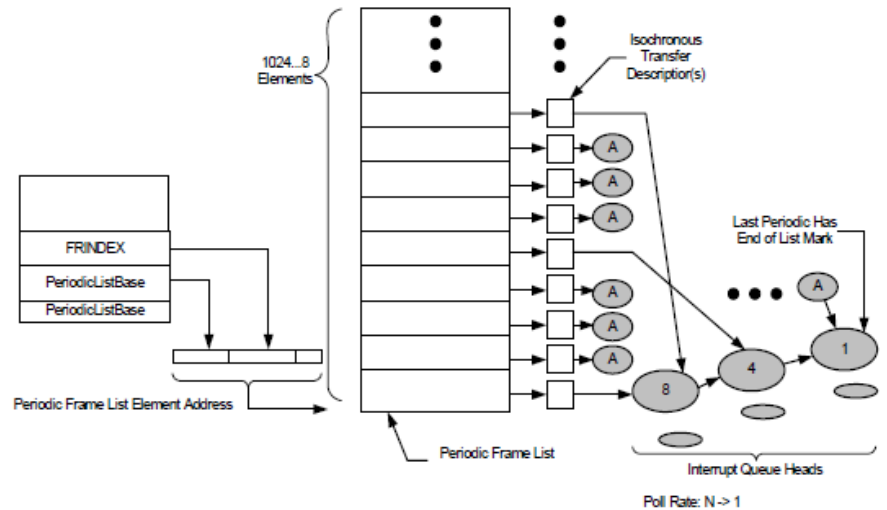
Figure 33: End Point Queue Head Organization



11.4.1 Host Data Structure

The Host data structures are used to communicate control, status, and data between software and the Host Controller. The Periodic Frame List, as shown in Figure 34 is an array of pointers for the periodic schedule. A sliding window on the Periodic Frame List is used. The Asynchronous Transfer List is where all the control and bulk transfers are managed.

Figure 34: Periodic Schedule Organization



11.5 USB Controller Operation

11.5.1 FIFO Operation in Device Mode

11.5.1.1 Streaming Mode

Device mode - ISO IN - Streaming mode

- The controller starts fetching data after the software primes the endpoint, and fills the TX FIFO to the available space.
- When the respective IN arrives from the Host, the controller sends the data, fetching more data from system memory to TX FIFO as space becomes available.

Device mode - ISO OUT - Streaming mode

- When ISO OUT arrives and data is sent from Host, the device controller begins storing it to RX buffer.
- The device controller starts sending data from RX FIFO to system memory as soon as a burst size of data is available.
- After all the ISO OUT data is received, and while the RX FIFO still has data inside waiting to be transferred to system memory, the device controller can receive other packets.
- If only 1 RX FIFO position is free, or full, the device "BTO" the OUT/DATA from Host.
- If there is more than 1 position free, the device accepts OUT/DATA from Host but if the FIFO is not read to system memory, an overflow occurs anyway, and the packet is NAKed.

Device mode - Bulk IN - Streaming mode

- When the Host sends the IN, if the device has not primed the EP yet, it "NAKs" the IN.
- After EP is primed, DMA engine in the device controller starts fetching data to the TX buffer.

- When the protocol engine part of the controller is primed (after synchronization), the device controller responds to an IN with data.
- Device controller continues to fetch data as space is available in TX buffer (at least one burst worth of data of free space available).
- If during a packet the system bus is unable to buffer all the data for that packet on time, the packet is cut short due to underrun.
- During a transfer, the device controller “NAKs” an IN if the TX buffer is empty, or more specifically, if no data was loaded for the next packet.

Device mode - Bulk OUT - Streaming mode

- When the Bulk OUT arrives and data is sent from Host, the device controller begins storing it to the RX buffer.
- The device controller starts sending data from RX FIFO to system memory as soon as a burst worth of data is available.
- After all the Bulk OUT data is received and the RX FIFO still has data inside waiting to be transferred to system memory, the device controller can receive other packets.
- If only 1 RX FIFO position is free, or full, the device “BTOs” the OUT/DATA from Host.
- If there is more than 1 position free, the device accepts OUT/DATA from the Host; if the FIFO is not read to system memory, an overflow occurs anyway, and the packet NAKed.
- This behavior is the same as for ISO OUT.

11.5.1.2 Additional Notes on TX FIFO Buffering – IN Endpoints

Initial Priming:

- When priming has started on DMA side, the controller loads the leading data into the TX buffer, and only then completes the priming operation (PE is primed). This pre-buffering is performed for the entire first packet, or until the TX FIFO is full.

Buffering after the first packet:

- After the first packet, the second packet in a dTD is sent as long as at least one byte was loaded to the FIFO for the second packet.
- Once FIFO is loaded with the first packet, the entire dTD (all the packets in one dTD) is sent for every IN from Host, and the system bus must continue back-filling the TX FIFO to keep up with data being sent for the several packets.

BTO or NAK from Host to a Data Packet:

- If a packet is NAKed or BTOed by Host, device flushes TX buffer and removes the priming state.
- It then returns to repeat the buffer operation.
- If an IN arrives from the host in the meantime, it is NAKed.
- Once the packet is fully loaded inside the TX buffer again, or the TX buffer is full, the prime state in the PE is set to active.
- Only then does the device controller respond to the Host IN token with the data packet.

Underrun of the Device TX FIFO:

- If a TX FIFO underrun occurs, device clears the prime, flushes TX buffer, and then reloads all of failing packet to the TXFIFO.
- At the same time, the device “NAKs” an IN from the Host.
- Only when entire packet is in FIFO again or FIFO is full does the Device respond to the Host IN with a data packet.

Buffering between dTDs:

- When the Host sends an ACK to the last data packet of the first dTD, the device disables the primed state.
- This last data packet can either be max packet sized, a short packet or a zero-length packet, depending on the transfer total length and the ZLT bit in the Queue head. Either one is ACKed by the Host, and the priming disabled after that.
- The device then retires the dTD back to system memory, and loads the new dTD.
- In the meantime, any IN packet from the Host is NAKed.
- After loading the new dTD, the device starts buffering the first packet into the TX buffer.
- Once the packet is fully loaded inside the TX buffer again, or the TX buffer is full, the prime state in the PE is set to active.
- Only then does the device controller responds to the Host IN token with the data packet.
- This behavior is similar to the initial priming of the first dTD.

11.5.1.3 Non-Streaming Mode

Device Mode - ISO IN - Non-Streaming mode

- Same as Streaming mode

Device Mode - ISO OUT - Non-Streaming mode

- Same as Streaming mode

Device mode - Bulk IN - Non-Streaming mode

- When the Host sends the IN, if the device has not yet primed the EP; it "NAKs" the IN.
- After EP is primed, the DMA engine in device controller starts fetching data to the TX buffer.
- The device controller continues fetching data as space is available in the TX buffer.
- The device controller responds only to the first IN with data when the entire packet is fetched to the TX FIFO or the TX FIFO is full. In the meantime, it responds to the IN with NAK.
- The following packets are sent only if next packet is fully in FIFO, or if the FIFO is full again.

Device Mode - Bulk OUT - Non-Streaming mode

- When Bulk OUT arrives and data is sent from Host, the device controller begins storing it to the RX buffer.
- The device controller starts sending data from RX FIFO to system memory as soon as a burst size of data is available.
- The device controller responds with NYET to this first packet of a transfer, so the Host sends PINGs after that.
- The device controller sends NAKs to the PINGs as long as data is still in the RX FIFO (RX FIFO not yet empty).
- Only when the RX FIFO is empty again does the device controller ACK the PING, and the Host sends the next IN.

11.5.1.4 FIFO Operation in Host Mode

Host Mode - ISO IN/OUT - Streaming mode

- Taking the example of using Streaming mode: It is a 760-byte transfer, using a max packet size of 370 bytes, MULT=3.

For OUT direction:

- Assuming a watermark value of 512 bytes (larger than the 370-byte packet), the controller uses the packet size as watermark.
- Behavior is the same as in Non-Streaming mode.
- Host controller starts fetching the data to TX FIFO after the SOF is sent.
- In Host ISO OUT, the TXFIFOTHRES watermark applies to every packet in a transfer.
- Assuming a packet size of 1024 bytes, TX FIFO with a size of 1024 bytes, and a TX watermark of 512 bytes as well: The DMA writes the Packet Start TAG to the TX FIFO, and then proceeds to fill with data. As the Start TAG is read just after it is written, 512 bytes are again available on the FIFO. The DMA then fills the TX FIFO to the 512 bytes watermark.
- Only then does the controller send the OUT token and start the packet. As the controller reads the first bytes from the TX FIFO, it makes space available for the missing End Packet TAG, which then fits without any problems.
- If the packet size is 1024 for a transfer size of 3072, MULT=3, and using a TX watermark of 512 bytes, the behavior is the same as in the previous example, except that when the first packet starts being sent, the controller backfills the TX FIFO with the remaining data for the packet, as soon as one burst of data space is available. The behavior is the same for each of the MULT=3 packets.
- Using an ISO packet size larger than the TX FIFO size is not a problem, as long as the system bus has enough bandwidth to backfill the TX buffer after the packet started being sent.
- For a packet size = 1024 bytes, TX watermark set to 512 and TX FIFO size is 1024, the controller fills the TX FIFO to 512 bytes before sending the OUT token, and then continues backfilling the FIFO.

For IN direction:

- Behavior is the same as in Non-Streaming mode in the sense that the Host Controller is working with one packet at a time.
- Controller starts sending data from RX FIFO to system memory as soon as one burst of data is available.
- Controller always waits for all the packet data to be stored in system memory, and only then proceeds to the next packet; that is, it sends only the IN token for the next packet when the RX FIFO is empty.
- If the packet size is 1024 for a transfer size of 3072, MULT=3, the behavior is the same, each packet is taken care of as described above.

Host mode - ISO IN/OUT - Non-Streaming mode

The available slot for HS ISO within a micro-frame is up to 80%, which means that the buffering could extend until that time is exhausted, with the resulting side effect of wasted bandwidth. It does work, but can become a low efficiency method.

Using Non-Streaming mode, 760 byte transfer, using a Max packet size of 370 bytes, MULT=3.

For OUT direction:

- After the SOF is sent, the host controller begins fetching the data from memory to the TX FIFO.
- Only after the first 370 bytes are fetched (a full packet), it sends the first OUT token to the line.
- During the first OUT/DATA, the controller continues fetching data for the remaining packets (370+20 bytes).
- Controller continues fetching data (within the limit of the buffer size) until all data is fetched and all packets are sent;
- Non-Streaming mode is the same behavior as Streaming mode with the TX Fill level set to the same size as the TX FIFO.

- If the packet size is 1024 for a transfer size of 3072, MULT=3, the behavior is the same as in the previous example, except that when the first packet starts being sent, the controller backfills the TX FIFO with the remaining data for the packet, as soon as one burst of data space is available. The behavior is the same for each of the MULT=3 packets.
- For packet size = 1024 bytes, TX watermark set to 512 and TX FIFO size is 2048, the controller fills the TX FIFO to 1024 bytes before sending the OUT token.

For IN direction:

- After the SOF is sent, the Host Controller starts fetching the iTD from system memory. As soon as the iTD has been read, it issues the IN token if the RX buffer is empty.
- While data is being received from the device and being stored to the RX buffer, the controller writes data to system memory as soon as one burst worth of data is available.
- Only after all data has been stored from the RX buffer to system memory does the Host Controller issue the second IN, and the same for the third IN.
- When sending the IN, if the RX buffer is not empty at the calculated time, the host delays issuing IN token until the buffer is empty of packet data.
- If the packet size is 1024 for a transfer size of 3072, MULT=3, the behavior is the same, each packet is taken care of as described above.

Host mode - Bulk OUT - Streaming mode

For every packet in a transfer, the Host Controller pre-fetches the data until the TX FIFO is filled up to the level specified by the TXFIFOTHRES Register. Only then is the OUT token sent while the controller continues fetching more data to the TX FIFO. This is the same TXFIFOTHRES behavior as in ISO OUT.

TXFIFOTHRES is set in number of bursts; that is, if TXFIFOTHRES=2, the actual watermark level is, for example, 2xINCR8, or 2x (INCR of VUSB_HS_TX_BURST length)

- If the TX FIFO is full, the Host Controller "refetches" data for the next packet as soon as there is one burst worth of free space available on the TX FIFO, regardless of the current packet being sent. So as soon as the first bytes are transmitted for the current packet, the Host Controller starts fetching for the next one. This method is valid in Streaming and Non-Streaming modes.

Host mode - Bulk IN - Streaming mode

Host issues IN token when the RX FIFO is empty which is valid for all packets in a transfer (qTD).

- The Host Controller starts sending data from RX FIFO to system memory as soon as there is one burst worth of data available on the RX FIFO (example: 1x INCR8, or 1x(INCR of VUSB_HS_TX_BURST length)), regardless of the current packet being received. This is valid in Streaming and Non-Streaming modes.

Host mode - Bulk OUT - Non-Streaming mode

Host issues OUT token when entire data packet is in the TX FIFO, or the FIFO is full.

- The Host Controller starts fetching data for the next packet as soon as there is one burst worth of free space available on the TX FIFO, regardless of the current packet being sent. So as soon as the first bytes are transmitted for the current packet, the Host Controller starts fetching for the next one. This is valid in Streaming and Non-Streaming modes.

Host mode - Bulk IN - Non-Streaming mode

Host issues IN token when the RX FIFO is empty.

The Host Controller starts sending data to system memory as soon as there is one burst worth of data available on the RX FIFO, regardless of the current packet being received. This is valid in Streaming and Non-Streaming modes.

11.5.2 Clock Control and Enables

The USB OTG and Host Controllers have separate clock enables for their system and USB clocks.

The USB Clock/Reset Control Register (APULL_CTRL0) is used to enable the AXI interface to the USB OTG and Host controllers.

11.5.3 Programming Guidelines

Both USB controllers receive their clocks from their respective USB UTMI physical layer interfaces.

To use the USB EHCI host controller or the USB EHCI OTG controller their physical layer interfaces must be initialized prior to any USB controller operation such as register write access.

11.6 Register Descriptions

A complete description of the USB Host and OTG Controller Registers is located in Appendix A.

12 WatchDog Timer (WDT)

The watchdog timer regains control in case of system failure (due to a software error) to increase application reliability. The WDT can generate a reset or an interrupt when the counter reaches a given timeout value.

- The 88MC200 WDT supports the following features:
- The WDT module gets the clock from APB clock
- 32-bit down counter with the minimal timeout value of 65536
- Configurable reset or interrupt generation with the given timeout value
- Support eight types of reset pulse length

12.1 Functional Description

12.1.1 Counter Operation

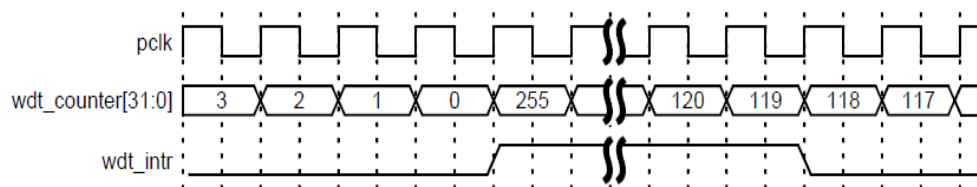
The Watchdog counter descends from a preset (timeout) value to zero. The timeout value is obtained by the formula of $2^{(16 + \text{WDT.TORR.TOP\_INIT})}$ or $2^{(16 + \text{WDT.TORR.TOP})}$. The register bit WDT.TORR.TOP_INIT is only used to initialize timeout period for the first counter restarts, which should be written after reset and before the WDT is enabled. The register bit WDT.TORR.TOP is used to select the timeout period from which the WDT count restarts. Depending on the output response mode selected, when the counter reaches zero, either a system reset or an interrupt occurs. The output response mode is set using the WDT.CR.RMOD register bit. WDT.CR.RMOD = 0 generates a system reset and WDT.CR.RMOD = 1 first generates an interrupt. If it is not cleared before a second timeout occurs then, a system reset is generated.

Users can restart the counter to its initial value (timeout value) by writing to the restart register WDT.CRR[7:0] at any time. The process of restarting the watchdog counter is sometimes referred to as "kicking the dog." As a safety feature to prevent accidental restarts, the value 0x76 must be written to the current counter value register (WDT.CRR).

12.1.2 Interrupt

The WDT can be programmed to generate an interrupt (and then a system reset) when a timeout occurs. When WDT.CR.RMOD is programmed to 1, the WDT generates an interrupt. If it is not cleared by the time a second timeout occurs, then it generates a system reset. If a restart occurs at the same time the watchdog counter reaches zero, an interrupt is not generated. [Figure 35](#) shows the timing diagram of the interrupt being generated and cleared. The interrupt is cleared by reading the WDT.EOI register in which no kick is required. The interrupt can also be cleared by a "kick" (watchdog counter restart).

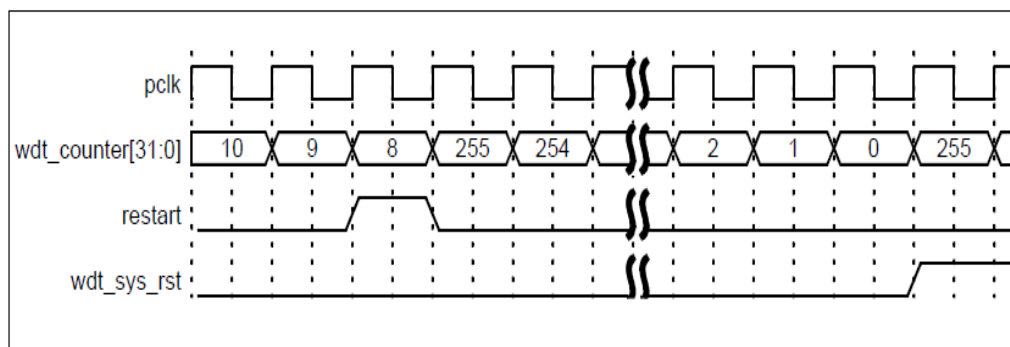
Figure 35: Interrupt Generation



12.1.3 System Reset

When bit WDT.CR.RMOD is programmed to 0, the WDT generates a system reset when a timeout occurs. [Figure 36](#) shows the timing diagram of the WDT system reset.

Figure 36: Counter Restart and System Reset



12.1.4 Reset Pulse Length

The reset pulse length is the number of pclk cycles for which a system reset is asserted. When a system reset is generated, it remains asserted for the number of cycles specified by the reset pulse length or until the system is reset (by the Reset Controller). A counter restart has no effect on the system reset once it has been asserted. The reset pulse length is set by programmed the WDT.CR.RPL register field. The register bits can be programmed to eight types of pulse length. The reset pulse is selected by balancing reset reliability and reset latency. A longer reset pulse provides a more reliable reset but may result in longer reset latency.

12.2 Initialization Sequence

When the counter reaches zero, depending on the output response mode selected, either system reset or an interrupt occurs.

The following sequence of operations must be followed to start the watchdog timer.

- Configure the WDT in Generate Reset mode by setting WDT.CR.RMOD to 0

- Configure timeout value
- Set WDT.CR.WDT.EN to 1'b1

The following sequence of operations must be followed to start the watchdog timer to generate an interrupt.

- Configure WDT in Generate Interrupt mode by setting WDT.CR.RMOD to 1
- Configure timeout value
- Set WDT.CR.WDT.EN to 1'b1

12.3 Register Description

A detailed description of the WDT Registers is located in Appendix A.



88MC200 Microcontroller Datasheet

13

Quad Serial Peripheral Interface (QSPI) Controller

This chapter describes the connectivity and operation of the two Quad-Serial Peripheral Interface Controllers QSPI0 and QSPI1 that are present on Marvell 88MC200 microcontroller. Both QSPI0 and QSPI1 are identical in operation. QSPI0 is connected with in-package serial flash and QSPI1 can be used to connect with external slave devices.

13.1 Overview

The QSPI controller is a synchronous serial peripheral that can be connected to a variety of slave devices that communicate using SPI protocol for data transfer. The QSPI controller always operates as a master and supports standard single bit, and high performance dual/quad output SPI as well as dual/quad I/O SPI. The QSPI Controller has an extremely flexible architecture where the command type, instruction encode, amount of data to be transferred and other parameters are all configurable through memory mapped registers.

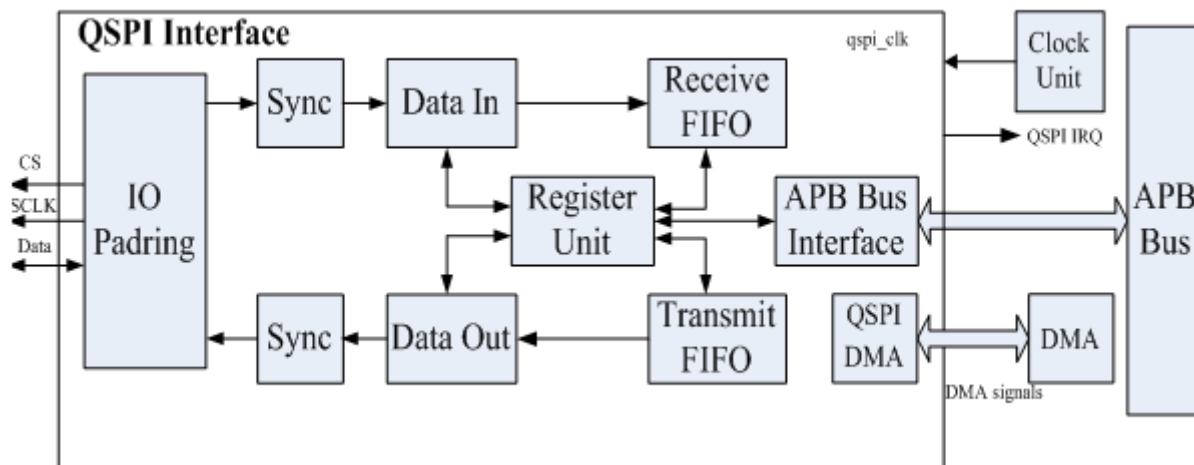
13.2 Features List

- Supports Standard SPI protocol with single bit Data In and Data Out
- Supports dual/quad output operations
- Supports dual/quad I/O operations
- Supports DMA and non-DMA modes for data transfer
- Separate FIFO for transmit and receive with the length of 8*32 bit
- Support for interrupts for a variety of events and conditions related to FIFOs
- 200 Mbps maximum serial data rate in quad mode with 50 MHz functional clock

13.3 Block Diagram

[Figure 37](#) is a diagram of the QSPI controller block.

Figure 37: Block Diagram of the QSPI Controller



13.4 IO Description

Table 37 lists the external signals between QSPI1 controllers on the 88MC200 microcontroller and the external peripherals.

Table 37: External Interface (QSPI1)

Input/output	Name	Width/bit	Description
output	QSPI1_CLK	1	QSPI1 bit clock
output	QSPI1_SS _n	1	QSPI1 chip select, Active low
IO	QSPI1_D0	1	Data I/O 0
IO	QSPI1_D1	1	Data I/O 1
IO	QSPI1_D2	1	Data I/O 2
IO	QSPI1_D3	1	Data I/O 3

Note: General Purpose I/O pins are multiplexed with QSPI1 pins and therefore software must configure the appropriate registers in the Pinmux to use them as QSPI1 pins. Refer to Chapter 3 Pinmux and Appendix Section1 Pinmux registers for more details.

13.5 Functional Description

Serial data is transferred between 88MC200 processor and serial peripheral through the FIFOs in the QSPI controller. QSPI always operates as a master providing the Serial bit clock and Chip-Select or Frame-Sync. The controller supports both the DMA and non-DMA modes of transferring data.

13.5.1 Basic Operation

QSPI always operates as a master and can be configured to generate read or write transactions to the attached slave device. There are two varieties of frames that can be generated by the QSPI – Read frame and Write frame. A Read frame basically consists of instruction encode, address of the location to read from and data itself. A Write frame consists of a similar structure and consists of

again instruction, address followed by outputting data. Some attached slaves that need a few extra cycles for data setup for high performance operation can be supported by configuring the QSPI to generate “dummy clocks.”

For Write transactions, the required fields are:

- Instr.INSTR -- determines the instruction encode or the command
- HdrCnt.INSTR_CNT -- determines the number of clocks or bytes required for instruction encode
- HdrCnt.ADDR_CNT – determines number of bytes or clocks of address
- Addr.ADDR – determines the address inside the slave
- HdrCnt.DUMMY_CNT – determines the number of extra or dummy clocks required by a slave

For Read transaction, the required fields are:

- Instr.INSTR -- determines the instruction encode or the command.
- HdrCnt.INSTR_CNT -- determines the number of clocks or bytes required for instruction encode
- HdrCnt.ADDR_CNT – determines number of bytes or clocks of address.
- Addr.ADDR – determines the address inside the slave
- HdrCnt.DUMMY_CNT – determines the number of extra or dummy clocks required by a slave
- DInCnt.DATA_IN_CNT – number of bytes of data to be transferred

QSPI Single, Dual or Quad mode operation is configurable by programming the Conf.DATA_PIN and Conf.ADDR_PIN fields.

Different values on Conf.DATA_PIN signify:

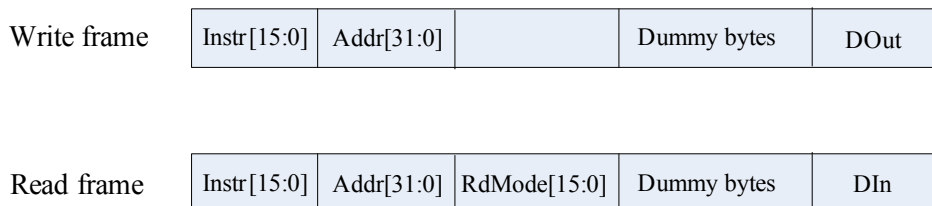
- 00: Use 1 serial interface pin (use in single mode)
- 01: Use 2 serial interface pins (use in dual mode)
- 10: Use 4 serial interface pins (use in quad mode)

Different values on Conf.ADDR_PIN signify:

- 0: Use one serial interface pin
- 1: Use the number of pins as indicated in Conf.DATA_PIN ()

13.5.2 Serial Flash Data Format

Figure 38: Frame of Data Format for Serial Flash Access



- Instr - Serial Interface Instruction
 After Conf.XFER_START is set to 1, the content of the Instr register is shifted out to the serial interface. HdrCnt.INSTR_CNT determines how the content is shifted out.
 - When HdrCnt.INSTR_CNT = 0, the content of this register is not shifted out to the serial interface

- When HdrCnt.INSTR_CNT = 1, bits [7:0] are shifted out
- When HdrCnt.INSTR_CNT = 2, bits [15:8] are shifted out first, followed by bits [7:0]

■ Addr - Serial Interface Address

After the contents of the Instr register is shifted out, the content of the Addr register is shifted out to the serial interface. HdrCnt.ADDR_CNT determines how the content of the Addr register is shifted out on the serial interface.

- When HdrCnt.ADDR_CNT = 0, the content of this register is not shifted out to the serial interface
- When HdrCnt.ADDR_CNT = 1, bits [7:0] are shifted out
- When HdrCnt.ADDR_CNT = 2, bits [15:8] are shifted out first, followed by bits [7:0]
- When HdrCnt.ADDR_CNT = 3, bits [23:16] are shifted out first, followed by bits [15:8], then bits [7:0]
- When HdrCnt.ADDR_CNT = 4, bits [31:24] are shifted out first, followed by bits [23:16], then bits [15:8] and finally bits [7:0]

■ RdMode - Serial Interface Read Mode

After the contents of the Addr register is shifted out, the content of the RdMode register is shifted out to the serial interface. HdrCnt.RM_CNT determines how the contents of the RdMode register are shifted out.

- When HdrCnt.RM_CNT = 0, the content of this register is not shifted out to the serial interface
- When HdrCnt.RM_CNT = 1, bits [7:0] are shifted out
- When HdrCnt.RM_CNT = 2, bits [15:8] are shifted out first, followed by bits [7:0]
- Dummy_byte bytes to shift out to the serial interface after the contents of RdMode register is shifted out. The number of dummy bytes shifted out is determined by HdrCnt.DUMMY_CNT. Different values on HdrCnt.DUMMY_CNT signify
 - 00: 0 byte
 - 01: 1 byte
 - 10: 2 bytes
 - 11: 3 bytes

■ DOut - Serial Interface Data Out

Data written to the DOut register is stored in the 8X32 bit Write FIFO. After the contents of the Instruction register (Instr), the Address register (Addr), the Read Mode register (RdMode) and Dummy value are transferred out to the serial interface, the data in the Write FIFO is shifted out. The serial interface clock stops when a Write FIFO empty condition occurs, i.e. Cntl.WFIFO_EMPTY = 1. The clock restarts when Write FIFO is not empty, i.e. Cntl.WFIFO_EMPTY = 0. Conf.BYTE_LEN determines the number of bytes shifted out on the serial interface.

- When Conf.BYTE_LEN = 0, only the first byte i.e bits [7:0] of the Write FIFO is shifted out with bit 7 shifted out first and bit 0 shifted out last.
- When Conf.BYTE_LEN = 1, all four bytes from each the Write FIFO are shifted out with bits [7:0] are shifted out (bit 7 shifted out first and bit 0 shifted out last), followed by bits [15:8] (bit 15 shifted out first and bit 8 shifted out last), then bits [23:16] (bit 23 shifted out first and bit 16 shifted out last) and finally bits [31:24] (bit 31 shifted out first and bit 24 shifted out last).



Note

To avoid a Write FIFO overflow condition (Cntl.WFIFO_OVRFLW = 1), check if Cntl.WFIFO_FULL = 0 before writing to the DOut register.

■ **DIn - Serial Interface Data In**

For read transfers, Conf.RW_EN = 0, data from the serial interface input pins are shifted in and stored in a 8X32 bit Read FIFO. The contents of the Read FIFO are read from this register. The serial interface clock stops when a Read FIFO full condition occurs, i.e Cntl.RFIFO_FULL= 1. The clock restarts when Read FIFO is not full, that is, Cntl.RFIFO_FULL= 0.

- When Conf.BYTE_LEN = 0, data is shifted into bits [7:0] of the Read FIFO
- When Conf.BYTE_LEN = 1, data is shifted into bits [7:0] first, followed by bits [15:8], then bits [23:16] and finally bits [31:24]



Note

To avoid a Read FIFO underflow condition, Conf.RFIFO_UNDRFLW = 1, check if Conf.RFIFO_EMPTY=0 before reading the DIn register.

DMA transfer is supported in QAPI functions. The specific bits in register QSPI.CONF2 must be set for the DMA transfer. [Figure 39](#), [Figure 40](#), [Figure 41](#), and [Figure 42](#) are flow charts for Read and Write transactions using DMA and non-DMA operation of the QSPI Controller.

Figure 39: Non-DMA Mode Read Flow

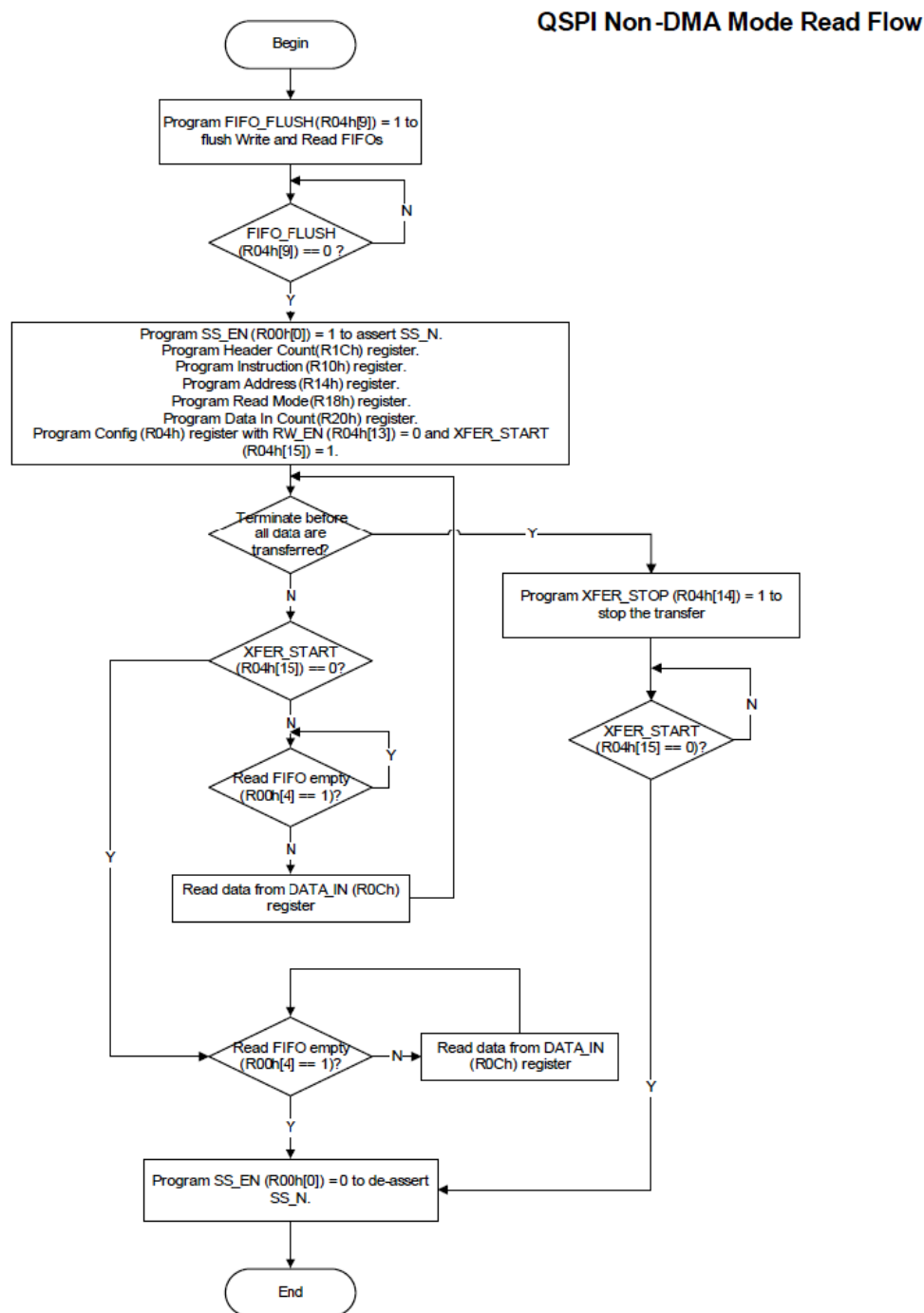


Figure 40: Non-DMA Mode Write Flow

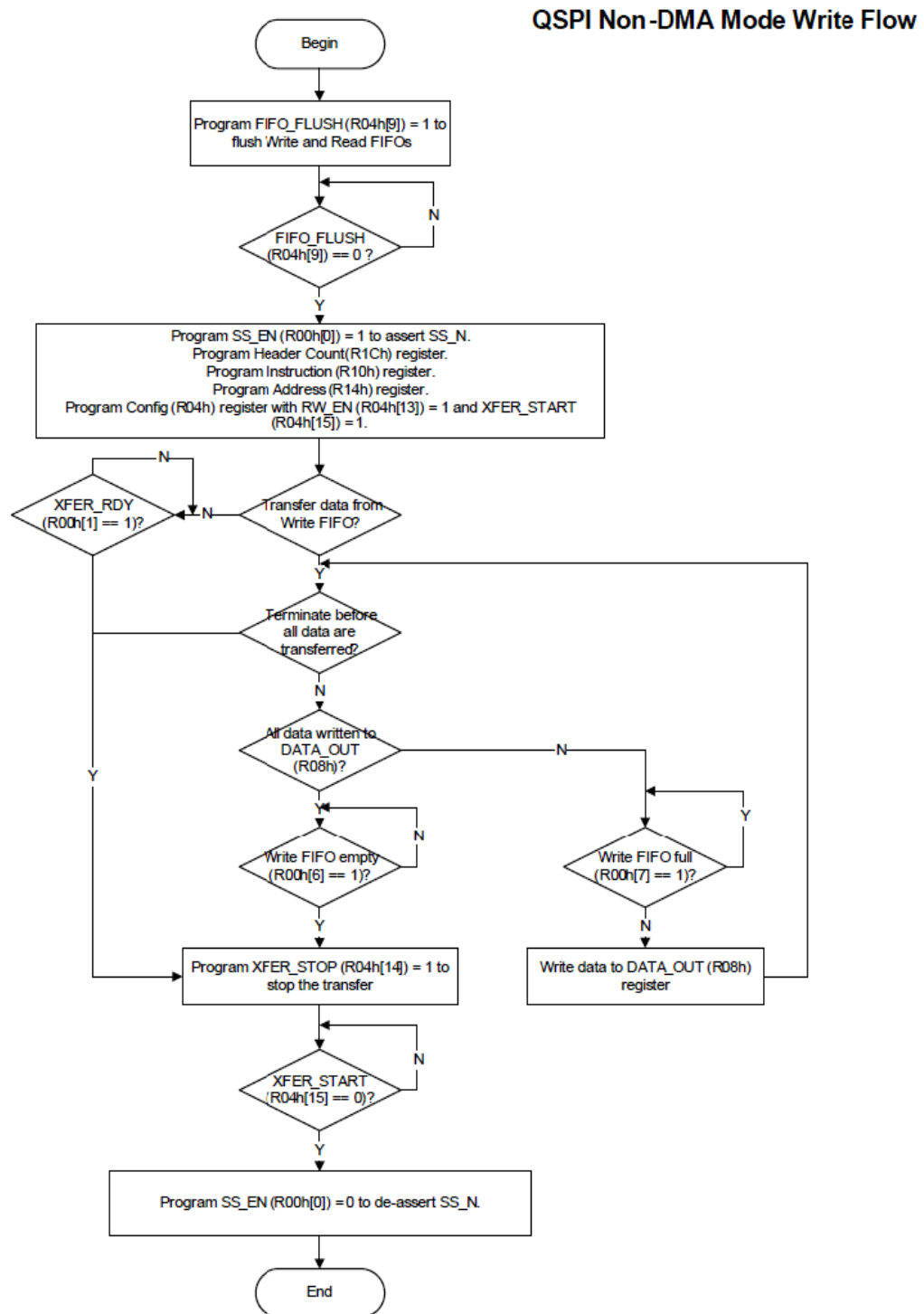


Figure 41: DMA Mode Write Flow

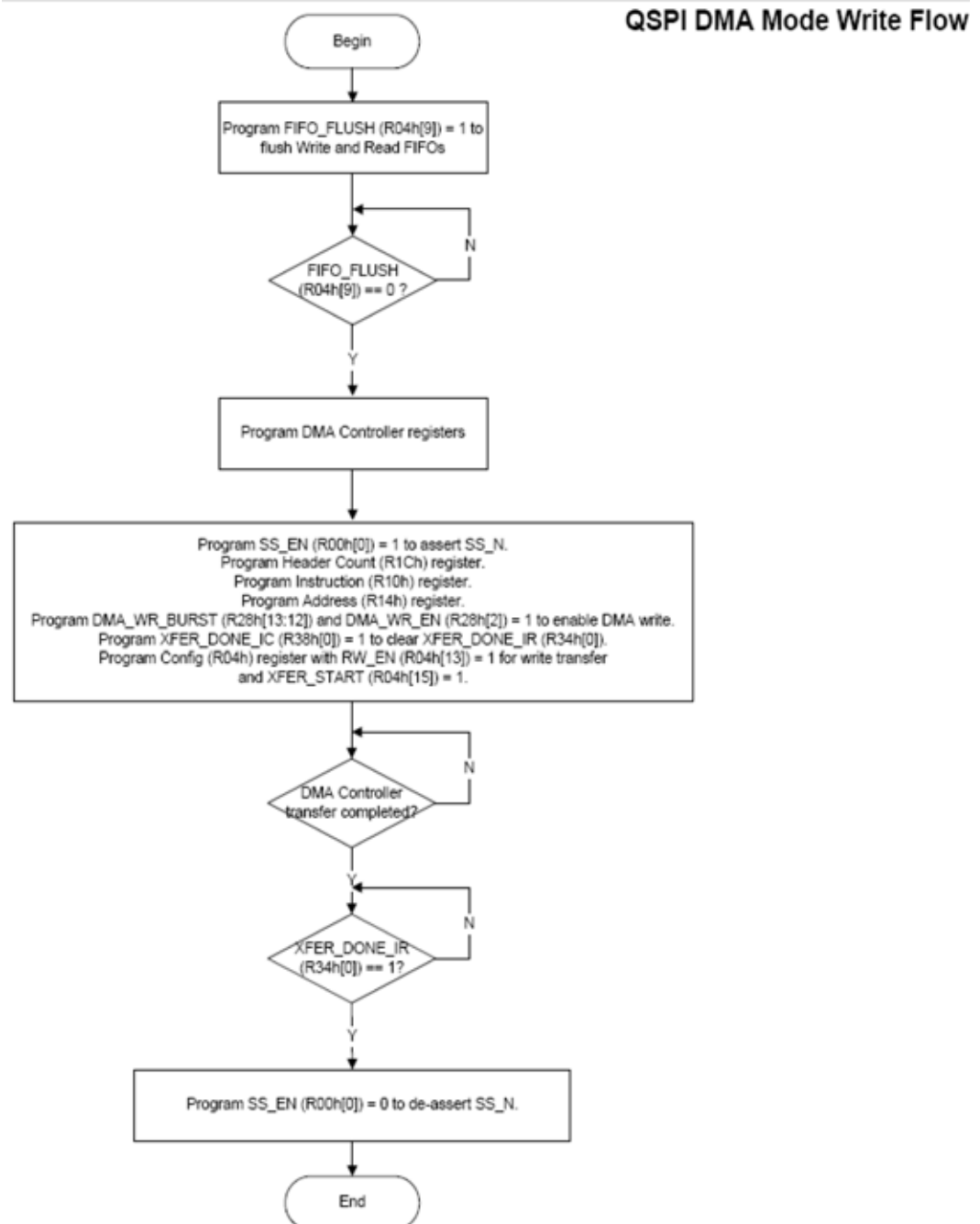
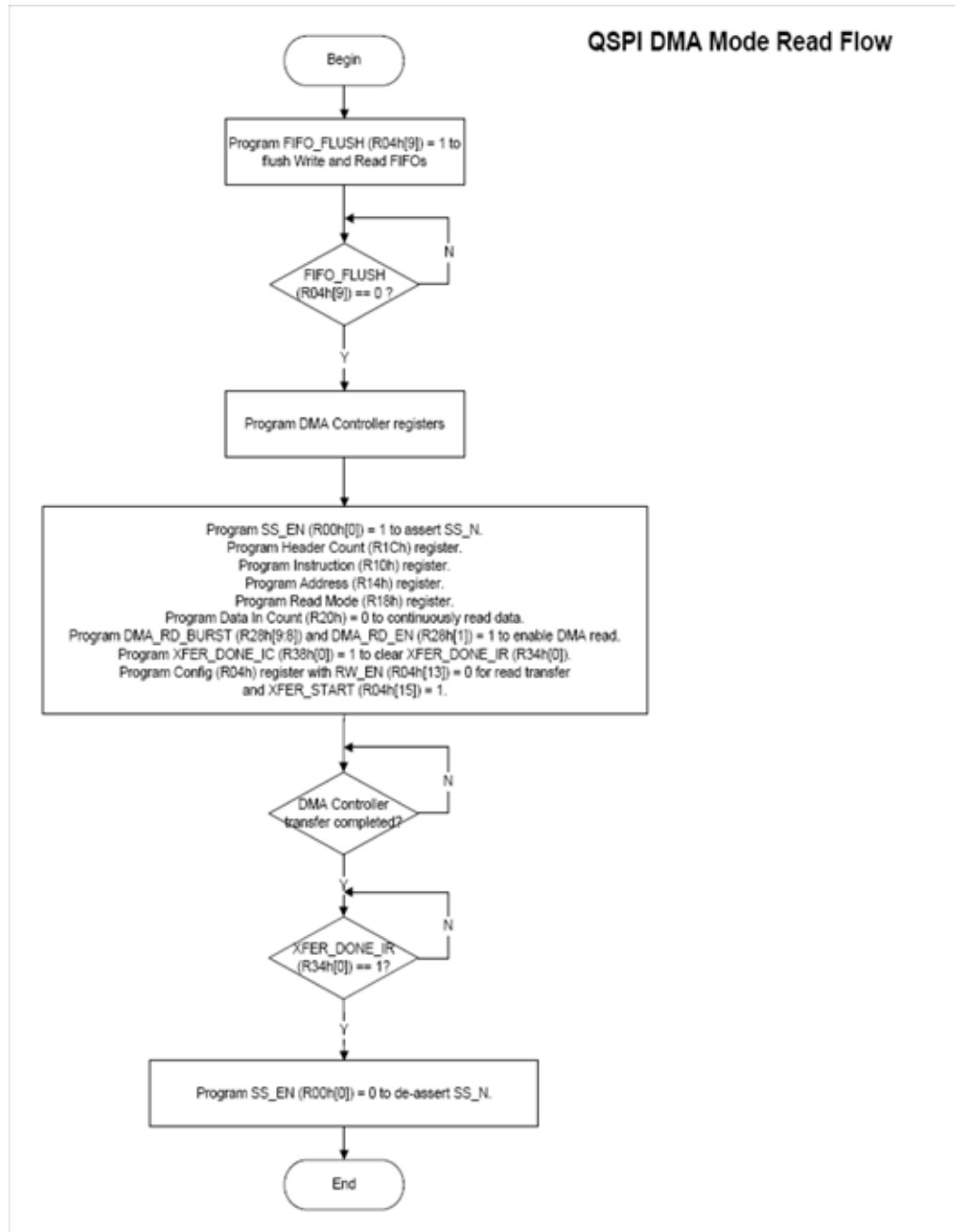


Figure 42: DMA Read Flow



13.6 Usage Models and Programming Notes

This section describes how the QSPI controller can be configured to achieve any basic command sequence as supported by slave serial interface.

13.7 QSPI0 Interface to In-package Serial Flash

The QSPI0 controller is dedicated to communicate with in-package serial flash memory.

13.7.1 Basic Read to Serial Flash Without DMA and Using Polling

A basic Read command or semantic for serial flash memory consists of an Instruction phase, address phase, and data phases. Required fields are Cntl.SS_EN, Conf.XFER_START, Conf.RW_EN, HdrCnt.INSTR_CNT, HdrCnt.ADDR_CNT, Instr.INSTR, Addr.ADDR, DInCnt.DATA_IN_CNT.

The flow below describes how to perform the Read Data (03h) command supported by serial flash and read 4 bytes from address 32'h0000_0000.

1. Program Conf.FIFO_FLUSH=1
2. Wait for Conf.FIFO_FLUSH=0
3. Program Cntl.SS_EN=1
4. Program HdrCnt.INSTR_CNT=2'b01
5. Program HdrCnt.ADDR_CNT=3'b011
6. Program Instr.INSTR=16'h0003
7. Program Addr.ADDR=32'h00000000
8. Program DInCnt.DATA_IN_CNT=20'h00004
9. Program Conf.RW_EN=0
10. Program Conf.XFER_START=1
11. Wait for Cntl.RFIFO_EMPTY=0
12. Read Data from Din register
13. Program Cntl.SS_EN=0

13.7.1.1 Page Program to Serial Flash Without DMA and Using Polling

A Page program semantic for flash memory consists of Instruction phase, address phase and data phases. The page program command (02h) must be preceded with a Write Enable instruction and the status register bit WEL should be set to 1 before the device accepts the page program. Required fields are Cntl.SS_EN, Conf.XFER_START, Conf.RW_EN, HdrCnt.INSTR_CNT, HdrCnt.ADDR_CNT, Instr.INSTR, Addr.ADDR, DOut.DATA_OUT.

The flow below explains how page program (02h) command is performed to write 4 bytes to address 32'h0000_0000.

1. Program Conf.FIFO_FLUSH=1
2. Wait for Conf.FIFO_FLUSH=0
3. Program Cntl.SS_EN=1
4. Program HdrCnt.INSTR_CNT=2'b01
5. Program HdrCnt.ADDR_CNT=3'b011
6. Program Instr.INSTR=16'h0002
7. Program Addr.ADDR=32'h00000000
8. Program Conf.RW_EN=1
9. Program Conf.XFER_START=1
10. Wait for Cntl.WFIFO_FULL=0
11. Write Data to DOut register

12. Wait for Cntl.WFIFO_EMPTY=1
13. Program Cntl.SS_EN=0

13.8 Register Description

A complete description of the QSPI0 and QSPI1 registers is located in Appendix A.



88MC200 Microcontroller Datasheet

14 In-Package Flash

14.1 Overview

The 88MC200 microcontroller has an 8Mbit on-chip serial flash memory. Internal QSPI0 interface is dedicated to the access of on-chip serial flash.

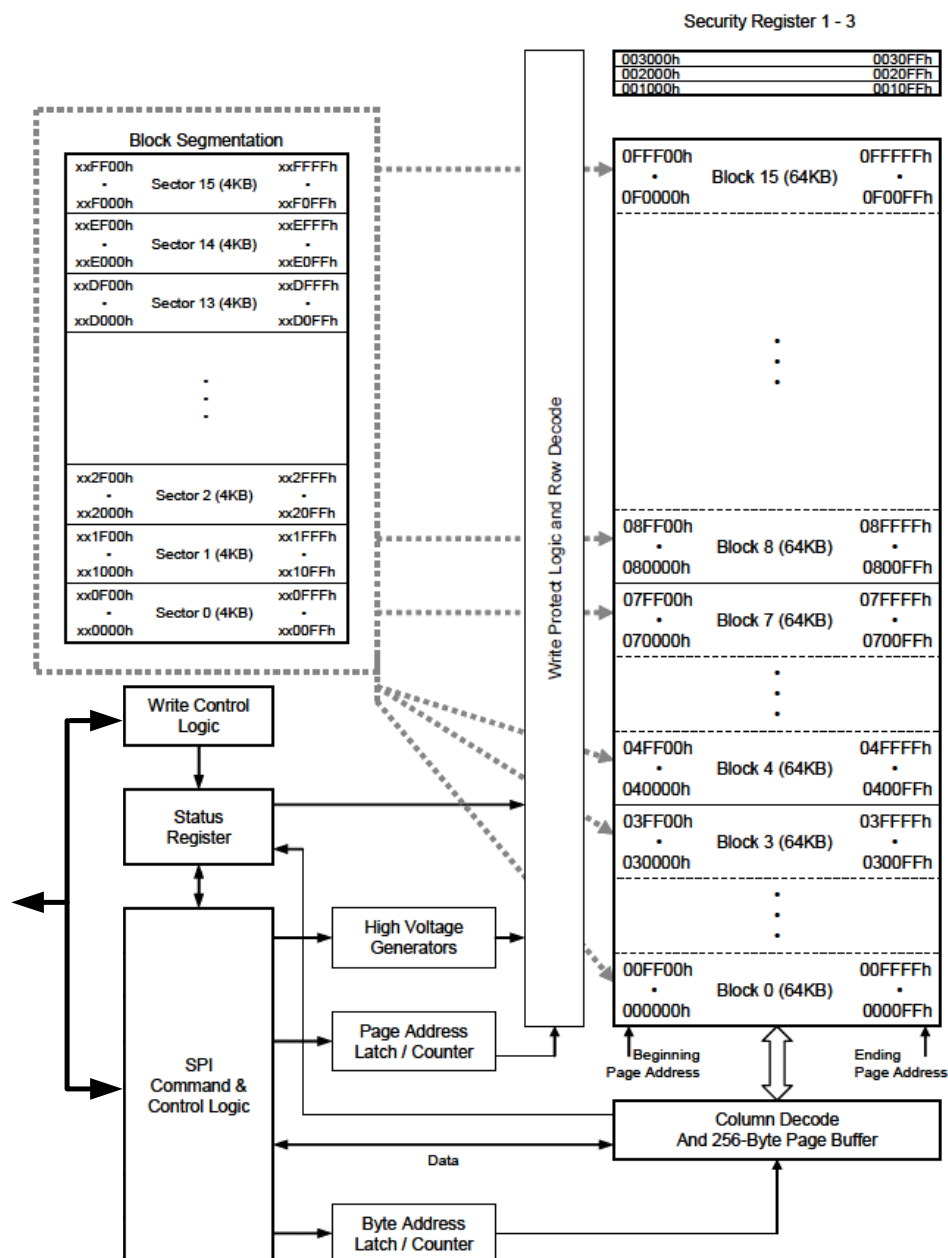
The serial flash is organized into 4,096 programmable pages of 256-bytes each. Up to 256 bytes can be programmed at a time. Pages can be erased in groups of 16 (4KB sector erase), groups of 128 (32 KB block erase), groups of 256 (64 KB block erase) or the entire chip (chip erase). The device has 256 erasable sectors and 16 erasable blocks, respectively. The small 4KB sectors allow for greater flexibility in applications that require data and parameter storage. (See [Figure 43.](#))

14.2 Features

- 8M-bit/1M-byte (1,048,576)
- 256-byte per programmable page with configurable length 1 to 256
- Uniform Sector Erase (4K-bytes)
- Uniform Block Erase (32K and 64K-bytes)
- Erase/Program Suspend & Resume
- Standard/Dual/Quad SPI support
- Efficient “Continuous Read Mode”
- Top/Bottom, 4KB complement array protection
- Lock-Down and OTP array protection
- 64-Bit Unique Serial Number for each device
- 200 Mbps maximum serial data rate in quad mode with 50 MHz functional clock
- More than 100,000 erase/program cycles
- More than 20-year data retention

14.3 Block Diagram

Figure 43: Serial Flash Memory Block Diagram



14.4 Functional Description

14.4.1 QSPI0 Interface

14.4.1.1 Standard SPI Operation

Standard SPI operation uses the single data bit to serially write instructions, addresses or data to the device on the rising edge of clock and read data or status from the device on the falling edge of clock. SPI bus operation Mode 0 (0,0) and 3 (1,1) are supported. The primary difference between Mode 0 and Mode 3 concerns the normal state of clock when the SPI bus master is in standby and data is not being transferred to the Serial Flash. For Mode 0, the clock is normally low. For Mode 3, the clock is normally high.

14.4.1.2 Dual SPI Operation

The device supports Dual SPI operation when using the “Fast Read Dual Output (3Bh)” and “Fast Read Dual I/O (BBh)” instructions. These instructions allow data to be transferred to or from the device at two to three times the rate of ordinary Serial Flash devices.

14.4.1.3 Quad SPI Operation

The device supports Quad SPI operation when using the “Fast Read Quad Output (6Bh)”, “Fast Read Quad I/O (EBh)”, “Word Read Quad I/O (E7h)” and “Octal Word Read Quad I/O (E3h)” instructions. These instructions allow data to be transferred to or from the device six to eight times the rate of ordinary Serial Flash. The Quad Read instructions offer a significant improvement in continuous and random access transfer rates. Quad SPI instructions require the non-volatile Quad Enable bit (QE) in Status Register-2 to be set.

14.4.2 Write Protection

Applications that use non-volatile memory must take into consideration the possibility of noise and other adverse system conditions that may compromise data integrity. To address this concern, the device provides several means to protect the data from inadvertent writes.

14.4.2.1 Write Protect Features

- Time delay write disable after Power-up
- Write enable/disable instructions and automatic write disable after erase or program
- Software write protection using Status Register
- Write Protection using Powerdown instruction
- Lock Down write protection until next power-up

After power-up the device is automatically placed in a write-disabled state with the Status Register Write Enable Latch (WEL) set to a 0. A Write Enable instruction must be issued before a Page Program, Sector Erase, Block Erase, Chip Erase or Write Status Register instruction is accepted. After completing a program, erase or write instruction the Write Enable Latch (WEL) is automatically cleared to a write- disabled state of 0.

Software controlled write protection is facilitated using the Write Status Register instruction and setting the Status Register Protect (SRP0, SRP1) and Block Protect (CMP, SEC, TB, BP2, BP1 and BP0) bits. These settings allow a portion as small as 4KB sector or the entire memory array to be configured as read only. Refer to the Status Register section for further information. Additionally, the Powerdown instruction offers an extra level of write protection as all instructions are ignored except the Release Powerdown instruction.

14.5 Control and Status Registers

The Read Status Register-1 and Status Register-2 instructions can be used to provide status on the availability of the Flash memory array, if the device is write enabled or disabled, the state of write protection, Quad SPI setting, Security Register lock status and Erase/Program Suspend status. The Write Status Register instruction can be used to configure the device write protection features, Quad SPI setting and Security Register OTP lock. Write access to the Status Register is controlled by the state of the non-volatile Status Register Protect bits (SRP0, SRP1), the Write Enable instruction, and during Standard/Dual SPI operations.

14.5.1 Status Register

14.5.1.1 BUSY

BUSY is a read only bit in the status register (S0) that is set to a 1 state when the device is executing a Page Program, Quad Page Program, Sector Erase, Block Erase, Chip Erase, Write Status Register or Erase/Program Security Register instruction. During this time the device ignores further instructions except for the Read Status Register and Erase/Program Suspend instruction (see tW, tPP, tSE, tBE, and tCE in AC Characteristics). When the program, erase or write status/security register instruction has completed, the BUSY bit is cleared to a 0 state indicating the device is ready for further instructions.

14.5.1.2 Write Enable Latch (WEL)

Write Enable Latch (WEL) is a read only bit in the status register (S1) that is set to 1 after executing a Write Enable Instruction. The WEL status bit is cleared to 0 when the device is write disabled. A write disable state occurs upon power-up or after any of the following instructions: Write Disable, Page Program, Quad Page Program, Sector Erase, Block Erase, Chip Erase, Write Status Register, Erase Security Register and Program Security Register.

14.5.1.3 Block Protect Bits (BP2, BP1, BP0)

The Block Protect Bits (BP2, BP1, BP0) are non-volatile read/write bits in the status register (S4, S3, and S2) that provide Write Protection control and status. Block Protect bits can be set using the Write Status Register Instruction (see tW in AC characteristics). All, none or a portion of the memory array can be protected from Program and Erase instructions (see Status Register Memory Protection table). The factory default setting for the Block Protection Bits is 0, none of the array is protected.

14.5.1.4 Top/Bottom Block Protect (TB)

The non-volatile Top/Bottom bit (TB) controls if the Block Protect Bits (BP2, BP1, BP0) protect from the Top (TB=0) or the Bottom (TB=1) of the array as shown in the Status Register Memory Protection table. The factory default setting is TB=0. The TB bit can be set with the Write Status Register Instruction depending on the state of the SRP0, SRP1 and WEL bits.

14.5.1.5 Sector/Block Protect (SEC)

The non-volatile Sector/Block Protect bit (SEC) controls if the Block Protect Bits (BP2, BP1, BP0) protect either 4KB Sectors (SEC=1) or 64KB Blocks (SEC=0) in the Top (TB=0) or the Bottom (TB=1) of the array as shown in the Status Register Memory Protection table. The default setting is SEC=0.

14.5.1.6 Complement Protect (CMP)

The Complement Protect bit (CMP) is a non-volatile read/write bit in the status register (S14). It is used in conjunction with SEC, TB, BP2, BP1 and BP0 bits to provide more flexibility for the array protection. Once CMP is set to 1, previous array protection set by SEC, TB, BP2, BP1 and BP0 is

reversed. For instance, when CMP=0, a top 4KB sector can be protected while the remainder of the array is not; when CMP=1, the top 4KB sector becomes unprotected while the remainder of the array becomes read only. Refer to the Status Register Memory Protection table for details. The default setting is CMP=0.

14.5.1.7 Status Register Protect (SRP1, SRP0)

The Status Register Protect bits (SRP1 and SRP0) are non-volatile read/write bits in the status register (S8 and S7). The SRP bits control the method of write protection: software protection, hardware protection, power supply lock-down, or one-time programmable (OTP) protection. See [Table 38](#).

Table 38: Status Register Protect (SRP1, SR0) Bits

SRP1	SRP0	Status Register	Description
0	X	Software Protection	The Status register can be written to after a Write Enable instruction, WEL=1. [Factory Default is SRP1, SRP0 = (0, 0)]
1	0	Power Supply Lock-Down	Status Register is protected and can not be written to again until the next power-down, power-up cycle. ¹
1	1	One Time Program ²	Status Register is permanently protected and cannot be written to.
¹ When SRP1, SRP0 = (1, 0) a powerdown/powerup cycle changes SRP1, SRP0 to (0, 0) state			
² This feature is available only by special order			

14.5.1.8 Erase/Program Suspend Status (SUS)

The Suspend Status bit is a read-only bit in the status register (S15) that is set to 1 after executing a Erase/Program Suspend (75h) instruction. The SUS status bit is cleared to 0 by Erase/Program Resume (7Ah) instruction as well as a powerdown, powerup cycle.

14.5.1.9 Security Register Lock Bits (LB3, LB2, LB1)

The Security Register Lock Bits (LB3, LB2, LB1) are non-volatile One Time Program (OTP) bits in Status Register (S13, S12, S11) that provide the write protect control and status to the Security Registers. The default state of LB[3:1] is 0, Security Registers are unlocked. LB [3:1] can be set to 1 individually using the Write Status Register instruction. LB[3:1] are One Time Programmable (OTP), once it's set to 1, the corresponding 256-Byte Security Register will become read-only permanently.

14.5.1.10 Quad Enable (QE)

The Quad Enable (QE) bit is a non-volatile read/write bit in the status register (S9) that allows Quad SPI operation. See [Figure 44](#) and [Figure 45](#).

Figure 44: Status Register (1)

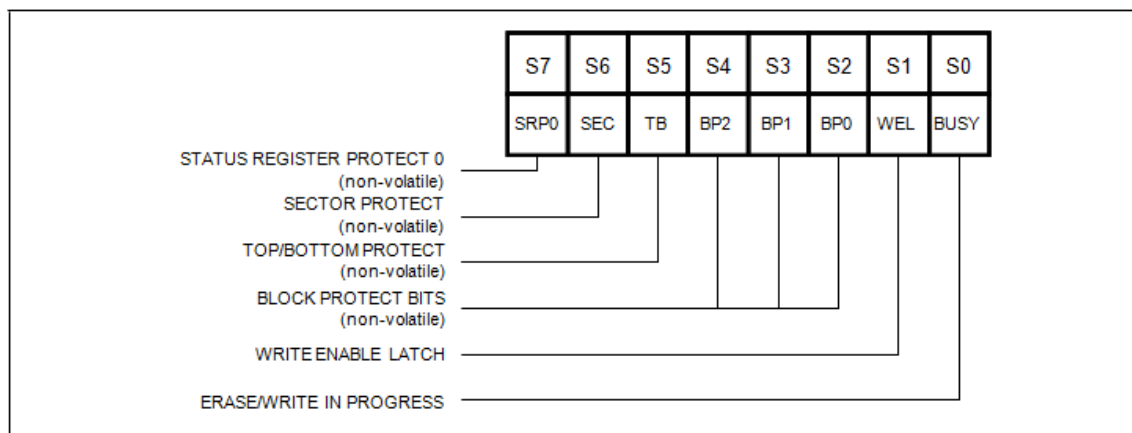


Figure 45: Status Register (2)

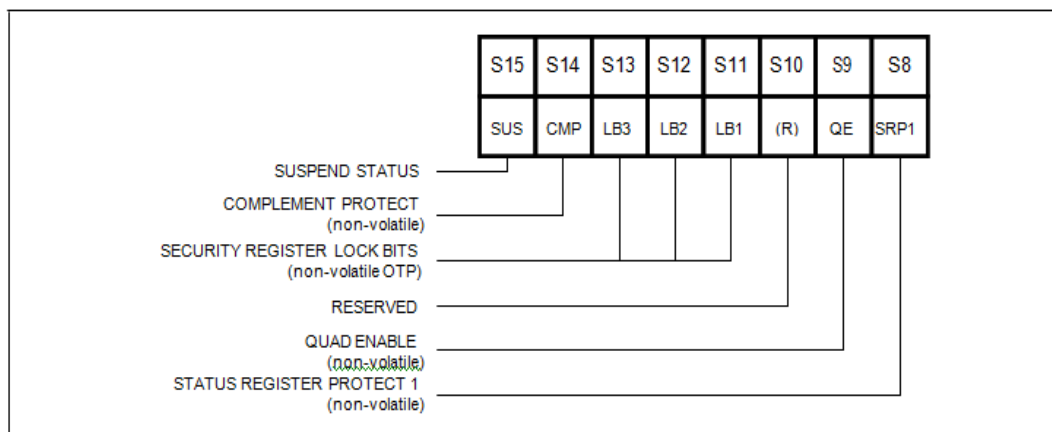


Table 39 describes the Status Register memory protection for CMP=0, while Table 40 does the same for CMP=1.

Table 39: Status Register Memory Protection (CMP = 0)

STATUS REGISTER(1)					(8M-BIT) MEMORY PROTECTION(2)			
SEC	TB	BP2	BP1	BP0	BLOCK(S)	ADDRESSES	DENSITY	PORTION
X	X	0	0	0	NONE	NONE	NONE	NONE
0	0	0	0	1	15	0F0000h – 0FFFFFFh	64KB	Upper 1/16
0	0	0	1	0	14 and 15	0E0000h – 0FFFFFFh	128KB	Upper 1/8
0	0	0	1	1	12 thru 15	0C0000h – 0FFFFFFh	256KB	Upper 1/4
0	0	1	0	0	8 thru 15	080000h – 0FFFFFFh	512KB	Upper 1/2
0	1	0	0	1	0	000000h – 00FFFFh	64KB	Lower 1/16
0	1	0	1	0	0 and 1	000000h – 01FFFFh	128KB	Lower 1/8
0	1	0	1	1	0 thru 3	000000h – 03FFFFh	256KB	Lower 1/4
0	1	1	0	0	0 thru 7	000000h – 07FFFFh	512KB	Lower 1/2
0	0	1	1	X	0 thru 15	000000h – 0FFFFFFh	1MB	ALL
0	X	1	0	1	0 thru 15	000000h – 0FFFFFFh	1MB	ALL
0	1	1	11	X	0 thru 15	000000h – 0FFFFFFh	1MB	ALL
1	X	1	1	1	0 thru 15	000000h – 0FFFFFFh	1MB	ALL
1	0	0	0	1	15	0FF000h – 0FFFFFFh	4KB	Upper 1/256
1	0	0	1	0	15	0FE000h – 0FFFFFFh	8KB	Upper 1/128
1	0	0	1	1	15	0FC000h – 0FFFFFFh	16KB	Upper 1/64
1	0	1	0	X	15	0F8000h – 0FFFFFFh	32KB	Upper 1/32
1	0	1	1	0	15	0F8000h – 0FFFFFFh	32KB	Upper 1/32
1	1	0	0	1	0	000000h – 000FFFh	4KB	Lower 1/256
1	1	0	1	0	0	000000h – 001FFFh	8KB	Lower 1/128
1	1	0	1	1	0	000000h – 003FFFh	16KB	Lower 1/64
1	1	1	0	X	0	000000h – 007FFFh	32KB	Lower 1/32
1	1	1	1	0	0	000000h – 007FFFh	32KB	Lower 1/32

Notes:
1. X = don't care
2. If any Erase or Program command specifies a memory region that contains protected data portion, this command will be ignored.

Table 40: Status Register Memory Protection (CMP=1)

STATUS REGISTER(1)					(8M-BIT) MEMORY PROTECTION(2)			
SEC	TB	BP2	BP1	BP0	BLOCK(S)	ADDRESSES	DENSITY	PORTION
X	X	0	0	0	0 thru 15	000000h – 0FFFFFFh	1MB	ALL
0	0	0	0	1	0 thru 14	000000h – 0EFFFFh	960KB	Lower 15/16
0	0	0	1	0	0 thru 13	000000h – 0DFFFFh	896KB	Lower 7/8
0	0	0	1	1	0 thru 11	000000h – 0BFFFFh	768KB	Lower 3/4
0	0	1	0	0	0 thru 7	000000h – 07FFFFh	512KB	Lower 1/2
0	1	0	0	1	1 thru 15	010000h – 0FFFFFFh	960KB	Upper 15/16
0	1	0	1	0	2 thru 15	020000h – 0FFFFFFh	896KB	Upper 7/8
0	1	0	1	1	4 thru 15	040000h – 0FFFFFFh	768KB	Upper 3/4
0	1	1	0	0	8 thru 15	080000h – 0FFFFFFh	512KB	Upper 1/2
0	0	1	1	X	NONE	NONE	NONE	NONE
0	X	1	0	1	NONE	NONE	NONE	NONE
0	1	1	11	X	NONE	NONE	NONE	NONE
1	X	1	1	1	NONE	NONE	NONE	NONE
1	0	0	0	1	0 thru 15	000000h – 0EFFFFh	1,020KB	Lower 255/256
1	0	0	1	0	0 thru 15	000000h – 0DFFFFh	1,016KB	Lower 127/128
1	0	0	1	1	0 thru 15	000000h – 0BFFFFh	1,008KB	Lower 63/64
1	0	1	0	X	0 thru 15	000000h – 0F7FFFh	992KB	Lower 31/32
1	0	1	1	0	0 thru 15	000000h – 0F7FFFh	992KB	Lower 31/32
1	1	0	0	1	0 thru 15	001000h – 0FFFFFFh	1,020KB	Upper 255/256
1	1	0	1	0	0 thru 15	002000h – 0FFFFFFh	1,016KB	Upper 127/128
1	1	0	1	1	0 thru 15	004000h – 0FFFFFFh	1,008KB	Upper 63/64
1	1	1	0	X	0 thru 15	008000h – 0FFFFFFh	992KB	Upper 31/32
1	1	1	1	0	0 thru 15	008000h – 0FFFFFFh	992KB	Upper 31/32

Notes:

1. X = don't care
2. If any Erase or Program command specifies a memory region that contains protected data portion, this command will be ignored.

14.5.2 Instructions

The instruction set consists of 35 basic instructions that are fully controlled through the SPI bus (see Instruction Set Table 1 [Table 41], Instruction Set Table 2 [Table 42], and Instruction Set Table 3

[Table 43]). Instructions are initiated with the falling edge of Chip Select. The first byte of data clocked for Data Input provides the instruction code. Data on the Data Input is sampled on the rising edge of clock with most significant bit (MSB) first.

Instructions vary in length from a single byte to several bytes and may be followed by address bytes, data bytes, dummy bytes (don't care), and in some cases, a combination. Instructions are completed with the rising edge of Chip Select. All Read instructions can be completed after any clocked bit. However, all instructions that Write, Program or Erase must complete on a byte boundary (Chip Select driven high after a full 8-bits have been clocked) otherwise the instruction is ignored. This feature further protects the device from inadvertent Writes. Additionally, while the memory is being programmed or erased, or when the Status Register is being written, all instructions except for Read Status Register are ignored until the program or erase cycle has completed.

14.5.2.1 Instruction Set Table 1 (Erase, Program Instructions) ¹

Table 41: Instruction Set Table

INSTRUCTION NAME	BYTE 1 (CODE)	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6
Write Enable	06h					
Write Enable for Volatile Status Register	50h					
Write Disable	04h					
Read Status Register-1	05h	(S7–S0) (2)				
Read Status Register-2	35h	(S15–S8) (2)				
Write Status Register	01h	S7–S0	S15–S8			
Page Program	02h	A23–A16	A15–A8	A7–A0	D7–D0	
Quad Page Program	32h	A23–A16	A15–A8	A7–A0	D7–D0, ... (3)	
Sector Erase (4KB)	20h	A23–A16	A15–A8	A7–A0		
Block Erase (32KB)	52h	A23–A16	A15–A8	A7–A0		
Block Erase (64KB)	D8h	A23–A16	A15–A8	A7–A0		
Chip Erase	C7h/60h					
Erase / Program Suspend	75h					
Erase / Program Resume	7Ah					
Power-down	B9h					
Continuous Read Mode Reset (4)	FFh	FFh				
Notes: 1. Data bytes are shifted with Most Significant Bit first. Byte fields with data in parenthesis “()” indicate data being read from the device. 2. The Status Register contents repeat continuously until Chip Select terminates the instruction. 3. Quad Page Program Input Data: - o IO0 = D4, D0, ... - o IO1 - o = D5, D1, ... - o IO2 = - o D6, D2, ... - o IO3 = - o D7, D3, ... 4. This instruction is recommended when using the Dual or Quad “Continuous Read Mode” feature						

14.5.2.2 Instruction Set Table 2 (Read Instructions)

Table 42: Instruction Set Table 2 (Read Instructions)

INSTRUCTION NAME	BYTE 1 (CODE)	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6
Read Data	03h	A23-A16	A15-A8	A7-A0	(D7-D0)	
Fast Read	0Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0)
Fast Read Dual Output	3Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0, ...) ⁽¹⁾
Fast Read Quad Output	6Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0, ...) ⁽³⁾
Fast Read Dual I/O	BBh	A23-A8 ⁽²⁾	A7-A0, M7-M0 ⁽²⁾	(D7-D0, ...) ⁽¹⁾		
Fast Read Quad I/O	EBh	A23-A0, M7-M0 ⁽⁴⁾	(x,x,x,x, D7-D0, ...) ⁽⁵⁾	(D7-D0, ...) ⁽³⁾		
Word Read Quad I/O ⁽⁷⁾	E7h	A23-A0, M7-M0 ⁽⁴⁾	(x,x, D7-D0, ...) ⁽⁶⁾	(D7-D0, ...) ⁽³⁾		
Octal Word Read Quad I/O ⁽⁸⁾	E3h	A23-A0, M7-M0 ⁽⁴⁾	(D7-D0, ...) ⁽³⁾			
Set Burst with Wrap	77h	xxxxxx, W6-W4 ⁽⁴⁾				

Notes:

1. Dual Output data

IO0 = (D6, D4, D2, D0)

IO1 = (D7, D5, D3, D1)

2. Dual Input Address

IO0 = A22, A20, A18, A16, A14, A12, A10, A8 A6, A4, A2, A0, M6, M4, M2, M0

IO1 = A23, A21, A19, A17, A15, A13, A11, A9 A7, A5, A3, A1, M7, M5, M3, M1

3. Quad Output Data

IO0 = (D4, D0, ...)

IO1 = (D5, D1, ...)

IO2 = (D6, D2, ...)

IO3 = (D7, D3, ...)

4. Quad Input Address

Set Burst with Wrap Input

IO0 = A20, A16, A12, A8, A4, A0, M4, M0 IO0 = x, x, x, x, x, x, W4, x

IO1 = A21, A17, A13, A9, A5, A1, M5, M1 IO1 = x, x, x, x, x, x, W5, x

IO2 = A22, A18, A14, A10, A6, A2, M6, M2 IO2 = x, x, x, x, x, x, W6, x

IO3 = A23, A19, A15, A11, A7, A3, M7, M3 IO3 = x, x, x, x, x, x, x, x

5. Fast Read Quad I/O Data

IO0 = (x, x, x, x, D4, D0, ...) IO1 = (x, x, x, x, D5, D1, ...)

IO2 = (x, x, x, x, D6, D2, ...) IO3 = (x, x, x, x, D7, D3, ...)

6. Word Read Quad I/O Data

IO0 = (x, x, D4, D0, ...) IO1 = (x, x, D5, D1, ...)

IO2 = (x, x, D6, D2, ...) IO3 = (x, x, D7, D3, ...)

7. The lowest address bit must be 0. (A0 = 0)

8. The lowest 4 address bits must be 0. (A0, A1, A2, A3 = 0)

14.5.2.3 Instruction Set Table 3 (ID, Security Instructions)

Table 43: Instruction Set Table 3 (ID, Security Instructions)

INSTRUCTION NAME	BYTE 1 (CODE)	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6
Release Powerdown/ Device ID	ABh	dummy	dummy	dummy	(ID7-ID0) ⁽¹⁾	
Read Unique ID	4Bh	dummy	dummy	dummy	dummy	(ID63-ID0)
Erase Security Registers(2)	44h	A23–A16	A15–A8	A7–A0		
Program Security Registers(2)	42h	A23–A16	A15–A8	A7–A0	D7-D0	D7-D0
Read Security Registers(2)	48h	A23–A16	A15–A8	A7–A0	dummy	(D7-0)

NOTE:

1. The Device ID repeats continuously until Chip Select terminates the instruction.
2. Security Register Address:
Security Register 1: A23-16 = 00h; A15-8 = 10h; A7-0 = byte address
Security Register 2: A23-16 = 00h; A15-8 = 20h; A7-0 = byte address
Security Register 3: A23-16 = 00h; A15-8 = 30h; A7-0 = byte address

15 General Purpose Input Output (GPIO)

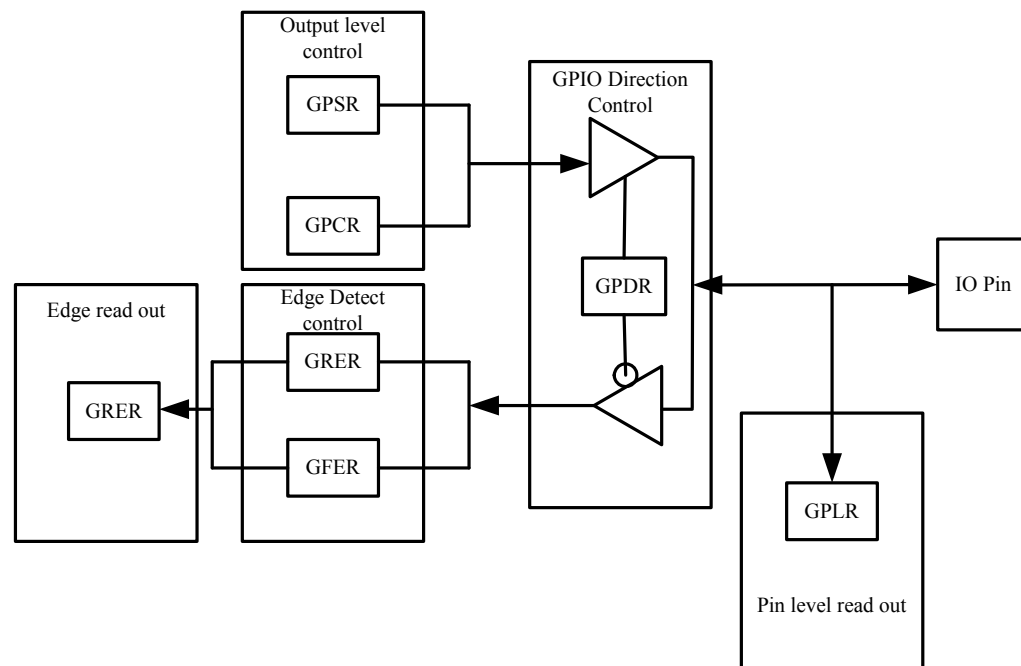
15.1 Overview

The GPIO unit provides as many as 63 GPIO pins.. All ports are brought out of the device using alternate function multiplexing. The GPIO function can be multiplexed on a multi-function I/O pin by selecting the GPIO alternate function in the pad configuration registers and configuring the GPIO internal registers. The GPIO registers are accessed through the APB interface.

15.2 GPIO Block Diagram

Figure 46 is a general-purpose I/O block diagram.

Figure 46: General Purpose I/O Block Diagram



15.3 GPIO Function Description

15.3.1 GPIO Ports

The GPIO pins are mapped to three port groups **GPIO_PORT0**, **GPIO_PORT1** and **GPIO_PORT2** of 32, 32 and 16 pins, respectively. Individual GPIO pins within a port are numbered from 0 to 31 according to their bit positions within the GPIO registers.

15.3.2 I/O Control

All the GPIO pins are inputs by default. The direction of the GPIO pins is configured by programming the GPIO Pin Direction Register - GPDR (0, 1, and 2) registers or the GPIO Set Direction Register-GSDR and GPIO Clear Direction Register-GCDR (0, 1 and 2) registers through the APB interface. Marvell recommends that, when the system is in low power mode, input enable to the GPIOs is disabled by setting the di_en (bit[3]) bit of the corresponding GPIO Configuration register in the Pinmux. Refer to the Appendix for Pinmux register details.

When configured as an input, GPIO can read the data on the external I/O pads and also serve as an interrupt. Interrupts are generated when the GPIO Rising Edge detect enable -GRER (0, 1 and 2) or GPIO Falling edge detect enable-GFER (0, 1 and 2) registers are configured. Rising and falling edges are detected using APB Clock Synchronized GPIO inputs. The status of edge detection can be read through the GPIO edge detect status-GEDR (0, 1 and 2) registers. The edge detect status is used to generate a combined interrupt from the GPIO block. Specific GPIO interrupts can also be masked by programming the APMASK register.

When configured as an output, the GPSR and GPCR (0, 1, and 2) registers are programmed to define the GPIO output port status.

The value of each GPIO port can be read through the GPIO Pin Level register-GPLR (read only) when the GPIO is configured as an input or output. This register can be read at any time to confirm the port state for the input configuration.

15.3.3 GPIO Interrupt

The GPIOs can be programmed to accept external signals as interrupt sources on any bit of the signal. The type of interrupt is programmable with either a rising or falling edge. When the GPIO is configured as an output, it requires a few GPIO clock cycles for the value to be updated in the GPLR register. The number of cycles required for updating the value depends on the CORE and GPIO clock frequencies and is specified as follows:

1. Core runs at 32M and GPIO runs at 32M: After writing to GPSR, correct GPLR value is reflected in three GPIO clock cycles
2. Core runs at 200M and GPIO runs at 50M: After writing to GPSR, correct GPLR value is reflected in five GPIO clock cycles

When the GPIO is configured as an input, this register is updated with the current level of the GPIO input after 12 cycles of the 200 MHz clock in the system.

The interrupts can be masked by programming the APMASK register. The interrupt status can be read before masking (called raw status) and after masking. A single combined interrupt is generated as output from the GPIO. All individual edges detected (as recorded in the GEDR registers) have to be masked, to mask the interrupt output. If the pin direction register is reprogrammed to output, then any pending interrupts are not lost. However, no new interrupts are generated.

When an edge is detected on a port that matches the type of edge programmed in the GRER and/or GFER registers, the corresponding status bit is set in GEDR registers. GEDR register value is updated with the current edge-detect status value after 12 cycles of the 200 MHz clock in the system from the occurrence of the edge.

15.3.4 External Interrupts

The Marvell 88MC200 microcontroller implements as many as 62 external interrupt sources. Since these external interrupt sources directly connect to the Cortex M3 NVIC module, an external interrupt pin can be used simultaneously by a peripheral device. All the external interrupts are active high. The external interrupt mapping table for peripheral interrupts and interrupts from the GPIO can be found in Chapter 6 Memory Subsystem, Interrupts and AHB Bus Fabric.

15.4 GPIO Register

There are a total of 42 registers in the GPIO register block. For each 32 bits forming a GPIO port there is a set of 14 registers. For up to 63 GPIOs in the 88MC200 system, three GPIO_ports are defined. Thus, there are three instances of each of the 14 registers. A complete description of the GPIO Registers is found in Appendix A.



88MC200 Microcontroller Datasheet

16 Advanced Encryption Standard (AES)

16.1 Features

The AES engine provides fast and energy efficient hardware encryption and decryption service for 88MC200 microcontroller. The main features of the 88MC200 AES engine are:

- Supports as many as six block cipher modes: ECB, CBC, CTR, CCM*, MMO, and Bypass
- Supports 128-, 192-, and 256-bit keys
- Efficient CPU/DMA access support
- Interrupt on finished AES operation, input FIFO full and output FIFO empty
- Error indications for each block cipher mode
- Separate 4*32-bit input and output FIFO

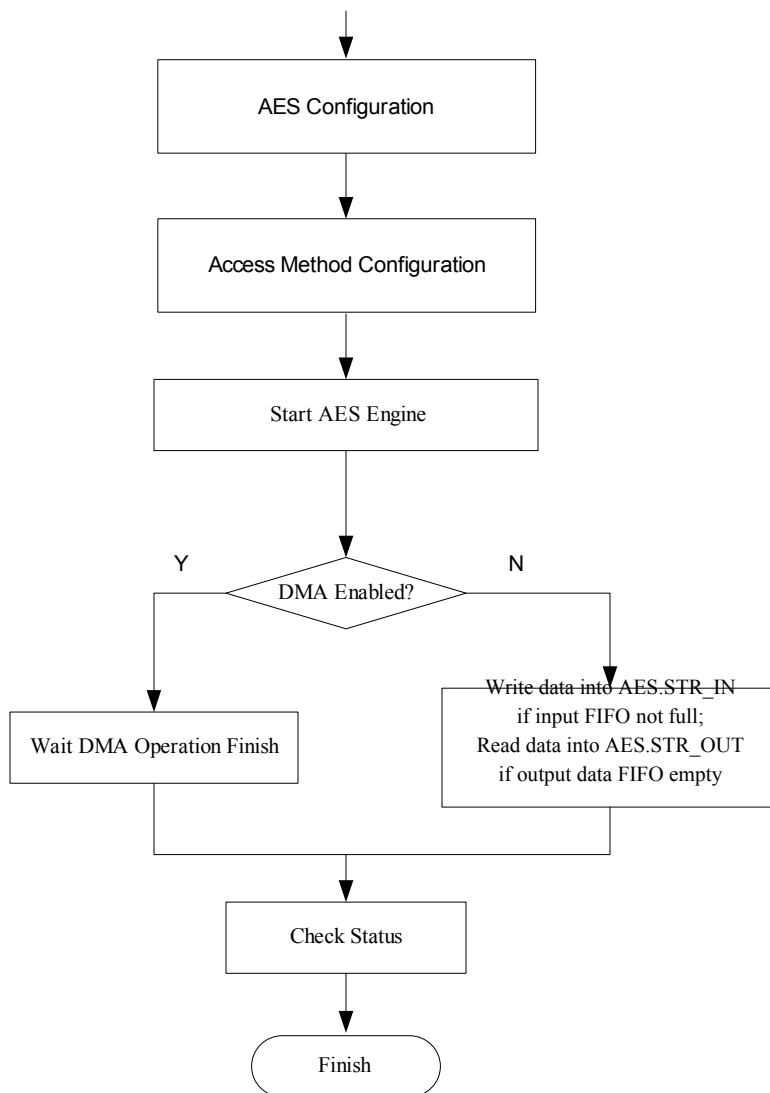
16.2 Functional Description

The AES module implements ECB, CBC, CTR, CCM*, MMO, and Bypass block cipher modes by efficient hardware.

16.2.1 AES Operational Flow

See [Figure 47](#) for the AES operational flow.

Figure 47: AES Operational Flow



16.2.2 AES Configuration

Ensure correct configuration before starting the AES engine by following these steps:

- Set AES engine to encrypt or decrypt by clearing/setting AES.CTRL1.DECRYPT
- Configure AES block cipher mode by setting AES.CTRL1.MODE, 0 for ECB mode, 1 for CBC mode, 2 for CTR mode, 5 for CCM* mode, 6 for MMO mode and 7 for BYPASS mode
- Configure AES key size. AES engine supports three types of key size: 128-, 192-, and 256- bit. Configure AES key size parameter by setting AES.CTRL1.KEY_SIZE

- Fill the key according to key size. AES engine contains eight 32-bit key registers defined as AES.KEY0, AES.KEY1, AES.KEY2, AES.KEY3, AES.KEY4, AES.KEY5, AES.KEY6 and AES.KEY7. When key size is set to 128-bit, then AES.KEY7/6/5/4 is used. When key size is set to 192-bit, then AES.KEY7/6/5/4/3/2 is used. When key size is set to 256-bit, then AES.KEY7/6/5/4/3/2/1/0 is used. However, MMO does not support 192- and 256- bit key size, and key size is ignored in Bypass mode.
- For all modes except CCM* mode, set input data size by setting AES.MSTR_LEN. For CCM* mode, set associate data size by setting AES.ASTR_LEN, set message data size by setting AES.MSTR_LEN.
- If AES block cipher mode is CTR mode, set CTR mode's counter modular by setting AES.CTRL1.CTR_MODE.
- For CCM* encryption or MMO mode, If MIC/HASH is needed, set AES.CTRL1.OUT_MIC bit to 1 to append MIC/HASH at the end of output stream. If only MIC/HASH is needed, we can block the encrypted data into output FIFO (set AES.CTRL1.OUT_MSG bit to 1), and get MIC/HASH from AES.OV3/2/1/0.
- For CCM* mode, set AES.CTRL1.OUT_HDR bit to 1 to output B0 at the beginning of the output stream if it is necessary.
- Fill with initial value according to AES block cipher mode. AES engine contains four 32-bit initial vector registers naming by AES.IV0, AES.IV1, AES.IV2, AES.IV3. For ECB/MMO/BYPASS mode, there are no initial vectors needed to be configured. For CTR mode, set AES.IV0= initial counter, AES.IV1= Nonce[31:0], AES.IV2=Nonce[63:32], AES.IV3=Nonce[95:64]. For CCM* mode, set AES.IV0=Nonce [31:0], AES.IV1=Nonce[63:32], AES.IV2=Nonce[85:64], AES.IV3=15-Nonce Bytes.



Note

For Bypass mode, the AES engine ignores input data; it passes it along unchanged to the output.

16.2.3 Data Access Method

The AES module contains separate 4*32-bit input and output FIFO. The input and output FIFO can be accessed by DMA or CPU.

When setting AES.CTRL1.IO_SRC bit to 1 and AES.CTRL1.DMA_EN bit to 1, the input and output FIFO are accessed by DMA. Two channels are required: one for input data and the other for output data. Before starting AES engine, the DMA controller must be configured, the transfer size is the input data length and output data length. The AES operation finishes as the DMA operation finishes.

When setting AES.CTRL1.IO_SRC bit to 0 and AES.CTRL1.DMA_EN bit to 0, the input and output FIFO are accessed by CPU. Then it writes data into AES.STR_IN if the input FIFO is not full; read data from AES.STR_OUT if output FIFO is not empty. The AES operation finishes as the transfer data size reaches input and output data size.

16.2.4 Starting the AES Engine

Clear AES input and output FIFO and reset AES core before starting the AES engine. The AES input and output FIFO can be cleared by setting the AES.CTRL1.IF_CLR bit and AES.CTRL1.OF_CLR bit to 1. The AES core can be reset by setting and then clearing AES.CTRL2.CORE_RESET.

16.2.5 Interrupt Request

There are three interrupts for the AES engine: input FIFO full interrupt, output FIFO empty interrupt, and AES operation done interrupt. Each interrupt can be masked or cleared by setting AES.IMR/AES.IC registers.

16.2.6 Partial Code Support

The AES engine can automatically pad the input data when the input data length is not a multiple of 128 bits. The AES module supports the following padding scheme for different AES block cipher modes, shown as [Table 44](#).

Table 44: Padding Scheme

Mode	Description
CCM*	Automatically padding 0 for both A string and M string
MMO	Automatically padding "100...00"+2 bytes length information
CBC	Cipher stealing is performed to partial codeword
CTR	Partial code word don't affect the operation
ECB	Check partial case, assert error when partial cases detected

16.2.7 Error Status Check

Register AES.STATUS.STATUS records the error status for the AES engine when the AES operation has finished. [Table 45](#) shows the error status for different AES block cipher modes.

Table 45: Error Status for Different AES Block Cipher Modes

Mode	Status[2]	Status[1]	Status[0]
ECB	N/A	Data is not multiple of 16 bytes	Input data size less than 16 bytes
CBC	N/A	N/A	
CTR	N/A	N/A	
CCM*	MIC mismatch during decryption	N/A	N/A
MMO	N/A	Data is more than $2^{13}-1$ bytes	N/A
By-pass	N/A	N/A	N/A

16.2.8 Output Vector

The output vector provides some useful information, such as the last cipher block in CBC mode, last counter in CTR mode, MIC value in CCM* mode and HASH value in MMO mode. Register AES.OV3/2/1/0 records useful information for different AES block cipher modes. [Table 46](#) shows the recorded information in AES output vector for different AES block cipher mode.

Table 46: AES Output Vector

Block Cipher Mode	Output Vector
ECB	N/A
CBC	Last cipher block AES.OV0 = cipher[31:0] AES.OV1 = cipher[63:32] AES.OV2 = cipher[95:64] AES.OV3 = cipher[127:96]
CTR	Last counter AES.OV0 = counter[31:0] AES.OV1 = counter[63:32] AES.OV2 = counter[95:64] AES.OV3 = counter[127:96]
CCM*	Encryption: MIC value. If MIC is less than 32 byte, always MSB byte is used. Example: 8 byte MIC AES.OV2 = MIC[31:0] AES.OV3 = MIC[63:32]
MMO	HASH value AES.OV0 = HASH[31:0] AES.OV1 = HASH[63:32] AES.OV2 = HASH[95:64] AES.OV3 = HASH[127:96]
By-pass	N/A

16.2.9 AES Operation Pseudo Code

```

AES_Config_Type aesConfig

aesConfig.mode <- AES_MODE_CBC
aesConfig.encDecSel <- AES_MODE_ENCRYPTION
aesConfig.keySize <- keysize
aesConfig.mStrLen <- length

for i=1 to keysize do
  aesConfig.key[i] = key[i]

for i=1 to 4 do
  aesConfig.initVect[i] = vector[i]

while j < length or k < length do
  if AES input fifo not full do
    feed the data plain_text[j]
    j++

```

```
if AES output fifo not empty do  
    read the encryption data  
k++
```

16.3 References for AES Standard

- [1] www.nist.gov/aes (AES development, historical site)
- [2] <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf> (AES standard and test vectors)
- [3] http://csrc.nist.gov/groups/ST/toolkit/block_ciphers.html (overview over block ciphers, key handling and test vectors)
- [4] Schneier, B.: Applied Cryptography, 2nd ed., Wiley 1996
- [5] Wobst, R.: Cryptology Unlocked, Wiley 2007
- [6] http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation
- [7] http://en.wikipedia.org/wiki/CCM_mode

16.4 Register Description

A detailed description of the AES Registers is located in Appendix A.

17 Cyclic Redundancy Check (CRC)

17.1 Overview

A cyclic redundancy check (CRC) or polynomial code checksum is a hash function designed to detect data integrity. The CRC unit calculates a short, fixed-length binary sequence, known as the CRC code. For each block of data, CRC code and original data are sent or stored together. When a block of data is used, the same CRC calculation is processed. If the new CRC does not match the one pre-calculated earlier in the block of data, then the block contains a data error and the device may take corrective action such as resending or requesting the block again; otherwise the data is assumed to be error free (though, with some small probability, it may contain undetected errors; this is the fundamental nature of error-checking).

17.2 Features

A standard AHB slave interface is used to configure the module, receive the bit stream, and output the CRC result.

- Supports 32-bit parallel bit stream input, and supports up to 32-bit CRC output
- Supports up to 2^{32} (4294967296) byte length to calculate CRC
- Supports the following CRC standards
 - CRC-16-CCITT, the polynomial is $x^{16}+x^{12}+x^5+1$
 - CRC-16-IBM, the polynomial is $x^{16}+x^{15}+x^2+1$
 - CRC-16-T10-DIF, the polynomial is $x^{16}+x^{15}+x^{11}+x^9+x^8+x^7+x^5+x^4+x^2+x+1$
 - CRC-32-IEEE 802.3, the polynomial is $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$
 - CRC-16-DNP, the polynomial is $x^{16}+x^{13}+x^{12}+x^{11}+x^{10}+x^8+x^6+x^5+x^2+1$

17.3 CRC Operation Flow

1. Disable CRC (CRC.CTRL.ENABLE to 0).
2. Disable the interrupt (set CRC.IMR.MASK to 1).
3. Clear all the interrupts (set CRC_ICR.CLEAR to 1).
4. Configure the stream length (set register CRC.STREAM_LEN_M1).
5. Configure CRC mode (set bit CRC.CTRL.MODE).
6. Enable the interrupt (set CRC.IMR.MASK to 0).
7. Enable CRC (set CRC.CTRL.ENABLE to 1).
8. Write stream in and waiting for interrupt to occur. (If interrupt occurred, go to 9)
9. Get CRC calculation result (Read register CRC.RESULT).
10. CRC operation complete.



Note

The CRC input stream registers accepts a word (32 bit) at a time. If the input data is not 4 bytes aligned, pad zeros at the start of the data stream. For example, if the data stream consists of 5 bytes starting from the lower address: 0xA1, 0xA2, 0xA3, 0xA4, 0xA5.

The following two words should be written to the stream input register:

- 0xA1000000
- 0xA5A4A3A2

The CRC result bit order is:

16 bit CRC: $x_0 \sim x_{15}$ [msb->lsb]

32 bit CRC: $x_0 \sim x_{31}$ [msb->lsb]

17.4 Register Descriptions

A detailed description of the CRC registers is located in Appendix A.

18 Universal Asynchronous Receiver Transmitter (UART)

18.1 Overview

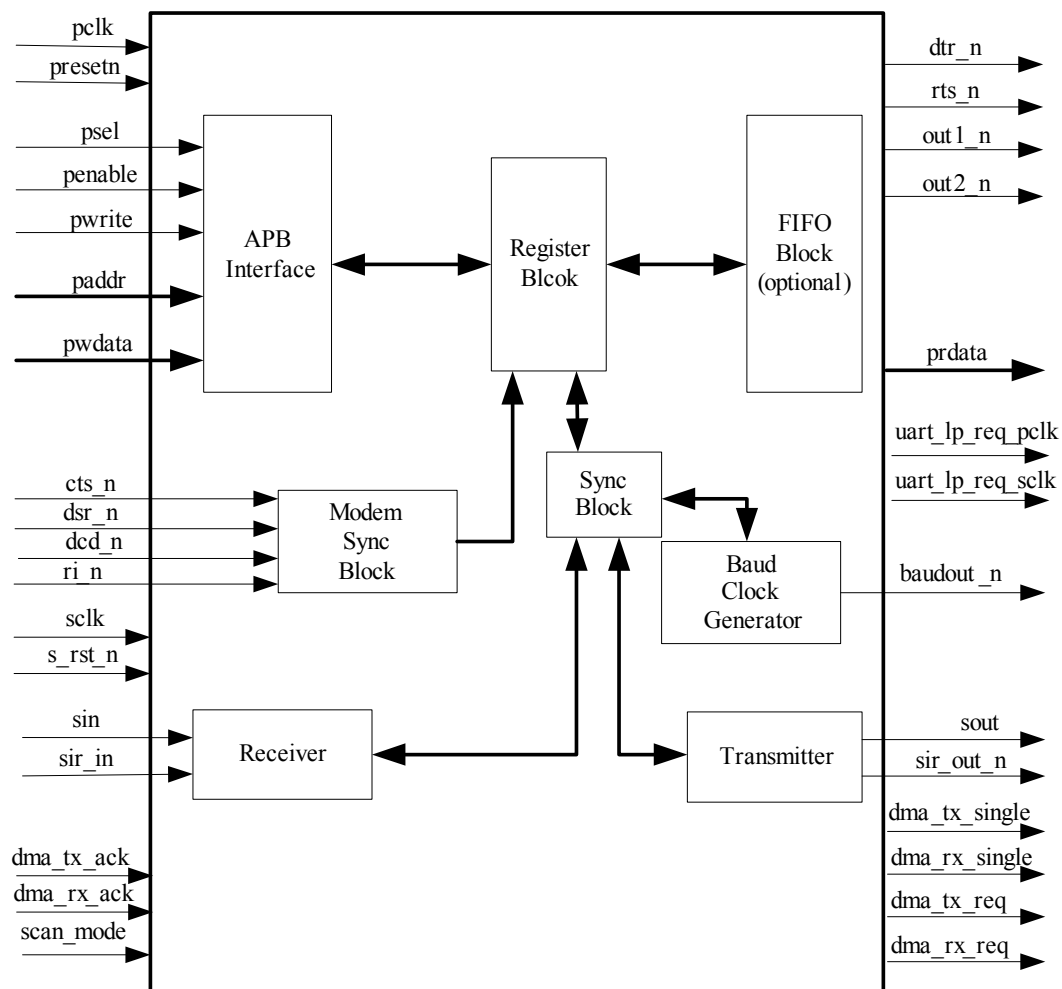
The 88MC200 device integrates four Universal Asynchronous Receiver Transmitter (UART) modules with the following features:

- Programmable FIFO access mode for 16 x 8 bits transmit and receive FIFO
- All four UARTs have DMA request capability
- Auto flow control support
- Programmable data format:
 - 5–8 data bits plus parity
 - Odd, even, no parity
 - One, one-and-a-half, or two stop bits
- Six interrupt type with flags:
 - Receiver line status
 - Receiver Data Available
 - Character Timeout (in FIFO mode only)
 - Transmitter Holding Register Empty or FIFO at/below threshold (Programmable THRE interrupt mode enable)
 - Modem Status
 - Busy Detect Indication
- Seven additional shadow registers to be used to reduce the software overhead
- Additional FIFO status registers
- IrDA 1.0 SIR mode supports up to 115200 baud rate and pulse duration (width) of 3/16x bit
- IrDA 1.0 SIR low-power reception capabilities

18.2 Block Diagram

[Figure 48](#) is a block diagram of the UART.

Figure 48: Block Diagram



18.3 Function Description

18.3.1 External Signal Descriptions

[Table 47](#) describes the UART Interface bus signals.

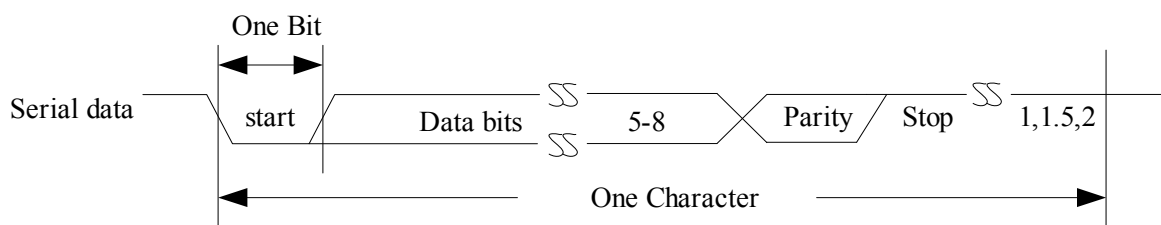
Table 47: Serial Interface Signal Descriptions

Signal Name	Description
TXD	Transmit data output
RXD	Receive data input
CTS	Clear To Send input
RTS	Request To Send output
DSRn	Data Set Ready Modem Status input
DCDn	Data Carrier Detect Modem Status input
Rin	Ring Indicator Modem Status input
DTRn	Modem Control Data Terminal Ready output
SIR_OUT	IrDA SIR output
SIR_IN	IrDA SIR input

18.3.2 Protocol

Figure 49 shows the data structure of UART serial protocol (RS232).

Figure 49: Serial Data Structure



The structure of serial data accompanied by start bit and stop bits(1, 1.5 or 2) is referred to as a character, as shown in Figure 49. The individual bits of the data word(5-8 bits) are sent after the start bit, starting with the least significant bit (LSB).

An additional parity bit may be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure to provide the UART with the ability to perform simple error checking on the received data.

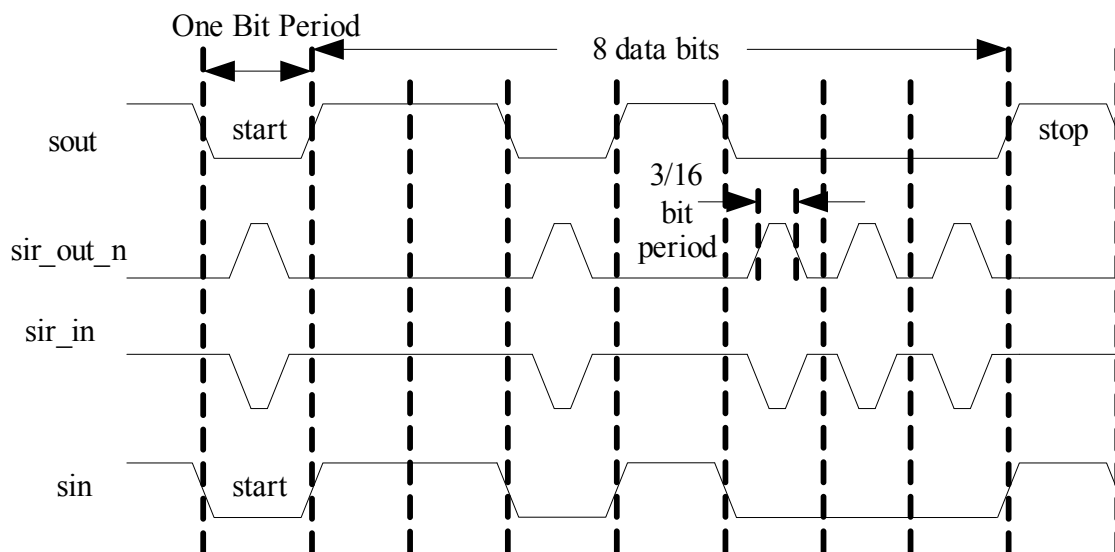
All the bits are transmitted for exactly the same time duration, which equals 16 baud clocks. The UART_LCR register is used to control the serial character characteristics.

18.3.3 SIR Protocol

Serial Infrared mode (IrDA 1.0) supports up to 115.2K baud rate bi-directional communication with remote devices, which can be enabled by setting UART_MCR [6] bit. The data format (sir_out_n and sir_in) is similar to the standard serial data format (sout and sin). Each character begins with a start bit, followed by 8 data bits, and ends with one stop bit. No parity information can be supplied and

only one stop bit is used while in this mode. Refer to [Figure 50](#).

Figure 50: SIR Data Format



When SIR mode is enabled and active by setting the MCR[6] bit, serial data is transmitted and received on the sir_out_n and sir_in ports, respectively. Trying to adjust the number of data bits sent or enable parity with the Line Control Register (LCR) has no effect.

Transmitting a single infrared pulse signals a logic zero, while a logic one is represented by not sending a pulse. The width of each pulse is 3/16ths of a normal serial bit time. Thus, each new character begins with an infrared pulse for the start bit. However, received data is inverted from transmitted data due to the infrared pulses energizing the photo transistor base of the IrDA receiver, pulling its output low. This inverted transistor output is then fed to uart sir_in, which then has correct UART polarity. [Figure 50](#) shows the timing diagram for the IrDA SIR data format in comparison to the standard serial format.

The UART can be configured to support a low-power reception mode. When the UART is configured in this mode, the reception of SIR pulses of 1.41 microseconds (minimum pulse duration) is possible, as well as nominal 3/16 of a normal serial bit time. Using this low-power reception mode requires programming the Low Power Divisor Latch (LPDLL/LPDH) registers. For all sclk frequencies greater than or equal to 7.37MHz (and obey the requirements of the Low Power Divisor Latch registers), pulses of 1.41µs are detectable. However there are several values of sclk that do not allow the detection of such a narrow pulse.

18.3.4 FIFO Access

Each UART has 16 x 8 bits receive and transmit FIFO respectively. FIFO Control Register (UART_FCR) controls the FIFO access mode. If disable FIFO (set UART_FCR [0] = 0), only a single receive data byte and transmit data byte can be stored at a time in the UART_RBP and UART_THR. Trigger level can be selected for receive and transmit FIFO to generate an interrupt. For detailed descriptions, refer to FIFO Control Register (UART_FCR).

18.3.5 Calculating Baud Rates

The UART has two system clocks (pclk and sclk). The sclk clock is used as the functional clock source. There are two programmable fractional dividers to generate the required uart sclk in the PMU module. By changing the nominator and denominator fields of UART_FAST_CLK_DIV or UART_SLOW_CLK_DIV register, users can obtain the more precise baud rate.

$$Sclk = SPLL\_Output * denominator / nominator$$

Divisor Latch Registers (UART_DLH and UART_DLL) contain the baud rate divisor for the UART. When the DLAB bit (UART_LCR [7]) is set and the UART is not busy (UART_USR [0] is zero), the Divisor Latch Registers are writeable.

The baud-rate calculation is:

$$Baud\ rate = (sclk\ frequency) / (16 * Divisor)$$

18.3.6 Interrupts

The assertion of the UART interrupt occurs whenever one of the several prioritized interrupt types is enabled and active. The following interrupt types can be enabled with the UART_IER register:

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE interrupt mode which is indicated by the UART register UART_LSR [5])
- Modem Status
- Busy Detect Indication
- When an interrupt occurs, the master accesses the IIR register to determine which interrupt occurred. For detailed descriptions, refer to the Interrupt Enable Register (IER) and Interrupt Identification Register (IIR).

18.3.7 DMA Support

The UART uses two DMA channels, one for the transmit data and one for the receive data. The UART supports DMA signaling with handshaking signals. Four output request signals indicate when data is ready to be read or when the transmit FIFO is empty. Two input acknowledge signals indicate when the DMA operation is completed.

The request signal is asserted under the following conditions in transmit mode:

- The transmitter Holding Register is empty in non-FIFO mode.
- The transmitter FIFO is empty in FIFO mode when Programmable THRE interrupt mode disabled.
- The transmitter FIFO is at or below the programmed threshold when Programmable THRE interrupt mode enabled.

The request signal is asserted under the following conditions in receive mode:

- There is a single character available in the Receive Buffer Register in non-FIFO mode.
- The Receive FIFO is at or above the programmed trigger level in FIFO mode.
- The de-assertion of the DMA transmit and receive request is respectively controlled by the assertion of the DMA transmit and receive acknowledge.

When UART auto flow control mode is enabled, the deadlock condition can be avoided if:

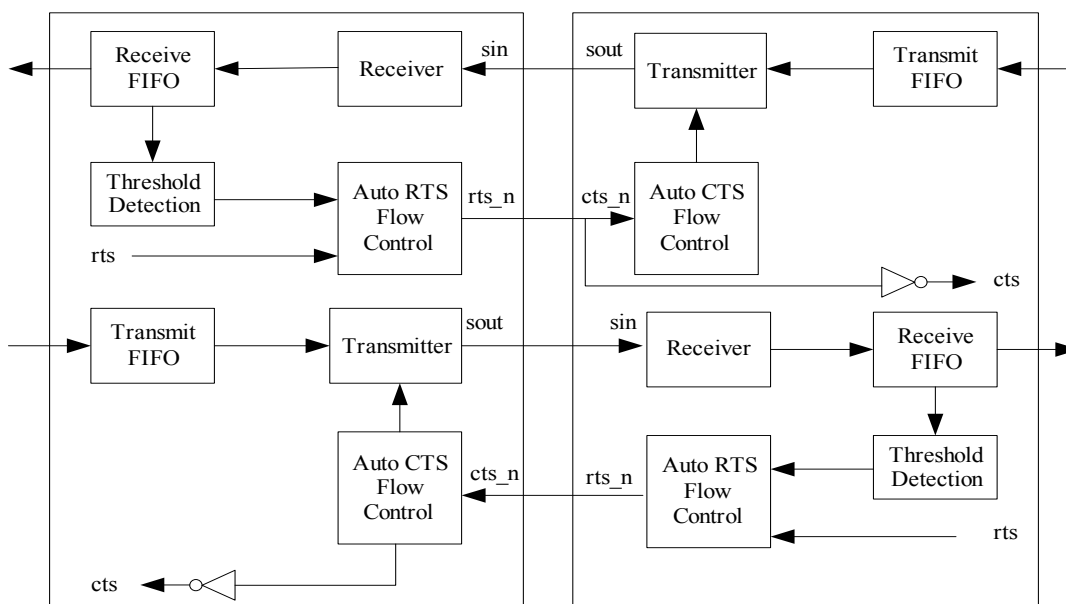
- UART_rx FIFO threshold should be set to a value equal or smaller than DMA burst transaction size.
- The DMA block size is set to a value smaller than twice the DMA burst transaction size.

The DMA controller is responsible for the data flow, which is controlled by the programmed burst transaction lengths.

18.3.8 Auto Flow Control

Auto Flow Control can be enabled with the Modem Control Register (UART_MCR [5]) when FIFOs are enabled. [Figure 51](#) shows a block diagram of the Auto Flow Control.

Figure 51: Auto Flow Control Block Diagram



Auto RTS becomes active under the following conditions:

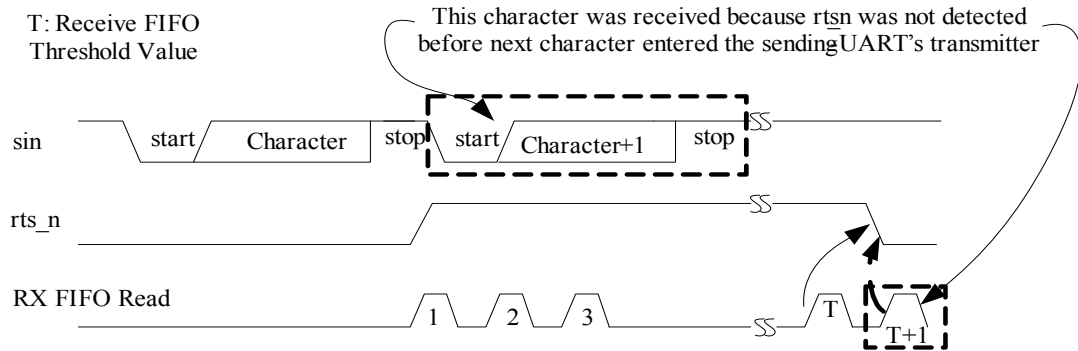
- Auto request to send is enabled (UART_MCR[1] bit and UART_MCR[5] bit are both set)
- FIFOs are enabled (UART_FCR[0] bit is set)
- SIR mode is disabled (UART_MCR[6] bit is not set)

When Auto RTS is enabled (active), the rts_n output is forced inactive (high) when the receiver FIFO level reaches the threshold set by UART_FCR [7:6]. When rts_n is connected to the cts_n input of another UART device, the other UART stops sending serial data until rts_n is active again. Once the receiver FIFO becomes completely empty by reading the Receiver Buffer Register (RBR), rts_n becomes active (low), signaling the other UART to continue sending data.

When Auto RTS is not implemented or disabled, rts_n is controlled solely by UART_MCR [1].

[Figure 52](#) shows a timing diagram of Auto RTS operation.

Figure 52: Auto RTS Timing



Auto CTS becomes active under the following conditions:

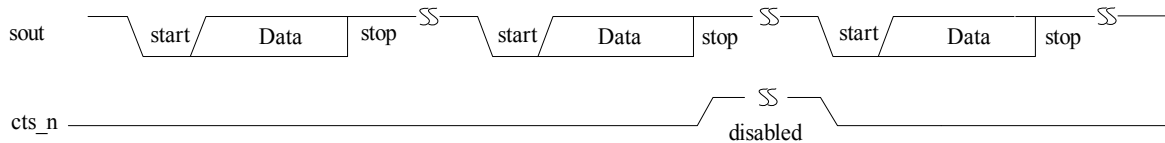
- Auto flow control is enabled (UART_MCR [5] bit is set).
- FIFOs are enabled (UART_FCR [0] bit is set).
- SIR mode is disabled (UART_MCR [6] bit is not set)

When Auto CTS is enabled (active), the UART transmitter is disabled whenever the **cts_n** input becomes inactive (high). This prevents overflowing the FIFO of the receiving UART. When the **cts_n** input becomes active (low) again, transmission resumes.

If the **cts_n** input is not inactivated before the middle of the last stop bit, another character is transmitted before the transmitter is disabled. While the transmitter is disabled, the transmitter FIFO can still be written to, and even overflowed. Therefore, software can poll the Transmitter FIFO status before each write.

When Auto CTS is not implemented or disabled, the transmitter is unaffected by **cts_n**. [Figure 53](#) shows a timing diagram of Auto CTS operation.

Figure 53: Auto CTS Timing



If the FIFOs are disabled (UART_FCR [0] bit is not set) or the UART is in SIR mode (UART_MCR [6] bit is set), auto flow control is also disabled, even if everything else is selected.

18.4 Register Descriptions

A complete description of the UART Registers is located in Appendix A.

19 Inter-Integrated Circuit (I²C)

19.1 Overview

The I²C bus interface complies with the common I²C protocol and can operate in standard mode (with data rates up to 100 Kb/s), fast mode (with data rates up to 400 Kb/s) and high-speed mode (with data rates up to 2Mb/s). Additionally, high-speed mode devices and fast mode devices are downward compatible. It also supports DMA capability.

The 88MC200 microcontroller contains three I²C interfaces: I2C0, I2C1 and I2C2, all of which are identical in function.

19.2 Features

The I²C bus interface unit has the following features:

- Three I²C serial interfaces consisting of a serial data line (SDA) and serial clock (SCL)
- Three speeds:
 - Standard mode (up to 100 Kb/s)
 - Fast mode (up to 400 Kb/s)
 - High-speed mode (2Mb/s)
- Clock synchronization
- Master or Slave I²C operation, multi-master, multi-slave operation, and arbitration support
- 7- or 10-bit addressing and General Call
- 7- or 10-bit combined format transfers
- Bulk transmit mode in slave
- 16 * 32 bits deep transmit and receive buffers, respectively
- Interrupt operation
- DMA function support

19.3 Signal Descriptions

I²C uses a two-pin interface as shown in [Table 48](#).

Table 48: I²C Signal Descriptions

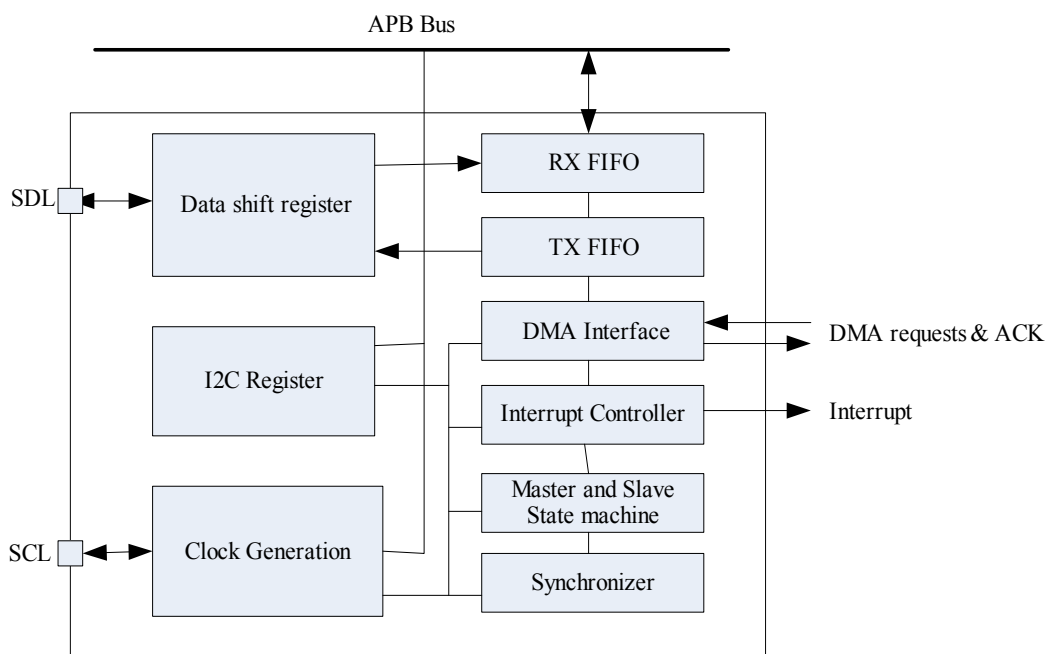
Signal Name	Input/Output	Description
SDA	Bidirection	Data signal line (Serial Data)
SCL	Bidirection	Clock signal line (Serial Clock)

19.4 Operation

19.4.1 I²C Block Diagram

The I²C consists of an APB slave interface, a I²C interface, and FIFO logic to maintain coherency between the two interfaces. A simplified block diagram of the component is illustrated in [Figure 54](#).

Figure 54: I²C Block Diagram



19.4.2 I²C Bus Terminology

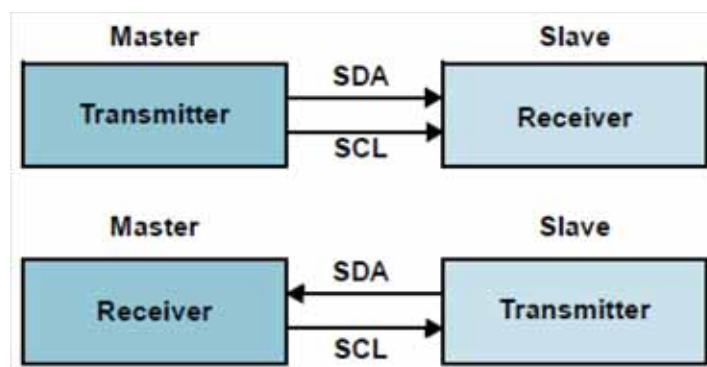
[Table 49](#) describes I²C bus terminology.

Table 49: I²C Bus Terminologies

I ² C Device	Definition
Transmitter	Sends data over the I ² C bus. A transmitter can either be a device that initiates the data transmission to the bus (a <i>master-transmitter</i>) or responds to a request from the master to send data to the bus (a <i>slave-transmitter</i>).
Receiver	Receives data over the I ² C bus. A receiver can either be a device that receives data on its own request (a <i>master-receiver</i>) or in response to a request from the master (a <i>slave-receiver</i>).

Master	The component that initiates a transfer (START command), generated the clock (SCL) signal and terminates the transfer (STOP command). A master can be either a transmitter or a receiver.
Slave	The device addressed by the master. A slave can be either a receiver or transmitter (see Figure 55).
Multi-master	The ability for more than one master to co-exist on the bus at the same time without collision or data loss.
Arbitration	The predefined procedure that authorizes only one master at a time to take control of the bus. Refer to "Multiple Master Arbitration" for more information.
Synchronization	The predefined procedure that synchronizes the clock signals provided by two or more masters. For more information about this feature, refer to "Clock Synchronization."
Bus Transfer Terminology The following terms are specific to data transfers that occur to/from the I²C bus.	
START (RESTART)	Data transfer begins with a START or RESTART condition. The level of the SDA data line changes from high to low, while the SCL clock line remains high. When this occurs, the bus becomes busy. Note: START and RESTART conditions are functionally identical.
STOP	Data transfer is terminated by a STOP condition that occurs when the level on the SDA data line passes from the low state to the high state, while the SCL clock line remains high. When the data transfer has been terminated, the bus is free or idle once again. The bus stays busy if a RESTART is generated instead of a STOP condition.

Figure 55: Master/Slave and Transmitter/Receiver Relationship



19.5 I²C Behavior

The I²C can be controlled via software to be either:

- An I²C master only, communicating with other I²C slaves; OR
- An I²C slave only, communicating with one more I²C masters.

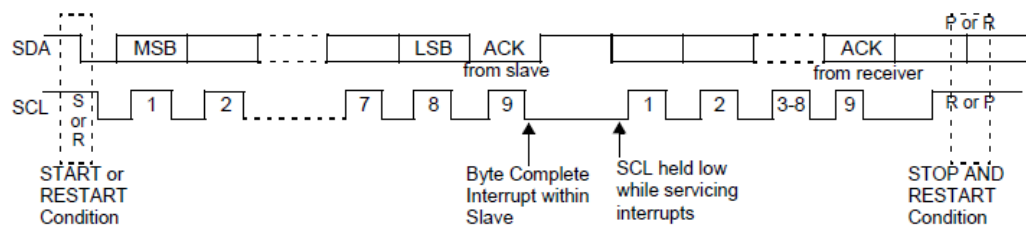
The master is responsible for generating the clock and controlling the transfer of data. The slave is responsible for either transmitting or receiving data to/from the master. The acknowledgement of

data is sent by the device that is receiving data, which can be either a master or a slave. The I²C protocol also allows multiple masters to reside on the I²C bus and uses an arbitration procedure to determine bus ownership.

Each slave has a unique address that is determined by the system designer. When a master wants to communicate with a slave, the master transmits a START/RESTART condition that is then followed by the slave's address and a control bit (R/W) to determine if the master wants to transmit data or receive data from the slave. The slave then sends an acknowledge (ACK) pulse after the address.

If the master (master-transmitter) is writing to the slave (slave-receiver), the receiver gets one byte of data. This transaction continues until the master terminates the transmission with a STOP condition. If the master is reading from a slave (master-receiver), the slave transmits (slave-transmitter) a byte of data to the master, and the master then acknowledges the transaction with the ACK pulse. This transaction continues until the master terminates the transmission by not acknowledging (NACK) the transaction after the last byte is received, and then the master issues a STOP condition or addresses another slave after issuing a RESTART condition. This behavior is illustrated in Figure 56.

Figure 56: Data Transfer on the I²C Bus



The I²C is a synchronous serial interface. The SDA line is a bidirectional signal and changes only while the SCL line is low, except for STOP, START, and RESTART conditions. The output drivers are open-drain or open-collector to perform wire-AND functions on the bus. The maximum number of devices on the bus is limited by only the maximum capacitance specification of 400 pF. Data is transmitted in byte packages.



Note

Placing data into the FIFO generates a START, and emptying the FIFO generates a STOP. For more information, refer to "START and STOP Generation".

19.5.1 START and STOP Generation

When operating as a I²C master, placing data into the Transmit FIFO causes the I²C to generate a START condition on the I²C bus. Allowing the Transmit FIFO to empty causes the I²C to generate a STOP condition on the I²C bus.

When operating as a slave, the I²C does not generate START and STOP conditions, as per the protocol. However, if a read request is made to the I²C, it holds the SCL line low until read data has been supplied to it. This action stalls the I²C bus until read data is provided to the slave I²C, or the I²C slave is disabled by writing a 0 to ENABLE in register I2C.ENABLE.

19.5.1.1 Combined Formats

The I²C I supports mixed read and write combined format transactions in both 7-bit and 10-bit addressing modes.

The I²C does not support mixed address and mixed address format—that is, a 7-bit address transaction followed by a 10-bit address transaction or vice versa—combined format transactions.

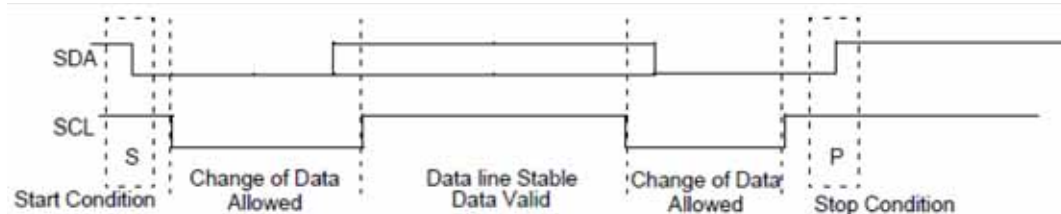
To initiate combined format transfers, set the register CON.RESTART_EN to 1. With this value set and operating as a master, when the I²C completes a I²C transfer, it checks the Transmit FIFO and executes the next transfer. If the direction of this transfer differs from the previous transfer, the combined format is used to issue the transfer. If the Transmit FIFO is empty when the current I²C transfer completes, a STOP is issued and the next transfer is issued following a START condition.

19.5.2 I²C Protocols

19.5.2.1 START and STOP Conditions

When the bus is idle, both the SCL and SDA signals are pulled high through external pullup resistors on the bus. When the master wants to start a transmission on the bus, the master issues a START condition. This is defined to be a high-to-low transition of the SDA signal while SCL is 1. When the master must terminate the transmission, it issues a STOP condition, which is defined to be a low-to-high transition of the SDA line while SCL is 1. Figure 57 shows the timing of the START and STOP conditions. When data is being transmitted on the bus, the SDA line must be stable when SCL is 1.

Figure 57: START and STOP Condition



Note

The signal transitions for the START/STOP conditions, as depicted in Figure 19-4, reflect those observed at the output signals of the Master driving the I²C bus. Use caution when observing the SDA/SCL signals at the input signals of the Slave(s), because unequal line delays may result in an incorrect SDA/SCL timing relationship.

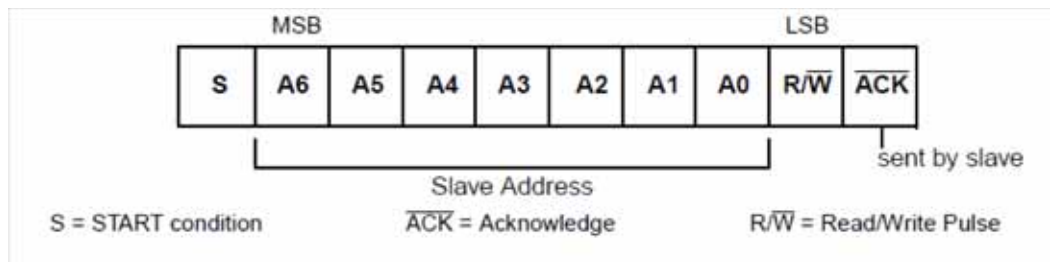
19.5.2.2 Addressing Slave Protocol

There are two address formats: the 7-bit and 10-bit address formats.

7-Bit Address Format

During the 7-bit address format, the first seven bits (bits 7:1) of the first byte set the slave address and the LSB bit (bit 0) is the R/W bit as shown in Figure 58. When bit 0 (R/W) is set to 0, the master writes to the slave. When bit 0 (R/W) is set to 1, the master reads from the slave.

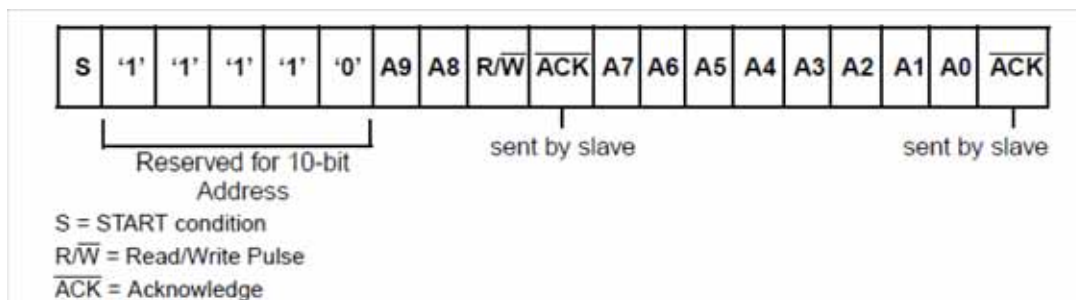
Figure 58: 7-Bit Address Format



10-Bit Address Format

During 10-bit addressing, two bytes are transferred to set the 10-bit address. The transfer of the first byte contains the following bit definition. The first five bits (bits 7:3) notify the slaves that this is a 10-bit transfer followed by the next two bits (bits 2:1), which set the slaves address bits 9:8, and the LSB bit (bit 0) is the /W bit. The second byte transferred sets bits 7:0 of the slave address. [Figure 59](#) shows the 10-bit address format.

Figure 59: 10-Bit Address Format



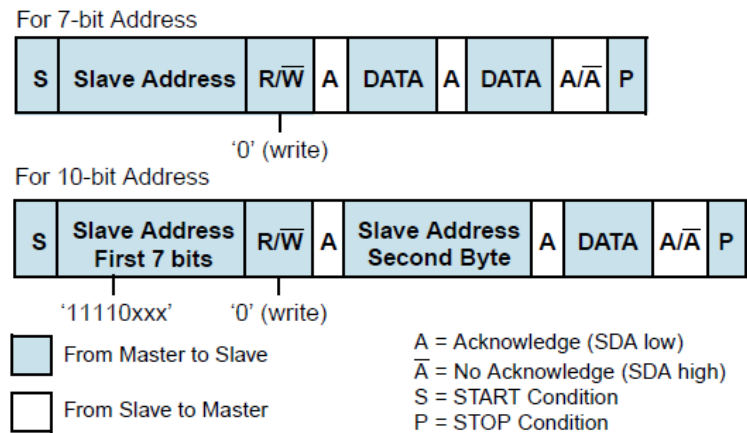
19.5.2.3 Transmitting and Receiving Protocol

The master can initiate data transmission and reception to/from the bus, acting as either a master-transmitter or master-receiver. A slave responds to requests from the master to either transmit data or receive data to/from the bus, acting as either a slave-transmitter or slave-receiver, respectively.

Master-Transmitter and Slave-Receiver

All data is transmitted in byte format, with no limit on the number of bytes transferred per data transfer. After the master sends the address and R/W bit or the master transmits a byte of data to the slave, the slave-receiver must respond with the acknowledge signal (ACK). When a slave-receiver does not respond with an ACK pulse, the master aborts the transfer by issuing a STOP condition. The slave must leave the SDA line high so that the master can abort the transfer.

Figure 60: Master-Transmitter Protocol

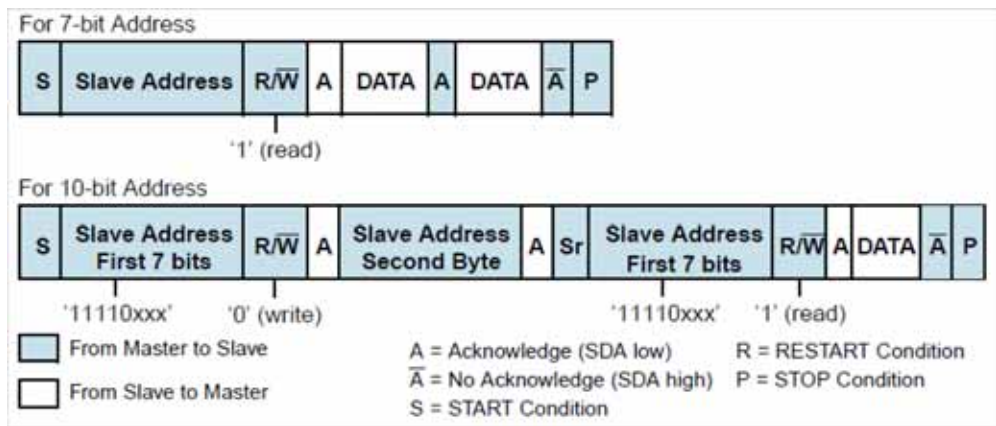


Master-Receiver and Slave-Transmitter

If the master is receiving data as shown in Figure 61, then the master responds to the slave-transmitter with an acknowledge pulse after a byte of data has been received, except for the last byte. This method is how the master-receiver notifies the slave-transmitter that this is the last byte. The slave-transmitter relinquishes the SDA line after detecting the No Acknowledge (NACK) so that the master can issue a STOP condition.

When a master refuses to relinquish the bus with a STOP condition, the master can issue a RESTART condition. This is identical to a START condition except it occurs after the ACK pulse. Operating in master mode, the I²C can then communicate with the same slave using a transfer of a different direction. For a description of the combined format transactions that the I²C supports, refer to “Combined Formats.”

Figure 61: Master-Receive Protocol



19.5.3 Multiple Master Arbitration

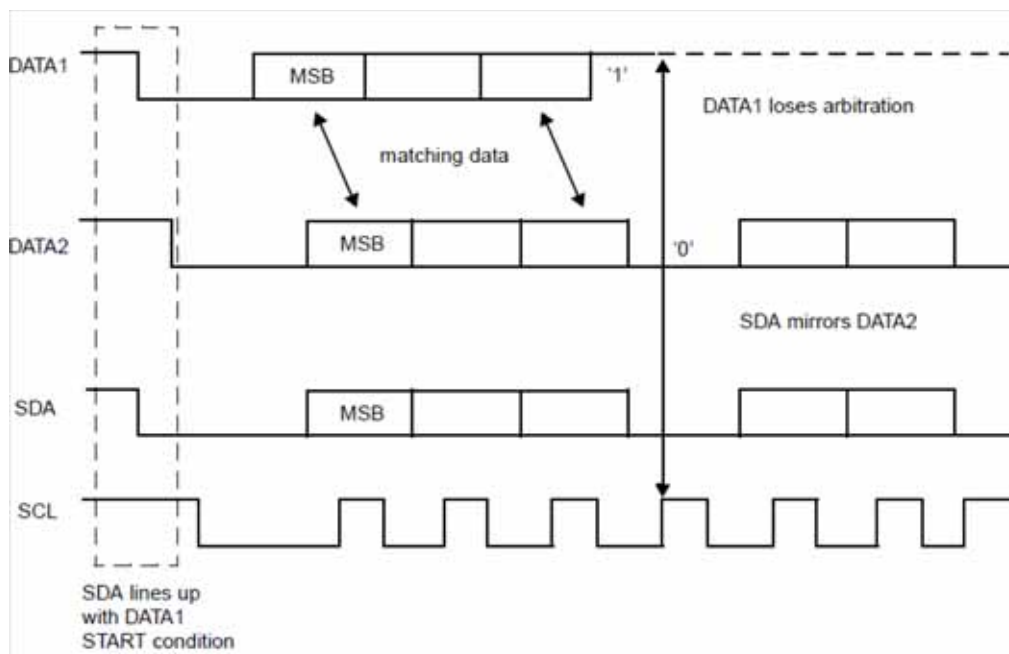
The I²C bus protocol allows multiple masters to reside on the same bus. If there are two masters on the same I²C bus, there is an arbitration procedure if both try to take control of the bus at the same time by generating a START condition at the same time. Once a master (for example, a microcontroller) has control of the bus, no other master can take control until the first master sends a STOP condition and places the bus in an idle state.

Arbitration occurs on the SDA line, while the SCL line is 1. The master, which transmits a 1 while the other master transmits 0, loses arbitration and turns off its data output stage. The master that lost arbitration can continue to generate clocks until the end of the byte transfer. If both masters are addressing the same slave device, the arbitration could go into the data phase.

The I²C stops generating SCL when it has detected it has lost arbitration to another master.

Figure 62 illustrates the timing of when two masters are arbitrating on the bus.

Figure 62: Multiple Master Arbitration



Arbitration is not allowed between the following conditions:

- A RESTART condition and a data bit
- A STOP condition and a data bit
- A RESTART condition and a STOP condition

Slaves are not involved in the arbitration process.

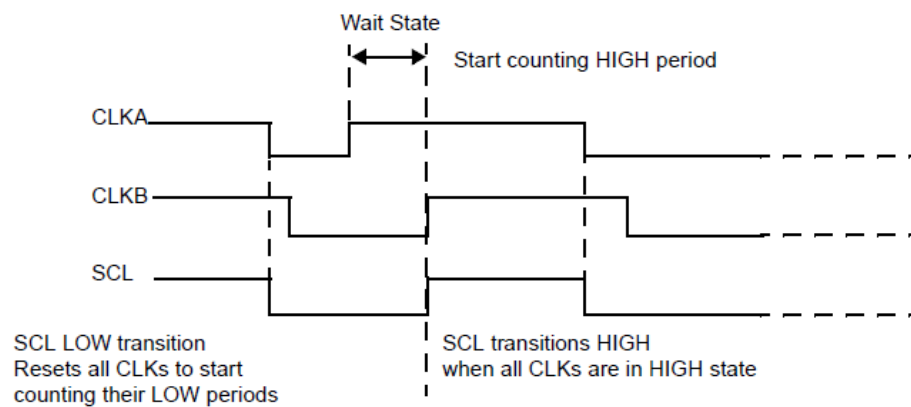
19.5.4 Clock Synchronization

When two or more masters try to transfer information on the bus at the same time, they must arbitrate and synchronize the SCL clock. All masters generate their own clock to transfer messages. Data is valid only during the high period of SCL clock. Clock synchronization is performed using the

wired-AND connection to the SCL signal. When the master transitions the SCL clock to 0, the master starts counting the low time of the SCL clock and transitions the SCL clock signal to 1 at the beginning of the next clock period. However, if another master is holding the SCL line to 0, then the master goes into a HIGH wait state until the SCL clock line transitions to 1.

All masters then count off their high time, and the master with the shortest high time transitions the SCL line to 0. The masters then counts out their low time and the one with the longest low time forces the other master into a HIGH wait state. Therefore, a synchronized SCL clock is generated, which is illustrated in Figure 19-10. Optionally, slaves may hold the SCL line low to slow down the timing on the I²C bus.

Figure 63: Multi-Master Clock Synchronization



19.5.5 Operation Modes

This section provides information on operation modes.



Note

The I²C should only be set to operate as a I²C Master, or I²C Slave, but not both simultaneously. This is achieved by ensuring that Bit 6 (SLAVE_DISABLE) and Bit 0 (MASTER_MODE) of the IIC_CON register are never set to 0 and 1, respectively.

19.5.5.1 Slave Mode Operation

This section discusses slave mode procedures.

Initial Configuration

To use the I²C as a slave, perform the following steps:

1. Disable the I²C by writing a 0 to Bit 0 of the I2C_ENABLE register.
2. Write to the I2C_SAR register (bits 9:0) to set the slave address. This is the address to which the I²C responds.
3. Write to the IIC_CON register to specify which type of addressing is supported (7-bit or 10-bit by setting Bit 3). Enable the I²C in slave-only mode by writing a 0 into bit 6 (SLAVE_DISABLE) and a 0 to Bit 0 (MASTER_MODE).


Note

Slaves and masters do not have to be programmed with the same type of addressing 7- or 10-bit address. For instance, a slave can be programmed with 7-bit addressing and a master with 10-bit addressing, and vice versa.

Enable the TWSI by writing a 1 to Bit 0 of the IIC_ENABLE register.

Slave-Transmitter Operation for a Single Byte

When another I²C master device on the bus addresses the I²C and requests data, the I²C acts as a slave-transmitter and the following steps occur:

1. The other I²C master device initiates a I²C transfer with an address that matches the slave address in the I2C_SAR register of the I²C.
2. The I²C acknowledges the sent address and recognizes the direction of the transfer to indicate that it is acting as a slave-transmitter.
3. The I²C asserts the RD_REQ interrupt (Bit 5 of the I2C_RAW_INTR_STAT register) and holds the SCL line low. It is in a wait state until software responds. If the RD_REQ interrupt has been masked, due to I2C_INTR_MASK [5] register (M_RD_REQ bit field) being set to 0, then Marvell recommends that a hardware and/or software timing routine be used to instruct the CPU to perform periodic reads of the I2C_RAW_INTR_STAT register.
 - a) Reads that indicate I2C_RAW_INTR_STAT [5] (R_RD_REQ bit field) being set to 1 must be treated as the equivalent of the RD_REQ interrupt being asserted.
 - b) Software must then act to satisfy the I²C transfer.
 - c) The timing interval used should be in the order of 10 times the fastest SCL clock period the I²C can handle. For example, for 400 kb/s, the timing interval is 25 μ s.


Note

The value of 10 is recommended here because this is approximately the amount of time required for a single byte of data transferred on the I²C bus.

4. If there is any data remaining in the TX FIFO before receiving the read request, then the I²C asserts a TX_ABRT interrupt (Bit 6 of the I2C_RAW_INTR_STAT register) to flush the old data from the TX FIFO.


Note

Because the I²C TX FIFO is forced into a flushed/reset state whenever a TX_ABRT event occurs, software must release the I²C from this state by reading the I2C_CLR_TX_ABRT register before attempting to write into the TX FIFO. See register I2C_RAW_INTR_STAT for more details.

If the TX_ABRT interrupt has been masked, due to of I2C_INTR_MASK [6] register (M_TX_ABRT bit field) being set to 0, then Marvell recommends that re-using the timing routine (described in the previous step), or a similar one be used to read the I2C_RAW_INTR_STAT register.

- a) Reads that indicate Bit 6 (R_TX_ABRT) being set to 1 must be treated as the equivalent of the TX_ABRT interrupt being asserted.
- b) There is no further action required from software.
- c) The timing interval used should be similar to that described in the previous step for the I2C_RAW_INTR_STAT [5] register.

5. Software writes to the IIC_DATA_CMD register with the data to be written (by writing a 0 in Bit 8).
6. Software must clear the RD_REQ and TX_ABRT interrupts (bits 5 and 6, respectively) of the I2C_RAW_INTR_STAT register before proceeding. If the RD_REQ and/or TX_ABRT interrupts have been masked, then clearing of the I2C_RAW_INTR_STAT register will have already been performed when either the R_RD_REQ or R_TX_ABRT bit has been read as 1.
7. The I²C releases the SCL and transmits the byte.
8. The master may hold the I²C bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

Slave-Receiver Operation for a Single Byte

When another I²C master device on the bus addresses the I²C and is sending data, the I²C acts as a slave-receiver and the following steps occur:

1. The other I²C master device initiates a I²C transfer with an address that matches the I²C slave address in the I2C_SAR register.
2. The I²C acknowledges the sent address and recognizes the direction of the transfer to indicate that the I²C is acting as a slave-receiver.
3. I²C receives the transmitted byte and places it in the receive buffer.



Note

If the RX FIFO is completely filled with data when a byte is pushed, then an overflow occurs and the I²C continues with subsequent I²C transfers. Because a NACK is not generated, software must recognize the overflow when indicated by the I²C (by the R_RX_OVER bit in the IIC_INTR_STAT register) and take appropriate actions to recover from lost data. Therefore, there is a real-time constraint on software to service the RX FIFO before the latter overflow as there is no way to reapply pressure to the remote transmitting master. Users must select a deep enough RX FIFO depth to satisfy the interrupt service interval of their system.

4. I²C asserts the RX_FULL interrupt (I2C_RAW_INTR_STAT [2] register). If the RX_FULL interrupt has been masked, due to setting I2C_INTR_MASK [2] register to 0 or setting I2C_TX_TL to a value larger than 0, then Marvell recommends that a timing routine (described in “Slave-Transmitter Operation for a Single Byte”) be implemented for periodic reads of the I2C_STATUS register. Reads of the I2C_STATUS register, with Bit 3 (RFNE) set at 1, must then be treated by software as the equivalent of the RX_FULL interrupt being asserted.
5. Software may read the byte from the IIC_DATA_CMD register (bits 7:0).
6. The other master device may hold the I²C bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

Slave-Transfer Operation for Bulk Transfers

In the standard I²C protocol, all transactions are single byte transactions and the programmer responds to a remote master read request by writing one byte into the slave TX FIFO. When a slave (slave-transmitter) is issued with a read request (RD_REQ) from the remote master (master-receiver), at a minimum there should be at least one entry placed into the slave-transmitter TX FIFO. I²C is designed to handle more data in the TX FIFO so that subsequent read requests can take that data without raising an interrupt to get more data. Ultimately, this eliminates the possibility of significant latencies being incurred between raising the interrupt for data each time had there been a restriction of having only one entry placed in the TX FIFO.

This mode only occurs when I²C is acting as a slave-transmitter. If the remote master acknowledges the data sent by the slave-transmitter and there is no data in the slave TX FIFO, the I²C holds the

I²C SCL line low while it raises the read request interrupt (RD_REQ) and waits for data to be written into the TX FIFO before it can be sent to the remote master.

If the RD_REQ interrupt is masked, due to Bit 5 (M_RD_REQ) of the I2C_INTR_STAT register being set to 0, then Marvell recommends that a timing routine be used to activate periodic reads of the I2C_RAW_INTR_STAT register. Reads of I2C_RAW_INTR_STAT that return Bit 5 (R_RD_REQ) set to 1 must be treated as the equivalent of the RD_REQ interrupt referred to in this section. This timing routine is similar to that described in “Slave-Transmitter Operation for a Single Byte”.

The RD_REQ interrupt is raised upon a read request, and like interrupts, must be cleared when exiting the interrupt service handling routine (ISR). The ISR allows users to either write one byte or more than one byte into the TX FIFO. During the transmission of these bytes to the master, if the master acknowledges the last byte, then the slave must raise the RD_REQ again because the master is requesting for more data.

If the programmer knows in advance that the remote master is requesting a packet of n bytes, then when another master addresses I²C and requests data, the TX FIFO could be written with n number bytes and the remote master receives it as a continuous stream of data. For example, the I²C slave continues to send data to the remote master as long as the remote master is acknowledging the data sent and there is data available in the TX FIFO. There is no need to hold the SCL line low or to issue RD_REQ again.

If the remote master is to receive n bytes from the I²C but the programmer wrote a number of bytes larger than n to the TX FIFO, then when the slave finishes sending the requested n bytes, it clears the TX FIFO and ignores any excess bytes.

The I²C generates a transmit abort (TX_ABRT) event to indicate the clearing of the TX FIFO in this example. At the time an ACK/NACK is expected, if a NACK is received, then the remote master has all the data it wants. At this time, a flag is raised within the slave state machine to clear the leftover data in the TX FIFO. This flag is transferred to the processor bus clock domain where the FIFO exists and the contents of the TX FIFO are cleared at that time.

19.5.5.2 Master Mode Operation

This section discusses master mode procedures.

Initial Configuration

Perform the following steps to use the I²C as a master:

1. Disable the I²C by writing 0 to the I2C_ENABLE register.
2. Write to the I2C_CON register to set the maximum speed mode supported for slave operation (bits 2:1) and to specify whether the I²C starts its transfers in 7/10 bit addressing mode when the device is a slave (Bit 3).
3. Write to the I2C_TAR register the address of the I²C device to be addressed. It also indicates whether a General Call or a START BYTE command is going to be performed by I²C. The required speed of the I²C master-initiated transfers, either 7-bit or 10-bit addressing, is controlled by the BIT Offset address10_MASTER bit field (bit 12).
4. Enable the I²C by writing a 1 in the I2C_ENABLE register.
5. Now write the transfer direction and data to be sent to the I2C_DATA_CMD register. If the I2C_DATA_CMD register is written before the I²C is enabled, the data and commands are lost as the buffers are kept cleared when I²C is not enabled.



Note

For multiple I²C transfers, perform additional writes to the TX FIFO such that the TX FIFO does not become empty during the I²C transaction. If the TX FIFO is completely emptied at any stage, then further writes to the TX FIFO result in an independent I²C transaction.

Dynamic TAR or BIT Offset address10_MASTER Update

The I²C supports dynamic updating of the TAR (bits 9:0) and BIT Offset address10_MASTER (Bit 12) bit fields of the IIC_TAR register. Users can dynamically write to the I2C_TAR register provided the following conditions are met:

- I²C is not enabled (I2C_ENABLE [0] =0); OR I²C is enabled (I2C_ENABLE [0] =1); AND
- I²C is NOT engaged in any Master (tx, rx) operation (I2C_STATUS [5] =0); AND
- I²C is enabled to operate in Master Mode (I2C_CON [0] =1); AND
- There are NO entries in the TX FIFO (I2C_STATUS [2] =1)

Master Transmit and Master Receive

The TWSI supports switching back and forth between reading and writing dynamically. To transmit data, write the data to be written to the lower byte of the I²C Rx/Tx Data Buffer and Command Register (I2C_DATA_CMD). The CMD bit [8] should be written to 0 for I²C write operations. Subsequently, a read command may be issued by writing “don't cares” to the lower byte of the I2C_DATA_CMD register, and a 1 should be written to the CMD bit. The I²C master continues to initiate transfers as long as there are commands present in the transmit FIFO. If the transmit FIFO becomes empty, the I²C inserts a STOP condition after completing the current transfers.

19.5.6 I2C.CLK Frequency Configuration

When the I²C is configured as a master, the *CNT registers must be set before any I²C bus transaction can occur to ensure proper I/O timing.

The *CNT registers are:

- I2C.SS_SCL_HCNT
- I2C.SS_SCL_LCNT
- I2C.FS_SCL_HCNT
- I2C.FS_SCL_LCNT

19.5.6.1 Calculating High and Low Counts

This section shows how to calculate SCL high and low counts for each speed mode in the I²C. In this TWSI module ic_clk is 16 MHz. The equation to calculate the proper number of ic_clk signals required for setting the proper SCL clocks high and low times is as follows:

$$IIC\_xCNT = (\text{ROUNDUP} (\text{MIN\_SCL\_xxxtime} * \text{OSCFREQ}, 0))$$

ROUNDUP is an explicit Excel function call that is used to convert a real number to its equivalent integer number.

MIN_SCL_HIGHTime = Minimum High Period

MIN_SCL_HIGHTime = 4000 ns for 100 kbps

600 ns for 400 kbps

60 ns for 3.4 Mbs, bus loading = 100pF

160 ns for 3.4 Mbs, bus loading = 400pF

MIN_SCL_LOWtime = Minimum Low Period
MIN_SCL_LOWtime = 4700 ns for 100 kbps
1300 ns for 400 kbps
120 ns for 2.4 Mbs, bus loading = 100pF
320 ns for 2.4 Mbs, bus loading = 400pF
OSCFREQ = ic_clk Clock Frequency (Hz)

For example:

OSCFREQ = 100 MHz
I2Cmode = fast, 400 kbit/s
MIN_SCL_HIGHTime = 600 ns.
MIN_SCL_LOWtime = 1300 ns.
IIC_xCNT = (ROUNDUP(MIN_SCL_HIGH_LOWtime*OSCFREQ,0))
IIC_HCNT = (ROUNDUP(600 ns * 100 MHz,0))
IIC_HCNTSCL PERIOD = 60
IIC_LCNT = (ROUNDUP(1300 ns * 100 MHz,0))
IIC_LCNTSCL PERIOD = 130
Actual MIN_SCL_HIGHTime = 60*(1/100 MHz) = 600 ns
Actual MIN_SCL_LOWtime = 130*(1/100 MHz) = 1300 ns

19.5.7 DMA Controller Interface

The I²C has a built-in DMA capability to request and control transfers. To enable the DMA Controller interface on the I²C, write the DMA Control Register (DMAC.I2C_DMA_CR). Writing a 1 into the TDMAE bit field of DMAC.I2C_DMA_CR register enables the I²C transmit handshaking interface. Writing a 1 into the RDMAE bit field of the DMAC.I2C_DMA_CR register enables the I²C receive handshaking interface.

Refer to the DMA chapter for more information about DMA operation.

19.6 Register Descriptions

This section describes the programmable registers of the I²C. There are a total of 35 registers. A complete description of the I²C registers is located in Appendix A.

20 Synchronous Serial Protocol (SSP)

20.1 Overview

The SSP port is a synchronous serial controller that can be connected to a variety of external Analog-to-Digital converters (ADC), audio and telecommunication CODECs, and many other devices that use serial protocols for data transfer.

The 88MC200 microcontroller contains three SSP interfaces: SSP0, SSP1, and SSP2. The SSP ports are configurable to operate in Master mode (the attached peripheral functions as a slave) or Slave mode (the attached peripheral functions as a master). The SSP ports support serial bit rates from 6.3Kbps (minimum recommended speed) up to 25 Mbps. Serial data sample size can be set to 8, 16, 18, or 32 bits in length. A FIFO is provided for Transmit data and a second independent FIFO is provided for Receive data. The two FIFOs are both 16 x 32 bits wide or both 32 x 16 bits wide. The FIFOs can be loaded or emptied by the Cortex M3 Processor or by DMA burst transfers.

20.2 Features

The enhanced SSP port features are as follows:

- Directly supports Texas Instruments* Synchronous Serial Protocol (SSP), and Motorola* Serial Peripheral Interface (SPI).
- The I²S protocol is supported by programming the PSP
 - I²S Philips standard
 - MSB-justified standard (left justified)
 - Master or Slave mode operation
 - Data transfer up to 25 Mbps
 - Programmable data frame size: 8, 16, 18, or 32 bits
 - Separate FIFO for transmit and receive with 16 x 32 or 32 x 16 bit length
 - Receive-without-Transmit operation
 - Network mode with as many as eight time slots for PSP formats
- Independent transmit/receive in any, all, or none of the time slots
- Supports DMA transfer

20.3 External Signal Descriptions

[Table 50](#) describes the SSP Interface bus signals.

Table 50: SSP Interface Signal Descriptions

Signal Name	Type	Description
SSPx_RXD	Input	Synchronous Serial Protocol Receive Data Serial data in. Sample length is selected by the <Extended Data Size Select> and <Data Size Select> fields in the SSP Control Register 0,

Table 50: SSP Interface Signal Descriptions (Continued)

Signal Name	Type	Description
SSPx_TXD	Output	Synchronous Serial Protocol Transmit Data Serial data out. Sample length is selected by the <Extended Data Size Select> and <Data Size Select> fields in the SSP Control Register 0,
SSPx_CLK	Input/Output	Synchronous Serial Protocol Serial Clock Controls the timing of a serial transfer. SSPx_CLK can be generated internally (master mode) or taken from an external source (slave mode)
SSPx_FRM	Input/Output	Synchronous Serial Protocol Serial Frame Indicator Indicates the beginning and the end of a serialized data sample. The SSPx_FRM can be generated internally (master mode) or taken from an external source (slave mode).

20.4 Operation

Serial data is transferred between the Cortex M3 Processor and an external peripheral through FIFOs in the SSPx port. Data transfers between an SSPx port and memory are initiated by the core or by DMA bursts. Separate data paths for transmitting and receiving permit simultaneous transaction in both directions, depending on the protocols chosen. The core and DMA bursts transaction can transfer data between:

- Memory to the FIFO Data Register for the TXFIFO
- The FIFO Data Register to memory for the RXFIFO

Data written to the FIFO Data Register by either the core or DMA is automatically transferred to the Transmit FIFO. When reading the FIFO Data Register by either the core or DMA, the data in the Receive FIFO is transferred automatically from the FIFO data register.

20.4.1 FIFO Operation

Two separate and independent FIFOs are present for transmitting (TXFIFO to peripheral) and receiving (RXFIFO from peripheral) serial data. The FIFOs are filled or emptied by the Cortex-M3 core or DMA bursts. The data is accessed through the TXFIFO and RXFIFO. The Cortex-M3 accesses are normally triggered by an interrupt caused by an SSP Status Register event (see the Appendix in this manual) and must always be 32 bits wide. The Cortex-M3 writes to the TXFIFO are 32-bits wide, but bits beyond the programmed FIFO data size are ignored. The Cortex-M3 Reads from the RXFIFO are also 32 bits wide with zeroes inserted in the MSBs down to the programmed data size.

The TXFIFO and RXFIFO can also be accessed by DMA bursts, which must be 8, 16, or 32 bytes in length, and must transfer one FIFO entry per access. When the SSP_SSCR0[EDSS] bit is set, the SSPx port must be configured as a 32-bit peripheral. The DMA burst transaction width must be programmed to at least the same data size programmed into this SSP_SSCR0[EDSS] and SSP_SSCR0[DSS] fields.

The TXFIFO and RXFIFO are each accessed as one 32-bit location by the Cortex-M3 core. For data transmission, the SSPx port transmits the data from the TXFIFO to the external peripheral via the SSPx_TXD interface. Data received from the external peripheral via the SSPx_RXD interface is converted to parallel words and written into the RXFIFO.

An interrupt or DMA service request is generated if a programmable FIFO trigger threshold exceeded which signals the Cortex-M3 or DMA to empty the RXFIFO or refill the TXFIFO.

The TXFIFO and RXFIFO are differentiated by whether the access is a Read or a Write transfer. Reads from the Data Register automatically target the RXFIFO. Writes to the FIFO Data Register automatically target the TXFIFO. From a memory-map perspective, the TXFIFO and the RXFIFO are at the same address. Each FIFO is 16 rows deep x 32 bits wide for a total of 16 data samples. Each sample can be 8, 16, 18, or 32 bits in length.

20.4.1.1 Parallel Data Formats for FIFO Storage

Data in the FIFOs is either stored with one 32-bit value per data sample (in non-packed or data size greater than 16 bits) or in a 16-bit value in packed mode when the data is 8 or 16 bits. Within each 32- or 16-bit field, the stored data sample is right-justified, with the LSB of the word in Bit 0. In the Receive FIFO, unused bits are packed as zeroes above the MSB. In the Transmit FIFO, unused “don’t-care” bits are above the MSB. For example, DMA accesses do not have to write to the unused bit locations. Logic in the SSP automatically formats data in the Transmit FIFO so that the sample is properly transmitted on SSPx_TXD in the selected frame format.

20.4.1.2 FIFO Operation in Packed Mode

When the TXFIFO and RXFIFO are operating in packed mode, each FIFO is 32 rows deep x 16-bits wide for a total of 32 data samples. For packed mode, each sample can be 8 or 16 bits in length. When the data is serialized and transmitted, Bits 15 to 0 are transmitted first, followed by Bits 31 to 16. When the TXFIFO and RXFIFO are operating in packed mode, they may best be thought of as a single entry of 32 bits holding two 8- or 16-bit samples. Thus, the Cortex-M3 Processor or the DMA should write and read 32 bits of data at a time where each Write or Read transfers two samples. The entire FIFO width (32 bits) must be read/written in this mode. The SSPx port does not support writing two separate 16-bit samples in this mode. The SSP FIFO thresholds align in 32-bit data size.

20.4.1.3 Trailing Bytes in RXFIFO

When the number of samples in the RXFIFO is less than its trigger threshold level and no additional data is received, the remaining bytes are called *RXFIFO trailing bytes*. RXFIFO trailing bytes can be handled by the Cortex-M3 core. RXFIFO trailing bytes are identified by means of a time-out mechanism and the existence of data within the RXFIFO after timeout.



Note

When FIFO packed mode is used, the DMA cannot be used to handle the RXFIFO trailing bytes. The RXFIFO trailing bytes must be handled by the Cortex-M3 core.

Timeout

A timeout condition exists when the RXFIFO has been idle for a period of time defined by the value programmed within the SSP_SSTO[Timeout Value] field. When a timeout occurs, the Receiver Time-outInterrupt bit, SSP_SSSR[TINT], is set to 1, and if the Receiver Time-out Interrupt Enable bit, SSP_SSCR1[TINTE], is set, a timeout interrupt signals the Cortex-M3 processor that a timeout condition has occurred. The timeout timer is reset after a new data sample is received into the RXFIFO. Once the SSP_SSSR[TINT] bit is set, it must be cleared by writing 0x1 to the SSP_SSCR1[TINT] bit. Clearing it also causes the timeout interrupt, if enabled, to be de-asserted.

Removing FIFO Trailing Bytes

When the Trailing Byte, SSP_SSCR1[TRAIL], bit is cleared, trailing bytes left in the RXFIFO are handled by the Cortex-M3 Processor programmed I/O method by default.

If a timeout occurs, the Cortex-M3 Processor is only interrupted by a timeout interrupt if it has been enabled by setting the Receiver Time-out Interrupt Enable SSP_SSCR1[TINTE] field. To read out the trailing bytes from the RXFIFO, software should wait for the timeout interrupt and then read all trailing bytes as indicated by the Odd Sample Status, SSP_SSSR[OSS], Receive FIFO Level,

SSP_SSSR[RFL], and Receive FIFO Not Empty, SSP_SSSR[RNE] fields in the SSP Status Register. To remove trailing bytes using PIO, enable the timeout interrupt by setting the SSP_SSCR1[TINTE] field.


Note

If FIFO Packed mode is enabled (SSP_SSRC0[FPCKE]=1), trailing bytes must be removed using programmed I/O. If the SSP_SSSR[OSS] field in SSP Status Register is set to 1, then the last FIFO line only contains one sample.

20.4.2 Using Programmed I/O Data Transfers

FIFO filling and emptying can be performed by the Cortex-M3 Processor in response to an interrupt from the FIFO logic. Each FIFO has a programmable FIFO trigger threshold that triggers an interrupt. When the number of entries in the RXFIFO exceeds the RXFIFO Trigger Threshold (SSP_SSCR1[RFT]) field in the SSP Control Register 1, an interrupt is generated (if enabled) that signals Cortex-M3 Processor to empty the RXFIFO. When the number of entries in the TXFIFO is less than or equal to the TXFIFO Trigger Threshold (SSP_SSCR1[TFT]) field in the SSP Control Register 1 plus 1, an interrupt is generated (if enabled) that signals the Cortex-M3 Processor to refill the TXFIFO.

The SSP Status Register can be polled to determine how many samples are in a FIFO and whether the FIFO is full or empty. Software is responsible for ensuring that the proper RXFIFO Trigger Threshold and TXFIFO Trigger Threshold values are chosen to prevent Receive FIFO Overrun and Transmit FIFO Underrun (in the SSP Status Register) error conditions.

20.4.3 Using DMA Data Transfers

The DMA controller can also be programmed to transfer data to and from the FIFOs. To prevent overruns of the TXFIFO or underruns of the RXFIFO when using the DMA, be careful when setting the FIFO trigger threshold levels by setting the correct DMA burst sizes. TXFIFO overruns and RXFIFO underruns are silent errors: There is no indication of the overrun or underrun condition other than missing data at the receiving end of the link. The DMA burst size must be smaller than the trigger threshold.

When not using packed mode, the SSP stores one data sample per FIFO location where each FIFO has 16 locations. When using packed mode, the SSP stores two data samples per FIFO location where each FIFO has 16 locations.

Because the SSP is not flow controlled and has only 16 location FIFOs, software must program the TXFIFO threshold (SSP_SSCR1[TFT]) field, RXFIFO threshold (SSP_SSCR1[RFT]) field, and the DMA burst size to ensure that a TXFIFO overrun or RXFIFO underrun does not occur.

Software must also ensure that the SSP DMA requests are properly prioritized in the system to prevent overruns and underruns.

20.4.4 Data Formats

This section describes the types of formats used to transfer serial data between the Cortex-M3 core and external peripherals.

20.4.4.1 Serial Data Formats for Transfer to/from Peripherals

Four interface signals for each SSPx port transfer data between the Cortex-M3 core and external peripherals. Although serial-data formats exist, each has the same basic structure, and in all cases, the interface signals used are:

- SSPx_CLK – Defines the bit rate at which serial data is driven onto and sampled from the port
- SSPx_FRM – Defines the boundaries of a basic data “unit” which is comprised of multiple serial bits

- SSPx_TXD – The serial datapath for transmitted data from the SSPx port to the peripheral
- SSPx_RXD – The serial datapath for received data from peripheral to the SSPx port

A data frame can contain 8, 16, 18, or 32 bits (see SSP_SSCR0[EDSS] and SSP_SSCR0[DSS] fields in the Cortex-M3 Processor Registers. Serial data is transmitted with the MSb first. The formats directly supported are the Motorola SPI and Texas Instruments SSP. The I²S protocol is supported by programming the PSP format.

The SSPx_FRM function and use varies between each format:

SPI format: SSPx_FRM functions as a chip select to enable the external device (target of the transfer) and is held active-low during the data transfer. During continuous transfers, the SSPx_FRM signal can be either held low or pulsed depending upon the value of the Motorola* SPI SSPx_CLK phase setting, SSP_SSCR1[SPH], field in the SSP Control Register 1. Master and Slave modes are supported. SPI is a full-duplex format.

SSP format: SSPx_FRM is pulsed high for one (serial) data period at the start of each frame. Master and Slave modes are supported. SSP is a full-duplex format.

PSP format (I²S): SSPx_FRM is programmable in direction, delay, polarity, and width. Master and Slave modes are supported. PSP can be programmed to be either full- or half-duplex format.

The SSPx_CLK function and use varies between each format:

SPI format: Programmers choose which edge of SSPx_CLK to use for switching Transmit data and for sampling Receive data. In addition, moving the phase of SSPx_CLK can be user-initiated, shifting its active state one-half cycle earlier or later at the start and end of a frame. Master and Slave modes are supported, and in both, the SSPx_CLK only toggles during active transfers (does not run continuously).

SSP format: Data sources switch Transmit data on the rising edge of SSPx_CLK and sample Receive data on the falling edge. Master and Slave modes are supported. When driven by the SSPx port, the SSPx_CLK only toggles during active transfers (not continuously) unless the SSP_SSCR1[SCFR], SSP_SSCR1[ECRA], or SSP_SSCR1[ECRB] functions are used.

When the SSPx_CLK is driven by another device, it is allowed to be either continuous or only driven during transfers.

PSP format (I²S): Programmers choose which edge of SSPx_CLK to use for switching Transmit data and for sampling Receive data. In addition, programmers can control the Idle state for SSPx_CLK and the number of active clocks that precede and follow the data transmission. Master and Slave modes are supported. When driven by the SSPx port, the SSPx_CLK toggles only during active transfers, not continuously, unless the SSP_SSCR1[SCFR], SSP_SSCR1[ECRA], or SSP_SSCR1[ECRB] functions are used. When the SSPx_CLK is driven by another device, it is allowed to be either continuous or driven only during transfers, but certain restrictions on PSP parameters apply (see Programmable Serial Protocol (PSP) Format).

Normally, if the serial clock (SSPx_CLK) is driven by the SSPx port, it toggles only while an active data transfer is underway. However, there are several conditions that may cause the clock to run continuously. If the Receive-without-Transmit mode is enabled by setting the Receive Without Transmit SSP_SSCR1[RWOT], field *and* the frame format is not Microwire then the SSPx_CLK toggles regardless of whether Transmit data exists within the Transmit FIFO. The SSPx_CLK also toggles continuously if the SSPx port is in Network mode, or if the SSP_SSCR1[ECRA] or SSP_SSCR1[ECRB] bits are enabled. At other times, SSPx_CLK is held in an inactive or idle state, as defined by the specified protocol under which it operates.

20.4.4.2 TI-SSP Format Details

When outgoing data in the SSP controller is ready to transmit, SSPx_FRM asserts for one clock period. On the following clock, data to be transmitted is driven on SSPx_TXD one bit at a time, with the MSB first. For Receive data, the peripheral similarly drives data on the SSPx_RXD pin. Word

length can be 8, 16, 18, or 32 bits. All output transitions occur on the rising edge of SSPx_CLK while data sampling occurs on the falling edge. The SSPx_TXD signal either retains the value of the last bit sent (bit 0) or goes to a high impedance state at the end of the transfer. If the SSPx port is disabled or reset, SSPx_TXD is forced to zero (unless the TXD Tri-State Enable, SSP_SSCR1[TTE], bit is set, in which case it goes into a high impedance state).

Figure 64 shows the TI Synchronous Serial Protocol for a single transmitted frame. Figure 65 shows the TI Synchronous Serial Protocol when back-to-back frames are transmitted. Once the Transmit FIFO contains data, SSPx_FRM is pulsed high for one SSPx_CLK period and the value to be transmitted is transferred from the Transmit FIFO to the Transmit Logic Serial Shift register. On the next rising edge of SSPx_CLK, the most significant bit of the eight to 32-bit data frame is shifted to the SSPx_TXD pin. Likewise, the MSB of the received data is shifted onto the SSPx_RXD pin by the off-chip serial slave device. Both the SSP port and the off-chip serial slave device then latch each data bit into the serial shifter on the falling edge of each SSPx_CLK. The received data is transferred from the serial shifter to the Receive FIFO on the first rising edge of SSPx_CLK after the last bit has been latched.

For back-to-back transfers, the start of one frame immediately follows the completion of the previous. The MSb of one transfer immediately follows the LSb of the preceding with no "dead" time between them.

When the enhanced SSPx port is a master to the frame synch (SSPx_FRM) and a slave to the clock (SSPx_CLK), then at least three extra clocks (SSPx_CLK) are needed at the beginning and end of each block of transfers to synchronize control signals from the ARM peripheral bus (APB) clock domain into the SSP clock domain (a block of transfers is a group of back-to-back continuous transfers).



Note

When configured as either master or slave to SSPx_CLK or SSPx_FRM, the SSP port continues to drive SSPx_TXD until the last bit of data is sent (the LSB) or the SSPx_TXD line becomes high impedance. If SSP_SSCR0[SSE] is cleared, the SSPx_TXD line goes low. SSP_SSPP[EDTS] has no effect when in SSP mode. SSPx_RXD is undefined before the MSB is sent and after the LSB is sent. SSPx_RXD must not float.

Figure 64: Texas Instruments Synchronous Serial Frame Protocol (Single Transfers)

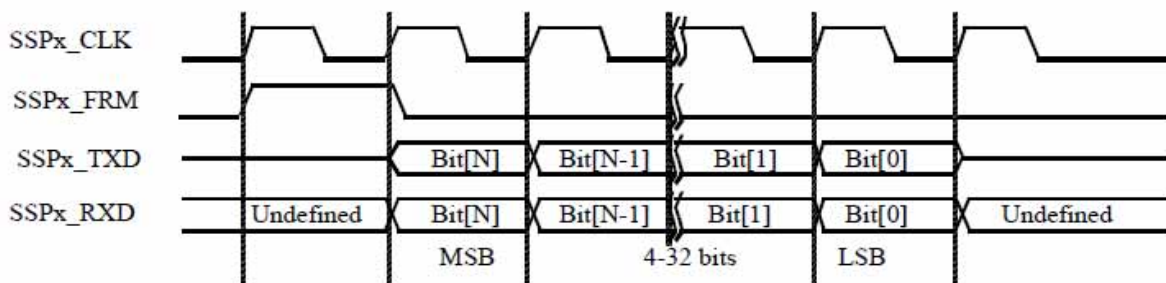
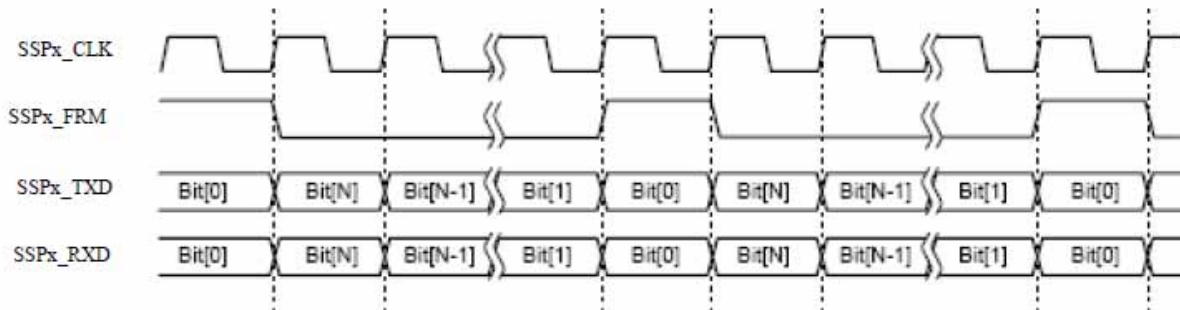


Figure 65: Texas Instruments Synchronous Serial Frame Protocol (Multiple Transfers)



20.4.4.3 Motorola SPI Format Details

The SPI format has four possible sub-modes depending on the SSPx_CLK edges selected for driving data and sampling received data and on the selection of the phase mode of SSPx_CLK (see [Serial Clock Phase \(SPH\)](#), on page 238, for a complete description of each sub-mode).

When the SSP port is disabled or in idle mode, SSPx_CLK and SSPx_TXD are low and SSPx_FRM is high. When transmit data is ready to be sent, SSPx_FRM goes low (one clock period before the first rising edge of SSPx_CLK) and stays low for the remainder of the frame. The most significant bit of the serial data is driven onto SSPx_TXD one half-cycle later. Halfway into the first bit period, SSPx_CLK asserts high and continues toggling for the remaining data bits. Data transitions on the falling edge of SSPx_CLK and is sampled on the rising edge of SSPx_CLK. 8, 16, 18, or 32 bits can be transferred per frame.

With the assertion of SSPx_FRM, Receive data is driven simultaneously from the peripheral on SSPx_RXD, MSb first. Data transitions on SSPx_CLK falling edges and is sampled by the controller on SSPx_CLK rising edges. At the end of the frame, SSPx_FRM is de-asserted high one clock period (one half clock cycle after the last falling edge of SSPx_CLK) after the last bit has been latched at its destination and the completed incoming word is shifted into the "incoming" FIFO. The peripheral can drive SSPx_RXD to a high-impedance state after sending the last bit of the frame. SSPx_TXD retains the last value transmitted when the controller goes into Idle mode, unless the enhanced SSP port is disabled or reset (which forces SSPx_TXD to zero).

For back-to-back transfers, start and completion are like those of a single transfer, but SSPx_FRM does not de-assert between words. Both transmitter and receiver are configured for the word length and internally track the start and end of frames. There are no "dead" bits; the LSb of one frame is followed immediately by the MSb of the next.

When in Motorola SPI format, the enhanced SSPx port can be either a master or a slave device, but the clock and frame direction must be the same. For example, the SSP Serial Bit Rate Clock Direction, SSP_SSCR1[SCLKDIR], and the SSP Frame Direction, SSP_SSCR1[SFRMDIR], fields must either both be set or cleared.

When in Motorola SPI format, if the SSP port is the master and SSP_SSPSP[ETDS] is cleared, the end-of-transfer data state for SSPx_TXD is low. If the SSP port is the master and SSP_SSPSP[ETDS] is set, the end-of-transfer data state for SSPx_TXD remains at the last bit transmitted (LSB). If the SSP port is the slave, then the SSP_SSPSP[ETDS] is undefined.

SSPx_RXD is undefined before the frame is active and after the LSB is received. SSPRXD must not float. When the SSP port is configured as a master and SSP_SSCR1[TTE] is set, SSP_SSPSP[ETDS] is ignored and SSPx_TXD becomes high impedance between active transfers).



Note

The input clock to the SSPx port must not be active when SSPx_FRM is de-asserted. When the SSP port is slave to clock and frame, SSP_SSCR1[SCFR] must be set.

Serial Clock Phase (SPH)

The phase relationship between SSPx_CLK and SSPx_FRM when the Motorola SPI protocol is selected is controlled by SSP_SSCR1[SPH].

The combination of the SSP_SSCR1[SPO] and SSP_SSCR1[SPH] settings determine when SSPx_CLK is active during the assertion of SSPx_FRM and which SSPx_CLK edge transmits and receives data on SSPx_TXD and SSPx_RXD.

When SPH is cleared, SSPx_CLK remains in its inactive (idle) state (as determined by SSP_SSCR1[SPO]) for one full cycle after SSPx_FRM is asserted low at the beginning of a frame. SSPx_CLK continues to toggle for the rest of the frame. It is then held in its inactive state for one-half of an SSPx_CLK period before SSPx_FRM is de-asserted high at the end of the frame. When SPH is set, SSPx_CLK remains in its inactive or idle state (as determined by SSP_SSCR1[SPO]) for one-half cycle after SSPx_FRM is asserted low at the beginning of a frame. SSPx_CLK continues to toggle for the remainder of the frame and is then held in its inactive state for one full SSPx_CLK period before SSPx_FRM is de-asserted high at the end of the frame. When programming SSP_SSCR1[SPO] and SSP_SSCR1[SPH] to the same value (both set or both cleared), transmit data is driven on the falling edge of SSPx_CLK and receive data is latched on the rising edge of SSPx_CLK. When programming SSP_SSCR1[SPO] and SSP_SSCR1[SPH] to opposite values (one set and the other cleared), transmit data is driven on the rising edge of SSPx_CLK and receive data is latched on the falling edge of SSPx_CLK. Refer to [Figure 66](#), [Figure 67](#), [Figure 68](#), and [Figure 69](#).

Figure 66: Motorola SPI Frame Protocol (Single Transfers)

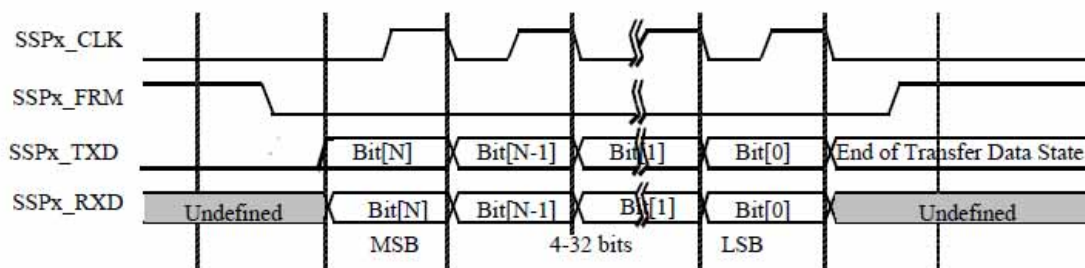


Figure 67: Motorola SPI Frame Protocol (Multiple Transfers)

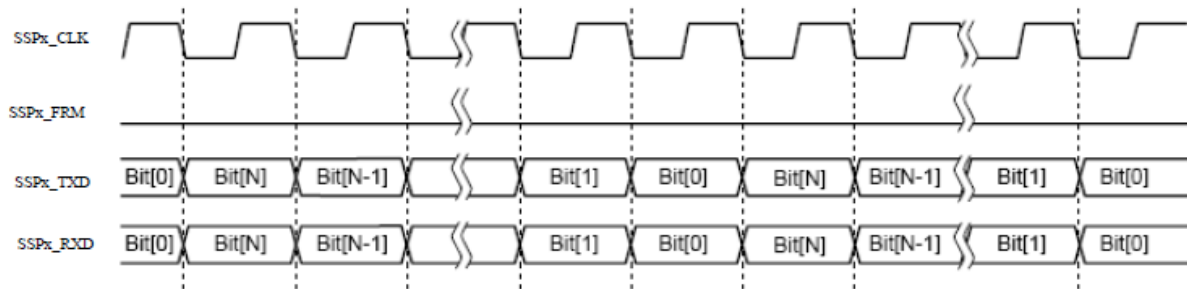
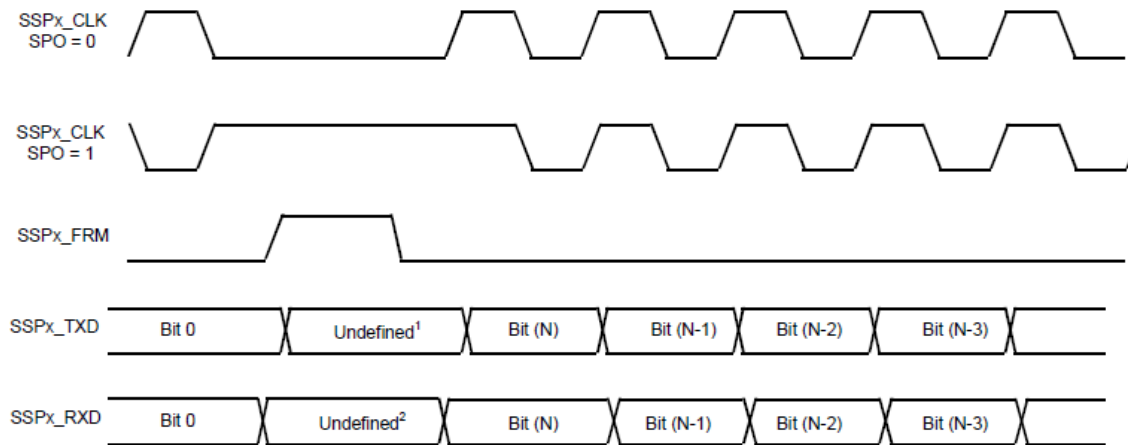
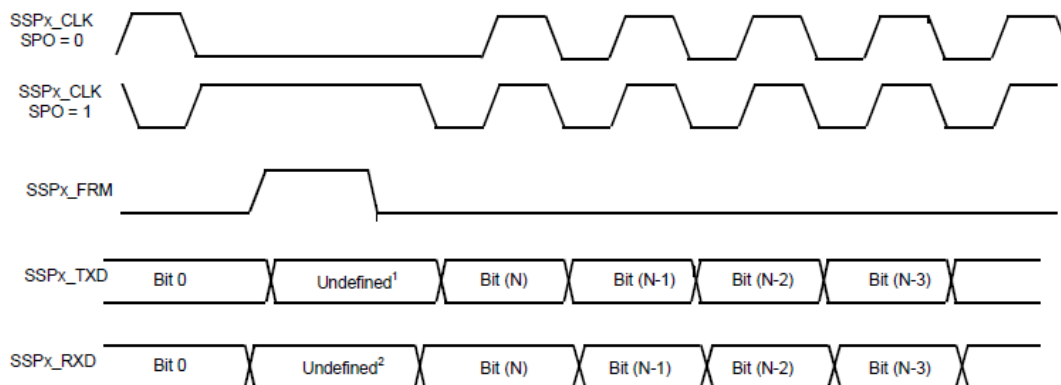


Figure 68: Motorola SPI Frame Protocols for SPO and SPH Programming (SPH Set)



1. SSPx_TXD will be tri-stated at this point if the TTE bit is set.
2. SSPx_RXD should not float
3. SPH = 1

Figure 69: Motorola SPI Frame Protocols for SPO and SPH Programming (SPH Cleared)



1. SSPx_TXD will be tri-stated at this point if the TTE bit is set.
2. SSPx_RXD should not float
3. SPH = 0

20.4.5 Programmable Serial Protocol (PSP) Format

The PSP format defines programmable parameters that determine the transfer timings between data samples. Four serial clock modes are defined in the Serial Bit-rate Clock Mode, SSP_SSPSP[SCMODE], field in the SSP Programmable Serial Protocol Register. These modes select the SSPx_CLK rising and falling edges for driving data, sampling received data, and the SSPx_CLK idle state.

As shown in [Table 51](#), the Idle and Disable modes of the SSPx_TXD, SSPx_CLK, and SSPx_FRM interface signals are programmable using the following fields in the SSP Programmable Serial Protocol Register: End Of Transfer Data State (SSP_SSPSP[ETDS]), Serial Frame Polarity (SSP_SSPSP[SFRMP]), and Serial Bit-rate Clock Mode (SSP_SSPSP[SCMODE]). When Transmit data is ready, SSPx_CLK remains in its Idle state for the number of serial clock (SSPx_CLK) periods programmed into the Start Delay (SSP_SSPSP[STRTDLY]) field in the SSP Programmable Serial Protocol Register.

SSPx_CLK then starts toggling. SSPx_TXD remains in the idle state for the number of serial clock periods programmed into the Dummy Start (SSP_SSPSP[DMYSTRT]) field in the SSP Programmable Serial Protocol Register. SSPx_FRM is asserted after the number of half serial clock periods programmed into the Serial Frame Delay (SSP_SSPSP[SFRMDLY]) field. SSPx_FRM remains asserted for the number of serial clock periods programmed into the Serial Frame Width (SSP_SSPSP[SFRMWDTH]) field in the SSP Programmable Serial Protocol Register, then SSPx_FRM de-asserts.

Serial data of 8, 16, 18, or 32 bits can be transferred per frame by setting the SSP_SSCR0[EDSS] and SSP_SSCR0[DSS] fields to the preferred data size select. Once the last bit (LSB) is transferred, SSPx_CLK continues toggling for the number of serial clock periods programmed into the Dummy Stop (SSP_SSPSP[DMYSTOP]) field. Depending on the value programmed into the End Of Transfer Data State (SSP_SSPSP[EDTS]) field when the SSPx port goes into Idle mode, SSPx_TXD either retains the last bit-value transmitted or is forced to 0 unless the SSPx port is disabled or reset, which forces SSPx_TXD to 0.

With the assertion of SSPx_FRM, Receive data is driven simultaneously from the peripheral onto SSPx_RXD, MSb first. Data transitions on the SSPx_CLK edge based on the serial-clock mode that is selected (SSP_SSPSP[SCMODE]) and is sampled by the SSPx port on the opposite clock edge. When the SSPx port is a master to SSPx_FRM and a slave to SSPx_CLK, at least three extra SSPSCLKs are needed at the beginning and end of each block of transfers to synchronize control signals from the APB clock domain into the SSP clock domain (a block of transfers is a group of back-to-back continuous transfers).

In general, because of the programmable nature of the PSP protocol, this protocol can be used to achieve a variety of serial protocols. For example: some DigRF protocol timing can be achieved by programming these values: Start Delay (SPP_SSPSP[STRTDLY]) = 0, Dummy Start (SPP_SSPSP[DMYSTRT]) = 0, Dummy Stop (SPP_SSPSP[DMYSTOP]) = 0, and Serial Frame Delay (SPP_SSPSP[SFRMDLY]) = 0. The SSPx port should be configured as a clock slave (SSP Serial Bit Rate Clock Direction (SSP_SSCR1[SCLKDIR]) = 1) and frame master (SSP Frame Direction (SSP_SSCR1[SFRMDIR]) = 0). Also, the Frame Sync Relative Timing Bit (SSP_SSPSP[FSRT]) field in the SSP Programmable Serial Protocol Register must be set for continuous transfers, and the Serial Frame Width (SSP_SSPSP[SFRMWDTH]) field should be equal to the data sample size.

Table 51: Programmable Protocol Parameters

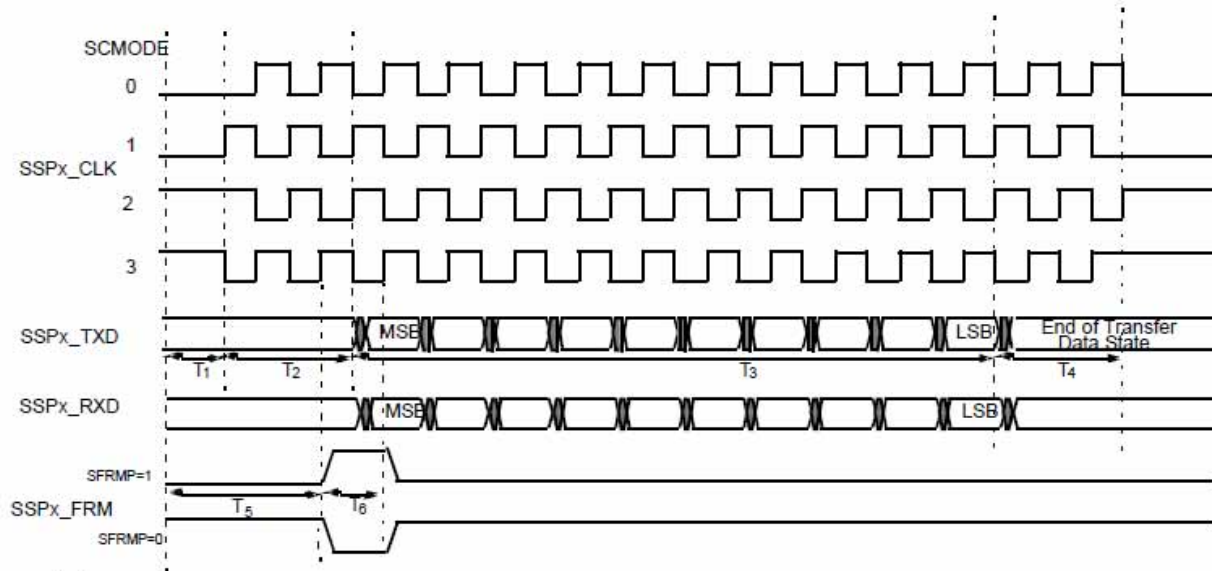
Symbol	Definition	Range	Units
-	Serial clock mode (SSP_SSPSP[SCMODE])	(Drive, Sample, SSPx_CLK Idle) 0 = Fall, rise, low 1 = Rise, fall, low 2 = Rise, fall, high 3 = Fall, rise, high	-
-	Serial frame polarity (SSP_SSPSP[SFRMP])	High or low	-
T1	Start delay (SSP_SSPSP[STRTDLY])	0 to 7	Clock period
T2	Dummy start (SSP_SSPSP[EDMYSTRT] + SSP_SSPSP[DMYSTRT])	0 to 15	Clock period
T3	Data size (SSP_SSCR0[EDSS] and SSP_SSCR0[DSS])	4 to 32	Clock period
T4	Dummy stop (SSP_SSPSP[EDMYSTOP] + SSP_SSPSP[DMYSTOP])	0 to 31	Clock period
T5	SSPSFRM delay (SSP_SSPSP[SFRMDLY])	0 to 127	Half-clock period
T6	SSPSFRM width (SSP_SSPSP[SFRMWDTH])	1 to 63	Clock period
-	End of transfer data state (SSP_SSPSP[ETDS])	Low or Bit 0	-

The SSPx_FRM delay (T5) must not extend beyond the end of T4. The SSPx_FRM width (T6) must be asserted for at least one SSPx_CLK period and should be de-asserted before the end of T4 (for example, in terms of time, not bit values

$$(T5 + T6) \leq (T1 + T2 + T3 + T4), 1 \leq T6 < (T2 + T3 + T4), \text{ and } (T5 + T6) \geq (T1 + 1)$$

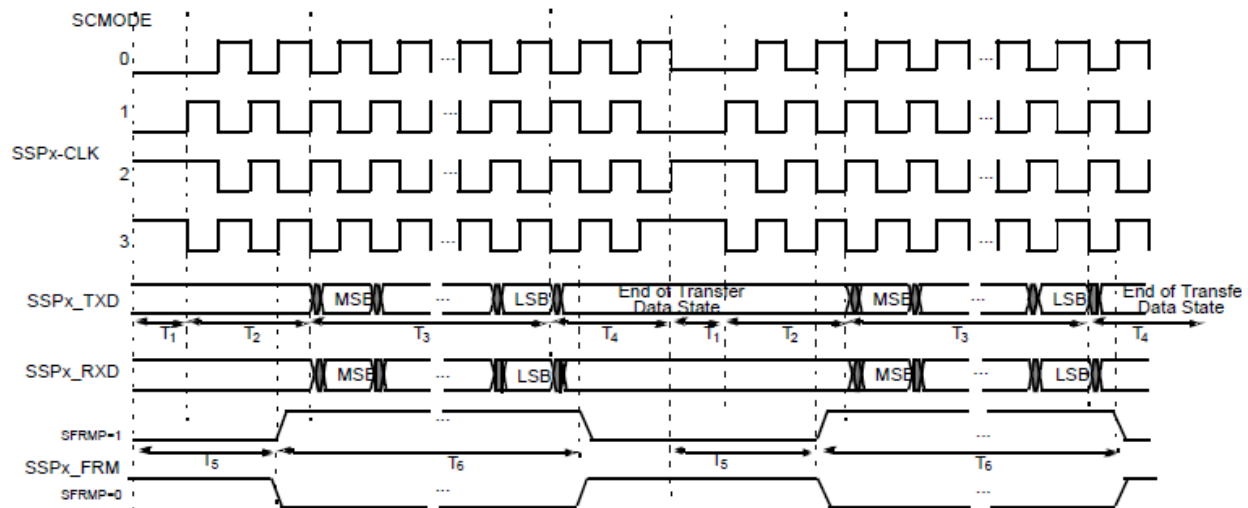
to ensure that SSPx_FRM is asserted for at least two edges of SSPx_CLK). Program T1 to 0b0 when SSPx_CLK is enabled by any of the SSP_SSCR1[SCFR], SSP_SSCR1[ECRA], or SSP_SSCR1[ECRB] fields in the SSP Control Register 1. While the PSP can be programmed to generate the assertion of SSPx_FRM during the middle of the data transfer (for example, after the MSB has been sent), the SSPx port is unable to Receive data in frame-Slave mode (SSP_SSPSP[SFRMDIR] is set, if the assertion of the frame is not before the MSB is sent (for example, $T5 \leq T2$ if the SSP_SSCR1[SFRMDIR] bit is set). Transmit data transitions from the end-of-transfer-data state (SSP_SSPSP[ETDS]) to the next MSB data value upon assertion of the internal version of SSPx_FRM. Program the SSP_SSPSP[STRTDLY] field to 0x00 whenever SSPx_CLK or SSPx_FRM is configured as an input (for example, SSP_SSCR1[SCLKDIR] and SSP_SSCR1[SFRMDIR] are cleared. See [Figure 70](#) and [Figure 71](#).

Figure 70: Programmable Serial Protocol Format



Note: If SSPx port is the master of SSPx_CLK (output) and SSPSP_x[ETDS=0], the end-of-transfer-data state for the SSPx_TXD line is 0. If the SSP is the master of the clock, and the SSPSP[ETDS] bit is set, then the SSPx_TXD line remains at the last bit transmitted (LSB). If the SSPx port is a slave to SSPx_CLK (input), and modes 1 or 3 are used, then the End of Transfer data state (ETDS) can only change from the LSB if more SSPx_CLKs are sent to the SSPx port

Figure 71: Programmable Protocol Format (Consecutive Transfers)

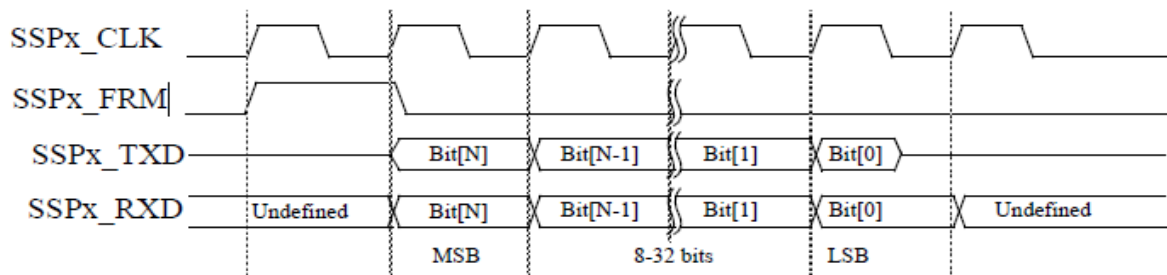


20.4.5.1 High Impedance on SSPx_TXD

The SSP supports placing the SSPx_TXD into high impedance during idle times instead of driving SSPx_TXD as controlled by the TXD Tri-State Enable (SSP_SSCR1[TTE]) and TXD Tri-State Enable On Last Phase (SSP_SSCR1[TTELP]) fields in the SSP Control Register 1. The SSP_SSCR1[TTE] enables a high-impedance state on SSPx_TXD. The SSP_SSCR1[TTELP] determines on which SSPx_CLK phase SSPx_TXD becomes high impedance.

See [Figure 72](#), [Figure 73](#), [Figure 74](#), [Figure 75](#), and [Figure 76](#).

Figure 72: TI SSP with SSP_SSCR1[TTE] = 1 and SSP_SSCR1[TTELP] = 0



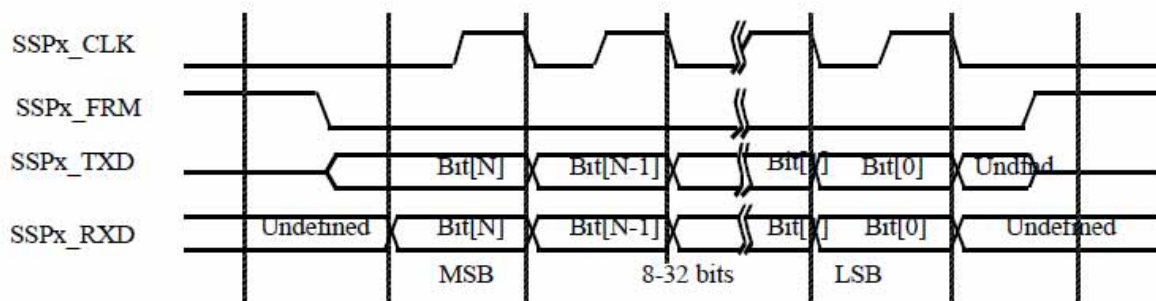
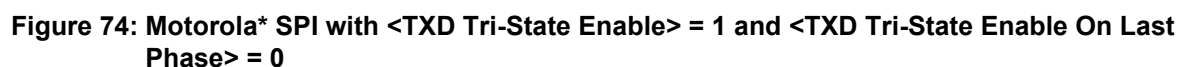


Figure 75: PSP Format with SSP_SSCR1[TTE] = 1, SSP_SSCR1[TTELP] = 0, and SSP_SSCR1[SFRMDIR] = 1

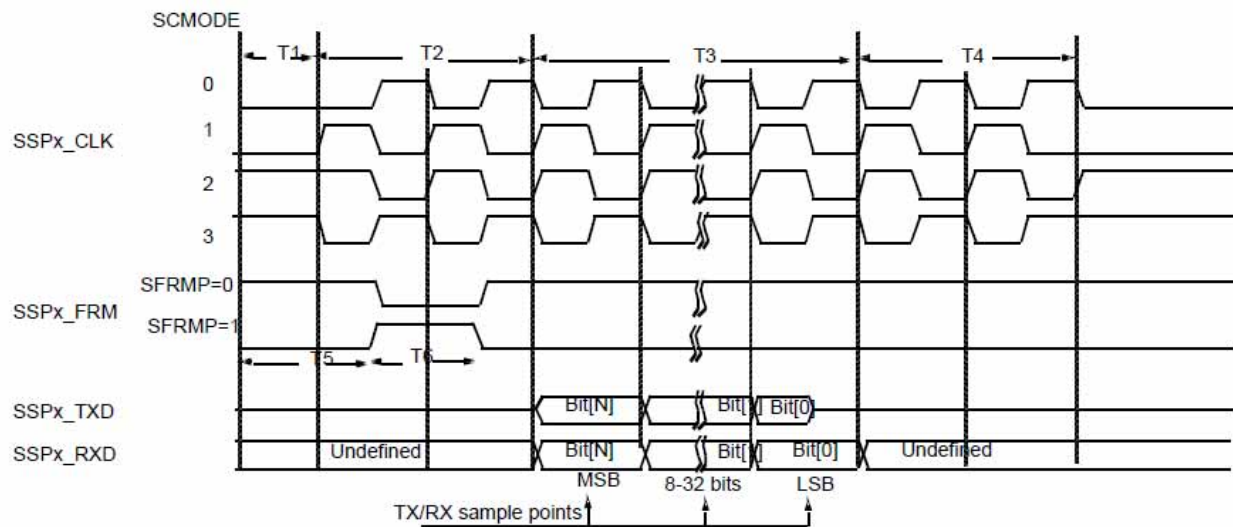
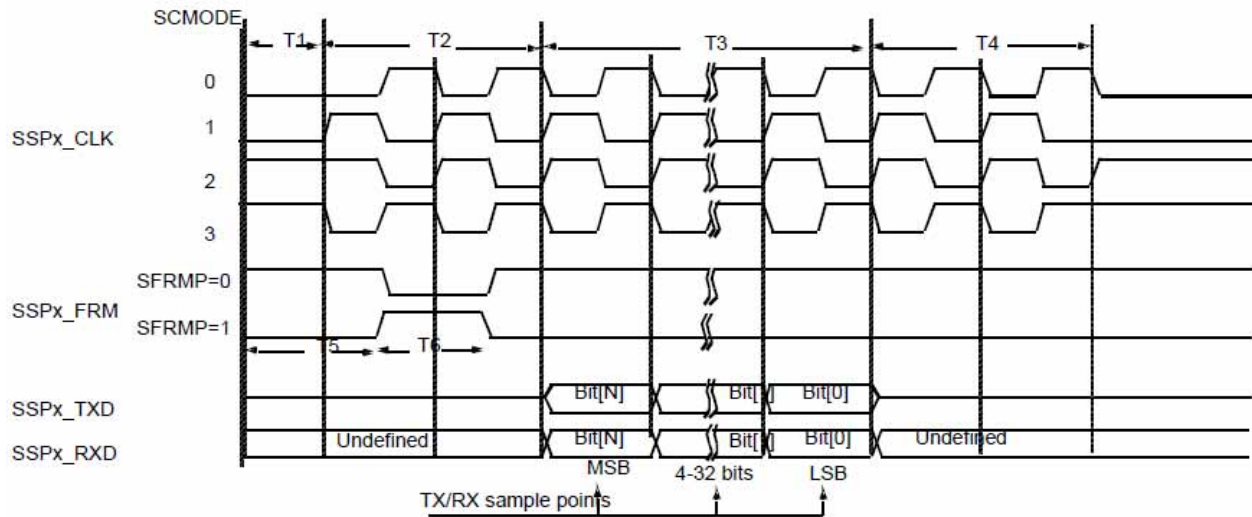


Figure 76: PSP Format with SSP_SSCR1[TTE] = 1, and either SSP_SSCR1[TTELP] = 1, or SSP_SSCR1[SFRMDIR] = 0



20.4.6 Network Mode

The SSP_SSCR0[MOD] bit selects between Normal and Network modes. Normal mode (MOD 0x0) is used when using the Texas Instruments* Synchronous Serial Protocol (SSP), and the Motorola* Serial Peripheral Interface (SPI). Network mode (MOD = 0x1) is used for emulating the I2S protocol. Software should set MOD only when using the PSP format. If the SSPx port is a master of the clock and SSP_SSCR1[SCLKDIR] is cleared, then setting MOD causes the SSPx_CLK to run continuously.

When in Network mode, only one SSPx_FRM is sent (master mode) or received (slave mode) for the number of time slots programmed into the SSP_SSCR0[FRDC] field. When beginning in Network mode, while the SSPx port is a master to the SSPx_FRM interface signal, the first SSPx_FRM signal does not occur until after data is in the TXFIFO. After assertion of the first SSPx_FRM signal, if the SSP is a master to SSPx_FRM, subsequent SSPx_FRM signals continue to assert regardless of whether data resides in the TXFIFO. Therefore, the transmit underrun bit, SSP_SSSR[TUR], is set to 0b1 if there is no data in the TXFIFO and the SSPx port is programmed to drive SSPx_TXD data in the current time slot, even if the SSPx port is master to SSPx_FRM. When using PSP format in Network mode, the parameters SFRMDLY, STRTDLY, DMYSTOP, DMYSTRT must all be 0b0. The other parameters SFRMP, SCSMODE, FSRT, SFRMWDTH are programmable.

When the SSPx port is a master to the SSPx_FRM signal and a need arises to exit from Network mode, software should:

- Clear the SSP_SSCR0[MOD] bit. SSP_SSCR0[SSE] does not need to change.
- Wait until SSP_SSTSS[NMBSY] is cleared.
- Disable the SSPx port by clearing SSP_SSCR0[SSE].
- Before exiting Network mode, verify the TXFIFO is empty (SSP_SSSR[TFL]=0b0000 and SSP_SSSR[TNF]=0b1.)

If data remains in the TXFIFO after the Network mode is exited, a non-Network mode frame will be sent.

Due to synchronization delay between the internal bus and the SSPx port clock domain, one extra frame may be transmitted after software clears the SSP_SSCR0[MOD] bit. The SSPx port continues to drive SSPx_CLK (if SSP_SSCR1[SCLKDIR] is cleared) and SSPx_FRM (if SSP_SSCR1[SFRMDIR] is cleared) until the end of the last valid time slot.

If the SSPx port is a slave to both SSPx_CLK (SSP_SSCR1[SCLKDIR] set) and SSPx_FRM (SSP_SSCR1[SFRMDIR] set), the SSP_SSTSS[NMBSY] bit remains asserted until the SSP_SSCR0[MOD] bit is cleared or until one SSPx_CLK after the end of the last valid time slot.



Note

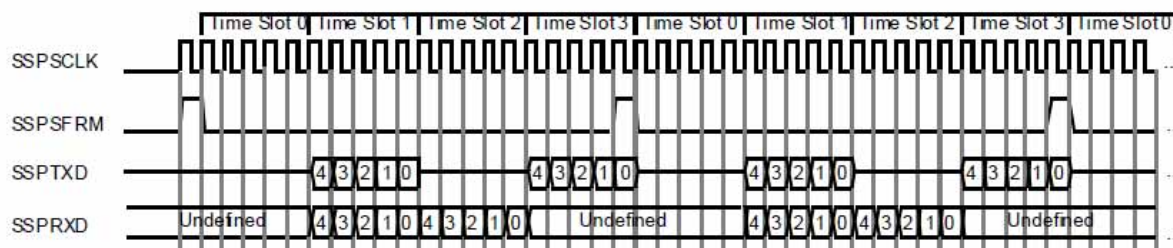
When operating in Slave mode (SSP_SSCR1[SCLKDIR] = SSP_SSCR1[SFRMDIR] = 1) the external codec must provide the correct number of bits as determined by the frame width (SSP_SSPSP[SFRMWDTH]) and number of time slots (SSP_SSCR0[FRDC]). When the number of bits read in during a Frame cycle do not match the number expected, a Bit Count Error will occur (SSP_SSSR[BCE]) and the contents in the FIFO cannot be guaranteed. Software must be used to handle any error conditions when they occur.

In the next example, a data format of 5 is used. However, a data format of 5 is not a supported data size for the SSP controller. A data size of 5 was used only to reduce the size of the diagram.

20.4.6.1 Network Mode Registers

The register bits and fields that must be programmed for the Network Mode are SSP_SSCR0[MOD], SSP_SSCR0[FRDC], SSP_SSPSP[FSRT], SSP_SSPSP[SFRMWDTH], SSP_SSPSP[SCMODE], SSP_SSPSP[SFRMP], and SSP_SSPSP[SFRMDLY]. The definitions of each of these bits and fields is located in the registers document. See [Figure 77](#).

Figure 77: Network Mode (Example Using 4 Time Slots)



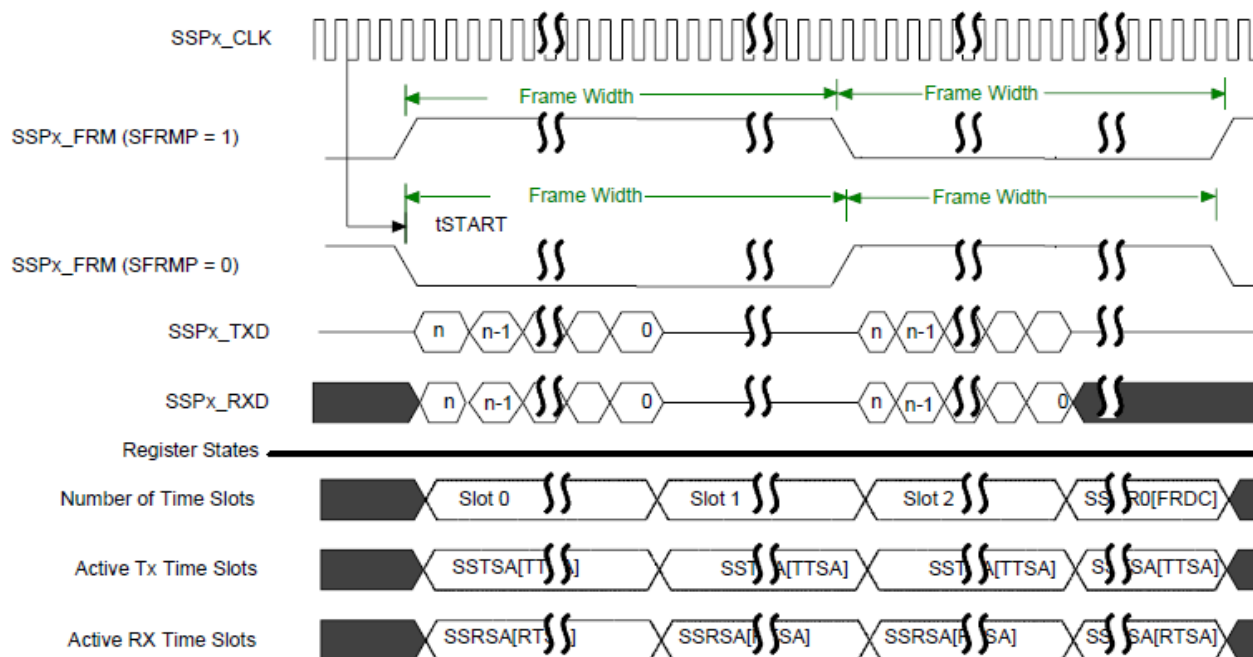
Note: This example has 5 bits of data per sample. The TXD three-state enable bit and the TXD three-state enable on last phase bit are both 0b1. The SSP is a master of SSPSCLK and SSPSFRM. The SSP has been programmed for 4 time slots (SSCR0_x[FRDC]=0b11), the TX Time Slot Active register is programmed to 0x0000 000A (SSTSA[TSA] = b00001010), and the RX Time Slot Active register has been set to 0x0000 0006 (SRSA[RTSA] = b00000110).

NOTE: A data format of 5 is not a supported data size for the SSP controller. A data size of 5 was used only to reduce the size of the diagram.

20.4.7 I²S Emulation Using SSP

Network Mode (SSCR0[MOD] = 1) is used along with the Programmable Serial Protocol (PSP) Format to emulate I²S mode. Figure 78 shows a frame cycle where four time slots are being used and time slots 0 and 3 are being accessed. The total number of time slots to use is defined by the SSCR0[FRDC] register. The slots that data is read into is defined by the RTSA field in the SSTR register, and the slots that data is transmitted out on is defined by the TTSA field in the SSTR register. The number of bits used for each time slot is defined by the Data Size Select bits (EDSS, DSS) in the SSCR0 register. When using master mode (SSCR1[SFRMDIR] = 0x0) the Serial Frame Width (SSPSP[SFRMWDTH]) is used to determine the number of SSPx_CLK for the left and right channels.

Figure 78: Network Mode and PSP Frame Format

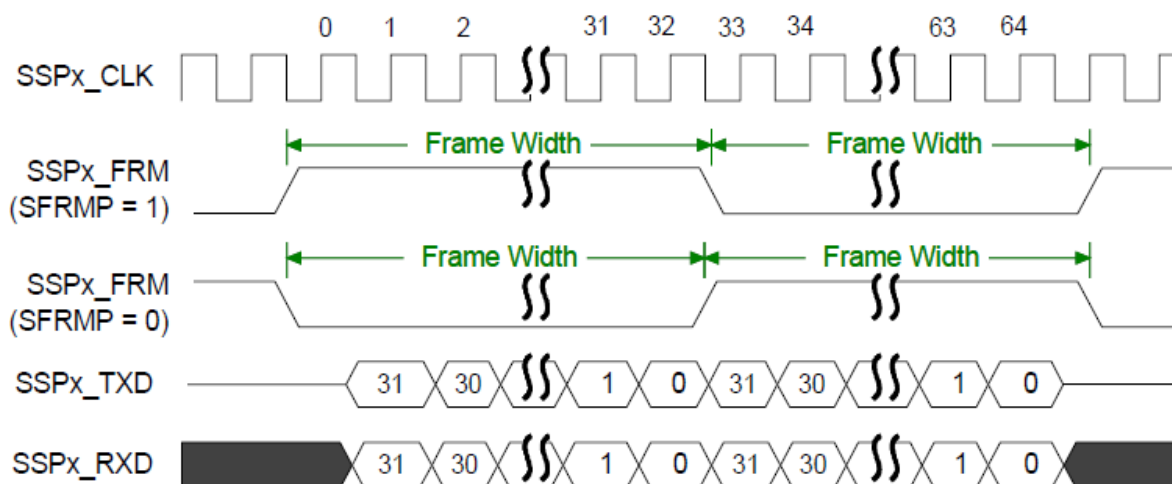


“Normal” Mode

The following bit fields must be configured for normal I²S mode as shown in Figure 79:

- SSP_SSCR0[EDSS] = 0b1 (32-bit data)
- SSP_SSCR0[DSS] = 0b1111 (32-bit data)
- SSP_SSCR0[FRF] = 0b11 (PSP format)
- SSP_SSCR0[FRDC] = 0b01 (Number of Time Slots+1)
- SSP_SSCR1[SCLKDIR] = 0b0 (SSP port is master of SSPx_CLK)
- SSP_SSCR1[SFRMDIR] = 0b0 (SSP port is master of SSPx_FRM)
- SSP_SSPSP[SFRMWDTH] = 0b100000 (Frame Width - 32 SSPx_CLK cycles)
- SSP_SSPSP[SFRMP] = 0b0 (Frame Polarity - Low)

- SSP_SSPSP[FSRT] = 0b1 (Frame Sync Timing - Delay audio data 1 SSPx_CLK cycle after SSPx_FRM transition)
- SSP_SSPSP[SCMODE] = 0x0 (Data driven on falling edge and sampled on rising)
- SSP_SSPSP[DMYSTOP] = 0b00 (Extended Dummy Stop)
- SSP_SSPSP[EDMYSTOP] = 0b000 (Extended Dummy Stop)
- SSP_SSTSA[TTSA] = 0x3 (Transmit Active Time Slots)
- SSP_SSRSA[RTSA] = 0x3 (Receive Active Time Slots)

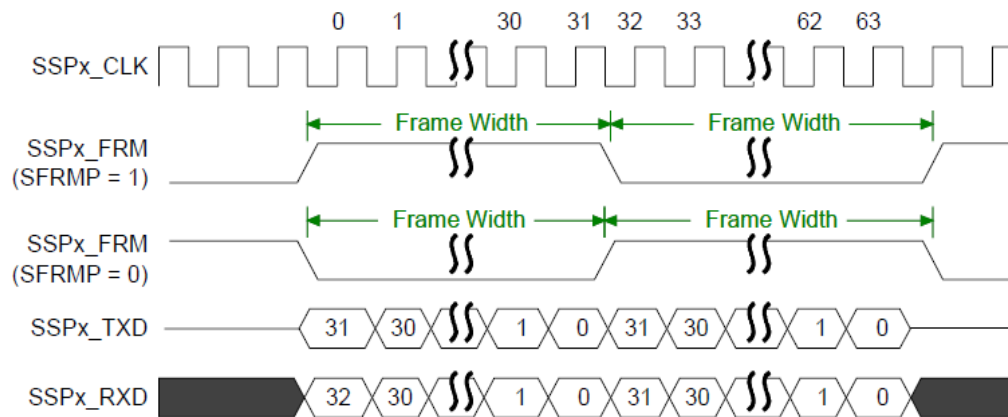
Figure 79: Normal I²S Format


“MSB-justified” Mode

The following bit fields must be configured for MSB-Justified I²S mode as shown in [Figure 80](#):

- SSP_SSCR0[EDSS] = 0b1 (32-bit data)
- SSP_SSCR0[DSS] = 0b1111 (32-bit data)
- SSP_SSCR0[FRF] = 0b11 (PSP format)
- SSP_SSCR0[FRDC] = 0b01 (Number of Time Slots+1)
- SSP_SSCR1[SCLKDIR] = 0b0 (SSP port is master of SSPx_CLK)
- SSP_SSCR1[SFRMDIR] = 0b0 (SSP port is master of SSPx_FRM)
- SSP_SSPSP[SFRMWDTH] = 0b100000 (Frame Width - 32 SSPx_CLK cycles)
- SSP_SSPSP[SFRMP] = 0b0 (Frame Polarity - Low)
- SSP_SSPSP[FSRT] = 0b0 (Frame Sync Timing - Audio data aligned with SSPx_FRM)
- SSP_SSPSP[SCMODE] = 0x0 (Data driven on falling edge and sampled on rising)
- SSP_SSPSP[EDMYSTOP] = 0b000 (Extended Dummy Stop)
- SSP_SSTSA[TTSA] = 0x3 (Transmit Active Time Slots)
- SSP_SSRSA[RTSA] = 0x3 (Receive Active Time Slots)

Figure 80: MSB-Justified I²S Format



20.5 Register Descriptions

A detailed description of SSP registers is located in Appendix A.



88MC200 Microcontroller Datasheet

21 Analog Digital Converter (ADC)

21.1 Overview

The Marvell 88MC200 microcontroller integrates two identical ADCs (ADC0 and ADC1) which can be programmed separately. The ADC is a second-order sigma-delta converter with up to 16-bit resolution. It includes an analog multiplexer (AMUX) and a programmable gain amplifier (PGA) with as many as eight individually configurable channels, a reference voltage generator, and a digital filter. The conversion results can be written to memory through DMA. Several modes of operation are available for the ADC.

21.2 Features

The main features of the ADC are as follows:

- Selectable decimation rates with also set the effective resolution (10 to 16 bits)
- Throughput rate as fast as 4 μ s (250 kHz)
- Single-ended and differential conversions from 8 external and 6 internal sources
- PGA setting support: 2x, 1x and 0.5x
- Selectable reference voltage (Vref)
 - Internal reference 1.2V (Vref_12)
 - Vref_18
 - External reference (do not exceed 1.8V)
- Input voltage ranges (differential)
 - -Vref/PGA to + Vref/PGA
 - Do not exceed VBAT voltage level
 - Do not exceed VDD_IOx_y voltage level
- Offset and gain calibration
- Embedded temperature sensor with internal or external diode options
- DAC dual inputs
- Interrupt generation and/or DMA request
- Internal GPT trigger on ADC conversion
- Battery measurement capability

21.3 External Signal Description

The external interface of the ADC module is mainly given by the power and analog pads. The ADC clock is provided by one of the clock sources: MAINXTAL, RC32M, and AUPLL. The analog unit uses ADC clock divided-down clock (divided by 4/8/16/32). The access to the configuration and data registers of the ADC is provided through an APB. The digital unit internal to the ADC runs with the APB bus clock. See [Table 52](#) for details.

Table 52: ADC Module External Interface Signals

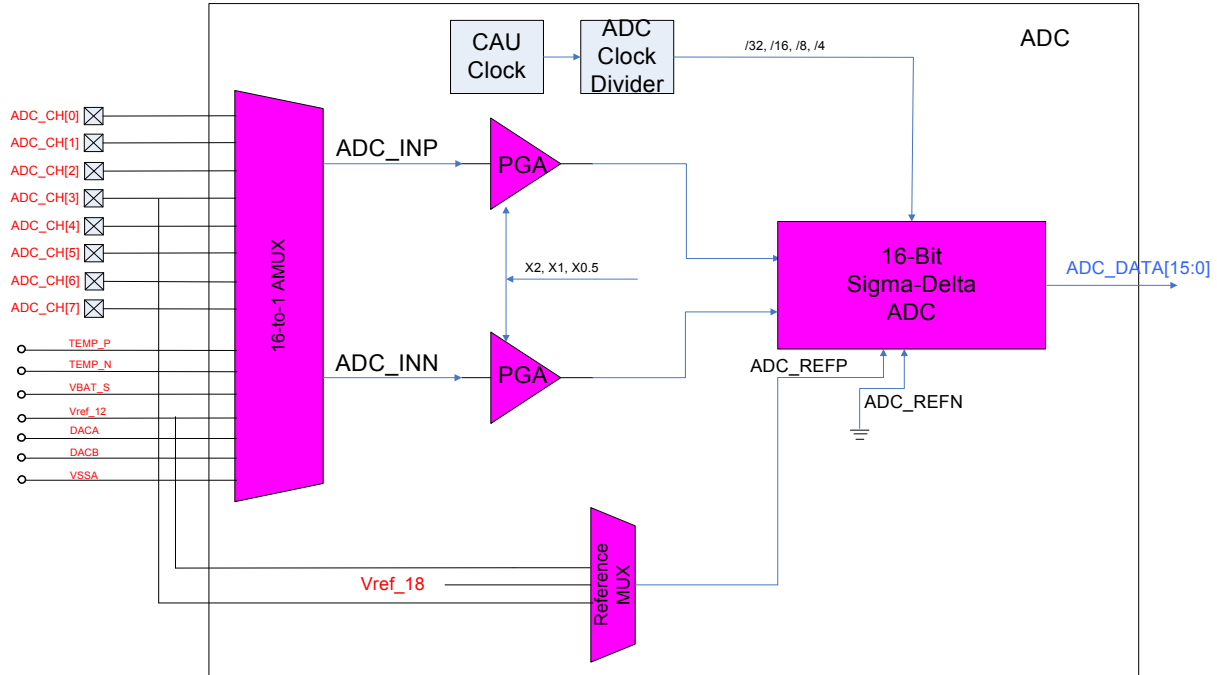
Pin Name	Type	Default Value	Source/ Destination	Function
ADC0_CH [7:0]	AI	N/A	From GPIO to ADC0	External analog inputs from GPIO. ADC0_CH [7]: GPIO_0 ADC0_CH [6]: GPIO_1 ADC0_CH [5]: GPIO_2 ADC0_CH [4]: GPIO_3 ADC0_CH [3]: GPIO_4 or External voltage reference ADC0_CH [2]: GPIO_5 ADC0_CH [1]: GPIO_6 or External diode negative side ADC0_CH [0]: GPIO_7 or External diode positive side
ADC1_CH [3:0]	AI	N/A	From GPIO to ADC1	External analog inputs from GPIO. ADC1_CH [3]: GPIO_11 or External voltage reference ADC1_CH [2]: GPIO_10 ADC1_CH [1]: GPIO_9 or External diode negative side ADC1_CH [0]: GPIO_8 or External diode positive side

21.4 ADC Functional Description

21.4.1 ADC Block Diagram

See [Figure 81](#) for the ADC block diagram.

Figure 81: ADC Block Diagram



21.4.2 ADC On-Off Control and Conversion Trigger

The ADC can be directly reset by system reset or ADC.CLKRST.SOFT_RST bit. The ADC is powered up by setting the ADC.PWR.GLOBAL_EN bit to 1'b1, and it is fully powered down when this bit is set to 1'b0.

After the ADC is powered up, the data conversion can be activated by writing a 1'b1 to the ADC.CMD.CONV_START bit if the ADC is in software control mode (ADC.CMD.TRIGGER_EN = 1'b0).

The actual conversion starts after ADC wakes up (TWARM) from the Powerdown mode where TWARM is 16μs by default. The conversions can be stopped by writing a 1'b0 to the ADC.CMD.CONV_START bit.

The ADC.STATUS.ACT bit is set to high when the ADC is actively converting or it is reset to low when conversions have stopped.

21.4.3 ADC Input

Each ADC module can support as many as eight external inputs. The actual input channels depend on the package types. Refer to the pinmux function for details.

Eight external inputs can be selected as eight different single-ended inputs or four differential inputs. In addition, it is possible to select six single-ended internal inputs. The available selections are given in the ADC.ANA.SINGLEDIFF and ADC.ANA.AMUX_SEL[3:0] bits.

Table 53 shows the various ADC input configurations.

Table 53: ADC Input Configurations

SINGLEDIFF/ AMUX_SEL[3:0]	ADC Positive Input	ADC Negative Input	Description
0/0000	ADC_CH[0]	VSSA	External single-ended or Temperature sensor (external diode)
0/0001	ADC_CH [1]	VSSA	External single-ended
0/0010	ADC_CH [2]	VSSA	External single-ended
0/0011	ADC_CH [3]	VSSA	External single-ended
0/0100	ADC_CH [4]	VSSA	External single-ended
0/0101	ADC_CH [5]	VSSA	External single-ended
0/0110	ADC_CH [6]	VSSA	External single-ended
0/0111	ADC_CH [7]	VSSA	External single-ended
0/1000	VBAT_S	VSSA	Internal single-ended (nominal 1/3 VBAT)
0/1001	Vref_12	VSSA	Internal single-ended (internal reference 1.2V)
0/1010	DACA	VSSA	Internal single-ended (DACA internal output)
0/1011	DACB	VSSA	Internal single-ended (DACB internal output)
0/1100	VSSA	VSSA	Internal single-ended (internal analog ground)
0/1101-0/1110	Reserved	Reserved	Reserved
0/1111	TEMP_P	VSSA	Temperature Sensor (internal diode)
1/0000	ADC_CH [0]	ADC_CH [1]	External differential or Temperature sensor (external diode)
1/0001	ADC_CH [2]	ADC_CH [3]	External differential
1/0010	ADC_CH [4]	ADC_CH [5]	External differential
1/0011	ADC_CH [6]	ADC_CH [7]	External differential
1/0100	DACA	DACB	Internal differential
1/1111	TEMP_P	TEMP_N	Temperature sensor (internal diode)
1/0101-1/1110	Reserved	Reserved	Reserved

21.4.4 Input Range

When configured to single-ended input, the voltage sampled is the difference between the input channel and VSSA:

$$\Delta V \text{ (differential voltage)} = VIN\_CH \text{ (input channel)} - VSSA$$

When configured to differential input, the voltage sampled is the difference between the odd and even channels:

$$\Delta V \text{ (differential voltage)} = VIN\_EVEN \text{ (even channel)} - VIN\_ODD \text{ (odd channel)}$$

Table 54 shows the detailed input range.

The input voltage for each external channel must be positive and cannot exceed the VBAT voltage level and VDD_IOx_y voltage level.

Table 54: Detailed Input Range

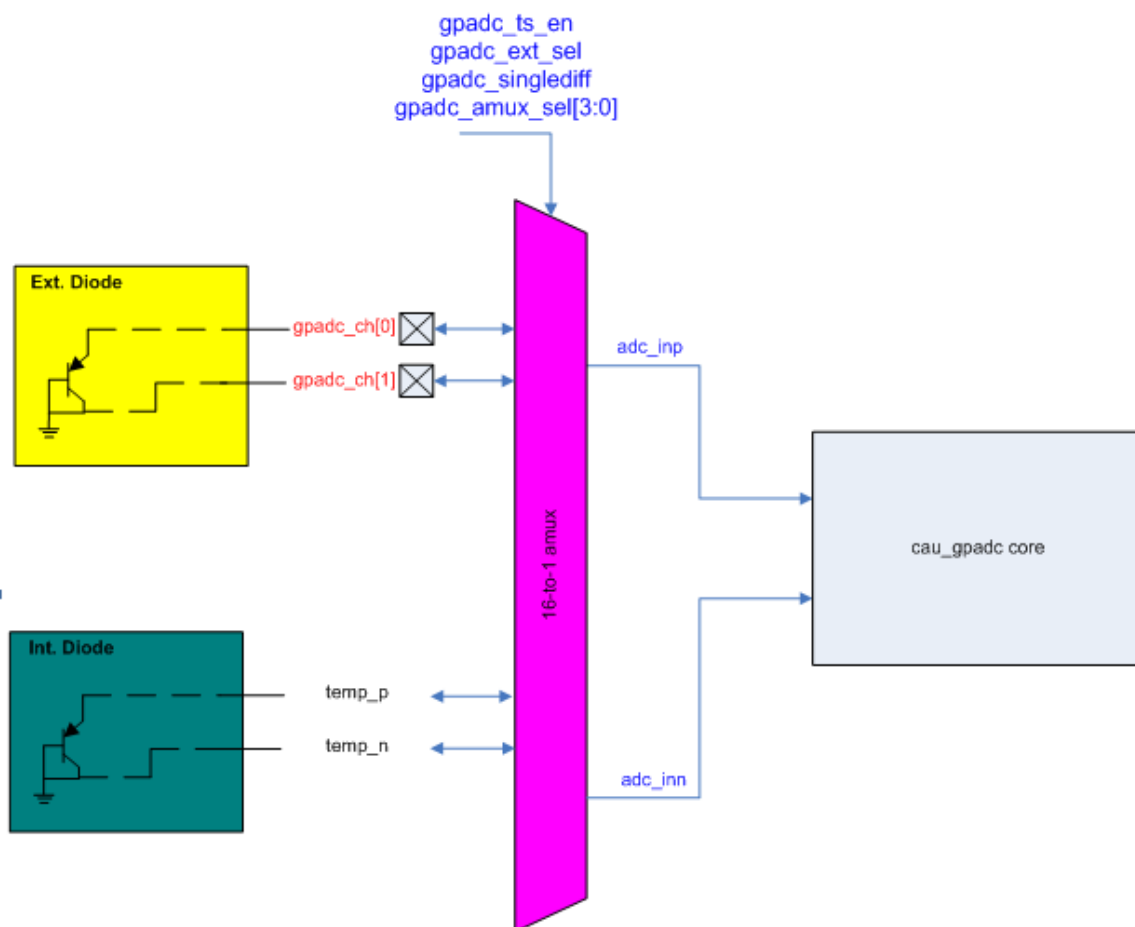
Single-ended or Differential input (ADC.ANA.SINGLEDIFF bit)	PGA Setting (ADC.ANA.PGA[1:0] bits)	Input Range (ΔV)
Single-ended	0.5	0 to 2*Vref
Single-ended	1	0 to Vref
Single-ended	2	0 to 0.5*Vref
Differential	0.5	-2*Vref to 2*Vref
Differential	1	-Vref to Vref
Differential	2	-0.5*Vref to 0.5*Vref

21.4.5 Temperature Measurement

The on-chip temperature sensor is used either to provide an absolute measurement of the device temperature or to detect changes in the ambient temperature (refer to Figure 82). The emitter and the collector/base of a PNP can be selected as two differential inputs of the ADC for temperature measurement. To do so, set the ADC.ANA.TS_EN bit to 1'b1.

Users have the option to measure the off-chip temperature by connecting an external diode (for example 2N3906) onto GPIO input pins (ADC_CH[0]- ADC_CH [1]) with ADC.ANA.EXT_SEL = 1'b1, or to monitor the on-chip temperature by using an embedded PNP with ADC.ANA.EXT_SEL = 1'b0.

Figure 82: ADC Temperature Sensor Mode with External Diode



By selecting internal voltage reference 1.2V (`Vref_12`), 14-bit ADC accuracy and by measuring the internal temperature sensor, the temperature is calculated according to the following formula:

$$T_{\text{meas}} (\text{in } ^\circ\text{C}) = (\text{ADC.RESULT.DATA}[15:0] - \text{TS\_OFFSET}) / \text{TS\_GAIN}$$

Equation Notes:

1. `ADC.RESULT.DATA` is denoted as signed 16 bits
2. `TS_OFFSET` and `TS_GAIN` are by default equal to:
 - For internal sensor: `TS_OFFSET` = 458; `TS_GAIN` = 1.7

21.4.6 ADC Reference Voltage

`ADC.ANA.VREF_SEL` is used to select the reference voltage. Change the reference voltage only when no conversion is running.

The positive reference voltage for analog-to-digital conversions is selectable as either the internal reference 1.2V (Vref_12), Vref_18, or external reference applied to the GPIO pin (GPIO_4 for ADC0 or GPIO_11 for ADC1, shared with ADC_CH[3]). The external reference should not exceed 1.8V.

21.4.7 ADC Throughput and Resolution

When the ADC clock is 32 MHz, through programming ADC.CLKRST.INT_CLK_DIV[4:0] and ADC.ANA.OSR[1:0], the ADC throughput and resolution is listed in [Table 55](#).

Table 55: ADC Conversion Time and Throughput Rate Lookup Table

INT_CLK_DIV [4:0]	OSR [1:0]	32 MHz Main Clock		Significant Bit
		One-Shot latency	Throughput Rate	
11111 (divide-by-32)	00	96 μ s	31.2ksps	10
	01	192 μ s	15.6ksps	12
	10	384 μ s	7.8ksps	14
	11	768 μ s	3.9ksps	16
01111 (divide-by-16)	00	48 μ s	62.5ksps	10
	01	96 μ s	31.2ksps	12
	10	192 μ s	15.6ksps	14
	11	384 μ s	7.8ksps	16
00111 (divide-by-8)	00	24 μ s	125ksps	10
	01	48 μ s	62.5ksps	12
	10	96 μ s	31.2ksps	14
	11	192 μ s	15.6ksps	16
00011 (divide-by-4)	00	12 μ s	250ksps	10
	01	24 μ s	125ksps	12
	10	48 μ s	62.5ksps	14
	11	96 μ s	31.2ksps	16

21.4.8 ADC Conversion Results

The digital conversion result is represented in 2's complement form (see [Table 56](#), [Table 57](#), [Table 58](#), and [Table 59](#)). The digital conversion result is available in ADC.RESULT.DATA[15:0] when ADC.IRSR.RDY is set to 1'b1.

Table 56: ADC Conversion Result Format (PGA=1 and OSR[1:0]=2'b11)

$\Delta V / V_{ref}$	Results	
	Binary	Hex Value
1	0111111111111111	7FFF
0.5	0011111111111111	3FFF
1/32768	0000000000000001	0001

Table 56: ADC Conversion Result Format (PGA=1 and OSR[1:0]=2'b11)

$\Delta V / V_{ref}$	Results	
	Binary	Hex Value
0	0000000000000000	0000
-1/32768	1111111111111111	FFFF
-0.5	1011111111111111	BFFF
-1	1000000000000000	8000

Table 57: ADC Conversion Result Format (PGA=1 and OSR[1:0]=2'b10)

$\Delta V / V_{ref}$	Results	
	Binary	Hex Value
1	0001111111111111	1FFF
0.5	0000111111111111	0FFF
1/8192	0000000000000001	0001
0	0000000000000000	0000
-1/8192	1111111111111111	FFFF
-0.5	1110111111111111	EFFF
-1	1110000000000000	E000

Table 58: ADC Conversion Result Format (PGA=1 and OSR[1:0]=2'b01)

$\Delta V / V_{ref}$	Results	
	Binary	Hex Value
1	0000111111111111	07FF
0.5	0000011111111111	03FF
1/2048	0000000000000001	0001
0	0000000000000000	0000
-1/2048	1111111111111111	FFFF
-0.5	1111011111111111	FCFF
-1	1111000000000000	F800

Table 59: ADC Conversion Result Format (PGA=1 and OSR[1:0]=2'b00)

$\Delta V / V_{ref}$	Results	
	Binary	Hex Value
1	0000001111111111	01FF
0.5	0000000111111111	00FF

Table 59: ADC Conversion Result Format (PGA=1 and OSR[1:0]=2'b00)

$\Delta V / V_{ref}$	Results	
	Binary	Hex Value
1/512	0000000000000001	0001
0	0000000000000000	0000
-1/512	1111111111111111	FFFF
-0.5	1111111011111111	FEFF
-1	1111111000000000	FE00

21.4.9 ADC Interrupts

The ADC outputs a single active high-level interrupt signal when any one of the following exceptions is pending:

- A single conversion has completed and 16-bit final data in ADC.RESULT.DATA[15:0] is ready for reading.
- An overflow occurred in the self-offset and system-offset calibration processes.
- An overflow occurred in the gain calibration process.
- An overflow occurred in the digital filtering process.
- A data transfer error occurred in the DMA control process.

ADC.ISR, the interrupt status register, is read only. Every non-reserved bit in this register represents one exception. A bit read as 1 means the corresponding exception is pending, the masked exception is not captured in this register.

ADC.IMR, the interrupt mask register, is readable and writable. This register is used to mask off unused exceptions. When a bit is set to 1, the corresponding exception is masked off and it does not trigger the interrupt signal. By default, all the non-reserved bits should be 1.

ADC.IRSR, the interrupt raw status register, is read-only. All the pending exceptions are captured in this register regardless of the values in the interrupt mask register.

ADC.ICR, the interrupt clear register, is write only. Write a 1 to a bit to clear the corresponding pending exception.

21.4.10 ADC Calibration

Sampling of internal connections VSSA or Vref_12 allows for self/system offset and gain calibration of the ADC to correct error due to process and temperature variations where absolute accuracy is important. This calibration must be done individually for each reference used and each ADC decimation rate. The ADC.OFF_CAL and ADC.GAIN_CAL registers contain 3 register fields for calibrating both offset and gain.

Reset initializes the offset to zero (ADC.OFF_CAL = 0) and gain factor to 1 (ADC.GAIN_CAL.GAIN_CAL = 0[15:0] = x8000).

Table 60 shows the equations used to calculate the gain and offset correction values.

Table 60: Equations for Gain and Offset Correction

Calibration Type	Calibration Correction Value (PGA = 1)
Self Offset, input buffer disabled. Vref = Vref_12	$N_{(VSSA-VSSA)}$

Table 60: Equations for Gain and Offset Correction (Continued)

Calibration Type	Calibration Correction Value (PGA = 1)
Self Gain, input buffer disabled. Vref = Vref_12	$0x7FFF/N_{(1.2-VSSA)}$
System Gain, input buffer disabled. Vref = external 1.25V	$0x7AE1/N_{(1.2-VSSA)}$

Equation Notes:

- All N are 16-bit 2's complement numbers.
- $N_{(VSSA-VSSA)}$ is a differential sampling of 0 with input positive and negative sides connected to VSSA when offset calibration mode and it yields a 2's complement value close to 0.
- $N_{1.2-VSSA}$ is a sampling of Vref_12 and it yields a 2's complement value close to 0x7FFF in OSR[1:0] = 2'b11 mode.
- $N_{1.2-VSSA}$ is a sampling of Vref_12 with external accurate voltage reference (1.25V from ADC_CH[3]) and it yields a 2's complement value close to 0x7AE1 in OSR[1:0] = 2'b11 mode.

21.4.11 DMA Request

With ADC.DMAR.DMA_EN to 1'b1, the ADC sends out a DMA request every time a conversion from any channel has completed. This request is automatically cleared when the corresponding result register is read and a DMA finish signal sent from DMA has received (optional).

21.4.12 Battery Monitor

The internal power supply monitor allows the battery voltage to be measured by programming ADC.ANA.SINGLEDIFF to 1'b0 and ADC.ANA.AMUX_SEL[3:0] to 4'b1000. This monitoring is achieved with a potential divider that reduces the voltage by a factor of 0.33 (nominal), allowing it to fall inside the ADC input range. After the battery voltage measurement is finished, ADC.ANA.AMUX_SEL[3:0] should be assigned another value other than 4'b1000 to disable the resistor chain that performs the voltage reduction, thereby avoiding a continuous drain on the power supply.

21.4.13 External Trigger from GPT

It is also possible to trigger conversions from an event. Once the ADC is powered up by setting the ADC.PWR.GLOBAL_EN bit to 1, the data conversion can be activated by external trigger source (GPT0 for ADC0 and GPT1 for ADC1) by writing 1 to the ADC.CMD.TRIGGER_EN bit to enable the Event Trigger mode.

21.5 Register Description

A detailed description of the ADC registers is located in Appendix A.

22 Digital Analog Converter (DAC)

22.1 Overview

The 88MC200 integrates a register string-based DAC with true 10-bit resolution. It includes two channels, every channel can output single ended signal or combine two channels to output differential signal.

22.2 Features

The main features of the 88MC200 DAC are:

- 10-bit resolution
- Throughput rate as fast as 2 μ s (500 kHz)
- Capable of directly driving a piezo speaker with 1000-ohm load
- Flexible waveform generator (sinusoidal, triangle, noise, etc.) at various frequency range
- Selectable output mode: single-ended or differential
- Internal or external reference voltage
- Interrupt generation and/or DMA request
- Three selectable output ranges
- Supports event trigger from GPT or GPIO

22.3 External Signal Description

The DAC External signal interface is listed in [Table 61](#).

Table 61: DAC External Signal Description

Pin Name	Type	Default Value	Source/ Destination	Function
DAC_REF	AI	N/A	From GPIO to DAC	External voltage reference from GPIO_10
DACA	AO	N/A	From DAC to GPIO	DAC channel A output to GPIO_4
DACB	AO	N/A	From DAC to GPIO	DAC channel B output to GPIO_11

22.4 DAC Configuration

The 88MC200 DAC can support two independent single-ended channels, or one differential channel, set by DAC.BCTRL.B_WAVE[1:0] register bits. In single-ended mode, each channel output can be sent to pad by setting DAC.xCTRL.x_IO_EN register bit (x can be A or B). Each channel can be powered on by the DAC.xCTRL.x_EN register bit (x can be A or B). Output voltage is calculated using [Table 62](#).

Table 62: Output Voltage Calculation Formula

A_Range[1:0]	REF_SEL	Output
00	0	$0.20 + (0.72 * \text{input data} / 1023)$
01/10	0	$0.24 + (1.14 * \text{input data} / 1023)$
11	0	$0.22 + (1.6 * \text{input data} / 1023)$
00	1	$0.1 * V_{\text{ref\_ext}}^1 + (0.4 * V_{\text{ref\_ext}} * \text{input data} / 1023)$
01/10	1	$0.1125 * V_{\text{ref\_ext}} + (0.6 * V_{\text{ref\_ext}} * \text{input data} / 1023)$
11	1	$0.1 * V_{\text{ref\_ext}} + (0.8 * V_{\text{ref\_ext}} * \text{input data} / 1023)$

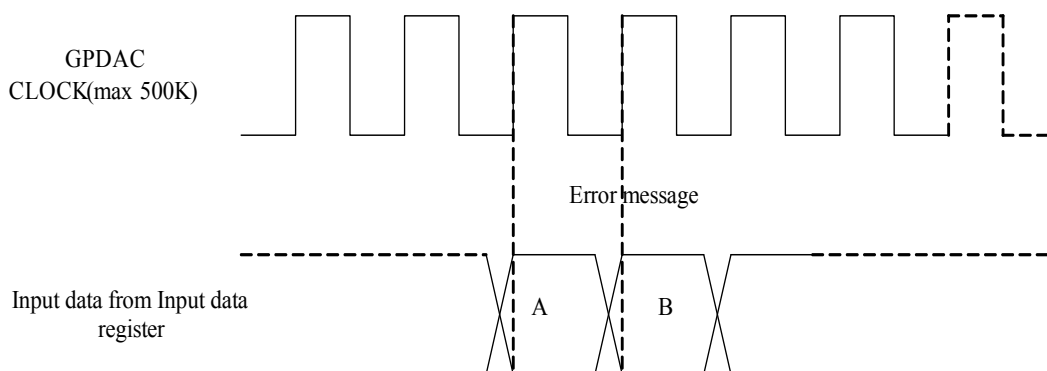
NOTE: ¹ Vref_ext is the voltage value of DAC_REF.

- DAC.ACTRL.A_RANGE[1:0] bits impact the output range of both channel A and B simultaneously
- The DAC can be operated in the differential mode by setting DAC.BCTRL.B_WAVE[1:0] to 2'b11
- The DAC has an internal clock prescaler to divide the clock, set by the DAC.CLK.CLK_CTRL[1:0] bits
- Each DAC channel can be powered on by setting the corresponding DAC.xCTRL.x_EN bit to 1 (x can be A or B)

22.4.1 Synchronous Mode

Each DAC channel can operate in synchronous mode by setting the DAC.xCTRL.x_MODE register bit (x can be A or B). In this mode, each DAC channel has a timing requirement for input data refresh speed, as illustrated in [Figure 83](#).

Figure 83: Synchronous Mode



22.4.2 Asynchronous Mode

In this mode, the DAC works in passive mode, and does not have a time-sequence requirement for input data refreshing.

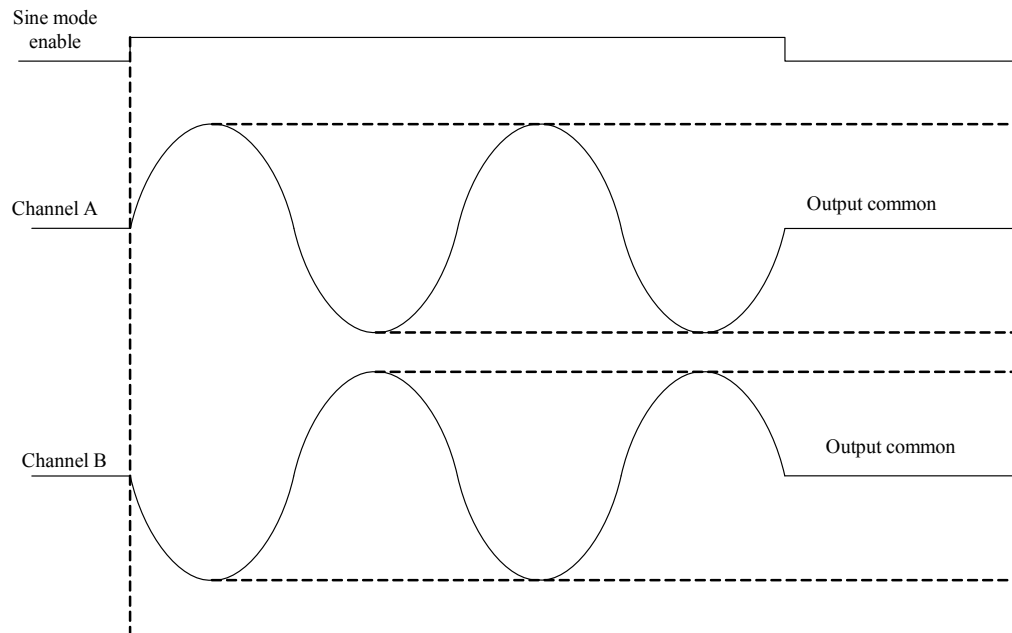
22.4.3 Sinusoidal Waveform Generation

This function is enabled by setting DAC.ACTRL.A_WAVE[1:0] register bits to 2'b10. The amplitude of the sine signal is controlled by the DAC.ACTRL.A_RANGE[1:0] register bits. Each period, starting at 0 degree consists of 16 samples and the frequency is given by the equation:

$$f_{\text{sine}} = f_{\text{clk}} / 16$$

The sine wave is output on Channel A. In differential mode, the sine wave is output on both channels (if two channels have been enabled), but inverted. See [Figure 84](#).

Figure 84: Sinusoidal Waveform Generation



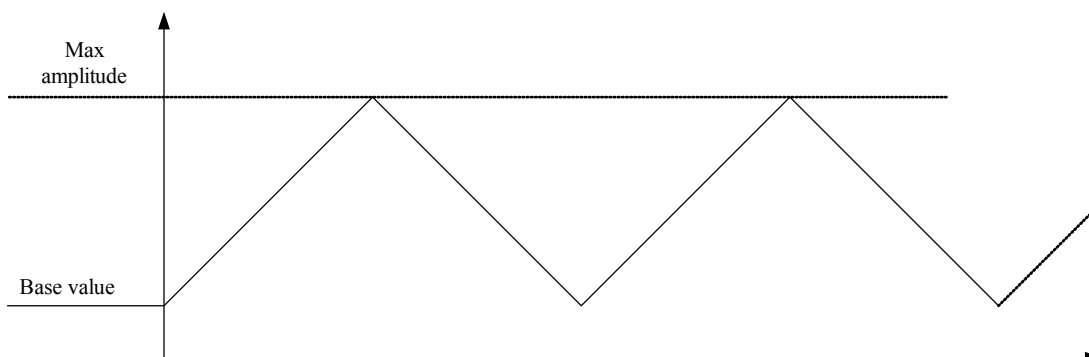
22.4.4 Triangle Waveform Generation

22.4.4.1 Up and Down Mode

- The amplitude of the triangle wave is configured through the DAC.ACTRL.A_RANGE[1:0] register bits.
- The triangle counter increases three clock cycles after each trigger event.
- The triangle counter increases while it is less than the maximum amplitude set by the DAC.ACTRL.A_TRIA_MAMP_SEL[3:0] register field.
- The increment step is defined by the DAC.ACTRL.A_TRIA_STEP_SEL[1:0] register.
- Once the configured amplitude is reached, the counter is decremented down to the base value defined by the DAC.ADATA register.

See [Figure 85](#).

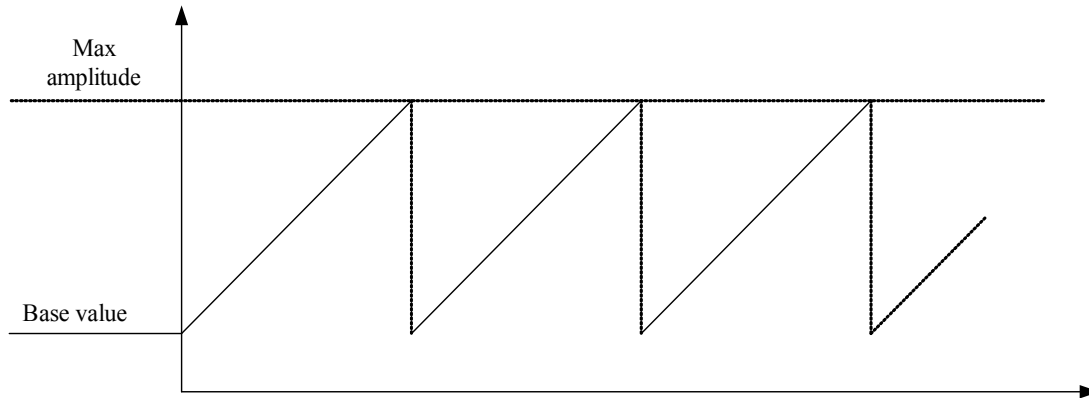
Figure 85: Full Triangle Generation Mode



22.4.4.2 Up Mode

This mode is set by the DAC.ACTRL.TRIA_HALF bit in the DAC.ACTRL register. The remaining control information with respect to setting the amplitude, maximum amplitude, increment step are configured using the DAC.ACTRL.A_RANGE[1:0], DAC.ACTRL.A_TRIA_MAMP_SEL[3:0] and DAC.ACTRL.A_TRIA_STEP_SEL[1:0] register fields. Refer to the Up and Down Mode section for a detailed description of how the register fields are used. The difference from the Up and Down mode is that once the configured amplitude is reached, the counter is directly down to the base value. See [Figure 86](#).

Figure 86: Half Triangle Generation Mode



22.4.5 Noise Generation

The DAC can generate pseudo noise.

22.4.6 DMA Request

Each DAC channel supports DMA data transfer. A DAC DMA request is generated each time when previous data conversion is complete and new data is requested to be loaded to DAC.xDATA.x_DATA[9:0] while the DAC.xCTRL.x_DEN register field (x can be A or B) is set to 1. If the DAC.xCTRL.x_DEN register field is set for both channel A and B, two DMA requests are generated.

22.4.7 Event Trigger from GPT or GPIO

Events from GPT or GPIO can trigger the reload of new data from DAC.xData.x_DATA (x can be A or B) to DAC for conversion. The DAC event trigger mode is activated by writing DAC.xCTRL.x_EN bit to 1 and DAC.xCTRL.x_TRIG_EN bit to 1 (x can be A or B).

Each DAC channel accepts up to 4 trigger sources: GPT2, GPT3, GPIO_44, and GPIO_45. DAC.xCTRL.x_TRIG_SEL[1:0] defines the appropriate trigger .

GPT match interrupt can generate the trigger event. In addition, the transition edge of selected external GPIO source can trigger the reload of new data. Rising edge, falling edge, or both edges can be selected by DAC.xCTRL.x_TRIG_TPY[1:0].

When there is no event occurred, the DAC continuously converts previous 10-bit data and hold the analog conversion result at the output.

When a trigger event is generated, a new 10- bit data block is loaded in the DAC and a new analog conversion result is presented at the output.

22.5 Registers Description

A detailed description of the DAC registers is located in Appendix A.



88MC200 Microcontroller Datasheet

23 Analog Comparator (ACOMP)

23.1 Overview

The 88MC200 microcontroller has two analog identical comparators, ACOMP0 and ACOMP1, which are designed to have true rail-to-rail inputs and operate over the full voltage range of the power supply VBAT. The comparator outputs are latched and can be used as interrupts.

23.1.1 Features

The main features of the analog comparator are as follows:

- Eight selectable external positive inputs
- Eight selectable external negative inputs
- Two selectable internal positive inputs
 - DACA output
 - DACB output
- Five selectable internal negative inputs
 - DACA output
 - DACB output
 - VBAT scaled by 4 selectable factors
 - Internal reference 1.2V (Vref_12)
 - VSSA
- Selectable positive and negative hysteresis between 0 and 70mV with 10mV step
- Selectable response time as fast as 110ns
- Interrupt generation on selectable edges (rising edge and/or falling edge) or levels.
- Extremely low power mode
- Configurable output when inactive
- Comparator output on GPIOs through alternate functionality, output inversion available

23.2 External Signal Description

The external interface signals of the ACOMP0/1 include power supply and analog inputs. Table [Table 63](#) lists the interface signals.

Table 63: ACOMP Module Interface Signals

Pin Name	Type	Default Value	Source/ Destination	Function
ACOMP0_IN [7:0]	AI	N/A	From GPIO to ACOMP	External analog inputs from GPIO. ACOMP_CH[0]: GPIO_7 ACOMP_CH[1]: GPIO_6 ACOMP_CH[2]: GPIO_5 ACOMP_CH[3]: GPIO_4 ACOMP_CH[4]: GPIO_3 ACOMP_CH[5]: GPIO_2 ACOMP_CH[6]: GPIO_1 ACOMP_CH[7]: GPIO_0
ACOMP0_GPIO_OUT	DO	N/A	From ACOMP to GPIO	ACOMP0 synchronized or asynchronous comparison output to GPIO_25/27/29
ACOMP1_GPIO_OUT	DO	N/A	From ACOMP to GPIO	ACOMP1 synchronized or asynchronous comparison output to GPIO_25/27/29
ACOMP0_EDGE_PULSE	DO	N/A	From ACOMP to GPIO	ACOMP0 edge detector output to GPIO_26/28/30
ACOMP1_EDGE_PULSE	DO	N/A	From ACOMP to GPIO	ACOMP1 edge detector output to GPIO_26/28/30

23.3 Functional Description

23.3.1 ACOMP0/1 Control Signals

The ACOMP.CTRL[x].POS_SEL[3:0] field and the ACOMP.CTRL[x].NEG_SEL[3:0] field select which signals are connected to the two inputs of the comparator.

23.3.1.1 Warmup Time

When the ACOMP.CTRL[x].EN bit is set from low to high, ACOMPx is turned on and compares the two analog inputs. The warm-up time of the comparator after turn-on is programmable in the ACOMP.CTRL[x].WARMTIME[1:0] field. When it is warmed up, the ACOMP.STATUS[x].ACT bit is set high.

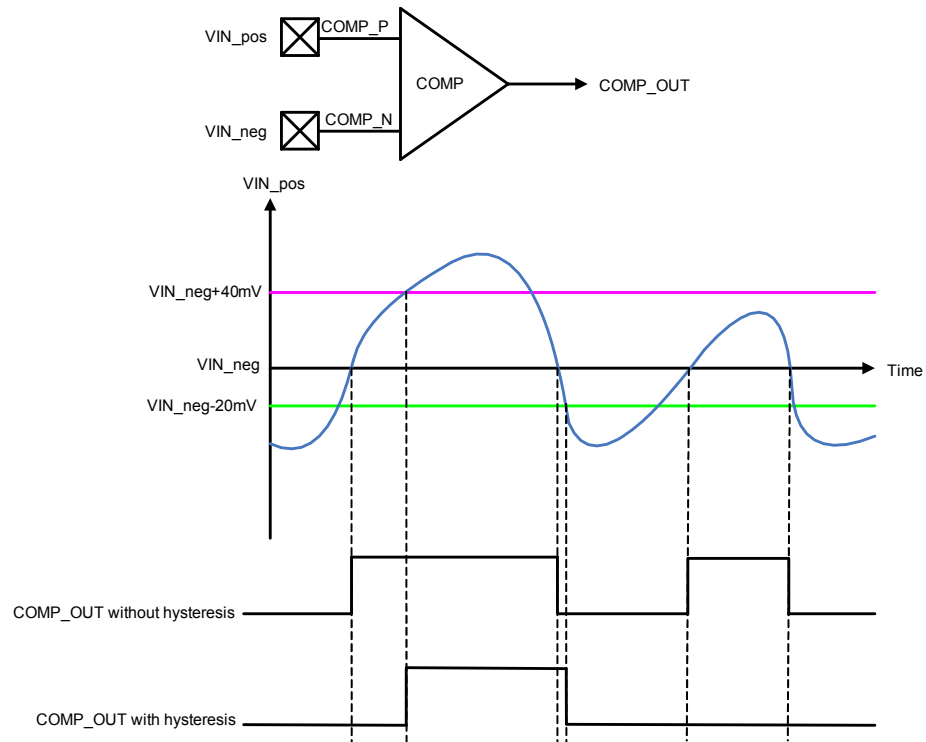
23.3.1.2 Response Time

When the voltage of input signals changes and triggers a polarity flip at the comparator output, the delay from input to output is the response time of the comparator. The response time is also programmable through the ACOMP.CTRL[x].BIAS_PROG[1:0] field.

23.3.1.3 Hysteresis

The programmable hysteresis in ACOMP0/1 can be used to filter input fluctuations due to noise, and only changes that are big enough to reach the hysteresis threshold trigger an output change, as illustrated in [Figure 87](#).

Figure 87: Comparator Hysteresis



The hysteresis voltage levels for the positive input and the negative input are set in the ACOMP.CTRL[x].HYST_SELN[2:0] and ACOMP.CTRL[x].HYST_SELN[2:0] field

23.3.2 Comparator Output

The outputs from ACOMP0/1 are available in ACOMP.STATUS[x].OUT, or as alternate functions to the GPIO pins. Set the ACOMP.ROUTE[x].PE bit to 1 to enable output to pin.

23.3.2.1 Asynchronous Comparison Output at Register

When comparator is enabled by ACOMP.CTRL[x].EN, real time comparator output is available in ACOMP.STATUS[x].OUT.

When comparator is disabled, comparator output in ACOMP.STATUS[x].OUT can be set by ACOMP.CTRL[x].INACT_VAL.

23.3.2.2 Synchronous/Asynchronous Comparison Output at GPIO

The comparator output at GPIO can be programmed to be synchronized with the main clock of the comparator, or asynchronized by setting ACOMP.ROUTE[x].OUTSEL.

When powered down, the output values can be set through register bit ACOMP.CTRL[x].INACT_VAL.

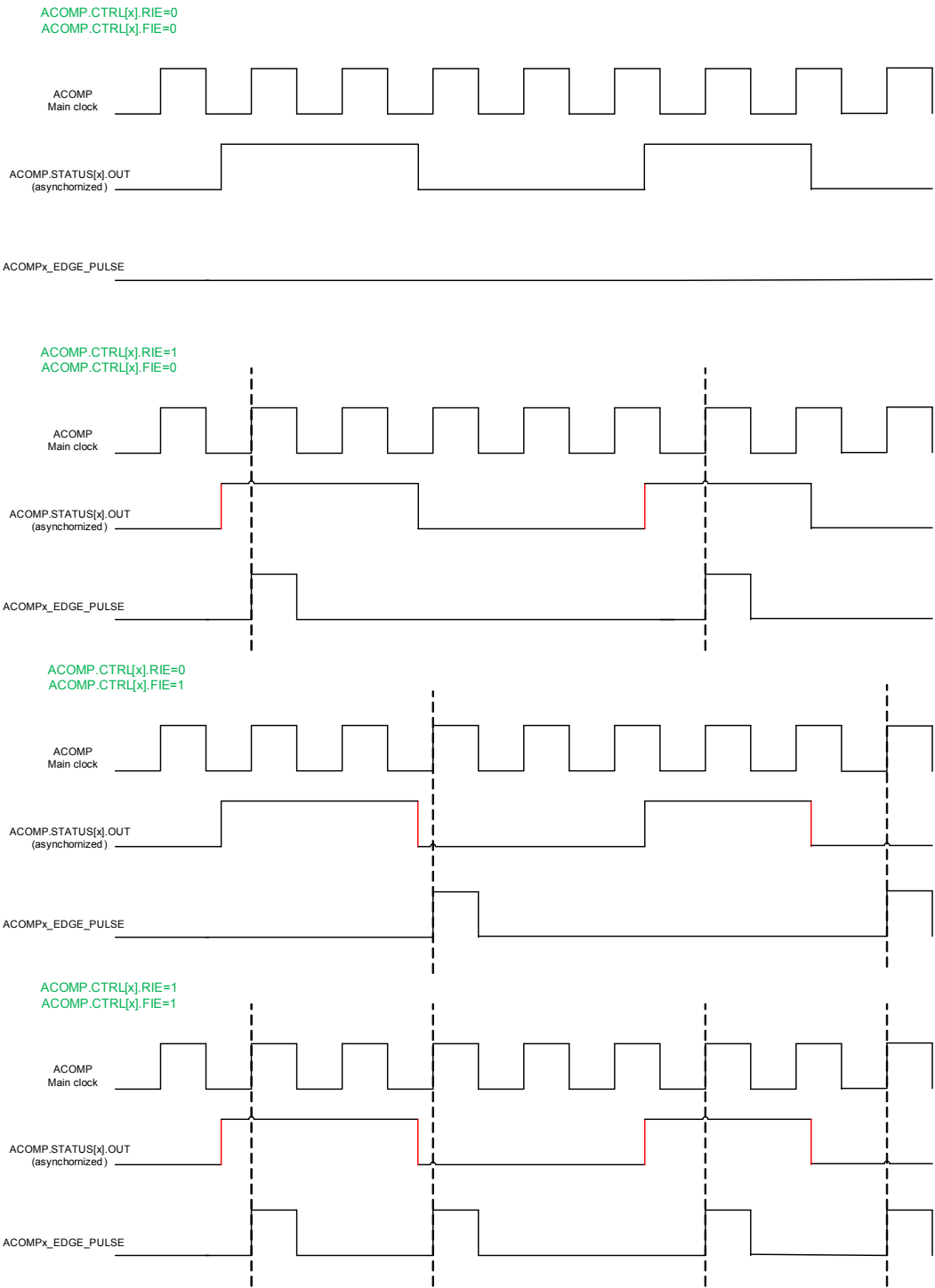
23.3.2.3 Comparison Output Inversion

The comparison output to GPIO can be inverted by the ACOMP.CTRL[x].GPIOINV bit.

23.3.3 Comparator Output Edge Detection

The comparator output edge detection can be enabled by selecting ACOMP.CTRL[x]. RIE (rising edge) or ACOMP.CTRL[x]. FIE (falling edge). The edge detection results are routed to GPIOs. See [Figure 89](#).

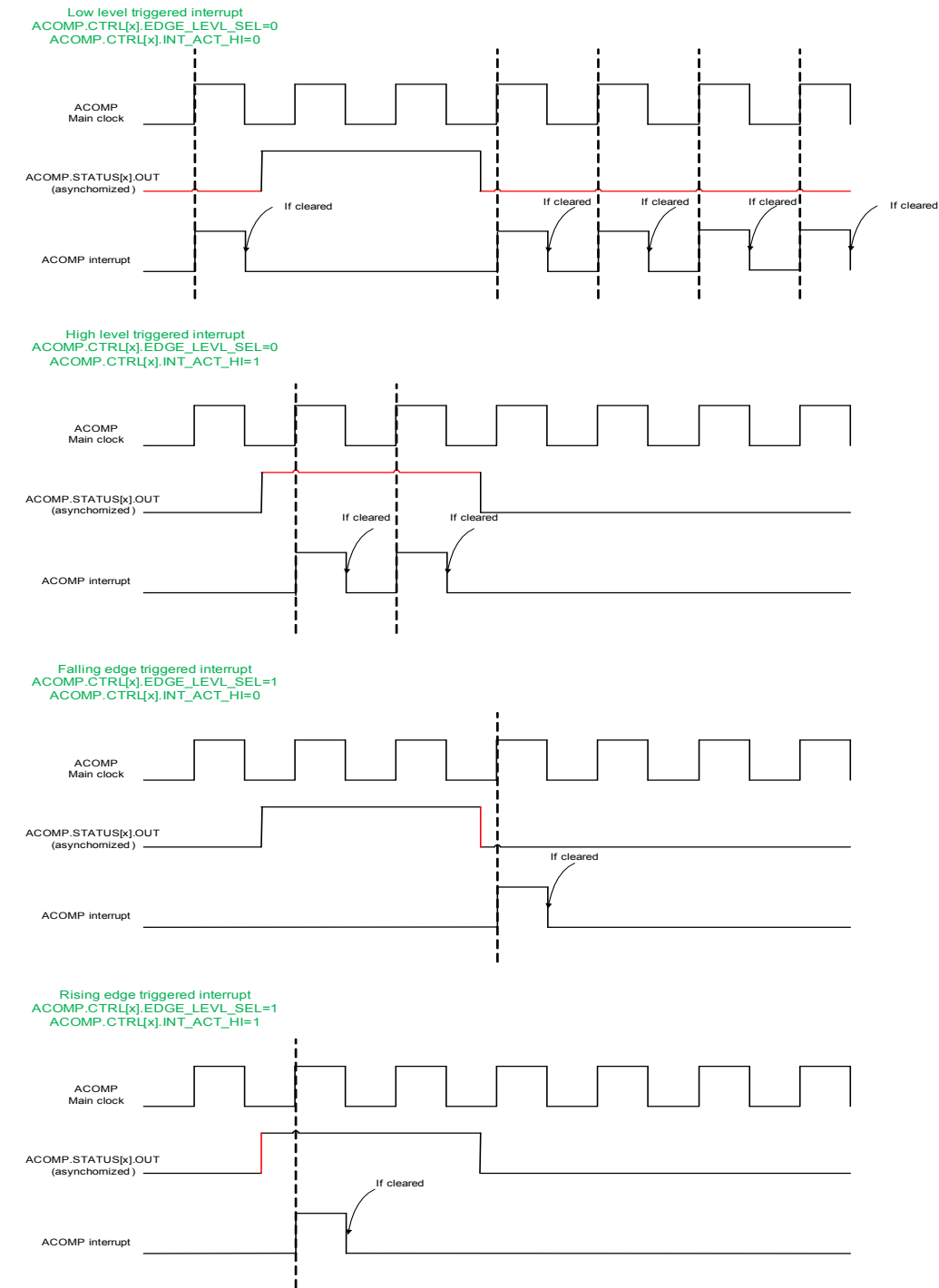
Figure 88: Comparator Output Edge Detection



23.3.4 Interrupt

An interrupt is generated upon detection of level or edge changes of ACOMP0/1 comparison results. Interrupt trigger type and active mode can be selected by ACOMP.CTRL[x].EDGE_LEVEL_SEL and ACOMP.CTRL[x].INT_ACT_HI. See [Figure 89](#).

Figure 89: Interrupt



23.4 Register Description

A detailed description of the ACOMP registers is located in Appendix A.

24 Boot ROM

24.1 Overview

Boot ROM is located in code space with address 0x0000_0000 to 0x0000_0xFFFF with a code size of 4 KB. After reset, the Cortex-M3 reads data from address 0x0000_0000 to set the MSP initial value and reads data from address 0x0000_0004 to set the PC initial value to allow the Boot ROM to take control of the chip. Once Boot ROM is running, it uploads user code from Flash or the UART port to the destination address then transfers the control to user code. When Boot ROM is running, it uses memory space 0x2001_0000 ~ 0x2001_046A for STACK and variables. Due to limited space, there is no vector table in Boot ROM and no interrupt service routine (ISR) in use.

The following features are implemented in this Boot ROM:

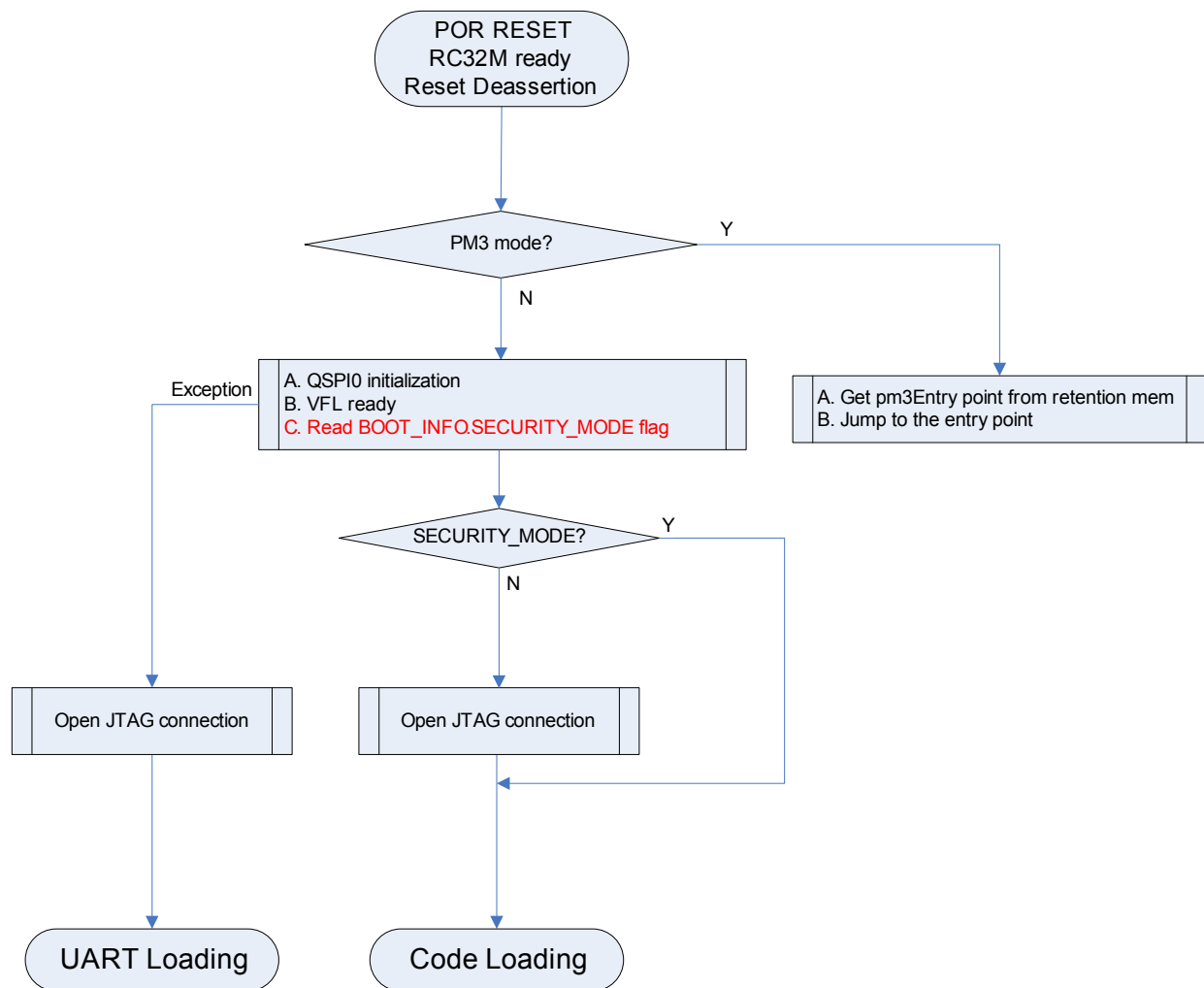
- Code loading interface
 - UART
 - Embedded flash
- Code loading
 - Configurable code loading interface by pin BOOT (GPIO_27)
 - Skip code loading from embedded flash if non-valid code in flash and switch to UART interface
 - Auto UART baud rate detection
 - Check for corruptions in image by CRC checksum
 - Only valid code which passes CRC check is loaded to RAM
 - Fast boot support waking up from PM3 mode

24.2 Boot ROM Flow Charts

Boot ROM is automatically activated by applying a reset. The value on the BOOT pin is latched on the de-assertion of reset, which is used to select the boot interface. Depending on the pin configuration used, either the Flash or UART is selected as the boot interface.

- Pin Boot 1 = Load code image to SRAM from Flash
- Pin Boot 0 = Load code image to SRAM from UART

Figure 90: Boot ROM Flow Chart 1



*Components in red indicate the access to embedded Flash

Figure 91: Boot ROM Flow Chart 2

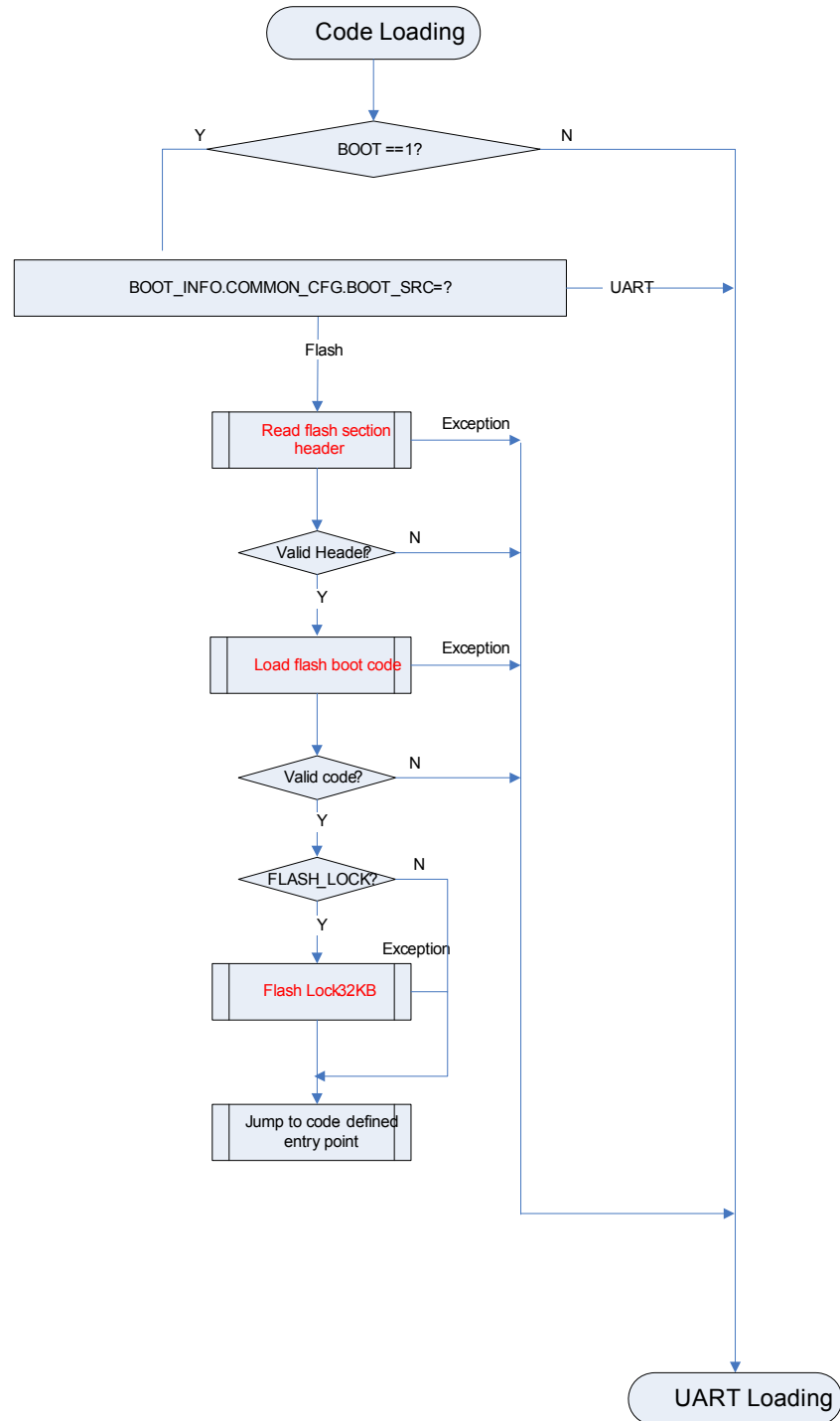
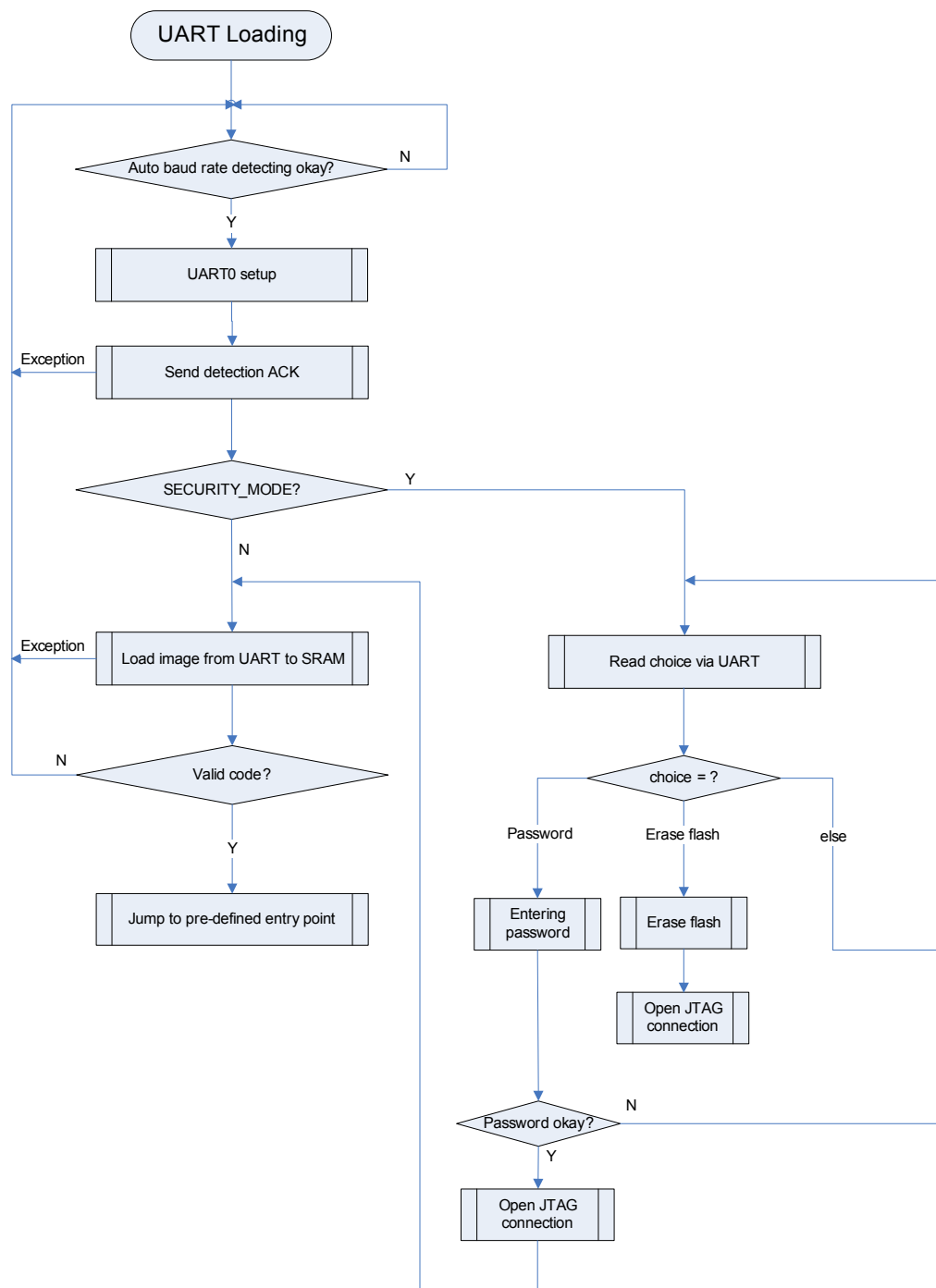


Figure 92: Boot ROM Flow Chart 3



24.2.1 Loading Code Through UART

(Pin BOOT = 0 at reset de-assertion)

3. QSPI0 initialization (switch on JTAG then go to Step 4 for any exception)
4. Read BOOT_INFO.SECURITY_MODE (switch on JTAG then go to Step 4 for any exception)
5. Switch JTAG per SECURITY_MODE
6. Auto baud-rate detection
7. Configure pin mux registers
8. Configure UART registers
9. Load code from UART if password matches BOOT_INFO.MAINPASSWD (go to Step 4 for any exception)
10. Jump to specified entry point

24.2.2 Loading Code Directly from Flash

1. QSPI0 initialization (switch on JTAG then go to Step 4 for any exception)
2. Read BOOT_INFO.SECURITY_MODE (switch on JTAG then go to Step 4 for any exception)
3. Switch JTAG per SECURITY_MODE
4. Read FLASH Loader Header (switch to UART code loading if any exception)
5. Load code from Flash address specified by FLASH Loader Header (switch to UART code loading if any exception)
6. Jump to specified entry point in FLASH Loader Header

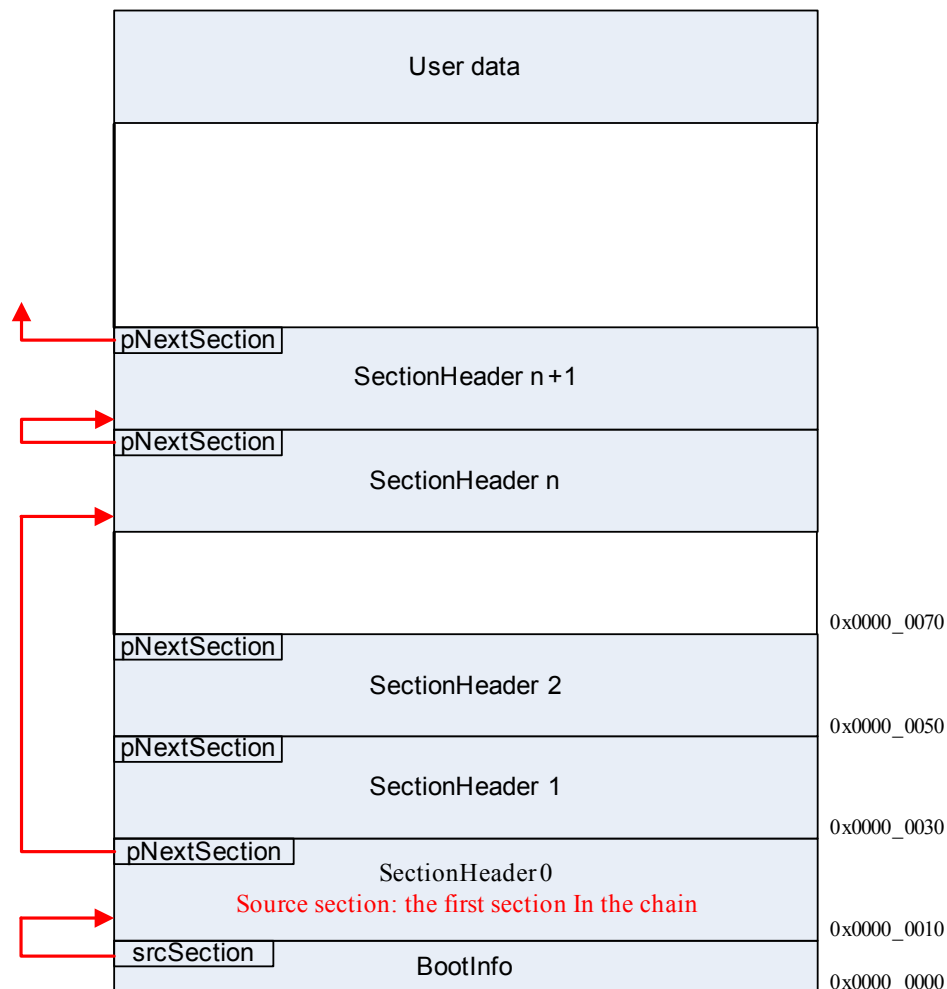
24.2.3 PM3 Wakeup

1. Wakeup from PM3 mode
2. Get 32-bit PM3 wakeup entry point from retention memory 0x480C0008
3. PC goes to the specified entry point

24.3 Flash Image Format

A Flash image consists of bootInfo (comprised of bootInfoHeader, flashBootHeader, spiBootHeader, usbBootHeader), code images and data images. BootInfo contains important information used during SoC boot. Code images include the primary firmware code (PFC, containing OS and driver), and possible customer applications. Data images include data segment used by system and that used by customers. With this version of BootROM, BootInfo is always at the beginning of the Flash region, and the primary firmware can be anywhere, just its start address should be appropriately defined in bootHeaders. On the other hand, the layout of other code images and data image are open to customization. An illustration of a flash image is shown in Figure 24-4.

Figure 93: Flash Image Memory Mapping



24.3.1 BootInfo/Section Header

Table 64: BootInfo Layout

Section Name	Address	Size	Field Name	Description
bootInfoHeader	0x00	32 bits	securityMode	Key 0x00010204 activates security mode
	0x04	32 bits	mainPasswd	Password used to quit security mode
	0x08	32 bits	commonCfg0	Common configuration 0
	0x0C	32 bits	commonCfg1	Common configuration 1

Table 64: BootInfo Layout (Continued)

Section Name	Address	Size	Field Name	Description
Section Header 0 (flash loader)	0x10	32 bits	codeSig	Code signature for CRC check
	0x14	32 bits	codeLen	Code length
	0x18	32 bits	flashStartAddr	Start address in FLASH to load PFC
	0x1C	32 bits	sramStartAddr	Start address in RAM to load PFC
	0x20	32 bits	sramEntryAddr	This field is valid only for the first section header in the section chain. Address of entry function
	0x24	32 bits	bootCfg0	Boot configuration 0
	0x28	32 bits	bootCfg1	Boot configuration 1
	0x2C	32 bits	N/A	Reserved, keep 0xFFFFFFFF

Table 65: Sub-Field in common Dfg0

Bit	Sub-Field Name	Default	Description
31	noWriteStatus	1	1: No change to flash status register. 0: Overwrite flash status register with statusBit0[1:0] and statusBit1[5:0]
30:29	statusBit0	2'b11	2'b11 or 2'b00: SRP[1:0]=2'b00 2'b10: SRP[1:0]=2'b10 2'b01: SRP[1:0]=2'b01
28:23	statusBit1	6'b111111	CMP: statusBit1[5] SEC: statusBit1[4] TB: statusBit1[3] BP[2:0]: statusBit1[2:0]
22:18	qspiPrescaler	5'b00010	QSPI clock prescaler
17	qspiPrescalerOff	1	1: use default prescaler (0x2) 0: use qspiPrescaler[4:0]
16:10	N/A	All 1s	Reserved
9:8	clockSel	2'b11	Set system clock source 2'b11: use RC32M for system clock 2'b10: use X32M for system clock 2'b01: use SFLL_200M for system clock, and SFLL reference clock is RC32M 2'b00: use SFLL_200M for system clock, and SFLL reference clock is X32M
7:0	srcSection	0x00	Decide which boot section header is to be applied 0x00: FLASH copy Others: reserved

Table 66: Sub-Field in common Cfg1

Bit	Sub-Field Name	Default	Description
31-0	N/A	All 1s	Reserved

Table 67: Sub-Field in bootCfg0

Bit	Sub-Field Name	Default	Description
31	emptyCfg	1	This field is valid only for the first section header in the section chain. 0: bootCfg1 is applied 1: bootCfg1 is ignored
30:16	N/A	All 1s	Reserved
15:0	pNextSection	0xFFFF	Next section index. Used to form section chain. 0xFFFF: end section Else: next section index

Table 68: Sub-Field for bootCfg1

Bit	Sub-Field Name	Default	Description
31:15	N/A	All 1s	Reserved
14:12	aesMode	3'b111	This field is valid only for the first section header in the section chain Decide which AES mode will be applied
11:9	crcMode	3'b111	This field is valid only for the first section header in the section chain Decide which CRC mode is to be applied 3'b000: CRC mode 0, is CRC-16-CCITT with polynomial 0x8408 3'b001: CRC mode 1, is CRC-16-IBM with polynomial 0xA001 3'b010: CRC mode 2, is CRC-16-T10-DIF with polynomial 0xEDD1 3'b011: CRC mode 3, is CRC-32-IEEE with polynomial 0xEDB88320 Else: No CRC support
8:6	qspiMode	3'b000	This field is valid only for the first section header in the section chain QSPI mode
5:3	memCfg1	3'b000	This field is valid only for the first section header in the section chain Memory configuration after code loading
2:0	memCfg0	3'b000	This field is valid only for the first section header in the section chain Memory configuration before code loading

24.3.2 Code Image

The code image is the binary images of a firmware which is generated by any tool chain for the ARM Cortex-M3 (such as the ARM RVCT or IAR compiler and linker). Depending on the value of memCfg0, a code image can be loaded into the space from 0x100000 to 0x16FFFF through D-bus or the one from 0x20000000 to 0x2004FFFF through system bus. The loading speed of the former is faster than the latter. On the other hand, the latter can access 320 KB RAM with memCfg0 3'b001, which enables loading code into system-bus only memory. D-bus can only access 448 KB RAM with memCfg0 3'b010.

Marvell does not suggest loading all code images by using Boot ROM. Instead, Boot ROM can load a bootloader into RAM and launch it to perform more sophisticated boot-loading tasks. Actually, the PFC could be a bootloader that defines the rest of the Flash/SRAM layout, an OS + application image; or diagnostic software.

24.3.3 Retention Data Format

The 88MC200 microcontroller consists of 4K-byte retention memory which is located in 0x480C0000. This memory module belongs to the VDD_AON domain so its data can be reserved in all power modes. The first 32 bytes of this memory area are used by Boot ROM. Table 24-6 shows the data structure.

Table 69: Retention Data Structure

Offset	Field Name	Description
0x00 ~ 0x07	N/A	Reserved
0x08 ~ 0x0B	pm3EntryAddr ¹	Address of entry function for PM3 mode wake up
0x0C ~ 0x1F	N/A	Reserved

¹ Users can place a function pointer in this address for PM3 fast boot. If waking from PM3 mode, Boot ROM fetches the pointer, go to the specified function.

24.4 UART Download Protocol

The basic download process via the UART port is as follows:

- Host sends detection byte
- Boot sends Header Request upon receipt of detection byte
- Host acknowledges request, then responds with required header
- Boot sends Data Request
- Host acknowledges request, then responds with required data
- Process continues until all the data bytes are downloaded to target
- Boot jumps the entry address



88MC200 Microcontroller Datasheet

25 Electrical, Mechanical and Thermal Specifications

25.1 Package Information

This chapter provides the package marking and mechanical specifications for the Marvell 88MC200 MCU. The 88MC200 device has two packages, 68 pin and 88 pin. The details are in [Figure 94](#) and [Table 70](#), [Figure 95](#) and [Table 71](#).

Figure 94: Mechanical Drawing for 68-pin Package

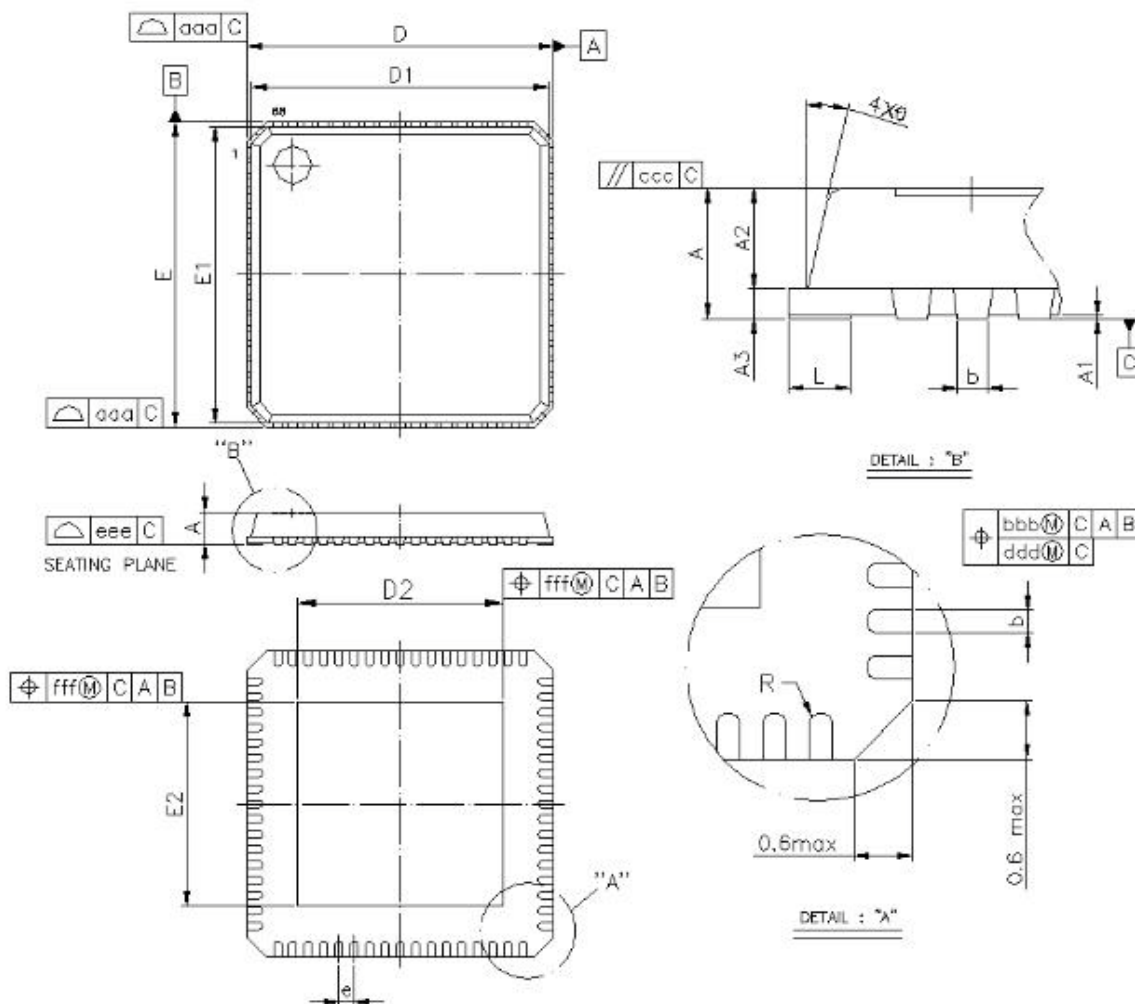


Table 70: Package Information for 68-pin Package

Symbol	Dimension in mm			Dimension in inches		
	MIN	NOM	MAX	MIN	NOM	MAX
A	0.80	0.85	1.00	0.031	0.033	0.039
A1	0.00	0.02	0.05	0.000	0.001	0.002
A2	0.60	0.65	0.80	0.024	0.026	0.031
A3	0.20 REF			0.008 REF		

Table 70: Package Information for 68-pin Package (Continued)

Symbol	Dimension in mm			Dimension in inches		
	MIN	NOM	MAX	MIN	NOM	MAX
b	0.15	0.20	0.25	0.006	0.008	0.010
D/E	8.00 BSC			0.315 BSC		
D1/E1	7.75 BSC			0.305 BSC		
D2/E2	5.20	5.35	5.50	0.205	0.211	0.217
e	0.40 BSC			0.016 BSC		
L	0.30	0.40	0.50	0.012	0.016	0.020
Θ	0°	---	14°	0°	---	14°
R	0.075	---	---	0.003	---	---
aaa	---	---	0.15	---	---	0.006
bbb	---	---	0.10	---	---	0.004
ccc	---	---	0.10	---	---	0.004
cdd	---	---	0.05	---	---	0.002
eee	---	---	0.08	---	---	0.003
fff	---	---	0.10	---	---	0.004

NOTE for Table 70 and Table 71:

1. Controlling dimension = millimeter
2. Reference document = JEDEC MO-220

Figure 95: Mechanical Drawing for 88-pin Package

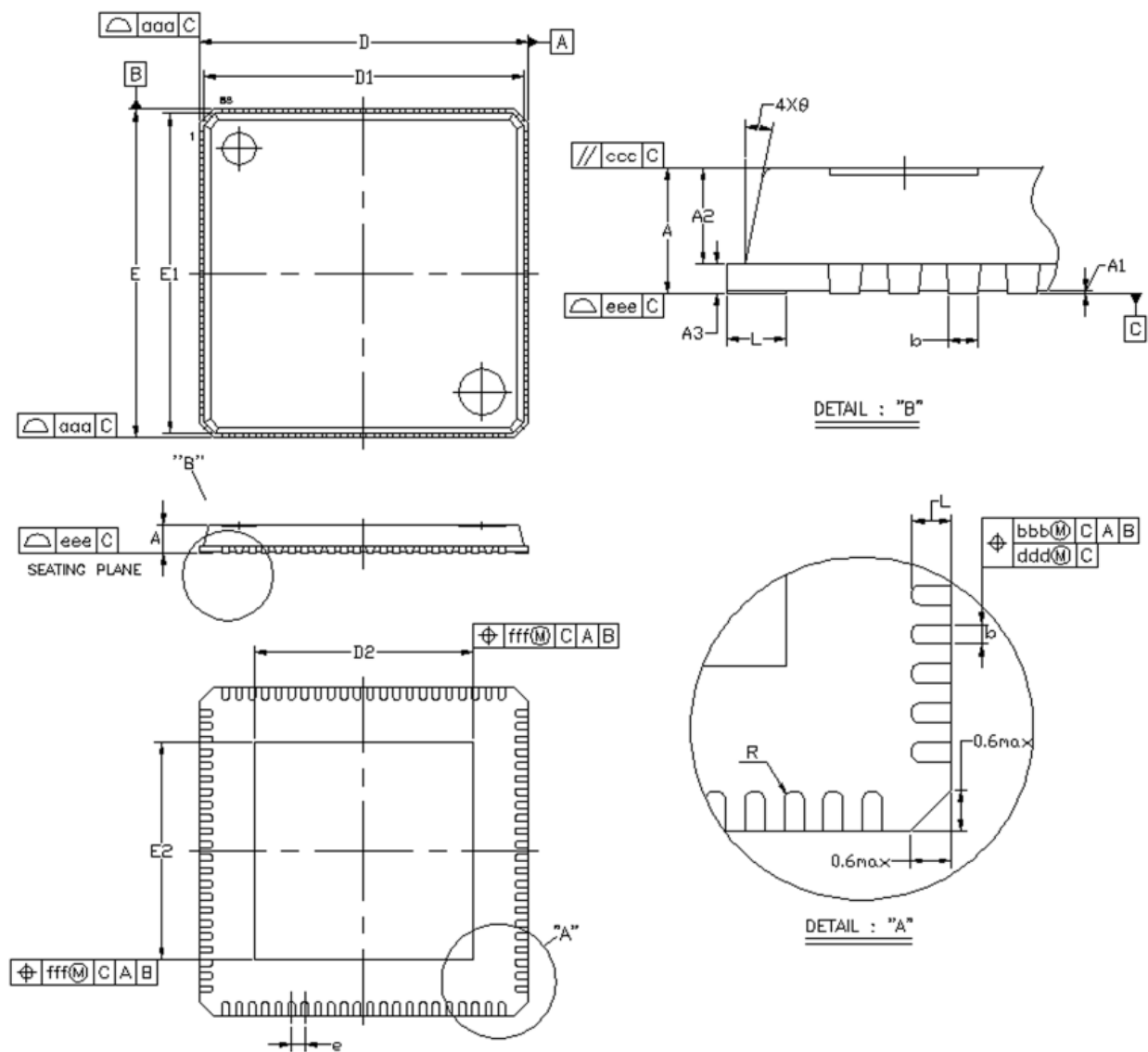


Table 71: Package Information for 88-pin Package (See Note under [Table 70](#))

Symbol	Dimension in mm			Dimension in inches		
	MIN	NOM	MAX	MIN	NOM	MAX
A	0.80	0.85	0.90	0.031	0.033	0.035
A1	0.00	0.02	0.05	0.000	0.001	0.002
A2	0.60	0.65	0.70	0.024	0.026	0.028
A3	0.20 REF			0.008 REF		
b	0.15	0.20	0.25	0.006	0.008	0.010
D/E	9.90	10.00	10.10	0.390	0.394	0.398
D1/E1	9.75 BSC			0.384 BSC		
D2/E2	5.85	6.00	6.15	0.230	0.236	0.242
e	0.40 BSC			0.016 BSC		
L	0.30	0.40	0.50	0.012	0.016	0.020
Θ	0°	---	14°	0°	---	14°
R	0.075	---	---	0.003	---	---
aaa	0.10			0.004		
bbb	0.07			0.003		
ccc	0.10			0.004		
ddd	0.05			0.002		
eee	0.08			0.003		
fff	0.10			0.004		

25.2 Maximum Ratings and Operating Conditions

25.2.1 Absolute Maximum Ratings

Table 72: Absolute Maximum Ratings

Symbol	Description	Min	Max	Units
Ts	Storage temperature	−55	125	°C
VCC_HV	Voltage applied to IO peripherals VDD_IO0, VDD_IO1, VDD_IO2, VDD_IO3, VDD_IO4	VSS−0.3	VSS+4.0	V
	USB_AVDD	VSS−0.3	VSS+4.0	V
VCC_MV	Voltage applied to VBAT supply pins	VSS−0.5	VSS+3.6	V

Table 72: Absolute Maximum Ratings (Continued)

Symbol	Description		Min	Max	Units
VESD	Maximum ESD stress voltage, three stresses maximum: - Any pin to any supply pin, either polarity, or - Any pin to all non-supply pins together, either polarity	HBM ¹	—	2000	V
		CDM ²	—	500	V
IEOS	Maximum DC input current (electrical overstress) for any non-supply pin		—	5	mA
¹ HBM = human body model ² CDM = charge device model					

25.2.2 Operating Conditions

This section discusses operating voltage, frequency, and temperature specifications for the Marvell 88MC200 MCU.

Refer to Chapter 5 for supported frequencies and clock-register settings as listed in [Table 73](#).

Table 73: Voltage, Temperature, and Frequency Electrical Specification

Symbol	Description	Min	Typical	Max	Units	Notes
Operating Temperature						
Tamb	Ambient Temperature	-40	--	+85	°C	1
VBAT Main Voltage						
Vbat	Main voltage	1.8		3.6	V	2
Tpwrramp	Ramp Rate	1.0	—	25.00	mV/μs	
VDD _ IO {0, 1, 2, 3, 4 } Voltage						
VDD_IOx_y	Voltage applied when using 1.8V devices	1.62	1.80	1.98	V	—
	Voltage applied when using 3.3V devices	2.97	3.30	3.63	V	—
Tsysramp	Ramp Rate	—	—	25.00	mV/μs	—
USB_AVDD Voltage						
USB_AVDD	Voltage applied on AVDD_OTG	3.00	3.30	3.6	V	—
Tsysramp	Ramp Rate	--	--	25.00	mV/μs	—

Table 73: Voltage, Temperature, and Frequency Electrical Specification (Continued)

Symbol	Description	Min	Typical	Max	Units	Notes
NOTE:						
1. Minimum ambient temperature depends on SKU.						
2. To ensure the ADC, DAC and Analog comparator modules work correctly, the suggested Vbat range is 2.0V ~ 3.6V.						

25.3 Electrical Characteristics

25.3.1 DC Voltage and Current Characteristics

25.3.2 I/O Port Characteristics

Table 74: I/O Static Characteristics on 3.3V VDD_IO

Specifications	Condition	Min	Nominal	Max	Unit
VDDO	--	2.97	3.3	3.63	V
VIL	--	-0.4	—	VDDO*30%	V
VIH	--	VDDO*70%	—	VDDO+0.4	V
Pullup strength (applicable to pullup pad only)	V(PAD)=0.5*VDDO	10	—	50	μA
Pulldown strength (applicable to pulldown pad only)	V(PAD)=0.5*VDDO	10	—	50	μA
I _{ol} @0.4V	--	4	--	--	mA
I _{oh} @VDDO-0.4V	--	4	--	--	mA
Input capacitance	--	--	--	5	pF
Input leakage 1	VDDO is ON, 0<V(PAD)<VDDO	--	--	2	μA

Table 75: I/O Static Characteristics on 1.8V VDD_IO

Specifications	Condition	Min	Nominal	Max	Unit
VDDO	--	1.62	1.8	1.98	V
VIL	--	-0.4	—	VDDO*30%	V
VIH	--	VDDO*70%	—	VDDO+0.4	V
Pullup strength (applicable to pullup pad only)	V(PAD)=0.5*VDDO	10	—	50	μA
Pulldown strength (applicable to pulldown pad only)	V(PAD)=0.5*VDDO	10	—	50	μA

Table 75: I/O Static Characteristics on 1.8V VDD_IO (Continued)

Specifications	Condition	Min	Nominal	Max	Unit
Iol@0.2V	--	2	--	--	mA
Loh@VDDo-0.2V	--	2	--	--	mA
Input capacitance	--	--	--	5	pF
Input leakage 1	VDDO is ON, 0<V(PAD)<VDDO	--	--	2	μA

Table 76: Current Consumption

Power Mode	Condition	Typical	Max	Unit
PM0-active (Power Mode = 00)	pll = on, MAINXTAL = on, RC32M = off, system clock = 200 MHz, typical peripherals on	60		mA
	pll = off, MAINXTAL = off, RC32M = on, system clock = 32 MHz, all peripherals off	14		mA
PM1-Idle (Power Mode = 00, RC32M is on)	Typical peripherals clock on, pll = off	12		mA
	All peripherals clock off, pll = off	10		mA
PM2-standby (Power Mode = 01, RC32M is on)	Flash is in powerdown mode	2		mA
PM3-sleep (Power Mode = 10, RC32K is on)	SRAM is in retention mode, RTC is on, ULP Comparator is off, brownout detection is off	40		μA
PM4-deep sleep (Power Mode = 11)	AON domain is on SRAM is in retention mode, RTC is on, ULP Comparator is off, brownout detection is off	9.0		μA

25.3.3 Clock Characteristics

This section provides the characteristics of various clock sources.

Table 77: MAINXTAL Oscillator

Element	Values
Supported local oscillator frequencies	4-50 MHz
Input clock signal type	CMOS and low-swing sine wave
Voltage range for square wave	Vih = 0.8 to 1.95V Vil = 0 to 0.25V
Voltage range for sine wave	0.2 to 1.2Vpp
Phase noise requirements	[1kHz, 10 kHz, 100 kHz, 1MHz] to [-120, -130, -135, 140 dBc/HZ]
Frequency accuracy tolerance	500 ppm
Startup time	<0.3 ms>
Slew rate (10-90%)	25 ns

Table 77: MAINXTAL Oscillator (Continued)

Element	Values
Duty cycle tolerance	35/65%

For Table 78: -40 to 85° C, VBAT = 3.6V with default setting unless otherwise specified.

Table 78: RC32K Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Units
Frequency Before Calibration		18.6	31.8	39.8	kHz
Startup Time			0.9		ms
After-Calibration Frequency Accuracy	Use crystal 32.768 kHz as reference clock	32.3	32.,7	33.1	kHz
Temperature Tolerance			65		ppm/C
Duty Cycle		40	50	60	%

25.3.4 Power and Brownout Detection

Table 79: Power-on Reset

Parameter	Condition	Min	Typical	Max	Unit
Power-on reset threshold (rising edge)	---	---	1.25	---	V

For Table 80: -40 to 85° C, with default setting unless otherwise specified.

Table 80: VBAT Brownout Detection (BOD) Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Units
V _{trig(BOD)} - V _{bat} Brownout Trigger Level	BRNTRIG_VBAT_CNTL = 0X0		1.70		V
	BRNTRIG_VBAT_CNTL = 0X1		1.80		
	BRNTRIG_VBAT_CNTL = 0X2		1.90		
	BRNTRIG_VBAT_CNTL = 0X3		2.00		
	BRNTRIG_VBAT_CNTL = 0X4		2.10		
	BRNTRIG_VBAT_CNTL = 0X5		2.20		

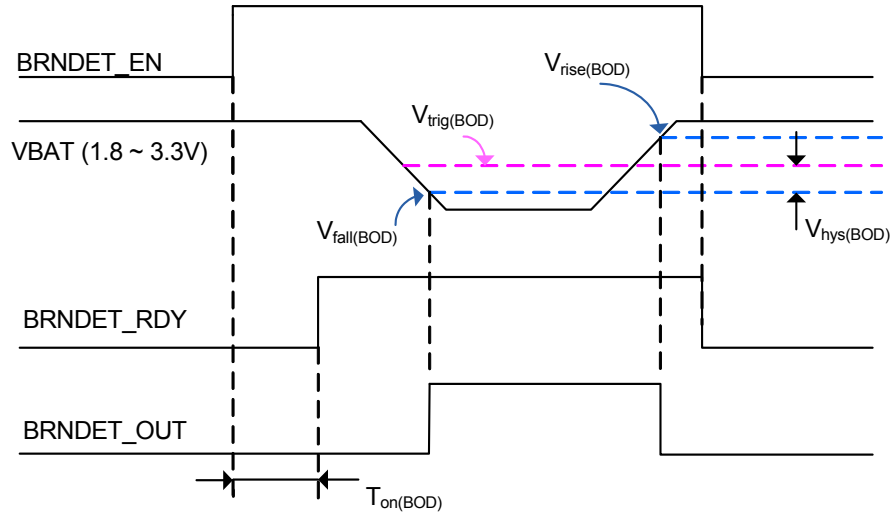
Table 80: VBAT Brownout Detection (BOD) Electrical Characteristics (Continued)

Parameter	Conditions	Min	Typ	Max	Units
$V_{\text{trig(BOD)}} - V_{\text{bat}}$ Brownout Trigger Level	BRNTRIG_VBAT_CNTL = 0X5		2.20		V
	BRNTRIG_VBAT_CNTL = 0X6		2.30		
	BRNTRIG_VBAT_CNTL = 0X7		2.40		
$V_{\text{hys(BOD)}} - V_{\text{bat}}$ Brownout Hysteresis	BRNHYST_VBAT_CNTL = 0X0		+/- 0		mV
	BRNHYST_VBAT_CNTL = 0X1		+/- 40		
	BRNHYST_VBAT_CNTL = 0X2		+/- 70		
	BRNHYST_VBAT_CNTL = 0X3		+/- 90		
$T_{\text{on(BOD)}} - V_{\text{bat}}$ Brownout Detector Turn on Time			200		μs

NOTE:

1. $V_{\text{fall(BOD)}}$ - BOD falling edge. $V_{\text{fall(BOD)}} = V_{\text{trig(BOD)}} - V_{\text{hys(BOD)}}$.
2. $V_{\text{rise(BOD)}}$ - BOD rising edge. $V_{\text{rise(BOD)}} = V_{\text{trig(BOD)}} + V_{\text{hys(BOD)}}$.

Figure 96: BOD Sequence Trigger Level



25.3.5 ADC Electrical Characteristics

The parameters in Table 81 are derived from design conditions of : $T_A = 25^\circ\text{C}$, $V_{bat} = 3.3\text{V}$ unless otherwise noted.

Table 81: ADC Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Units
Analog Supply Voltage	VBAT	2.0	3.3	3.6	V
Operation Temperature		-40		85	$^\circ\text{C}$
ADC Main Clock		30	32		MHz
Reference Voltage					
Internal Reference Voltage		1.19	1.20	1.21	V
External Reference Voltage		0		1.8	V
Analog Inputs					
Input Voltage	Input buffer disabled	0	-	VBAT	V
	Input buffer enabled	0.2	-	VBAT-0.2	

Table 81: ADC Electrical Characteristics (Continued)

Parameter	Conditions	Min	Typ	Max	Units
ADC Voltage Conversion Range	Refer to Table 54	-2*Vref ¹ or -VBAGT	-	2*Vref or VBAT	V
Internal Sampling and Hold Capacitance C_{ADC}			500		fF
Internal Sampling Switch Resistance R_{ADC}		1			k Ω
Input Multiplexer Impedance R_{MUX}		1			k Ω
External Input Resistance ² R_S	8MHz ADC operating clock without input buffer 16-bit settling accuracy			4	k Ω
	1MHz ADC operating clock without input buffer 16-bit settling accuracy			46	
Input Frequency Range ³	With input buffer		31		MHz
	Without input buffer		130		
Conversion Rate⁴					
ADC Sampling Clock Frequency	Fast Mode		4	8	MHz
	Low Power Mode		1	2	
Conversion Time in ADC Clocks	10-bit setting		32		cycles
	12-bit setting		64		
	14-bit setting		128		
	16-bit setting		256		
One Shot Latency	16-bit setting@ 8MHz	96+T _{WARM} ⁵			μ s
	10-bit setting@ 8MHz	12+T _{WARM} ⁵			
	16-bit setting@ 1MHz	768+T _{WARM} ⁵			
	10-bit setting@ 1MHz	96+T _{WARM} ⁵			
Data Rate	16-bit setting@ 8MHz		31.2		ksps
	10-bit setting@ 8MHz		250		
	16-bit setting@ 1MHz		3.9		
	10-bit setting@1MHz		31.2		
DC Accuracy					
Resolution	Single-ended			15	bits
	Differential			16	
Differential Integral Nonlinearity	Internal 1.2V reference, 16-bit setting, 1MHz ADC clock, 20 Hz sine wave single-ended input	-	+/- 1		LSB
Offset Error	Before calibration 16-bit setting		+/- 30		LSB

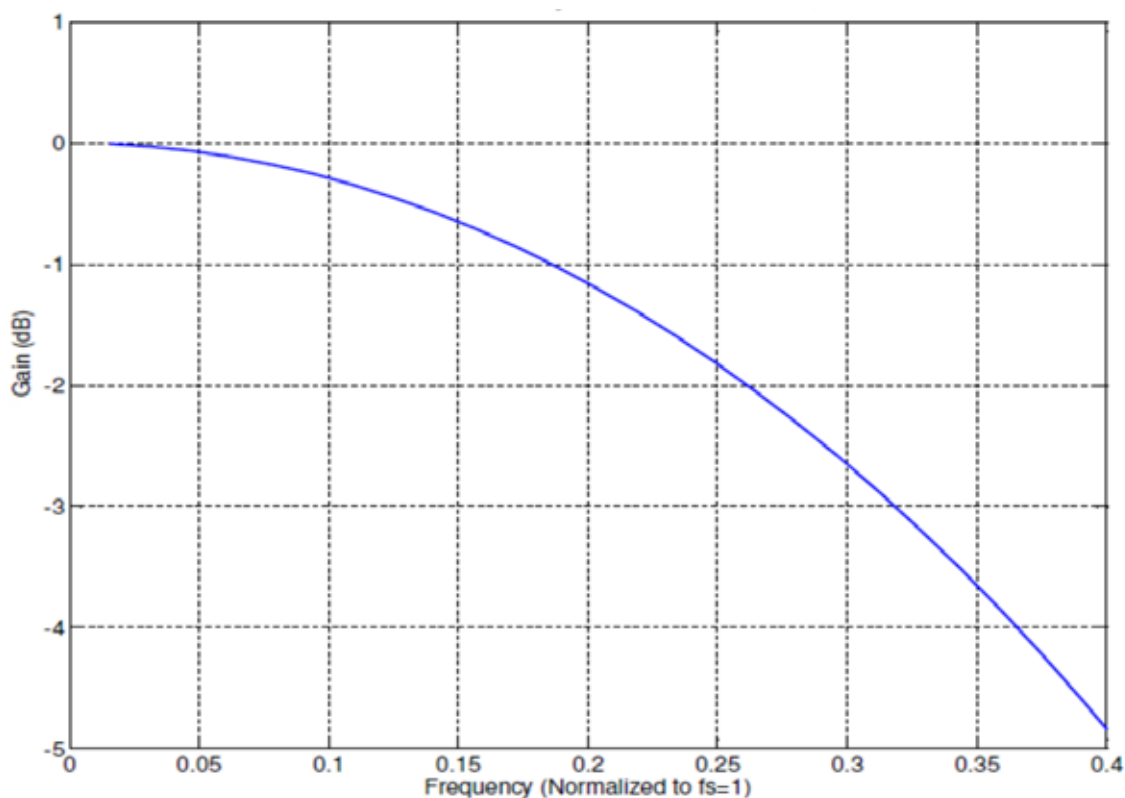
Table 81: ADC Electrical Characteristics (Continued)

Parameter	Conditions	Min	Typ	Max	Units
	After calibration ¹ 16-bit setting		+/- 2		
Dynamic Performance (200 Hz sine wave differential input, -1db of full scale, 1 MHz ADC clock)					
ENOB (Effective Number of Bit)	10-bit setting internal 1.2V reference		9.2		Bit
	12-bit setting internal 1.2V reference		11.2		
	14-bit setting internal 1.2V reference		12.6		
	16-bit setting internal 1.2V reference		14		
	16-bit setting internal 1.2V reference		14		
SNDR (Signal to Noise and Distortion Ratio)	10-bit setting internal 1.2V reference		57		dB
	12-bit setting internal 1.2V reference		69		
	14-bit setting internal 1.2V reference		77		
	16-bit setting internal 1.2V reference		86		
	16-bit setting internal 1.2V reference		86		
THD (Total Harmonic Distortion)	10-bit setting		-71		dB
	12-bit setting		-80		
	14-bit setting		-83		
	16-bit setting		-86		
Warm-up Time					
Warmup Time of ADC and internal reference generator (T _{WARM})	Programmable ADC main clock frequency = 32 MHz	1		32	μs
¹ Vref stands for the voltage reference of ADC. It could be Vref_12, Vref_18, or external voltage (<1.8V). ² External input resistance (R_s) max formula $R_s < \frac{1/f_{ADC}}{4 * \ln(2^{15}) * C_{ADC}} - (R_{ADC} + R_{MUX})$ C_{ADC} : Internal Sampling and Hold Capacitance R_{ADC} : Internal Sampling Switch Resistance R_{MUX} : Input Multiplexer Impedance ³ Defined as input signal -3dB bandwidth through analog path. ⁴ ADC main clock frequency is 32 MHz. Conversion rate is linearly scaled when clock frequency is 30 MHz. ⁵ Marvell recommends performing a calibration after each power-up sequence.					

Table 82: ADC Digital Filter Electrical Characteristics

Parameter	Conditions	Min	Typical	Max	Units
-3dB Passband				0.32	Fs ¹
Passband with flatness within +/- 0.1dB				0.06	Fs
NOTE: ¹ Digital filter response (see graph below) is clock dependent and scaled with Fs, where Fs is the data rate					

Figure 97: Digital Filter Frequency Response



25.3.6 Analog Temperature Sensor Characteristics

All parameters in Table 83 are listed with conditions of $T_A = 25^\circ\text{C}$, $V_{BAT} = 3.3\text{V}$, unless otherwise noted. On-chip temperature can be measured by using ADC with 1.2V internal voltage reference and 14-bit resolution setting.

Table 83: Analog Temperature Sensor Characteristics

Parameter	Conditions	Min	Typical	Max	Unit
Analog Supply Voltage	VBAT	2.0	3.3	3.6	V
Operation Temperature		-40		85	°C
Internal Reference Voltage		1.19	1.2	1.21	V
Conversion Time in ADC Clocks	14-bit resolution setting		512		clock cycles
Throughput Rate ¹	8MHz ADC operating clock		15.6		ksps
	1MHz ADC operating clock		1.95		
Measurement Latency ¹	8MHz ADC operating clock		64+T _{WARM}		μs
	1MHz ADC operating clock		512+T _{WARM}		
Resolution	Traditional Method 1LSB=1.2V/2 ¹³		0.61		°C/LSB
Measurement Accuracy	Initial accuracy w/o calibration		±5		°C
	with 1 temperature calibration		±3		

¹ ADC main clock frequency is 32 MHz. Conversion rate is linearly scaled when clock frequency is 30 MHz.

25.3.7 ACOMP Electrical Characteristics

All parameters in [Table 84](#) are listed with conditions of T_A = 25°C, VBAT = 3.3V, unless otherwise noted.

Table 84: ACOMP Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Units
Analog Supply Voltage	VBAT	2.0	3.3	3.6	V
Operation Temperature		-40	-	85	°C
Analog Input					
Analog Input Voltage	Any pin (in Analog Input Mode)	0	-	VBAT	V
Common Mode Input Range		0	-	VBAT	V

Table 84: ACOMP Electrical Characteristics (Continued)

Parameter	Conditions	Min	Typ	Max	Units
Reference Voltage					
Internal Reference Voltage		1.19	1.20	1.21	V
Analog Response Time (No digital delay ¹)					
Fast Response Mode MODE 3, $V_{cm}^2 = 1.5V$	Overdrive (COMP_P-COMP_N) = +/- 100mV		110		ns
Medium Response Mode Mode 2, $V_{cm}^2 = 1.5V$	Overdrive (COMP_P-COMP_N) = +/- 100mV		190		ns
Slow Response Mode Mode 1, $V_{cm}^2 = 1.5V$	Overdrive (COMP_P-COMP_N) = +/- 100mV		450		ns
DC Offset					
Offset Voltage			+/- 10		mV
Hysteresis	Programmed in 7 steps and zero		10 20 30 40 50 60 70		mV
Warm-up Time					
Warmup Time of ACOMP and internal reference generator (T_{WARM})	VBAT = 3.6V		0.6		μs
	VBAT = 3.0V		1.0		
	VBAT = 2.4V		1.8		
¹ Digital Delay can be up to max of two 16 MHz clock periods					
² V_{cm} is the common-mode voltage on COMP_P and COMP_N.					

25.3.8 DAC Electrical Characteristics

All parameters in [Table 85](#) are listed with conditions of $T_A = 25^\circ C$, VBAT = 3.3V, unless otherwise noted.

Table 85: DAC Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Units
Analog Supply Voltage	VBAT	2.0	3.3	3.6	V
Operation Temperature		-40		85	$^\circ C$

Table 85: DAC Electrical Characteristics (Continued)

Parameter	Conditions	Min	Typ	Max	Units
Reference Voltage (Vref)					
Internal Reference Voltage		0.72		1.6	V
External Reference Voltage		0.72		VBAT	V
Conversion Range					
Analog Output Range	Single-ended; Refer to Table 62 Output Voltage Calculation formula	0.2		VBAT-0.2	V
Output Load					
Resistive Load (Minimum resistive load between DAC output and VSS)	Single-ended	1K			Ohm
	Differential	1K			
Capacitive Load (Maximum capacitive load at DAT output)				50	pF
Conversion Rate					
Conversion Rate	Max frequency for a correct DAC output change with 10-bit setting	62.5		500	kHz
DC Accuracy					
Resolution	Single-ended			10	bits
	Differential			10	
Differential Nonlinearity (RMS)	Guaranteed monotonic, internal 1.2V reference	-	+/- 0.5	+/- 1	LSB ¹
Integral Nonlinearity (best-fit method)	Internal reference	-	+/- 1.5	+/-4	LSB
Offset Error	A_range = 2'b11, internal Vref (refer to Table 62); difference between measured value at code (0x200)		5		mV
Gain Error (after offset removal)	A_range = 2'b11, internal Vref (refer to Table 62)		1		%
¹ LSB = Vref/1024					

-40 to 85°C, VBAT = 3.6V with default setting unless otherwise specified for [Table 86](#).

Table 86: Ultra Low-Power Comparator Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Units
Input Range		0	$V_{BAT}/2$	V_{BAT}	V
ULP Comparator Turn-on Time			100		μs
ULP Comparator Hysteresis Window	COMP_HYST = 0 (single-ended mode)		0		mV
	COMP_HYST = 0 (differential mode)		0		
	COMP_HYST = 1 (single-ended mode)		4		
	COMP_HYST = 1 (differential mode)		8		
	COMP_HYST = 2 (single-ended mode)		8		
	COMP_HYST = 2 (differential mode)		17		
	COMP_HYST = 3 (single-ended mode)		22		
	COMP_HYST = 3 (differential mode)		52		
Response Time			5		μs

25.3.9 AC Electrical Characteristics

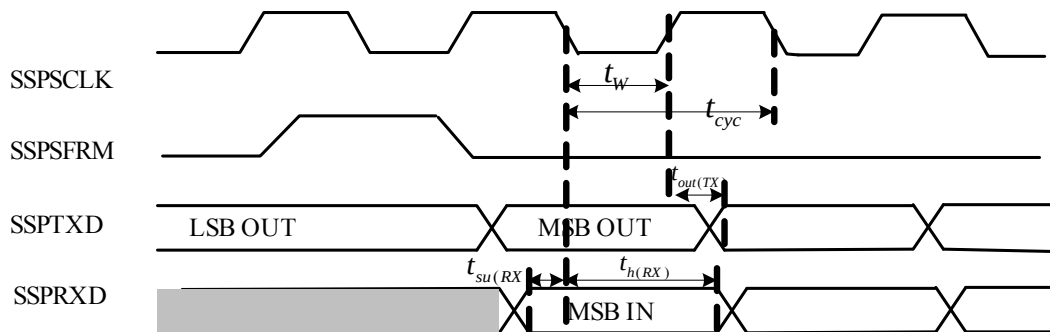
This section includes alternating-current (AC) characteristics, timing diagrams and timing parameters for the Marvell 88MC200 MCU controllers/interfaces listed below:

25.3.9.1 SSP Timing Diagram and Specifications

Table 87: SSP Timing

SYMBOL	PARAMETER	MASTER OF CLOCK		SLAVE TO CLOCK		UNIT
		MIN	MAX	MIN	MAX	
$t_{out(TX)}$	TX delay time	1.6		12		ns
$t_{su(RX)}$	Set up time RX valid before clock low	10		2.2		ns
$t_{h(RX)}$	Hold time, RX Data valid after clock low	0		3.7		ns
T_{cyc}	Serial Bit Clock cycle time	40		40		ns
t_w	Serial Clock high/low time	$T_{cyc}/2-13$		$T_{cyc}/2-13$		ns

Figure 98: SSP Serial Frame Format



25.3.9.2 QSPI Timing Diagram and Specifications

Table 88: QSPI Characteristics

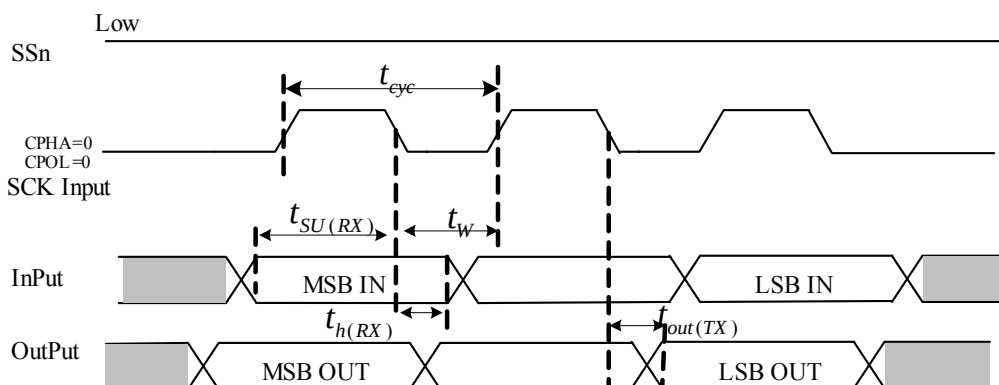
SYMBOL	PARAMETER	MASTER		UNIT
		MIN	MAX	
t_{cyc}	QSPI clock cycle time	20		ns
t_w	Clock high and low time	$T_{cyc}/2-6.67$		ns
$t_{su(RX)}$	Data input setup time	5.5		ns
$t_{h(RX)}$	Data input hold time,	0		ns

Table 88: QSPI Characteristics (Continued)

SYMBOL	PARAMETER	MASTER		UNIT
		MIN	MAX	
$t_{out(TX)}$	Data output delay time	3.2		ns

Condition : Capacitive load C=5pF ; The min annotated_transition = 0.5ns ; The max annotated_transition=1.5ns.

Figure 99: QSPI Timing Diagram



Shaded areas are not valid.

25.3.9.3 SDIO Timing Diagram and Specifications

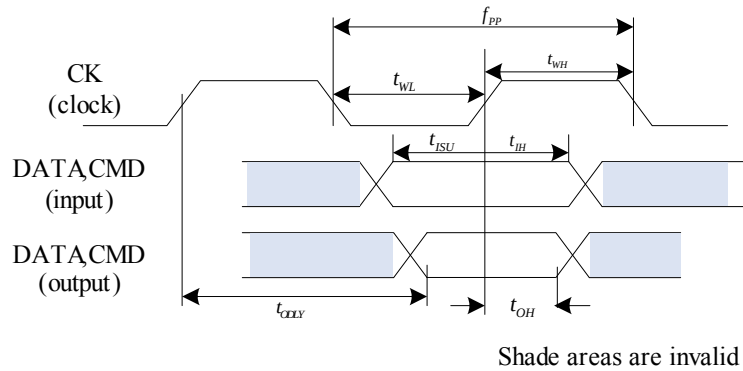
Table 89: SDIO Characteristics in Full Speed Mode

Parameter	Symbol	Min	Max	Unit
Clock				
Clock frequency	f_{PP}	0	25	MHz
Clock low time	t_{WL}	10		ns
Clock high time	t_{WH}	10		ns
Inputs CMD, DAT (reference to clock)				
Input setup time	t_{ISU}	7.8		ns
Input hold time	t_{IH}	0		ns
Outputs CMD, DATA (reference to clock)				
Outputs delay time during data transfer Mode	t_{ODLY}		20.24	ns
Output Hold time	t_{OH}	14.63		ns

Table 90: SDIO Timing Specification High-Speed Mode

Parameter	Symbol	Min	Max	Unit
Clock				
Clock frequency data transfer Mode	f_{PP}	0	50	MHz
Clock low time	t_{WL}	7		ns
Clock high time	t_{WH}	7		ns
Inputs CMD, DAT (reference to clock)				
Input setup time	t_{ISU}	7.8		ns
Input hold time	t_{IH}	0		ns
Outputs CMD, DATA (reference to clock)				
Outputs delay time during data transfer Mode	t_{ODLY}		10.24	ns
Output Hold time	t_{OH}	4.63		ns

Figure 100:SDIO DC Parameter: Bus Timing (PAD)



25.3.9.4 RESETn Pin Specification

-40 to +85°C, VBAT=3.6V with default settings unless otherwise specified.

Parameter	Condition	Min	Typical	Max	Unit
Minimum reset pulse width on RESETn Pin ¹	--	80	--	--	μs

¹ From design, not production



88MC200 Microcontroller Datasheet



88MC200 Microcontroller

Datasheet Appendix A:
Register Tables



88MC200 Microcontroller Register Tables

Table of Contents

88MC200 Register Information	5
DMA Register Information	7
USBC Register Information	49
SDIO Register Information	81
AES Register Information	107
CRC Register Information	127
I2C Register Information	132
QSPI Register Information	168
SSP Register Information	186
UART Register Information	202
GPIO Register Information	242
GPT Register Information	258
RC32M Register Information	275
ADC Register Information	281
DAC Register Information	294
ACOMP Register Information	305
PINMUX Register Information	321
WDT Register Information	378
RTC Register Information	382
PMU Register Information	388
SYS_CTRL Register Information	431



A 88MC200 Register Information

Table 1: 88MC200 Memory Map

Unit	Base Address	Details
(DMA)	0x4400_0000	Page: 7
(USBC)	0x4400_1000	Page: 49
(SDIO)	0x4400_2000	Page: 81
(AES)	0x4400_4000	Page: 107
(CRC)	0x4400_5000	Page: 127
(I2C0)	0x4600_0000	Page: 132
(QSPI0)	0x4601_0000	Page: 168
SSP (SSP0)	0x4602_0000	Page: 186
SSP (SSP1)	0x4603_0000	Page: 186
(UART0)	0x4604_0000	Page: 202
(UART1)	0x4605_0000	Page: 202
(GPIO)	0x4606_0000	Page: 242
(GPT0)	0x4607_0000	Page: 258
(GPT1)	0x4608_0000	Page: 258
(QSPI1)	0x4609_0000	Page: 168
(RC32M)	0x460A_0000	Page: 275
(ADC0)	0x460B_0000	Page: 281
(ADC1)	0x460B_0100	Page: 281
(DAC)	0x460B_0200	Page: 294
(ACOMP)	0x460B_0300	Page: 305
SSP (SSP2)	0x4800_0000	Page: 186
(PIN_MUX)	0x4801_0000	Page: 321
(UART2)	0x4802_0000	Page: 202
(UART3)	0x4803_0000	Page: 202
(WDT)	0x4804_0000	Page: 378
(I2C1)	0x4805_0000	Page: 132
(I2C2)	0x4806_0000	Page: 132
(GPT2)	0x4807_0000	Page: 258

Table 1: 88MC200 Memory Map

Unit	Base Address	Details
(GPT3)	0x4808_0000	Page: 258
(RTC)	0x4809_0000	Page: 382
(PMU)	0x480A_0000	Page: 388
(SYS_CTRL)	0x480B_0000	Page: 431

A.1 DMA Register Information

Channel 0 registers

Table 2: DMA Register Summary

Offset	Name	Description	Details
0x000	SAR0		Page: 9
0x008	DAR0		Page: 9
0x018	CTL0		Page: 10
0x040	CFG0		Page: 11
0x058	SAR1		Page: 12
0x060	DAR1		Page: 12
0x070	CTL1		Page: 13
0x098	CFG1		Page: 14
0x0B0	SAR2		Page: 15
0x0B8	DAR2		Page: 15
0x0C8	CTL2		Page: 16
0x0F0	CFG2		Page: 17
0x108	SAR3		Page: 18
0x110	DAR3		Page: 18
0x120	CTL3		Page: 19
0x148	CFG3		Page: 20
0x160	SAR4		Page: 21
0x168	DAR4		Page: 21
0x178	CTL4		Page: 22
0x1A0	CFG4		Page: 23
0x1B8	SAR5		Page: 24
0x1C0	DAR5		Page: 24
0x1D0	CTL5		Page: 25
0x1F8	CFG5		Page: 26
0x210	SAR6		Page: 27
0x218	DAR6		Page: 27
0x228	CTL6		Page: 28
0x250	CFG6		Page: 29

Table 2: DMA Register Summary

Offset	Name	Description	Details
0x268	SAR7		Page: 30
0x270	DAR7		Page: 30
0x280	CTL7		Page: 31
0x2A8	CFG7		Page: 32
0x2C0	RAWTFR		Page: 33
0x2C8	RAWBLOCK		Page: 33
0x2D0	RAWSRCTRAN		Page: 34
0x2D8	RAWDSTTRAN		Page: 34
0x2E0	RAWERR		Page: 35
0x2E8	STATUSTFR		Page: 35
0x2F0	STATUSBLOCK		Page: 36
0x2F8	STATUSSRCTRAN		Page: 36
0x300	STATUSDSTTRAN		Page: 37
0x308	STATUSERR		Page: 37
0x310	MASKTFR		Page: 38
0x318	MASKBLOCK		Page: 38
0x320	MASKSRCTRAN		Page: 39
0x328	MASKDSTTRAN		Page: 39
0x330	MASKERR		Page: 40
0x338	CLEARTFR		Page: 40
0x340	CLEARBLOCK		Page: 41
0x348	CLEARSRCTRAN		Page: 41
0x350	CLEARDSTTRAN		Page: 42
0x358	CLEARERR		Page: 42
0x360	STATUSINT		Page: 43
0x368	REQSRCREG		Page: 43
0x370	REQDSTREG		Page: 44
0x378	SGLREQSRCREG		Page: 44
0x380	SGLREQDSTREG		Page: 45
0x388	RESERVED	Reserved	Page: 45
0x390	RESERVED	Reserved	Page: 46

Table 2: DMA Register Summary

Offset	Name	Description	Details
0x398	DMACFGREG		Page: 46
0x3A0	CHENREG		Page: 47
0x3A8	RESERVED	Reserved	Page: 47
0x3B0	RESERVED	Reserved	Page: 48
0x3F8	RESERVED	Reserved	Page: 48

A.1.1 (SAR0)

Channel 0 source address

Instance Name
SAR0

Offset
0x000

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3: (SAR0)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	SAR	R/W	0x0	Source address

A.1.2 (DAR0)

Channel 0 destination address

Instance Name
DAR0

Offset
0x008

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4: (DAR0)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	DAR	R/W	0x0	Destination address

A.1.3 (CTL0)

Channel 0 control

Instance Name

CTL0

Offset

0x018

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	
Field	Reserved																					BLOCK_TS											
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	1	0		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved		Reserved								Reserved	TT_FC		Reserved				SRC_MSIZ		Reserved	DEST_MSIZ		SINC		DINC		SRC_TR_WIDTH			DST_TR_WIDTH			INT_EN
Default	?	?	?	0	0	0	0	0	0	?	1	1	?	?	?	?	0	1	?	0	1	0	0	0	0	0	0	0	0	0	0	1	

Table 5: (CTL0)

Bits	Name	Type	Reset	Description
63:42	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
41:32	BLOCK_TS	R/W	0x2	Block length
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
28:23	Reserved	R/W	0x0	Reserved. Do not change the reset value.
22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21:20	TT_FC	R/W	0x3	Flow control
19:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:14	SRC_MSIZ	R/W	0x1	Source transaction burst length
13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12:11	DEST_MSIZ	R/W	0x1	Destination transaction burst length
10:9	SINC	R/W	0x0	Source address direction control
8:7	DINC	R/W	0x0	Destination address direction control
6:4	SRC_TR_WIDTH	R/W	0x0	Source transfer width
3:1	DST_TR_WIDTH	R/W	0x0	Destination transfer width

Table 5: (CTL0)

Bits	Name	Type	Reset	Description
0	INT_EN	R/W	0x1	Interrupt enable

A.1.4 (CFG0)

Channel 0 configuration

Instance Name
CFG0

Offset
0x040

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																DEST_PER				SRC_PER				Reserved		Reserved			FIFO_MODE	FCMODE	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	?	?	0	0	1	0	0	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												Reserved		Reserved					HS_SEL_SRC	HS_SEL_DST	FIFO_EMPTY	CH_SUSP	CH_PRIOR		Reserved						
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	?	?	?	1	1	0	0	0	0	0	?	?	?	?	?

Table 6: (CFG0)

Bits	Name	Type	Reset	Description
63:47	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
46:43	DEST_PER	R/W	0x0	Destination hardware interface
42:39	SRC_PER	R/W	0x0	Source hardware interface
38:37	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
36:34	Reserved	R/W	0x1	Reserved. Do not change the reset value.
33	FIFO_MODE	R/W	0x0	Channel FIFO mode control
32	FCMODE	R/W	0x0	Channel flow control mode
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	Reserved	R/W	0x0	Reserved. Do not change the reset value.
17:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	HS_SEL_SRC	R/W	0x1	Source handshake select
10	HS_SEL_DST	R/W	0x1	Destination handshake select
9	FIFO_EMPTY	R	0x0	FIFO empty status
8	CH_SUSP	R/W	0x0	Channel Suspend control

Table 6: (CFG0)

Bits	Name	Type	Reset	Description
7:5	CH_PRIOR	R/W	0x0	Channel Priority
4:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.1.5 (SAR1)

Channel 1 source address

Instance Name

SAR1

Offset

0x058

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7: (SAR1)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	SAR	R/W	0x0	Source address

A.1.6 (DAR1)

Channel 1 destination address

Instance Name

DAR1

Offset

0x060

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8: (DAR1)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	DAR	R/W	0x0	Destination address

A.1.7 (CTL1)

Channel 1 control

Instance Name

CTL1

Offset

0x070

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																					BLOCK_TS										
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	1	0	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved		Reserved						Reserved	TT_FC		Reserved				SRC_MSIZ		Reserved	DEST_MSIZ		SINC		DINC		SRC_TR_WIDTH			DST_TR_WIDTH			INT_EN	
Default	?	?	?	0	0	0	0	0	0	?	1	1	?	?	?	?	0	1	?	0	1	0	0	0	0	0	0	0	0	0	0	1

Table 9: (CTL1)

Bits	Name	Type	Reset	Description
63:42	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
41:32	BLOCK_TS	R/W	0x2	Block length
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
28:23	Reserved	R/W	0x0	Reserved. Do not change the reset value.
22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21:20	TT_FC	R/W	0x3	Flow control
19:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:14	SRC_MSIZE	R/W	0x1	Source transaction burst length
13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12:11	DEST_MSIZE	R/W	0x1	Destination transaction burst length
10:9	SINC	R/W	0x0	Source address direction control
8:7	DINC	R/W	0x0	Destination address direction control
6:4	SRC_TR_WIDTH	R/W	0x0	Source transfer width
3:1	DST_TR_WIDTH	R/W	0x0	Destination transfer width
0	INT_EN	R/W	0x1	Interrupt enable

A.1.8 (CFG1)

Channel 1 configuration

Instance Name

CFG1

Offset

0x098

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																DEST_PER				SRC_PER				Reserved		Reserved		FIFO_MODE	FCMODE		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	?	?	0	0	1	0	0
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												Reserved		Reserved				HS_SEL_SRC		HS_SEL_DST	FIFO_EMPTY	CH_SUSP	CH_PRIOR		Reserved						
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	?	?	?	1	1	0	0	0	0	1	?	?	?	?	?

Table 10: (CFG1)

Bits	Name	Type	Reset	Description
63:47	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
46:43	DEST_PER	R/W	0x0	Destination hardware interface
42:39	SRC_PER	R/W	0x0	Source hardware interface
38:37	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
36:34	Reserved	R/W	0x1	Reserved. Do not change the reset value.
33	FIFO_MODE	R/W	0x0	FIFO mode control
32	FCMODE	R/W	0x0	Flow control mode
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	Reserved	R/W	0x0	Reserved. Do not change the reset value.
17:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	HS_SEL_SRC	R/W	0x1	Source handshake select
10	HS_SEL_DST	R/W	0x1	Destination handshake select
9	FIFO_EMPTY	R	0x0	FIFO empty status
8	CH_SUSP	R/W	0x0	Channel Suspend control
7:5	CH_PRIOR	R/W	0x1	Channel Priority
4:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.1.9 (SAR2)

Channel 2 source address

Instance Name

SAR2

Offset

0x0B0

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11: (SAR2)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	SAR	R/W	0x0	Source address

A.1.10 (DAR2)

Channel 2 destination address

Instance Name

DAR2

Offset

0x0B8

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 12: (DAR2)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	DAR	R/W	0x0	Destination address

A.1.11 (CTL2)

Channel 2 control

Instance Name

CTL2

Offset

0x0C8

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																					BLOCK_TS										
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	1	0	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved		Reserved							Reserved	TT_FC		Reserved				SRC_MSIZ		Reserved	DEST_MSIZ		SINC		DINC		SRC_TR_WIDTH			DST_TR_WIDTH			INT_EN
Default	?	?	?	0	0	0	0	0	0	?	1	1	?	?	?	?	0	1	?	0	1	0	0	0	0	0	0	0	0	0	0	1

Table 13: (CTL2)

Bits	Name	Type	Reset	Description
63:42	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
41:32	BLOCK_TS	R/W	0x2	Block length
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
28:23	Reserved	R/W	0x0	Reserved. Do not change the reset value.
22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21:20	TT_FC	R/W	0x3	Flow control
19:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:14	SRC_MSIZ	R/W	0x1	Source transaction burst length
13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12:11	DEST_MSIZ	R/W	0x1	Destination transaction burst length
10:9	SINC	R/W	0x0	Source address direction control
8:7	DINC	R/W	0x0	Destination address direction control
6:4	SRC_TR_WIDTH	R/W	0x0	Source transfer width
3:1	DST_TR_WIDTH	R/W	0x0	Destination transfer width
0	INT_EN	R/W	0x1	Interrupt enable

A.1.12 (CFG2)

Channel 2 configuration

Instance Name

CFG2

Offset

0x0F0

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																DEST_PER				SRC_PER				Reserved		Reserved		FIFO_MODE	FCMODE		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	?	?	0	0	1	0	0
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												Reserved		Reserved						HS_SEL_SRC	HS_SEL_DST	FIFO_EMPTY	CH_SUSP	CH_PRIOR		Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	?	?	?	1	1	0	0	0	1	0	?	?	?	?	?

Table 14: (CFG2)

Bits	Name	Type	Reset	Description
63:47	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
46:43	DEST_PER	R/W	0x0	Destination hardware interface
42:39	SRC_PER	R/W	0x0	Source hardware interface
38:37	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
36:34	Reserved	R/W	0x1	Reserved. Do not change the reset value.
33	FIFO_MODE	R/W	0x0	FIFO mode control
32	FCMODE	R/W	0x0	Flow control mode
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	Reserved	R/W	0x0	Reserved. Do not change the reset value.
17:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	HS_SEL_SRC	R/W	0x1	Source handshake select
10	HS_SEL_DST	R/W	0x1	Destination handshake select
9	FIFO_EMPTY	R	0x0	FIFO empty status
8	CH_SUSP	R/W	0x0	Channel Suspend control
7:5	CH_PRIOR	R/W	0x2	Channel Priority
4:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.1.13 (SAR3)

Channel 3 source address

Instance Name

SAR3

Offset

0x108

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15: (SAR3)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	SAR	R/W	0x0	Source address

A.1.14 (DAR3)

Channel 3 destination address

Instance Name

DAR3

Offset

0x110

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 16: (DAR3)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	DAR	R/W	0x0	Destination address

A.1.15 (CTL3)

Channel 3 control

Instance Name

CTL3

Offset

0x120

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																					BLOCK_TS										
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	1	0	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved		Reserved						Reserved	TT_FC		Reserved				SRC_MSIZ		Reserved	DEST_MSIZ		SINC		DINC		SRC_TR_WIDTH			DST_TR_WIDTH			INT_EN	
Default	?	?	?	0	0	0	0	0	0	?	1	1	?	?	?	?	0	1	?	0	1	0	0	0	0	0	0	0	0	0	0	1

Table 17: (CTL3)

Bits	Name	Type	Reset	Description
63:42	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
41:32	BLOCK_TS	R/W	0x2	Block length
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
28:23	Reserved	R/W	0x0	Reserved. Do not change the reset value.
22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21:20	TT_FC	R/W	0x3	Flow control
19:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:14	SRC_MSIZE	R/W	0x1	Source transaction burst length
13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12:11	DEST_MSIZE	R/W	0x1	Destination transaction burst length
10:9	SINC	R/W	0x0	Source address direction control
8:7	DINC	R/W	0x0	Destination address direction control
6:4	SRC_TR_WIDTH	R/W	0x0	Source transfer width
3:1	DST_TR_WIDTH	R/W	0x0	Destination transfer width
0	INT_EN	R/W	0x1	Interrupt enable

A.1.16 (CFG3)

Channel 3 configuration

Instance Name

CFG3

Offset

0x148

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																DEST_PER				SRC_PER				Reserved		Reserved		FIFO_MODE	FCMODE		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	?	?	0	0	1	0	0
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												Reserved		Reserved				HS_SEL_SRC		HS_SEL_DST	FIFO_EMPTY	CH_SUSP	CH_PRIOR		Reserved						
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	?	?	?	1	1	0	0	0	1	1	?	?	?	?	?

Table 18: (CFG3)

Bits	Name	Type	Reset	Description
63:47	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
46:43	DEST_PER	R/W	0x0	Destination hardware interface
42:39	SRC_PER	R/W	0x0	Source hardware interface
38:37	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
36:34	Reserved	R/W	0x1	Reserved. Do not change the reset value.
33	FIFO_MODE	R/W	0x0	FIFO mode control
32	FCMODE	R/W	0x0	Flow control mode
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	Reserved	R/W	0x0	Reserved. Do not change the reset value.
17:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	HS_SEL_SRC	R/W	0x1	Source handshake select
10	HS_SEL_DST	R/W	0x1	Destination handshake select
9	FIFO_EMPTY	R	0x0	FIFO empty status
8	CH_SUSP	R/W	0x0	Channel Suspend control
7:5	CH_PRIOR	R/W	0x3	Channel Priority
4:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.1.17 (SAR4)

Channel 4 source address

Instance Name

SAR4

Offset

0x160

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 19: (SAR4)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	SAR	R/W	0x0	Source address

A.1.18 (DAR4)

Channel 4 destination address

Instance Name

DAR4

Offset

0x168

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20: (DAR4)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	DAR	R/W	0x0	Destination address

A.1.19 (CTL4)

Channel 4 control

Instance Name

CTL4

Offset

0x178

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	
Field	Reserved																					BLOCK_TS											
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	1	0	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved		Reserved							Reserved	TT_FC		Reserved				SRC_MSIZ		Reserved		DEST_MSIZ		SINC		DINC		SRC_TR_WIDTH			DST_TR_WIDTH			INT_EN
Default	?	?	?	0	0	0	0	0	0	?	1	1	?	?	?	?	0	1	?	0	1	0	0	0	0	0	0	0	0	0	0	1	

Table 21: (CTL4)

Bits	Name	Type	Reset	Description
63:42	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
41:32	BLOCK_TS	R/W	0x2	Block length
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
28:23	Reserved	R/W	0x0	Reserved. Do not change the reset value.
22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21:20	TT_FC	R/W	0x3	Flow control
19:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:14	SRC_MSIZ	R/W	0x1	Source transaction burst length
13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12:11	DEST_MSIZ	R/W	0x1	Destination transaction burst length
10:9	SINC	R/W	0x0	Source address direction control
8:7	DINC	R/W	0x0	Destination address direction control
6:4	SRC_TR_WIDTH	R/W	0x0	Source transfer width
3:1	DST_TR_WIDTH	R/W	0x0	Destination transfer width
0	INT_EN	R/W	0x1	Interrupt enable

A.1.20 (CFG4)

Channel 4 configuration

Instance Name

CFG4

Offset

0x1A0

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																DEST_PER				SRC_PER				Reserved		Reserved		Reserved		FIFO_MODE	FCMODE
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	?	?	0	0	1	0	0
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												Reserved		Reserved						HS_SEL_SRC	HS_SEL_DST	FIFO_EMPTY	CH_SUSP	CH_PRIOR		Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	?	?	?	1	1	0	0	1	0	0	?	?	?	?	?

Table 22: (CFG4)

Bits	Name	Type	Reset	Description
63:47	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
46:43	DEST_PER	R/W	0x0	Destination hardware interface
42:39	SRC_PER	R/W	0x0	Source hardware interface
38:37	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
36:34	Reserved	R/W	0x1	Reserved. Do not change the reset value.
33	FIFO_MODE	R/W	0x0	FIFO mode control
32	FCMODE	R/W	0x0	Flow control mode
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	Reserved	R/W	0x0	Reserved. Do not change the reset value.
17:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	HS_SEL_SRC	R/W	0x1	Source handshake select
10	HS_SEL_DST	R/W	0x1	Destination handshake select
9	FIFO_EMPTY	R	0x0	FIFO empty status
8	CH_SUSP	R/W	0x0	Channel Suspend control
7:5	CH_PRIOR	R/W	0x4	Channel Priority
4:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.1.21 (SAR5)

Channel 5 source address

Instance Name

SAR5

Offset

0x1B8

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 23: (SAR5)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	SAR	R/W	0x0	Source address

A.1.22 (DAR5)

Channel 5 destination address

Instance Name

DAR5

Offset

0x1C0

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24: (DAR5)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	DAR	R/W	0x0	Destination address

A.1.23 (CTL5)

Channel 5 control

Instance Name

CTL5

Offset

0x1D0

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	
Field	Reserved																					BLOCK_TS											
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	1	0		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved		Reserved						Reserved	TT_FC		Reserved				SRC_MSIZ		Reserved	DEST_MSIZ		SINC		DINC		SRC_TR_WIDTH			DST_TR_WIDTH			INT_EN		
Default	?	?	?	0	0	0	0	0	0	?	1	1	?	?	?	?	0	1	?	0	1	0	0	0	0	0	0	0	0	0	0	1	

Table 25: (CTL5)

Bits	Name	Type	Reset	Description
63:42	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
41:32	BLOCK_TS	R/W	0x2	Block length
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
28:23	Reserved	R/W	0x0	Reserved. Do not change the reset value.
22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21:20	TT_FC	R/W	0x3	Flow control
19:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:14	SRC_MSIZE	R/W	0x1	Source transaction burst length
13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12:11	DEST_MSIZE	R/W	0x1	Destination transaction burst length
10:9	SINC	R/W	0x0	Source address direction control
8:7	DINC	R/W	0x0	Destination address direction control
6:4	SRC_TR_WIDTH	R/W	0x0	Source transfer width
3:1	DST_TR_WIDTH	R/W	0x0	Destination transfer width
0	INT_EN	R/W	0x1	Interrupt enable

A.1.24 (CFG5)

Channel 5 configuration

Instance Name

CFG5

Offset

0x1F8

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																DEST_PER				SRC_PER				Reserved		Reserved		FIFO_MODE		FCMODE	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	?	?	0	0	1	0	0
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												Reserved		Reserved					HS_SEL_SRC		HS_SEL_DST	FIFO_EMPTY	CH_SUSP	CH_PRIOR		Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	?	?	?	1	1	0	0	1	0	1	?	?	?	?	?

Table 26: (CFG5)

Bits	Name	Type	Reset	Description
63:47	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
46:43	DEST_PER	R/W	0x0	Destination hardware interface
42:39	SRC_PER	R/W	0x0	Source hardware interface
38:37	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
36:34	Reserved	R/W	0x1	Reserved. Do not change the reset value.
33	FIFO_MODE	R/W	0x0	FIFO mode control
32	FCMODE	R/W	0x0	Flow control mode
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	Reserved	R/W	0x0	Reserved. Do not change the reset value.
17:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	HS_SEL_SRC	R/W	0x1	Source handshake select
10	HS_SEL_DST	R/W	0x1	Destination handshake select
9	FIFO_EMPTY	R	0x0	FIFO empty status
8	CH_SUSP	R/W	0x0	Channel Suspend control
7:5	CH_PRIOR	R/W	0x5	Channel Priority
4:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.1.25 (SAR6)

Channel 6 source address

Instance Name

SAR6

Offset

0x210

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 27: (SAR6)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	SAR	R/W	0x0	Source address

A.1.26 (DAR6)

Channel 6 destination address

Instance Name

DAR6

Offset

0x218

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28: (DAR6)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	DAR	R/W	0x0	Destination address

A.1.27 (CTL6)

Channel 6 control

Instance Name

CTL6

Offset

0x228

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																					BLOCK_TS										
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	1	0	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved		Reserved						Reserved	TT_FC	Reserved				SRC_MSIZE		Reserved	DEST_MSIZE	SINC		DINC		SRC_TR_WIDTH			DST_TR_WIDTH			INT_EN			
Default	?	?	?	0	0	0	0	0	0	?	1	1	?	?	?	?	0	1	?	0	1	0	0	0	0	0	0	0	0	0	0	1

Table 29: (CTL6)

Bits	Name	Type	Reset	Description
63:42	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
41:32	BLOCK_TS	R/W	0x2	Block length
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
28:23	Reserved	R/W	0x0	Reserved. Do not change the reset value.
22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21:20	TT_FC	R/W	0x3	Flow control
19:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:14	SRC_MSIZE	R/W	0x1	Source transaction burst length
13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12:11	DEST_MSIZE	R/W	0x1	Destination transaction burst length
10:9	SINC	R/W	0x0	Source address direction control
8:7	DINC	R/W	0x0	Destination address direction control
6:4	SRC_TR_WIDTH	R/W	0x0	Source transfer width
3:1	DST_TR_WIDTH	R/W	0x0	Destination transfer width
0	INT_EN	R/W	0x1	Interrupt enable

A.1.28 (CFG6)

Channel 6 configuration

Instance Name

CFG6

Offset

0x250

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																DEST_PER				SRC_PER				Reserved		Reserved		Reserved		FIFO_MODE	FCMODE
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	?	?	0	0	1	0	0
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												Reserved		Reserved						HS_SEL_SRC	HS_SEL_DST	FIFO_EMPTY	CH_SUSP	CH_PRIOR		Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	?	?	?	1	1	0	0	1	1	0	?	?	?	?	?

Table 30: (CFG6)

Bits	Name	Type	Reset	Description
63:47	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
46:43	DEST_PER	R/W	0x0	Destination hardware interface
42:39	SRC_PER	R/W	0x0	Source hardware interface
38:37	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
36:34	Reserved	R/W	0x1	Reserved. Do not change the reset value.
33	FIFO_MODE	R/W	0x0	FIFO mode control
32	FCMODE	R/W	0x0	Flow control mode
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	Reserved	R/W	0x0	Reserved. Do not change the reset value.
17:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	HS_SEL_SRC	R/W	0x1	Source handshake select
10	HS_SEL_DST	R/W	0x1	Destination handshake select
9	FIFO_EMPTY	R	0x0	FIFO empty status
8	CH_SUSP	R/W	0x0	Channel Suspend control
7:5	CH_PRIOR	R/W	0x6	Channel Priority
4:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.1.29 (SAR7)

Channel 7 source address

Instance Name

SAR7

Offset

0x268

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 31: (SAR7)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	SAR	R/W	0x0	Source address

A.1.30 (DAR7)

Channel 7 destination address

Instance Name

DAR7

Offset

0x270

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DAR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 32: (DAR7)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	DAR	R/W	0x0	Destination address

A.1.31 (CTL7)

Channel 7 control

Instance Name

CTL7

Offset

0x280

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	
Field	Reserved																					BLOCK_TS											
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	1	0		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved		Reserved						Reserved	TT_FC		Reserved					SRC_MSIZ		Reserved	DEST_MSIZ		SINC		DINC		SRC_TR_WIDTH			DST_TR_WIDTH			INT_EN	
Default	?	?	?	0	0	0	0	0	0	?	1	1	?	?	?	?	0	1	?	0	1	0	0	0	0	0	0	0	0	0	0	1	

Table 33: (CTL7)

Bits	Name	Type	Reset	Description
63:42	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
41:32	BLOCK_TS	R/W	0x2	Block length
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
28:23	Reserved	R/W	0x0	Reserved. Do not change the reset value.
22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21:20	TT_FC	R/W	0x3	Flow control
19:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:14	SRC_MSIZE	R/W	0x1	Source transaction burst length
13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12:11	DEST_MSIZE	R/W	0x1	Destination transaction burst length
10:9	SINC	R/W	0x0	Source address direction control
8:7	DINC	R/W	0x0	Destination address direction control
6:4	SRC_TR_WIDTH	R/W	0x0	Source transfer width
3:1	DST_TR_WIDTH	R/W	0x0	Destination transfer width
0	INT_EN	R/W	0x1	Interrupt enable

A.1.32 (CFG7)

Channel 7 configuration

Instance Name

CFG7

Offset

0x2A8

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																DEST_PER				SRC_PER				Reserved		Reserved		FIFO_MODE	FCMODE		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	?	?	0	0	1	0	0
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												Reserved		Reserved				HS_SEL_SRC		HS_SEL_DST	FIFO_EMPTY	CH_SUSP	CH_PRIOR		Reserved						
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	?	?	?	1	1	0	0	1	1	1	?	?	?	?	?

Table 34: (CFG7)

Bits	Name	Type	Reset	Description
63:47	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
46:43	DEST_PER	R/W	0x0	Destination hardware interface
42:39	SRC_PER	R/W	0x0	Source hardware interface
38:37	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
36:34	Reserved	R/W	0x1	Reserved. Do not change the reset value.
33	FIFO_MODE	R/W	0x0	FIFO mode control
32	FCMODE	R/W	0x0	Flow control mode
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	Reserved	R/W	0x0	Reserved. Do not change the reset value.
17:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	HS_SEL_SRC	R/W	0x1	Source handshake select
10	HS_SEL_DST	R/W	0x1	Destination handshake select
9	FIFO_EMPTY	R	0x0	FIFO empty status
8	CH_SUSP	R/W	0x0	Channel Suspend control
7:5	CH_PRIOR	R/W	0x7	Channel Priority
4:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.1.33 (RAWTFR)

Raw Status for IntTfr Interrupt

Instance Name

RAWTFR

Offset

0x2C0

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								RAW							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 35: (RAWTFR)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	RAW	R/W	0x0	Raw Status for IntTfr Interrupt

A.1.34 (RAWBLOCK)

Raw Status for IntBlock Interrupt

Instance Name

RAWBLOCK

Offset

0x2C8

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								RAW							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 36: (RAWBLOCK)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	RAW	R/W	0x0	Raw Status for IntBlock Interrupt

A.1.35 (RAWSRCTRAN)

Raw Status for IntSrcTran Interrupt

Instance Name

RAWSRCTRAN

Offset

0x2D0

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								RAW							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 37: (RAWSRCTRAN)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	RAW	R/W	0x0	Raw Status for IntSrcTran Interrupt

A.1.36 (RAWDSTTRAN)

Raw Status for IntDstTran Interrupt

Instance Name

RAWDSTTRAN

Offset

0x2D8

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								RAW							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 38: (RAWDSTTRAN)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	RAW	R/W	0x0	Raw Status for IntDstTran Interrupt

A.1.37 (RAWERR)

Raw Status for IntErr Interrupt

Instance Name

RAWERR

Offset

0x2E0

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							RAW								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 39: (RAWERR)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	RAW	R/W	0x0	Raw Status for IntErr Interrupt

A.1.38 (STATUSTFR)

Status for IntTfr Interrupt

Instance Name

STATUSTFR

Offset

0x2E8

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							STATUS								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 40: (STATUSTFR)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	STATUS	R	0x0	Status for IntTfr Interrupt

A.1.39 (STATUSBLOCK)

Status for IntBlock Interrupt

Instance Name

STATUSBLOCK

Offset

0x2F0

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								STATUS							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 41: (STATUSBLOCK)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	STATUS	R	0x0	Status for IntBlock Interrupt

A.1.40 (STATUSSRCTRAN)

Status for IntSrcTran Interrupt

Instance Name

STATUSSRCTRAN

Offset

0x2F8

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								STATUS							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 42: (STATUSSRCTRAN)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	STATUS	R	0x0	Status for IntSrcTran Interrupt

A.1.41 (STATUSDSTTRAN)

Status for IntDstTran Interrupt

Instance Name
STATUSDSTTRAN**Offset**
0x300

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							STATUS								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0

Table 43: (STATUSDSTTRAN)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	STATUS	R	0x0	Status for IntDstTran Interrupt

A.1.42 (STATUSERR)

Status for IntErr Interrupt

Instance Name
STATUSERR**Offset**
0x308

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							STATUS								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0

Table 44: (STATUSERR)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	STATUS	R	0x0	Status for IntErr Interrupt

A.1.43 (MASKTFR)

Mask for IntTfr Interrupt

Instance Name

MASKTFR

Offset

0x310

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																INT_MASK_WE						INT_MASK									
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 45: (MASKTFR)

Bits	Name	Type	Reset	Description
63:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	INT_MASK_WE	R/W	0x0	Interrupt Mask Write Enable
7:0	INT_MASK	R/W	0x0	Mask for IntTfr Interrupt

A.1.44 (MASKBLOCK)

Mask for IntBlock Interrupt

Instance Name

MASKBLOCK

Offset

0x318

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																INT_MASK_WE						INT_MASK									
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 46: (MASKBLOCK)

Bits	Name	Type	Reset	Description
63:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	INT_MASK_WE	R/W	0x0	Interrupt Mask Write Enable
7:0	INT_MASK	R/W	0x0	Mask for IntBlock Interrupt

A.1.45 (MASKSRCTRAN)

Mask for IntSrcTran Interrupt

Instance Name
MASKSRCTRAN**Offset**
0x320

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																INT_MASK_WE								INT_MASK							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 47: (MASKSRCTRAN)

Bits	Name	Type	Reset	Description
63:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	INT_MASK_WE	R/W	0x0	Interrupt Mask Write Enable
7:0	INT_MASK	R/W	0x0	Mask for IntSrcTran Interrupt

A.1.46 (MASKDSTTRAN)

Mask for IntDstTran Interrupt

Instance Name
MASKDSTTRAN**Offset**
0x328

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																INT_MASK_WE								INT_MASK							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 48: (MASKDSTTRAN)

Bits	Name	Type	Reset	Description
63:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	INT_MASK_WE	R/W	0x0	Interrupt Mask Write Enable
7:0	INT_MASK	R/W	0x0	Mask for IntDstTran Interrupt

A.1.47 (MASKERR)

Mask for IntErr Interrupt

Instance Name

MASKERR

Offset

0x330

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																INT_MASK_WE						INT_MASK									
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 49: (MASKERR)

Bits	Name	Type	Reset	Description
63:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	INT_MASK_WE	R/W	0x0	Interrupt Mask Write Enable
7:0	INT_MASK	R/W	0x0	Mask for IntErr Interrupt

A.1.48 (CLEARTFR)

Clear for IntTfr Interrupt

Instance Name

CLEARTFR

Offset

0x338

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								CLEAR							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 50: (CLEARTFR)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	CLEAR	W	0x0	Clear for IntTfr Interrupt

A.1.49 (CLEARBLOCK)

Clear for IntBlock Interrupt

Instance Name

CLEARBLOCK

Offset

0x340

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								CLEAR							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 51: (CLEARBLOCK)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	CLEAR	W	0x0	Clear for IntBlock Interrupt

A.1.50 (CLEARSRCTAN)

Clear for IntSrcTran Interrupt

Instance Name

CLEARSRCTAN

Offset

0x348

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								CLEAR							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 52: (CLEARSRCTAN)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	CLEAR	W	0x0	Clear for IntSrcTran Interrupt

A.1.51 (CLEARDSTTRAN)

Clear for IntDstTran Interrupt

Instance Name
CLEARDSTTRAN

Offset
0x350

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								CLEAR							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 53: (CLEARDSTTRAN)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	CLEAR	W	0x0	Clear for IntDstTran Interrupt

A.1.52 (CLEARERR)

Clear for IntErr Interrupt

Instance Name
CLEARERR

Offset
0x358

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								CLEAR							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 54: (CLEARERR)

Bits	Name	Type	Reset	Description
63:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	CLEAR	W	0x0	Clear for IntErr Interrupt

A.1.53 (STATUSINT)

Status for each Interrupt type

Instance Name

STATUSINT

Offset

0x360

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	ERR	DSTT	SRCT	BLOCK	TFR
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0

Table 55: (STATUSINT)

Bits	Name	Type	Reset	Description
63:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	ERR	R	0x0	Status for each Interrupt type
3	DSTT	R	0x0	Status for each Interrupt type
2	SRCT	R	0x0	Status for each Interrupt type
1	BLOCK	R	0x0	Status for each Interrupt type
0	TFR	R	0x0	Status for each Interrupt type

A.1.54 (REQSRCREG)

Source Software Transaction Request register

Instance Name

REQSRCREG

Offset

0x368

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																SRC_REQ_WE								SRC_REQ							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 56: (REQSRCREG)

Bits	Name	Type	Reset	Description
63:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	SRC_REQ_WE	R/W	0x0	Source Software Transaction Request write enable
7:0	SRC_REQ	R/W	0x0	Source Software Transaction Request register

A.1.55 (REQDSTREG)

Destination Software Transaction Request register

Instance Name

REQDSTREG

Offset

0x370

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																DST_REQ_WE						DST_REQ									
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 57: (REQDSTREG)

Bits	Name	Type	Reset	Description
63:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	DST_REQ_WE	R/W	0x0	Destination Software Transaction Request write enable
7:0	DST_REQ	R/W	0x0	Destination Software Transaction Request register

A.1.56 (SGLRQSRCREG)

Source Single Transaction Request register

Instance Name

SGLREQSRCREG

Offset

0x378

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																SRC_SGLREQ_WE						SRC_SGLREQ									
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 58: (SGLRQSRCREG)

Bits	Name	Type	Reset	Description
63:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	SRC_SGLREQ_WE	R/W	0x0	Source Single Transaction Request write enable
7:0	SRC_SGLREQ	R/W	0x0	Source Single Transaction Request register

A.1.57 (SGLRQDSTREG)

Destination Single Transaction Request register

Instance Name
SGLREQDSTREG

Offset
0x380

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																DST_SGLREQ_WE						DST_SGLREQ									
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 59: (SGLRQDSTREG)

Bits	Name	Type	Reset	Description
63:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	DST_SGLREQ_WE	R/W	0x0	Destination Single Transaction Request write enable
7:0	DST_SGLREQ	R/W	0x0	Destination Single Transaction Request register

A.1.58 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x388

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																Reserved															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 60: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
63:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.1.59 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x390

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																Reserved															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 61: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
63:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.1.60 (DMACFGREG)

DMA Configuration Register

Instance Name

DMACFGREG

Offset

0x398

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															DMA_EN
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Table 62: (DMACFGREG)

Bits	Name	Type	Reset	Description
63:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	DMA_EN	R/W	0x0	DMA global enable

A.1.61 (CHENREG)

Channel enable register

Instance Name

CHENREG

Offset

0x3A0

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																CH_EN_WE						CH_EN									
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 63: (CHENREG)

Bits	Name	Type	Reset	Description
63:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	CH_EN_WE	R/W	0x0	Channel enable register
7:0	CH_EN	R/W	0x0	Channel enable register

A.1.62 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x3A8

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 64: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
63:32	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.1.63 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x3B0

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	
Field	Reserved																																
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																																Reserved
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	

Table 65: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
63:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.1.64 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x3F8

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Field	Reserved																															
Default	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	1	0	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0

Table 66: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
63:0	Reserved	R	0x3231_372A_4457_1110	Reserved. Do not change the reset value.

A.2 USBC Register Information

Table 67: USBC Register Summary

Offset	Name	Description	Details
0x000	ID	The ID Register Identifies the USB-HS 2.0 Core and Its Revision.	Page: 51
0x004	HWGENERAL	General Hardware Parameters as Defined	Page: 51
0x008	HWHOST	Host Hardware Parameters as Defined	Page: 52
0x00C	HWDEVICE	Device Hardware Parameters as Defined	Page: 52
0x010	HWTXBUF	TX Buffer Hardware Parameters	Page: 53
0x014	HWRXBUF	RX Bbuffer Hardware Parameters	Page: 53
0x080	GPTIMER0LD		Page: 53
0x084	GPTIMER0CTRL		Page: 54
0x088	GPTIMER1LD		Page: 54
0x08C	GPTIMER1CTRL		Page: 55
0x090	SBUSCFG	This Register Contains the Control for The System Bus Interface	Page: 55
0x100	CAPLENGTH_HCVERSION	Indicate Which Offset to Add to the Register Base Address	Page: 56
0x104	HCSPARAMS	Port Steering Logic Capabilities	Page: 56
0x108	HCCPARAMS	Identifies Multiple Mode Control	Page: 57
0x120	DCVERSION		Page: 57
0x124	DCCPARAMS	Describe the Overall Host/Device Capability of the Controller.	Page: 58
0x140	USBCMD	The Serial Bus Host/Device Controller Executes the Command Indicated in This Register.	Page: 58
0x144	USBSTS	Indicates Various States of the Controller and any Pending Interrupts	Page: 59
0x148	USBINTR	Interrupt Sources	Page: 60
0x14C	FRINDEX	Used by the Host Controller to Index the Periodic Frame List	Page: 61
0x154	PERIODICLISTBASE_HOST	Host Controller FrameList Base Address	Page: 62
0x154	PERIODICLISTBASE_DEVICE	Device Controller Sub Device Address	Page: 62
0x158	ASYNCLISTADDR_HOST	Contains the Address of the Top of the Endpoint List in System Memory	Page: 62

Table 67: USBC Register Summary

Offset	Name	Description	Details
0x158	ASYNCLISTADDR_DEVICE	Contains the Address of the Top of the Endpoint List in System Memory	Page: 63
0x15C	TTCTRL	Contains Parameters Needed for Internal TT Operations.	Page: 63
0x160	BURSTSIZE	Controls the Burst Size Used during Data Movement	Page: 64
0x164	TXFILLTUNING		Page: 64
0x168	TXTTFILLTUNING		Page: 65
0x16C	IC_USB	Enable and Controls the IC_USB FS/LS Transceiver.	Page: 65
0x170	ULPI_VIEWPORT	Provides Indirect Access to the ULPI PHY Register Set.	Page: 66
0x178	ENDPTNAK		Page: 67
0x17C	ENDPTNAKEN		Page: 67
0x184	PORTSC1		Page: 67
0x1A4	OTGSC	This Register Only Exists in a OTG Implementation.	Page: 69
0x1A8	USBMODE		Page: 70
0x1AC	ENDPTSETUPSTAT	Setup Endpoint Status	Page: 71
0x1B0	ENDPTPRIME		Page: 71
0x1B4	ENDPTFLUSH		Page: 71
0x1B8	ENDPTSTAT		Page: 72
0x1BC	ENDPTCOMPLETE		Page: 72
0x1C0	ENDPTCTRL0	Every Device Will Implement Endpoint0 as a Control Endpoint.	Page: 72
0x1C4	ENDPTCTRL1		Page: 73
0x1C8	ENDPTCTRL2		Page: 74
0x1CC	ENDPTCTRL3		Page: 75
0x1D0	ENDPTCTRL4		Page: 76
0x1D4	ENDPTCTRL5		Page: 76
0x1D8	ENDPTCTRL6		Page: 77
0x1DC	ENDPTCTRL7		Page: 78
0x1E0	ENDPTCTRL8		Page: 79

A.2.1 The ID Register Identifies the USB-HS 2.0 Core and Its Revision. (ID)

Instance Name

ID

Offset

0x000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	CIVERSION			VERSION				REVISION				TAG				Reserved		NID						Reserved		ID						
Default	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	1	?	?	1	1	1	0	1	0	?	?	0	0	0	1	0	1

Table 68: The ID Register Identifies the USB-HS 2.0 Core and Its Revision. (ID)

Bits	Name	Type	Reset	Description
31:29	CIVERSION	R	0x7	Identifies the CI version
28:25	VERSION	R	0x2	Identifies the version of the core
24:21	REVISION	R	0x0	Revision number of the core
20:16	TAG	R	0x1	Identifies the tag of the core
15:14	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
13:8	NID	R	0x3A	Complement of ID
7:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5:0	ID	R	0x5	Configuration number

A.2.2 General Hardware Parameters as Defined (HWGENERAL)

Instance Name

HWGENERAL

Offset

0x004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																				SM		PHYM			PHYW		BWT	CLKC		RT	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	0	0	1	1	0		0	0	1	0

Table 69: General Hardware Parameters as Defined (HWGENERAL)

Bits	Name	Type	Reset	Description
31:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11:10	SM	R	0x1	VUSB_HS_PHY_SERIAL
9:6	PHYM	R	0x3	VUSB_HS_PHY_TYPE
5:4	PHYW	R	0x0	VUSB_HS_PHY16_8
3	BWT	R	0x0	

Table 69: General Hardware Parameters as Defined (HWGENERAL)

Bits	Name	Type	Reset	Description
2:1	CLKC	R	0x2	VUSB_HS_CLOCK_CONFIGURATION
0	RT	R	0x1	VUSB_HS_RESET_TYPE

A.2.3 Host Hardware Parameters as Defined (HWHOST)

Instance Name
HWHOST

Offset
0x008

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TTPER								TTASY								Reserved												NPORT		HC	
Default	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	1

Table 70: Host Hardware Parameters as Defined (HWHOST)

Bits	Name	Type	Reset	Description
31:24	TTPER	R	0x10	VUSB_HS_TT_PERIODIC_CONTEXTS
23:16	TTASY	R	0x2	VUSB_HS_TT_ASYNC_CONTEXTS
15:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:1	NPORT	R	0x0	VUSB_HS_NUM_PORT-1
0	HC	R	0x1	VUSB_HS_HOST

A.2.4 Device Hardware Parameters as Defined (HWDEVICE)

Instance Name
HWDEVICE

Offset
0x00C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										DEVEP			DO		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	0	0	0	1	

Table 71: Device Hardware Parameters as Defined (HWDEVICE)

Bits	Name	Type	Reset	Description
31:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5:1	DEVEP	R	0x8	VUSB_HS_DEV_EP
0	DC	R	0x1	Device capable

A.2.5 TX Buffer Hardware Parameters (HWTXBUF)

Instance Name
HWTXBUF

Offset
0x010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								TXCHANADD								TXADD								TXBURST							
Default	?	?	?	?	?	?	?	?	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0

Table 72: TX Buffer Hardware Parameters (HWTXBUF)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23:16	TXCHANADD	R	0x6	VUSB_HS_TX_CHAN_ADD
15:8	TXADD	R	0x9	VUSB_HS_TX_ADD
7:0	TXBURST	R	0x8	VUSB_HS_TX_BURST

A.2.6 RX Bbuffer Hardware Parameters (HWRXBUF)

Instance Name
HWRXBUF

Offset
0x014

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																RXADD								RXBURST								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

Table 73: RX Bbuffer Hardware Parameters (HWRXBUF)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	RXADD	R	0x8	VUSB_HS_RX_ADD
7:0	RXBURST	R	0x8	VUSB_HS_RX_BURST

A.2.7 (GPTIMER0LD)

Instance Name
GPTIMER0LD

Offset
0x080

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								GPTLD																							
Default	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 74: (GPTIMER0LD)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23:0	GPTLD	R/W	0x0	

A.2.8 (GPTIMER0CTRL)

Instance Name
GPTIMER0CTRL

Offset
0x084

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GPTRUN	GPTRST	Reserved					GPTMODE	GPTCNT																							
Default	0	0	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 75: (GPTIMER0CTRL)

Bits	Name	Type	Reset	Description
31	GPTRUN	R/W	0x0	
30	GPTRST	R	0x0	
29:25	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
24	GPTMODE	R/W	0x0	
23:0	GPTCNT	R	0x0	

A.2.9 (GPTTIMER1LD)

Instance Name
GPTTIMER1LD

Offset
0x088

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								GPTLD																							
Default	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 76: (GPTTIMER1LD)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23:0	GPTLD	R/W	0x0	

A.2.10 (GPTIMER1CTRL)

Instance Name
GPTIMER1CTRL

Offset
0x08C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GPTRUN	GPTRST	Reserved					GPTMODE	GPTCNT																							
Default	0	0	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 77: (GPTIMER1CTRL)

Bits	Name	Type	Reset	Description
31	GPTRUN	R/W	0x0	
30	GPTRST	R	0x0	
29:25	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
24	GPTMODE	R/W	0x0	
23:0	GPTCNT	R	0x0	

A.2.11 This Register Contains the Control for The System Bus Interface (SBUSCFG)

Instance Name
SBUSCFG

Offset
0x090

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														AHBBRST	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0		0

Table 78: This Register Contains the Control for The System Bus Interface (SBUSCFG)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	AHBBRST	R/W	0x0	AMBA AHB burst configuration

A.2.12 Indicate Which Offset to Add to the Register Base Address (CAPLENGTH_HCIVERSION)

Instance Name
CAPLENGTH_HCIVERSION

Offset
0x100

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	HCIVERSION																Reserved								CAPLENGTH							
Default	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	0	1	0	0	0	0	0	0

Table 79: Indicate Which Offset to Add to the Register Base Address (CAPLENGTH_HCIVERSION)

Bits	Name	Type	Reset	Description
31:16	HCIVERSION	R	0x100	
15:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	CAPLENGTH	R	0x40	

A.2.13 Port Steering Logic Capabilities (HCSPARAMS)

Instance Name
HCSPARAMS

Offset
0x104

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved				N_TT				N_PTT				Reserved		PI	N_CC				N_PCC				Reserved		PPC	N_PORTS					
Default	?	?	?	?	0	0	0	0	0	0	0	0	?	?	?	1	0	0	0	0	0	0	0	0	?	?	?	0	0	0	0	0

Table 80: Port Steering Logic Capabilities (HCSPARAMS)

Bits	Name	Type	Reset	Description
31:28	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
27:24	N_TT	R	0x0	Number of Transaction Translators
23:20	N_PTT	R	0x0	Number of Ports per Transaction Translator
19:17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16	PI	R	0x1	Port Indicator
15:12	N_CC	R	0x0	Number of Companion Controller
11:8	N_PCC	R	0x0	Number of Ports per Companion Controller
7:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	PPC	R	0x0	Port Power Control
3:0	N_PORTS	R	0x0	Number of downstream ports

A.2.14 Identifies Multiple Mode Control (HCCPARAMS)

Instance Name
HCCPARAMS

Offset
0x108

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																EECP								IST				Reserved	ASP	PFL	ADC
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	?	1	1	0

Table 81: Identifies Multiple Mode Control (HCCPARAMS)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	EECP	R	0x0	
7:4	IST	R	0x0	Isochronous Scheduling Threshold
3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2	ASP	R	0x1	Asynchronous Schedule Park Capability
1	PFL	R	0x1	Programmable Frame List Flag
0	ADC	R	0x0	64-bit Addressing Capability

A.2.15 (DCIVERSION)

Instance Name
DCIVERSION

Offset
0x120

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																DCIVERSION															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 82: (DCIVERSION)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	DCIVERSION	R	0x1	

A.2.16 Describe the Overall Host/Device Capability of the Controller. (DCCPARAMS)

Instance Name
DCCPARAMS

Offset
0x124

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							HC	DC	Reserved	DEN					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0		?	?	0	0	0	0

Table 83: Describe the Overall Host/Device Capability of the Controller. (DCCPARAMS)

Bits	Name	Type	Reset	Description
31:9	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
8	HC	R	0x0	Host Capable
7	DC	R	0x0	Device Capable
6:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4:0	DEN	R	0x1	Device Endpoint Number

A.2.17 The Serial Bus Host/Device Controller Executes the Command Indicated in This Register. (USBCMD)

Instance Name
USBCMD

Offset
0x140

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								ITC								FS2	ATDTW	SUTW	Reserved	ASPE	Reserved	ASP1	ASP0	LR	IAA	ASE	PSE	FS1	FS0	RST	RS
Default	?	?	?	?	?	?	?	?	0	0	0	0	1	0	0	0	0	0	0	?	1	?	1	1	0	0	0	0	0	0	0	0

Table 84: The Serial Bus Host/Device Controller Executes the Command Indicated in This Register. (USBCMD)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23:16	ITC	R/W	0x8	Interrupt Threshold Control
15	FS2	R/W	0x0	Frame List Size
14	ATDTW	R/W	0x0	Add dTD TripWire
13	SUTW	R/W	0x0	Setup TripWire
12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	ASPE	R/W	0x1	Asynchronous Schedule Park Mode Enable

Table 84: The Serial Bus Host/Device Controller Executes the Command Indicated in This Register. (USBCMD)

Bits	Name	Type	Reset	Description
10	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
9	ASP1	R/W	0x1	Asynchronous Schedule Park Mode Count
8	ASP0	R/W	0x1	Asynchronous Schedule Park Mode Count
7	LR	R	0x0	Light Host/Device Controller Reset
6	IAA	R/W	0x0	Interrupt on Async Advance Doorbell
5	ASE	R/W	0x0	Asynchronous Schedule Enable
4	PSE	R/W	0x0	Periodic Schedule Enable
3	FS1	R/W	0x0	Frame List Size
2	FS0	R/W	0x0	Frame List Size
1	RST	R/W	0x0	Controller Reset
0	RS	R/W	0x0	Run/Stop

A.2.18 Indicates Various States of the Controller and any Pending Interrupts (USBSTS)

Instance Name
USBSTS

Offset
0x144

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved						TI1	TI0	Reserved				UPI	UAI	Reserved	NAKI	AS	PS	RCL	HCH	UALTI	ULPII	Reserved	SLI	SRI	URI	AAI	SEI	FRI	PCI	UEI	UI
Default	?	?	?	?	?	?	0	0	?	?	?	?	0	0	?	0	0	0	0	1	0	0	?	0	0	0	0	0	0	0	0	0

Table 85: Indicates Various States of the Controller and any Pending Interrupts (USBSTS)

Bits	Name	Type	Reset	Description
31:26	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
25	TI1	R/W	0x0	General Purpose Timer Interrupt 1
24	TI0	R/W	0x0	General Purpose Timer Interrupt 0
23:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19	UPI	R/W	0x0	USB Host Periodic Interrupt
18	UAI	R/W	0x0	USB Host Asynchronous Interrupt
17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16	NAKI	R	0x0	NAK Interrupt

Table 85: Indicates Various States of the Controller and any Pending Interrupts (USBSTS)

Bits	Name	Type	Reset	Description
15	AS	R	0x0	Asynchronous Schedule Status
14	PS	R	0x0	Periodic Schedule Status
13	RCL	R	0x0	Reclamation
12	HCH	R	0x1	HCHalted
11	UALTI	R	0x0	ULPI alt_int Interrupt
10	ULPII	R/W	0x0	ULPI Interrupt
9	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
8	SLI	R/W	0x0	DCSuspend
7	SRI	R/W	0x0	SOF Received
6	URI	R/W	0x0	USB Reset Received
5	AAI	R/W	0x0	Interrupt on Async Advance
4	SEI	R/W	0x0	System Error
3	FRI	R/W	0x0	Frame List Rollover
2	PCI	R/W	0x0	Port Change Detect
1	UEI	R/W	0x0	USB Error Interrupt
0	UI	R/W	0x0	USB Interrupt

A.2.19 Interrupt Sources (USBINTR)

Instance Name
USBINTR

Offset
0x148

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved						TIE1	TIE0	Reserved				UPIE	UAIE	Reserved	NAKE	Reserved				UALTIE	ULPIE	Reserved	SLE	SRE	URE	AAE	SEE	FRE	PCE	UEE	UE
Default	?	?	?	?	?	?	0	0	?	?	?	?	0	0	?	0	?	?	?	?	0	0	?	0	0	0	0	0	0	0	0	0

Table 86: Interrupt Sources (USBINTR)

Bits	Name	Type	Reset	Description
31:26	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
25	TIE1	R/W	0x0	General Purpose Timer Interrupt Enable 1
24	TIE0	R/W	0x0	General Purpose Timer Interrupt Enable 0
23:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 86: Interrupt Sources (USBINTR)

Bits	Name	Type	Reset	Description
19	UPIE	R/W	0x0	USB Host Periodic Interrupt Enable
18	UAIE	R/W	0x0	USB Host Asynchronous Interrupt Enable
17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16	NAKE	R	0x0	NAK Interrupt Enable
15:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	UALTIE	R/W	0x0	ULPI alt_int Interrupt Enable
10	ULPIE	R/W	0x0	ULPI Interrupt Enable
9	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
8	SLE	R/W	0x0	DCSuspend Interrupt Enable
7	SRE	R/W	0x0	SOF Received Interrupt Enable
6	URE	R/W	0x0	USB Reset Received Interrupt Enable
5	AAE	R/W	0x0	Interrupt on Async Advance Enable
4	SEE	R/W	0x0	System Error Interrupt Enable
3	FRE	R/W	0x0	Frame List Rollover Interrupt Enable
2	PCE	R/W	0x0	Port Change Detect Interrupt Enable
1	UEE	R/W	0x0	USB Error Interrupt
0	UE	R/W	0x0	USB Interrupt Enable

A.2.20 Used by the Host Controller to Index the Periodic Frame List (FRINDEX)

Instance Name

FRINDEX

Offset

0x14C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																		FRINDEX													
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 87: Used by the Host Controller to Index the Periodic Frame List (FRINDEX)

Bits	Name	Type	Reset	Description
31:14	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
13:0	FRINDEX	R/W	0x0	Frame Index

A.2.21 Host Controller FrameList Base Address (PERIODICLISTBASE_HOST)

Instance Name
PERIODICLISTBASE_HOST

Offset
0x154

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	PERBASE																				Reserved											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?

Table 88: Host Controller FrameList Base Address (PERIODICLISTBASE_HOST)

Bits	Name	Type	Reset	Description
31:12	PERBASE	R/W	0x0	Periodic List Base Address
11:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.2.22 Device Controller Sub Device Address (PERIODICLISTBASE_DEVICE)

Instance Name
PERIODICLISTBASE_DEVICE

Offset
0x154

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	USBADR							USBADRA	Reserved																							
Default	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Table 89: Device Controller Sub Device Address (PERIODICLISTBASE_DEVICE)

Bits	Name	Type	Reset	Description
31:25	USBADR	R/W	0x0	USB Device address
24	USBADRA	R/W	0x0	Device address advance
23:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.2.23 Contains the Address of the Top of the Endpoint List in System Memory (ASYNCLISTADDR_HOST)

Instance Name
ASYNCLISTADDR_HOST

Offset
0x158

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ASYBASE																				Reserved											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?

Table 90: Contains the Address of the Top of the Endpoint List in System Memory (ASYNCLISTADDR_HOST)

Bits	Name	Type	Reset	Description
31:5	ASYBASE	R/W	0x0	Asynchronous List Base Address
4:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.2.24 Contains the Address of the Top of the Endpoint List in System Memory (ASYNCLISTADDR_DEVICE)

Instance Name
ASYNCLISTADDR_DEVICE

Offset
0x158

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	EPBASE																				Reserved											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	

Table 91: Contains the Address of the Top of the Endpoint List in System Memory (ASYNCLISTADDR_DEVICE)

Bits	Name	Type	Reset	Description
31:11	EPBASE	R/W	0x0	Endpoint List Base Address
10:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.2.25 Contains Parameters Needed for Internal TT Operations. (TTCTRL)

Instance Name
TTCTRL

Offset
0x15C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved	TTHA								Reserved																						TTAC	TTAS
Default	?	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	

Table 92: Contains Parameters Needed for Internal TT Operations. (TTCTRL)

Bits	Name	Type	Reset	Description
31	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
30:24	TTHA	R/W	0x0	Internal TT Hub Address Representation
23:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	TTAC	R/W	0x0	Embedded TT Asynchronous Buffers Clear
0	TTAS	R	0x0	Embedded TT Async Buffers Status

A.2.26 Controls the Burst Size Used during Data Movement (BURSTSIZE)

Instance Name

BURSTSIZE

Offset

0x160

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																TXPBURST								RXPBURST								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

Table 93: Controls the Burst Size Used during Data Movement (BURSTSIZE)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:8	TXPBURST	R/W	0x8	Programmable TX Burst Length
7:0	RXPBURST	R/W	0x8	Programmable RX Burst Length

A.2.27 (TXFILLTUNING)

Instance Name

TXFILLTUNING

Offset

0x164

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved										TXFIFOTHRES						Reserved			TXSCHHEALTH				Reserved	TXSCHOH							
Default	?	?	?	?	?	?	?	?	?	?	0	0	0	0	1	0	?	?	?	0	0	0	0	0	?	0	0	0	0	0	0	0

Table 94: (TXFILLTUNING)

Bits	Name	Type	Reset	Description
31:22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21:16	TXFIFOTHRES	R/W	0x2	FIFO Burst Threshold
15:13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12:8	TXSCHHEALTH	R/W	0x0	Scheduler Health Counter
7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
6:0	TXSCHOH	R/W	0x0	Scheduler Overhead

Table 96: Enable and Controls the IC_USB FS/LS Transceiver. (IC_USB)

Bits	Name	Type	Reset	Description
18:16	IC_VDD5	R/W	0x0	Inter-Chip voltage selection
15	IC4	R/W	0x0	Inter-Chip transceiver enable
14:12	IC_VDD4	R/W	0x0	Inter-Chip voltage selection
11	IC3	R/W	0x0	Inter-Chip transceiver enable
10:8	IC_VDD3	R/W	0x0	Inter-Chip voltage selection
7	IC2	R/W	0x0	Inter-Chip transceiver enable
6:4	IC_VDD2	R/W	0x0	Inter-Chip voltage selection
3	IC1	R/W	0x0	Inter-Chip transceiver enable
2:0	IC_VDD1	R/W	0x0	Inter-Chip voltage selection

A.2.30 Provides Indirect Access to the ULPI PHY Register Set. (ULPI_VIEWPORT)

Instance Name
ULPI_VIEWPORT

Offset
0x170

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ULPIWU	ULPIRUN	ULPIRW	Reserved	ULPISS		ULPIPORT	ULPIADDR								ULPIDATRD								ULPIDATWR								
Default	0	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 97: Provides Indirect Access to the ULPI PHY Register Set. (ULPI_VIEWPORT)

Bits	Name	Type	Reset	Description
31	ULPIWU	R/W	0x0	ULPI Wakeup
30	ULPIRUN	R/W	0x0	ULPIRUN
29	ULPIRW	R/W	0x0	ULPI Read/Write Control
28	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
27	ULPISS	R/W	0x0	ULPI Synchronous State
26:24	ULPIPORT	R/W	0x0	ULPI Port Number
23:16	ULPIADDR	R/W	0x0	ULPI Data Address
15:8	ULPIDATRD	R	0x0	ULPI Data Read
7:0	ULPIDATWR	R/W	0x0	ULPI Data Write

[illegible]

Bits	Name	Type	Reset	Description
31:16	EPTN	R/W	0x0	TX Endpoint NAK
15:0	EPRN	R/W	0x0	RX Endpoint NAK

A.2.32 (ENDPTNAKEN)

[illegible]

Bits	Name	Type	Reset	Description
31:16	EPTNE	R/W	0x0	TX Endpoint NAK enable
15:0	EPRNE	R/W	0x0	RX Endpoint NAK enable

A.2.33 (PORTSC1)

[illegible]

Bits	Name	Type	Reset	Description
31:30	PTS	R/W	0x3	Parallel Transceiver Select
29	STS	R/W	0x1	Serial Transceiver Select

Table 100: (PORTSC1)

Bits	Name	Type	Reset	Description
28	PTW	R/W	0x0	Parallel Transceiver Width
27:26	PSPD	R	0x3	Port Speed
25	PTS2	R/W	0x0	Parallel Transceiver Select
24	PFSC	R/W	0x0	Port Force Full Speed Connect
23	PHCD	R/W	0x0	PHY Low Power Clock Disable
22	WKOC	R/W	0x0	Wake on Over-current Enable
21	WKDS	R/W	0x0	Wake on Disconnect Enable
20	WKN	R/W	0x0	Wake on Connect Enable
19:16	PTC	R/W	0x0	Port Test Control
15:14	PIC	R/W	0x0	Port Indicator Control
13	PO	R	0x0	Port Owner
12	PP	R/W	0x0	Port Power
11:10	LS	R	0x0	Line Status
9	HSP	R	0x0	High-Speed Port
8	PR	R/W	0x0	Port Reset
7	SUSP	R/W	0x0	Suspend
6	FPR	R/W	0x0	Force Port Resume
5	OCC	R/W	0x0	Over-current Change
4	OCA	R	0x0	Over-current Active
3	PEC	R/W	0x0	Port Enabled Change
2	PE	R/W	0x0	Port Enabled
1	CSC	R/W	0x0	Connect Status Change
0	CCS	R	0x0	Current Connect Status

A.2.34 This Register Only Exists in a OTG Implementation. (OTGSC)

Instance Name

OTGSC

Offset

0x1A4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved	DPIE	1MSE	BSEIE	BSVIE	ASVIE	AVVIE	IDIE	Reserved	DPIS	1MSS	BSEIS	BSVIS	ASVIS	AVVIS	IDIS	Reserved	DPS	1MST	BSE	BSV	ASV	AVV	ID	HABA	HADP	IDPU	DP	OT	HAAR	VC	VD
Default	?	0	0	0	0	0	0	0	?	0	0	0	0	0	0	0	?	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Table 101: This Register Only Exists in a OTG Implementation. (OTGSC)

Bits	Name	Type	Reset	Description
31	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
30	DPIE	R/W	0x0	Data Pulse Interrupt Enable
29	1MSE	R/W	0x0	1 millisecond timer Interrupt Enable
28	BSEIE	R/W	0x0	B Session End Interrupt Enable
27	BSVIE	R/W	0x0	B Session Valid Interrupt Enable
26	ASVIE	R/W	0x0	A Session Valid Interrupt Enable
25	AVVIE	R/W	0x0	A VBus Valid Interrupt Enable
24	IDIE	R/W	0x0	USB ID Interrupt Enable
23	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
22	DPIS	R/W	0x0	Data Pulse Interrupt Status
21	1MSS	R/W	0x0	1 millisecond timer Interrupt Status
20	BSEIS	R/W	0x0	B Session End Interrupt Status
19	BSVIS	R/W	0x0	B Session Valid Interrupt Status
18	ASVIS	R/W	0x0	A Session Valid Interrupt Status
17	AVVIS	R/W	0x0	Frame Index
16	IDIS	R/W	0x0	USB ID Interrupt Status
15	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
14	DPS	R	0x0	Data Bus Pulsing Status
13	1MST	R	0x0	1 millisecond timer toggle
12	BSE	R	0x0	B Session End
11	BSV	R	0x0	B Session Valid
10	ASV	R	0x0	A Session Valid
9	AVV	R	0x0	A VBus Valid

Table 101: This Register Only Exists in a OTG Implementation. (OTGSC)

Bits	Name	Type	Reset	Description
8	ID	R	0x0	USB ID
7	HABA	R/W	0x0	Hardware Assist B-Disconnect to A-connect
6	HADP	R/W	0x0	Hardware Assist Data-Pulse
5	IDPU	R/W	0x1	ID Pullup
4	DP	R/W	0x0	Data Pulsing
3	OT	R/W	0x0	OTG Termination
2	HAAR	R/W	0x0	Hardware Assist Auto-Reset
1	VC	R/W	0x0	VBUS Charge
0	VD	R/W	0x0	VBUS Discharge

A.2.35 (USBMODE)

Instance Name
USBMODE

Offset
0x1A8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																SRT	Reserved										VBPS	SDIS	SLOM	ES	CM
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0

Table 102: (USBMODE)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	SRT	R/W	0x0	Shorten Reset Time
14:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5	VBPS	R/W	0x0	Vbus Power Select
4	SDIS	R/W	0x0	Stream Disable Mode
3	SLOM	R/W	0x0	Setup Lockout Mode
2	ES	R/W	0x0	Endian Select
1:0	CM	R/W	0x0	Controller Mode

A.2.36 Setup Endpoint Status (ENDPTSETUPSTAT)

Instance Name
ENDPTSETUPSTAT

Offset
0x1AC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																ENDPTSETUPSTAT															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 103: Setup Endpoint Status (ENDPTSETUPSTAT)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	ENDPTSETUPSTAT	R/W	0x0	Setup Endpoint Status

A.2.37 (ENDPTPRIME)

Instance Name
ENDPTPRIME

Offset
0x1B0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	PETB																PERB															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 104: (ENDPTPRIME)

Bits	Name	Type	Reset	Description
31:16	PETB	R/W	0x0	Prime Endpoint Transmit Buffer
15:0	PERB	R/W	0x0	Prime Endpoint Receive Buffer

A.2.38 (ENDPTFLUSH)

Instance Name
ENDPTFLUSH

Offset
0x1B4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	FETB																FERB															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 105: (ENDPTFLUSH)

Bits	Name	Type	Reset	Description
31:16	FETB	R/W	0x0	Flush Endpoint Transmit Buffer
15:0	FERB	R/W	0x0	Flush Endpoint Receive Buffer

A.2.39 (ENDPTSTAT)

Instance Name
ENDPTSTAT

Offset
0x1B8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ETBR																ERBR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 106: (ENDPTSTAT)

Bits	Name	Type	Reset	Description
31:16	ETBR	R	0x0	Endpoint Transmit Buffer Ready
15:0	ERBR	R	0x0	Endpoint Receive Buffer Ready

A.2.40 (ENDPTCOMPLETE)

Instance Name
ENDPTCOMPLETE

Offset
0x1BC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ETCE																ERCE															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 107: (ENDPTCOMPLETE)

Bits	Name	Type	Reset	Description
31:16	ETCE	R/W	0x0	Endpoint Transmit Complete Event
15:0	ERCE	R/W	0x0	Endpoint Receive Complete Event

A.2.41 Every Device Will Implement Endpoint0 as a Control Endpoint. (ENDPTCTRL0)

Instance Name
ENDPTCTRL0

Offset
0x1C0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								TXE	Reserved			TXT		Reserved	TXS	Reserved								RXE	Reserved			RXT		Reserved	RXS
Default	?	?	?	?	?	?	?	?	1	?	?	?	0	0	?	1	?	?	?	?	?	?	?	?	1	?	?	?	0	0	?	0

Table 108: Every Device Will Implement Endpoint0 as a Control Endpoint. (ENDPTCTRL0)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	TXE	R	0x1	TX Endpoint Enable
22:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	TXT	R	0x0	TX Endpoint Type
17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16	TXS	R/W	0x1	TX Endpoint Stall
15:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	RXE	R	0x1	RX Endpoint Enable
6:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:2	RXT	R	0x0	RX Endpoint Type
1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	RXS	R/W	0x0	RX Endpoint Stall

A.2.42 (ENDPTCTRL1)

Instance Name
ENDPTCTRL1

Offset
0x1C4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Default	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0

Table 109: (ENDPTCTRL1)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	TXE	R/W	0x0	TX Endpoint Enable
22	TXR	R/W	0x0	TX Data Toggle Reset
21	TXI	R/W	0x0	TX Data Toggle Inhibit
20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	TXT	R/W	0x0	TX Endpoint Type
17	TXD	R/W	0x0	TX Endpoint Data Source
16	TXS	R/W	0x0	TX Endpoint Stall

Table 109: (ENDPTCTRL1)

Bits	Name	Type	Reset	Description
15:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	RXE	R/W	0x0	RX Endpoint Enable
6	RXR	R/W	0x0	RX Data Toggle Reset
5	RXI	R/W	0x0	RX Data Toggle Inhibit
4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:2	RXT	R/W	0x0	RX Endpoint Type
1	RXD	R/W	0x0	RX Endpoint Data Sink
0	RXS	R/W	0x0	RX Endpoint Stall

A.2.43 (ENDPTCTRL2)

Instance Name
 ENDPTCTRL2

Offset
 0x1C8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Default	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0

Table 110: (ENDPTCTRL2)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	TXE	R/W	0x0	TX Endpoint Enable
22	TXR	R/W	0x0	TX Data Toggle Reset
21	TXI	R/W	0x0	TX Data Toggle Inhibit
20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	TXT	R/W	0x0	TX Endpoint Type
17	TXD	R/W	0x0	TX Endpoint Data Source
16	TXS	R/W	0x0	TX Endpoint Stall
15:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	RXE	R/W	0x0	RX Endpoint Enable
6	RXR	R/W	0x0	RX Data Toggle Reset
5	RXI	R/W	0x0	RX Data Toggle Inhibit

Table 110: (ENDPTCTRL2)

Bits	Name	Type	Reset	Description
4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:2	RXT	R/W	0x0	RX Endpoint Type
1	RXD	R/W	0x0	RX Endpoint Data Sink
0	RXS	R/W	0x0	RX Endpoint Stall

A.2.44 (ENDPTCTRL3)

Instance Name
ENDPTCTRL3

Offset
0x1CC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Default	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0

Table 111: (ENDPTCTRL3)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	TXE	R/W	0x0	TX Endpoint Enable
22	TXR	R/W	0x0	TX Data Toggle Reset
21	TXI	R/W	0x0	TX Data Toggle Inhibit
20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	TXT	R/W	0x0	TX Endpoint Type
17	TXD	R/W	0x0	TX Endpoint Data Source
16	TXS	R/W	0x0	TX Endpoint Stall
15:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	RXE	R/W	0x0	RX Endpoint Enable
6	RXR	R/W	0x0	RX Data Toggle Reset
5	RXI	R/W	0x0	RX Data Toggle Inhibit
4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:2	RXT	R/W	0x0	RX Endpoint Type
1	RXD	R/W	0x0	RX Endpoint Data Sink
0	RXS	R/W	0x0	RX Endpoint Stall

A.2.45 (ENDPTCTRL4)

Instance Name
ENDPTCTRL4

Offset
0x1D0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Default	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0

Table 112: (ENDPTCTRL4)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	TXE	R/W	0x0	TX Endpoint Enable
22	TXR	R/W	0x0	TX Data Toggle Reset
21	TXI	R/W	0x0	TX Data Toggle Inhibit
20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	TXT	R/W	0x0	TX Endpoint Type
17	TXD	R/W	0x0	TX Endpoint Data Source
16	TXS	R/W	0x0	TX Endpoint Stall
15:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	RXE	R/W	0x0	RX Endpoint Enable
6	RXR	R/W	0x0	RX Data Toggle Reset
5	RXI	R/W	0x0	RX Data Toggle Inhibit
4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:2	RXT	R/W	0x0	RX Endpoint Type
1	RXD	R/W	0x0	RX Endpoint Data Sink
0	RXS	R/W	0x0	RX Endpoint Stall

A.2.46 (ENDPTCTRL5)

Instance Name
ENDPTCTRL5

Offset
0x1D4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved									TXE	TXR	TXI	Reserved	TXT		TXD	TXS	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Default	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0	

Table 113: (ENDPTCTRL5)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	TXE	R/W	0x0	TX Endpoint Enable
22	TXR	R/W	0x0	TX Data Toggle Reset
21	TXI	R/W	0x0	TX Data Toggle Inhibit
20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	TXT	R/W	0x0	TX Endpoint Type
17	TXD	R/W	0x0	TX Endpoint Data Source
16	TXS	R/W	0x0	TX Endpoint Stall
15:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	RXE	R/W	0x0	RX Endpoint Enable
6	RXR	R/W	0x0	RX Data Toggle Reset
5	RXI	R/W	0x0	RX Data Toggle Inhibit
4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:2	RXT	R/W	0x0	RX Endpoint Type
1	RXD	R/W	0x0	RX Endpoint Data Sink
0	RXS	R/W	0x0	RX Endpoint Stall

A.2.47 (ENDPTCTRL6)

Instance Name
ENDPTCTRL6

Offset
0x1D8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Default	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0

Table 114: (ENDPTCTRL6)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	TXE	R/W	0x0	TX Endpoint Enable
22	TXR	R/W	0x0	TX Data Toggle Reset
21	TXI	R/W	0x0	TX Data Toggle Inhibit

Table 114: (ENDPTCTRL6)

Bits	Name	Type	Reset	Description
20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	TXT	R/W	0x0	TX Endpoint Type
17	TXD	R/W	0x0	TX Endpoint Data Source
16	TXS	R/W	0x0	TX Endpoint Stall
15:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	RXE	R/W	0x0	RX Endpoint Enable
6	RXR	R/W	0x0	RX Data Toggle Reset
5	RXI	R/W	0x0	RX Data Toggle Inhibit
4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:2	RXT	R/W	0x0	RX Endpoint Type
1	RXD	R/W	0x0	RX Endpoint Data Sink
0	RXS	R/W	0x0	RX Endpoint Stall

A.2.48 (ENDPTCTRL7)

Instance Name
ENDPTCTRL7

Offset
0x1DC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Default	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0

Table 115: (ENDPTCTRL7)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	TXE	R/W	0x0	TX Endpoint Enable
22	TXR	R/W	0x0	TX Data Toggle Reset
21	TXI	R/W	0x0	TX Data Toggle Inhibit
20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	TXT	R/W	0x0	TX Endpoint Type
17	TXD	R/W	0x0	TX Endpoint Data Source
16	TXS	R/W	0x0	TX Endpoint Stall

Table 115: (ENDPTCTRL7)

Bits	Name	Type	Reset	Description
15:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	RXE	R/W	0x0	RX Endpoint Enable
6	RXR	R/W	0x0	RX Data Toggle Reset
5	RXI	R/W	0x0	RX Data Toggle Inhibit
4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:2	RXT	R/W	0x0	RX Endpoint Type
1	RXD	R/W	0x0	RX Endpoint Data Sink
0	RXS	R/W	0x0	RX Endpoint Stall

A.2.49 (ENDPTCTRL8)

Instance Name
ENDPTCTRL8

Offset
0x1E0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Default	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0	0

Table 116: (ENDPTCTRL8)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	TXE	R/W	0x0	TX Endpoint Enable
22	TXR	R/W	0x0	TX Data Toggle Reset
21	TXI	R/W	0x0	TX Data Toggle Inhibit
20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	TXT	R/W	0x0	TX Endpoint Type
17	TXD	R/W	0x0	TX Endpoint Data Source
16	TXS	R/W	0x0	TX Endpoint Stall
15:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	RXE	R/W	0x0	RX Endpoint Enable
6	RXR	R/W	0x0	RX Data Toggle Reset
5	RXI	R/W	0x0	RX Data Toggle Inhibit

Table 116: (ENDPTCTRL8)

Bits	Name	Type	Reset	Description
4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:2	RXT	R/W	0x0	RX Endpoint Type
1	RXD	R/W	0x0	RX Endpoint Data Sink
0	RXS	R/W	0x0	RX Endpoint Stall

A.3 SDIO Register Information

Table 117: SDIO Register Summary

Offset	Name	Description	Details
0x00	SYSADDR	SDIO System Address Register	Page: 81
0x04	BLK_CNTL	SDIO Block Control Register	Page: 82
0x08	ARG	SDIO Command Argument Register	Page: 84
0x0C	CMD_XFRMD	SDIO Command and Transfer Mode Register	Page: 85
0x10	RESP0	SDIO Command Response Register0	Page: 87
0x14	RESP1	SDIO Command Response Register1	Page: 87
0x18	RESP2	SDIO Command Response Register2	Page: 87
0x1C	RESP3	SDIO Command Response Register3	Page: 88
0x20	DP	SDIO Buffer Data Port Register	Page: 88
0x24	STATE	SDIO Present State Register	Page: 88
0x28	CNTL1	SDIO Host Control Register 1	Page: 92
0x2C	CNTL2	SDIO Host Control Register 2	Page: 94
0x30	I_STAT	SDIO Interrupt Status Register	Page: 96
0x34	I_STAT_EN	SDIO Interrupt Status Enable Register	Page: 100
0x38	I_SIG_EN	SDIO Interrupt Signal Enable Register	Page: 102
0x40	CAP0	SDIO Capabilities Register 0	Page: 104
0xFC	VER	SDIO Controller Version Status	Page: 106

A.3.1 SDIO System Address Register (SYSADDR)

Instance Name
SYSADDR

Offset
0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SYSADDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 118: SDIO System Address Register (SYSADDR)

Bits	Name	Type	Reset	Description
31:0	SYSADDR	R/W	0x0	DMA System Address. This register contains the system memory address for a DMA transfer. When the controller stops a DMA transfer, this register shall point to the system address of the next contiguous data position. It can be accessed only if no transaction is executing (i.e. after a transaction has stopped). Read operations during a data transfer return an invalid value. The host driver software shall initialize this register before starting a DMA transaction. After DMA has stopped, the next system address of the next contiguous data position can be read from this register. The DMA transfer waits at every address boundary specified by the DMA_BUFSZ in the BLK_CNTL register. The controller generates a DMA Interrupt to request to update this register. The software sets the next system address of the next data position to this register. Note: When bits [31:24] of this register are written, the controller will restart the DMA transfer. Writing bits [23:0] will not cause the controller to restart a DMA transfer. When restarting DMA by the resume command or by setting CONTREQ in the CNTL1 register, the controller shall start at the next contiguous address stored here in the System Address register

A.3.2 SDIO Block Control Register (BLK_CNTL)

Instance Name

BLK_CNTL

Offset

0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	BLK_CNT																Reserved	DMA_BUFSZ		XFR_BLKSZ												
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 119: SDIO Block Control Register (BLK_CNTL)

Bits	Name	Type	Reset	Description
31:16	BLK_CNT	R/W	0x0	Block Count for Current Transfer. This register is enabled when the BLKCNTEN bit in the CMD_XFRMD register is set to 1 and is valid only for multiple block transfers. The controller decrements the block count after each block transfer and stops when the count reaches zero. It can be accessed only if no transaction is executing (i.e. after a transaction has stopped). Read operations during transfer return an invalid value and write operations shall be ignored. When saving transfer context as a result of Suspend command, the number of blocks yet to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, the software shall restore the previously saved block count. 0x0: Stop Count 0x1: 1 block 0x2: 2 blocks ... 0xFFFF: 65535 blocks
15	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
14:12	DMA_BUFSZ	R/W	0x0	Host DMA Buffer Size. To perform a long DMA transfer, the SYSADDR register shall be updated at every system memory buffer boundary during the DMA transfer. These bits specify the size of contiguous buffer in the system memory. The DMA transfer shall wait at the system memory buffer boundary specified by these fields and the controller generates the DMA Interrupt to request the software to update the SYSADDR register. These bits are functional when the DMAPRT bit in the CAP0 register is set to 1 and this function is active when the DMA_EN bit in the CMD_XFRMD register is set to 1. 0x0: 4KB 0x1: 8KB 0x2: 16KB 0x3: 32KB 0x4: 64KB 0x5: 128KB 0x6: 256KB 0x7: 512KB

Table 119: SDIO Block Control Register (BLK_CNTL)

Bits	Name	Type	Reset	Description
11:0	XFR_BLKSZ	R/W	0x0	Transfer Block Size. This register specifies the block size for block data transfers for CMD53. It can be accessed only if no transaction is executing (i.e. after a transaction has stopped). Read operations during transfer return an invalid value and write operations shall be ignored. 0x0: No Data Transfer 0x1: 1 Byte 0x2: 2 Bytes 0x3: 3 Bytes 0x4: 4 Bytes ... 0x1FF: 511 Bytes 0x200: 512 Bytes ... 0x800: 2048 Bytes 0x801: FFFh : reserved

A.3.3 SDIO Command Argument Register (ARG)

Instance Name

ARG

Offset

0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ARG																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 120: SDIO Command Argument Register (ARG)

Bits	Name	Type	Reset	Description
31:0	ARG	R/W	0x0	Command Argument. The Command Argument is specified as bit 39-8 of Command-Format. (SDIO Card Specification Version 1.0)

A.3.4 SDIO Command and Transfer Mode Register (CMD_XFRMD)

Instance Name
CMD_XFRMD

Offset
0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved		CMD_IDX						CMD_TYPE		DPSEL	IDXCHKEN	CRCCHKEN	Reserved	RES_TYPE		Reserved										MS_BLKSEL	DXFRDIR	Reserved		BLKCNTEEN	DMA_EN
Default	?	?	0	0	0	0	0	0	0	0	0	0	0	?	0	0	?	?	?	?	?	?	?	?	?	?	0	0	?	?	0	0

Table 121: SDIO Command and Transfer Mode Register (CMD_XFRMD)

Bits	Name	Type	Reset	Description
31:30	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
29:24	CMD_IDX	R/W	0x0	Command Index. These bits shall be set to the command number (CMD0-63, ACMD0-63). (SDIO Card Specification Version 1.0)
23:22	CMD_TYPE	R/W	0x0	Command Type. These bits shall be set to 00b for all commands except suspend, resume and abort commands. Suspend Command: If the Suspend command succeeds, the controller shall assume the SDIO bus has been released and that it is possible to issue the next command which uses the data line. The controller shall de-assert Read Wait for read transactions and stop checking busy for write transactions. The Interrupt cycle shall start, if the device is in 4-bit mode. If the Suspend command fails, the controller shall maintain its current state and the software shall restart the transfer by setting CONTREQ in the CNTL1 register. Resume Command: The software re-starts the data transfer by restoring the registers in the range of 000-00Dh. The controller shall check for busy before starting write transfers. Abort Command: If this command is set when executing a read transfer, the controller shall stop reads to the buffer. If this command is set when executing a write transfer, the controller shall stop driving the data line. After issuing the abort command, the software should issue a software reset. 0x0: Normal 0x1: Suspend 0x2: Resume 0x3: Abort
21	DPSEL	R/W	0x0	Data Present Select. This bit is set to 1 to indicate that data is present and shall be transferred using the SDIO_x line. It is set to 0 for the following: 1. Commands using only SDIO_CMD line (ex. CMD52) 2. Commands with no data transfer but using busy signal on SDIO_0 line 3. Resume Command 0x0: No Data Present 0x1: Data Present

Table 121: SDIO Command and Transfer Mode Register (CMD_XFRMD)

Bits	Name	Type	Reset	Description
20	IDXCHKEN	R/W	0x0	Command Index Check Enable. If this bit is set to 1, the controller shall check the index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked. 0x0: Disable 0x1: Enable
19	CRCCHKEN	R/W	0x0	Command CRC Check Enable. If this bit is set to 1, the controller shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. 0x0: Disable 0x1: Enable
18	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
17:16	RES_TYPE	R/W	0x0	Response Type Select 0x0: No Response 0x1: Response length 136 bits 0x2: Response length 48 bits 0x3: Response length 48 bits
15:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5	MS_BLKSEL	R/W	0x0	Multi / Single Block Select. This bit enables multiple block SDIO_x line data transfers. 0x0: Single Block 0x1: Multiple Block
4	DXFRDIR	R/W	0x0	Data Transfer Direction Select. This bit defines the direction of data line data transfers. 0x0: Write (Host to Card) 0x1: Read (Card to Host)
3:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	BLKCNTEN	R/W	0x0	Block Count Enable. This bit is used to enable the BLK_CNT in the BLK_CNTL register, which is only relevant for multiple block transfers. When this bit is 0, the block count register is disabled, which is useful when executing an infinite transfer. 0x0: Disable 0x1: Enable
0	DMA_EN	R/W	0x0	DMA Enable. If this bit is set to 1, a DMA operation shall begin when the software writes to the CMD_IDX of the CMD_XFRMD register. 0x0: Disable 0x1: Enable

A.3.5 SDIO Command Response Register0 (RESP0)

Instance Name Offset
RESP0 0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RESP[31:0]																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 122: SDIO Command Response Register0 (RESP0)

Bits	Name	Type	Reset	Description
31:0	RESP[31:0]	R	0x0	Command Response. This register contains bits 31:0 of the command response.

A.3.6 SDIO Command Response Register1 (RESP1)

Instance Name Offset
RESP1 0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RESP[63:32]																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 123: SDIO Command Response Register1 (RESP1)

Bits	Name	Type	Reset	Description
31:0	RESP[63:32]	R	0x0	Command Response. This register contains bits 63:32 of the command response.

A.3.7 SDIO Command Response Register2 (RESP2)

Instance Name Offset
RESP2 0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RESP[95:64]																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 124: SDIO Command Response Register2 (RESP2)

Bits	Name	Type	Reset	Description
31:0	RESP[95:64]	R	0x0	Command Response. This register contains bits 95:64 of the command response.

A.3.8 SDIO Command Response Register3 (RESP3)

Instance Name Offset
RESP3 0x1C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RESP[127:96]																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 125: SDIO Command Response Register3 (RESP3)

Bits	Name	Type	Reset	Description
31:0	RESP[127:96]	R	0x0	Command Response. This register contains bits 127:96 of the command response.

A.3.9 SDIO Buffer Data Port Register (DP)

Instance Name Offset
DP 0x20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	BFR_DATA																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 126: SDIO Buffer Data Port Register (DP)

Bits	Name	Type	Reset	Description
31:0	BFR_DATA	R/W	0x0	Buffer Data. The Controller Buffer can be accessed through this 32-bit Data Port Register.

A.3.10 SDIO Present State Register (STATE)

Instance Name Offset
STATE 0x24

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved			UPRDLVL				CMDLVL	LWRDLVL				WPSWLVL	CDDTLVL	CDSTBL	CDINSTD	Reserved				BUFRDEN	BUFWREN	RDACTV	WRAC TV	Reserved				DATACTV	DCMDINHBT	CCMDINHBT	
Default	?	?	?	1	1	1	1	1	1	1	1	1	1	0	0	0	?	?	?	?	0	0	0	0	?	?	?	?	?	0	0	0

Table 127: SDIO Present State Register (STATE)

Bits	Name	Type	Reset	Description
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 127: SDIO Present State Register (STATE)

Bits	Name	Type	Reset	Description
28:25	UPRDLVL	R	0xF	DAT[7:4] Line Signal Level. This status is used to check the upper nibble SDIO_x line level to recover from errors, and for debugging. D25 : DAT[4] D26 : DAT[5] D27 : DAT[6] D28 : DAT[7]
24	CMDLVL	R	0x1	CMD Line Signal Level. This status is used to check SDIO_CMD line level to recover from errors, and for debugging. This bit reflects the state of the SDIO_CMD pin.
23:20	LWRDLVL	R	0xF	DAT[3:0] Line Signal Level. This status is used to check SDIO_x line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DAT[0]. D23 : DAT[3] D22 : DAT[2] D21 : DAT[1] D20 : DAT[0]
19	WPSWLVL	R	0x1	Write Protect Switch Pin Level. The Write Protect Switch is supported for memory and SDIO card. This bit reflects the inverse of the SDWP# pin. 0x0: Write protected (SDWP# = 1) 0x1: Write enabled (SDWP# = 0)
18	CDETLVL	R	0x0	Card Detect Pin Level. This bit reflects the inverse value of the SDCD# pin. 0x0: No Card present (SDCD# = 1) 0x1: Card present (SDCD# = 0)
17	CDSTBL	R	0x0	Card State Stable. This bit is used for testing. If it is 0, the CDETLVL is not stable. If this bit is set to 1, it means the CDETLVL is stable. The MSWRST bit in the CNTL2 register shall not affect this bit. 0x0: Reset or CDETLVL is not stable 0x1: CDETLVL is stable
16	CDINST	R	0x0	Card Inserted. This bit indicates whether a card has been inserted. Changing from 0 to 1 generates a CDINS interrupt in the I_STAT register and changing from 1 to 0 generates a CDREM Interrupt in the I_STAT register. The MSWRST bit in the CNTL2 register shall not affect this bit. If a Card is removed while its power is on and its clock is oscillating, the controller shall clear the BUSPWR bit in the CNTL1 register and CLKEN in the CNTL2 register. In addition the software should reset the controller by the MSWRST bit in CNTL2 register. The card detect is active regardless of the Bus Power. 0x0: Reset or Debouncing or No Card 0x1: Card Inserted

Table 127: SDIO Present State Register (STATE)

Bits	Name	Type	Reset	Description
15:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	BUFRDEN	R	0x0	Buffer Read Enable. This status is used for non-DMA read transfers. This read only flag indicates that valid data exists in the FIFO buffer for reading. This data can then be read by reading the DP register. If this bit is 1, readable data exists in the buffer. A change of this bit from 1 to 0 occurs when all the block data is read from the buffer. A change of this bit from 0 to 1 occurs when all the block data is ready in the buffer and generates the BUFRDRDY Interrupt. 0x0: Read Disable 0x1: Read Enable.
10	BUFWREN	R	0x0	Buffer Write Enable. This status is used for non-DMA write transfers. This read only flag indicates if space is available for write data. If this bit is 1, data can be written to the buffer by writing to the DP register. A change of this bit from 1 to 0 occurs when all the block data is written to the buffer. A change of this bit from 0 to 1 occurs when block data can be written to the FIFO buffer and generates the BUFWRRDY Interrupt. 0x0: Write Disable 0x1: Write Enable.
9	RDACTV	R	0x0	Read Transfer Active. This status is used for detecting completion of a read transfer. This bit is set to 1 for either of the following conditions: 1) After the end bit of the read command 2) When writing a 1 to CONTREQ in the CNTL1 register to restart a read transfer. This bit is cleared to 0 for either of the following conditions: 1) When the last data block as specified by block length is transferred to the system. 2) When all valid data blocks have been transferred to the system and no current block transfers are being sent as a result of the BGREQSTP bit in the CNTL1 register being set to 1. An XFRCOMP interrupt is generated when this bit changes from 1 to 0. 0x0: Not transferring data 0x1: Transferring data

Table 127: SDIO Present State Register (STATE)

Bits	Name	Type	Reset	Description
8	WRACTV	R	0x0	Write Transfer Active. This status indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the controller. This bit is set in either of the following cases: 1) After the end bit of the write command. 2) When writing a 1 to CONTREQ in the CNTL1 register to restart a write transfer. This bit is cleared in either of the following cases: 1) After getting the CRC status of the last data block as specified by the transfer count (Single or Multiple) 2) After getting a CRC status of any block where data transmission is about to be stopped by the assertion of BGREQSTP in the CNTL1 register. During a write transaction, a BGEVNT interrupt is generated when this bit is changed to 0, as a result of the BGREQSTP being set. This status is useful for the software in determining when to issue commands during write busy. 0x0: Not transferring data 0x1: Transferring data
7:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2	DATACTV	R	0x0	DAT Line Active. This bit indicates whether one of the SDIO_x lines on bus is in use. 0x0: SDIO_x line(s) inactive 0x1: SDIO_x line(s) active
1	DCMDINHBT	R	0x0	Command Inhibit (DAT). This status bit is generated if either the DATACTV or the RDATACTV is set to 1. If this bit is 0, it indicates the controller can issue the next command. Commands with busy signal affect DCMDINHBT (ex. R1b, R5b type). Changing from 1 to 0 generates a XFRCOMP interrupt in the I_STAT register. Note: The Host Driver can save registers in the range of 000-00Dh for a suspend transaction after this bit has changed from 1 to 0. 0x0: Can issue command which uses the SDIO_x line(s) 0x1: Cannot issue command which uses the SDIO_x line(s)
0	CCMDINHBT	R	0x0	Command Inhibit (CMD). If this bit is 0, it indicates the SDIO_CMD line is not in use and the controller can issue a command using the SDIO_CMD line. This bit is set immediately after the CMD_XFRMD register is written. This bit is cleared when the command response is received. Even if the DCMDINHBT bit is set to 1, Commands using only the SDIO_CMD line can be issued if this bit is 0. Changing from 1 to 0 generates a CMDCOMP interrupt in the I_STAT register. If the controller cannot issue the command because of a command conflict error, this bit shall remain 1 and the CMDCOMP bit is not set.

A.3.11 SDIO Host Control Register 1 (CNTL1)

Instance Name
CNTL1

Offset
0x28

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												BGIRQEN	RDWTCNTL	CONTREQ	BGREQSTP	Reserved				VLTGSEL	Reserved			BUSPWR	Reserved				HISPEED	4BITMD	LEDCNTL
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	?	?	?	?	0	0	0	0	?	?	?	?	?	0	0	0

Table 128: SDIO Host Control Register 1 (CNTL1)

Bits	Name	Type	Reset	Description
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19	BGIRQEN	R/W	0x0	Interrupt at Block Gap. This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. If the card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. When the software detects a card insertion, it shall set this bit according to the CCCR of the SDIO card.
18	RDWTCNTL	R/W	0x0	Read Wait Control. The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using SDIO_2 line. Otherwise the controller has to stop the SDIO_CLK to hold read data, which restricts command generation. When the software detects a card insertion, it shall set this bit according to the CCCR of the SDIO card. If the card does not support read wait, this bit shall never be set to 1 otherwise SDIO_x line conflict may occur. If this bit is set to 0, Suspend / Resume cannot be supported 0x0: Disable Read Wait Control 0x1: Enable Read Wait Control
17	CONTREQ	R/W	0x0	Continue Request. This bit is used to restart a transaction which was stopped using the BGREQSTP bit in this register. To cancel stop at the block gap, set BGREQSTP to 0 and set this bit to restart the transfer. The controller automatically clears this bit in either of the following cases: 1) In the case of a read transaction, the DATACTV changes from 0 to 1 as a read transaction restarts. 2) In the case of a write transaction, the WRACTV bit changes from 0 to 1 as the write transaction restarts. Therefore it is not necessary for the host driver to set this bit to 0. If BGREQSTP is set to 1, any write to this bit is ignored. 0x0: Ignored 0x1: Restart

Table 128: SDIO Host Control Register 1 (CNTL1)

Bits	Name	Type	Reset	Description
16	BGREQSTP	R/W	0x0	Stop at Block Gap Request. This bit is used to stop executing a transaction at the next block gap for DMA or non-DMA transfers. Until the XFRCOMP bit in the I_STAT register is set to 1, indicating a transfer completion the software shall leave this bit set to 1. Clearing both the BGREQSTP and CONTREQ shall not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The controller shall honor BGREQSTP for write transfers, but for read transfers it requires that the card supports Read Wait. Therefore the software shall not set this bit during read transfers unless the card supports Read Wait and has set RDWTCNTL to 1. In case of write transfers in which the software writes data to the DP register, the software shall set this bit after all block data is written. If this bit is set to 1, the software shall not write data to DP register. This bit affects RDACTV, WRACTV, DATACTV and DCMDINHBT in the STATE register. 0x0: Transfer 0x1: Stop
15:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11:9	VLTGSEL	R/W	0x0	Bus Voltage Select. By setting these bits, the software selects the voltage level for the card. Before setting this register, the software shall check the voltage support bits in the CAP0 register. The system software should only program this field to a value that is supported. 0x7: 3.3 V(Typ.) 0x6: 3.0 V(Typ.) 0x5: 1.8 V(Typ.) 0x4: 000b : Reserved
8	BUSPWR	R/W	0x0	Bus Power. Before setting this bit, the host driver shall set Bus Voltage Select. If the controller detects the No Card State, this bit shall be cleared. 0x0: Power off 0x1: Power on
7:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 128: SDIO Host Control Register 1 (CNTL1)

Bits	Name	Type	Reset	Description
2	HISPEED	R/W	0x0	High Speed Enable. This bit is optional. Before setting this bit, the software shall check the High Speed Support in the CAP0 register. If this bit is set to 0 (default), the controller outputs SDIO_CMD line and SDIO_x lines at the falling edge of the SDIO_CLK clock (up to 25 MHz). If this bit is set to 1, the controller outputs SDIO_CMD line and SDIO_x lines at the rising edge of the SDIO_CLK clock (up to 50MHz). The device's EXT_CSD register needs to be checked for high speed mode support, and the device needs to be set to that mode (using a CMD52 command) before setting this bit to a 1. 0x0: Normal Speed Mode 0x1: High Speed Mode
1	4BITMD	R/W	0x0	4 Bit Mode. This bit selects the data width of the controller. The software shall select it to match the data width of the card. 0x0: 1 bit mode 0x1: 4 bit mode
0	LEDCNTL	R/W	0x0	LED Control. This bit is used to caution the user not to remove the card while the card is being accessed. If the software is going to issue multiple commands, this bit can be set during all transactions. It is not necessary to change for each transaction. 0x0: LED off 0x1: LED on

A.3.12 SDIO Host Control Register 2 (CNTL2)

Instance Name
 CNTL2

Offset
 0x2C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved					DATSWRST	CMDSWRST	MSWRST	Reserved				DTCNTR				Reserved													CLKEN	Reserved	INTCLKEN
Default	?	?	?	?	?	0	0	0	?	?	?	?	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	0

Table 129: SDIO Host Control Register 2 (CNTL2)

Bits	Name	Type	Reset	Description
31:27	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 129: SDIO Host Control Register 2 (CNTL2)

Bits	Name	Type	Reset	Description
26	DATSWRST	R/W	0x0	Software Reset for SDIO_x lines. Only part of the data circuit is reset. The following registers and bits are cleared by this bit: DP register : Buffer is cleared and Initialized. STATE register: bits of BUFRDEN, BUFWREN, RDACTV, WRACTV, DATACTV, DCMDINHBT CNTL1 register: bits of CONTREQ, BGREQSTP I_STAT register: bits of BUFRDRDY, BUFWRRDY, BGEVNT, XFRCOMP 0x0: Functional 0x1: Reset
25	CMDSWRST	R/W	0x0	Software Reset for SDIO_CMD line. Only part of the command circuit is reset. The following registers and bits are cleared by this bit: STATE register, CCMDINHBT, I_STAT register, CMDCOMP 0x0: Functional 0x1: Reset
24	MSWRST	R/W	0x0	Software Reset for All. This reset affects the entire controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared to 0. During its initialization, the software shall set this bit to 1 to reset the controller. The controller shall reset this bit to 0 when the CAP0 register is valid and the software can read it. Additional use of Software Reset For All may not affect the value of the CAP0 register. If this bit is set to 1, the BUSPWR bit in the CNTL1 register is cleared, causing the bus power to be removed. When the bus power is restored, the card shall reset itself and must be reinitialized by the software. 0x0: Functional 0x1: Reset
23:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:16	DTOCNTR	R/W	0x0	Data Timeout Counter Value. This value determines the interval by which DAT line time-outs are detected. Refer to the DTOERR in the I_STAT register for information on factors that dictate time-out generation. Time-out clock frequency will be generated by dividing the base timeout clock by this value. When setting this register, prevent inadvertent time-out events by clearing the DTOSTEN (in the I_STAT_EN register) 0xF: Reserved 0xE: (timeout clock)*2²⁷ 0x1: (timeout clock)*2¹⁴ 0x0: (timeout clock)*2¹³
15:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 129: SDIO Host Control Register 2 (CNTL2)

Bits	Name	Type	Reset	Description
2	CLKEN	R/W	0x0	SDIO Clock Enable. The controller shall stop SDIO_CLK when writing this bit to 0. If the controller detects the No Card state, this bit shall be cleared. 0x0: Disable 0x1: Enable
1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	INTCLKEN	R/W	0x0	Internal clock enable. This bit needs to be set in order to enable the controller's internal clock.

A.3.13 SDIO Interrupt Status Register (I_STAT)

Instance Name
I_STAT

Offset
0x30

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved			AHBTERR	Reserved				ILMTERR	DENDERR	DCRCERR	DTOERR	CIDXERR	CENDERR	CCRCERR	CTOERR	ERRINT	Reserved						CDINT	CDREM	CDINS	BUFRDRDY	BUFWRDRDY	DMAINT	BGEVNT	XFERCOMP	CMDCOMP
Default	?	?	?	0	?	?	?	?	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0

Table 130: SDIO Interrupt Status Register (I_STAT)

Bits	Name	Type	Reset	Description
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
28	AHBERR	R/W	0x0	AHB Target Error. When the controller's AHB master interface receives an error response from the AHB target, this bit is set to a 1, and the AHB transaction is aborted. 0x0: No Error 0x1: Error
27:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 130: SDIO Interrupt Status Register (I_STAT)

Bits	Name	Type	Reset	Description
23	ILMTERR	R/W	0x0	Current Limit Error. By setting the BUSPWR bit in the CNTL1 register, the controller is requested to supply power for the SDIO Bus. If the controller supports the Current Limit Function, it can be protected from a card that consumes excessive current by stopping power supply to the card in which case this bit indicates a failure status. Reading 1 means the controller is not supplying power to SDIO card due to some failure. Reading 0 means that the controller is supplying power and no error has occurred. This bit shall always set to be 0, if the controller does not support this function. 0x0: No Error 0x1: Power Fail
22	DENDERR	R/W	0x0	Data End Bit Error. Occurs when detecting 0 at the end bit position of read data which uses the DAT line or the end bit position of the CRC status. 0x0: No Error 0x1: Error
21	DCRCERR	R/W	0x0	Data CRC Error. Occurs when detecting CRC error when transferring read data which uses the DAT line or when detecting the Write CRC Status having a value of other than "010". 0x0: No Error 0x1: Error
20	DTOERR	R/W	0x0	Data Timeout Error. Occurs when detecting one of following timeout conditions. 1) Busy Timeout for R1b, R5b type. 2) Busy Timeout after Write CRC status 3) Write CRC status Timeout 4) Read Data Timeout 0x0: No Error 0x1: Timeout
19	CIDXERR	R/W	0x0	Command Index Error. Occurs if a Command Index error occurs in the Command Response. 0x0: No Error 0x1: Error
18	CENDERR	R/W	0x0	Command End Bit Error. Occurs when detecting that the end bit of a command response is 0. 0x0: No Error 0x1: End Bit Error Generated

Table 130: SDIO Interrupt Status Register (I_STAT)

Bits	Name	Type	Reset	Description
17	CCRCERR	R/W	0x0	Command CRC Error. Command CRC Error is generated in two cases. 1) If a response is returned and the CTOERR is set to 0, this bit is set to 1 when detecting a CRT error in the command response. 2) The controller detects a SDIO_CLK line conflict by monitoring the SDIO_CMD line when a command is issued. If the controller drives the SDIO_CMD line to 1 level, but detects 0 level on the SDIO_CMD line at the next SDIO_CLK edge, then the controller shall abort the command (Stop driving SDIO_CMD line) and set this bit to 1. The Command Timeout Error shall also be set to 1 to distinguish SDIO_CMD line conflict. 0x0: No Error 0x1: CRC Error Generated
16	CTOERR	R/W	0x0	Command Timeout Error. Occurs only if the no response is returned within 64 SDIO_CLK cycles from the end bit of the command. If the controller detects a SDIO_CMD line conflict, in which case CCRCERR shall also be set. This bit shall be set without waiting for 64 SDIO_CLK cycles because the command will be aborted by the controller. 0x0: No Error 0x1: Timeout
15	ERRINT	R	0x0	Error Interrupt. If any of the error bits (bits 24:16) of this register are set, then this bit is set. Therefore the software can test for an error by checking this bit first. 0x0: No Error 0x1: Error
14:9	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
8	CDINT	R	0x0	Card Interrupt. Writing this bit to 1 does not clear this bit. It is cleared by resetting the SDIO card interrupt factor. In 1-bit mode, the controller shall detect the CDINT without SDIO_CLK. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the card and the interrupt to the Host system. When this status has been set and the software needs to start this interrupt service, CDINTSTEN in the I_STAT register shall be set to 0 in order to clear the card interrupt statuses latched in the controller and stop driving the Host System. After completion of the card interrupt service (the reset factor in the SDIO card and the interrupt signal may not be asserted), set Card Interrupt Status Enable to 1 and start sampling the interrupt signal again. 0x0: No Card Interrupt 0x1: Generate Card Interrupt

Table 130: SDIO Interrupt Status Register (I_STAT)

Bits	Name	Type	Reset	Description
7	CDREM	R/W	0x0	Card Removal. This status is set if the CDINSTD in the STATE register changes from 1 to 0. When the software writes this bit to 1 to clear this status the status of the Card Inserted in the STATE register should be confirmed. Because the card detect may possibly be changed when the software clear this bit an Interrupt event may not be generated. 0x0: Card State Stable or Debouncing 0x1: Card Removed
6	CDINS	R/W	0x0	Card Insertion. This status is set if the CDINSTD in the STATE register changes from 0 to 1. When the software writes this bit to 1 to clear this status the status of the Card Inserted in the STATE register should be confirmed. Because the card detect may possibly be changed when the software clear this bit an Interrupt event may not be generated. 0x0: Card State Stable or Debouncing 0x1: Card Inserted
5	BUFRDRDY	R/W	0x0	Buffer Read Ready. This status is set if the BUFRDEN bit changes from 0 to 1. 0x0: Not Ready to read Buffer 0x1: Ready to read Buffer
4	BUFWRRDY	R/W	0x0	Buffer Write Ready. This status is set if the BUFWREN bit changes from 0 to 1. 0x0: Not Ready to Write Buffer 0x1: Ready to Write Buffer
3	DMAINT	R/W	0x0	DMA Interrupt. This status is set if the controller detects the system memory buffer boundary in the Block Size register. 0x0: No DMA Interrupt 0x1: DMA Interrupt is Generated
2	BGEVNT	R/W	0x0	Block Gap Event. If the BGREQSTP in the CNTL1 register is set, this bit is set. Read Transaction: This bit is set at the falling edge of the DACTV Status (When the transaction is stopped at SDIO Bus timing. The Read Wait must be supported in order to use this function). Write Transaction: This bit is set at the falling edge of WRACTV Status (After getting CRC status at SDIO Bus timing). 0x0: No Block Gap Event 0x1: Transaction stopped at Block Gap

Table 130: SDIO Interrupt Status Register (I_STAT)

Bits	Name	Type	Reset	Description
1	XFRCOMP	R/W	0x0	Transfer Complete. This bit is set when a read / write transaction is completed. Read Transaction: This bit is set at the falling edge of RDACTV Status. There are two cases in which the Interrupt is generated. The first is when a data transfer is completed as specified by data length (After the last data has been read to the Host System). The second is when data has stopped at the block gap and completed the data transfer by setting the BGREQSTP in the CNTL1 register (After valid data has been read to the Host System). Write Transaction: This bit is set at the falling edge of the DATACTV Status. There are two cases in which the Interrupt is generated. The first is when the last data is written to the card as specified by data length and busy signal is released. The second is when data transfers are stopped at the block gap by setting BGREQSTP in the CNTL1 register and data transfers completed. (After valid data is written to the SDIO card and the busy signal is released). Note: XFRCOMP has higher priority than DTOERR. If both bits are set to 1, the data transfer can be considered complete. 0x0: No Data Transfer Complete 0x1: Data Transfer Complete
0	CMDCOMP	R/W	0x0	Command Complete. This bit is set when the controller gets the end bit of the command response. Note: CTOERR has higher priority than CMDCOMP. If both are set to 1, it can be considered that the response was not received correctly. 0x0: No Command Complete 0x1: Command Complete

A.3.14 SDIO Interrupt Status Enable Register (I_STAT_EN)

Instance Name

I_STAT_EN

Offset

0x34

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved			ATERRSTEN	Reserved				ILMTSTEN	DENDSTEN	DCRCSTEN	DTOSTEN	CIDXSTEN	CENDSTEN	CCRCSTEN	CTOSTEN	Reserved							CDINTSTEN	CDREMSTEN	CDINSSTEN	BUFRDRDYSTEN	BUFWRDRDYSTEN	DMAINTSTEN	BGEVNTSTEN	XFRCOMPSTEN	CMDCOMPSTEN
Default	?	?	?	0	?	?	?	?	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0

Table 131: SDIO Interrupt Status Enable Register (I_STAT_EN)

Bits	Name	Type	Reset	Description
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
28	ATERRSTEN	R/W	0x0	AHB Target Error Status Enable 0x0: Masked 0x1: Enabled
27:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	ILMTSTEN	R/W	0x0	Current Limit Error Status Enable 0x0: Masked 0x1: Enabled
22	DENDSTEN	R/W	0x0	Data End Bit Error Status Enable 0x0: Masked 0x1: Enabled
21	DCRCSTEN	R/W	0x0	Data CRC Error Status Enable 0x0: Masked 0x1: Enabled
20	DTOSTEN	R/W	0x0	Data Timeout Error Status Enable 0x0: Masked 0x1: Enabled
19	CIDXSTEN	R/W	0x0	Command Index Error Status Enable 0x0: Masked 0x1: Enabled
18	CENDSTEN	R/W	0x0	Command End Bit Error Status Enable 0x0: Masked 0x1: Enabled
17	CCRCSTEN	R/W	0x0	Command CRC Error Status Enable 0x0: Masked 0x1: Enabled
16	CTOSTEN	R/W	0x0	Command Timeout Error Status Enable 0x0: Masked 0x1: Enabled
15:9	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
8	CDINTSTEN	R/W	0x0	Card Interrupt Status Enable. If this bit is set to 0, the controller shall clear Interrupt request to the System. The CDINT detection is stopped when this bit is cleared and restarted when this bit is set to 1. The software should clear the CDINTSTEN before servicing the CDINT and should set this bit again after all Interrupt requests from the card are cleared to prevent inadvertent Interrupts. 0x0: Masked 0x1: Enabled

Table 131: SDIO Interrupt Status Enable Register (I_STAT_EN)

Bits	Name	Type	Reset	Description
7	CDREMSTEN	R/W	0x0	Card Removal Status Enable 0x0: Masked 0x1: Enabled
6	CDINSSTEN	R/W	0x0	Card Insertion Status Enable 0x0: Masked 0x1: Enabled
5	BUFRDRDYSTEN	R/W	0x0	Buffer Read Ready Status Enable 0x0: Masked 0x1: Enabled
4	BUFWRDRDYSTEN	R/W	0x0	Buffer Write Ready Status Enable 0x0: Masked 0x1: Enabled
3	DMAINTSTEN	R/W	0x0	DMA Interrupt Status Enable 0x0: Masked 0x1: Enabled
2	BGEVNTSTEN	R/W	0x0	Block Gap Event Status Enable 0x0: Masked 0x1: Enabled
1	XFRCOMPSTEN	R/W	0x0	Transfer Complete Status Enable 0x0: Masked 0x1: Enabled
0	CMDCOMPSTEN	R/W	0x0	Command Complete Status Enable 0x0: Masked 0x1: Enabled

A.3.15 SDIO Interrupt Signal Enable Register (I_SIG_EN)

Instance Name

I_SIG_EN

Offset

0x38

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved			ATERRSGEN	Reserved				ILMTSGEN	DENDSGEN	DCRCSGEN	DTOSGEN	CIDXSGEN	CENDSGEN	CCRCSGEN	CTOSGEN	Reserved								CDINTSGEN	CDREMSGEN	CDINSSGEN	BUFRDRDYSGEN	BUFWRDRDYSGEN	DMAINTSGEN	BGEWNTSGEN	XFCRMPMSGEN	CMDCMPMSGEN
Default									?	?	?	0	?	?	?	?									0	0	0	0	0	0	0	?	?

Table 132: SDIO Interrupt Signal Enable Register (I_SIG_EN)

Bits	Name	Type	Reset	Description
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
28	ATERRSGEN	R/W	0x0	AHB Target Error Signal Enable 0x0: Masked 0x1: Enabled
27:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	ILMTSGEN	R/W	0x0	Current Limit Error Signal Enable 0x0: Masked 0x1: Enabled
22	DENDSGEN	R/W	0x0	Data End Bit Error Signal Enable 0x0: Masked 0x1: Enabled
21	DCRCSGEN	R/W	0x0	Data CRC Error Signal Enable 0x0: Masked 0x1: Enabled
20	DTOSGEN	R/W	0x0	Data Timeout Error Signal Enable 0x0: Masked 0x1: Enabled
19	CIDXSGEN	R/W	0x0	Command Index Error Signal Enable 0x0: Masked 0x1: Enabled
18	CENDSGEN	R/W	0x0	Command End Bit Error Signal Enable 0x0: Masked 0x1: Enabled
17	CCRCSGEN	R/W	0x0	Command CRC Error Signal Enable 0x0: Masked 0x1: Enabled
16	CTOSGEN	R/W	0x0	Command Timeout Error Signal Enable 0x0: Masked 0x1: Enabled
15:9	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
8	CDINTSGEN	R/W	0x0	Card Interrupt Signal Enable 0x0: Masked 0x1: Enabled
7	CDREMSGEN	R/W	0x0	Card Removal Signal Enable 0x0: Masked 0x1: Enabled
6	CDINSSGEN	R/W	0x0	Card insertion signal enable 0x0: Masked 0x1: Enabled

Table 132: SDIO Interrupt Signal Enable Register (I_SIG_EN)

Bits	Name	Type	Reset	Description
5	BUFRDRDYSGEN	R/W	0x0	Buffer Read ready signal enable 0x0: Masked 0x1: Enabled
4	BUFWRDYSGEN	R/W	0x0	Buffer write ready signal enable 0x0: Masked 0x1: Enabled
3	DMAINTSGEN	R/W	0x0	DMA interrupt signal enable 0x0: Masked 0x1: Enabled
2	BGEVNTSGEN	R/W	0x0	Block gap event signal enable 0x0: Masked 0x1: Enabled
1	XFRCOMPSTGEN	R/W	0x0	Transfer complete signal enable 0x0: Masked 0x1: Enabled
0	CMDCOMPSTGEN	R/W	0x0	Command complete signal enable 0x0: Masked 0x1: Enabled

A.3.16 SDIO Capabilities Register 0 (CAP0)

Instance Name
CAP0

Offset
0x40

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved				IRQMODE	1.8VSPRT	3.0VSPRT	3.3VSPRT	SUSP/RES	DMA SPRT	HISPD SPRT	Reserved			MAXBLEN		Reserved		BSCLKFREQ						TOCLKUNIT	Reserved	TOCLKFREQ					
Default	?	?	?	?	1	0	0	1	1	1	1	?	?	?	1	1	?	?	1	1	0	0	0	0	1	?	1	1	0	0	0	0

Table 133: SDIO Capabilities Register 0 (CAP0)

Bits	Name	Type	Reset	Description
31:28	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
27	IRQMODE	R	0x1	Interrupt mode 0x0: Not Supported 0x1: Supported
26	1.8VSPRT	R	0x0	1.8V Voltage Support 0x0: 1.8 V Not Supported 0x1: 1.8 V Supported

Table 133: SDIO Capabilities Register 0 (CAP0)

Bits	Name	Type	Reset	Description
25	3.0VSPRT	R	0x0	3.0V Voltage Support. 0x0: 3.0V Not Supported 0x1: 3.0V Supported
24	3.3VSPRT	R	0x1	3.3V Voltage Support. 0x0: 3.3V Not Supported 0x1: 3.3V Supported
23	SUSP/RES	R	0x1	Suspend / Resume Support. This bit indicates whether the controller supports Suspend / Resume functionality. If this bit is 0, the Suspend and Resume mechanism are not supported and the software shall not issue either Suspend / Resume commands. 0x0: Not Supported 0x1: Supported
22	DMASPT	R	0x1	DMA Support. This bit indicates whether the controller is capable of using DMA to transfer data between system memory and the controller directly. 0x0: DMA Not Supported 0x1: DMA Supported
21	HISPDSPRT	R	0x1	High Speed Support. This bit indicates whether the controller and the Host System support High Speed mode and they can supply SDIO_CLK frequency from 25MHz to 50MHz. 0x0: High Speed Not Supported 0x1: High Speed Supported
20:18	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
17:16	MAXBLEN	R	0x3	Max Block Length. This value indicates the maximum block size that the software can read and write to the buffer in the controller. The buffer shall transfer this block size without wait cycles. Three sizes can be defined as indicated below. 0x0: 512 byte 0x1: 1024 byte 0x2: 2048 byte 0x3: Reserved
15:14	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
13:8	BSCLKFREQ	R	0x30	Base Clock Frequency for SDIO_CLK. This value indicates the base (maximum) clock frequency for the SDIO_CLK. Unit values are 1MHz. If the real frequency is 16.5 MHz, the larger value shall be set 010001b (17MHz) because the software uses this value to calculate the clock divider value and it shall not exceed the upper limit of the SDIO_CLK frequency. If these bits are all 0, the Host System has to get information via another method. 0x0: Reserved

Table 133: SDIO Capabilities Register 0 (CAP0)

Bits	Name	Type	Reset	Description
7	TOCLKUNIT	R	0x1	Timeout Clock Unit. This bit shows the unit of base clock frequency used to detect Data Timeout Error. 0x0: KHz 0x1: MHz
6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5:0	TOCLKFREQ	R	0x30	Timeout Clock Frequency. This bit shows the base clock frequency used to detect Data Timeout Error. 0x0: Reserved

A.3.17 SDIO Controller Version Status (VER)

Instance Name
VER

Offset
0xFC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	VNDRVER								SPECVER								Reserved															
Default	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	1	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Table 134: SDIO Controller Version Status (VER)

Bits	Name	Type	Reset	Description
31:24	VNDRVER	R	0x79	Vendor Version Number. This status is reserved for the vendor version number. The software should not use this status.
23:16	SPECVER	R	0x1	Specification Version Number. This Status indicates the Controller Spec Version. The Upper and Lower 4 bits indicate the version. 0x0: SDIO Host Specification version1.0 others: Reserved
15:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.4 AES Register Information

Table 135: AES Register Summary

Offset	Name	Description	Details
0x00	CTRL1	AES Control Register 1	Page: 108
0x04	CTRL2	AES Control Register 2	Page: 110
0x08	STATUS	AES Status Register	Page: 110
0x0C	ASTR_LEN	AES Astr Length Register	Page: 112
0x10	MSTR_LEN	AES Mstr Length Register	Page: 112
0x14	STR_IN	AES Stream Input Register	Page: 112
0x18	IV0	AES Input Vector Register 0	Page: 113
0x1C	IV1	AES Input Vector Register 1	Page: 113
0x20	IV2	AES Input Vector Register 2	Page: 113
0x24	IV3	AES Input Vector Register 3	Page: 114
0x28	KEY0	AES Key 0	Page: 114
0x2C	KEY1	AES Key 1	Page: 114
0x30	KEY2	AES Key 2	Page: 115
0x34	KEY3	AES Key 3	Page: 115
0x38	KEY4	AES Key 4	Page: 115
0x3C	KEY5	AES Key 5	Page: 116
0x40	KEY6	AES Key 6	Page: 116
0x44	KEY7	AES Key 7	Page: 116
0x48	STR_OUT	AES Stream Output Port	Page: 117
0x4C	OV0	AES Output Vector 0	Page: 117
0x50	OV1	AES Output Vector 1	Page: 117
0x54	OV2	AES Output Vector 2	Page: 118
0x58	OV3	AES Output Vector 3	Page: 118
0x5C	ISR	AES Interrupt Status Register	Page: 118
0x60	IMR	AES Interrupt Mask Register	Page: 119
0x64	IRSR	AES Interrupt Raw Status Register	Page: 120
0x68	ICR	AES Interrupt Clear Register	Page: 120
0x8C	RESERVED	Reserved	Page: 121

Table 135: AES Register Summary

Offset	Name	Description	Details
0x90	RESERVED	Reserved	Page: 121
0x94	RESERVED	Reserved	Page: 122
0x98	RESERVED	Reserved	Page: 122
0x9C	RESERVED	Reserved	Page: 123
0xA0	RESERVED	Reserved	Page: 123
0xA4	RESERVED	Reserved	Page: 123
0xA8	RESERVED	Reserved	Page: 124
0xAC	RESERVED	Reserved	Page: 124
0xB0	RESERVED	Reserved	Page: 124
0xB4	RESERVED	Reserved	Page: 125
0xB8	RESERVED	Reserved	Page: 125
0xBC	RESERVED	Reserved	Page: 126

A.4.1 AES Control Register 1 (CTRL1)

Instance Name
CTRL1

Offset
0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved						CTR_MOD						MODE				DECRYPT	OUT_MIC	MIC_LEN	KEY_SIZE		DMA_LEN	IO_SRC	Reserved	OUT_HDR	OUT_MSG	OF_CLR	IF_CLR	Reserved	START		
Default	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	?	?	0	1	0	0	?	0

Table 136: AES Control Register 1 (CTRL1)

Bits	Name	Type	Reset	Description
31:26	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
25:19	CTR_MOD	R/W	0x0	CTR mode's counter modular modular = 2^{128} : [7'h0-7'hF] modular = $2^{\text{ctr\_mod}}$: others

Table 136: AES Control Register 1 (CTRL1)

Bits	Name	Type	Reset	Description
18:16	MODE	R/W	0x0	AES running mode 0x0: ECB 0x1: CBC 0x2: CTR 0x3: RESERVED 0x4: RESERVED 0x5: CCM* 0x6: MMO 0x7: BYPASS
15	DECRYPT	R/W	0x0	Decrypt operation. Ignored in MMO and BYPASS Mode 0x0: Encryption 0x1: Decryption
14	OUT_MIC	R/W	0x1	Append MIC/HASH at the end of output stream in CCM* mode decryption/MMO mode. 0x0: Not append MIC/HASH at the end of output stream in CCM* mode decryption or MMO mode 0x1: Append MIC/HASH at the end of output stream in CCM* mode decryption or MMO mode
13:12	MIC_LEN	R/W	0x0	Length of MIC field 0x0: 0 bytes 0x1: 4 bytes 0x2: 8 bytes 0x3: 16 bytes
11:10	KEY_SIZE	R/W	0x0	Key size parameter 0x0: 16 bytes 0x1: 32 bytes 0x2: 24 bytes
9	DMA_EN	R/W	0x0	Enable DMA 0x0: Disable DMA 0x1: Enable DMA
8	IO_SRC	R/W	0x0	AES data input source 0x1: I/O through DMA 0x0: I/O through register
7:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5	OUT_HDR	R/W	0x0	Output B0 and I(a) in CCM* mode 0x0: don't output B0 and I(a) at the beginning of output stream 0x1: output B0 and I(a) at the beginning of output stream
4	OUT_MSG	R/W	0x1	Output stream to output FIFO 0x0: block output stream from output FIFO 0x1: forward output stream to output FIFO

Table 136: AES Control Register 1 (CTRL1)

Bits	Name	Type	Reset	Description
3	OF_CLR	R/W	0x0	Clear output FIFO write '1' will generate one-cycle pulse to clear output FIFO
2	IF_CLR	R/W	0x0	Clear input FIFO write '1' will generate one-cycle pulse to clear input FIFO
1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	START	R/W	0x0	Start AES write '1' will generate one-cycle pulse to start AES operation

A.4.2 AES Control Register 2 (CTRL2)

Instance Name
CTRL2

Offset
0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														RESET	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		?

Table 137: AES Control Register 2 (CTRL2)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	RESET	R/W	0x0	Reset AES 0x0: Un-reset AES 0x1: Reset AES

A.4.3 AES Status Register (STATUS)

Instance Name
STATUS

Offset
0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												OF_DEPTH		IF_DEPTH		STATUS				Reserved		OF_EMPTY	OF_RDY	Reserved	IF_FULL	Reserved		DONE			
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	?	?	?	1	0	?	0	?	?	?	1

Table 138: AES Status Register (STATUS)

Bits	Name	Type	Reset	Description
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 138: AES Status Register (STATUS)

Bits	Name	Type	Reset	Description
19:17	OF_DEPTH	R	0x0	The output FIFO depth
16:14	IF_DEPTH	R	0x0	The input FIFO depth
13:11	STATUS	R	0x0	AES operation error status status[2]: MIC Mismatch during CCM* Decryption status[1]: Data is not multiple of 16 bytes in ECB mode or Data is more than $2^{13}-1$ bytes in MMO mode status[0]: Input stream size less than 16 byte in ECB, CBC and CTR mode 0x0 : No operation error 0x1 : Input stream size less than 16 byte in ECB, CBC and CTR mode 0x2 : Data is not multiple of 16 bytes in ECB mode or Data is more than $2^{13}-1$ bytes in MMO mode 0x4 : MIC Mismatch during CCM* Decryption 0x3 : Data is not multiple of 16 bytes and less than 16 byte in ECB mode others : not valid
10:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	OF_EMPTY	R	0x1	Output FIFO empty 0x0 : Output FIFO is not empty 0x1 : Output FIFO is empty
6	OF_RDY	R	0x0	Output FIFO is ready to read 0x0 : Output FIFO is not ready to read 0x1 : Output FIFO is ready to read
5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	IF_FULL	R	0x0	Input FIFO full 0x0 : Input FIFO is not full 0x1 : Input FIFO is full
3:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	DONE	R	0x1	AES operation done 0x0 : AES operation has not done yet 0x1 : AES operation done

A.4.4 AES Astr Length Register (ASTR_LEN)

Instance Name

ASTR_LEN

Offset

0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ASTR_LEN																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 139: AES Astr Length Register (ASTR_LEN)

Bits	Name	Type	Reset	Description
31:0	ASTR_LEN	R/W	0x0	Size of associate string

A.4.5 AES Mstr Length Register (MSTR_LEN)

Instance Name

MSTR_LEN

Offset

0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	MSTR_LEN																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 140: AES Mstr Length Register (MSTR_LEN)

Bits	Name	Type	Reset	Description
31:0	MSTR_LEN	R/W	0x0	Size of message string

A.4.6 AES Stream Input Register (STR_IN)

Instance Name

STR_IN

Offset

0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	STR_IN																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 141: AES Stream Input Register (STR_IN)

Bits	Name	Type	Reset	Description
31:0	STR_IN	W	0x0	Input message word

A.4.7 AES Input Vector Register 0 (IV0)

Instance Name
IV0

Offset
0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	IV0																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 142: AES Input Vector Register 0 (IV0)

Bits	Name	Type	Reset	Description
31:0	IV0	R/W	0x0	Byte 0-3 of initial vector

A.4.8 AES Input Vector Register 1 (IV1)

Instance Name
IV1

Offset
0x1C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	IV1																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 143: AES Input Vector Register 1 (IV1)

Bits	Name	Type	Reset	Description
31:0	IV1	R/W	0x0	Byte 4-7 of initial vector

A.4.9 AES Input Vector Register 2 (IV2)

Instance Name
IV2

Offset
0x20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	IV2																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 144: AES Input Vector Register 2 (IV2)

Bits	Name	Type	Reset	Description
31:0	IV2	R/W	0x0	Byte 8-11 of initial vector

A.4.10 AES Input Vector Register 3 (IV3)

Instance Name

IV3

Offset

0x24

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	IV3																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 145: AES Input Vector Register 3 (IV3)

Bits	Name	Type	Reset	Description
31:0	IV3	R/W	0x0	Byte 12-15 of initial vector

A.4.11 AES Key 0 (KEY0)

Instance Name

KEY0

Offset

0x28

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	KEY0																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 146: AES Key 0 (KEY0)

Bits	Name	Type	Reset	Description
31:0	KEY0	R/W	0x0	Byte 0-3 of key

A.4.12 AES Key 1 (KEY1)

Instance Name

KEY1

Offset

0x2C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	KEY1																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 147: AES Key 1 (KEY1)

Bits	Name	Type	Reset	Description
31:0	KEY1	R/W	0x0	Byte 4-7 of key

A.4.13 AES Key 2 (KEY2)

Instance Name
KEY2

Offset
0x30

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	KEY2																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 148: AES Key 2 (KEY2)

Bits	Name	Type	Reset	Description
31:0	KEY2	R/W	0x0	Byte 8-11 of key

A.4.14 AES Key 3 (KEY3)

Instance Name
KEY3

Offset
0x34

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	KEY3																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 149: AES Key 3 (KEY3)

Bits	Name	Type	Reset	Description
31:0	KEY3	R/W	0x0	Byte 12-15 of key

A.4.15 AES Key 4 (KEY4)

Instance Name
KEY4

Offset
0x38

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	KEY4																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 150: AES Key 4 (KEY4)

Bits	Name	Type	Reset	Description
31:0	KEY4	R/W	0x0	Byte 16-19 of key

A.4.16 AES Key 5 (KEY5)

Instance Name
KEY5

Offset
0x3C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	KEY5																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 151: AES Key 5 (KEY5)

Bits	Name	Type	Reset	Description
31:0	KEY5	R/W	0x0	Byte 20-23 of key

A.4.17 AES Key 6 (KEY6)

Instance Name
KEY6

Offset
0x40

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	KEY6																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 152: AES Key 6 (KEY6)

Bits	Name	Type	Reset	Description
31:0	KEY6	R/W	0x0	Byte 24-27 of key

A.4.18 AES Key 7 (KEY7)

Instance Name
KEY7

Offset
0x44

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	KEY7																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 153: AES Key 7 (KEY7)

Bits	Name	Type	Reset	Description
31:0	KEY7	R/W	0x0	Byte 28-31 of key

A.4.19 AES Stream Output Port (STR_OUT)

Instance Name
STR_OUT

Offset
0x48

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	STR_OUT																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 154: AES Stream Output Port (STR_OUT)

Bits	Name	Type	Reset	Description
31:0	STR_OUT	R	0x0	Output message word

A.4.20 AES Output Vector 0 (OV0)

Instance Name
OV0

Offset
0x4C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	OV0																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 155: AES Output Vector 0 (OV0)

Bits	Name	Type	Reset	Description
31:0	OV0	R	0x0	Byte 0-3 of output vector

A.4.21 AES Output Vector 1 (OV1)

Instance Name
OV1

Offset
0x50

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	OV1																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 156: AES Output Vector 1 (OV1)

Bits	Name	Type	Reset	Description
31:0	OV1	R	0x0	Byte 4-7 of output vector

A.4.22 AES Output Vector 2 (OV2)

Instance Name Offset
OV2 0x54

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	OV2																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 157: AES Output Vector 2 (OV2)

Bits	Name	Type	Reset	Description
31:0	OV2	R	0x0	Byte 8-11 of output vector

A.4.23 AES Output Vector 3 (OV3)

Instance Name Offset
OV3 0x58

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	OV3																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 158: AES Output Vector 3 (OV3)

Bits	Name	Type	Reset	Description
31:0	OV3	R	0x0	Byte 12-15 of output vector

A.4.24 AES Interrupt Status Register (ISR)

Instance Name Offset
ISR 0x5C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																														STATUS[2]	STATUS[1]	STATUS[0]
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	

Table 159: AES Interrupt Status Register (ISR)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2	STATUS[2]	R	0x0	Status of AES output FIFO empty interrupt 0x0: AES output FIFO empty interrupt not occurred 0x1: AES output FIFO empty interrupt occurred

Table 159: AES Interrupt Status Register (ISR)

Bits	Name	Type	Reset	Description
1	STATUS[1]	R	0x0	Status of AES input FIFO full interrupt 0x0: AES input FIFO full interrupt not occurred 0x1: AES input FIFO full interrupt occurred
0	STATUS[0]	R	0x0	Status of AES output FIFO empty interrupt 0x0: AES operation done interrupt not occurred 0x1: AES operation done interrupt occurred

A.4.25 AES Interrupt Mask Register (IMR)

Instance Name

IMR

Offset

0x60

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																													MASK[2]	MASK[1]	MASK[0]
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	1

Table 160: AES Interrupt Mask Register (IMR)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2	MASK[2]	R/W	0x1	Mask of AES output FIFO empty interrupt 0x0: Enable AES output FIFO empty interrupt 0x1: Disable AES output FIFO empty interrupt
1	MASK[1]	R/W	0x1	Mask of AES input FIFO full interrupt 0x0: Enable AES input FIFO full interrupt 0x1: Disable AES input FIFO full interrupt
0	MASK[0]	R/W	0x1	Mask of AES operation done interrupt 0x0: Enable AES operation done interrupt 0x1: Disable AES operation done interrupt

A.4.26 AES Interrupt Raw Status Register (IRSR)

Instance Name
IRSR

Offset
0x64

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																													STATUS_RAW[2]	STATUS_RAW[1]	STATUS_RAW[0]
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Table 161: AES Interrupt Raw Status Register (IRSR)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2	STATUS_RAW[2]	R	0x0	AES output FIFO empty interrupt raw status regardless of mask 0x0: AES output FIFO empty interrupt not occurred 0x1: AES output FIFO empty interrupt
1	STATUS_RAW[1]	R	0x0	AES input FIFO full interrupt raw status regardless of mask 0x0: AES no input FIFO full interrupt not occurred 0x1: AES no input FIFO full interrupt occurred
0	STATUS_RAW[0]	R	0x0	AES operation done interrupt raw status regardless of mask 0x0: AES operation done interrupt not occurred 0x1: AES operation done interrupt occurred

A.4.27 AES Interrupt Clear Register (ICR)

Instance Name
ICR

Offset
0x68

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																													CLEAR[2]	CLEAR[1]	CLEAR[0]
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Table 162: AES Interrupt Clear Register (ICR)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 162: AES Interrupt Clear Register (ICR)

Bits	Name	Type	Reset	Description
2	CLEAR[2]	R/W	0x0	Clearance of AES output FIFO empty interrupt status and raw status Write '1' will generate a single-cycle pulse that clears both AES output FIFO empty interrupt status and raw status
1	CLEAR[1]	R/W	0x0	Clearance of AES input FIFO full interrupt status and raw status Write '1' will generate a single-cycle pulse that clears both AES input FIFO full interrupt status and raw status
0	CLEAR[0]	R/W	0x0	Clearance of AES operation done interrupt status and raw status Write '1' will generate a single-cycle pulse that clears both AES operation done interrupt status and raw status

A.4.28 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x8C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								Reserved							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	1	0	0	1	0

Table 163: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	Reserved	R	0x12	Reserved. Do not change the reset value.

A.4.29 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x90

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																												Reserved			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		?	0	0

Table 164: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.4.30 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x94

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 165: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.4.31 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x98

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 166: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.4.32 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x9C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 167: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.4.33 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0xA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 168: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.4.34 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0xA4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 169: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.4.35 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0xA8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																Reserved															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 170: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.4.36 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0xAC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							Reserved								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0

Table 171: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:9	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
8:0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.4.37 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0xB0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 172: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.4.38 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0xB4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 173: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.4.39 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0xB8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 174: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.4.40 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0xBC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 175: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.5 CRC Register Information

Table 176: CRC Register Summary

Offset	Name	Description	Details
0x00	ISR	Interrupt Status Register	Page: 127
0x04	IRSR	Interrupt Raw Status Register	Page: 128
0x08	ICR	Interrupt Clear Register	Page: 128
0x0C	IMR	Interrupt Mask Register	Page: 129
0x10	CTRL	CRC Module Control Register	Page: 129
0x14	STREAM_LEN_M1	Stream Length Minus 1 Register	Page: 130
0x18	STREAM_IN	Stream Input Register	Page: 130
0x1C	RESULT	CRC Calculation Result	Page: 131
0x3C	RESERVED	Reserved	Page: 131

A.5.1 Interrupt Status Register (ISR)

Instance Name

ISR

Offset

0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														STATUS	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 177: Interrupt Status Register (ISR)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	STATUS	R	0x0	CRC calculation done interrupt status after mask status[0]: CRC done 0x0: interrupt is not occurred 0x1: interrupt is occurred

A.5.2 Interrupt Raw Status Register (IRSR)

Instance Name

IRSR

Offset

0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															STATUS_RAW
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 178: Interrupt Raw Status Register (IRSR)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	STATUS_RAW	R	0x0	CRC calculation done interrupt raw status regardless of mask 0x0: interrupt is not occurred 0x1: interrupt is occurred

A.5.3 Interrupt Clear Register (ICR)

Instance Name

ICR

Offset

0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															CLEAR
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 179: Interrupt Clear Register (ICR)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLEAR	W	0x0	Clearance of status[0] and status_raw[0] Write '1' will generate a single-cycle pulse that clears both status[0] and status_raw[0]

A.5.4 Interrupt Mask Register (IMR)

Instance Name
IMR

Offset
0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														MASK	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1

Table 180: Interrupt Mask Register (IMR)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	MASK	R/W	0x1	Mask of interrupt 0x1: disable generation of IRQ and corresponding status[0] 0x0: enable generation of IRQ and corresponding status[0]

A.5.5 CRC Module Control Register (CTRL)

Instance Name
CTRL

Offset
0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																												MODE			ENABLE
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	

Table 181: CRC Module Control Register (CTRL)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:1	MODE	R/W	0x0	CRC mode select 0x0: $x^{**16}+x^{**12}+x^{**5}+1$ (CRC-16-CCITT, CRC-CCITT) 0x1: $x^{**16}+x^{**15}+x^{**2}+1$ (CRC-16, CRC-16-IBM, CRC-16-ANSI) 0x2: $x^{**16}+x^{**15}+x^{**11}+x^{**9}+x^{**8}+x^{**7}+x^{**5}+x^{**4}+x^{**2}+x+1$ (CRC-16-T10-DIF) 0x3: $x^{**32}+x^{**26}+x^{**23}+x^{**22}+x^{**16}+x^{**12}+x^{**11}+x^{**10}+x^{**8}+x^{**7}+x^{**5}+x^{**4}+x^{**2}+x+1$ (CRC-32-IEEE802.3) 0x4: $x^{**16}+x^{**13}+x^{**12}+x^{**11}+x^{**10}+x^{**8}+x^{**6}+x^{**5}+x^{**2}+1$ (CRC-16-DNP) others: reserved

Table 181: CRC Module Control Register (CTRL)

Bits	Name	Type	Reset	Description
0	ENABLE	R/W	0x0	CRC calculate enable 0x1 : enable CRC calculation, it is automatically cleared when the CRC calculation is finished 0x0 : disable CRC calculation

A.5.6 Stream Length Minus 1 Register (STREAM_LEN_M1)

Instance Name Offset
STREAM_LEN_M1 0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	LENGTH_M1																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 182: Stream Length Minus 1 Register (STREAM_LEN_M1)

Bits	Name	Type	Reset	Description
31:0	LENGTH_M1	R/W	0x0	Input stream length minus 1 (in unit of byte) Example: 0x0000_00FF : input stream length of 256 bytes 0x0000_0000 : input stream length of 1 byte

A.5.7 Stream Input Register (STREAM_IN)

Instance Name Offset
STREAM_IN 0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DATA																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 183: Stream Input Register (STREAM_IN)

Bits	Name	Type	Reset	Description
31:0	DATA	R/W	0x0	Stream input data register

A.5.8 CRC Calculation Result (RESULT)

Instance Name
RESULT

Offset
0x1C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DATA																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 184: CRC Calculation Result (RESULT)

Bits	Name	Type	Reset	Description
31:0	DATA	R	0x0	CRC calculation result

A.5.9 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x3C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								Reserved							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	1	0	0	1	0	

Table 185: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	Reserved	R	0x12	Reserved. Do not change the reset value.

A.6 I2C Register Information

Table 186: I2C Register Summary

Offset	Name	Description	Details
0x00	CON		Page: 133
0x04	TAR		Page: 135
0x08	SAR		Page: 136
0x0C	HS_MADDR		Page: 137
0x10	DATA_CMD		Page: 137
0x14	SS_SCL_HCNT		Page: 138
0x18	SS_SCL_LCNT		Page: 139
0x1C	FS_SCL_HCNT		Page: 140
0x20	FS_SCL_LCNT		Page: 140
0x24	HS_SCL_HCNT		Page: 141
0x28	HS_SCL_LCNT		Page: 142
0x2C	INTR_STAT		Page: 142
0x30	INTR_MASK		Page: 145
0x34	RAW_INTR_STAT		Page: 147
0x38	RX_TL		Page: 149
0x3C	TX_TL		Page: 150
0x40	CLR_INTR		Page: 150
0x44	CLR_RX_UNDER		Page: 151
0x48	CLR_RX_OVER		Page: 151
0x4C	CLR_TX_OVER		Page: 152
0x50	CLR_RD_REQ		Page: 152
0x54	CLR_TX_ABRT		Page: 153
0x58	CLR_RX_DONE		Page: 153
0x5C	CLR_ACTIVITY		Page: 154
0x60	CLR_STOP_DET		Page: 154
0x64	CLR_START_DET		Page: 155
0x68	CLR_GEN_CALL		Page: 155
0x6C	ENABLE		Page: 156

Table 186: I2C Register Summary

Offset	Name	Description	Details
0x70	STATUS		Page: 156
0x74	TXFLR		Page: 158
0x78	RXFLR		Page: 158
0x7C	RESERVED	Reserved	Page: 158
0x80	TX_ABRT_SOURCE		Page: 159
0x84	RESERVED	Reserved	Page: 161
0x88	DMA_CR		Page: 162
0x8C	DMA_TDLR		Page: 162
0x90	DMA_RDLR		Page: 163
0x94	SDA_SETUP		Page: 163
0x98	ACK_GENERAL_CALL		Page: 164
0x9C	ENABLE_STATUS		Page: 164
0xA0	RESERVED	Reserved	Page: 166
0xA4	RESERVED	Reserved	Page: 166
0xF4	RESERVED	Reserved	Page: 166
0xF8	RESERVED	Reserved	Page: 167
0xFC	RESERVED	Reserved	Page: 167

A.6.1 (IC_CON)

I2C Control Register

Instance Name

CON

Offset

0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Field	Reserved																										SLAVE_DISABLE	RESTART_EN	BITADDR10_MASTER_RD_ONLY		BITADDR10_SLAVE	SPEED		MASTER_MODE
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	1	1	1	1	1		

Table 187: (IC_CON)

Bits	Name	Type	Reset	Description
31:7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
6	SLAVE_DISABLE	R/W	0x1	This bit controls whether I2C has its slave disabled, which means once the preseln signal is applied, then this bit takes on the value of the configuration parameter SLAVE_DISABLE. You have the choice of having the slave enabled or disabled after reset is applied, which means software does not have to configure the slave. By default, the slave is always enabled (in reset state as well). If you need to disable it after reset, set this bit to 1. If this bit is set (slave is disabled), I2C functions only as a master and does not perform any action that requires a slave. NOTE: Software should ensure that if this bit is written with 0, then bit 0 should also be written with a 0. 0x0: slave is enabled 0x1: slave is disabled
5	RESTART_EN	R/W	0x1	Determines whether RESTART conditions may be sent when acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in several I2C operations. When RESTART is disabled, the master is prohibited from performing the following functions: - Change direction within a transfer (split) - Send a START BYTE - High-speed mode operation - Combined format transfers in 7-bit addressing modes - Read operation with a 10-bit address - Send multiple bytes per transfer By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple I2C transfers. If the above operations are performed, it will result in setting bit 6 (TX_ABRT) of the RAW_INTR_STAT register. 0x0: disable 0x1: enable
4	BITADDR10_MASTE R_RD_ONLY	R	0x1	The function of this bit is handled by bit 12 of TAR register, and becomes a read-only copy. 0x0: 7-bit addressing 0x1: 10-bit addressing
3	BITADDR10_SLAVE	R/W	0x1	When acting as a slave, this bit controls whether the I2C responds to 7- or 10-bit addresses. 0x0: 7-bit addressing. The I2C ignores transactions that involve 10-bit addressing; for 7-bit addressing, only the lower 7 bits of the IC_SAR register are compared. 0x1: 10-bit addressing. The I2C responds to only 10-bit addressing transfers that match the full 10 bits of the IC_SAR register.

Table 187: (IC_CON)

Bits	Name	Type	Reset	Description
2:1	SPEED	R/W	0x3	These bits control at which speed the I2C operates; its setting is relevant only if one is operating the I2C in master mode. Hardware protects against illegal values being programmed by software. This register should be programmed only with a value in the range of 1 to 3; otherwise, hardware updates this register with the value of 3. 0x1: standard mode (100 kbit/s) 0x2: fast mode (400 kbit/s) 0x3: high speed mode (3.4 Mbit/s)
0	MASTER_MODE	R/W	0x1	This bit controls whether the I2C master is enabled. • NOTE: Software should ensure that if this bit is written with '1' then bit 6 should also be written with a '1'. 0x0: master disabled 0x1: master enabled

A.6.2 (IC_TAR)

I2C Target Address Register

Instance Name

Offset

TAR

0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																			BITADDR10_MASTER	SPECIAL	GC_OR_START	TAR									
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	0	0	0	0	0	1	0	1	0	1	0	1

Table 188: (IC_TAR)

Bits	Name	Type	Reset	Description
31:13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12	BITADDR10_MASTER	R/W	0x1	This bit controls whether the I2C starts its transfers in 7- or 10-bit addressing mode when acting as a master. 0x0: 7-bit addressing 0x1: 10-bit addressing

Table 188: (IC_TAR)

Bits	Name	Type	Reset	Description
11	SPECIAL	R/W	0x0	This bit indicates whether software performs a General Call or START BYTE command. 0x0: ignore bit 10 GC_OR_START and use IC_TAR normally 0x1: perform special I2C command as specified in GC_OR_START bit
10	GC_OR_START	R/W	0x0	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a General Call or START byte command is to be performed by the I2C. 0x0: General Call Address after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. The I2C remains in General Call mode until the SPECIAL bit value (bit 11) is cleared. 0x1: START BYTE
9:0	TAR	R/W	0x55	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits. If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a slave.

A.6.3 (IC_SAR)

I2C Slave Address Register

Instance Name
SAR

Offset
0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																						SAR									
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	1	0	1	0	1	0	1

Table 189: (IC_SAR)

Bits	Name	Type	Reset	Description
31:10	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 189: (IC_SAR)

Bits	Name	Type	Reset	Description
9:0	SAR	R/W	0x55	The SAR holds the slave address when the I2C is operating as a slave. For 7-bit addressing, only SAR[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the ENABLE register being set to 0. Writes at other times have no effect. Note: The default values cannot be any of the reserved address locations: that is, 0x00 to 0x07, or 0x78 to 0x7f. The correct operation of the device is not guaranteed if you program the SAR or TAR to a reserved value.

A.6.4 (IC_HS_MADDR)

I2C High Speed Master Mode Code Address Register

Instance Name
HS_MADDR

Offset
0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																													HS_MAR		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1

Table 190: (IC_HS_MADDR)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	HS_MAR	R/W	0x1	This bit field holds the value of the I2C HS mode master code. HS-mode master codes are reserved 8-bit codes (00001xxx) that are not used for slave addressing or other purposes. Each master has its unique master code; up to eight high-speed mode masters can be present on the same I2C bus system. Valid values are from 0 to 7. This register can be written only when the I2C interface is disabled, which corresponds to the ENABLE register being set to 0. Writes at other times have no effect.

A.6.5 (IC_DATA_CMD)

I2C Rx/Tx Data Buffer and Command Register

Instance Name
DATA_CMD

Offset
0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Field	Reserved																							CMD	DAT									
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0		

Table 191: (IC_DATA_CMD)

Bits	Name	Type	Reset	Description
31:9	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
8	CMD	R/W	0x0	This bit controls whether a read or a write is performed. This bit does not control the direction when the I2C acts as a slave. It controls only the direction when it acts as a master. When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In slave-receiver mode, this bit is a 'don't care' because writes to this register are not required. In slave-transmitter mode, a '0' indicates that CPU data is to be transmitted and as DAT or IC_DATA_CMD[7:0]. When programming this bit, you should remember the following: attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the IC_RAW_INTR_STAT register), unless bit 11 (SPECIAL) in the IC_TAR register has been cleared. If a '1' is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs. NOTE: It is possible that while attempting a master I2C read transfer on I2C, a RD_REQ interrupt may have occurred simultaneously due to a remote I2C master addressing I2C. In this type of scenario, I2C ignores the IC_DATA_CMD write, generates a TX_ABRT interrupt, and waits to service the RD_REQ interrupt. 0x1: Read 0x0: Write
7:0	DAT	R/W	0x0	This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the I2C. However, when you read this register, these bits return the value of data received on the I2C interface.

A.6.6 (IC_SS_SCL_HCNT)

Standard Speed I2C Clock SCL High Count Register

Instance Name

SS_SCL_HCNT

Offset

0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																SS_SCL_HCNT															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0

Table 192: (IC_SS_SCL_HCNT)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 192: (IC_SS_SCL_HCNT)

Bits	Name	Type	Reset	Description
15:0	SS_SCL_HCNT	R/W	0x1F4	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed. These values apply only if the ic_clk is set to the given frequency in the table. This register can be written only when the I2C interface is disabled which corresponds to the ENABLE register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of the I2C. The lower byte must be programmed first. Then the upper byte is programmed.

A.6.7 (IC_SS_SCL_LCNT)

Standard Speed I2C Clock SCL Low Count Register

Instance Name
SS_SCL_LCNT

Offset
0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																SS_SCL_LCNT															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0

Table 193: (IC_SS_SCL_LCNT)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	SS_SCL_LCNT	R/W	0x24C	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed. The table below shows some sample IC_SS_SCL_LCNT calculations. These values apply only if the ic_clk is set to the given frequency in the table. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted, results in 8 being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of I2C. The lower byte must be programmed first, and then the upper byte is programmed.

A.6.8 (IC_FS_SCL_HCNT)

Fast Speed I2C Clock SCL High Count Register

Instance Name

FS_SCL_HCNT

Offset

0x1C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																FS_SCL_HCNT															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1

Table 194: (IC_FS_SCL_HCNT)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	FS_SCL_HCNT	R/W	0x4B	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the I2C. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter HC_COUNT_VALUES is set to 1, this register is read only.

A.6.9 (IC_FS_SCL_LCNT)

Fast Speed I2C Clock SCL Low Count Register

Instance Name

FS_SCL_LCNT

Offset

0x20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																FS_SCL_LCNT															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1

Table 195: (IC_FS_SCL_LCNT)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 195: (IC_FS_SCL_LCNT)

Bits	Name	Type	Reset	Description
15:0	FS_SCL_LCNT	R/W	0xA3	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. This register can be written only when the I2C interface is disabled, which corresponds to the ENABLE register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the I2C. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than 8 then the count value gets changed to 8.

A.6.10 (IC_HS_SCL_HCNT)

High Speed I2C Clock SCL High Count Register

Instance Name

HS_SCL_HCNT

Offset

0x24

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																HS_SCL_HCNT															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Table 196: (IC_HS_SCL_HCNT)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	HS_SCL_HCNT	R/W	0x8	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high period count for high speed. The SCL High time depends on the loading of the bus. For 100pF loading, the SCL High time is 60ns; for 400pF loading, the SCL High time is 120ns. This register goes away and becomes read-only returning 0s if MAX_SPEED_MODE != high. This register can be written only when the I2C interface is disabled, which corresponds to the ENABLE register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the I2C. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter HC_COUNT_VALUES is set to 1, this register is read only.

A.6.11 (IC_HS_SCL_LCNT)

High Speed I2C Clock SCL Low Count Register

Instance Name

HS_SCL_LCNT

Offset

0x28

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																HS_SCL_LCNT															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Table 197: (IC_HS_SCL_LCNT)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	HS_SCL_LCNT	R/W	0x14	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for high speed. The SCL low time depends on the loading of the bus. For 100pF loading, the SCL low time is 160ns; for 400pF loading, the SCL low time is 320ns. This register goes away and becomes read-only returning 0s if MAX_SPEED_MODE != high. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the I2C. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than 8 then the count value gets changed to 8. When the configuration parameter HC_COUNT_VALUES is set to 1, this register is read only.

A.6.12 (IC_INTR_STAT)

I2C Interrupt Status Register

Instance Name

INTR_STAT

Offset

0x2C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																				R_GEN_CALL	R_START_DET	R_STOP_DET	R_ACTIVITY	R_RX_DONE	R_TX_ABORT	R_RD_REQ	R_TX_EMPTY	R_TX_OVER	R_RX_FULL	R_RX_OVER	R_RX_UNDER
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0

Table 198: (IC_INTR_STAT)

Bits	Name	Type	Reset	Description
31:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	R_GEN_CALL	R	0x0	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling I2C or when the CPU reads bit 0 of the CLR_GEN_CALL register. I2C stores the received data in the Rx buffer.
10	R_START_DET	R	0x0	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode.
9	R_STOP_DET	R	0x0	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode.
8	R_ACTIVITY	R	0x0	This bit captures I2C activity and stays set until it is cleared. There are four ways to clear it: - Disabling the I2C - Reading the IC_CLR_ACTIVITY register - Reading the IC_CLR_INTR register - System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the I2C module is idle, this bit remains set until cleared, indicating that there was activity on the bus.
7	R_RX_DONE	R	0x0	When the I2C is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.
6	R_TX_ABRT	R	0x0	This bit indicates if I2C, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a 'transmit abort'. When this bit is set to 1, the TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. NOTE: The I2C flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface.
5	R_RD_REQ	R	0x0	This bit is set to 1 when I2C is acting as a slave and another I2C master is attempting to read data from I2C. The I2C holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the DATA_CMD register. This bit is set to 0 just after the processor reads the CLR_RD_REQ register.

Table 198: (IC_INTR_STAT)

Bits	Name	Type	Reset	Description
4	R_TX_EMPTY	R	0x0	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en=0, this bit is set to 0.
3	R_TX_OVER	R	0x0	Set during transmit if the transmit buffer is filled to TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
2	R_RX_FULL	R	0x0	Set when the receive buffer reaches or goes above the RX_TL threshold in the RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.
1	R_RX_OVER	R	0x0	Set if the receive buffer is completely filled to RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The I2C acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
0	R_RX_UNDER	R	0x0	Set if the processor attempts to read the receive buffer when it is empty by reading from the DATA_CMD register. If the module is disabled (ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.

A.6.13 (IC_INTR_MASK)

I2C Interrupt Mask Register

Instance Name

INTR_MASK

Offset

0x30

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																				M_GEN_CALL	M_START_DET	M_STOP_DET	M_ACTIVITY	M_RX_DONE	M_TX_ABORT	M_RD_REQ	M_TX_EMPTY	M_TX_OVER	M_RX_FULL	M_RX_OVER	M_RX_UNDER
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	0	0	0	1	1	1	1	1	1	1	1

Table 199: (IC_INTR_MASK)

Bits	Name	Type	Reset	Description
31:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	M_GEN_CALL	R/W	0x1	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling I2C or when the CPU reads bit 0 of the CLR_GEN_CALL register. I2C stores the received data in the Rx buffer.
10	M_START_DET	R/W	0x0	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode.
9	M_STOP_DET	R/W	0x0	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode.
8	M_ACTIVITY	R/W	0x0	This bit captures I2C activity and stays set until it is cleared. There are four ways to clear it: - Disabling the I2C - Reading the IC_CLR_ACTIVITY register - Reading the IC_CLR_INTR register - System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the I2C module is idle, this bit remains set until cleared, indicating that there was activity on the bus.
7	M_RX_DONE	R/W	0x1	When the I2C is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.

Table 199: (IC_INTR_MASK)

Bits	Name	Type	Reset	Description
6	M_TX_ABRT	R/W	0x1	This bit indicates if I2C, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a 'transmit abort'. When this bit is set to 1, the TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. NOTE: The I2C flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface.
5	M_RD_REQ	R/W	0x1	This bit is set to 1 when i2c is acting as a slave and another I2C master is attempting to read data from I2C. The I2C holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the DATA_CMD register. This bit is set to 0 just after the processor reads the CLR_RD_REQ register.
4	M_TX_EMPTY	R/W	0x1	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en=0, this bit is set to 0.
3	M_TX_OVER	R/W	0x1	Set during transmit if the transmit buffer is filled to TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
2	M_RX_FULL	R/W	0x1	Set when the receive buffer reaches or goes above the RX_TL threshold in the RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.
1	M_RX_OVER	R/W	0x1	Set if the receive buffer is completely filled to RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The I2C acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.

Table 199: (IC_INTR_MASK)

Bits	Name	Type	Reset	Description
0	M_RX_UNDER	R/W	0x1	Set if the processor attempts to read the receive buffer when it is empty by reading from the DATA_CMD register. If the module is disabled (ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.

A.6.14 (IC_RAW_INTR_STAT)

I2C Raw Interrupt Status Register

Instance Name
RAW_INTR_STAT

Offset
0x34

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																				GEN_CALL	START_DET	STOP_DET	ACTIVITY	RX_DONE	TX_ABORT	RD_REQ	TX_EMPTY	TX_OVER	RX_FULL	RX_OVER	RX_UNDER
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0

Table 200: (IC_RAW_INTR_STAT)

Bits	Name	Type	Reset	Description
31:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	GEN_CALL	R	0x0	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling I2C or when the CPU reads bit 0 of the CLR_GEN_CALL register. I2C stores the received data in the Rx buffer.
10	START_DET	R	0x0	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode.
9	STOP_DET	R	0x0	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode.
8	ACTIVITY	R	0x0	This bit captures I2C activity and stays set until it is cleared. There are four ways to clear it: - Disabling the I2C - Reading the IC_CLR_ACTIVITY register - Reading the IC_CLR_INTR register - System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the I2C module is idle, this bit remains set until cleared, indicating that there was activity on the bus.

Table 200: (IC_RAW_INTR_STAT)

Bits	Name	Type	Reset	Description
7	RX_DONE	R	0x0	When the I2C is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.
6	TX_ABRT	R	0x0	This bit indicates if I2C, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a 'transmit abort'. When this bit is set to 1, the TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. NOTE: The I2C flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface.
5	RD_REQ	R	0x0	This bit is set to 1 when I2C is acting as a slave and another I2C master is attempting to read data from I2C. The I2C holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the DATA_CMD register. This bit is set to 0 just after the processor reads the CLR_RD_REQ register.
4	TX_EMPTY	R	0x0	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en=0, this bit is set to 0.
3	TX_OVER	R	0x0	Set during transmit if the transmit buffer is filled to TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
2	RX_FULL	R	0x0	Set when the receive buffer reaches or goes above the RX_TL threshold in the RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.

Table 200: (IC_RAW_INTR_STAT)

Bits	Name	Type	Reset	Description
1	RX_OVER	R	0x0	Set if the receive buffer is completely filled to RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The I2C acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
0	RX_UNDER	R	0x0	Set if the processor attempts to read the receive buffer when it is empty by reading from the DATA_CMD register. If the module is disabled (ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.

A.6.15 (IC_RX_TL)

I2C Receive FIFO Threshold Register

Instance Name

RX_TL

Offset

0x38

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											RX_TL				
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0

Table 201: (IC_RX_TL)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4:0	RX_TL	R/W	0x0	Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the RX_FULL interrupt (bit 2 in RAW_INTR_STAT register). The valid range is 0-255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries.

A.6.16 (IC_TX_TL)

I2C Transmit FIFO Threshold Register

Instance Name

TX_TL

Offset

0x3C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											TX_TL				
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	

Table 202: (IC_TX_TL)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4:0	TX_TL	R/W	0x0	Transmit FIFO Threshold Level Controls the level of entries (or below) that trigger the TX_EMPTY interrupt (bit 4 in RAW_INTR_STAT register). The valid range is 0-255, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 255 sets the threshold for 255 entries.

A.6.17 (IC_CLR_INTR)

Clear Combined and Individual Interrupt Register

Instance Name

CLR_INTR

Offset

0x40

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															CLR_INTR
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 203: (IC_CLR_INTR)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLR_INTR	R	0x0	Read this register to clear the combined interrupt, all individual interrupts, and the TX_ABRT_SOURCE register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the TX_ABRT_SOURCE register for an exception to clearing TX_ABRT_SOURCE.

A.6.18 (IC_CLR_RX_UNDER)

Clear RX_UNDER Interrupt Register

Instance Name
CLR_RX_UNDER**Offset**
0x44

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															CLR_RX_UNDER
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 204: (IC_CLR_RX_UNDER)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLR_RX_UNDER	R	0x0	Read this register to clear the RX_UNDER interrupt (bit 0) of the RAW_INTR_STAT register. • Reset value: 0x0

A.6.19 (IC_CLR_RX_OVER)

Clear RX_OVER Interrupt Register

Instance Name
CLR_RX_OVER**Offset**
0x48

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															CLR_RX_OVER
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 205: (IC_CLR_RX_OVER)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLR_RX_OVER	R	0x0	Read this register to clear the RX_OVER interrupt (bit 1) of the RAW_INTR_STAT register.

A.6.20 (IC_CLR_TX_OVER)

Clear TX_OVER Interrupt Register

Instance Name

CLR_TX_OVER

Offset

0x4C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															CLR_TX_OVER
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 206: (IC_CLR_TX_OVER)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLR_TX_OVER	R	0x0	Read this register to clear the TX_OVER interrupt (bit 3) of the RAW_INTR_STAT register.

A.6.21 (IC_CLR_RD_REQ)

Clear RD_REQ Interrupt Register

Instance Name

CLR_RD_REQ

Offset

0x50

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															CLR_RD_REQ
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 207: (IC_CLR_RD_REQ)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLR_RD_REQ	R	0x0	Read this register to clear the RD_REQ interrupt (bit 5) of the RAW_INTR_STAT register.

A.6.22 (IC_CLR_TX_ABRT)

Clear TX_ABRT Interrupt Register

Instance Name

CLR_TX_ABRT

Offset

0x54

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														CLR TX ABRT	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 208: (IC_CLR_TX_ABRT)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLR_TX_ABRT	R	0x0	Read this register to clear the TX_ABRT interrupt (bit 6) of the RAW_INTR_STAT register, and the TX_ABRT_SOURCE register. This also releases the TX FIFO from the flushed/reset state, allowing more writes to the TX FIFO. Refer to Bit 9 of the TX_ABRT_SOURCE register for an exception to clearing TX_ABRT_SOURCE.

A.6.23 (IC_CLR_RX_DONE)

Clear RX_DONE Interrupt Register

Instance Name

CLR_RX_DONE

Offset

0x58

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																														CLR	RX	DONE
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	

Table 209: (IC_CLR_RX_DONE)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLR_RX_DONE	R	0x0	Read this register to clear the RX_DONE interrupt (bit 7) of the RAW_INTR_STAT register.

A.6.24 (IC_CLR_ACTIVITY)

Clear ACTIVITY Interrupt Register

Instance Name

CLR_ACTIVITY

Offset

0x5C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															CLR_ACTIVITY
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 210: (IC_CLR_ACTIVITY)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLR_ACTIVITY	R	0x0	Reading this register clears the ACTIVITY interrupt if the I2C is not active anymore. If the I2C module is still active on the bus, the ACTIVITY interrupt bit continues to be set. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY interrupt (bit 8) of the RAW_INTR_STAT register.

A.6.25 (IC_CLR_STOP_DET)

Clear STOP_DET Interrupt Register

Instance Name

CLR_STOP_DET

Offset

0x60

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															CLR_STOP_DET
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 211: (IC_CLR_STOP_DET)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLR_STOP_DET	R	0x0	Read this register to clear the STOP_DET interrupt (bit 9) of the RAW_INTR_STAT register.

A.6.26 (IC_CLR_START_DET)

Clear START_DET Interrupt Register

Instance Name
CLR_START_DET**Offset**
0x64

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														CLR_START_DET	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 212: (IC_CLR_START_DET)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLR_START_DET	R	0x0	Read this register to clear the START_DET interrupt (bit 10) of the RAW_INTR_STAT register.

A.6.27 (IC_CLR_GEN_CALL)

Clear GEN_CALL Interrupt Register

Instance Name
CLR_GEN_CALL**Offset**
0x68

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														CLR_GEN_CALL	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 213: (IC_CLR_GEN_CALL)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLR_GEN_CALL	R	0x0	Read this register to clear the GEN_CALL interrupt (bit 11) of RAW_INTR_STAT register.

A.6.28 (IC_ENABLE)

I2C Enable Register

Instance Name
ENABLE

Offset
0x6C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															ENABLE
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 214: (IC_ENABLE)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	ENABLE	R/W	0x0	Controls whether the I2C is enabled. However, it is important that care be taken to ensure that i2c is disabled properly. When I2C is disabled, the following occurs: - The TX FIFO and RX FIFO get flushed. - Status bits in the INTR_STAT register are still active until i2c goes into IDLE state. If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the I2C stops the current transfer at the end of the current byte and does not acknowledge the transfer. In systems with asynchronous pclk and ic_clk when CLK_TYPE parameter set to asynchronous, there is a two ic_clk delay when enabling or disabling the I2C. 0x0: Disables I2C (TX and RX FIFOs are held in an erased state) 0x1: Enables I2C Software can disable I2C while it is active. However, it is important that care be taken to ensure that I2C is disabled properly. When I2C is disabled, the following occurs:

A.6.29 (IC_STATUS)

I2C Status Register

Instance Name
STATUS

Offset
0x70

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																										SLV_ACTIVITY	MST_ACTIVITY	RFF	RFNE	TFE	TFNF	ACTIVITY
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	1	1	0	

Table 215: (IC_STATUS)

Bits	Name	Type	Reset	Description
31:7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
6	SLV_ACTIVITY	R	0x0	Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0x0: Slave FSM is in IDLE state so the Slave part of I2C is not Active 0x1: Slave FSM is not in IDLE state so the Slave part of I2C is Active
5	MST_ACTIVITY	R	0x0	Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set. Note: IC_STATUS[0]-that is, ACTIVITY bit-is the OR of SLV_ACTIVITY and MST_ACTIVITY bits. 0x0: Master FSM is in IDLE state so the Master part of I2C is not Active 0x1: Master FSM is not in IDLE state so the Master part of I2C is Active Note IC_STATUS[0]-that is, ACTIVITY bit-is the OR of SLV_ACTIVITY and MST_ACTIVITY bits.
4	RFF	R	0x0	Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0x0: Receive FIFO is not full 0x1: Receive FIFO is full
3	RFNE	R	0x0	Receive FIFO Not Empty. This bit is set when the receive FIFO contains one or more entries; it is cleared when the receive FIFO is empty. 0x0: Receive FIFO is empty 0x1: Receive FIFO is not empty
2	TFE	R	0x1	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request 0x0: Transmit FIFO is not empty 0x1: Transmit FIFO is empty
1	TFNF	R	0x1	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0x0: Transmit FIFO is full 0x1: Transmit FIFO is not full
0	ACTIVITY	R	0x0	I2C Activity Status. • Reset value: 0x0

A.6.30 (IC_TXFLR)

I2C Transmit FIFO Level Register

Instance Name

TXFLR

Offset

0x74

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										TXFLR					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	

Table 216: (IC_TXFLR)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4:0	TXFLR	R	0x0	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.

A.6.31 (IC_RXFLR)

I2C Receive FIFO Level Register

Instance Name

RXFLR

Offset

0x78

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										RXFLR					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	

Table 217: (IC_RXFLR)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4:0	RXFLR	R	0x0	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.

A.6.32 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x7C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																Reserved															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 218: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	Reserved	R/W	0x1	Reserved. Do not change the reset value.

A.6.33 (IC_TX_ABRT_SOURCE)

I2C Transmit Abort Source Register

Instance Name

TX_ABRT_SOURCE

Offset

0x80

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																ABRT_SLVRD_INTX	ABRT_SLV_ARBLOST	ABRT_SLVFLUSH_TXFIFO	ARB_LOST	ABRT_MASTER_DIS	ABRT_10B_RD_NORSTRT	ABRT_SBYTE_NORSTRT	ABRT_HS_NORSTRT	ABRT_SBYTE_ACKDET	ABRT_HS_ACKDET	ABRT_GCALL_READ	ABRT_GCALL_NOACK	ABRT_TXDATA_NOACK	ABRT_10ADDR2_NOACK	ABRT_10ADDR1_NOACK	ABRT_7B_ADDR_NOACK
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 219: (IC_TX_ABRT_SOURCE)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	ABRT_SLVRD_INTX	R	0x0	Role of I2C: Slave-Transmitter 0x1: When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of IC_DATA_CMD register.
14	ABRT_SLV_ARBLOST	R	0x0	- Note: Even though the slave never 'owns' the bus, something could go wrong on the bus. This is a failsafe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then I2C no longer own the bus. - Role of i2c: Slave-Transmitter 0x1: Slave lost the bus while transmitting data to a remote master. IC_TX_ABRT_SOURCE[12] is set at the same time.
13	ABRT_SLVFLUSH_TXFIFO	R	0x0	Role of I2C: Slave-Transmitter 0x1: Slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO.

Table 219: (IC_TX_ABRT_SOURCE)

Bits	Name	Type	Reset	Description
12	ARB_LOST	R	0x0	- Note: I2C can be both master and slave at the same time. - Role of I2C: Master-Transmitter or Slave-Transmitter 0x1: Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration.
11	ABRT_MASTER_DIS	R	0x0	Role of I2C: Master-Transmitter or Master-Receiver 0x1: User tries to initiate a Master operation with the Master mode disabled.
10	ABRT_10B_RD_NO RSTRT	R	0x0	Role of I2C: Master-Receiver 0x1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the master sends a read command in 10-bit addressing mode.
9	ABRT_SBYTE_NOR STRT	R	0x0	To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (CON[5]=1), the SPECIAL bit must be cleared (TAR[11]), or the GC_OR_START bit must be cleared (TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets reasserted. Role of I2C: Master 0x1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the user is trying to send a START Byte.
8	ABRT_HS_NORSTR T	R	0x0	Role of I2C: Master-Transmitter or Master-Receiver 0x1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the user is trying to use the master to transfer data in High Speed mode.
7	ABRT_SBYTE_ACK DET	R	0x0	Role of I2C: Master 0x1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior).
6	ABRT_HS_ACKDET	R	0x0	Role of I2C: Master 0x1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).
5	ABRT_GCALL_REA D	R	0x0	Role of I2C: Master-Transmitter 0x1: I2C in master mode sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1).
4	ABRT_GCALL_NOA CK	R	0x0	Role of I2C: Master-Transmitter 0x1: I2C in master mode sent a General Call and no slave on the bus acknowledged the General Call.

Table 219: (IC_TX_ABRT_SOURCE)

Bits	Name	Type	Reset	Description
3	ABRT_TXDATA_NO ACK	R	0x0	Role of I2C: Master-Transmitter 0x1: This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s).
2	ABRT_10ADDR2_N OACK	R	0x0	Role of I2C: Master-Transmitter or Master-Receiver 0x1: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave.
1	ABRT_10ADDR1_N OACK	R	0x0	Role of I2C: Master-Transmitter or Master-Receiver 0x1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.
0	ABRT_7B_ADDR_N OACK	R	0x0	Role of I2C: Master-Transmitter or Master-Receiver 0x1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.

A.6.34 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

Offset

RESERVED

0x84

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															Reserved
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 220: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.6.35 (IC_DMA_CR)

DMA Control Register

Instance Name

DMA_CR

Offset

0x88

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														TDMAE	RDMAE
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 221: (IC_DMA_CR)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	TDMAE	R/W	0x0	Transmit DMA Enable. //This bit enables/disables the transmit FIFO DMA channel. 0x0 : Transmit DMA disabled 0x1 : Transmit DMA enabled
0	RDMAE	R/W	0x0	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. 0x0 : Receive DMA disabled 0x1 : Receive DMA enabled

A.6.36 (IC_DMA_TDLR)

DMA Transmit Data Level Register

Instance Name

DMA_TDLR

Offset

0x8C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																												DMATDL			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	

Table 222: (IC_DMA_TDLR)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	DMATDL	R/W	0x0	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.

A.6.37 (IC_DMA_RDLR)

I2C Receive Data Level Register

Instance Name

DMA_RDLR

Offset

0x90

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										DMARDL					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	

Table 223: (IC_DMA_RDLR)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	DMARDL	R/W	0x0	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE =1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO.

A.6.38 (IC_SDA_SETUP)

I2C SDA Setup Register

Instance Name

SDA_SETUP

Offset

0x94

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								SDA_SETUP							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	1	0	0	1	0	0	

Table 224: (IC_SDA_SETUP)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SDA_SETUP	R/W	0x64	SDA Setup. It is recommended that if the required delay is 1000ns, then for an ic_clk frequency of 10 MHz, SDA_SETUP should be programmed to a value of 11.

A.6.39 (IC_ACK_GENERAL_CALL)

I2C ACK General Call Register

Instance Name

ACK_GENERAL_CALL

Offset

0x98

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														ACK GEN CALL	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1

Table 225: (IC_ACK_GENERAL_CALL)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	ACK_GEN_CALL	R/W	0x1	ACK General Call. When set to 1, I2C responds with a ACK (by asserting ic_data_oe) when it receives a General Call. Otherwise, I2C responds with a NACK (by negating ic_data_oe).

A.6.40 (IC_ENABLE_STATUS)

I2C Enable Status Register

Instance Name

ENABLE_STATUS

Offset

0x9C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																														SLV_RX_DATA_LOST	SLV_DISABLED_WHILE_BUSY	EN
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	

Table 226: (IC_ENABLE_STATUS)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 226: (IC_ENABLE_STATUS)

Bits	Name	Type	Reset	Description
2	SLV_RX_DATA_LOST	R	0x0	Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting of IC_ENABLE from 1 to 0. When read as 1, I2C is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. - NOTE: If the remote I2C master terminates the transfer with a STOP condition before the I2C has a chance to NACK a transfer, and ENABLE has been set to 0, then this bit is also set to 1. When read as 0, I2C is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. - NOTE: The CPU can safely read this bit when EN (bit 0) is read as 0.
1	SLV_DISABLED_WHILE_BUSY	R	0x0	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting of the ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the ENABLE register while: (a) I2C is receiving the address byte of the Slave-Transmitter operation from a remote master; OR, (b) address and data bytes of the Slave-Receiver operation from a remote master. When read as 1, I2C is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the slave address set in I2C (SAR register) OR if the transfer is completed before ENABLE is set to 0 but has not taken effect. - NOTE: If the remote I2C master terminates the transfer with a STOP condition before the i2c has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit will also be set to 1. When read as 0, I2C is deemed to have been disabled when there is master activity, or when the I2C bus is idle. - NOTE: The CPU can safely read this bit when EN (bit 0) is read as 0.
0	EN	R	0x0	ic_en Status. This bit always reflects the value driven on the output port ic_en. When read as 1, I2C is deemed to be in an enabled state. When read as 0, I2C is deemed completely inactive. NOTE: The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read SLV_RX_DATA_LOST (bit 2) and SLV_DISABLED_WHILE_BUSY (bit 1).

A.6.41 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0xA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							Reserved								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	1	1	0	

Table 227: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	Reserved	R/W	0x6	Reserved. Do not change the reset value.

A.6.42 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0xA4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							Reserved								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	1	0	

Table 228: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	Reserved	R/W	0x2	Reserved. Do not change the reset value.

A.6.43 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0xF4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								Reserved																							
Default	?	?	?	?	?	?	?	?	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0

Table 229: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23:0	Reserved	R	0xF_0FE E	Reserved. Do not change the reset value.

A.6.44 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0xF8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	0	1	0	1	0	0	1	0	1	0	1	0

Table 230: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x3131_ 352A	Reserved. Do not change the reset value.

A.6.45 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0xFC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0

Table 231: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x4457_ 0140	Reserved. Do not change the reset value.

A.7 QSPI Register Information

Table 232: QSPI Register Summary

Offset	Name	Description	Details
0x00	CNTL	Serial Interface Control	Page: 168
0x04	CONF	Serial Interface Configuration	Page: 170
0x08	DOUT	Serial Interface Data Out	Page: 172
0x0C	DIN	Serial Interface Data In	Page: 173
0x10	INSTR	Serial Interface Instruction	Page: 174
0x14	ADDR	Serial Interface Address	Page: 175
0x18	RDMODE	Serial Interface Read Mode	Page: 175
0x1C	HDRCNT	Serial Interface Header Count	Page: 176
0x20	DINCNT	Serial Interface Data In Count	Page: 177
0x24	TIMING	Serial Interface Timing	Page: 177
0x28	CONF2	Serial Interface Configuration 2	Page: 179
0x2C	ISR	Serial Interface Interrupt Status	Page: 180
0x30	IMR	Serial Interface Interrupt Mask	Page: 182
0x34	IRSR	Serial Interface Interrupt Raw Status	Page: 183
0x38	ISC	Serial Interface Interrupt Clear	Page: 185

A.7.1 Serial Interface Control (CNTL)

Instance Name

CNTL

Offset

0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																				WFIFO_OVRFLW	WFIFO_UNDRFLW	RFIFO_OVRFLW	RFIFO_UNDRFLW	WFIFO_FULL	WFIFO_EMPTY	RFIFO_FULL	RFIFO_EMPTY	Reserved		XFER_RDY	SS_EN
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	1	0	1	?	?	1	0

Table 233: Serial Interface Control (CNTL)

Bits	Name	Type	Reset	Description
31:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 233: Serial Interface Control (CNTL)

Bits	Name	Type	Reset	Description
11	WFIFO_OVRFLW	R	0x0	Write FIFO Overflow. 0x0: Write FIFO is not overflowed. 0x1: Write FIFO is overflowed.
10	WFIFO_UNDRFLW	R	0x0	Write FIFO Underflow. 0x0: Write FIFO is not underflowed. 0x1: Write FIFO is underflowed.
9	RFIFO_OVRFLW	R	0x0	Read FIFO Overflow. 0x0: Read FIFO is not overflowed. 0x1: Read FIFO is overflowed
8	RFIFO_UNDRFLW	R	0x0	Read FIFO Underflow. 0x0: Read FIFO is not underflowed. 0x1: Read FIFO is underflowed.
7	WFIFO_FULL	R	0x0	Write FIFO Full. 0x0: Write FIFO is not filled. 0x1: Write FIFO is filled.
6	WFIFO_EMPTY	R	0x1	Write FIFO Empty. 0x0: Write FIFO is not emptied. 0x1: Write FIFO is emptied.
5	RFIFO_FULL	R	0x0	Read FIFO Full. 0x0: Read FIFO is not filled. 0x1: Read FIFO is filled.
4	RFIFO_EMPTY	R	0x1	Read FIFO Empty. 0x0: Read FIFO is not emptied. 0x1: Read FIFO is emptied
3:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	XFER_RDY	R	0x1	Serial Interface Transfer Ready. 0x0: Serial Interface is currently transferring data. 0x1: Serial Interface is ready for a new transfer.
0	SS_EN	R/W	0x0	Serial Select Enable. 0x0: Serial select is de-activated, ss_n (serial interface select) output is driven high. 0x1: Serial select is activated, ss_n (serial interface select) output is driven low

A.7.2 Serial Interface Configuration (CONF)

Instance Name
CONF

Offset
0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																XFER_START	XFER_STOP	RW_EN	ADDR_PIN	DATA_PIN	FIFO_FLUSH	CLK_POL	CLK_PHA	Reserved	BYTE_LEN	CLK_PRESCALE						
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	?	0	0	0	0	1	0	

Table 234: Serial Interface Configuration (CONF)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	XFER_START	R/W	0x0	Transfer Start. This bit starts the serial interface I/O transfer. For read transfers, RW_EN (R04[13]) = 0, the hardware resets this bit to 0 when the number of bytes indicated in DInCnt (R20h) register have been read in from the interface. For write transfers, RW_EN (R04[13]) = 1, firmware sets XFER_STOP (R04h [14]) = 1 when all data have been written to the WFIFO and WFIFO_EMPTY (R00h [6]) = 1. Hardware resets this bit to 0 when all data have been written out the interface. 0x0: Transfer has completed. 0x1: Transfer has started.
14	XFER_STOP	R/W	0x0	Transfer Stop. This bit stops the serial interface I/O transfer. The transfer stops at either a 1-byte or 4-byte boundary, depending on the setting of BYTE_LEN (R04h [5]). Once the byte boundary is reached, the hardware resets XFER_START (R04h [15]) to 0. Hardware resets this bit to 0 after XFER_START has been reset. 0x0: Continue current transfer 0x1: Stop current transfer.
13	RW_EN	R/W	0x0	Read Write Enable. 0x0: Read data from the serial interface. 0x1: Write data to the serial interface.
12	ADDR_PIN	R/W	0x0	Address Transfer Pin. Number of pins used for transferring the content of the Addr (R14h) register. 0x0: Use one serial interface pin. 0x1: Use the number of pins as indicated in DATA_PIN (R04h [11:10]).

Table 234: Serial Interface Configuration (CONF)

Bits	Name	Type	Reset	Description
11:10	DATA_PIN	R/W	0x0	Data Transfer Pin. Number of pins used for transferring the non-command and nonaddress portions of each serial interface I/O transfer. 0x0: Use 1 serial interface pin (use in single mode). 0x1: Use 2 serial interface pins (use in dual mode). 0x2: Use 4 serial interface pins (use in quad mode). 0x3: Reserved.
9	FIFO_FLUSH	R/W	0x0	Flush Read and Write FIFOs. This bit flushes the Read and Write FIFOs. The FIFOs are emptied after being flushed. Hardware resets this bit to 0 after flushing. 0x0: Read and Write FIFOs are not flushed. 0x1: Read and Write FIFOs are flushed.
8	CLK_POL	R/W	0x0	Serial Interface Clock Polarity. Selects the serial interface clock as high or low when inactive. 0x0: Serial interface clock is low when inactive. 0x1: Serial interface clock is high when inactive.
7	CLK_PHA	R/W	0x0	Serial Interface Clock Phase. Selects the serial interface clock phase. 0x0: Data is latched at the rising edge of the serial interface clock when CLK_POL (R04h [8]) = 0, and at the falling edge of the serial interface clock when CLK_POL = 1. 0x1: Data is latched at the falling edge of the serial interface clock when CLK_POL = 0, and at the rising edge of the serial interface clock when CLK_POL = 1.
6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5	BYTE_LEN	R/W	0x0	Byte Length. The number of bytes in each serial interface I/O transfer. 0x0: 1 byte. 0x1: 4 bytes.

Table 234: Serial Interface Configuration (CONF)

Bits	Name	Type	Reset	Description
4:0	CLK_PRESCALE	R/W	0x2	Serial Interface Clock Prescaler (from SPI clock) 00h: SPI clock/1 01h: SPI clock/1 02h: SPI clock/2 03h: SPI clock/3 04h: SPI clock/4 05h: SPI clock/5 0Dh: SPI clock/13 0Eh: SPI clock/14 0Fh: SPI clock/15 10h: SPI clock/2 11h: SPI clock/2 12h: SPI clock/4 13h: SPI clock/6 14h: SPI clock/8 15h: SPI clock/10 1Dh: SPI clock/26 1Eh: SPI clock/28 1Fh: SPI clock/30

A.7.3 Serial Interface Data Out (DOUT)

Instance Name

DOUT

Offset

0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DATA_OUT																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 235: Serial Interface Data Out (DOUT)

Bits	Name	Type	Reset	Description
31:0	DATA_OUT	R/W	0x0	Serial Interface Data Out. Writing to this register stores the data in a 8X32 bit Write FIFO. After the contents of the Instruction register (R10h), the Address register (R14h), the Read Mode register (R18h) and Dummy value are transferred out to the serial interface, the data in the Write FIFO is shifted out. The serial interface clock stops when a Write FIFO empty condition occurs, WFIFO_EMPTY (R00h [6]) = 1. The clock restarts when Write FIFO is not empty, WFIFO_EMPTY (R00h [6]) = 0. When BYTE_LEN (R04h [5]) = 0, bits [7:0] are shifted out with bit 7 shifted out first and bit 0 shifted out last. When BYTE_LEN (R04h [5]) = 1, bits [7:0] are shifted out (bit 7 shifted out first and bit 0 shifted out last), followed by bits [15:8] (bit 15 shifted out first and bit 8 shifted out last), then bits [23:16] (bit 23 shifted out first and bit 16 shifted out last) and finally bits [31:24] (bit 31 shifted out first and bit 24 shifted out last). Note: To avoid a Write FIFO overflow condition (WFIFO_OVRFLW (R00h [11]) = 1), check WFIFO_FULL (R00h [7]) is equal to 0 before writing to this register.

A.7.4 Serial Interface Data In (DIN)

Instance Name

DIN

Offset

0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DATA_IN																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 236: Serial Interface Data In (DIN)

Bits	Name	Type	Reset	Description
31:0	DATA_IN	R	0x0	Serial Interface Data In. For read transfers, RW_EN (R04h [13]) = 0, data from the serial interface input pins are shifted in and stores in a 8X32 bit Read FIFO. The contents of the Read FIFO is read from this register. The serial interface clock stops when a Read FIFO full condition occurs, RFIFO_FULL (R00h [5]) = 1. The clock restarts when Read FIFO is not full, RFIFO_FULL (R00h [5]) = 0. When BYTE_LEN (R04h [5]) = 0, data is shifted into bits [7:0]. When BYTE_LEN (R04h [5]) = 1, data is shifted into bits [7:0] first, followed by bits [15:8], then bits [23:16] and finally bits [31:24]. Note: To avoid a Read FIFO underflow condition, RFIFO_UNDRFLW (R00h [8]) = 1, check RFIFO_EMPTY (R00h [4]) is equal to 0 before reading this register.

A.7.5 Serial Interface Instruction (INSTR)

Instance Name

INSTR

Offset

0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																INSTR															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 237: Serial Interface Instruction (INSTR)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	INSTR	R/W	0x0	After XFER_START (R04h [15]) is set to 1, the content of this register is shifted out to the serial interface. When INSTR_CNT (R1Ch [1:0]) = 0, the content of this register is not shifted out to the serial interface. When INSTR_CNT (R1Ch [1:0]) = 1, bits [7:0] are shifted out. When INSTR_CNT (R1Ch [1:0]) = 2, bits [15:8] are shifted out first, followed by bits [7:0].

A.7.6 Serial Interface Address (ADDR)

Instance Name
ADDR

Offset
0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ADDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 238: Serial Interface Address (ADDR)

Bits	Name	Type	Reset	Description
31:0	ADDR	R/W	0x0	Serial Interface Address. After Instr (R10h) is shifted out, the content of this register is shifted out to the serial interface. When ADDR_CNT (R1Ch [6:4]) = 0, the content of this register is not shifted out to the serial interface. When ADDR_CNT (R1Ch [6:4]) = 1, bits [7:0] are shifted out. When ADDR_CNT (R1Ch [6:4]) = 2, bits [15:8] are shifted out first, followed by bits [7:0]. When ADDR_CNT (R1Ch [6:4]) = 3, bits [23:16] are shifted out first, followed by bits [15:8], then bits [7:0]. When ADDR_CNT (R1Ch [6:4]) = 4, bits [31:24] are shifted out first, followed by bits [23:16], then bits [15:8] and finally bits [7:0].

A.7.7 Serial Interface Read Mode (RDMODE)

Instance Name
RDMODE

Offset
0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																RMODE															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 239: Serial Interface Read Mode (RDMODE)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	RMODE	R/W	0x0	Serial Interface Read Mode. After Addr (R14h) is shifted out, the content of this register is shifted out to the serial interface. When RM_CNT (R1Ch [9:8]) = 0, the content of this register is not shifted out to the serial interface. When RM_CNT (R1Ch [9:8]) = 1, bits [7:0] are shifted out. When RM_CNT (R1Ch [9:8]) = 2, bits [15:8] are shifted out first, followed by bits [7:0].

A.7.8 Serial Interface Header Count (HDRCNT)

Instance Name Offset
HDRCNT 0x1C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																		DUMMY_CNT		Reserved		RM_CNT		Reserved		ADDR_CNT		Reserved		INSTR_CNT	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	0	0	?	0	0	0	?	?	0	0

Table 240: Serial Interface Header Count (HDRCNT)

Bits	Name	Type	Reset	Description
31:14	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
13:12	DUMMY_CNT	R/W	0x0	Dummy Count. Number of bytes to shift out to the serial interface after the content of RdMode (R18h) register is shifted out. Note: The value being shifted out is 0. 0x0: 0 byte. 0x1: 1 byte. 0x2: 2 bytes. 0x3: 3 bytes.
11:10	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
9:8	RM_CNT	R/W	0x0	Read Mode Count. Number of bytes in RdMode (R18h) register to shift out to the serial interface. 0x0: 0 byte. 0x1: 1 byte. 0x2: 2 bytes. 0x3: Reserved.
7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
6:4	ADDR_CNT	R/W	0x0	Address Count. Number of bytes in Addr (R14h) register to shift out to the serial interface. 0x0: 0 byte. 0x1: 1 byte. 0x2: 2 bytes. 0x3: 3 bytes. 0x4: 4 bytes. others: Reserved
3:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 240: Serial Interface Header Count (HDCNT)

Bits	Name	Type	Reset	Description
1:0	INSTR_CNT	R/W	0x0	Instruction Count. Number of bytes in Instr (R10h) register to shift out to the serial interface. 0x0: 0 byte. 0x1: 1 byte. 0x2: 2 bytes. 0x3: Reserved.

A.7.9 Serial Interface Data In Count (DINCNT)

Instance Name

DINCNT

Offset

0x20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												DATA_IN_CNT																			
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 241: Serial Interface Data In Count (DINCNT)

Bits	Name	Type	Reset	Description
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:0	DATA_IN_CNT	R/W	0x0	Serial Interface Data In Count. For read transfers, RW_EN (R04h [13]) = 0, this register indicates the number of bytes of data to shift in from the serial interface and stores in the 8X32 bit Read FIFO. When this register is set to 0, data is shifted in continuously until XFER_STOP (R04h [14]) bit is set to 1 by firmware.

A.7.10 Serial Interface Timing (TIMING)

Instance Name

TIMING

Offset

0x24

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																						CLK_OUT_DLY	Reserved	CLK_CAPT_EDGE	CLK_IN_DLY		Reserved	DATA_IN_DLY			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	?	0	0	1	?	?	0	1

Table 242: Serial Interface Timing (TIMING)

Bits	Name	Type	Reset	Description
31:10	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
9:8	CLK_OUT_DLY	R/W	0x1	Serial Interface Clock Out Delay. Programmable delay added to the serial interface output clock to adjust the timing. 0x0: No delay. 0x1: Add 0.2 ns delay. 0x2: Add 0.4 ns delay. 0x3: Add 0.6 ns delay.
7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
6	CLK_CAPT_EDGE	R/W	0x0	Serial Interface Capture Clock Edge. Capture serial interface input data on either the rising or falling edge of the serial interface clock. This bit is used to allow more time to capture the input data. When CLK_POL (R04h [8]) = 0 and CLK_PHA (R04h [7]) = 1, capture input data on the falling edge of the serial interface clock. When CLK_POL (R04h [8]) = 1 and CLK_PHA (R04h [7]) = 0, capture input data on the falling edge of the serial interface clock. When CLK_POL (R04h [8]) = 1 and CLK_PHA (R04h [7]) = 1, capture input data on the rising edge of the serial interface clock. When CLK_POL (R04h [8]) = 1 and CLK_PHA (R04h [7]) = 0, capture input data on the rising edge of the serial interface clock. Note: When CLK_PHA (R04h [7]) = 1, this bit must be set to 0. 0x0: When CLK_POL (R04h [8]) = 0 and CLK_PHA (R04h [7]) = 0, capture input data on rising edge of the serial interface clock. 0x1: When CLK_POL (R04h [8]) = 0 and CLK_PHA (R04h [7]) = 0, capture input data on the falling edge of the serial interface clock.
5:4	CLK_IN_DLY	R/W	0x1	Serial Interface Clock In Delay. Programmable delay added to the serial interface input data capture clock. 0x0: No delay. 0x1: Add 0.2 ns delay. 0x2: Add 0.4 ns delay 0x3: Add 0.6 ns delay
3:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 242: Serial Interface Timing (TIMING)

Bits	Name	Type	Reset	Description
1:0	DATA_IN_DLY	R/W	0x1	Serial Interface Data Input Delay Delay the transfer of captured serial input data to the read register. 0x0: No delay. 0x1: 1 serial interface clock delay. 0x2: 2 serial interface clock delay. 0x3: Reserved

A.7.11 Serial Interface Configuration 2 (CONF2)

Instance Name
CONF2

Offset
0x28

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Field	Reserved																		DMA_WR_BURST		Reserved		DMA_RD_BURST		Reserved				DMA_WR_EN		DMA_RD_EN		SRST	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	?	?	0	1	?	?	?	?	?	0	0	0		

Table 243: Serial Interface Configuration 2 (CONF2)

Bits	Name	Type	Reset	Description
31:14	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
13:12	DMA_WR_BURST	R/W	0x1	DMA Write Burst. Number of data, each of width BYTE_LEN (R04h [5]), which can be written to the Write FIFO, without overflowing the FIFO, before a transmit burst request is made to the DMA controller. 0x0: 1 data. 0x1: 4 data. 0x2: 8 data. 0x3: Reserved.
11:10	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
9:8	DMA_RD_BURST	R/W	0x1	DMA Read Burst. Number of data, each of width BYTE_LEN (R04[5]), which is available in the Read FIFO before a receive burst request is made to the DMA controller. 0x0: 1 data. 0x1: 4 data. 0x2: 8 data. 0x3: Reserved.

Table 243: Serial Interface Configuration 2 (CONF2)

Bits	Name	Type	Reset	Description
7:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2	DMA_WR_EN	R/W	0x0	DMA Write Enable. 0x0: DMA write is disabled. 0x1: DMA write is enabled.
1	DMA_RD_EN	R/W	0x0	DMA Read Enable. 0x0: DMA read is disabled. 0x1: DMA read is enabled.
0	SRST	R/W	0x0	Soft Reset. Allows firmware to reset the hardware. Note: After setting this bit to 1, firmware has to reset this bit to 0 before starting any transfer 0x0: Hardware is not in reset. 0x1: Hardware is in reset.

A.7.12 Serial Interface Interrupt Status (ISR)

Instance Name

Offset

ISR

0x2C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																				WFIFO_OVRFLW_IS	WFIFO_UNDRFLW_IS	RFIFO_OVRFLW_IS	RFIFO_UNDRFLW_IS	WFIFO_FULL_IS	WFIFO_EMPTY_IS	RFIFO_FULL_IS	RFIFO_EMPTY_IS	WFIFO_DMA_BURST_IS	RFIFO_DMA_BURST_IS	XFER_RDY_IS	XFER_DONE_IS
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0

Table 244: Serial Interface Interrupt Status (ISR)

Bits	Name	Type	Reset	Description
31:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	WFIFO_OVRFLW_IS	R	0x0	Write FIFO Overflow Interrupt Status. 0x0: Write FIFO is not overflowed after masking. 0x1: Write FIFO is overflowed after masking.
10	WFIFO_UNDRFLW_IS	R	0x0	Write FIFO Underflow Interrupt Status. 0x0: Write FIFO is not underflowed after masking. 0x1: Write FIFO is underflowed after masking.

Table 244: Serial Interface Interrupt Status (ISR)

Bits	Name	Type	Reset	Description
9	RFIFO_OVRFLW_IS	R	0x0	Read FIFO Overflow Interrupt Status. 0x0: Read FIFO is not overflowed after masking. 0x1: Read FIFO is overflowed after masking.
8	RFIFO_UNDRFLW_IS	R	0x0	Read FIFO Underflow Interrupt Status. 0x0: Read FIFO is not underflowed after masking. 0x1: Read FIFO is underflowed after masking.
7	WFIFO_FULL_IS	R	0x0	Write FIFO Full Interrupt Status. 0x0: Write FIFO is not filled after masking. 0x1: Write FIFO is filled after masking.
6	WFIFO_EMPTY_IS	R	0x0	Write FIFO Empty Interrupt Status. 0x0: Write FIFO is not emptied after masking. 0x1: Write FIFO is emptied after masking.
5	RFIFO_FULL_IS	R	0x0	Read FIFO Full Interrupt Status. 0x0: Read FIFO is not filled after masking. 0x1: Read FIFO is filled after masking.
4	RFIFO_EMPTY_IS	R	0x0	Read FIFO Empty Interrupt Status. 0x0: Read FIFO is not emptied after masking. 0x1: Read FIFO is empty after masking.
3	WFIFO_DMA_BURST_IS	R	0x0	Write FIFO DMA burst Interrupt Status. 0x0: Number of unused entries in the Write FIFO is less than DMA_WR_BURST (R28h [13:12]) after masking. 0x1: Number of unused entries in the Write FIFO is greater than or equal to DMA_WR_BURST (R28h [13:12]) after masking.
2	RFIFO_DMA_BURST_IS	R	0x0	Read FIFO DMA Burst Interrupt Status. 0x0: Number of available data in the Read FIFO is less than DMA_RD_BURST (R28h [9:8]) after masking. 0x1: Number of available data in the Read FIFO is greater than or equal to DMA_RD_BURST (R28h [9:8]) after masking.
1	XFER_RDY_IS	R	0x0	Serial Interface Transfer Ready Interrupt Status 0x0: Serial interface is currently transferring data after masking. 0x1: Serial interface is ready for a new transfer after masking.
0	XFER_DONE_IS	R	0x0	Transfer Done Interrupt Status. 0x0: Transfer has not completed after masking. 0x1: Transfer has completed after masking.

A.7.13 Serial Interface Interrupt Mask (IMR)

Instance Name

IMR

Offset

0x30

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																				WFIFO_OVRFLW_IM	WFIFO_UNDRFLW_IM	RFIFO_OVRFLW_IM	RFIFO_UNDRFLW_IM	WFIFO_FULL_IM	WFIFO_EMPTY_IM	RFIFO_FULL_IM	RFIFO_EMPTY_IM	WFIFO_DMA_BURST_IM	RFIFO_DMA_BURST_IM	XFER_RDY_IM	XFER_DONE_IM
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	1	1	1	1	1	1	1	1	1	1

Table 245: Serial Interface Interrupt Mask (IMR)

Bits	Name	Type	Reset	Description
31:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	WFIFO_OVRFLW_IM	R	0x1	Write FIFO Overflow Interrupt Mask. 0x0: Write FIFO overflow interrupt is not masked. 0x1: Write FIFO overflow interrupt is masked.
10	WFIFO_UNDRFLW_IM	R	0x1	Write FIFO Underflow Interrupt Mask. 0x0: Write FIFO underflow interrupt is not masked. 0x1: Write FIFO underflow interrupt is masked.
9	RFIFO_OVRFLW_IM	R	0x1	Read FIFO Overflow Interrupt Mask. 0x0: Read FIFO overflow interrupt is not masked. 0x1: Read FIFO overflow interrupt is masked.
8	RFIFO_UNDRFLW_IM	R	0x1	Read FIFO Underflow Interrupt Mask. 0x0: Read FIFO underflow interrupt is not masked. 0x1: Read FIFO underflow interrupt is masked.
7	WFIFO_FULL_IM	R	0x1	Write FIFO Full Interrupt Mask. 0x0: Write FIFO full interrupt is not masked. 0x1: Write FIFO full interrupt is masked.
6	WFIFO_EMPTY_IM	R	0x1	Write FIFO Empty Interrupt Mask. 0x0: Write FIFO empty interrupt is not masked. 0x1: Write FIFO empty interrupt is masked.
5	RFIFO_FULL_IM	R	0x1	Read FIFO Full Interrupt Mask. 0x0: Read FIFO full interrupt is not masked. 0x1: Read FIFO full interrupt is masked.
4	RFIFO_EMPTY_IM	R	0x1	Read FIFO Empty Interrupt Mask. 0x0: Read FIFO empty interrupt is not masked. 0x1: Read FIFO empty interrupt is masked.

Table 245: Serial Interface Interrupt Mask (IMR)

Bits	Name	Type	Reset	Description
3	WFIFO_DMA_BURST_IM	R	0x1	Write FIFO DMA Burst Interrupt Mask. 0x0: Write FIFO DMA burst interrupt is not masked. 0x1: Write FIFO DMA burst interrupt is masked.
2	RFIFO_DMA_BURST_IM	R	0x1	Read FIFO DMA Burst Interrupt Mask. 0x0: Read FIFO DMA burst interrupt is not masked. 0x1: Read FIFO DMA burst interrupt is masked.
1	XFER_RDY_IM	R	0x1	Serial Interface Transfer Ready Mask. 0x0: Transfer ready interrupt is not masked. 0x1: Transfer ready interrupt is masked.
0	XFER_DONE_IM	R	0x1	Transfer Done Interrupt Mask. 0x0: Transfer done interrupt is not masked. 0x1: Transfer done interrupt is masked.

A.7.14 Serial Interface Interrupt Raw Status (IRSR)

Instance Name

Offset

IRSR

0x34

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																				WFIFO_OVRFLW_IR	WFIFO_UNDRFLW_IR	RFIFO_OVRFLW_IR	RFIFO_UNDRFLW_IR	WFIFO_FULL_IR	WFIFO_EMPTY_IR	RFIFO_FULL_IR	RFIFO_EMPTY_IR	WFIFO_DMA_BURST_IR	RFIFO_DMA_BURST_IR	XFER_RDY_IR	XFER_DONE_IR
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	1	0	1	1	0	1	1

Table 246: Serial Interface Interrupt Raw Status (IRSR)

Bits	Name	Type	Reset	Description
31:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	WFIFO_OVRFLW_IR	R	0x0	Write FIFO Overflow Interrupt Raw. 0x0: Write FIFO is not overflowed before masking. 0x1: Write FIFO is overflowed before masking.
10	WFIFO_UNDRFLW_IR	R	0x0	Write FIFO Underflow Interrupt Raw. 0x0: Write FIFO is not underflowed before masking. 0x1: Write FIFO is underflowed before masking.
9	RFIFO_OVRFLW_IR	R	0x0	Read FIFO Overflow Interrupt Raw. 0x0: Read FIFO is not overflowed before masking. 0x1: Read FIFO is overflowed before masking.

Table 246: Serial Interface Interrupt Raw Status (IRSR)

Bits	Name	Type	Reset	Description
8	RFIFO_UNDRFLW_IR	R	0x0	Read FIFO Underflow Interrupt Raw. 0x0: Read FIFO is not underflowed before masking. 0x1: Read FIFO is underflowed before masking.
7	WFIFO_FULL_IR	R	0x0	Write FIFO Full Interrupt Raw. 0x0: Write FIFO is not filled before masking. 0x1: Write FIFO is filled before masking.
6	WFIFO_EMPTY_IR	R	0x1	Write FIFO Empty Interrupt Raw. 0x0: Write FIFO is not emptied before masking. 0x1: Write FIFO is emptied before masking.
5	RFIFO_FULL_IR	R	0x0	Read FIFO Full Interrupt Raw. 0x0: Read FIFO is not filled before masking. 0x1: Read FIFO is filled before masking.
4	RFIFO_EMPTY_IR	R	0x1	Read FIFO Empty Interrupt Raw. 0x0: Read FIFO is not emptied before masking. 0x1: Read FIFO is empty before masking.
3	WFIFO_DMA_BURST_IR	R	0x1	Write FIFO DMA Burst Interrupt Raw. 0x0: Number of unused entries in the Write FIFO is less than DMA_WR_BURST (R28h [13:12]) before masking. 0x1: Number of unused entries in the Write FIFO is greater than or equal to DMA_WR_BURST (R28h [13:12]) before masking.
2	RFIFO_DMA_BURST_IR	R	0x0	Read FIFO DMA Burst Interrupt Raw. 0x0: Number of available data in the Read FIFO is less than DMA_RD_BURST (R28h [9:8]) before masking. 0x1: Number of available data in the Read FIFO is greater than or equal to DMA_RD_BURST (R28h [9:8]) before masking.
1	XFER_RDY_IR	R	0x1	Serial Interface Transfer Ready Raw. 0x0: Serial interface is currently transferring data before masking. 0x1: Serial interface is ready for a new transfer before masking.
0	XFER_DONE_IR	R	0x1	Transfer Done Interrupt Raw. 0x0: Transfer has not completed before masking. 0x1: Transfer has completed before masking.

A.7.15 Serial Interface Interrupt Clear (ISC)

Instance Name
ISC

Offset
0x38

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														XFER_DONE_IC	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 247: Serial Interface Interrupt Clear (ISC)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	XFER_DONE_IC	W	0x0	Transfer Done Interrupt Clear. 0x0: Transfer done interrupt is not cleared. 0x1: Transfer done interrupt is cleared.

A.8 SSP Register Information

Table 248: SSP Register Summary

Offset	Name	Description	Details
0x00	SSCR0	SSP Control Register 0	Page: 186
0x04	SSCR1	SSP Control Register 1	Page: 188
0x08	SSSR	SSP Status Register	Page: 191
0x0C	SSITR	SSP Interrupt Test Register	Page: 194
0x10	SSDR	SSP Data Register	Page: 194
0x28	SSTO	SSP Time Out Register	Page: 195
0x2C	SSPSP	SSP Programmable Serial Protocol Register	Page: 196
0x30	SSTSA	SSP TX Time Slot Active Register	Page: 198
0x34	SSRSA	SSP RX Time Slot Active Register	Page: 198
0x38	SSTSS	SSP Time Slot Status Register	Page: 199
0x3C	RESERVED	Reserved	Page: 200
0x40	RESERVED	Reserved	Page: 200

A.8.1 SSP Control Register 0 (SSP_SSCR0)

The SSP Control 0 registers contain 13 different bit fields that control various functions within the SSP port. The following table shows the bit locations corresponding to the different control bit fields within the SSP Control 0 Register. The reset state of all bits are as shown, but they must be programmed to their preferred values before enabling the SSP port.

Writes to reserved bits must be 0b0, and read values of reserved bits are undefined.

Instance Name													Offset																			
SSCR0													0x00																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	MOD	Reserved	FPCKE	Reserved	Reserved	FRDC			TIM	RIM	Reserved	EDSS	Reserved										SSE	Reserved	FRF		DSS					
Default	0	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 249: SSP Control Register 0 (SSP_SSCR0)

Bits	Name	Type	Reset	Description
31	MOD	R/W	0x0	Mode 0x0: Normal SSP mode 0x1: Network mode
30	Reserved	R/W	0x0	Reserved. Do not change the reset value.

Table 249: SSP Control Register 0 (SSP_SSCR0)

Bits	Name	Type	Reset	Description
29	FPCKE	R/W	0x0	FIFO Packing Enable 0x0: FIFO packing mode disabled 0x1: FIFO packing mode enabled
28	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
27	Reserved	R/W	0x0	Reserved. Do not change the reset value.
26:24	FRDC	R/W	0x0	Frame Rate Divider Control Value of 0x0-0x7 specifies the number of time slots per frame when in network mode (the actual number of time slots is this field +1, so 1 to 8 time slots can be specified).
23	TIM	R/W	0x0	Transmit FIFO Underrun Interrupt Mask 0x0: TUR events generate an SSP interrupt 0x1: TUR events do NOT generate an SSP interrupt
22	RIM	R/W	0x0	Receive FIFO Overrun Interrupt Mask 0x0: ROR events generate an SSP interrupt 0x1: ROR events do NOT generate an SSP interrupt
21	Reserved	R/W	0x0	Reserved. Do not change the reset value.
20	EDSS	R/W	0x0	Extended Data Size Select 0x0: A 0 is pre-appended to the DSS value to set the DSS range from 8 to 16 bits. 0x1: A 1 is pre-appended to the DSS value to set the DSS range from 18 to 32 bits.
19:8	Reserved	R/W	0x0	Reserved. Do not change the reset value.
7	SSE	R/W	0x0	Synchronous Serial Port Enable 0x0: SSPx port is disabled 0x1: SSPx port is enabled
6	Reserved	R/W	0x0	Reserved. Do not change the reset value.
5:4	FRF	R/W	0x0	Frame Format This field must be written with 0x3 = to select the PSP format. 0x0: Motorola* Serial Peripheral Interface (SPI) 0x1: Texas Instruments* Synchronous Serial Protocol (SSP) 0x2: Reserved, undefined 0x3: Programmable Serial Protocol (PSP)

Table 249: SSP Control Register 0 (SSP_SSCR0)

Bits	Name	Type	Reset	Description
3:0	DSS	R/W	0x0	Data Size Select EDSS DSS Data Size 1 0000 reserved, undefined 1 0001 18-bit data 1 0010 reserved, undefined 1 0011 reserved, undefined 1 0100 reserved, undefined 1 0101 reserved, undefined 1 0110 reserved, undefined 1 0111 reserved, undefined 1 1000 reserved, undefined 1 1001 reserved, undefined 1 1010 reserved, undefined 1 1011 reserved, undefined 1 1100 reserved, undefined 1 1101 reserved, undefined 1 1110 reserved, undefined 1 1111 32-bit data EDSS DSS Data Size 0 0000 reserved, undefined 0 0001 reserved, undefined 0 0010 reserved, undefined 0 0011 reserved, undefined 0 0100 reserved, undefined 0 0101 reserved, undefined 0 0110 reserved, undefined 0 0111 8-bit data 0 1000 reserved, undefined 0 1001 reserved, undefined 0 1010 reserved, undefined 0 1011 reserved, undefined 0 1100 reserved, undefined 0 1101 reserved, undefined 0 1110 reserved, undefined 0 1111 16-bit data

A.8.2 SSP Control Register 1 (SSP_SSCR1)

The SSP Port Control 1 registers contain bit fields that control various SSP port functions. The following table shows bit locations corresponding to control bit fields in SSP_SSCR1. The reset state of all bits is shown, but must be set to the preferred value before enabling the SSP port by setting the <Synchronous Serial Port Enable> field in the SSP Control Register 0.

Write 0b0 to reserved bits, the read values of reserved bits are undetermined.

Instance Name
SSCR1

Offset
0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TTELP	TTE	EBCEI	SCFR	Reserved		SCLKDIR	SFRMDIR	RWOT	TRAIL	TSRE	RSRE	TINTE	Reserved	Reserved	IFS	STRF	EFWR	RFT				TFT				Reserved	SPH	SPO	LBM	TIE	RIE
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	0	0	0	0	0	0	0	0	0	0	0	?	0	0	0	0	0

Table 250: SSP Control Register 1 (SSP_SSCR1)

Bits	Name	Type	Reset	Description
31	TTELP	R/W	0x0	TXD Tri-stated Enable On Last Phase 0x0: TXDx is tri-stated 1/2 clock cycle after the beginning of the LSB. 0x1: TXDx output signal is tri-stated on the clock edge that ends the LSB.
30	TTE	R/W	0x0	TXD Tri-States Enable 0x0: TXDx output signal is not tri-stated. 0x1: TXD is tri-stated when not transmitting data.
29	EBCEI	R/W	0x0	Enable Bit Count Error Interrupt 0x0: Interrupt due to a bit count error is disabled. 0x1: Interrupt due to a bit count error is enabled.
28	SCFR	R/W	0x0	Slave Clock Free Running 0x0: Clock input to SSPSCLKx is not active when SSCR1_SCLKDIR is set to 1 0x1: Clock input to SSPSCLKx is active when SSCR1_SCLKDIR is set to 1
27:26	Reserved	R/W	0x0	Reserved. Do not change the reset value.
25	SCLKDIR	R/W	0x0	SSP Serial Bit Rate Clock (SSPSCLKx) Direction 0x0: Master mode, SSPx port drives SSPSCLKx 0x1: Slave mode, SSPx port receives SSPSCLKx
24	SFRMDIR	R/W	0x0	SSP Frame (SSPSFRMx) Direction 0x0: Master mode, SSPx port drives SSPSFRMx 0x1: Slave mode, SSPx port receives SSPSFRMx
23	RWOT	R/W	0x0	Receive Without Transmit 0x0: Transmit/Receive mode 0x1: Receive without Transmit mode
22	TRAIL	R/W	0x0	Trailing Byte 0x0: Trailing bytes are handled by CPU 0x1: Reserved, undefined
21	TSRE	R/W	0x0	Transmit Service Request Enable 0x0: DMA service request is disabled 0x1: DMA service request is enabled

Table 250: SSP Control Register 1 (SSP_SSCR1)

Bits	Name	Type	Reset	Description
20	RSRE	R/W	0x0	Receive Service Request Enable 0x0: DMA service request is disabled 0x1: DMA service request is enabled
19	TINTE	R/W	0x0	Receiver Time-out Interrupt Enable 0x0: Receiver time-out interrupt is disabled. 0x1: Receiver time-out interrupt is enabled.
18	Reserved	R/W	0x0	Reserved. Do not change the reset value.
17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16	IFS	R/W	0x0	Invert Frame Signal 0x0: SSPSFRMx polarity is determined by the PSP polarity bits. 0x1: SSPSFRMx is inverted from normal-SSPSFRMx (as defined by the PSP polarity bits). (Works in all frame formats: SPI, SSP, and PSP)
15	STRF	R/W	0x0	Select FIFO For Efwr Select FIFO For Efwr (Test Mode Bit). Only when EFWR = 1 0x0: TXFIFO is selected for both writes and reads through the SSP Data Register. 0x1: RXFIFO is selected for both writes and reads through the SSP Data Register.
14	EFWR	R/W	0x0	Enable FIFO Write/read Enable FIFO Write/read (Test Mode Bit) 0x0: FIFO write/read special function is disabled (normal SSPx operational mode). 0x1: FIFO write/read special function is enabled.
13:10	RFT	R/W	0x0	RXFIFO Trigger Threshold Sets threshold level at which RXFIFO asserts interrupt. Level should be set to the preferred threshold value minus 1.
9:6	TFT	R/W	0x0	TXFIFO Trigger Threshold Sets threshold level at which TXFIFO asserts interrupt. Level should be set to the preferred threshold value minus 1.
5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	SPH	R/W	0x0	Motorola SPI SSPSCLK phase setting 0x0: SSPSCLKx is inactive until one cycle after the start of a frame and active until 1/2 cycle before the end of a frame. 0x1: SSPSCLKx is inactive until 1/2 cycle after the start of a frame and active until one cycle
3	SPO	R/W	0x0	Motorola SPI SSPSCLK Polarity Setting 0x0: The inactive or idle state of SSPSCLKx is low. 0x1: The inactive or idle state of SSPSCLKx is high.

Table 250: SSP Control Register 1 (SSP_SSCR1)

Bits	Name	Type	Reset	Description
2	LBM	R/W	0x0	Loopback Mode Loopback Mode (Test Mode Bit) 0x0: Normal serial port operation is enabled. 0x1: Output of TX serial shifter is internally connected to input of RX serial shifter.
1	TIE	R/W	0x0	Transmit FIFO Interrupt Enable 0x0: TXFIFO threshold-level-reached interrupt is disabled. 0x1: TXFIFO threshold-level-reached interrupt is enabled.
0	RIE	R/W	0x0	Receive FIFO Interrupt Enable 0x0: RXFIFO threshold-level-reached interrupt is disabled. 0x1: RXFIFO threshold-level-reached interrupt is enabled.

A.8.3 SSP Status Register (SSP_SSSR)

The SSP Port Status registers contain bits that signal overrun errors as well as the TXFIFO and RXFIFO service requests. Each of these hardware-detected events signals an interrupt request to the interrupt controller, or a DMA request. The Status register also contains flags that indicate if the SSPx port is actively transmitting data, if the TXFIFO is not full, and if the RXFIFO is not empty. A signal- interrupt signal is sent to the interrupt controller for each SSPx port. These events can cause an interrupt request or a DMA request: end-of-chain, receiver time out, peripheral trailing byte, RXFIFO overrun, RXFIFO service request, and TXFIFO service request.

Bits that cause an interrupt request remain set until they are cleared by writing a 0b1 to each bit. Once a status bit is cleared, the interrupt is cleared. Read-write bits are called status bits (status bits are referred to as 'sticky'; and once set by hardware, they can only be cleared by writing a 0b1 to each bit), read-only bits are called flags. Writing a 0b1 to a sticky status bit clears it, writing a 0b0 has no effect. Read-only flags are set to 0b1 and are cleared automatically to 0b0 by hardware, and writes have no effect. Some bits that cause interrupt requests have corresponding mask bits in the Control registers and are indicated in the section headings that follow.

The following table shows the bit locations corresponding to the status and flag bits within the SSP Port Status Register. All bits are read-only except the <Receive FIFO Overrun>, <Peripheral Trailing Byte Interrupt>, <Transmit FIFO Underrun>, <Bit Count Error>, <Receiver Time-out Interrupt>, and <End of Chain>, which are all read-write. The reset state of read-write bits is 0b0 and all bits return to their reset state when <Synchronous Serial Port Enable> field in the SSP Control Register 0 is cleared.

Write 0b0 to reserved bits, reads from reserved bits are undetermined.

Instance Name
SSSR

Offset
0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	OSS	TX_OSS	Reserved						BCE	CSS	TUR	Reserved	TINT	Reserved	Reserved	RFL				TFL				ROR	RFS	TFS	BSY	RNE	TNF	Reserved		
Default	0	0	?	?	?	?	?	?	0	0	0	0	0	0	?	?	1	1	1	1	0	0	0	0	0	0	0	0	0	1	?	?

Table 251: SSP Status Register (SSP_SSSR)

Bits	Name	Type	Reset	Description
31	OSS	R	0x0	Odd Sample Status Note: that this bit needs to be looked at only when FIFO Packing is enabled (<FIFO Packing Enable> field in SSP Control Register 0 is set). Otherwise this bit is zero. When SSPx port is in Packed mode, and the CPU is used instead of DMA to read the RxFIFO, CPU should make sure that <Receive FIFO Not Empty>=1 AND this field=0 before it attempts to read the RxFIFO. 0x0: RxFIFO entry has 2 samples 0x1: RxFIFO entry has 1 sample.
30	TX_OSS	R	0x0	TX FIFO Odd Sample Status When SSPx port is in packed mode, the number of samples in the TX FIFO is: (<Transmit FIFO Level>*2 + this field), when <Transmit FIFO Not Full>=1 32, when <Transmit FIFO Not Full>=0. The TX FIFO cannot accept new data when <Transmit FIFO Not Full>=1 and <Transmit FIFO Level>=15 and this field=1. (The TX FIFO has 31 samples). Note: that this bit needs to be read only when FIFO Packing is enabled (<FIFO Packing Enable> in the SSP Control Register 0 set). Otherwise this bit is zero. 0x0: TxFIFO entry has an even number of samples 0x1: TxFIFO entry has an odd number of sample.
29:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	BCE	R/W	0x0	Bit Count Error 0x0: The SSPx port has not experienced a bit count error. 0x1: The SSPSFRMx signal was asserted when the bit counter was not zero.
22	CSS	R	0x0	Clock Synchronization Status 0x0: The SSPx port is ready for slave clock operations. 0x1: The SSPx port is currently busy synchronizing slave mode signals.
21	TUR	R/W	0x0	Transmit FIFO Underrun 0x0: The TXFIFO has not experienced an underrun. 0x1: A read from the TXFIFO was attempted when the TXFIFO was empty, causes an interrupt if it is enabled (<Transmit FIFO Underrun Interrupt Mask> in the SSP Control Register 0 is 0).
20	Reserved	R/W	0x0	Reserved. Do not change the reset value.

Table 251: SSP Status Register (SSP_SSSR)

Bits	Name	Type	Reset	Description
19	TINT	R/W	0x0	Receiver Time-out Interrupt 0x0: No receiver time-out is pending. 0x1: Receiver time-out pending, causes an interrupt request.
18	Reserved	R/W	0x0	Reserved. Do not change the reset value.
17:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:12	RFL	R	0xF	Receive FIFO Level Number of entries minus one in RXFIFO. Note: When the value 0xF is read, the RXFIFO is either empty or full, and software should read the <Receive FIFO Not Empty> field.
11:8	TFL	R	0x0	Transmit FIFO Level Number of entries in TXFIFO. Note: When the value 0x0 is read, the TXFIFO is either empty or full, and software should read the <Transmit FIFO Not Full> field.
7	ROR	R/W	0x0	Receive FIFO Overrun 0x0: RXFIFO has not experienced an overrun. 0x1: Attempted data write to full RXFIFO, causes an interrupt request.
6	RFS	R	0x0	Receive FIFO Service Request 0x0: RXFIFO level is at or below RFT threshold (RFT), or SSPx port is disabled 0x1: RXFIFO level exceeds RFT threshold (RFT), causes an interrupt request
5	TFS	R	0x0	Transmit FIFO Service Request 0x0: TX FIFO level exceeds the TFT threshold (TFT + 1), or SSPx port disabled. 0x1: TXFIFO level is at or below TFT threshold (TFT + 1), causes an interrupt request.
4	BSY	R	0x0	SSP Busy 0x0: SSPx port is idle or disabled. 0x1: SSPx port is currently transmitting or receiving framed data.
3	RNE	R	0x0	Receive FIFO Not Empty 0x0: RXFIFO is empty. 0x1: RXFIFO is not empty.
2	TNF	R	0x1	Transmit FIFO Not Full 0x0: TXFIFO is full. 0x1: TXFIFO is not full.
1:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.8.4 SSP Interrupt Test Register (SSP_SSITR)

Setting the <Test TXFIFO Service Request> field generates a non-maskable interrupt request and a DMA service request for the TXFIFO.

Write 0b0 to reserved bits, reads from reserved bits are undetermined.

Instance Name

SSITR

Offset

0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							TROR	TRFS	TIFS	Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?

Table 252: SSP Interrupt Test Register (SSP_SSITR)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	TROR	R/W	0x0	Test RXFIFO Overrun 0x0: No RXFIFO-overflow service request. 0x1: Generates a non-maskable RXFIFO-overflow interrupt request. No DMA request is generated
6	TRFS	R/W	0x0	Test RXFIFO Service Request 0x0: No RXFIFO-service request. 0x1: Generates a non-maskable RXFIFO-service interrupt request and DMA request
5	TTFS	R/W	0x0	Test TXFIFO Service Request 0x0: No TXFIFO-service request 0x1: Generates a non-maskable TXFIFO-service interrupt request and DMA request
4:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.8.5 SSP Data Register (SSP_SSDR)

The SSP Data Registers are each two physical registers that have a common address. One SSDR_x is temporary storage for data that is transferred automatically into the TXFIFO, the other SSDR_x is temporary storage for data that is transferred automatically from the RXFIFO.

As programmed I/O or DMA access the SSDR_x, the TXFIFO or RXFIFO control logic transfers data automatically between the SSDR_x and the FIFO as fast as the system moves it. Data in the TXFIFO shifts up to accommodate new data that is written to the SSDR_X, unless it is an attempted write to a full TXFIFO. Data in the RXFIFO shifts down to accommodate data that is read from the SSP Data Register. The <Transmit FIFO Level>, <Receive FIFO Level>, <Receive FIFO Not Empty>, and <Transmit FIFO Not Full> fields in the SSP Status Register show whether the FIFO is full, above/below a programmable FIFO trigger threshold level, or empty.

When using programmed I/O, data can be written to the SSP Data Register anytime the TXFIFO falls below its trigger threshold level.

When a data sample size of less than 32-bits is selected, or 16 bits for packed mode, software should right-justify the data that is written to the SSP Data Register for automatic insertion into the TXFIFO. The transmit logic left-justifies the data and ignores any unused bits. Received data of less than 32 bits is right-justified automatically in the RXFIFO (thus, you cannot perform a write in packed mode of less than 32 bits wide). The TXFIFO and RXFIFO are cleared to 0b0 when the SSPx port is reset or disabled (by writing a 0b0 to the <Synchronous Serial Port Enable> field in the SSP Control Register 0).

The reset state of SSSDR_x is undetermined. The following table shows the location of the SSPx port SSSDR_x.

Instance Name

Offset

SSDR

0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DATA																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 253: SSP Data Register (SSP_SSDR)

Bits	Name	Type	Reset	Description
31:0	DATA	R/W	0x0	DATA Data to be written to the TXFIFO read from the RXFIFO

A.8.6 SSP Time Out Register (SSP_SSTO)

The SSPx Time Out registers specify the time-out (TIMEOUT) value used to signal a period of inactivity within the RXFIFO, see 'Timeout' section in the Aspen (88AP168) Processor Specification. When a timeout occurs, the <Receiver Time-out Interrupt> field in the SSP Status Register is set. When the TIMEOUT value is set to 0x000000, no timeout occurs and the <Receiver Time-out Interrupt> field is not set. The TIMEOUT interval is given by the calculation in the TIMEOUT Interval Equation.

Skipped paragraph type: EquationTitle Content: TIMEOUT Interval Equation

Skipped paragraph type: Formula Content: TimeOut Interval = <Timeout Value> / Low-Speed I/O Bus Clock Frequency



Note

Low-Speed I/O Bus Clock Frequency = 26 MHz

Write 0b0 to reserved bits, reads from reserved bits are undetermined.

Instance Name

Offset

SSTO

0x28

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								TIMEOUT																							
Default	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 254: SSP Time Out Register (SSP_SSTO)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23:0	TIMEOUT	R/W	0x0	Timeout Value TIMEOUT value is the value (0 to $2^{24}-1$) that defines the time-out interval. The time-out interval is given by the equation shown in the TIMEOUT Interval Equation.

A.8.7 SSP Programmable Serial Protocol Register (SSP_SSPSP)

The SSP Programmable Serial Protocol registers contain eight fields that program the various programmable serial-protocol (PSP) parameters. When using Programmable Serial Protocol (PSP) format in network mode, the parameters <Serial Frame Delay>, <Serial Frame Delay>, <Start Delay>, <Dummy Stop>, <Extended Dummy Stop>, <Dummy Start>, and <Extended Dummy Start> must be set to 0b0. The other parameters <Serial Frame Polarity>, <Serial Bit-rate Clock Mode>, <Frame Sync Relative Timing Bit>, and <Serial Frame Width> are programmable.

Writes 0b0 to reserved bits, reads from reserved bits are undetermined.

Instance Name
SSPSP

Offset
0x2C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved	EDMYSTOP			EDMYSTRT		FSRT	DMYSTOP		Reserved	SFRMWDTH					SFRMDLY						DMYSTRT		STRTDLY			ETDS	SFRMP	SCMODE			
Default	?	0	0	0	0	0	0	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 255: SSP Programmable Serial Protocol Register (SSP_SSPSP)

Bits	Name	Type	Reset	Description
31	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
30:28	EDMYSTOP	R/W	0x0	Extended Dummy Stop The most-significant bits of the dummy stop delay Note: Do not use in PSP Network mode.
27:26	EDMYSTRT	R/W	0x0	Extended Dummy Start The most-significant bits of the dummy start delay Note: Do not use in PSP Network mode.
25	FSRT	R/W	0x0	Frame Sync Relative Timing Bit 0x0: Next frame is asserted after the end of the DMTSTOP timing. 0x1: Next frame is asserted with the LSB of the previous frame.

Table 255: SSP Programmable Serial Protocol Register (SSP_SSPSP)

Bits	Name	Type	Reset	Description
24:23	DMYSTOP	R/W	0x0	Dummy Stop The least-significant bits of the dummy stop delay Programmed value of <Extended Dummy Stop> + this field specifies the number (0-31) of active clocks (SSPCLKx) that follow the end of the transmitted data. Note: Do not use in PSP Network mode.
22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21:16	SFRMWIDTH	R/W	0x0	Serial Frame Width Least-significant bits of the serial frame width Programmed value of this field specifies the frame width from 0x00 (one SSPCLKx cycle) to 0x3F (63 SSPCLKx cycles).
15:9	SFRMDLY	R/W	0x0	Serial Frame Delay Programmed value specifies the number (0 -127) of active one-half clocks (SSPCLKx) asserted from the most-significant bit of TXDx (output) or RXD (input) being driven to SSPSRMx. Note: Do not use in PSP Network mode.
8:7	DMYSTRT	R/W	0x0	Dummy Start Least-significant bits of the dummy start delay Programmed value of this field specifies the number (0-15) of active clocks (SSPCLKs) between the end of start delay and when the most-significant bit of transmit/receive data is driven Note: Do not use in PSP Network mode.
6:4	STRTDLY	R/W	0x0	Start Delay Programmed value specifies the number (0-7) of non-active clocks (SSPCLKx) that define the duration of idle time Note: Do not use in PSP Network mode.
3	ETDS	R/W	0x0	End Of Transfer Data State
2	SFRMP	R/W	0x0	Serial Frame Polarity 0x0: SSPSRMx is active low (0b0). 0x1: SSPSRMx is active high (0b1).
1:0	SCMODE	R/W	0x0	Serial Bit-rate Clock Mode 0x0: Data Driven (Falling), Data Sampled (Rising), Idle State (Low) 0x1: Data Driven (Rising), Data Sampled (Falling), Idle State (Low) 0x2: Data Driven (Rising), Data Sampled (Falling), Idle State (High) 0x3: Data Driven (Falling), Data Sampled (Rising), Idle State (High)

A.8.8 SSP TX Time Slot Active Register (SSP_SSTSA)

Only used in Network mode (<Mode> in SSP Control Register 0 set), the read-write SSP TX Time Slot Active registers specify in which time slot the SSPx port transmits data. See 'Network Mode (Example Using 4 Time Slots)' figure for an example of the use of time slots when in Network mode.

The eight-bit <TX Time Slot Active> field specifies in which time slots the SSPx port transmits data and in which time slots the SSPx port does not transmit data. Bits beyond the <Frame Rate Divider Control> field in the SSP Control Register 0 value are ignored (for example, if <Frame Rate Divider Control>= 0x3, specifying that four time slots are used, then <TX Time Slot Active> bits 7:4 are ignored). If the <TXD Tri-state Enable> field in the SSP Control Register 1 is set, the SSPx port tri-states the SSPTXDx interface output signal line during time slots that have associated TTSA bits programmed to 0b0.

Write 0b0 to reserved bits, reads from reserved bits are undetermined.

Instance Name
SSTSA

Offset
0x30

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							TTSA								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 256: SSP TX Time Slot Active Register (SSP_SSTSA)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	TTSA	R/W	0x0	TX Time Slot Active 0x0: SSPx port does NOT transmit data in this time slot. 0x1: SSPx port does transmit data in this time slot.

A.8.9 SSP RX Time Slot Active Register (SSP_SSRSA)

Only used in Network mode (<Mode> in SSP Control Register 0 set), the read-write SSP RX Time Slot Active registers specify in which time slot the SSPx port receives data. See 'Network Mode (Example Using 4 Time Slots)' figure for an example of the use of time slots when in Network mode.

The eight-bit <RX Time Slot Active> field specifies in which time slots the SSPx port receives data and in which time slots the SSPx port does not receive data. Bits beyond the <Frame Rate Divider Control> field in the SSP Control Register 0 value are ignored. For example, if <Frame Rate Divider Control>=0x3, specifying that four time slots are used, then <RX Time Slot Active> bits 7:4 are ignored.

Write 0b0 to reserved bits, reads from reserved bits are undetermined.

Instance Name
SSRSA

Offset
0x34

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							RTSA								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 257: SSP RX Time Slot Active Register (SSP_SSRSA)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	RTSA	R/W	0x0	RX Time Slot Active 0x0: SSPx port does not receive data in this time slot. 0x1: SSPx port receives data in this time slot.

A.8.10 SSP Time Slot Status Register (SSP_SSTSS)

The <Network Mode Busy> field shows when the SSPx port is within a frame only when in Network mode (<Mode> in SSP Control Register 0 set). It can be used by software to determine when a clean shutdown of the SSPx port can be initiated. Software should:

1. Determine that the TXFIFO is either empty or is emptied at the end of the next frame.

1. Deactivate the TXFIFO DMA service requests.
2. Clear <Mode> in SSP Control Register 0 (to exit network mode).
3. Then poll <Network Mode Busy> until it is cleared before disabling the SSPx port by clearing the <Synchronous Serial Port Enable> field in the SSP Control Register 0.

When the SSPx port is a master of the frame signal (<SSP Frame (SSPSFRMx) Direction> field in SSP Control Register 1 set), <Network Mode Busy> is set as long as the port remains in Network mode. When the SSPx port is a slave of the frame signal, the <Network Mode Busy> field is cleared if the current frame (number of bits per sample * number of time slots per frame) has not expired since the last SSPSFRMx interface signal (in/out) was asserted.

Time Slot Status (TSS)

The three-bit <Time Slot Status> field value identifies the time slot in which the SSPx port is operating. Due to synchronization between the SSPSCLKx domain and an internal bus clock domain, the TSS value becomes stable approximately two internal bus clock cycles after the beginning of the associated time slot. The <Time Slot Status> value is not valid if the <Network Mode Busy> field is cleared.

Write 0b0 to reserved bits, reads from reserved bits are undetermined.

Instance Name

Offset

SSTSS

0x38

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	NMBSY	Reserved																													TSS		
Default	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	

Table 258: SSP Time Slot Status Register (SSP_SSTSS)

Bits	Name	Type	Reset	Description
31	NMBSY	R	0x0	Network Mode Busy 0x0: SSPx port is in network mode and no frame is currently active. 0x1: SSPx port is in network mode and a frame is currently active.

Table 258: SSP Time Slot Status Register (SSP_SSTSS)

Bits	Name	Type	Reset	Description
30:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	TSS	R	0x0	Time Slot Status Value indicates which time slot is currently active. Because of synchronization between the SSPx port's SSPSCLKx domain and an internal bus clock domain, the value in this field becomes stable between the beginning and end of the currently active time slot.

A.8.11 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x3C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							Reserved								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 259: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.8.12 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x40

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved	Reserved														Reserved				Reserved												
Default		?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0

Table 260: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
30:16	Reserved	R/W	0x0	Reserved. Do not change the reset value.

Table 260: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
15:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11:0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.9 UART Register Information

Table 261: UART Register Summary

Offset	Name	Description	Details
0x00	RBR_DLL_THR	Receive Buffer Register	Page: 203
0x04	IER_DLH	Interrupt Enable Register	Page: 205
0x08	IIR_FCR	Interrupt Identification Register	Page: 206
0x0C	LCR	Line Control Register	Page: 208
0x10	MCR	Modem Control Register	Page: 210
0x14	LSR	Line Status Register	Page: 213
0x18	MSR	Modem Status Register	Page: 215
0x1C	SCR	Scratchpad Register	Page: 218
0x20	LPDLL	Low Power Divisor Latch Low Register	Page: 218
0x24	LPDLH	Low Power Divisor Latch High Register	Page: 219
0x30	SRBR_STHR0	Shadow Receive Buffer Register 0	Page: 220
0x34	SRBR_STHR1	Shadow Receive Buffer Register 1	Page: 221
0x38	SRBR_STHR2	Shadow Receive Buffer Register 2	Page: 222
0x3C	SRBR_STHR3	Shadow Receive Buffer Register 3	Page: 222
0x40	SRBR_STHR4	Shadow Receive Buffer Register 4	Page: 223
0x44	SRBR_STHR5	Shadow Receive Buffer Register 5	Page: 223
0x48	SRBR_STHR6	Shadow Receive Buffer Register 6	Page: 223
0x4C	SRBR_STHR7	Shadow Receive Buffer Register 7	Page: 224
0x50	SRBR_STHR8	Shadow Receive Buffer Register 8	Page: 224
0x54	SRBR_STHR9	Shadow Receive Buffer Register 9	Page: 225
0x58	SRBR_STHR10	Shadow Receive Buffer Register 10	Page: 225
0x5C	SRBR_STHR11	Shadow Receive Buffer Register 11	Page: 225
0x60	SRBR_STHR12	Shadow Receive Buffer Register 12	Page: 226
0x64	SRBR_STHR13	Shadow Receive Buffer Register 13	Page: 226
0x68	SRBR_STHR14	Shadow Receive Buffer Register 14	Page: 227
0x6C	SRBR_STHR15	Shadow Receive Buffer Register 15	Page: 227
0x70	FAR	FIFO Access Register	Page: 227
0x74	TFR	Transmit FIFO Read Register	Page: 228

Table 261: UART Register Summary

Offset	Name	Description	Details
0x78	RFW	Receive FIFO Write Register	Page: 229
0x7C	USR	UART Status Register	Page: 229
0x80	TFL	Transmit FIFO Level Register	Page: 231
0x84	RFL	Receive FIFO Level Register	Page: 232
0x88	SRR	Software Reset Register	Page: 232
0x8C	SRTS	Shadow Request-to-Send Register	Page: 233
0x90	SBCR	Shadow Break Control Register	Page: 234
0x94	SDMAM	Shadow DMA Mode Register	Page: 235
0x98	SFE	Shadow FIFO Enable Register	Page: 235
0x9C	SRT	Shadow RCVR Trigger Register	Page: 236
0xA0	STET	Shadow TX Empty Trigger Register	Page: 237
0xA4	HTX	Halt TX Register	Page: 237
0xA8	DMASA	DMA Software Acknowledge Register	Page: 238
0xF4	CPR	Component Parameter Register	Page: 239
0xF8	UCV	Component Version Register	Page: 241
0xFC	CTR	Component Type Register	Page: 241

A.9.1 Receive Buffer Register (RBR_THR_DLL)

Receive Buffer Register, reading this register when the DLAB bit is zero; Transmit Holding Register, writing to this register when the DLAB is zero; Divisor Latch (Low), when DLAB bit is one

Instance Name

RBR_DLL_THR

Offset

0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							RBR_THR_DLL								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 262: Receive Buffer Register (RBR_THR_DLL)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 262: Receive Buffer Register (RBR_THR_DLL)

Bits	Name	Type	Reset	Description
7:0	RBR_THR_DLL	R/W	0x0	Receive Buffer Register: This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur. Transmit Holding Register: This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost. Divisor Latch (Low): This register makes up the lower 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. If UART_16550_COMPATIBLE == No, then this register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero), otherwise this register may only be accessed when the DLAB bit (LCR[7]) is set. The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design (CLOCK_MODE == Enabled)) frequency divided by sixteen times the value of the baud rate divisor, as follows: $\text{baud rate} = (\text{serial clock freq}) / (16 * \text{divisor})$ Note: that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications will occur. Also, once the DLL is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

A.9.2 Interrupt Enable Register (IER_DLH)

Interrupt Enable Register, when the DLAB bit is zero; Divisor Latch (High), when the DLAB bit is one.

Interrupt Enable Register:

Each of the bits used has a different function and will be detailed in the bit field descriptions.

Divisor Latch (High):

This register makes up the upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. If UART_16550_COMPATIBLE == NO, then this register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero), otherwise this register may only be accessed when the DLAB bit (LCR[7]) is set. The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design (CLOCK_MODE == Enabled)) frequency divided by sixteen times the value of the baud rate divisor, as follows:

$\text{baud rate} = (\text{serial clock freq}) / (16 * \text{divisor})$



Note

that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications will occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

Instance Name

IER_DLH

Offset

0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							PTIME_DLH7	Reserved		EDSSI_DLH3	ELSI_DLH2	ETBEL_DLH1	ERBEL_DLH0		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?	?	0	0	0	0

Table 263: Interrupt Enable Register (IER_DLH)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	PTIME_DLH7	R/W	0x0	- Interrupt Enable Register: PTIME, Programmable THRE Interrupt Mode Enable. Writeable only when THRE_MODE_USER == Enabled, always readable. This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled - Divisor Latch (High): Bit[7] of the 8 bit DLH register. 0x0: disabled 0x1: enabled
6:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 263: Interrupt Enable Register (IER_DLH)

Bits	Name	Type	Reset	Description
3	EDSSI_DLH3	R/W	0x0	- Interrupt Enable Register: EDSSI, Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 = disabled 1 = enabled - Divisor Latch (High): Bit[3] of the 8 bit DLH register. 0x0: disabled 0x1: enabled
2	ELSI_DHL2	R/W	0x0	- Interrupt Enable Register: ELSI, Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled - Divisor Latch (High): Bit[2] of the 8 bit DLH register. 0x0: disabled 0x1: enabled
1	ETBEI_DLH1	R/W	0x0	- Interrupt Enable Register: ETBEI, Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = disabled 1 = enabled - Divisor Latch (High): Bit[1] of the 8 bit DLH register. 0x0: disabled 0x1: enabled
0	ERBFI_DLH0	R/W	0x0	- Interrupt Enable Register: ERBFI, Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled - Divisor Latch (High): Bit[0] of the 8 bit DLH register. 0x0: disabled 0x1: enabled

A.9.3 Interrupt Identification Register (IIR_FCR)

Interrupt Identification Register, reading this register; FIFO Control Register, writing to this register.

Interrupt Identification Register:

Bits[7:6], FIFOs Enabled (or FIFOSE):

This is used to indicate whether the FIFOs are enabled or disabled.

- 00 = disabled.
- 11 = enabled.

Bits[3:0], Interrupt ID (or IID):

This indicates the highest priority pending interrupt which can be one of the following types:

- 0000 = modem status.
- 0001 = no interrupt pending.
- 0010 = THR empty.
- 0100 = received data available.
- 0110 = receiver line status.
- 0111 = busy detect.
- 1100 = character timeout.



Note

an interrupt of type 0111 (busy detect) will never get indicated if
UART_16550_COMPATIBLE == YES in coreConsultant.

FIFO Control Register:

This register is only valid when the UART is configured to have FIFOs implemented (FIFO_MODE != NONE). If FIFOs are not implemented, this register does not exist and writing to this register address will have no effect.

Bits[7:6], RCVR Trigger (or RT):

This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt will be generated. In auto flow control mode it is used to determine when the rts_n signal will be de-asserted. It also determines when the dma_rx_req_n signal will be asserted when in certain modes of operation. See section 5.9 on page 56 for details on DMA support. The following trigger levels are supported:

- 00 = 1 character in the FIFO
- 01 = FIFO 1/4 full
- 10 = FIFO 1/2 full
- 11 = FIFO 2 less than full

Bits[5:4], TX Empty Trigger (or TET):

Writes will have no effect when THRE_MODE_USER == Disabled. This is used to select the empty threshold level at which the THRE Interrupts will be generated when the mode is active. It also determines when the dma_tx_req_n signal will be asserted when in certain modes of operation. See section 5.9 on page 56 for details on DMA support. The following trigger levels are supported:

- 00 = FIFO empty
- 01 = 2 characters in the FIFO
- 10 = FIFO 1/4 full
- 11 = FIFO 1/2 full

Bit[3], DMA Mode (or DMAM):

This determines the DMA signaling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA == NO). See section 5.9 on page 56 for details on DMA support.

- 0 = mode 0
- 1 = mode 1

Bit[2], XMIT FIFO Reset (or XFIFOR):

This resets the control portion of the transmit FIFO and treats the FIFO as empty. This will also de-assert the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing' and it is not necessary to clear this bit.

Bit[1], RCVR FIFO Reset (or RFIFOR):

This resets the control portion of the receive FIFO and treats the FIFO as empty. This will also de-assert the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing' and it is not necessary to clear this bit.

Bit[0], FIFO Enable (or FIFOE):

This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs will be reset.

Instance Name

Offset

IIR_FCR

0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DATA																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 264: Interrupt Identification Register (IIR_FCR)

Bits	Name	Type	Reset	Description
31:0	DATA	R/W	0x0	

A.9.4 Line Control Register (LCR)

Line Control Register

Instance Name

Offset

LCR

0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								DLAB	BREAK	Reserved	EPS	PEN	STOP	DLS	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	0	0	0	0	0

Table 265: Line Control Register (LCR)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 265: Line Control Register (LCR)

Bits	Name	Type	Reset	Description
7	DLAB	R/W	0x0	Divisor Latch Access Bit. If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable, always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.
6	BREAK	R/W	0x0	Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE == Enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.
5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	EPS	R/W	0x0	Even Parity Select. If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable, always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic '1's is transmitted or checked. If set to zero, an odd number of logic '1's is transmitted or checked.
3	PEN	R/W	0x0	Parity Enable. If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable, always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled 0x0: parity disabled 0x1: parity enabled

Table 265: Line Control Register (LCR)

Bits	Name	Type	Reset	Description
2	STOP	R/W	0x0	Number of stop bits. If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable, always readable. This is used to select the number of stop bits per character that the peripheral will transmit and receive. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected the receiver will only check the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit 0x0: 1 stop bit 0x1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit
1:0	DLS	R/W	0x0	Data Length Select. If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable, always readable. This is used to select the number of data bits per character that the peripheral will transmit and receive. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits 0x0: 5 bits 0x1: 6 bits 0x2: 7 bits 0x3: 8 bits

A.9.5 Modem Control Register (MCR)

Modem Control Register

Instance Name

MCR

Offset

0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																									SIRE	AFCE	LOOPBACK	OUT2	OUT1	RTS	DTR
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0

Table 266: Modem Control Register (MCR)

Bits	Name	Type	Reset	Description
31:7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
6	SIRE	R/W	0x0	SIR Mode Enable. Writeable only when SIR_MODE == Enabled, always readable. This is used to enable/ disable the IrDA SIR Mode features as described in section 5.2 on page 47. 0 = IrDA SIR Mode disabled 1 = IrDA SIR Mode enabled 0x0: IrDA SIR Mode disabled 0x1: IrDA SIR Mode enabled
5	AFCE	R/W	0x0	Auto Flow Control Enable. Writeable only when AFCE_MODE == Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled as described in section 5.6 on page 51. 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled 0x0: Auto Flow Control Mode disabled 0x1: Auto Flow Control Mode enabled
4	LOOPBACK	R/W	0x0	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled OR NOT active, MCR[6] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.
3	OUT2	R/W	0x0	OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 0 = out2_n de-asserted (logic 1) 1 = out2_n asserted (logic 0) Note: that in Loopback mode (MCR[4] set to one), the out2_n output is held inactive high while the value of this location is internally looped back to an input. 0x0: out2_n de-asserted (logic 1) 0x1: out2_n asserted (logic 0)

Table 266: Modem Control Register (MCR)

Bits	Name	Type	Reset	Description
2	OUT1	R/W	0x0	OUT1. This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is: 0 = out1_n de-asserted (logic 1) 1 = out1_n asserted (logic 0) Note: that in Loopback mode (MCR[4] set to one), the out1_n output is held inactive high while the value of this location is internally looped back to an input. 0x0: out1_n de-asserted (logic 1) 0x1: out1_n asserted (logic 0)
1	RTS	R/W	0x0	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal will be de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.
0	DTR	R/W	0x0	Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 0 = dtr_n de-asserted (logic 1) 1 = dtr_n asserted (logic 0) The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] set to one), the dtr_n output is held inactive high while the value of this location is internally looped back to an input. 0x0: dtr_n de-asserted (logic 1) 0x1: dtr_n asserted (logic 0)

A.9.6 Line Status Register (LSR)

Line Status Register

Instance Name

LSR

Offset

0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							RFE	TEMT	THRE	BI	FE	PE	OE	DR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	1	0	0	0	0	0

Table 267: Line Status Register (LSR)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	RFE	R	0x0	Receiver FIFO Error bit. This bit is only relevant when FIFO_MODE != NONE AND FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. That is: 0 = no error in RX FIFO 1 = error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO. 0x0: no error in RX FIFO 0x1: error in RX FIFO
6	TEMT	R	0x1	Transmitter Empty bit. If in FIFO mode (FIFO_MODE != NONE) and FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in the non-FIFO mode or FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.
5	THRE	R	0x1	Transmit Holding Register Empty bit. If THRE_MODE_USER == Disabled or THRE mode is disabled (IER[7] set to zero) and regardless of FIFOs being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If THRE_MODE_USER == Enabled AND FIFO_MODE != NONE and both modes are active (IER[7] set to one and FCR[0] set to one respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting. Programmable THRE interrupt mode operation is described in detail in section 5.7 on page 52.

Table 267: Line Status Register (LSR)

Bits	Name	Type	Reset	Description
4	BI	R	0x0	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. If in UART mode it is set whenever the serial input, <i>sin</i>, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits. If in infrared mode it is set whenever the serial input, <i>sir_in</i>, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read.
3	FE	R	0x0	Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs the UART will try resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) will be set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no framing error 1 = framing error Reading the LSR clears the FE bit. 0x0: no framing error 0x1: framing error
2	PE	R	0x0	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) will be set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no parity error 1 = parity error Reading the LSR clears the PE bit. 0x0: no parity error 0x1: parity error

Table 267: Line Status Register (LSR)

Bits	Name	Type	Reset	Description
1	OE	R	0x0	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 0 = no overrun error 1 = overrun error Reading the LSR clears the OE bit. 0x0: no overrun error 0x1: overrun error
0	DR	R	0x0	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready This bit is cleared when the RBR is read in the non-FIFO mode, or when the receiver FIFO is empty, in the FIFO mode. 0x0: no data ready 0x1: data ready

A.9.7 Modem Status Register (MSR)

Modem Status Register

It should be noted that whenever bits 0, 1, 2 or 3 is set to logic one, to indicate a change on the modem control inputs, a modem status interrupt will be generated if enabled via the IER regardless of when the change occurred. Since the delta bits (bits 0, 1, 3) can get set after a reset if their respective modem signals are active (see individual bits for details), a read of the MSR after reset can be performed to prevent unwanted interrupts.

Instance Name

MSR

Offset

0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 268: Modem Status Register (MSR)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 268: Modem Status Register (MSR)

Bits	Name	Type	Reset	Description
7	DCD	R	0x0	Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n. That is this bit is the complement dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set. 0 = dcd_n input is de-asserted (logic 1) 1 = dcd_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), DCD is the same as MCR[3] (Out2). 0x0: dcd_n input is de-asserted (logic 1) 0x1: dcd_n input is asserted (logic 0)
6	RI	R	0x0	Ring Indicator. This is used to indicate the current state of the modem control line ri_n. That is this bit is the complement ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set. 0 = ri_n input is de-asserted (logic 1) 1 = ri_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), RI is the same as MCR[2] (Out1). 0x0: ri_n input is de-asserted (logic 1) 0x1: ri_n input is asserted (logic 0)
5	DSR	R	0x0	Data Set Ready. This is used to indicate the current state of the modem control line dsr_n. That is this bit is the complement dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with the UART. 0 = dsr_n input is de-asserted (logic 1) 1 = dsr_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), DSR is the same as MCR[0] (DTR). 0x0: dsr_n input is de-asserted (logic 1) 0x1: dsr_n input is asserted (logic 0)
4	CTS	R	0x0	Clear to Send. This is used to indicate the current state of the modem control line cts_n. That is, this bit is the complement cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with the UART. 0 = cts_n input is de-asserted (logic 1) 1 = cts_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), CTS is the same as MCR[1] (RTS). 0x0: cts_n input is de-asserted (logic 1) 0x1: cts_n input is asserted (logic 0)

Table 268: Modem Status Register (MSR)

Bits	Name	Type	Reset	Description
3	DDCD	R	0x0	Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read. That is: 0 = no change on dcd_n since last read of MSR 1 = change on dcd_n since last read of MSR Reading the MSR clears the DDCD bit. In Loopback Mode (MCR[4] set to one), DDCD reflects changes on MCR[3] (Out2). Note: if the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit will get set when the reset is removed if the dcd_n signal remains asserted. 0x0: no change on dcd_n since last read of MSR 0x1: change on dcd_n since last read of MSR
2	TERI	R	0x0	Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active low, to an inactive high state) has occurred since the last time the MSR was read. That is: 0 = no change on ri_n since last read of MSR 1 = change on ri_n since last read of MSR Reading the MSR clears the TERI bit. In Loopback Mode (MCR[4] set to one), TERI reflects when MCR[2] (Out1) has changed state from a high to a low. 0x0: no change on ri_n since last read of MSR 0x1: change on ri_n since last read of MSR
1	DDSR	R	0x0	Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read. That is: 0 = no change on dsr_n since last read of MSR 1 = change on dsr_n since last read of MSR Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] set to one), DDSR reflects changes on MCR[0] (DTR). Note: if the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDSR bit will get set when the reset is removed if the dsr_n signal remains asserted. 0x0: no change on dsr_n since last read of MSR 0x1: change on dsr_n since last read of MSR

Table 268: Modem Status Register (MSR)

Bits	Name	Type	Reset	Description
0	DCTS	R	0x0	Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read. That is: 0 = no change on cts_n since last read of MSR 1 = change on cts_n since last read of MSR Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] set to one), DCTS reflects changes on MCR[1] (RTS). Note: if the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit will get set when the reset is removed if the cts_n signal remains asserted. 0x0: no change on cts_n since last read of MSR 0x1: change on cts_n since last read of MSR

A.9.8 Scratchpad Register (SCR)

Scratchpad Register

Instance Name

SCR

Offset

0x1C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							SCR								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 269: Scratchpad Register (SCR)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SCR	R/W	0x0	This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART.

A.9.9 Low Power Divisor Latch Low Register (LPDLL)

Low Power Divisor Latch Low

Instance Name

LPDLL

Offset

0x20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							LPDLL								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 270: Low Power Divisor Latch Low Register (LPDLL)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	LPDLL	R/W	0x0	This register makes up the lower 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART which must give a baud rate of 115.2K. This is required for SiR Low Power (minimum pulse width) detection at the receiver. If <code>UART_16550_COMPATIBLE == NO</code>, then this register may only be accessed when the DLAB bit (<code>LCR[7]</code>) is set and the UART is not busy (<code>USR[0]</code> is zero), otherwise this register may only be accessed when the DLAB bit (<code>LCR[7]</code>) is set. The output low power baud rate is equal to the serial clock (<code>sclk</code>) frequency divided by sixteen times the value of the baud rate divisor, as follows: $\text{Low power baud rate} = (\text{serial clock freq}) / (16 * \text{divisor})$ Therefore a divisor must be selected to give a baud rate of 115.2K. Note: that with the Low Power Divisor Latch Registers (LPDLL and LPDLH) set to zero, the low power baud clock is disabled and no low power pulse detection (or any pulse detection for that matter) will occur at the receiver. Also, once the LPDLL is set at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

A.9.10 Low Power Divisor Latch High Register (LPDLH)

Low Power Divisor Latch High

Instance Name

LPDLH

Offset

0x24

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							LPDLH								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 271: Low Power Divisor Latch High Register (LPDLH)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 271: Low Power Divisor Latch High Register (LPDLH)

Bits	Name	Type	Reset	Description
7:0	LPDLH	R/W	0x0	This register makes up the upper 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART which must give a baud rate of 115.2K. This is required for SiR Low Power (minimum pulse width) detection at the receiver. If UART_16550_COMPATIBLE == NO, then this register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero), otherwise this register may only be accessed when the DLAB bit (LCR[7]) is set. The output low power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: Low power baud rate = (serial clock freq) / (16 * divisor) Therefore a divisor must be selected to give a baud rate of 115.2K. Note: that with the Low Power Divisor Latch Registers (LPDLL and LPDLH) set to zero, the low power baud clock is disabled and no low power pulse detection (or any pulse detection for that matter) will occur at the receiver. Also, once the LPDLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

A.9.11 Shadow Receive Buffer Register 0 (SRBR_STHR0)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name

SRBR_STHR0

Offset

0x30

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							SRBR_STHR0								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 272: Shadow Receive Buffer Register 0 (SRBR_STHR0)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 272: Shadow Receive Buffer Register 0 (SRBR_STHR0)

Bits	Name	Type	Reset	Description
7:0	SRBR_STHR0	R/W	0x0	Shadow Receive Buffer Register 0: This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur. Shadow Transmit Holding Register 0: This is a shadow register for the THR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

A.9.12 Shadow Receive Buffer Register 1 (SRBR_STHR1)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR1

Offset
0x34

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							SRBR_STHR1								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 273: Shadow Receive Buffer Register 1 (SRBR_STHR1)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR1	R/W	0x0	See srbr_sthr0 description.

A.9.13 Shadow Receive Buffer Register 2 (SRBR_STHR2)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR2

Offset
0x38

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							SRBR_STHR2								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 274: Shadow Receive Buffer Register 2 (SRBR_STHR2)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR2	R/W	0x0	See srbr_sthr0 description.

A.9.14 Shadow Receive Buffer Register 3 (SRBR_STHR3)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR3

Offset
0x3C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							SRBR_STHR3								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 275: Shadow Receive Buffer Register 3 (SRBR_STHR3)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR3	R/W	0x0	See srbr_sthr0 description.

A.9.15 Shadow Receive Buffer Register 4 (SRBR_STHR4)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR4

Offset
0x40

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								SRBR_STHR4							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 276: Shadow Receive Buffer Register 4 (SRBR_STHR4)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR4	R/W	0x0	See srbr_sthr0 description.

A.9.16 Shadow Receive Buffer Register 5 (SRBR_STHR5)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR5

Offset
0x44

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								SRBR_STHR5							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 277: Shadow Receive Buffer Register 5 (SRBR_STHR5)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR5	R/W	0x0	See srbr_sthr0 description.

A.9.17 Shadow Receive Buffer Register 6 (SRBR_STHR6)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR6

Offset
0x48

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								SRBR_STHR6							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 278: Shadow Receive Buffer Register 6 (SRBR_STHR6)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR6	R/W	0x0	See srbr_sthr0 description.

A.9.18 Shadow Receive Buffer Register 7 (SRBR_STHR7)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR7

Offset
0x4C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							SRBR_STHR7								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 279: Shadow Receive Buffer Register 7 (SRBR_STHR7)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR7	R/W	0x0	See srbr_sthr0 description.

A.9.19 Shadow Receive Buffer Register 8 (SRBR_STHR8)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR8

Offset
0x50

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							SRBR_STHR8								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 280: Shadow Receive Buffer Register 8 (SRBR_STHR8)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR8	R/W	0x0	See srbr_sthr0 description.

A.9.20 Shadow Receive Buffer Register 9 (SRBR_STHR9)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR9

Offset
0x54

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								SRBR_STHR9							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 281: Shadow Receive Buffer Register 9 (SRBR_STHR9)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR9	R/W	0x0	See srbr_sthr0 description.

A.9.21 Shadow Receive Buffer Register 10 (SRBR_STHR10)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR10

Offset
0x58

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								SRBR_STHR10							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 282: Shadow Receive Buffer Register 10 (SRBR_STHR10)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR10	R/W	0x0	See srbr_sthr0 description.

A.9.22 Shadow Receive Buffer Register 11 (SRBR_STHR11)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR11

Offset
0x5C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								SRBR_STHR11							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 283: Shadow Receive Buffer Register 11 (SRBR_STHR11)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR11	R/W	0x0	See srbr_sthr0 description.

A.9.23 Shadow Receive Buffer Register 12 (SRBR_STHR12)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR12

Offset
0x60

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							SRBR_STHR12								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 284: Shadow Receive Buffer Register 12 (SRBR_STHR12)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR12	R/W	0x0	See srbr_sthr0 description.

A.9.24 Shadow Receive Buffer Register 13 (SRBR_STHR13)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR13

Offset
0x64

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							SRBR_STHR13								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 285: Shadow Receive Buffer Register 13 (SRBR_STHR13)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR13	R/W	0x0	See srbr_sthr0 description.

A.9.25 Shadow Receive Buffer Register 14 (SRBR_STHR14)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR14

Offset
0x68

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							SRBR_STHR14								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 286: Shadow Receive Buffer Register 14 (SRBR_STHR14)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR14	R/W	0x0	See srbr_sthr0 description.

A.9.26 Shadow Receive Buffer Register 15 (SRBR_STHR15)

Shadow Receive Buffer Register, when reading; Shadow Transmit Holding Register, when writing to this register

Instance Name
SRBR_STHR15

Offset
0x6C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							SRBR_STHR15								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 287: Shadow Receive Buffer Register 15 (SRBR_STHR15)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	SRBR_STHR15	R/W	0x0	See srbr_sthr0 description.

A.9.27 FIFO Access Register (FAR)

FIFO Access Register

Instance Name
FAR

Offset
0x70

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														FAR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 288: FIFO Access Register (FAR)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	FAR	R/W	0x0	Writes will have no effect when FIFO_ACCESS == No, always readable. This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled Note: that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFOs are treated as empty. 0x0: FIFO access mode disabled 0x1: FIFO access mode enabled

A.9.28 Transmit FIFO Read Register (TFR)

Transmit FIFO Read

Instance Name

TFR

Offset

0x74

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							TFR								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 289: Transmit FIFO Read Register (TFR)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	TFR	R	0x0	Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO. When FIFOs are not implemented or not enabled, reading this register gives the data in the THR.

A.9.29 Receive FIFO Write Register (RFW)

Receive FIFO Write

Instance Name

RFW

Offset

0x78

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Field	Reserved																						RFE	RFE	RFWD									
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0		

Table 290: Receive FIFO Write Register (RFW)

Bits	Name	Type	Reset	Description
31:10	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
9	RFFE	W	0x0	Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, this bit is used to write framing error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write framing error detection information to the RBR.
8	RFPE	W	0x0	Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, this bit is used to write parity error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write parity error detection information to the RBR.
7:0	RFWD	W	0x0	Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs are not implemented or not enabled, the data that is written to the RFWD is pushed into the RBR.

A.9.30 UART Status Register (USR)

UART Status register.

Instance Name

USR

Offset

0x7C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											RFF	RFNE	TFE	TFNF	BUSY
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1	1	0

Table 291: UART Status Register (USR)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	RFF	R	0x0	Receive FIFO Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO is completely full. That is: 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full. 0x0: Receive FIFO not full 0x1: Receive FIFO Full
3	RFNE	R	0x0	Receive FIFO Not Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty. 0x0: Receive FIFO is empty 0x1: Receive FIFO is not empty
2	TFE	R	0x1	Transmit FIFO Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty. 0x0: Transmit FIFO is not empty 0x1: Transmit FIFO is empty
1	TFNF	R	0x1	Transmit FIFO Not Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO is not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full. 0x0: Transmit FIFO is full 0x1: Transmit FIFO is not full

Table 291: UART Status Register (USR)

Bits	Name	Type	Reset	Description
0	BUSY	R	0x0	UART Busy. This bit is only valid when UART_16550_COMPATIBLE == NO. This indicates that a serial transfer is in progress, when cleared indicates that the UART is idle or inactive. 0 = UART is idle or inactive 1 = UART is busy (actively transferring data) Note: that it is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the UART has no data in the THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the UART. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE == Enabled) the assertion of this bit will also be delayed by several cycles of the slower clock. 0x0: uart is idle or inactive 0x1: uart is busy (actively transferring data)

A.9.31 Transmit FIFO Level Register (TFL)

Instance Name

TFL

Offset

0x80

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										TFL					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	

Table 292: Transmit FIFO Level Register (TFL)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4:0	TFL	R	0x0	Transmit FIFO Level. This indicates the number of data entries in the transmit FIFO.

A.9.32 Receive FIFO Level Register (RFL)

Receive FIFO Level.

Instance Name

RFL

Offset

0x84

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											RFL				
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0

Table 293: Receive FIFO Level Register (RFL)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4:0	RFL	R	0x0	Receive FIFO Level. This indicates the number of data entries in the receive FIFO.

A.9.33 Software Reset Register (SRR)

Software Reset Register.

Instance Name

SRR

Offset

0x88

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																													XFR	RFR	UR
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Table 294: Software Reset Register (SRR)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2	XFR	W	0x0	XMIT FIFO Reset. Writes will have no effect when FIFO_MODE == NONE. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the transmit FIFO. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This will also de-assert the DMA TX request and single signals when additional DMA handshaking signals are selected

Table 294: Software Reset Register (SRR)

Bits	Name	Type	Reset	Description
1	RFR	W	0x0	RCVR FIFO Reset. Writes will have no effect when FIFO_MODE == NONE. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO. This resets the control portion of the receive FIFO and treats the FIFO as empty. This will also de-assert the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing' and it is not necessary to clear this bit.
0	UR	W	0x0	UART Reset. This asynchronously resets the UART and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains will be reset. When UART reset is asserted, software should wait at least 4 sclk before programming other UART registers.

A.9.34 Shadow Request-to-Send Register (SRTS)

Shadow Request to Send.

Instance Name
SRTS

Offset
0x8C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														SRTS	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 295: Shadow Request-to-Send Register (SRTS)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 295: Shadow Request-to-Send Register (SRTS)

Bits	Name	Type	Reset	Description
0	SRTS	R/W	0x0	Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read modify write on the MCR. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.

A.9.35 Shadow Break Control Register (SBCR)

Shadow Break Control Register.

Instance Name
SBCR

Offset
0x90

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															SBCB
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 296: Shadow Break Control Register (SBCR)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	SBCB	R/W	0x0	Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE == Enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver.

A.9.36 Shadow DMA Mode Register (SDMAM)

Shadow DMA Mode.

Instance Name

SDMAM

Offset

0x94

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														SDMAM	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		?

Table 297: Shadow DMA Mode Register (SDMAM)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	SDMAM	R/W	0x0	Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the DMA Mode bit gets updated. This determines the DMA signaling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA == NO). See section 5.9 on page 54 for details on DMA support. 0 = mode 0 1 = mode 1 0x0: mode 0 0x1: mode 1

A.9.37 Shadow FIFO Enable Register (SFE)

Shadow FIFO Enable

Instance Name

SFE

Offset

0x98

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														SFE	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 298: Shadow FIFO Enable Register (SFE)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 298: Shadow FIFO Enable Register (SFE)

Bits	Name	Type	Reset	Description
0	SFE	R/W	0x0	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the FIFO enable bit gets updated. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. If this bit is set to zero (disabled) after being enabled then both the XMIT and RCVR controller portion of FIFOs will be reset.

A.9.38 Shadow RCVR Trigger Register (SRT)

Shadow RCVR Trigger

Instance Name

SRT

Offset

0x9C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														SRT	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 299: Shadow RCVR Trigger Register (SRT)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1:0	SRT	R/W	0x0	Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the RCVR trigger bit gets updated. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt will be generated. It also determines when the dma_rx_req_n signal will be asserted when DMA Mode (FCR[3]) is set to one. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO 1/4 full 10 = FIFO 1/2 full 11 = FIFO 2 less than full 0x0: 1 character in the FIFO 0x1: FIFO 1/4 full 0x2: FIFO 1/2 full 0x3: FIFO 2 less than full

A.9.39 Shadow TX Empty Trigger Register (STET)

Shadow TX Empty Trigger

Instance Name

STET

Offset

0xA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														STET	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 300: Shadow TX Empty Trigger Register (STET)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1:0	STET	R/W	0x0	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the TX empty trigger bit gets updated. Writes will have no effect when THRE_MODE_USER == Disabled. This is used to select the empty threshold level at which the THRE Interrupts will be generated when the mode is active. These threshold levels are also described in. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO 1/4 full 11 = FIFO 1/2 full 0x0: FIFO empty 0x1: 2 characters in the FIFO 0x2: FIFO 1/4 full 0x3: FIFO 1/2 full

A.9.40 Halt TX Register (HTX)

Halt TX

Instance Name

HTX

Offset

0xA4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														HTX	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 301: Halt TX Register (HTX)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 301: Halt TX Register (HTX)

Bits	Name	Type	Reset	Description
0	HTX	R/W	0x0	Halt TX. Writes will have no effect when FIFO_MODE == NONE, always readable. This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. Note, if FIFOs are implemented and not enabled the setting of the halt TX register will have no effect on operation. 0 = Halt TX disabled 1 = Halt TX enabled 0x0: Halt TX disabled 0x1: Halt TX enabled

A.9.41 DMA Software Acknowledge Register (DMASA)

DMA Software Acknowledge

Instance Name

DMASA

Offset

0xA8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														DMASA	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		?

Table 302: DMA Software Acknowledge Register (DMASA)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	DMASA	W	0x0	DMA Software Acknowledge. Writes will have no effect when DMA_EXTRA == No. This register is use to perform DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the UART should clear its request. This will cause the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing' and it is not necessary to clear this bit.

A.9.42 Component Parameter Register (CPR)

Component Parameter Register

Instance Name

CPR

Offset

0xF4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								FIFO_MODE								Reserved		DMA_EXTRA	UART_ADD_ENCODED_PARAMS	SHADOW	FIFO_STAT	FIFO_ACCESS	ADDITIONAL_FEAT	SIR_LP_MODE	SIR_MODE	THRE_MODE	AFCE_MODE	Reserved		APB_DATA_WIDTH	
Default	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	?	?	0	0	0	0	0	0	0	0	0	0	?	?	0	0

Table 303: Component Parameter Register (CPR)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23:16	FIFO_MODE	R	0x0	Encoding of FIFO_MODE configuration parameter value. 0x00 = 0, 0x01 = 16, 0x02 = 32, to 0x80 = 2048, 0x81- 0xff = reserved 0x0: 0, 0x1: 16, 0x2: 32, to 0x80: 2048, 0x81- 0xff = reserved
15:14	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
13	DMA_EXTRA	R	0x0	Encoding of DMA_EXTRA configuration parameter value. 0 = FALSE, 1 = TRUE 0x0: FALSE, 0x1: TRUE
12	UART_ADD_ENCODED_PARAMS	R	0x0	Encoding of UART_ADD_ENCODED_PARAMS configuration parameter value. 0 = FALSE, 1 = TRUE 0x0: FALSE, 0x1: TRUE

Table 303: Component Parameter Register (CPR)

Bits	Name	Type	Reset	Description
11	SHADOW	R	0x0	Encoding of SHADOW configuration parameter value. 0 = FALSE, 1 = TRUE 0x0: FALSE, 0x1: TRUE
10	FIFO_STAT	R	0x0	Encoding of FIFO_STAT configuration parameter value. 0 = FALSE, 1 = TRUE 0x0: FALSE, 0x1: TRUE
9	FIFO_ACCESS	R	0x0	Encoding of FIFO_ACCESS configuration parameter value. 0 = FALSE, 1 = TRUE 0x0: FALSE, 0x1: TRUE
8	ADDITIONAL_FEAT	R	0x0	Encoding of ADDITIONAL_FEATURES configuration parameter value. 0 = FALSE, 1 = TRUE 0x0: FALSE, 0x1: TRUE
7	SIR_LP_MODE	R	0x0	Encoding of SIR_LP_MODE configuration parameter value. 0 = FALSE, 1 = TRUE 0x0: FALSE, 0x1: TRUE
6	SIR_MODE	R	0x0	Encoding of SIR_MODE configuration parameter value. 0 = FALSE, 1 = TRUE 0x0: FALSE, 0x1: TRUE
5	THRE_MODE	R	0x0	Encoding of THRE_MODE configuration parameter value. 0 = FALSE, 1 = TRUE 0x0: FALSE, 0x1: TRUE
4	AFCE_MODE	R	0x0	Encoding of AFCE_MODE configuration parameter value. 0 = FALSE, 1 = TRUE 0x0: FALSE, 0x1: TRUE
3:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 303: Component Parameter Register (CPR)

Bits	Name	Type	Reset	Description
1:0	APB_DATA_WIDTH	R	0x0	Encoding of APB_DATA_WIDTH configuration parameter value. 00 = 8 bits, 01 = 16 bits, 10 = 32 bits, 11 = reserved 0x0: 8 bits, 0x1: 16 bits, 0x2: 32 bits, 0x3: reserved

A.9.43 Component Version Register (UCV)

Component Version

Instance Name

UCV

Offset

0xF8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	UART_COMPONENT_VERSION																															
Default	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	0	0	0	1	0	1	0	1	0

Table 304: Component Version Register (UCV)

Bits	Name	Type	Reset	Description
31:0	UART_COMPONENT_VERSION	R	0x3331_322A	ASCII value for each number in the version, followed by *. For example 32_30_31_2A represents the version 2.01*

A.9.44 Component Type Register (CTR)

Component Type Register

Instance Name

CTR

Offset

0xFC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	PERIPHERAL_ID																															
Default	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0

Table 305: Component Type Register (CTR)

Bits	Name	Type	Reset	Description
31:0	PERIPHERAL_ID	R	0x4457_0110	This register contains the peripherals identification code.

A.10 GPIO Register Information

Table 306: GPIO Register Summary

Offset	Name	Description	Details
0x00	GPLR0	GPIO Pin Level Register	Page: 243
0x04	GPLR1	GPIO Pin Level Register	Page: 244
0x08	GPLR2	GPIO Pin Level Register	Page: 244
0x0C	GPDR0	GPIO Pin Direction Register	Page: 244
0x10	GPDR1	GPIO Pin Direction Register	Page: 245
0x14	GPDR2	GPIO Pin Direction Register	Page: 245
0x18	GPSR0	GPIO Pin Output Set Register	Page: 245
0x1C	GPSR1	GPIO Pin Output Set Register	Page: 246
0x20	GPSR2	GPIO Pin Output Set Register	Page: 246
0x24	GPCR0	GPIO Pin Output Clear Register	Page: 246
0x28	GPCR1	GPIO Pin Output Clear Register	Page: 247
0x2C	GPCR2	GPIO Pin Output Clear Register	Page: 247
0x30	GRER0	GPIO Rising Edge detect Enable Register	Page: 247
0x34	GRER1	GPIO Rising Edge detect Enable Register	Page: 248
0x38	GRER2	GPIO Rising Edge detect Enable Register	Page: 248
0x3C	GFER0	GPIO Falling Edge detect Enable Register	Page: 248
0x40	GFER1	GPIO Falling Edge detect Enable Register	Page: 249
0x44	GFER2	GPIO Falling Edge detect Enable Register	Page: 249
0x48	GEDR0	GPIO Edge detect Status Register	Page: 249
0x4C	GEDR1	GPIO Edge detect Status Register	Page: 250
0x50	GEDR2	GPIO Edge detect Status Register	Page: 250
0x54	GSDR0	GPIO Pin Bitwise Set Direction Register	Page: 250
0x58	GSDR1	GPIO Pin Bitwise Set Direction Register	Page: 251
0x5C	GSDR2	GPIO Pin Bitwise Set Direction Register	Page: 251
0x60	GCDR0	GPIO Pin Bitwise Clear Direction Register	Page: 251
0x64	GCDR1	GPIO Pin Bitwise Clear Direction Register	Page: 252
0x68	GCDR2	GPIO Pin Bitwise Clear Direction Register	Page: 252
0x6C	GSRER0	GPIO Bitwise Set Rising Edge detect Enable Register	Page: 252

Table 306: GPIO Register Summary

Offset	Name	Description	Details
0x70	GSRER1	GPIO Bitwise set Rising Edge detect Enable Register	Page: 253
0x74	GSRER2	GPIO Bitwise set Rising Edge detect Enable Register	Page: 253
0x78	GCRER0	GPIO Bitwise Clear Rising Edge detect Enable Register	Page: 253
0x7C	GCRER1	GPIO Bitwise Clear Rising Edge detect Enable Register	Page: 254
0x80	GCRER2	GPIO Bitwise Clear Rising Edge detect Enable Register	Page: 254
0x84	GSFER0	GPIO Bitwise Set Falling Edge detect Enable Register	Page: 254
0x88	GSFER1	GPIO Bitwise set Falling Edge detect Enable Register	Page: 255
0x8C	GSFER2	GPIO Bitwise set Falling Edge detect Enable Register	Page: 255
0x90	GCFER0	GPIO Bitwise Clear Falling Edge detect Enable Register	Page: 255
0x94	GCFER1	GPIO Bitwise Clear Falling Edge detect Enable Register	Page: 256
0x98	GCFER2	GPIO Bitwise Clear Falling Edge detect Enable Register	Page: 256
0x9C	APMASK0	GPIO Bitwise mask of Edge detect Status Register	Page: 256
0xA0	APMASK1	GPIO Bitwise mask of Edge detect Status Register	Page: 257
0xA4	APMASK2	GPIO Bitwise mask of Edge detect Status Register	Page: 257

A.10.1 GPIO Pin Level Register (GPLR0)

Instance Name
GPLR0

Offset
0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GPLR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 307: GPIO Pin Level Register (GPLR0)

Bits	Name	Type	Reset	Description
31:0	GPLR	R	0x0	0 : Port State Low ; 1: Port State High

A.10.2 GPIO Pin Level Register (GPLR1)

Instance Name

GPLR1

Offset

0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GPLR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 308: GPIO Pin Level Register (GPLR1)

Bits	Name	Type	Reset	Description
31:0	GPLR	R	0x0	0 : Port State Low ; 1: Port State High

A.10.3 GPIO Pin Level Register (GPLR2)

Instance Name

GPLR2

Offset

0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GPLR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 309: GPIO Pin Level Register (GPLR2)

Bits	Name	Type	Reset	Description
31:0	GPLR	R	0x0	0 : Port State Low ; 1: Port State High

A.10.4 GPIO Pin Direction Register (GPDR0)

Instance Name

GPDR0

Offset

0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GPDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 310: GPIO Pin Direction Register (GPDR0)

Bits	Name	Type	Reset	Description
31:0	GPDR	R/W	0x0	0: Input port; 1: Output Port

A.10.5 GPIO Pin Direction Register (GPDR1)

Instance Name
GPDR1

Offset
0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GPDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 311: GPIO Pin Direction Register (GPDR1)

Bits	Name	Type	Reset	Description
31:0	GPDR	R/W	0x0	0: Input port; 1: Output Port

A.10.6 GPIO Pin Direction Register (GPDR2)

Instance Name
GPDR2

Offset
0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GPDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 312: GPIO Pin Direction Register (GPDR2)

Bits	Name	Type	Reset	Description
31:0	GPDR	R/W	0x0	0: Input port; 1: Output Port

A.10.7 GPIO Pin Output Set Register (GPSR0)

Instance Name
GPSR0

Offset
0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GPSR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 313: GPIO Pin Output Set Register (GPSR0)

Bits	Name	Type	Reset	Description
31:0	GPSR	W	0x0	0 : Unaffected; 1: Port set if GPIO is configured as output

A.10.8 GPIO Pin Output Set Register (GPSR1)

Instance Name

Offset

GPSR1

0x1C

[illegible]**Table 314: GPIO Pin Output Set Register (GPSR1)**

Bits	Name	Type	Reset	Description
31:0	GPSR	W	0x0	0 : Unaffected; 1: Port set if GPIO is configured as output

A.10.9 GPIO Pin Output Set Register (GPSR2)

Instance Name

Offset

GPSR2

0x20

[illegible]

Table 315: GPIO Pin Output Set Register (GPSR2)

Bits	Name	Type	Reset	Description
31:0	GPSR	W	0x0	0 : Unaffected; 1: Port set if GPIO is configured as output

A.10.10 GPIO Pin Output Clear Register (GPCR0)

Instance Name

Offset

GPCR0

0x24

[illegible]**Table 316: GPIO Pin Output Clear Register (GPCR0)**

Bits	Name	Type	Reset	Description
31:0	GPCR	W	0x0	0 : Unaffected; 1: Port Clear if GPIO is configured as output

A.10.11 GPIO Pin Output Clear Register (GPCR1)

Instance Name Offset
GPCR1 0x28

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GPCR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 317: GPIO Pin Output Clear Register (GPCR1)

Bits	Name	Type	Reset	Description
31:0	GPCR	W	0x0	0 : Unaffected; 1: Port Clear if GPIO is configured as output

A.10.12 GPIO Pin Output Clear Register (GPCR2)

Instance Name Offset
GPCR2 0x2C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GPCR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 318: GPIO Pin Output Clear Register (GPCR2)

Bits	Name	Type	Reset	Description
31:0	GPCR	W	0x0	0 : Unaffected; 1: Port Clear if GPIO is configured as output

A.10.13 GPIO Rising Edge detect Enable Register (GRER0)

Instance Name Offset
GRER0 0x30

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GRER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 319: GPIO Rising Edge detect Enable Register (GRER0)

Bits	Name	Type	Reset	Description
31:0	GRER	R/W	0x0	0 : Disable Rising Edge detection; 1: Set corresponding GEDR Status bit when Rising edge is detected on GPIO input

A.10.14 GPIO Rising Edge detect Enable Register (GRER1)

Instance Name
GRER1

Offset
0x34

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GRER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 320: GPIO Rising Edge detect Enable Register (GRER1)

Bits	Name	Type	Reset	Description
31:0	GRER	R/W	0x0	0 : Disable Rising Edge detection; 1: Set corresponding GEDR Status bit when Rising edge is detected on GPIO input

A.10.15 GPIO Rising Edge detect Enable Register (GRER2)

Instance Name
GRER2

Offset
0x38

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GRER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 321: GPIO Rising Edge detect Enable Register (GRER2)

Bits	Name	Type	Reset	Description
31:0	GRER	R/W	0x0	0 : Disable Rising Edge detection; 1: Set corresponding GEDR Status bit when Rising edge is detected on GPIO input

A.10.16 GPIO Falling Edge detect Enable Register (GFER0)

Instance Name
GFER0

Offset
0x3C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GFER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 322: GPIO Falling Edge detect Enable Register (GFER0)

Bits	Name	Type	Reset	Description
31:0	GFER	R/W	0x0	0 : Disable Falling Edge detection; 1: Set corresponding GEDR Status bit when Falling edge is detected on GPIO input

A.10.17 GPIO Falling Edge detect Enable Register (GFER1)

Instance Name
GFER1

Offset
0x40

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GFER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 323: GPIO Falling Edge detect Enable Register (GFER1)

Bits	Name	Type	Reset	Description
31:0	GFER	R/W	0x0	0 : Disable Falling Edge detection; 1: Set corresponding GEDR Status bit when Falling edge is detected on GPIO input

A.10.18 GPIO Falling Edge detect Enable Register (GFER2)

Instance Name
GFER2

Offset
0x44

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GFER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 324: GPIO Falling Edge detect Enable Register (GFER2)

Bits	Name	Type	Reset	Description
31:0	GFER	R/W	0x0	0 : Disable Falling Edge detection; 1: Set corresponding GEDR Status bit when Falling edge is detected on GPIO input

A.10.19 GPIO Edge detect Status Register (GEDR0)

Instance Name
GEDR0

Offset
0x48

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GEDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 325: GPIO Edge detect Status Register (GEDR0)

Bits	Name	Type	Reset	Description
31:0	GEDR	R/W1CLR	0x0	0: No edge detected on a port as specified by GRERx or GFERx; 1 : Edge detected on a port as specified by GRERx or GFERx

A.10.20 GPIO Edge detect Status Register (GEDR1)

Instance Name
GEDR1

Offset
0x4C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GEDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 326: GPIO Edge detect Status Register (GEDR1)

Bits	Name	Type	Reset	Description
31:0	GEDR	R/W1CLR	0x0	0: No edge detected on a port as specified by GRERx or GFERx; 1 : Edge detected on a port as specified by GRERx or GFERx

A.10.21 GPIO Edge detect Status Register (GEDR2)

Instance Name
GEDR2

Offset
0x50

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GEDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 327: GPIO Edge detect Status Register (GEDR2)

Bits	Name	Type	Reset	Description
31:0	GEDR	R/W1CLR	0x0	0: No edge detected on a port as specified by GRERx or GFERx; 1 : Edge detected on a port as specified by GRERx or GFERx

A.10.22 GPIO Pin Bitwise Set Direction Register (GSDR0)

Instance Name
GSDR0

Offset
0x54

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GSDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 328: GPIO Pin Bitwise Set Direction Register (GSDR0)

Bits	Name	Type	Reset	Description
31:0	GSDR	W	0x0	0: GPDR bit unaffected; 1 : GPDR Bit set and GPIO Pin is set as output

A.10.23 GPIO Pin Bitwise Set Direction Register (GSDR1)

Instance Name Offset
GSDR1 0x58

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GSDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 329: GPIO Pin Bitwise Set Direction Register (GSDR1)

Bits	Name	Type	Reset	Description
31:0	GSDR	W	0x0	0: GPDR bit unaffected; 1 : GPDR Bit set and GPIO Pin is set as output

A.10.24 GPIO Pin Bitwise Set Direction Register (GSDR2)

Instance Name Offset
GSDR2 0x5C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GSDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 330: GPIO Pin Bitwise Set Direction Register (GSDR2)

Bits	Name	Type	Reset	Description
31:0	GSDR	W	0x0	0: GPDR bit unaffected; 1 : GPDR Bit set and GPIO Pin is set as output

A.10.25 GPIO Pin Bitwise Clear Direction Register (GCDR0)

Instance Name Offset
GCDR0 0x60

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GCDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 331: GPIO Pin Bitwise Clear Direction Register (GCDR0)

Bits	Name	Type	Reset	Description
31:0	GCDR	W	0x0	0: GPDR bit unaffected; 1 : GPDR Bit clear and GPIO Pin is set as input

A.10.26 GPIO Pin Bitwise Clear Direction Register (GCDR1)

Instance Name

GCDR1

Offset

0x64

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GCDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 332: GPIO Pin Bitwise Clear Direction Register (GCDR1)

Bits	Name	Type	Reset	Description
31:0	GCDR	W	0x0	0: GPDR bit unaffected; 1 : GPDR Bit clear and GPIO Pin is set as input

A.10.27 GPIO Pin Bitwise Clear Direction Register (GCDR2)

Instance Name

GCDR2

Offset

0x68

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GCDR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 333: GPIO Pin Bitwise Clear Direction Register (GCDR2)

Bits	Name	Type	Reset	Description
31:0	GCDR	W	0x0	0: GPDR bit unaffected; 1 : GPDR Bit clear and GPIO Pin is set as input

A.10.28 GPIO Bitwise Set Rising Edge detect Enable Register (GSRER0)

Instance Name

GSRER0

Offset

0x6C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GSRER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 334: GPIO Bitwise Set Rising Edge detect Enable Register (GSRER0)

Bits	Name	Type	Reset	Description
31:0	GSRER	W	0x0	0 : GRER bit Unaffected; ; 1: GRER Bit set

A.10.29 GPIO Bitwise set Rising Edge detect Enable Register (GSRER1)

Instance Name

GSRER1

Offset

0x70

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GSRER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 335: GPIO Bitwise set Rising Edge detect Enable Register (GSRER1)

Bits	Name	Type	Reset	Description
31:0	GSRER	W	0x0	0 : GRER bit Unaffected; ; 1: GRER Bit set

A.10.30 GPIO Bitwise set Rising Edge detect Enable Register (GSRER2)

Instance Name

GSRER2

Offset

0x74

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GSRER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 336: GPIO Bitwise set Rising Edge detect Enable Register (GSRER2)

Bits	Name	Type	Reset	Description
31:0	GSRER	W	0x0	0 : GRER bit Unaffected; ; 1: GRER Bit Clear

A.10.31 GPIO Bitwise Clear Rising Edge detect Enable Register (GCRER0)

Instance Name

GCRER0

Offset

0x78

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GCRER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 337: GPIO Bitwise Clear Rising Edge detect Enable Register (GCRER0)

Bits	Name	Type	Reset	Description
31:0	GCRER	W	0x0	0 : GRER bit Unaffected; ; 1: GRER Bit Clear

A.10.32 GPIO Bitwise Clear Rising Edge detect Enable Register (GCRER1)

Instance Name

GCRER1

Offset

0x7C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GCRER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 338: GPIO Bitwise Clear Rising Edge detect Enable Register (GCRER1)

Bits	Name	Type	Reset	Description
31:0	GCRER	W	0x0	0 : GRER bit Unaffected; ; 1: GRER Bit Clear

A.10.33 GPIO Bitwise Clear Rising Edge detect Enable Register (GCRER2)

Instance Name

GCRER2

Offset

0x80

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GCRER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 339: GPIO Bitwise Clear Rising Edge detect Enable Register (GCRER2)

Bits	Name	Type	Reset	Description
31:0	GCRER	W	0x0	0 : GRER bit Unaffected; 1: GRER Bit Clear

A.10.34 GPIO Bitwise Set Falling Edge detect Enable Register (GSFER0)

Instance Name

GSFER0

Offset

0x84

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GSFER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 340: GPIO Bitwise Set Falling Edge detect Enable Register (GSFER0)

Bits	Name	Type	Reset	Description
31:0	GSFER	W	0x0	0 : GFER bit Unaffected; 1: GFER Bit set

A.10.35 GPIO Bitwise set Falling Edge detect Enable Register (GSFER1)

Instance Name

GSFER1

Offset

0x88

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GSFER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 341: GPIO Bitwise set Falling Edge detect Enable Register (GSFER1)

Bits	Name	Type	Reset	Description
31:0	GSFER	W	0x0	0 : GFER bit Unaffected; 1: GFER Bit set

A.10.36 GPIO Bitwise set Falling Edge detect Enable Register (GSFER2)

Instance Name

GSFER2

Offset

0x8C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GSFER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 342: GPIO Bitwise set Falling Edge detect Enable Register (GSFER2)

Bits	Name	Type	Reset	Description
31:0	GSFER	W	0x0	0 : GFER bit Unaffected; 1: GFER Bit Clear

A.10.37 GPIO Bitwise Clear Falling Edge detect Enable Register (GCFER0)

Instance Name

GCFER0

Offset

0x90

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GCFER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 343: GPIO Bitwise Clear Falling Edge detect Enable Register (GCFER0)

Bits	Name	Type	Reset	Description
31:0	GCFER	W	0x0	0 : GFER bit Unaffected; 1: GFER Bit Clear

A.10.38 GPIO Bitwise Clear Falling Edge detect Enable Register (GCFER1)

Instance Name

GCFER1

Offset

0x94

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GCFER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 344: GPIO Bitwise Clear Falling Edge detect Enable Register (GCFER1)

Bits	Name	Type	Reset	Description
31:0	GCFER	W	0x0	0 : GFER bit Unaffected; 1: GFER Bit Clear

A.10.39 GPIO Bitwise Clear Falling Edge detect Enable Register (GCFER2)

Instance Name

GCFER2

Offset

0x98

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GCFER																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 345: GPIO Bitwise Clear Falling Edge detect Enable Register (GCFER2)

Bits	Name	Type	Reset	Description
31:0	GCFER	W	0x0	0 : GFER bit Unaffected; 1: GFER Bit Clear

A.10.40 GPIO Bitwise mask of Edge detect Status Register (APMASK0)

Instance Name

APMASK0

Offset

0x9C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	APMASK																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 346: GPIO Bitwise mask of Edge detect Status Register (APMASK0)

Bits	Name	Type	Reset	Description
31:0	APMASK	R/W	0x0	0 : GPIO Edge detects are masked ; 1: GPIO Edge detects are not masked

A.10.41 GPIO Bitwise mask of Edge detect Status Register (APMASK1)

Instance Name

APMASK1

Offset

0xA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	APMASK																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 347: GPIO Bitwise mask of Edge detect Status Register (APMASK1)

Bits	Name	Type	Reset	Description
31:0	APMASK	R/W	0x0	0 : GPIO Edge detects are masked ; 1: GPIO Edge detects are not masked

A.10.42 GPIO Bitwise mask of Edge detect Status Register (APMASK2)

Instance Name

APMASK2

Offset

0xA4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	APMASK																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 348: GPIO Bitwise mask of Edge detect Status Register (APMASK2)

Bits	Name	Type	Reset	Description
31:0	APMASK	R/W	0x0	0 : GPIO Edge detects are masked ; 1: GPIO Edge detects are not masked

A.11 GPT Register Information

Table 349: GPT Register Summary

Offset	Name	Description	Details
0x000	CNT_EN	Counter Enable Register	Page: 259
0x020	STS	Status Register	Page: 260
0x024	INT	Interrupt Register	Page: 261
0x028	INT_MSK	Interrupt Mask Register	Page: 263
0x040	CNT_CNTL	Counter Control Register	Page: 264
0x050	CNT_VAL	Counter Value Register	Page: 265
0x060	CNT_UPP_VAL	Counter Upper Value Register	Page: 265
0x080	CLK_CNTL	Clock Control Register	Page: 266
0x088	IC_CNTL	Input Capture Control Register	Page: 266
0x0A0	DMA_CNTL_EN	DMA Control Enable Register	Page: 267
0x0A4	DMA_CNTL_CH	DMA Control Channel Register	Page: 268
0x0D0	TCR	ADC/DAC Trigger Control Register	Page: 268
0x0D8	TDR	ADC/DAC Trigger Delay Register	Page: 269
0x0F0	USER_REQ	User Request Register	Page: 270
0x200	CH0_CNTL	Channel 0 Control Register	Page: 272
0x210	CH0_CMRO	Channel 0 Counter Match Register 0	Page: 273
0x220	CH0_CMRI	Channel 0 Counter Match Register 1	Page: 273
0x240	CH1_CNTL	Channel 1 Control Register	Page: 272
0x250	CH1_CMRO	Channel 1 Counter Match Register 0	Page: 273
0x260	CH1_CMRI	Channel 1 Counter Match Register 1	Page: 273
0x280	CH2_CNTL	Channel 2 Control Register	Page: 272
0x290	CH2_CMRO	Channel 2 Counter Match Register 0	Page: 273
0x2A0	CH2_CMRI	Channel 2 Counter Match Register 1	Page: 273
0x2C0	CH3_CNTL	Channel 3 Control Register	Page: 272
0x2D0	CH3_CMRO	Channel 3 Counter Match Register 0	Page: 273
0x2E0	CH3_CMRI	Channel 3 Counter Match Register 1	Page: 273
0x300	CH4_CNTL	Channel 4 Control Register	Page: 272
0x310	CH4_CMRO	Channel 4 Counter Match Register 0	Page: 273

Table 349: GPT Register Summary

Offset	Name	Description	Details
0x320	CH4_CMR1	Channel 4 Counter Match Register 1	Page: 273
0x340	CH5_CNTL	Channel 5 Control Register	Page: 272
0x350	CH5_CMR0	Channel 5 Counter Match Register 0	Page: 273
0x360	CH5_CMR1	Channel 5 Counter Match Register 1	Page: 273

A.11.1 Counter Enable Register (CNT_EN)

Instance Name

CNT_EN

Offset

0x000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved													STS_RESETN	CNT_RST_DONE	CNT_RUN	Reserved										CNT_RESET	CNT_STOP	CNT_START			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Table 350: Counter Enable Register (CNT_EN)

Bits	Name	Type	Reset	Description
31:19	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
18	STS_RESETN	R	0x0	System Reset Status CPU must poll this bit for a 1 before accessing any other registers. 0x0: The system reset is still asserted 0x1: The system reset is deasserted
17	CNT_RST_DONE	R	0x0	Counter Reset Done Status Writing 1 to CNT_RESET will set this bit to 0 until the counter finishes resetting. 0x0: The counter is still resetting 0x1: The counter has been reset
16	CNT_RUN	R	0x0	Counter Enabled Status This bit can be polled to see when the counter is really enabled. 0x0: Counter is disabled 0x1: Counter is enabled
15:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 350: Counter Enable Register (CNT_EN)

Bits	Name	Type	Reset	Description
2	CNT_RESET	W	0x0	Counter Reset 0x0: No action 0x1: Reset the counter. Counter is reset to 0. Channel output states are also reset. Poll CNT_RST_DONE to see when the counter has been reset. Do not write to any other registers before CNT_RST_DONE turns to 1.
1	CNT_STOP	W	0x0	Counter Stop 0x0: No action 0x1: Disable the counter. Poll CNT_RUN to see the counter status. if CNT_RUN is 0, it means that the counter has been disabled internally.
0	CNT_START	W	0x0	Counter Start 0x0: No action 0x1: Enable the counter. Poll CNT_RUN to see the counter status. if CNT_RUN is 1, it means that the counter has been enabled internally.

A.11.2 Status Register (STS)

Instance Name
STS

Offset
0x020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved						DMA1_OF_STS.	DMA0_OF_STS.	Reserved								CNT_UPP_STS	Reserved		CH5_ERR_STS	CH4_ERR_STS	CH3_ERR_STS	CH2_ERR_STS	CH1_ERR_STS	CH0_ERR_STS	Reserved		CH5_STS	CH4_STS	CH3_STS	CH2_STS	CH1_STS	CH0_STS
Default	?	?	?	?	?	?	0	0	?	?	?	?	?	?	?	0	?	?	0	0	0	0	0	0	?	?	0	0	0	0	0	0	0

Table 351: Status Register (STS)

Bits	Name	Type	Reset	Description
31:26	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
25	DMA1_OF_STS.	R/W1CLR	0x0	See DMA0_OF_STS.
24	DMA0_OF_STS.	R/W1CLR	0x0	DMA Channel Overflow Status 0x0: Status cleared 0x1: Indicates that there has been a new input capture before this DMA channel could transfer the captured data away.
23:17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 351: Status Register (STS)

Bits	Name	Type	Reset	Description
16	CNT_UPP_STS	R/W1CLR	0x0	Counter-Reach-Upper Status Indicates that the counter has reached UPP_VAL. 0x0: Status cleared 0x1: The counter has reached UPP_VAL
15:14	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
13	CH5_ERR_STS	R/W1CLR	0x0	See CH0_ERR_STS.
12	CH4_ERR_STS	R/W1CLR	0x0	See CH0_ERR_STS.
11	CH3_ERR_STS	R/W1CLR	0x0	See CH0_ERR_STS.
10	CH2_ERR_STS	R/W1CLR	0x0	See CH0_ERR_STS.
9	CH1_ERR_STS	R/W1CLR	0x0	See CH0_ERR_STS.
8	CH0_ERR_STS	R/W1CLR	0x0	Channel Error Status This bit will be set when an error occurs during the update of CMR0 and CMR1. Please refer to the description of CH0_CMR_UPDT for details. 0x0: Status cleared 0x1: An error has occurred in this channel
7:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5	CH5_STS	R/W1CLR	0x0	See CH0_STS.
4	CH4_STS	R/W1CLR	0x0	See CH0_STS.
3	CH3_STS	R/W1CLR	0x0	See CH0_STS.
2	CH2_STS	R/W1CLR	0x0	See CH0_STS.
1	CH1_STS	R/W1CLR	0x0	See CH0_STS.
0	CH0_STS	R/W1CLR	0x0	Channel Status If this channel is in input-capture mode, this bit will be set when an edge is captured. If this channel is in PWM or one-shot mode, the value of CMR0 determines when this bit is set. For PWM mode, this bit will be set CMR0 counter ticks after the starting point of each period. This bit will be set at each period. For one-shot modes, the bit will be set CMR0 counter ticks after the starting point. This bit will be set only once. 0x0: Status cleared 0x1: Status bit for this channel has been set

A.11.3 Interrupt Register (INT)

INT_MSK is combined with STS to determine the value of this register. Masked bits will always be 0, while unmasked bits will be the same value as the corresponding bits in STS.

If any bits in this register is 1, an interrupt will be generated.

Instance Name
Offset

INT

0x024

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved						DMA1_OF_INTR	DMA0_OF_INTR	Reserved								CNT_UPP_INTR	Reserved		CH5_ERR_INTR	CH4_ERR_INTR	CH3_ERR_INTR	CH2_ERR_INTR	CH1_ERR_INTR	CH0_ERR_INTR	Reserved		CH5_INTR	CH4_INTR	CH3_INTR	CH2_INTR	CH1_INTR	CH0_INTR
Default	?	?	?	?	?	?	0	0	?	?	?	?	?	?	?	0	?	?	0	0	0	0	0	0	?	?	0	0	0	0	0	0	0

Table 352: Interrupt Register (INT)

Bits	Name	Type	Reset	Description
31:26	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
25	DMA1_OF_INTR	R	0x0	Masked signal of DMA1_OF_STS.
24	DMA0_OF_INTR	R	0x0	Masked signal of DMA0_OF_STS.
23:17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16	CNT_UPP_INTR	R	0x0	Masked signal of CNT_UPP_STS.
15:14	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
13	CH5_ERR_INTR	R	0x0	Masked signal of CH5_ERR_STS.
12	CH4_ERR_INTR	R	0x0	Masked signal of CH4_ERR_STS.
11	CH3_ERR_INTR	R	0x0	Masked signal of CH3_ERR_STS.
10	CH2_ERR_INTR	R	0x0	Masked signal of CH2_ERR_STS.
9	CH1_ERR_INTR	R	0x0	Masked signal of CH1_ERR_STS.
8	CH0_ERR_INTR	R	0x0	Masked signal of CH0_ERR_STS.
7:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5	CH5_INTR	R	0x0	Masked signal of CH5_STS.
4	CH4_INTR	R	0x0	Masked signal of CH4_STS.
3	CH3_INTR	R	0x0	Masked signal of CH3_STS.
2	CH2_INTR	R	0x0	Masked signal of CH2_STS.
1	CH1_INTR	R	0x0	Masked signal of CH1_STS.
0	CH0_INTR	R	0x0	Masked signal of CH0_STS.

A.11.4 Interrupt Mask Register (INT_MSK)

Instance Name

INT_MSK

Offset

0x028

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved						DMA1_OF_MSK	DMA0_OF_MSK	Reserved								CNT_UPP_MSK	Reserved		CH5_ERR_MSK	CH4_ERR_MSK	CH3_ERR_MSK	CH2_ERR_MSK	CH1_ERR_MSK	CH0_ERR_MSK	Reserved		CH5_MSK	CH4_MSK	CH3_MSK	CH2_MSK	CH1_MSK	CH0_MSK
Default	?	?	?	?	?	?	1	1	?	?	?	?	?	?	?	1	?	?	1	1	1	1	1	1	?	?	1	1	1	1	1	1	

Table 353: Interrupt Mask Register (INT_MSK)

Bits	Name	Type	Reset	Description
31:26	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
25	DMA1_OF_MSK	R/W	0x1	See DMA0_OF_MSK
24	DMA0_OF_MSK	R/W	0x1	DMA Channel Overflow Mask 0x0: Do not mask DMA0_OF_STS 0x1: Mask DMA0_OF_STS
23:17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16	CNT_UPP_MSK	R/W	0x1	Upper Value Interrupt Mask 0x0: Do not mask CNT_UPP_STS 0x1: Mask CNT_UPP_STS
15:14	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
13	CH5_ERR_MSK	R/W	0x1	See CH0_ERR_STS.
12	CH4_ERR_MSK	R/W	0x1	See CH0_ERR_STS.
11	CH3_ERR_MSK	R/W	0x1	See CH0_ERR_STS.
10	CH2_ERR_MSK	R/W	0x1	See CH0_ERR_STS.
9	CH1_ERR_MSK	R/W	0x1	See CH0_ERR_STS.
8	CH0_ERR_MSK	R/W	0x1	Channel Error Interrupt Mask 0x0: Do not mask CH0_ERR_STS 0x1: Mask CH0_ERR_STS
7:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5	CH5_MSK	R/W	0x1	See CH0_STS.
4	CH4_MSK	R/W	0x1	See CH0_STS.
3	CH3_MSK	R/W	0x1	See CH0_STS.
2	CH2_MSK	R/W	0x1	See CH0_STS.
1	CH1_MSK	R/W	0x1	See CH0_STS.

Table 353: Interrupt Mask Register (INT_MSK)

Bits	Name	Type	Reset	Description
0	CH0_MSK	R/W	0x1	Channel Interrupt Mask 0x0: Do not mask CH0_STS 0x1: Mask CH0_STS

A.11.5 Counter Control Register (CNT_CNTL)

Instance Name

CNT_CNTL

Offset

0x040

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																						CNT_UPDT_MOD	Reserved			CNT_DBG_ACT	Reserved				
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	0	?	?	?	?

Table 354: Counter Control Register (CNT_CNTL)

Bits	Name	Type	Reset	Description
31:10	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
9:8	CNT_UPDT_MOD	R/W	0x0	Counter Value Update Mode 0x0: Auto-update normal. Can be used for any clock relationship between the counter clock and the APB clock. Only every 3-4 counter ticks are updated to CNT_VAL. 0x1: Auto-update fast. Used when counter clock is at least 5 times slower than the APB clock. Every counter tick is updated to CNT_VAL. 0x2: Reserved 0x3: Update off. If CNT_VAL does not need to be read, CNT_UPDT_MOD can be set to off to save power.
7:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	CNT_DBG_ACT	R/W	0x0	Counter Action in Debug Mode 0x0: In debug mode, stop the counter 0x1: In debug mode, the counter is not affected
3:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.11.6 Counter Value Register (CNT_VAL)

Instance Name
CNT_VAL

Offset
0x050

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	CNT_VAL																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 355: Counter Value Register (CNT_VAL)

Bits	Name	Type	Reset	Description
31:0	CNT_VAL	R	0x0	Counter Value This register is used to view the current value of the main counter. The update mode of this register is based on CNT_UPDT_MOD.

A.11.7 Counter Upper Value Register (CNT_UPP_VAL)

Instance Name
CNT_UPP_VAL

Offset
0x060

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	UPP_VAL																															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Table 356: Counter Upper Value Register (CNT_UPP_VAL)

Bits	Name	Type	Reset	Description
31:0	UPP_VAL	R/W	0xFFFF_FFFF	Counter Upper Value Do not set it to 0. The reset value is the maximum value of the counter, where all bits are 1. In the event that the counter reaches this value (counter-reach-upper), the counter will overflow to 0. Setting this value to all 1s is equivalent to a free running up-counter. The value written to UPP_VAL will not be valid immediately. It will not be effective until the counter overflows. If you need the value is valid immediately, you can write 1 to CNT_RESET.

A.11.8 Clock Control Register (CLK_CNTL)

Instance Name Offset
CLK_CNTL 0x080

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved								CLK_PRE								Reserved				CLK_DIV				Reserved								CLK_SRC
Default	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	?	?	?	?	0	0	0	0	?	?	?	?	?	?	?	0	

Table 357: Clock Control Register (CLK_CNTL)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23:16	CLK_PRE	R/W	0x0	Clock Pre-Scalar This can be used together with CLK_DIV. The frequency of the prescaled clock (f_pre) is calculated from the frequency of the counter clock (f_clk) using this formula: $f_{pre} = f_{clk} / (CLK\_PRE + 1)$
15:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11:8	CLK_DIV	R/W	0x0	Clock Divider This can be used together with CLK_PRE. The frequency after the clock divider is calculated using this formula: $f_{div} = f_{pre} / (2 ^ CLK\_DIV)$
7:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	CLK_SRC	R/W	0x0	Counter Clock Select 0x0: Select clock 0 0x1: Select clock 1

A.11.9 Input Capture Control Register (IC_CNTL)

Instance Name Offset
IC_CNTL 0x088

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																									IC_DIV			Reserved	IC_WIDTH		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0

Table 358: Input Capture Control Register (IC_CNTL)

Bits	Name	Type	Reset	Description
31:7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 358: Input Capture Control Register (IC_CNTL)

Bits	Name	Type	Reset	Description
6:4	IC_DIV	R/W	0x0	Input Capture Sampling Clock Divider This field divides the sampling clock used to sample the input trigger. The frequency of the divided sampling clock (f_sdiv) is calculated from the frequency of the sampling clock (f_sclk) using the following formula: $f_{sdiv} = f_{sclk} / (2 \wedge IC\_DIV)$
3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	IC_WIDTH	R/W	0x0	Input Capture Filter Width It indicates how many consecutive cycles the input trigger signal must be sampled for before it is considered as a valid edge. Any glitch pulse shorter than it will be filtered. 0x0: No filtering (1 cycle) 0x1: 2 cycles 0x2: 3 cycles 0x3: 4 cycles 0x4: 5 cycles 0x5: 6 cycles 0x6: 7 cycles 0x7: Reserved

A.11.10 DMA Control Enable Register (DMA_CNTL_EN)

Instance Name Offset
DMA_CNTL_EN 0x0A0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														DMA1_EN	DMA0_EN
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 359: DMA Control Enable Register (DMA_CNTL_EN)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	DMA1_EN	R/W	0x0	See DMA0_EN.
0	DMA0_EN	R/W	0x0	DMA Channel Enable 0x0: Disable this DMA channel 0x1: Enable this DMA channel. In input capture mode, DMA controller will be notified when a value is captured.

A.11.11 DMA Control Channel Register (DMA_CNTL_CH)

Instance Name
DMA_CNTL_CH

Offset
0x0A4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																									DMA1_CH		Reserved	DMA0_CH			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	0	0	0

Table 360: DMA Control Channel Register (DMA_CNTL_CH)

Bits	Name	Type	Reset	Description
31:7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
6:4	DMA1_CH	R/W	0x0	See DMA0_CH.
3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	DMA0_CH	R/W	0x0	DMA Channel Select Select the GPT channel, to which this DMA channel is connected. 0x0: Connected to GPT channel 0 0x1: Connected to GPT channel 1 0x2: Connected to GPT channel 2 0x3: Connected to GPT channel 3 0x4: Connected to GPT channel 4 0x5: Connected to GPT channel 5 others: Reserved

A.11.12 ADC/DAC Trigger Control Register (TCR)

Instance Name
TCR

Offset
0x0D0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																							TRIG_EN	Reserved					TRIG_CHSEL		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?	?	?	?	0	0	0

Table 361: ADC/DAC Trigger Control Register (TCR)

Bits	Name	Type	Reset	Description
31:9	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 361: ADC/DAC Trigger Control Register (TCR)

Bits	Name	Type	Reset	Description
8	TRIG_EN	R/W	0x0	ADC/DAC Trigger Enable. The ADC trigger is in GPT0 and GPT1. The DAC trigger is in GPT2 and GPT3. 0x0: Disable the ADC/DAC trigger 0x1: Enable the ADC/DAC trigger
7:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	TRIG_CHSEL	R/W	0x0	ADC/DAC Trigger Channel Select. The ADC trigger is in GPT0 and GPT1. The DAC trigger is in GPT2 and GPT3. Select which GPT channel is used to generate the ADC/ DAC trigger signal. 0x0: GPT channel 0 0x1: GPT channel 1 0x2: GPT channel 2 0x3: GPT channel 3 0x4: GPT channel 4 0x5: GPT channel 5 others: Reserved

A.11.13 ADC/DAC Trigger Delay Register (TDR)

Instance Name

Offset

TDR

0x0D8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TRIG_DLY																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 362: ADC/DAC Trigger Delay Register (TDR)

Bits	Name	Type	Reset	Description
31:0	TRIG_DLY	R/W	0x0	ADC Trigger Delay. The ADC trigger is in GPT0 and GPT1. The DAC trigger is in GPT2 and GPT3. At the end of each PWM period, after the effective ADC/ DAC trigger delay (adly_eff), a trigger will be generated to signal the ADC/DAC to begin a conversion. adly_eff is 4 times TRIG_DLY, as shown below: $adly\_eff = TRIG\_DLY \times 4$ For the ADC/DAC trigger to work properly, adly_eff must be shorter than the PWM period. The PWM period must be longer than the ADC/DAC conversion period, with appropriate margins.

A.11.14 User Request Register (USER_REQ)

Instance Name

USER_REQ

Offset

0x0F0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved										CH5_CMUR_UPDT	CH4_CMUR_UPDT	CH3_CMUR_UPDT	CH2_CMUR_UPDT	CH1_CMUR_UPDT	CH0_CMUR_UPDT	Reserved		CH5_RST	CH4_RST	CH3_RST	CH2_RST	CH1_RST	CH0_RST	Reserved		CH5_USER_ITRIG	CH4_USER_ITRIG	CH3_USER_ITRIG	CH2_USER_ITRIG	CH1_USER_ITRIG	CH0_USER_ITRIG
Default	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	?	?	0	0	0	0	0	0	?	?	0	0	0	0	0	0

Table 363: User Request Register (USER_REQ)

Bits	Name	Type	Reset	Description
31:22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21	CH5_CMUR_UPDT	W	0x0	See CH0_CMUR_UPDT.
20	CH4_CMUR_UPDT	W	0x0	See CH0_CMUR_UPDT.
19	CH3_CMUR_UPDT	W	0x0	See CH0_CMUR_UPDT.
18	CH2_CMUR_UPDT	W	0x0	See CH0_CMUR_UPDT.
17	CH1_CMUR_UPDT	W	0x0	See CH0_CMUR_UPDT.
16	CH0_CMUR_UPDT	W	0x0	Channel CMUR Update To update the values of CMR0 and/or CMR1, you need to write the values to CMR0 and CMR1, and then write 1 to this field. Then CMR0 and CMR1 starts to be updated. This is useful for the case that the CMR0 and CMR1 need to be updated at the same time. In PWM mode, the update will not be finished until the end of a PWM period. In one-shot mode, the update will not be finished until this channel is idle. In any other channel mode, the update of CMR0 and CMR1 will be finished immediately after writing 1 to this field. In any channel mode, the CMR update will be completed immediately if there is a counter or channel reset. Writing to this field again before the first update is done will cause the second update to be discarded, and CH0_ERR_STS will be set. 0x0: No action 0x1: Update CMR0 and CMR1.
15:14	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
13	CH5_RST	W	0x0	See CH0_RST.
12	CH4_RST	W	0x0	See CH0_RST.
11	CH3_RST	W	0x0	See CH0_RST.
10	CH2_RST	W	0x0	See CH0_RST.

Table 363: User Request Register (USER_REQ)

Bits	Name	Type	Reset	Description
9	CH1_RST	W	0x0	See CH0_RST.
8	CH0_RST	W	0x0	Channel Reset Writing 1 to this register will reset the channel and kick start the corresponding mode determined by CHx_IO. Only write to this field when CNT_RUN is set. In One-shot pulse mode, the output state will be reset, and a pulse will occur. In One-shot edge mode, an edge transition will occur, but the output state will NOT be reset. In all other modes, the output state will be reset. Resetting multiple channels in the same APB write will synchronize the start periods of the PWM and One-shot outputs. 0x0: No action 0x1: Reset this channel
7:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5	CH5_USER_ITRIG	W	0x0	See CH0_USER_ITRIG.
4	CH4_USER_ITRIG	W	0x0	See CH0_USER_ITRIG.
3	CH3_USER_ITRIG	W	0x0	See CH0_USER_ITRIG.
2	CH2_USER_ITRIG	W	0x0	See CH0_USER_ITRIG.
1	CH1_USER_ITRIG	W	0x0	See CH0_USER_ITRIG.
0	CH0_USER_ITRIG	W	0x0	User Input Trigger 0x0: No action 0x1: If this channel is configured to input-capture mode, generate a manual input trigger to capture the current counter value. This trigger bypasses the input capture control register settings.

A.11.15 Channel x Control Register (CHX_CNTL)

Instance Name	Offset
CH0_CNTL	0x200
CH1_CNTL	0x240
CH2_CNTL	0x280
CH3_CNTL	0x2C0
CH4_CNTL	0x300
CH5_CNTL	0x340

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved															POL	Reserved	IC_EDGE			Reserved										CH_IO		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	0	0	0	?	?	?	?	?	?	?	?	?	?	0	0	0

Table 364: Channel x Control Register (CHX_CNTL)

Bits	Name	Type	Reset	Description
31:17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16	POL	R/W	0x0	Channel Polarity Default output state of this channel after reset. This field determines the waveform polarity in PWM modes and one-shot pulse mode. 0x0: Reset to 0 0x1: Reset to 1
15	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
14:12	IC_EDGE	R/W	0x0	Channel Input Capture Edge Determines which edge of the input trigger to capture. 0x0: Capture rising edge in CMR0 0x1: Capture falling edge in CMR0
11:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	CH_IO	R/W	0x0	Channel Operation Mode 0x0: No function 0x1: Input capture mode 0x4: One-shot mode (pulse) 0x5: One-shot mode (edge) 0x6: PWM mode (edge-aligned) 0x7: PWM mode (center-aligned) others: Reserved

A.11.16 Channel x Counter Match Register 0 (CHX_CMRO)

Instance Name	Offset
CH0_CMRO	0x210
CH1_CMRO	0x250
CH2_CMRO	0x290
CH3_CMRO	0x2D0
CH4_CMRO	0x310
CH5_CMRO	0x350

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	CMR0																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 365: Channel x Counter Match Register 0 (CHX_CMRO)

Bits	Name	Type	Reset	Description
31:0	CMR0	R/W	0x0	Channel x Counter Match Register 0 In order to make the values in CMR0 and CMR1 valid, write 1 to CHx_CMRO_UPDT to start the update of CMR0 and CMR1. Please refer to the description of CHx_CMRO_UPDT for details. In any PWM mode, setting both CMR0 and CMR1 to 0 will result in a degenerate case that shuts off the PWM generator. To reset the PWM, set at least one of CMR0 or CMR1 to a non-zero value and write to CHX_RST.

A.11.17 Channel x Counter Match Register 1 (CHX_CMRI)

Instance Name	Offset
CH0_CMRI	0x220
CH1_CMRI	0x260
CH2_CMRI	0x2A0
CH3_CMRI	0x2E0
CH4_CMRI	0x320
CH5_CMRI	0x360

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	CMR1																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 366: Channel x Counter Match Register 1 (CHX_CM1)

Bits	Name	Type	Reset	Description
31:0	CMR1	R/W	0x0	Channel x Counter Match Register 1 In order to make the values in CMR0 and CMR1 valid, write 1 to CHx_CM1_UPDT to start the update of CMR0 and CMR1. Please refer to the description of CHx_CM1_UPDT for details. In any PWM mode, setting both CMR0 and CMR1 to 0 will result in a degenerate case that shuts off the PWM generator. To reset the PWM, set at least one of CMR0 or CMR1 to a non-zero value and write to CHx_RST.

A.12 RC32M Register Information

Table 367: RC32M Register Summary

Offset	Name	Description	Details
0x00	CTRL	Control Register	Page: 275
0x04	STATUS	Status Register	Page: 276
0x08	ISR	Interrupt Status Register	Page: 276
0x0C	IMR	Interrupt Mask Register	Page: 277
0x10	IRSR	Interrupt Raw Status Register	Page: 277
0x14	ICR	Interrupt Clear Register	Page: 278
0x18	CLK	Clock Register	Page: 278
0x1C	RST	Soft Reset Register	Page: 279
0x20	RESERVED	Reserved	Page: 279
0x24	RESERVED	Reserved	Page: 280

A.12.1 Control Register (CTRL)

Instance Name
CTRLOffset
0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																			Reserved		CODE_FR_EXT							PD	EXT_CODE_EN	CAL_EN	EN
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	1	1	1	1	1	1	0	0	0	1

Table 368: Control Register (CTRL)

Bits	Name	Type	Reset	Description
31:13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12:11	Reserved	R/W	0x0	Reserved. Do not change the reset value.
10:4	CODE_FR_EXT	R/W	0x3F	External code input for calibration.
3	PD	R/W	0x0	Clock Power down 0x1: Power down 0x0: Power up.
2	EXT_CODE_EN	R/W	0x0	Calibration code from external enable 0x1: calibration code from external 0x0: calibration code from internal

Table 368: Control Register (CTRL)

Bits	Name	Type	Reset	Description
1	CAL_EN	R/W	0x0	Calibration enable 0x1: Enable 0x0: Disable
0	EN	R/W	0x1	Calibration reference clock enable 0x1: Enable 0x0: Disable

A.12.2 Status Register (STATUS)

Instance Name
STATUS

Offset
0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																						CODE_FR_CAL						CAL_DONE	CLK_RDY		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0

Table 369: Status Register (STATUS)

Bits	Name	Type	Reset	Description
31:9	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
8:2	CODE_FR_CAL	R	0x0	Calibration code
1	CAL_DONE	R	0x0	Calibration finish flag 0x1: calibration done 0x0: calibration not done
0	CLK_RDY	R	0x0	1 indicate whether RC32M clock is ready

A.12.3 Interrupt Status Register (ISR)

Instance Name
ISR

Offset
0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														CKRDY_INT	CALDON_INT
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 370: Interrupt Status Register (ISR)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	CKRDY_INT	R	0x0	1 indicates clock out ready interrupt flag if ckrdy_int_raw is not masked
0	CALDON_INT	R	0x0	1 indicates calibration done interrupt flag if caldon_int_raw is not masked

A.12.4 Interrupt Mask Register (IMR)

Instance Name

IMR

Offset

0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														CKRDY_INT_MSK	CALDON_INT_MSK
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1

Table 371: Interrupt Mask Register (IMR)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	CKRDY_INT_MSK	R/W	0x1	Set to 1 to mask off clk ready interrupt
0	CALDON_INT_MSK	R/W	0x1	Set to 1 to mask off cal done interrupt

A.12.5 Interrupt Raw Status Register (IRSR)

Instance Name

IRSR

Offset

0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														CKRDY_INT_RAW	CALDON_INT_RAW
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 372: Interrupt Raw Status Register (IRSR)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	CKRDY_INT_RAW	R	0x0	1 indicates clock out ready interrupt flag regardless the mask
0	CALDON_INT_RAW	R	0x0	1 indicates calibration done interrupt flag regardless the mask

A.12.6 Interrupt Clear Register (ICR)

Instance Name

ICR

Offset

0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														CKRDY_INT_CLR	CALDON_INT_CLR
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 373: Interrupt Clear Register (ICR)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	CKRDY_INT_CLR	R/W	0x0	Write to 1 to clean clk ready interrupt flag. By setting 1 to this bit, on next clock active edge, it will be cleared to 0 automatically.
0	CALDON_INT_CLR	R/W	0x0	Write to 1 to clean cal done interrupt flag. By setting 1 to this bit, on next clock active edge, it will be cleared to 0 automatically.

A.12.7 Clock Register (CLK)

Instance Name

CLK

Offset

0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																												SOFT_CLK_RST	SEL_32M	Reserved	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	0	0

Table 374: Clock Register (CLK)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	SOFT_CLK_RST	R/W	0x0	Soft reset for clock divider 0x1: Reset 0x0: No action
2	SEL_32M	R/W	0x1	Output clock frequency select. 0x1: 32M 0x0: 16M
1:0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.12.8 Soft Reset Register (RST)

Instance Name

RST

Offset

0x1C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														SOFT_RST	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 375: Soft Reset Register (RST)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	SOFT_RST	R/W	0x0	Soft reset for module, active high 0x1: Reset 0x0: No action

A.12.9 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																												Reserved			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		?	0	0

Table 376: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.12.10 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x24

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																Reserved															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Table 377: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	Reserved	R/W	0xFF00	Reserved. Do not change the reset value.

A.13 ADC Register Information

Table 378: ADC Register Summary

Offset	Name	Description	Details
0x00	PWR	Power Enable Register	Page: 281
0x04	CLKRST	Clock Control Register	Page: 282
0x08	CMD	Command Register	Page: 283
0x0C	INTERVAL	Time Interval Register	Page: 283
0x10	ANA	Analog Configuration Register	Page: 284
0x14	DMAR	DMA Control Register	Page: 287
0x18	RESERVED	Reserved	Page: 287
0x1C	STATUS	Status Register	Page: 288
0x20	ISR	Interrupt Status Register	Page: 288
0x24	IMR	Interrupt Mask Register	Page: 289
0x28	IRSR	Interrupt Raw Status Register	Page: 290
0x2C	ICR	Interrupt Clear Register	Page: 291
0x30	RESULT	Final Data Register	Page: 291
0x34	RESERVED	Reserved	Page: 292
0x38	OFF_CAL	Offset Calibration Data Register	Page: 292
0x3C	GAIN_CAL	Gain Calibration Data Register	Page: 292
0x40	RESERVED	Reserved	Page: 293
0x44	RESERVED	Reserved	Page: 293

A.13.1 Power Enable Register (PWR)

Instance Name

PWR

Offset

0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															GLOBAL_EN
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 379: Power Enable Register (PWR)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	GLOBAL_EN	R/W	0x0	Global enable bit. 0x0: ADC is fully powered off. 0x1: ADC is powered on - waiting for the conv_start or event trigger signal to start data conversion.

A.13.2 Clock Control Register (CLKRST)

Instance Name
CLKRST

Offset
0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																						INT_CLK_DIV					Reserved			SOFT_CLK_RST		SOFT_RST
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	1	1	1	1	1	0	0	0	

Table 380: Clock Control Register (CLKRST)

Bits	Name	Type	Reset	Description
31:10	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
9:5	INT_CLK_DIV	R/W	0x1F	Clock divider set. Select ADC internal operating frequency. 0x1F: divide-by-32 (1MHz if main clock is 32MHz). 0x0F: divide-by-16 (2MHz if main clock is 32MHz). 0x07: divide-by-8 (4MHz if main clock is 32MHz). 0x03: divide-by-4 (8MHz if main clock is 32MHz). others: reserved
4:2	Reserved	R/W	0x6	Reserved. Do not change the reset value.
1	SOFT_CLK_RST	R/W	0x0	Soft reset for clk divider. Active high. 0x0: normal 0x1: reset
0	SOFT_RST	R/W	0x0	ADC asynchronous reset signal. Will reset FSM and data registers in ADC without affecting APB register settings. Active high. 0x0: normal 0x1: reset

A.13.3 Command Register (CMD)

Instance Name
CMD

Offset
0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											PWR_MODE	TRIGGER_SEL	TRIGGER_EN	CONV_START	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	0	0	0	0

Table 381: Command Register (CMD)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	PWR_MODE	R/W	0x1	Power mode select bits. 0x0: power mode 0 - Analog biasing and reference block are powered up when conv_start is 1. 0x1: power mode 1 - Analog biasing and reference block are powered up once global_en is 1.
3:2	TRIGGER_SEL	R/W	0x0	Trigger source select bits. 0x0: ADC trigger source 0 (GPT0 for ADC0 and GPT1 for ADC1) 0x1: ADC trigger source 1 (reserved) 0x2: ADC trigger source 2 (reserved) 0x3: ADC trigger source 3 (reserved)
1	TRIGGER_EN	R/W	0x0	Event trigger enable bit. 0x0: conversion start controlled by software 0x1: conversion start controlled by event triggered level signal
0	CONV_START	R/W	0x0	Conversion start enable bit. 0x0: stop data conversion 0x1: start data conversion

A.13.4 Time Interval Register (INTERVAL)

Instance Name
INTERVAL

Offset
0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															Reserved										WARMUP_TIME						
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	1	1	1	1	0	1	1					1	1	0

Table 382: Time Interval Register (INTERVAL)

Bits	Name	Type	Reset	Description
31:17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16:5	Reserved	R/W	0x1EF	Reserved. Do not change the reset value.
4:0	WARMUP_TIME	R/W	0xF	Warm up time. Set warm up time period to (warmup_time + 1)us for ADC warm up sequence according to the 1MHz frequency. 0x00: ADC warm up time is 1us 0x1F: ADC warm up time is 32us others: ADC warm up time is (warmup_time + 1) us

A.13.5 Analog Configuration Register (ANA)

Instance Name

ANA

Offset

0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												IN_BFSEL	VREF_BFSEL	PGA		BIAS_SEL	OSR		SINGLEDIFF	AMUX_SEL				VREF_SEL		Reserved		CAL	TS_EN	EXT_SEL	Reserved
Default	?	?	?	?	?	?	?	?	?	?	?	?	1	1	0	1	0	1	1	0	0	0	0	0	1	1	1	1	0	0	0	1

Table 383: Analog Configuration Register (ANA)

Bits	Name	Type	Reset	Description
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19	IN_BFSEL	R/W	0x1	Input gain buffer enable bit. 0x0: input gain buffer disabled 0x1: input gain buffer enabled
18	VREF_BFSEL	R/W	0x1	Voltage reference buffer enable bit. 0x0: reference buffer disabled 0x1: reference buffer enabled
17:16	PGA	R/W	0x1	Gain control bits. This field also selects input voltage range. 0x0: The on-chip PGA gain is 0.5, input voltage range is 2 * Vref. 0x1: The on-chip PGA gain is 1, input voltage range is Vref. 0x2: The on-chip PGA gain is 2, input voltage range is 0.5 * Vref. 0x3: reserved
15	BIAS_SEL	R/W	0x0	Analog portion low power mode select. 0x0: full biasing current for modulator 0x1: half biasing current for modulator

Table 383: Analog Configuration Register (ANA)

Bits	Name	Type	Reset	Description
14:13	OSR	R/W	0x3	Sample time select bits. ADC sample time in clocks and the equivalent significant bits in the conversions. 0x0: 32 decimation rate (10-bit) 0x1: 64 decimation rate (12-bit) 0x2: 128 decimation rate (14-bit) 0x3: 256 decimation rate (16-bit)
12	SINGLEDIFF	R/W	0x0	Select single-ended or differential input. 0x0: single-ended input 0x1: differential input

Table 383: Analog Configuration Register (ANA)

Bits	Name	Type	Reset	Description
11:8	AMUX_SEL	R/W	0x0	Input channel select bits. Selects the channel number for 14 single-ended inputs and 6 pairs of differential inputs. 0x0: singlediff = 0: ADC_CH[0] (GPIO_7 for ADC0 and GPIO_8 for ADC1) and VSSA singlediff = 1: ADC_CH[0] (GPIO_7 for ADC0 and GPIO_8 for ADC1, positive input) and ADC_CH[1] (GPIO_6 for ADC0 and GPIO_9 for ADC1, negative input) 0x1: singlediff = 0: ADC_CH[1] (GPIO_6 for ADC0 and GPIO_9 for ADC1) and VSSA singlediff = 1: ADC_CH[2] (GPIO_5 for ADC0 and GPIO_10 for ADC1, positive input) and ADC_CH[3] (GPIO_4 for ADC0 and GPIO_11 for ADC1, negative input) 0x2: singlediff = 0: ADC_CH[2] (GPIO_5 for ADC0 and GPIO_10 for ADC1) and VSSA singlediff = 1: ADC_CH[4] (GPIO_3 for ADC0, positive input) and ADC_CH[5] (GPIO_2 for ADC0, negative input) 0x3: singlediff = 0: ADC_CH[3] (GPIO_4 for ADC0 and GPIO_11 for ADC1) and VSSA singlediff = 1: ADC_CH[6] (GPIO_1 for ADC0, positive input) and ADC_CH[7] (GPIO_0 for ADC0, negative input) 0x4: singlediff = 0: ADC_CH[4] (GPIO_3 for ADC0) and VSSA singlediff = 1: DACA (positive input) and DACB (negative input) 0x5: singlediff = 0: ADC_CH[5] (GPIO_2 for ADC0) and VSSA 0x6: singlediff = 0: ADC_CH[6] (GPIO_1 for ADC0) and VSSA 0x7: singlediff = 0: ADC_CH[7] (GPIO_0 for ADC0) and VSSA 0x8: singlediff = 0: VBAT_S and VSSA 0x9: singlediff = 0: Vref_12 and VSSA 0xA: singlediff = 0: DACA and VSSA 0xB: singlediff = 0: DACB and VSSA 0xC: singlediff = 0: VSSA and VSSA 0xF: singlediff = 0: TEMP_P and VSSA singlediff = 1: TEMP_P (positive input) and TEMP_N (negative input) others: reserved
7:6	VREF_SEL	R/W	0x3	Voltage reference select bits. 0x0: reserved 0x1: Vref_18 0x2: external single-ended reference 0x3: internal reference Vref_12
5:4	Reserved	R/W	0x3	Reserved. Do not change the reset value.

Table 383: Analog Configuration Register (ANA)

Bits	Name	Type	Reset	Description
3	CAL	R/W	0x0	Select ADC calibration mode. 0x0: normal operation 0x1: ADC calibration mode
2	TS_EN	R/W	0x0	Temperature sensor enable bit. 0x0: ADC mode 0x1: TSensor mode
1	EXT_SEL	R/W	0x0	Temperature sensor diode select bit. 0x0: internal diode mode 0x1: external diode mode
0	Reserved	R/W	0x1	Reserved. Do not change the reset value.

A.13.6 DMA Control Register (DMAR)

Instance Name

DMAR

Offset

0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														DMA_EN	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 384: DMA Control Register (DMAR)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	DMA_EN	R/W	0x0	DMA data transfer enable bit. 0x0: Final data transfer is done by CPU. 0x1: Final data transfer is done by DMA.

A.13.7 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																												Reserved				
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	0	1	0	0	

Table 385: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4:0	Reserved	R/W	0x14	Reserved. Do not change the reset value.

A.13.8 Status Register (STATUS)

Instance Name
STATUS

Offset
0x1C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														ACT	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 386: Status Register (STATUS)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	ACT	R	0x0	Conversion active flag. Will be cleared when ADC conversion is stopped. 0x0: ADC conversion stops. 0x1: ADC conversion starts.

A.13.9 Interrupt Status Register (ISR)

Instance Name
ISR

Offset
0x20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																												FILTERSAT	DMA_ERR	OFFSAT	GAINSAT	RDY
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	

Table 387: Interrupt Status Register (ISR)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	FILTERSAT	R	0x0	Digital filtering saturation interrupt status. Will be set to 1 when overflow happened during digital filtering if corresponding 'mask' bit is not set.

Table 387: Interrupt Status Register (ISR)

Bits	Name	Type	Reset	Description
3	DMA_ERR	R	0x0	DMA data transfer failure interrupt status. Will be set to 1 when final data register read and data transfer happened after the next data ready comes if corresponding 'mask' bit is not set.
2	OFFSAT	R	0x0	Offset correction saturation interrupt status. Will be set to 1 when overflow happened during offset calibration if corresponding 'mask' bit is not set.
1	GAINSAT	R	0x0	Gain correction saturation interrupt status. Will be set to 1 when overflow happened during gain calibration if corresponding 'mask' bit is not set.
0	RDY	R	0x0	Conversion data ready interrupt status. Will be set to 1 when data is ready for reading if corresponding 'mask' bit is not set.

A.13.10 Interrupt Mask Register (IMR)

Instance Name
IMR

Offset
0x24

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										FILTERSAT_MASK	DMA_ERR_MASK	OFFSAT_MASK	GAINSAT_MASK	RDY_MASK	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	1	1	1	

Table 388: Interrupt Mask Register (IMR)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	FILTERSAT_MASK	R/W	0x1	Digital filter overflow event interrupt mask bit. Write 1 to mask off the digital filter overflow event interrupt, and write 0 to unmask interrupt.
3	DMA_ERR_MASK	R/W	0x1	DMA data transfer failure event interrupt mask bit. Write 1 to mask off the DMA data transfer failure event interrupt, and write 0 to unmask interrupt.
2	OFFSAT_MASK	R/W	0x1	Offset calibration overflow event interrupt mask bit. Write 1 to mask off the offset calibration overflow event interrupt, and write 0 to unmask interrupt.
1	GAINSAT_MASK	R/W	0x1	Gain calibration overflow event interrupt mask bit. Write 1 to mask off the gain calibration overflow event interrupt, and write 0 to unmask interrupt.

Table 388: Interrupt Mask Register (IMR)

Bits	Name	Type	Reset	Description
0	RDY_MASK	R/W	0x1	Conversion data ready event interrupt mask bit. Write 1 to mask off the conversion data ready event interrupt, and write 0 to unmask interrupt.

A.13.11 Interrupt Raw Status Register (IRSR)

Instance Name
IRSR

Offset
0x28

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										FILTERSAT_RAW	DMA_ERR_RAW	OFFSAT_RAW	GAINSAT_RAW	RDY_RAW	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	

Table 389: Interrupt Raw Status Register (IRSR)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	FILTERSAT_RAW	R	0x0	Digital filtering saturation raw status. Will be set to 1 when overflow happened during digital filtering process regardless the 'mask' bit.
3	DMA_ERR_RAW	R	0x0	DMA data transfer failure raw status. Will be set to 1 when final data register read and data transfer happened after the next data ready comes regardless the 'mask' bit.
2	OFFSAT_RAW	R	0x0	Offset correction saturation raw status. Will be set to 1 when overflow happened during offset calibration regardless the 'mask' bit.
1	GAINSAT_RAW	R	0x0	Gain correction saturation raw status. Will be set to 1 when overflow happened during gain calibration regardless the 'mask' bit.
0	RDY_RAW	R	0x0	Conversion data ready raw status. Will be set to 1 when conversion data is ready for reading regardless the 'mask' bit.

A.13.12 Interrupt Clear Register (ICR)

Instance Name

ICR

Offset

0x2C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											FILTERSAT_CLR	DMA_ERR_CLR	OFFSAT_CLR	GAINSAT_CLR	RDY_CLR
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0

Table 390: Interrupt Clear Register (ICR)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	FILTERSAT_CLR	W	0x0	Digital filtering saturation interrupt flag clear signal. Write 1 to clear filtersat and filtersat_raw.
3	DMA_ERR_CLR	W	0x0	DMA data transfer failure interrupt flag clear signal. Write 1 to clear dma_err and dma_err_raw.
2	OFFSAT_CLR	W	0x0	Offset correction saturation interrupt flag clear signal. Write 1 to clear offsat and offsat_raw.
1	GAINSAT_CLR	W	0x0	Gain correction saturation interrupt flag clear signal. Write 1 to clear gainsat and gainsat_raw.
0	RDY_CLR	W	0x0	Data conversion ready interrupt flag clear signal. Write 1 to clear rdy and rdy_raw.

A.13.13 Final Data Register (RESULT)

Instance Name

RESULT

Offset

0x30

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																DATA															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 391: Final Data Register (RESULT)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	DATA	R	0x0	Conversion result data. The result is a signed 2's complement value. The significant bits of the value begin at bit 15 regardless of the decimation rate/resolution select.

A.13.14 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x34

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 392: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.13.15 Offset Calibration Data Register (OFF_CAL)

Instance Name

OFF_CAL

Offset

0x38

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SYS_CAL																SELF_CAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 393: Offset Calibration Data Register (OFF_CAL)

Bits	Name	Type	Reset	Description
31:16	SYS_CAL	R/W	0x0	System offset calibration value. Signed 16-bit offset subtracted from the basic ADC conversion result before gain correction is applied.
15:0	SELF_CAL	R	0x0	Self offset calibration value. Signed 16-bit offset subtracted from the basic ADC conversion result before system offset and gain correction is applied.

A.13.16 Gain Calibration Data Register (GAIN_CAL)

Instance Name

GAIN_CAL

Offset

0x3C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																GAIN_CAL															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 394: Gain Calibration Data Register (GAIN_CAL)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	GAIN_CAL	R/W	0x8000	Gain calibration value. Gain factor that is multiplied by the offset-corrected ADC result to produce the output value. The gain is a 16-bit unsigned scaled integer value with a binary decimal point between bit 15 and bit 14. It can represent values from 0 to (almost) 2. The reset value is a gain factor of 1.

A.13.17 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x40

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														Reserved	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0		0

Table 395: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.13.18 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x44

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																Reserved															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Table 396: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	Reserved	R/W	0xFF00	Reserved. Do not change the reset value.

A.14 DAC Register Information

Table 397: DAC Register Summary

Offset	Name	Description	Details
0x00	CTRL	DAC Control Register	Page: 294
0x04	STATUS	DAC Status Register	Page: 295
0x08	ACTRL	Channel A Control Register	Page: 295
0x0C	BCTRL	Channel B Control Register	Page: 297
0x10	ADATA	Channel A Data Register	Page: 298
0x14	BDATA	Channel B Data Register	Page: 299
0x18	ISR	Interrupt Status Register	Page: 299
0x1C	IMR	Interrupt Mask Register	Page: 300
0x20	IRSR	Interrupt Raw Status Register	Page: 301
0x24	ICR	Interrupt Clear Register	Page: 302
0x28	CLK	Clock Register	Page: 302
0x2C	RST	Soft Reset Register	Page: 303
0x30	RESERVED	Reserved	Page: 304
0x34	RESERVED	Reserved	Page: 304

A.14.1 DAC Control Register (CTRL)

Instance Name
CTRL

Offset
0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															REF_SEL
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 398: DAC Control Register (CTRL)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	REF_SEL	R/W	0x0	Reference voltage select bit. 0x0: internal reference 0x1: external reference

A.14.2 DAC Status Register (STATUS)

Instance Name
STATUS

Offset
0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														B_DV	A_DV
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 399: DAC Status Register (STATUS)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	B_DV	R	0x0	Channel B output valid status. 0x0: Channel B has new data for conversion. 0x1: Channel B conversion is complete.
0	A_DV	R	0x0	Channel A output valid status. 0x0: Channel A has new data for conversion. 0x1: Channel A conversion is complete.

A.14.3 Channel A Control Register (ACTRL)

Instance Name
ACTRL

Offset
0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Field	Reserved												A_RANGE		A_WAVE		A_TRIA_STEP_SEL		A_TRIA_MAMP_SEL			A_TRIA_HALF		A_MODE		A_DEN		A_TRIG_TYP		A_TRIG_SEL		A_TRIG_EN		A_IO_EN		A_EN
Default	?	?	?	?	?	?	?	?	?	?	?	?	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0				

Table 400: Channel A Control Register (ACTRL)

Bits	Name	Type	Reset	Description
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	A_RANGE	R/W	0x3	Channel A output range control bits (internal reference / external reference) 0x0: 0.2V ~ 1.0V / (0.1 ~ 0.5) * Vref 0x1: 0.225V ~ 1.425V / (0.1125 ~ 0.7125) * Vref 0x2: 0.225V ~ 1.425V / (0.1125 ~ 0.7125) * Vref 0x3: 0.2V ~ 1.8V / (0.1 ~ 0.9) * Vref

Table 400: Channel A Control Register (ACTRL)

Bits	Name	Type	Reset	Description
17:16	A_WAVE	R/W	0x0	Channel A wave type select bits. 0x0: normal 0x1: triangle wave 0x2: sine wave 0x3: noise
15:14	A_TRIA_STEP_SEL	R/W	0x0	Channel A triangle wave step select bits. 0x0: 1 0x1: 3 0x2: 15 0x3: 511
13:10	A_TRIA_MAMP_SEL	R/W	0x0	Channel A triangle wave max amplitude select bits. 0x0: 63 0x1: 127 0x2: 191 0x3: 255 0x4: 319 0x5: 383 0x6: 447 0x7: 511 0x8: 575 0x9: 639 0xA: 703 0xB: 767 0xC: 831 0xD: 895 0xE: 959 0xF: 1023
9	A_TRIA_HALF	R/W	0x0	Channel A triangle wave type select bit. 0x0: full triangle 0x1: half triangle
8	A_MODE	R/W	0x0	Channel A timing mode select bit. 0x0: non-timing related mode 0x1: timing related mode
7	A_DEN	R/W	0x0	Channel A DMA enable bit 0x0: disable DMA data transfer 0x1: enable DMA data transfer
6:5	A_TRIG_TYP	R/W	0x1	Channel A trigger type select bits. 0x0: reserved 0x1: rising edge trigger 0x2: falling edge trigger 0x3: both rising and falling edge trigger

Table 400: Channel A Control Register (ACTRL)

Bits	Name	Type	Reset	Description
4:3	A_TRIG_SEL	R/W	0x3	Channel A trigger select bits. 0x0: conversion triggered by GPT2 event 0x1: conversion triggered by GPT3 event 0x2: conversion triggered by GPIO45 event 0x3: conversion triggered by GPIO44 event
2	A_TRIG_EN	R/W	0x0	Channel A trigger enable bit. 0x0: disable channel A conversion triggered by external event 0x1: enable channel A conversion triggered by external event
1	A_IO_EN	R/W	0x0	Channel A conversion output to pad enable bit. 0x0: disable channel A conversion output to pad 0x1: enable channel A conversion output to pad
0	A_EN	R/W	0x0	Channel A enable bit. 0x0: disable channel A conversion 0x1: enable channel A conversion

A.14.4 Channel B Control Register (BCTRL)

Instance Name
BCTRL

Offset
0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																			Reserved	B_WAVE		B_MODE		B_DEN	B_TRIG_TYP		B_TRIG_SEL		B_TRIG_EN	B_IO_EN	B_EN
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	0	0	0	0	0	1	1	1	0	0	0

Table 401: Channel B Control Register (BCTRL)

Bits	Name	Type	Reset	Description
31:13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12:11	Reserved	R/W	0x3	Reserved. Do not change the reset value.
10:9	B_WAVE	R/W	0x0	Channel B wave type select bits. 0x0: normal 0x1: reserved 0x2: reserved 0x3: differential with channel A

Table 401: Channel B Control Register (BCTRL)

Bits	Name	Type	Reset	Description
8	B_MODE	R/W	0x0	Channel B timing mode select bit. 0x0: non-timing related mode 0x1: timing related mode
7	B_DEN	R/W	0x0	Channel B DMA enable bit 0x0: disable DMA data transfer 0x1: enable DMA data transfer
6:5	B_TRIG_TYP	R/W	0x1	Channel B trigger type select bits. 0x0: reserved 0x1: rising edge trigger 0x2: falling edge trigger 0x3: both rising and falling edge trigger
4:3	B_TRIG_SEL	R/W	0x3	Channel B trigger select bits. 0x0: conversion triggered by GPT2 event 0x1: conversion triggered by GPT3 event 0x2: conversion triggered by GPIO45 event 0x3: conversion triggered by GPIO44 event
2	B_TRIG_EN	R/W	0x0	Channel B trigger enable bit. 0x0: disable channel B conversion triggered by external event 0x1: enable channel B conversion triggered by external event
1	B_IO_EN	R/W	0x0	Channel B conversion output to pad enable bit. 0x0: disable channel B conversion output to pad 0x1: enable channel B conversion output to pad
0	B_EN	R/W	0x0	Channel B enable bit. 0x0: disable channel B conversion 0x1: enable channel B conversion

A.14.5 Channel A Data Register (ADATA)

Instance Name
ADATA

Offset
0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																						A_DATA									
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0

Table 402: Channel A Data Register (ADATA)

Bits	Name	Type	Reset	Description
31:10	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 402: Channel A Data Register (ADATA)

Bits	Name	Type	Reset	Description
9:0	A_DATA	R/W	0x0	Channel A input data. This field is also used as base value when triangle wave mode is selected.

A.14.6 Channel B Data Register (BDATA)

Instance Name
BDATA

Offset
0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																					B_DATA										
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0

Table 403: Channel B Data Register (BDATA)

Bits	Name	Type	Reset	Description
31:10	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
9:0	B_DATA	R/W	0x0	Channel B input data.

A.14.7 Interrupt Status Register (ISR)

Instance Name
ISR

Offset
0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											TRIA_OVFL_INT	B_TO_INT	A_TO_INT	B_RDY_INT	A_RDY_INT
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	

Table 404: Interrupt Status Register (ISR)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	TRIA_OVFL_INT	R	0x0	Triangle wave configuration overflow interrupt status. Will be set to 1 when triangle wave configuration is overflow if corresponding 'mask' bit is not set.
3	B_TO_INT	R	0x0	Channel B data refreshing timeout interrupt status. Will be set to 1 when channel B data refreshing is timeout if corresponding 'mask' bit is not set.

Table 404: Interrupt Status Register (ISR)

Bits	Name	Type	Reset	Description
2	A_TO_INT	R	0x0	Channel A data refreshing timeout interrupt status. Will be set to 1 when channel A data refreshing is timeout if corresponding 'mask' bit is not set.
1	B_RDY_INT	R	0x0	Channel B data ready interrupt status. Will be set to 1 when channel B data is ready if corresponding 'mask' bit is not set.
0	A_RDY_INT	R	0x0	Channel A data ready interrupt status. Will be set to 1 when channel A data is ready if corresponding 'mask' bit is not set.

A.14.8 Interrupt Mask Register (IMR)

Instance Name
IMR

Offset
0x1C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											TRIA_OVFL_INT_MSK	B_TO_INT_MSK	A_TO_INT_MSK	B_RDY_INT_MSK	A_RDY_INT_MSK
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	1	1	1

Table 405: Interrupt Mask Register (IMR)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	TRIA_OVFL_INT_MSK	R/W	0x1	Triangle wave configuration overflow interrupt mask bit. Write 1 to mask off triangle wave configuration overflow interrupt, and write 0 to unmask interrupt.
3	B_TO_INT_MSK	R/W	0x1	Channel B data refreshing timeout interrupt mask bit. Write 1 to mask off channel B timeout interrupt, and write 0 to unmask interrupt.
2	A_TO_INT_MSK	R/W	0x1	Channel A data refreshing timeout interrupt mask bit. Write 1 to mask off channel A timeout interrupt, and write 0 to unmask interrupt.
1	B_RDY_INT_MSK	R/W	0x1	Channel B data ready interrupt mask bit. Write 1 to mask off channel B data ready interrupt, and write 0 to unmask interrupt.

Table 405: Interrupt Mask Register (IMR)

Bits	Name	Type	Reset	Description
0	A_RDY_INT_MSK	R/W	0x1	Channel A data ready interrupt mask bit. Write 1 to mask off channel A data ready interrupt, and write 0 to unmask interrupt.

A.14.9 Interrupt Raw Status Register (IRSR)

Instance Name
IRSR

Offset
0x20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										TRIA_OVFL_INT_RAW	B_TO_INT_RAW	A_TO_INT_RAW	B_RDY_INT_RAW	A_RDY_INT_RAW	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	

Table 406: Interrupt Raw Status Register (IRSR)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	TRIA_OVFL_INT_RAW	R	0x0	Triangle wave configuration overflow interrupt raw status. Will be set to 1 when triangle wave configuration is overflow regardless of the 'mask' bit.
3	B_TO_INT_RAW	R	0x0	Channel B data refreshing timeout interrupt raw status. Will be set to 1 when channel B data refreshing is timeout regardless of the 'mask' bit.
2	A_TO_INT_RAW	R	0x0	Channel A data refreshing timeout interrupt raw status. Will be set to 1 when channel A data refreshing is timeout regardless of the 'mask' bit.
1	B_RDY_INT_RAW	R	0x0	Channel B data ready interrupt raw status. Will be set to 1 when channel B data is ready regardless of the 'mask' bit.
0	A_RDY_INT_RAW	R	0x0	Channel A data ready interrupt raw status. Will be set to 1 when channel A data is ready regardless of the 'mask' bit.

A.14.10 Interrupt Clear Register (ICR)

Instance Name

ICR

Offset

0x24

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											TRIA_OVFL_INT_CLR	B_TO_INT_CLR	A_TO_INT_CLR	B_RDY_INT_CLR	A_RDY_INT_CLR
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0

Table 407: Interrupt Clear Register (ICR)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	TRIA_OVFL_INT_CLR	W	0x0	Triangle wave configuration overflow interrupt clear bit. Write 1 to clear tria_ovfl_int and tria_ovfl_int_raw.
3	B_TO_INT_CLR	W	0x0	Channel B data refreshing timeout interrupt clear bit. Write 1 to clear b_to_int and b_to_int_raw.
2	A_TO_INT_CLR	W	0x0	Channel A data refreshing timeout interrupt clear bit. Write 1 to clear a_to_int and a_to_int_raw.
1	B_RDY_INT_CLR	W	0x0	Channel B data ready interrupt clear bit. Write 1 to clear b_rdy_int and b_rdy_int_raw.
0	A_RDY_INT_CLR	W	0x0	Channel A data ready interrupt clear bit. Write 1 to clear a_rdy_int and a_rdy_int_raw.

A.14.11 Clock Register (CLK)

Instance Name

CLK

Offset

0x28

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											SOFT_CLK_RST	CLK_INV_SEL	CLK_CTRL	FORCE_CLK_ON	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	

Table 408: Clock Register (CLK)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 408: Clock Register (CLK)

Bits	Name	Type	Reset	Description
4	SOFT_CLK_RST	R/W	0x0	Soft reset for clk divider. 0x0: normal 0x1: reset
3	CLK_INV_SEL	R/W	0x0	Selector of the clock inverse or not for digital use. 0x0: buffered clock 0x1: inverted clock
2:1	CLK_CTRL	R/W	0x0	DAC conversion rate selector. 0x0: 62.5K 0x1: 125K 0x2: 250K 0x3: 500K
0	FORCE_CLK_ON	R/W	0x0	Bypass the clock gating. 0x0: normal 0x1: bypass

A.14.12 Soft Reset Register (RST)

Instance Name
RST

Offset
0x2C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														B_SOFT_RST	A_SOFT_RST
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 409: Soft Reset Register (RST)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	B_SOFT_RST	R/W	0x0	Soft reset for DAC channel B. Active high. 0x0: normal 0x1: reset
0	A_SOFT_RST	R/W	0x0	Soft reset for DAC channel A. active high. 0x0: normal 0x1: reset

A.14.13 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x30

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	

Table 410: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.14.14 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x34

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																Reserved															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Table 411: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	Reserved	R/W	0xFF00	Reserved. Do not change the reset value.

A.15 ACOMP Register Information

Table 412: ACOMP Register Summary

Offset	Name	Description	Details
0x00	CTRL0	ACOMP0 Control Register	Page: 305
0x04	CTRL1	ACOMP1 Control Register	Page: 308
0x08	STATUS0	ACOMP0 Status Register	Page: 311
0x0C	STATUS1	ACOMP1 Status Register	Page: 312
0x10	ROUTE0	ACOMP0 Route Register	Page: 312
0x14	ROUTE1	ACOMP1 Route Register	Page: 313
0x18	ISR0	ACOMP0 Interrupt Status Register	Page: 313
0x1C	ISR1	ACOMP1 Interrupt Status Register	Page: 314
0x20	IMR0	ACOMP0 Interrupt Mask Register	Page: 314
0x24	IMR1	ACOMP1 Interrupt Mask Register	Page: 315
0x28	IRSR0	ACOMP0 Interrupt Raw Status Register	Page: 315
0x2C	IRSR1	ACOMP1 Interrupt Raw Status Register	Page: 316
0x30	ICR0	ACOMP0 Interrupt Clear Register	Page: 316
0x34	ICR1	ACOMP1 Interrupt Clear Register	Page: 317
0x38	RST0	ACOMP0 Soft Reset Register	Page: 317
0x3C	RST1	ACOMP1 Soft Reset Register	Page: 318
0x40	RESERVED	Reserved	Page: 318
0x44	RESERVED	Reserved	Page: 319
0x48	CLK	Clock Register	Page: 319
0x4C	RESERVED	Reserved	Page: 320

A.15.1 ACOMP0 Control Register (CTRL0)

Instance Name

CTRL0

Offset

0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	EDGE_LVL_SEL	INT_ACT_HI	FIE	RIE	INACT_VAL	MUXEN	POS_SEL				NEG_SEL				LEVEL_SEL						BIAS_PROG		HYST_SEL_P		HYST_SEL_N		WARMTIME		GPIOINV		EN	
Default	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 413: ACOMP0 Control Register (CTRL0)

Bits	Name	Type	Reset	Description
31	EDGE_LEVEL_SEL	R/W	0x0	ACOMP0 interrupt type select. 0x0: level triggered interrupt 0x1: edge triggered interrupt
30	INT_ACT_HI	R/W	0x1	ACOMP0 interrupt active mode select. 0x0: low level or falling edge triggered interrupt 0x1: high level or rising edge triggered interrupt
29	FIE	R/W	0x0	ACOMP0 enable/disable falling edge triggered edge pulse. 0x0: disable 0x1: enable
28	RIE	R/W	0x0	ACOMP0 enable/disable rising edge triggered edge pulse. 0x0: disable 0x1: enable
27	INACT_VAL	R/W	0x0	Set output value when ACOMP0 is inactive. 0x0: output 0 when ACOMP0 is inactive 0x1: output 1 when ACOMP0 is inactive
26	MUXEN	R/W	0x0	ACOMP0 input MUX enable bit. This bit should be asserted earlier than 'en' bit. 0x0: disable input MUX 0x1: enable input MUX
25:22	POS_SEL	R/W	0x0	ACOMP0 positive input select bits. 0x0: GPIO7 0x1: GPIO6 0x2: GPIO5 0x3: GPIO4 0x4: GPIO3 0x5: GPIO2 0x6: GPIO1 0x7: GPIO0 0x8: DACA 0x9: DACB others: reserved

Table 413: ACOMP0 Control Register (CTRL0)

Bits	Name	Type	Reset	Description
21:18	NEG_SEL	R/W	0x0	ACOMP0 negative input select bits. 0x0: GPIO7 0x1: GPIO6 0x2: GPIO5 0x3: GPIO4 0x4: GPIO3 0x5: GPIO2 0x6: GPIO1 0x7: GPIO0 0x8: DACA 0x9: DACB 0xA: Vref_12 0xB: VSSA 0xC: VBAT * scaling factor others: reserved
17:12	LEVEL_SEL	R/W	0x0	Scaling factor select bits for vat reference level. 0x0X: scaling factor = 0.25 0x1X: scaling factor = 0.5 0x2X: scaling factor = 0.75 0x3X: scaling factor = 1 others: reserved
11:10	BIAS_PROG	R/W	0x0	ACOMP0 bias current control bits Or response time control bits. 0x0: power mode 1 (slow response mode) 0x1: power mode 2 (medium response mode) 0x2: power mode 3 (fast response mode) 0x3: reserved
9:7	HYST_SELP	R/W	0x0	Select ACOMP0 positive hysteresis voltage level. 0x0: no hysteresis 0x1: 10mV hysteresis 0x2: 20mV hysteresis 0x3: 30mV hysteresis 0x4: 40mV hysteresis 0x5: 50mV hysteresis 0x6: 60mV hysteresis 0x7: 70mV hysteresis

Table 413: ACOMP0 Control Register (CTRL0)

Bits	Name	Type	Reset	Description
6:4	HYST_SELN	R/W	0x0	Select ACOMP0 negative hysteresis voltage level. 0x0: no hysteresis 0x1: 10mV hysteresis 0x2: 20mV hysteresis 0x3: 30mV hysteresis 0x4: 40mV hysteresis 0x5: 50mV hysteresis 0x6: 60mV hysteresis 0x7: 70mV hysteresis
3:2	WARMTIME	R/W	0x0	Set ACOMP0 warm-up time. 0x0: 16 clock cycles 0x1: 32 clock cycles 0x2: 64 clock cycles 0x3: 128 clock cycles
1	GPIOINV	R/W	0x0	Enable/disable inversion of ACOMP0 output to GPIO. 0x0: not invert ACOMP0 output 0x1: invert ACOMP0 output
0	EN	R/W	0x0	ACOMP0 enable bit. 0x0: disable 0x1: enable

A.15.2 ACOMP1 Control Register (CTRL1)

Instance Name

CTRL1

Offset

0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	EDGE_LVL_SEL	INT_ACT_HI	FIE	RIE	INACT_VAL	MUXEN	POS_SEL				NEG_SEL				LEVEL_SEL						BIAS_PROG		HYST_SEL_P			HYST_SEL_N			WARMTIME		GPIOINV		EN
Default	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 414: ACOMP1 Control Register (CTRL1)

Bits	Name	Type	Reset	Description
31	EDGE_LVL_SEL	R/W	0x0	ACOMP1 interrupt type select. 0x0: level triggered interrupt 0x1: edge triggered interrupt

Table 414: ACOMP1 Control Register (CTRL1)

Bits	Name	Type	Reset	Description
30	INT_ACT_HI	R/W	0x1	ACOMP1 interrupt active mode select. 0x0: low level or falling edge triggered interrupt 0x1: high level or rising edge triggered interrupt
29	FIE	R/W	0x0	ACOMP1 enable/disable falling edge triggered edge pulse. 0x0: disable 0x1: enable
28	RIE	R/W	0x0	ACOMP1 enable/disable rising edge triggered edge pulse. 0x0: disable 0x1: enable
27	INACT_VAL	R/W	0x0	Set output value when ACOMP1 is inactive. 0x0: output 0 when ACOMP1 is inactive 0x1: output 1 when ACOMP1 is inactive
26	MUXEN	R/W	0x0	ACOMP1 input MUX enable bit. This bit should be asserted earlier than 'en' bit. 0x0: disable input MUX 0x1: enable input MUX
25:22	POS_SEL	R/W	0x0	ACOMP1 positive input select bits. 0x0: GPIO7 0x1: GPIO6 0x2: GPIO5 0x3: GPIO4 0x4: GPIO3 0x5: GPIO2 0x6: GPIO1 0x7: GPIO0 0x8: DACA 0x9: DACB others: reserved

Table 414: ACOMP1 Control Register (CTRL1)

Bits	Name	Type	Reset	Description
21:18	NEG_SEL	R/W	0x0	ACOMP1 negative input select bits. 0x0: GPIO7 0x1: GPIO6 0x2: GPIO5 0x3: GPIO4 0x4: GPIO3 0x5: GPIO2 0x6: GPIO1 0x7: GPIO0 0x8: DACA 0x9: DACB 0xA: Vref_12 0xB: VSSA 0xC: VBAT * scaling factor others: reserved
17:12	LEVEL_SEL	R/W	0x0	Scaling factor select bits for vat reference level. 0x0X: scaling factor = 0.25 0x1X: scaling factor = 0.5 0x2X: scaling factor = 0.75 0x3X: scaling factor = 1 others: reserved
11:10	BIAS_PROG	R/W	0x0	ACOMP1 bias current control bits Or response time control bits. 0x0: power mode 1 (slow response mode) 0x1: power mode 2 (medium response mode) 0x2: power mode 3 (fast response mode) 0x3: reserved
9:7	HYST_SELP	R/W	0x0	Select ACOMP1 positive hysteresis voltage level. 0x0: no hysteresis 0x1: 10mV hysteresis 0x2: 20mV hysteresis 0x3: 30mV hysteresis 0x4: 40mV hysteresis 0x5: 50mV hysteresis 0x6: 60mV hysteresis 0x7: 70mV hysteresis

Table 414: ACOMP1 Control Register (CTRL1)

Bits	Name	Type	Reset	Description
6:4	HYST_SELN	R/W	0x0	Select ACOMP1 negative hysteresis voltage level. 0x0: no hysteresis 0x1: 10mV hysteresis 0x2: 20mV hysteresis 0x3: 30mV hysteresis 0x4: 40mV hysteresis 0x5: 50mV hysteresis 0x6: 60mV hysteresis 0x7: 70mV hysteresis
3:2	WARMTIME	R/W	0x0	Set ACOMP1 warm-up time. 0x0: 16 clock cycles 0x1: 32 clock cycles 0x2: 64 clock cycles 0x3: 128 clock cycles
1	GPIOINV	R/W	0x0	Enable/disable inversion of ACOMP1 output to GPIO. 0x0: not invert ACOMP1 output 0x1: invert ACOMP1 output
0	EN	R/W	0x0	ACOMP1 enable bit. 0x0: disable 0x1: enable

A.15.3 ACOMP0 Status Register (STATUS0)

Instance Name
STATUS0

Offset
0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														OUT	ACT
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 415: ACOMP0 Status Register (STATUS0)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	OUT	R	0x0	ACOMP0 comparison output value. 0x0: ACOMP0 comparison output value is 0. 0x1: ACOMP0 comparison output value is 1.
0	ACT	R	0x0	ACOMP0 active status. 0x0: ACOMP0 is inactive 0x1: ACOMP0 is active

A.15.4 ACOMP1 Status Register (STATUS1)

Instance Name Offset
STATUS1 0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														OUT	ACT
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 416: ACOMP1 Status Register (STATUS1)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	OUT	R	0x0	ACOMP1 comparison output value. 0x0: ACOMP1 comparison output value is 0. 0x1: ACOMP1 comparison output value is 1.
0	ACT	R	0x0	ACOMP1 active status. 0x0: ACOMP1 is inactive 0x1: ACOMP1 is active

A.15.5 ACOMP0 Route Register (ROUTE0)

Instance Name Offset
ROUTE0 0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														PE	OUTSEL
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 417: ACOMP0 Route Register (ROUTE0)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	PE	R/W	0x0	Enable/disable ACOMP0 output to pin. 0x0: disable 0x1: enable
0	OUTSEL	R/W	0x0	Select ACOMP0 synchronous or asynchronous output to pin. 0x0: synchronous output 0x1: asynchronous output

A.15.6 ACOMP1 Route Register (ROUTE1)

Instance Name
ROUTE1Offset
0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														PE	OUTSEL
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 418: ACOMP1 Route Register (ROUTE1)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	PE	R/W	0x0	Enable/disable ACOMP1 output to pin. 0x0: disable 0x1: enable
0	OUTSEL	R/W	0x0	Select ACOMP1 synchronous or asynchronous output to pin. 0x0: synchronous output 0x1: asynchronous output

A.15.7 ACOMP0 Interrupt Status Register (ISR0)

Instance Name
ISR0Offset
0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														OUTA_INT	OUT_INT
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 419: ACOMP0 Interrupt Status Register (ISR0)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	OUTA_INT	R	0x0	ACOMP0 asynchronous output interrupt status. Will be set to 1 when ACOMP0 asynchronous output changes from 0 to 1 if corresponding 'mask' bit is not set.
0	OUT_INT	R	0x0	ACOMP0 synchronous output interrupt status. Will be set to 1 when ACOMP0 synchronous output changes from 0 to 1 if corresponding 'mask' bit is not set.

A.15.8 ACOMP1 Interrupt Status Register (ISR1)

Instance Name
ISR1

Offset
0x1C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														OUTA_INT	OUT_INT
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 420: ACOMP1 Interrupt Status Register (ISR1)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	OUTA_INT	R	0x0	ACOMP1 asynchronous output interrupt status. Will be set to 1 when ACOMP1 asynchronous output changes from 0 to 1 if corresponding 'mask' bit is not set.
0	OUT_INT	R	0x0	ACOMP1 synchronous output interrupt status. Will be set to 1 when ACOMP1 synchronous output changes from 0 to 1 if corresponding 'mask' bit is not set.

A.15.9 ACOMP0 Interrupt Mask Register (IMR0)

Instance Name
IMR0

Offset
0x20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														OUTA_INT_MASK	OUT_INT_MASK
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1

Table 421: ACOMP0 Interrupt Mask Register (IMR0)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	OUTA_INT_MASK	R/W	0x1	ACOMP0 asynchronous output interrupt mask bit. Write 1 to mask off the ACOMP0 asynchronous output interrupt, and write 0 to unmask interrupt.
0	OUT_INT_MASK	R/W	0x1	ACOMP0 synchronous output interrupt mask bit. Write 1 to mask off the ACOMP0 synchronous output interrupt, and write 0 to unmask interrupt.

A.15.10 ACOMP1 Interrupt Mask Register (IMR1)

Instance Name

IMR1

Offset

0x24

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														OUTA_INT_MASK	OUT_INT_MASK
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1

Table 422: ACOMP1 Interrupt Mask Register (IMR1)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	OUTA_INT_MASK	R/W	0x1	ACOMP1 asynchronous output interrupt mask bit. Write 1 to mask off the ACOMP1 asynchronous output interrupt, and write 0 to unmask interrupt.
0	OUT_INT_MASK	R/W	0x1	ACOMP1 synchronous output interrupt mask bit. Write 1 to mask off the ACOMP1 synchronous output interrupt, and write 0 to unmask interrupt.

A.15.11 ACOMP0 Interrupt Raw Status Register (IRSR0)

Instance Name

IRSR0

Offset

0x28

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														OUTA_INT_RAW	OUT_INT_RAW
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 423: ACOMP0 Interrupt Raw Status Register (IRSR0)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	OUTA_INT_RAW	R	0x0	ACOMP0 asynchronous output interrupt raw status. Will be set to 1 when ACOMP0 asynchronous output changes from 0 to 1 regardless the 'mask' bit.
0	OUT_INT_RAW	R	0x0	ACOMP0 synchronous output interrupt raw status. Will be set to 1 when ACOMP0 synchronous output changes from 0 to 1 regardless the 'mask' bit.

A.15.12 ACOMP1 Interrupt Raw Status Register (IRSR1)

Instance Name
IRSR1

Offset
0x2C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														OUTA_INT_RAW	OUT_INT_RAW
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 424: ACOMP1 Interrupt Raw Status Register (IRSR1)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	OUTA_INT_RAW	R	0x0	ACOMP1 asynchronous output interrupt raw status. Will be set to 1 when ACOMP1 asynchronous output changes from 0 to 1 regardless the 'mask' bit.
0	OUT_INT_RAW	R	0x0	ACOMP1 synchronous output interrupt raw status. Will be set to 1 when ACOMP1 synchronous output changes from 0 to 1 regardless the 'mask' bit.

A.15.13 ACOMP0 Interrupt Clear Register (ICR0)

Instance Name
ICR0

Offset
0x30

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														OUTA_INT_CLR	OUT_INT_CLR
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 425: ACOMP0 Interrupt Clear Register (ICR0)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	OUTA_INT_CLR	W	0x0	ACOMP0 asynchronous output interrupt flag clear signal. Write 1 to clear ACOMP0 outa_int and outa_int_raw.
0	OUT_INT_CLR	W	0x0	ACOMP0 synchronous output interrupt flag clear signal. Write 1 to clear ACOMP0 out_int and out_int_raw.

A.15.14 ACOMP1 Interrupt Clear Register (ICR1)

Instance Name
ICR1

Offset
0x34

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														OUTA_INT_CLR	OUT_INT_CLR
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 426: ACOMP1 Interrupt Clear Register (ICR1)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	OUTA_INT_CLR	W	0x0	ACOMP1 asynchronous output interrupt flag clear signal. Write 1 to clear ACOMP1 outa_int and outa_int_raw.
0	OUT_INT_CLR	W	0x0	ACOMP1 synchronous output interrupt flag clear signal. Write 1 to clear ACOMP1 out_int and out_int_raw.

A.15.15 ACOMP0 Soft Reset Register (RST0)

Instance Name
RST0

Offset
0x38

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														SOFT_RST	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 427: ACOMP0 Soft Reset Register (RST0)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	SOFT_RST	R/W	0x0	Soft reset for ACOMP0. 0x0: normal 0x1: reset

A.15.16 ACOMP1 Soft Reset Register (RST1)

Instance Name
RST1

Offset
0x3C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															SOFT_RST
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 428: ACOMP1 Soft Reset Register (RST1)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	SOFT_RST	R/W	0x0	Soft reset for ACOMP1. 0x0: normal 0x1: reset

A.15.17 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x40

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														Reserved	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0		0

Table 429: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.15.18 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x44

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														Reserved	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0		0

Table 430: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.15.19 Clock Register (CLK)

Instance Name

CLK

Offset

0x48

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														SOFT_CLK_RST	FORCE_CLK_ON
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 431: Clock Register (CLK)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	SOFT_CLK_RST	R/W	0x0	Soft reset for clock divider. 0x0: normal 0x1: reset
0	FORCE_CLK_ON	R/W	0x0	Bypass the clock gating if set to 1. 0x0: Clock gates are functional. 0x1: Clock gates are bypassed.

A.15.20 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x4C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																Reserved															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Table 432: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:0	Reserved	R/W	0xFF00	Reserved. Do not change the reset value.

A.16 PINMUX Register Information

Table 433: PINMUX Register Summary

Offset	Name	Description	Details
0x000	GPIO_PINMUX0	GPIO_PINMUX Configuration Register	Page: 324
0x004	GPIO_PINMUX1	GPIO_PINMUX Configuration Register	Page: 324
0x008	GPIO_PINMUX2	GPIO_PINMUX Configuration Register	Page: 325
0x00C	GPIO_PINMUX3	GPIO_PINMUX Configuration Register	Page: 326
0x010	GPIO_PINMUX4	GPIO_PINMUX Configuration Register	Page: 326
0x014	GPIO_PINMUX5	GPIO_PINMUX Configuration Register	Page: 327
0x018	GPIO_PINMUX6	GPIO_PINMUX Configuration Register	Page: 328
0x01C	GPIO_PINMUX7	GPIO_PINMUX Configuration Register	Page: 328
0x020	GPIO_PINMUX8	GPIO_PINMUX Configuration Register	Page: 329
0x024	GPIO_PINMUX9	GPIO_PINMUX Configuration Register	Page: 330
0x028	GPIO_PINMUX10	GPIO_PINMUX Configuration Register	Page: 330
0x02C	GPIO_PINMUX11	GPIO_PINMUX Configuration Register	Page: 331
0x030	RESERVED	Reserved	Page: 332
0x034	RESERVED	Reserved	Page: 332
0x038	RESERVED	Reserved	Page: 333
0x03C	RESERVED	Reserved	Page: 333
0x040	GPIO_PINMUX16	GPIO_PINMUX Configuration Register	Page: 334
0x044	GPIO_PINMUX17	GPIO_PINMUX Configuration Register	Page: 334
0x048	GPIO_PINMUX18	GPIO_PINMUX Configuration Register	Page: 335
0x04C	GPIO_PINMUX19	GPIO_PINMUX Configuration Register	Page: 336
0x050	GPIO_PINMUX20	GPIO_PINMUX Configuration Register	Page: 336
0x054	GPIO_PINMUX21	GPIO_PINMUX Configuration Register	Page: 337
0x058	GPIO_PINMUX22	GPIO_PINMUX Configuration Register	Page: 338
0x05C	GPIO_PINMUX23	GPIO_PINMUX Configuration Register	Page: 338
0x060	GPIO_PINMUX24	GPIO_PINMUX Configuration Register	Page: 339
0x064	GPIO_PINMUX25	GPIO_PINMUX Configuration Register	Page: 340
0x068	GPIO_PINMUX26	GPIO_PINMUX Configuration Register	Page: 340
0x06C	GPIO_PINMUX27	GPIO_PINMUX Configuration Register	Page: 341

Table 433: PINMUX Register Summary

Offset	Name	Description	Details
0x070	GPIO_PINMUX28	GPIO_PINMUX Configuration Register	Page: 342
0x074	GPIO_PINMUX29	GPIO_PINMUX Configuration Register	Page: 342
0x078	GPIO_PINMUX30	GPIO_PINMUX Configuration Register	Page: 343
0x07C	RESERVED	Reserved	Page: 344
0x080	GPIO_PINMUX32	GPIO_PINMUX Configuration Register	Page: 344
0x084	GPIO_PINMUX33	GPIO_PINMUX Configuration Register	Page: 345
0x088	GPIO_PINMUX34	GPIO_PINMUX Configuration Register	Page: 346
0x08C	GPIO_PINMUX35	GPIO_PINMUX Configuration Register	Page: 346
0x090	RESERVED	Reserved	Page: 347
0x094	RESERVED	Reserved	Page: 348
0x098	RESERVED	Reserved	Page: 348
0x09C	RESERVED	Reserved	Page: 349
0x0A0	GPIO_PINMUX40	GPIO_PINMUX Configuration Register	Page: 349
0x0A4	GPIO_PINMUX41	GPIO_PINMUX Configuration Register	Page: 350
0x0A8	GPIO_PINMUX42	GPIO_PINMUX Configuration Register	Page: 351
0x0AC	GPIO_PINMUX43	GPIO_PINMUX Configuration Register	Page: 351
0x0B0	GPIO_PINMUX44	GPIO_PINMUX Configuration Register	Page: 352
0x0B4	GPIO_PINMUX45	GPIO_PINMUX Configuration Register	Page: 353
0x0B8	RESERVED	Reserved	Page: 353
0x0BC	RESERVED	Reserved	Page: 354
0x0C0	RESERVED	Reserved	Page: 354
0x0C4	RESERVED	Reserved	Page: 355
0x0C8	GPIO_PINMUX50	GPIO_PINMUX Configuration Register	Page: 355
0x0CC	GPIO_PINMUX51	GPIO_PINMUX Configuration Register	Page: 356
0x0D0	GPIO_PINMUX52	GPIO_PINMUX Configuration Register	Page: 357
0x0D4	GPIO_PINMUX53	GPIO_PINMUX Configuration Register	Page: 357
0x0D8	GPIO_PINMUX54	GPIO_PINMUX Configuration Register	Page: 358
0x0DC	GPIO_PINMUX55	GPIO_PINMUX Configuration Register	Page: 359
0x0E0	GPIO_PINMUX56	GPIO_PINMUX Configuration Register	Page: 359
0x0E4	GPIO_PINMUX57	GPIO_PINMUX Configuration Register	Page: 360
0x0E8	GPIO_PINMUX58	GPIO_PINMUX Configuration Register	Page: 361

Table 433: PINMUX Register Summary

Offset	Name	Description	Details
0x0EC	GPIO_PINMUX59	GPIO_PINMUX Configuration Register	Page: 361
0x0F0	GPIO_PINMUX60	GPIO_PINMUX Configuration Register	Page: 362
0x0F4	GPIO_PINMUX61	GPIO_PINMUX Configuration Register	Page: 363
0x0F8	GPIO_PINMUX62	GPIO_PINMUX Configuration Register	Page: 363
0x0FC	GPIO_PINMUX63	GPIO_PINMUX Configuration Register	Page: 364
0x100	GPIO_PINMUX64	GPIO_PINMUX Configuration Register	Page: 365
0x104	GPIO_PINMUX65	GPIO_PINMUX Configuration Register	Page: 365
0x108	GPIO_PINMUX66	GPIO_PINMUX Configuration Register	Page: 366
0x10C	RESERVED	Reserved	Page: 367
0x110	GPIO_PINMUX68	GPIO_PINMUX Configuration Register	Page: 367
0x114	RESERVED	Reserved	Page: 368
0x118	RESERVED	Reserved	Page: 368
0x11C	RESERVED	Reserved	Page: 369
0x120	GPIO_PINMUX72	GPIO_PINMUX Configuration Register	Page: 369
0x124	GPIO_PINMUX73	GPIO_PINMUX Configuration Register	Page: 370
0x128	GPIO_PINMUX74	GPIO_PINMUX Configuration Register	Page: 371
0x12C	GPIO_PINMUX75	GPIO_PINMUX Configuration Register	Page: 371
0x130	GPIO_PINMUX76	GPIO_PINMUX Configuration Register	Page: 372
0x134	GPIO_PINMUX77	GPIO_PINMUX Configuration Register	Page: 373
0x138	GPIO_PINMUX78	GPIO_PINMUX Configuration Register	Page: 373
0x13C	GPIO_PINMUX79	GPIO_PINMUX Configuration Register	Page: 374
0x140	RESERVED	Reserved	Page: 375
0x144	RESERVED	Reserved	Page: 375
0x148	RESERVED	Reserved	Page: 376
0x14C	RESERVED	Reserved	Page: 376
0x150	RESERVED	Reserved	Page: 377
0x154	RESERVED	Reserved	Page: 377

A.16.1 GPIO_PINMUX Configuration Register (GPIO_PINMUX0)

Instance Name
GPIO_PINMUX0

Offset
0x000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 434: GPIO_PINMUX Configuration Register (GPIO_PINMUX0)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.2 GPIO_PINMUX Configuration Register (GPIO_PINMUX1)

Instance Name
GPIO_PINMUX1

Offset
0x004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 435: GPIO_PINMUX Configuration Register (GPIO_PINMUX1)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 435: GPIO_PINMUX Configuration Register (GPIO_PINMUX1)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.3 GPIO_PINMUX Configuration Register (GPIO_PINMUX2)

Instance Name
GPIO_PINMUX2

Offset
0x008

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 436: GPIO_PINMUX Configuration Register (GPIO_PINMUX2)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.4 GPIO_PINMUX Configuration Register (GPIO_PINMUX3)

Instance Name
GPIO_PINMUX3

Offset
0x00C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved								DI_EN	FSEL_XR			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 437: GPIO_PINMUX Configuration Register (GPIO_PINMUX3)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.5 GPIO_PINMUX Configuration Register (GPIO_PINMUX4)

Instance Name
GPIO_PINMUX4

Offset
0x010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved								DI_EN	FSEL_XR			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 438: GPIO_PINMUX Configuration Register (GPIO_PINMUX4)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 438: GPIO_PINMUX Configuration Register (GPIO_PINMUX4)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.6 GPIO_PINMUX Configuration Register (GPIO_PINMUX5)

Instance Name
GPIO_PINMUX5

Offset
0x014

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 439: GPIO_PINMUX Configuration Register (GPIO_PINMUX5)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.7 GPIO_PINMUX Configuration Register (GPIO_PINMUX6)

Instance Name
GPIO_PINMUX6

Offset
0x018

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 440: GPIO_PINMUX Configuration Register (GPIO_PINMUX6)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.8 GPIO_PINMUX Configuration Register (GPIO_PINMUX7)

Instance Name
GPIO_PINMUX7

Offset
0x01C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 441: GPIO_PINMUX Configuration Register (GPIO_PINMUX7)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 441: GPIO_PINMUX Configuration Register (GPIO_PINMUX7)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.9 GPIO_PINMUX Configuration Register (GPIO_PINMUX8)

Instance Name
GPIO_PINMUX8

Offset
0x020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 442: GPIO_PINMUX Configuration Register (GPIO_PINMUX8)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.10 GPIO_PINMUX Configuration Register (GPIO_PINMUX9)

Instance Name
GPIO_PINMUX9

Offset
0x024

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 443: GPIO_PINMUX Configuration Register (GPIO_PINMUX9)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.11 GPIO_PINMUX Configuration Register (GPIO_PINMUX10)

Instance Name
GPIO_PINMUX10

Offset
0x028

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 444: GPIO_PINMUX Configuration Register (GPIO_PINMUX10)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 444: GPIO_PINMUX Configuration Register (GPIO_PINMUX10)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.12 GPIO_PINMUX Configuration Register (GPIO_PINMUX11)

Instance Name
GPIO_PINMUX11

Offset
0x02C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 445: GPIO_PINMUX Configuration Register (GPIO_PINMUX11)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.13 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x030

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																Reserved	Reserved										Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0

Table 446: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.14 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x034

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																Reserved			Reserved								Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 447: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.15 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x038

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																Reserved	Reserved								Reserved							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0

Table 448: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.16 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x03C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																Reserved	Reserved								Reserved							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0

Table 449: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.17 GPIO_PINMUX Configuration Register (GPIO_PINMUX16)

Instance Name
GPIO_PINMUX16

Offset
0x040

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 450: GPIO_PINMUX Configuration Register (GPIO_PINMUX16)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.18 GPIO_PINMUX Configuration Register (GPIO_PINMUX17)

Instance Name
GPIO_PINMUX17

Offset
0x044

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 451: GPIO_PINMUX Configuration Register (GPIO_PINMUX17)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 451: GPIO_PINMUX Configuration Register (GPIO_PINMUX17)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.19 GPIO_PINMUX Configuration Register (GPIO_PINMUX18)

Instance Name
GPIO_PINMUX18

Offset
0x048

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 452: GPIO_PINMUX Configuration Register (GPIO_PINMUX18)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.20 GPIO_PINMUX Configuration Register (GPIO_PINMUX19)

Instance Name
GPIO_PINMUX19

Offset
0x04C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 453: GPIO_PINMUX Configuration Register (GPIO_PINMUX19)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.21 GPIO_PINMUX Configuration Register (GPIO_PINMUX20)

Instance Name
GPIO_PINMUX20

Offset
0x050

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 454: GPIO_PINMUX Configuration Register (GPIO_PINMUX20)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 454: GPIO_PINMUX Configuration Register (GPIO_PINMUX20)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.22 GPIO_PINMUX Configuration Register (GPIO_PINMUX21)

Instance Name
GPIO_PINMUX21

Offset
0x054

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 455: GPIO_PINMUX Configuration Register (GPIO_PINMUX21)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.23 GPIO_PINMUX Configuration Register (GPIO_PINMUX22)

Instance Name
GPIO_PINMUX22

Offset
0x058

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 456: GPIO_PINMUX Configuration Register (GPIO_PINMUX22)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.24 GPIO_PINMUX Configuration Register (GPIO_PINMUX23)

Instance Name
GPIO_PINMUX23

Offset
0x05C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 457: GPIO_PINMUX Configuration Register (GPIO_PINMUX23)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 457: GPIO_PINMUX Configuration Register (GPIO_PINMUX23)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.25 GPIO_PINMUX Configuration Register (GPIO_PINMUX24)

Instance Name
GPIO_PINMUX24

Offset
0x060

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 458: GPIO_PINMUX Configuration Register (GPIO_PINMUX24)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.26 GPIO_PINMUX Configuration Register (GPIO_PINMUX25)

Instance Name
GPIO_PINMUX25

Offset
0x064

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 459: GPIO_PINMUX Configuration Register (GPIO_PINMUX25)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.27 GPIO_PINMUX Configuration Register (GPIO_PINMUX26)

Instance Name
GPIO_PINMUX26

Offset
0x068

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 460: GPIO_PINMUX Configuration Register (GPIO_PINMUX26)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 460: GPIO_PINMUX Configuration Register (GPIO_PINMUX26)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.28 GPIO_PINMUX Configuration Register (GPIO_PINMUX27)

Instance Name
GPIO_PINMUX27

Offset
0x06C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 461: GPIO_PINMUX Configuration Register (GPIO_PINMUX27)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.29 GPIO_PINMUX Configuration Register (GPIO_PINMUX28)

Instance Name
GPIO_PINMUX28

Offset
0x070

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 462: GPIO_PINMUX Configuration Register (GPIO_PINMUX28)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.30 GPIO_PINMUX Configuration Register (GPIO_PINMUX29)

Instance Name
GPIO_PINMUX29

Offset
0x074

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 463: GPIO_PINMUX Configuration Register (GPIO_PINMUX29)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 463: GPIO_PINMUX Configuration Register (GPIO_PINMUX29)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.31 GPIO_PINMUX Configuration Register (GPIO_PINMUX30)

Instance Name
GPIO_PINMUX30

Offset
0x078

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 464: GPIO_PINMUX Configuration Register (GPIO_PINMUX30)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.32 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x07C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																Reserved	Reserved								Reserved						
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 465: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.33 GPIO_PINMUX Configuration Register (GPIO_PINMUX32)

Instance Name
GPIO_PINMUX32

Offset
0x080

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved								DI_EN	FSEL_XR			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 466: GPIO_PINMUX Configuration Register (GPIO_PINMUX32)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 466: GPIO_PINMUX Configuration Register (GPIO_PINMUX32)

Bits	Name	Type	Reset	Description
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.34 GPIO_PINMUX Configuration Register (GPIO_PINMUX33)

Instance Name
GPIO_PINMUX33

Offset
0x084

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 467: GPIO_PINMUX Configuration Register (GPIO_PINMUX33)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.35 GPIO_PINMUX Configuration Register (GPIO_PINMUX34)

Instance Name
GPIO_PINMUX34

Offset
0x088

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 468: GPIO_PINMUX Configuration Register (GPIO_PINMUX34)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.36 GPIO_PINMUX Configuration Register (GPIO_PINMUX35)

Instance Name
GPIO_PINMUX35

Offset
0x08C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 469: GPIO_PINMUX Configuration Register (GPIO_PINMUX35)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 469: GPIO_PINMUX Configuration Register (GPIO_PINMUX35)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.37 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x090

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																Reserved			Reserved								Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 470: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.38 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x094

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																Reserved	Reserved										Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0

Table 471: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.39 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x098

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																Reserved			Reserved								Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 472: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.40 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x09C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																Reserved	Reserved								Reserved						
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 473: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.41 GPIO_PINMUX Configuration Register (GPIO_PINMUX40)

Instance Name

GPIO_PINMUX40

Offset

0x0A0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved								DI_EN	FSEL_XR			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 474: GPIO_PINMUX Configuration Register (GPIO_PINMUX40)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 474: GPIO_PINMUX Configuration Register (GPIO_PINMUX40)

Bits	Name	Type	Reset	Description
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.42 GPIO_PINMUX Configuration Register (GPIO_PINMUX41)

Instance Name

GPIO_PINMUX41

Offset

0x0A4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 475: GPIO_PINMUX Configuration Register (GPIO_PINMUX41)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.43 GPIO_PINMUX Configuration Register (GPIO_PINMUX42)

Instance Name
GPIO_PINMUX42

Offset
0x0A8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 476: GPIO_PINMUX Configuration Register (GPIO_PINMUX42)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.44 GPIO_PINMUX Configuration Register (GPIO_PINMUX43)

Instance Name
GPIO_PINMUX43

Offset
0x0AC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 477: GPIO_PINMUX Configuration Register (GPIO_PINMUX43)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 477: GPIO_PINMUX Configuration Register (GPIO_PINMUX43)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.45 GPIO_PINMUX Configuration Register (GPIO_PINMUX44)

Instance Name
GPIO_PINMUX44

Offset
0x0B0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 478: GPIO_PINMUX Configuration Register (GPIO_PINMUX44)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.46 GPIO_PINMUX Configuration Register (GPIO_PINMUX45)

Instance Name
GPIO_PINMUX45

Offset
0x0B4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 479: GPIO_PINMUX Configuration Register (GPIO_PINMUX45)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.47 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x0B8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved															Reserved			Reserved										Reserved				
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 480: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.

Table 480: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.48 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x0BC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																Reserved			Reserved								Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 481: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.49 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x0C0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																Reserved			Reserved								Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 482: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.

Table 482: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.50 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x0C4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved														Reserved	Reserved										Reserved						
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?		?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0

Table 483: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.51 GPIO_PINMUX Configuration Register (GPIO_PINMUX50)

Instance Name
GPIO_PINMUX50

Offset
0x0C8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0		

Table 484: GPIO_PINMUX Configuration Register (GPIO_PINMUX50)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 484: GPIO_PINMUX Configuration Register (GPIO_PINMUX50)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.52 GPIO_PINMUX Configuration Register (GPIO_PINMUX51)

Instance Name

GPIO_PINMUX51

Offset

0x0CC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 485: GPIO_PINMUX Configuration Register (GPIO_PINMUX51)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.53 GPIO_PINMUX Configuration Register (GPIO_PINMUX52)

Instance Name
GPIO_PINMUX52

Offset
0x0D0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 486: GPIO_PINMUX Configuration Register (GPIO_PINMUX52)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.54 GPIO_PINMUX Configuration Register (GPIO_PINMUX53)

Instance Name
GPIO_PINMUX53

Offset
0x0D4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 487: GPIO_PINMUX Configuration Register (GPIO_PINMUX53)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 487: GPIO_PINMUX Configuration Register (GPIO_PINMUX53)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.55 GPIO_PINMUX Configuration Register (GPIO_PINMUX54)

Instance Name
GPIO_PINMUX54

Offset
0x0D8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 488: GPIO_PINMUX Configuration Register (GPIO_PINMUX54)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.56 GPIO_PINMUX Configuration Register (GPIO_PINMUX55)

Instance Name
GPIO_PINMUX55

Offset
0x0DC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 489: GPIO_PINMUX Configuration Register (GPIO_PINMUX55)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.57 GPIO_PINMUX Configuration Register (GPIO_PINMUX56)

Instance Name
GPIO_PINMUX56

Offset
0x0E0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 490: GPIO_PINMUX Configuration Register (GPIO_PINMUX56)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 490: GPIO_PINMUX Configuration Register (GPIO_PINMUX56)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.58 GPIO_PINMUX Configuration Register (GPIO_PINMUX57)

Instance Name

GPIO_PINMUX57

Offset

0x0E4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 491: GPIO_PINMUX Configuration Register (GPIO_PINMUX57)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.59 GPIO_PINMUX Configuration Register (GPIO_PINMUX58)

Instance Name
GPIO_PINMUX58

Offset
0x0E8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 492: GPIO_PINMUX Configuration Register (GPIO_PINMUX58)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.60 GPIO_PINMUX Configuration Register (GPIO_PINMUX59)

Instance Name
GPIO_PINMUX59

Offset
0x0EC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 493: GPIO_PINMUX Configuration Register (GPIO_PINMUX59)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 493: GPIO_PINMUX Configuration Register (GPIO_PINMUX59)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.61 GPIO_PINMUX Configuration Register (GPIO_PINMUX60)

Instance Name
GPIO_PINMUX60

Offset
0x0F0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 494: GPIO_PINMUX Configuration Register (GPIO_PINMUX60)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.62 GPIO_PINMUX Configuration Register (GPIO_PINMUX61)

Instance Name
GPIO_PINMUX61

Offset
0x0F4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 495: GPIO_PINMUX Configuration Register (GPIO_PINMUX61)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.63 GPIO_PINMUX Configuration Register (GPIO_PINMUX62)

Instance Name
GPIO_PINMUX62

Offset
0x0F8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 496: GPIO_PINMUX Configuration Register (GPIO_PINMUX62)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 496: GPIO_PINMUX Configuration Register (GPIO_PINMUX62)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.64 GPIO_PINMUX Configuration Register (GPIO_PINMUX63)

Instance Name

GPIO_PINMUX63

Offset

0x0FC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 497: GPIO_PINMUX Configuration Register (GPIO_PINMUX63)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.65 GPIO_PINMUX Configuration Register (GPIO_PINMUX64)

Instance Name
GPIO_PINMUX64

Offset
0x100

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 498: GPIO_PINMUX Configuration Register (GPIO_PINMUX64)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.66 GPIO_PINMUX Configuration Register (GPIO_PINMUX65)

Instance Name
GPIO_PINMUX65

Offset
0x104

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 499: GPIO_PINMUX Configuration Register (GPIO_PINMUX65)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 499: GPIO_PINMUX Configuration Register (GPIO_PINMUX65)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.67 GPIO_PINMUX Configuration Register (GPIO_PINMUX66)

Instance Name

GPIO_PINMUX66

Offset

0x108

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 500: GPIO_PINMUX Configuration Register (GPIO_PINMUX66)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.68 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED**Offset**
0x10C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																Reserved	Reserved								Reserved						
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 501: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.69 GPIO_PINMUX Configuration Register (GPIO_PINMUX68)**Instance Name**
GPIO_PINMUX68**Offset**
0x110

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved								DI_EN	FSEL_XR			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 502: GPIO_PINMUX Configuration Register (GPIO_PINMUX68)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 502: GPIO_PINMUX Configuration Register (GPIO_PINMUX68)

Bits	Name	Type	Reset	Description
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.70 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x114

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															Reserved	Reserved							Reserved								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 503: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.71 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x118

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved															Reserved			Reserved							Reserved							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 504: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 504: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.72 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x11C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															Reserved	Reserved										Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 505: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.73 GPIO_PINMUX Configuration Register (GPIO_PINMUX72)

Instance Name
GPIO_PINMUX72

Offset
0x120

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 506: GPIO_PINMUX Configuration Register (GPIO_PINMUX72)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.74 GPIO_PINMUX Configuration Register (GPIO_PINMUX73)

Instance Name
GPIO_PINMUX73

Offset
0x124

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 507: GPIO_PINMUX Configuration Register (GPIO_PINMUX73)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.75 GPIO_PINMUX Configuration Register (GPIO_PINMUX74)

Instance Name
GPIO_PINMUX74

Offset
0x128

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 508: GPIO_PINMUX Configuration Register (GPIO_PINMUX74)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.76 GPIO_PINMUX Configuration Register (GPIO_PINMUX75)

Instance Name
GPIO_PINMUX75

Offset
0x12C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 509: GPIO_PINMUX Configuration Register (GPIO_PINMUX75)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 509: GPIO_PINMUX Configuration Register (GPIO_PINMUX75)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.77 GPIO_PINMUX Configuration Register (GPIO_PINMUX76)

Instance Name

GPIO_PINMUX76

Offset

0x130

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 510: GPIO_PINMUX Configuration Register (GPIO_PINMUX76)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.78 GPIO_PINMUX Configuration Register (GPIO_PINMUX77)

Instance Name
GPIO_PINMUX77

Offset
0x134

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 511: GPIO_PINMUX Configuration Register (GPIO_PINMUX77)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.79 GPIO_PINMUX Configuration Register (GPIO_PINMUX78)

Instance Name
GPIO_PINMUX78

Offset
0x138

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved										DI_EN	FSEL_XR	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 512: GPIO_PINMUX Configuration Register (GPIO_PINMUX78)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 512: GPIO_PINMUX Configuration Register (GPIO_PINMUX78)

Bits	Name	Type	Reset	Description
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.80 GPIO_PINMUX Configuration Register (GPIO_PINMUX79)

Instance Name

GPIO_PINMUX79

Offset

0x13C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																PIO_PULL_SEL_R	PIO_PULLUP_R	PIO_PULLDN_R	Reserved							DI_EN	FSEL_XR				
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 513: GPIO_PINMUX Configuration Register (GPIO_PINMUX79)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15	PIO_PULL_SEL_R	R/W	0x0	Pull up/down enable. 1: pull-up and pull-down enable; 0: pull-up and pull-down disable.
14	PIO_PULLUP_R	R/W	0x0	Pull up enable. 1: pull-up enable; 0: pull-up disable.
13	PIO_PULLDN_R	R/W	0x0	Pull down enable. 1: pull-down enable; 0: pull-down disable.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	DI_EN	R/W	0x1	Control input enable, actively high. 1: input always enable; 0: input tri-stated.
2:0	FSEL_XR	R/W	0x0	Pinmux function. The function number is specified from 0 to 7.

A.16.81 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x140

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved															Reserved	Reserved								Reserved								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0

Table 514: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.82 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x144

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved															Reserved	Reserved								Reserved								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		?	0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0

Table 515: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.83 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x148

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																Reserved	Reserved										Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0

Table 516: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.84 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x14C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																Reserved	Reserved								Reserved							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		0	0	0	?	?	?	?	?	?	?	?	?	?	1	0	0

Table 517: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.85 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x150

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															Reserved	Reserved										Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		0	0	0	?	?	?	?	?	?	?	?	?	1	0	0	0

Table 518: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.16.86 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x154

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																Reserved	Reserved										Reserved				
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		0	0	0	?	?	?	?	?	?	?	?	?	1	0	0

Table 519: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:13	Reserved	R/W	0x0	Reserved. Do not change the reset value.
12:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x8	Reserved. Do not change the reset value.

A.17 WDT Register Information

Table 520: WDT Register Summary

Offset	Name	Description	Details
0x00	CR	Control Register	Page: 378
0x04	TORR	Timeout Range Register	Page: 379
0x08	CCVR	Current Counter Value Register	Page: 380
0x0C	CRR	Counter Restart Register	Page: 380
0x10	STAT	Interrupt Status Register	Page: 380
0x14	EOI	Interrupt Clear Register	Page: 381

A.17.1 Control Register (CR)

Control Register

Instance Name

CR

Offset

0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	RESERVED																												RPL			RMOD	EN
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0		

Table 521: Control Register (CR)

Bits	Name	Type	Reset	Description
31:5	RESERVED	R	0x0	Reserved and read as 0.
4:2	RPL	R/W	0x2	Reset pulse length. This is used to select the number of pclk cycles for which the system reset stays asserted. The range of values available is 2 to 256 pclk cycles. 0x0: 2 pclk cycles 0x1: 4 pclk cycles 0x2: 8 pclk cycles 0x3: 16 pclk cycles 0x4: 32 pclk cycles 0x5: 64 pclk cycles 0x6: 128 pclk cycles 0x7: 256 pclk cycles
1	RMOD	R/W	0x1	Response mode. Selects the output response generated to a timeout. 0x0: Generate a system reset. 0x1: First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset.

Table 521: Control Register (CR)

Bits	Name	Type	Reset	Description
0	EN	R/W	0x0	WDT enable. This bit is used to enable and disable the WDT. When disabled, the counter does not decrement. Thus, no interrupts or system resets are generated. Once this bit has been enabled, it can be cleared only by a system reset. 0x0: WDT disabled. 0x1: WDT enabled.

A.17.2 Timeout Range Register (TORR)

Timeout Range Register

Instance Name

TORR

Offset

0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RESERVED																							TOP_INIT				TOP				
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 522: Timeout Range Register (TORR)

Bits	Name	Type	Reset	Description
31:8	RESERVED	R	0x0	Reserved and read as 0.
7:4	TOP_INIT	R/W	0x0	Timeout period for initialization. Used to select the timeout period that the watchdog counter restarts from for the first counter restart (kick). This register should be written after reset and before the WDT is enabled. A change of the TOP_INIT is seen only once the WDT has been enabled, and any change after the first kick is not seen as subsequent kicks use the period specified by the TOP bits. The range of values available for watchdog counter are: Where $i = \text{TOP\_INIT}$ and $t = \text{timeout period}$ For $i = 0$ to 15 $t = 2^{(16 + i)}$
3:0	TOP	R/W	0x0	Timeout period. This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick). The range of values available for watchdog counter are: Where $i = \text{TOP}$ and $t = \text{timeout period}$ For $i = 0$ to 15 $t = 2^{(16 + i)}$

A.17.3 Current Counter Value Register (CCVR)

Current Counter Value Register.

Instance Name

CCVR

Offset

0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	CCVR																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 523: Current Counter Value Register (CCVR)

Bits	Name	Type	Reset	Description
31:0	CCVR	R	0xFFFF	This register, when read, is the current value of the internal counter.

A.17.4 Counter Restart Register (CRR)

Counter Restart Register.

Instance Name

CRR

Offset

0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																								CRR							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Table 524: Counter Restart Register (CRR)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:0	CRR	W	0x0	This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero.

A.17.5 Interrupt Status Register (STAT)

Interrupt Status Register.

Instance Name

STAT

Offset

0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															STAT
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 525: Interrupt Status Register (STAT)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	STAT	R	0x0	This register shows the interrupt status of the WDT. 0x1 : Interrupt is active regardless of polarity. 0x0 : Interrupt is inactive.

A.17.6 Interrupt Clear Register (EOI)

Interrupt Clear Register.

Instance Name

EOI

Offset

0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														EOI	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 526: Interrupt Clear Register (EOI)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	EOI	R	0x0	Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter.

A.18 RTC Register Information

Table 527: RTC Register Summary

Offset	Name	Description	Details
0x00	CNT_EN	Counter Enable Register	Page: 382
0x20	INT_RAW	Interrupt Raw Register	Page: 383
0x24	INT	Interrupt Register	Page: 384
0x28	INT_MSK	Interrupt Mask Register	Page: 384
0x40	CNT_CNTL	Counter Control Register	Page: 385
0x50	CNT_VAL	Counter Value Register	Page: 386
0x60	CNT_UPP_VAL	Counter Upper Value Register	Page: 386
0x80	CLK_CNTL	Clock control register	Page: 386

A.18.1 Counter Enable Register (CNT_EN)

Instance Name

CNT_EN

Offset

0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved													STS_RESETN	CNT_RST_DONE	CNT_RUN	Reserved										Reserved			CNT_RESET	CNT_STOP	CNT_START
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Table 528: Counter Enable Register (CNT_EN)

Bits	Name	Type	Reset	Description
31:19	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
18	STS_RESETN	R	0x0	System Reset Status CPU must poll this bit for a 1 before accessing any other registers. 0x0: The system reset is still asserted 0x1: The system reset is deasserted
17	CNT_RST_DONE	R	0x0	Counter Reset Done Status Writing 1 to CNT_RESET will set this bit to 0 until the counter finishes resetting. 0x0: The counter is still resetting 0x1: The counter has been reset

Table 528: Counter Enable Register (CNT_EN)

Bits	Name	Type	Reset	Description
16	CNT_RUN	R	0x0	Counter Enabled Status This bit can be polled to see when the counter is really enabled. 0x0: Counter is disabled 0x1: Counter is enabled
15:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2	CNT_RESET	W	0x0	Counter Reset 0x0: No action 0x1: Reset the counter. Counter is reset to 0. Poll CNT_RST_DONE to see when the counter has been reset. Do not write to any other registers before CNT_RST_DONE turns to 1.
1	CNT_STOP	W	0x0	Counter Disable 0x0: No action 0x1: Disable the counter. Poll CNT_RUN to see the counter status. if CNT_RUN is 0, it means that the counter has been disabled internally.
0	CNT_START	W	0x0	Counter Enable 0x0: No action 0x1: Enable the counter. Poll CNT_RUN to see the counter status. if CNT_RUN is 1, it means that the counter has been enabled internally.

A.18.2 Interrupt Raw Register (INT_RAW)

Instance Name
INT_RAW

Offset
0x20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															CNT_UPP_INT	Reserved															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Table 529: Interrupt Raw Register (INT_RAW)

Bits	Name	Type	Reset	Description
31:17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16	CNT_UPP_INT	R/W1CLR	0x0	Counter-Reach-Upper Interrupt Status It indicates that the counter has reached UPP_VAL. 0x0: Status cleared 0x1: Counter has reached UPP_VAL

Table 529: Interrupt Raw Register (INT_RAW)

Bits	Name	Type	Reset	Description
15:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.18.3 Interrupt Register (INT)

INT_MSK is combined with INT_RAW to determine the value of this register. Masked bits will be always 0, while unmasked bits will be the same value as the corresponding bits in INT_RAW register.

If any bit in this register is 1, an interrupt will be generated.

Instance Name

INT

Offset

0x24

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															CNT_UPP_INTR	Reserved															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Table 530: Interrupt Register (INT)

Bits	Name	Type	Reset	Description
31:17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16	CNT_UPP_INTR	R	0x0	Masked Signal of CNT_UPP_INTR
15:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.18.4 Interrupt Mask Register (INT_MSK)

Instance Name

INT_MSK

Offset

0x28

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															CNT_UPP_MSK	Reserved															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Table 531: Interrupt Mask Register (INT_MSK)

Bits	Name	Type	Reset	Description
31:17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16	CNT_UPP_MSK	R/W	0x1	CNT_UPP_INT Interrupt Mask 0x0: Do not mask CNT_UPP_INT 0x1: Mask interrupt CNT_UPP_INT
15:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.18.5 Counter Control Register (CNT_CNTL)

Instance Name
CNT_CNTL

Offset
0x40

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																						CNT_UPDT_MOD		Reserved			CNT_DBG_ACT	Reserved			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	0	?	?	?	?

Table 532: Counter Control Register (CNT_CNTL)

Bits	Name	Type	Reset	Description
31:10	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
9:8	CNT_UPDT_MOD	R/W	0x0	Counter Update Mode 0x0: Update off 0x1: Reserved. 0x2: Auto-update. 0x3: Reserved
7:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	CNT_DBG_ACT	R/W	0x0	Counter Action in Debug Mode 0x0: In debug mode, stop the counter 0x1: In debug mode, counter is not affected
3:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.18.6 Counter Value Register (CNT_VAL)

Instance Name
CNT_VAL

Offset
0x50

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	CNT_VAL																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 533: Counter Value Register (CNT_VAL)

Bits	Name	Type	Reset	Description
31:0	CNT_VAL	R	0x0	Counter Value This register displays the current counter value. The update method for CNT_VAL is chosen in CNT_UPDT_MOD.

A.18.7 Counter Upper Value Register (CNT_UPP_VAL)

Instance Name
CNT_UPP_VAL

Offset
0x60

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	UPP_VAL																															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Table 534: Counter Upper Value Register (CNT_UPP_VAL)

Bits	Name	Type	Reset	Description
31:0	UPP_VAL	R/W	0xFFFF_FFFF	Counter Upper Value Do not set UPP_VAL to 0. The reset value is the maximum value of the counter, where all bits are 1. In the event that the counter reaches this value (counter-reach-upper), the counter will overflow to 0. Setting this value to all 1s is equivalent to a free running up-counter. The value written to UPP_VAL will not be valid immediately. It will not be effective until the counter overflows. If you need the value is valid immediately, you can write 1 to CNT_RESET.

A.18.8 Clock control register (CLK_CNTL)

Instance Name
CLK_CNTL

Offset
0x80

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																				CLK_DIV				Reserved							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	?	?	?	?	?	?	?	?

Table 535: Clock control register (CLK_CNTL)

Bits	Name	Type	Reset	Description
31:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11:8	CLK_DIV	R/W	0x0	Clock Divider The frequency of the divided clock (f_div) is calculated from the frequency of the counter clock (f_clk) using this formula: $f\_div = f\_clk / (2 ^ CLK\_DIV)$
7:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19 PMU Register Information

Table 536: PMU Register Summary

Offset	Name	Description	Details
0x00	PWR_MODE	Power Mode Control Register	Page: 390
0x04	BOOT_JTAG	BOOT_JTAG Register	Page: 390
0x08	LAST_RST_CAUSE	Last Reset Cause Register	Page: 391
0x0C	LAST_RST_CLR	Last Reset Cause Clear Register	Page: 391
0x10	WAKE_SRC_CLR	Wake Up Source Clear Register	Page: 392
0x18	CLK_SRC	Clock Source Selection Register	Page: 392
0x20	PMIP_BRN_INT_SEL	PMIP Brown Interrupt Select	Page: 393
0x28	CLK_RDY	Clock Ready Register	Page: 393
0x2C	RC32M_CTRL	RC 32M Control Register	Page: 394
0x34	SFLL_CTRL1	SFLL Control Register 1	Page: 395
0x38	ANA_GRP_CTRL0	MAINXTAL Clock Request Register	Page: 395
0x3C	SFLL_CTRL0	SFLL Control Register 2	Page: 396
0x44	PWR_STAT	Power Status Register	Page: 397
0x48	PAD_CTRL0_REG	PAD Control Register0	Page: 397
0x4C	PAD_CTRL1_REG	PAD Control Register1	Page: 398
0x54	PMIP_BRN_CFG	Brownout Config Register	Page: 399
0x58	RSVD	PMU MISC Register	Page: 399
0x5C	ANA_GRP_CTRL1	BG Control Register	Page: 400
0x60	PMIP_PWR_CONFIG	Power Configure Register	Page: 400
0x64	PMIP_CHP_CTRL0	Chargepump Control Register0	Page: 401
0x68	PMIP_CHP_CTRL1	Chargepump Control Register1	Page: 401
0x78	AUPLL_CTRL0	USB and Audio PLL Control Register	Page: 402
0x7C	PERI_CLK_EN	Peripheral Clock Gating Register	Page: 402
0x80	UART_FAST_CLK_DIV	UART Fast Clock Div Register	Page: 404
0x84	UART_SLOW_CLK_DIV	UART Slow Clock Div Register	Page: 404
0x88	UART_CLK_SEL	UART Clock Select Register	Page: 404
0x8C	MCU_CORE_CLK_DIV	MCU CORE Clock Divider Ratio Register	Page: 405
0x90	PERI0_CLK_DIV	Peripheral0 Clock Divider Ratio Register	Page: 405

Table 536: PMU Register Summary

Offset	Name	Description	Details
0x94	PERI1_CLK_DIV	Peripheral1 Clock Divider Ratio Register	Page: 406
0x98	PERI2_CLK_DIV	Peripheral2 Clock Divider Ratio Register	Page: 408
0x9C	CAU_CLK_SEL	Select Signal for cau mclk	Page: 409
0xA0	WAKEUP_PUPD_CTRL	WAKE_UP PAD Pull_up/down Control in PM3/PM4	Page: 410
0xA4	IO_PAD_PWR_CFG	Power Config Register	Page: 410
0xA8	EXT_SEL_REG0	Extra Interrupt Select Register0	Page: 412
0xAC	EXT_SEL_REG1	Extra Interrupt Select Register1	Page: 415
0xB0	AUPLL_CTRL1	USB and Audio PLL Control Register	Page: 417
0xB4	AUPLL_CTRL2	USB and Audio PLL Control Register	Page: 417
0xB8	CAU_CTRL	CAU Clock Control Register	Page: 418
0xBC	RC32K_CTRL	RC32K Control Register	Page: 419
0xC0	XTAL32K_CTRL	XTAL32K Control Register	Page: 420
0xC4	PMIP_CMP_CTRL	Comparator Control Register	Page: 421
0xC8	PMIP_CONFIG0	PMIP Config Register 0.	Page: 422
0xCC	PMIP_CONFIG1	PMIP Config Register 1.	Page: 422
0xD0	PMIP_BRNDET_VBAT	Vbat Brownout Detection Control Register	Page: 423
0xD4	PMIP_CONFIG2	PMIP Config Register 2.	Page: 424
0xD8	PMIP_LDO_CTRL	LDO Control Register	Page: 425
0xDC	PERI_CLK_SRC	Peripheral Clock Source Register	Page: 425
0xE4	GPT0_CTRL	GPT0 Clock Control Register	Page: 426
0xE8	GPT1_CTRL	GPT1 Clock Control Register	Page: 427
0xEC	GPT2_CTRL	GPT2 Clock Control Register	Page: 428
0xF0	GPT3_CTRL	GPT3 Clock Control Register	Page: 429
0xF4	WAKEUP_EDGE_DETECT	Wakeup Edge Detect Register	Page: 430

A.19.1 Power Mode Control Register (PWR_MODE)

Instance Name: PWR_MODE
Offset: 0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														PWR_MODE	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 537: Power Mode Control Register (PWR_MODE)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1:0	PWR_MODE	R/W	0x0	Power mode control. The power mode transition only happens when the Cortex-M3(CM3) execute WFI, WFE or SLEEPONEXIT values Power mode 2'b00 PM0 or PM1 2'b01 PM2 2'b10 PM3 2'b11 PM4

A.19.2 BOOT_JTAG Register (BOOT_JTAG)

Instance Name: BOOT_JTAG
Offset: 0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														BOOT_MODE_REG	JTAG_EN
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	0

Table 538: BOOT_JTAG Register (BOOT_JTAG)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	BOOT_MODE_REG	R	0x1	1: boot from flash
0	JTAG_EN	R/W	0x0	1:enable jtag

A.19.3 Last Reset Cause Register (LAST_RST_CAUSE)

Instance Name Offset
LAST_RST_CAUSE 0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										WDT_RST	CM3_LOCKUP	CM3_SYSRESETREQ	Reserved	BROWNOUT_VBAT	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	0

Table 539: Last Reset Cause Register (LAST_RST_CAUSE)

Bits	Name	Type	Reset	Description
31:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5	WDT_RST	R	0x0	1: reset cause is watchdog timer
4	CM3_LOCKUP	R	0x0	1: reset cause is lockup
3	CM3_SYSRESETREQ	R	0x0	1: reset cause is system software reset request
2:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	BROWNOUT_VBAT	R	0x0	1: Vbat power brown out detected

A.19.4 Last Reset Cause Clear Register (LAST_RST_CLR)

Instance Name Offset
LAST_RST_CLR 0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										WDT_RST_CLR	CM3_LOCKUP_CLR	CM3_SYSRESETREQ_CLR	Reserved		BROWNOUT_VBAT_CLR
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?	?	0

Table 540: Last Reset Cause Clear Register (LAST_RST_CLR)

Bits	Name	Type	Reset	Description
31:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5	WDT_RST_CLR	R/W	0x0	Clear the watchdog timer reset request. Write 1 to clear.

Table 540: Last Reset Cause Clear Register (LAST_RST_CLR)

Bits	Name	Type	Reset	Description
4	CM3_LOCKUP_CLR	R/W	0x0	Clear the lockup request. Write 1 to clear.
3	CM3_SYSRESETRE Q_CLR	R/W	0x0	Clear the system reset request. Write 1 to clear.
2:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	BROWNOUT_VBAT_ CLR	R/W	0x0	Write 1 to clear.

A.19.5 Wake Up Source Clear Register (WAKE_SRC_CLR)

Instance Name
WAKE_SRC_CLR

Offset
0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														CLR_PIN_INT1	CLR_PIN_INT0
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 541: Wake Up Source Clear Register (WAKE_SRC_CLR)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	CLR_PIN_INT1	R/W	0x0	Clear the pin1 interrupt request. Write 1 to clear.
0	CLR_PIN_INT0	R/W	0x0	Clear the pin0 interrupt request. Write 1 to clear.

A.19.6 Clock Source Selection Register (CLK_SRC)

Instance Name
CLK_SRC

Offset
0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														SYS_CLK_SEL	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 542: Clock Source Selection Register (CLK_SRC)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1:0	SYS_CLK_SEL	R/W	0x0	value clock source 2'b00 SPLL 200MHz clock 2'b01 RC 32MHz clock 2'b10 MAINXTAL clock 2'b11 RC 32MHz clock

A.19.7 PMIP Brown Interrupt Select (PMIP_BRN_INT_SEL)

Instance Name
PMIP_BRN_INT_SEL

Offset
0x20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														PMIP_BRN_INT_SEL	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 543: PMIP Brown Interrupt Select (PMIP_BRN_INT_SEL)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	PMIP_BRN_INT_SEL	R/W	0x0	

A.19.8 Clock Ready Register (CLK_RDY)

Instance Name
CLK_RDY

Offset
0x28

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																									MAINXTAL_CLK_RDY	Reserved		X32K_RDY	RC32M_RDY	Reserved		PLL_CLK_RDY
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?	0	0	?	0	

Table 544: Clock Ready Register (CLK_RDY)

Bits	Name	Type	Reset	Description
31:7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
6	MAINXTAL_CLK_RDY	R	0x0	1: MAINXTAL clock is ready for use
5:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	X32K_RDY	R	0x0	1: Xtal 32k clock is ready for use
2	RC32M_RDY	R	0x0	1: RC 32m clock is ready for use
1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	PLL_CLK_RDY	R	0x0	1: SFLL clock is ready for use

A.19.9 RC 32M Control Register (RC32M_CTRL)

Instance Name

RC32M_CTRL

Offset

0x2C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														CAL_ALLOW	CAL_IN_PROGRESS
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 545: RC 32M Control Register (RC32M_CTRL)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	CAL_ALLOW	R/W	0x0	Allow calibration command from PMU
0	CAL_IN_PROGRESS	R	0x0	Asserts high when calibration is in progress

A.19.10 SPLL Control Register 1 (SPLL_CTRL1)Instance Name
SPLL_CTRL1Offset
0x34

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved											SPLL_REFDIV								SPLL_FBDIV								Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	0	0	1	0	1	0	0	0	0	0	0	0	1	1	1	1		1	0	0	1	1

Table 546: SPLL Control Register 1 (SPLL_CTRL1)

Bits	Name	Type	Reset	Description
31:21	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
20:12	SPLL_REFDIV	R/W	0x50	Reference Divider Default Value set for INPUT clock = 32MHz
11:1	SPLL_FBDIV	R/W	0x1F3	Feedback Divider Default Value set for output clock = 200MHz
0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.11 MAINXTAL Clock Request Register (ANA_GRP_CTRL0)Instance Name
ANA_GRP_CTRL0Offset
0x38

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved							SPLL_READY_DET_LOW										SPLL_READY_DET_HIGH										PU		PU_XTAL	Reserved	
Default	?	?	?	?	?	?	?	0	0	1	1	1	1	0	1	0	0	1	0	0	1	1	1	1	1	1	1	0	1	1	1	?

Table 547: MAINXTAL Clock Request Register (ANA_GRP_CTRL0)

Bits	Name	Type	Reset	Description
31:25	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
24:14	SPLL_READY_DET_LOW	R/W	0x1E9	READY LOWER BOUND Based on FBDIV[1:0], value = FBDIV[1:0] * 0.98 (round down)
13:3	SPLL_READY_DET_HIGH	R/W	0x1FD	Threshold for PLL READY Higher BOUND Based on FBDIV[10:0], value = FBDIV * 1.02 (round up)
2	PU	R/W	0x1	Power-up signal for the whole block 1: power up, 0: power down
1	PU_XTAL	R/W	0x1	Power-up signal for the MAINXTAL OSC circuit. 1: power up, 0: power down
0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.12 SPLL Control Register 2 (SPLL_CTRL0)

Instance Name
SPLL_CTRL0

Offset
0x3C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved					SFLL_LOCK	SFLL_REFCLK_SEL	Reserved		SFLL_KVCO			Reserved			SFLL_DIV_SEL		Reserved												SFLL_PU		
Default	?	?	?	?	?	0	0	?	?	1	1	1	?	?	?	?	?	0	0	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 548: SPLL Control Register 2 (SPLL_CTRL0)

Bits	Name	Type	Reset	Description
31:27	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
26	SPLL_LOCK	R	0x0	1: PLL module is locked; 0: PLL module is unlocked
25	SPLL_REFCLK_SEL	R/W	0x0	1: select MAINXTAL output, 0: select RC32M output
24:23	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
22:20	SPLL_KVCO	R/W	0x7	Select VCO Running Range Default value for output clock=200M value VCO frequency 3'b000 100M - 110M 3'b001 110M - 121M 3'b010 121M - 133M 3'b011 133M - 146M 3'b100 146M - 160M 3'b101 160M - 175M 3'b110 175M - 192M 3'b111 192M - 200M
19:15	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
14:13	SPLL_DIV_SEL	R/W	0x0	Post Divider value post divisor 2'b00 divide by 1 2'b01 divide by 2 2'b10 divide by 4 2'b11 divide by 8
12:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	SPLL_PU	R/W	0x0	Power-up signal for the Flock 0x0: power down 0x1: power up

A.19.13 Power Status Register (PWR_STAT)

Instance Name Offset
PWR_STAT 0x44

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																							AV18_RDY	VDD_MCU_RDY		VDD_CAU_RDY		VDD_MEM_RDY		VDD_VFL_RDY		
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	

Table 549: Power Status Register (PWR_STAT)

Bits	Name	Type	Reset	Description
31:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7	AV18_RDY	R	0x0	av18_rdy
6:5	VDD_MCU_RDY	R	0x0	vdd_mcu_rdy
4:3	VDD_CAU_RDY	R	0x0	vdd_cau_rdy
2:1	VDD_MEM_RDY	R	0x0	vdd_mem_rdy
0	VDD_VFL_RDY	R	0x0	vdd_vfl_rdy

A.19.14 PAD Control Register0 (PAD_CTRL0_REG)

Instance Name Offset
PAD_CTRL0_REG 0x48

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											TDO_CTRL	XTAL32K_OUT_CTRL	XTAL32K_IN_CTRL	Reserved	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	?

Table 550: PAD Control Register0 (PAD_CTRL0_REG)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	TDO_CTRL	R/W	0x0	TDO PAD control: 0-normal mode 1-power saving mode
2	XTAL32K_OUT_CTRL	R/W	0x0	XTAL32K_OUT PAD control: 0-normal mode 1-power saving mode

Table 550: PAD Control Register0 (PAD_CTRL0_REG)

Bits	Name	Type	Reset	Description
1	XTAL32K_IN_CTRL	R/W	0x0	XTAL32K_IN PAD control: 0-normal mode 1-power saving mode
0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.15 PAD Control Register1 (PAD_CTRL1_REG)

Instance Name
PAD_CTRL1_REG

Offset
0x4C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											WAKEUP1_CTRL	WAKEUP0_CTRL	Reserved	GPIO_27_CTRL	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	0	

Table 551: PAD Control Register1 (PAD_CTRL1_REG)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	WAKEUP1_CTRL	R/W	0x0	WAKE_UP1 PAD control: 0-normal pin muxing 1-output xtal32k clock
2	WAKEUP0_CTRL	R/W	0x0	WAKE_UP0 PAD control: 0-normal pin muxing 1-output xtal32k clock
1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	GPIO_27_CTRL	R/W	0x0	GPIO_27 PAD control: 0-normal pin muxing 1-output xtal32k clock

A.19.16 Brownout Config Register (PMIP_BRN_CFG)

Instance Name
PMIP_BRN_CFG

Offset
0x54

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																													BRNDET_VBAT_RST_EN	Reserved	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?

Table 552: Brownout Config Register (PMIP_BRN_CFG)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	BRNDET_VBAT_RST_EN	R/W	0x0	1: enable Vbat brownout reset, 0: disable vbat brownout reset
0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.17 PMU MISC Register (RSVD)

Instance Name
RSVD

Offset
0x58

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RESERVED_OUT																													AUPLL LOCK STATUS	CAU CLOCK GATE	Reserved
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?

Table 553: PMU MISC Register (RSVD)

Bits	Name	Type	Reset	Description
31:3	RESERVED_OUT	R/W	0x0	
2	AUPLL LOCK STATUS	R	0x0	1: AUPLL is lock and ready to use
1	CAU CLOCK GATE	R/W	0x0	1: shut off pmu generated cau clock
0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.18 BG Control Register (ANA_GRP_CTRL1)

Instance Name
ANA_GRP_CTRL1

Offset
0x5C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																				BYPASS	Reserved											
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		?	0	?	?	?	?	?	?	?	?	?	?

Table 554: BG Control Register (ANA_GRP_CTRL1)

Bits	Name	Type	Reset	Description
31:11	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
10	BYPASS	R/W	0x0	MAINXTAL OSC bypass control signal. 1: use external clock
9:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.19 Power Configure Register (PMIP_PWR_CONFIG)

Instance Name
PMIP_PWR_CONFIG

Offset
0x60

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																													AV18_EXT	Reserved	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?

Table 555: Power Configure Register (PMIP_PWR_CONFIG)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2	AV18_EXT	R/W	0x0	Assert high if external DC/DC chip will provide AV18=1.8V during PM0/1 modes
1:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.20 Chargepump Control Register0 (PMIP_CHP_CTRL0)

Instance Name
PMIP_CHP_CTRL0

Offset
0x64

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																DEL_V12_SEL	Reserved															
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		1	1	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Table 556: Chargepump Control Register0 (PMIP_CHP_CTRL0)

Bits	Name	Type	Reset	Description
31:16	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
15:14	DEL_V12_SEL	R/W	0x3	V12 power delay control. 00: short delay;11: long delay
13:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.21 Chargepump Control Register1 (PMIP_CHP_CTRL1)

Instance Name
PMIP_CHP_CTRL1

Offset
0x68

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										CHP_SPREADSP	Reserved				
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		0	0	?	?	?

Table 557: Chargepump Control Register1 (PMIP_CHP_CTRL1)

Bits	Name	Type	Reset	Description
31:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5:4	CHP_SPREADSP	R/W	0x0	Enable spread-spectrum for CHP V12 reference
3:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.22 USB and Audio PLL Control Register (AUPLL_CTRL0)

Instance Name Offset
AUPLL_CTRL0 0x78

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																	PU	REFDIV					FBDIV								
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0

Table 558: USB and Audio PLL Control Register (AUPLL_CTRL0)

Bits	Name	Type	Reset	Description
31:15	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
14	PU	R/W	0x0	Power up signals for the PLL. 1: power up, 0: power down
13:9	REFDIV	R/W	0x8	Reference clock divider select
8:0	FBDIV	R/W	0x100	Feedback clock divider select

A.19.23 Peripheral Clock Gating Register (PERI_CLK_EN)

Instance Name Offset
PERI_CLK_EN 0x7C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved				USBC_CLK_EN	Reserved	SDIO_CLK_EN	QSPI1_CLK_EN	WDT_CLK_EN	GPT3_CLK_EN	GPT2_CLK_EN	I2C2_CLK_EN	I2C1_CLK_EN	Reserved	SSP2_CLK_EN	UART3_CLK_EN	UART2_CLK_EN	Reserved			GPT1_CLK_EN	GPT0_CLK_EN	SSP1_CLK_EN	SSP0_CLK_EN	I2C0_CLK_EN	UART1_CLK_EN	UART0_CLK_EN	GPIO_CLK_EN	RTC_CLK_EN	Reserved	QSPI0_CLK_EN	Reserved
Default	?	?	?	?	0	?	0	0	1	1	1	1	1	?	1	0	0	?	?	?	1	1	1	1	1	0	0	1	1	?	0	?

Table 559: Peripheral Clock Gating Register (PERI_CLK_EN)

Bits	Name	Type	Reset	Description
31:28	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
27	USBC_CLK_EN	R/W	0x0	usbc clock gate enable, 1: clock is disable, 0: clock is enabled
26	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
25	SDIO_CLK_EN	R/W	0x0	sdio clock gate enable, 1: clock is disable, 0: clock is enabled
24	QSPI1_CLK_EN	R/W	0x0	qspi1 clock gate enable, 1: clock is disable, 0: clock is enabled
23	WDT_CLK_EN	R/W	0x1	wdt clock gate enable, 1: clock is disable, 0: clock is enabled

Table 559: Peripheral Clock Gating Register (PERI_CLK_EN)

Bits	Name	Type	Reset	Description
22	GPT3_CLK_EN	R/W	0x1	gpt3 clock gate enable, 1: clock is disable, 0: clock is enabled
21	GPT2_CLK_EN	R/W	0x1	gpt2 clock gate enable, 1: clock is disable, 0: clock is enabled
20	I2C2_CLK_EN	R/W	0x1	i2c2 clock gate enable, 1: clock is disable, 0: clock is enabled
19	I2C1_CLK_EN	R/W	0x1	i2c1 clock gate enable, 1: clock is disable, 0: clock is enabled
18	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
17	SSP2_CLK_EN	R/W	0x1	ssp2 clock gate enable, 1: clock is disable, 0: clock is enabled
16	UART3_CLK_EN	R/W	0x0	uart3 clock gate enable, 1: clock is disable, 0: clock is enabled
15	UART2_CLK_EN	R/W	0x0	uart2 clock gate enable, 1: clock is disable, 0: clock is enabled
14:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	GPT1_CLK_EN	R/W	0x1	gpt1 clock gate enable, 1: clock is disable, 0: clock is enabled
10	GPT0_CLK_EN	R/W	0x1	gpt0 clock gate enable, 1: clock is disable, 0: clock is enabled
9	SSP1_CLK_EN	R/W	0x1	ssp1 clock gate enable, 1: clock is disable, 0: clock is enabled
8	SSP0_CLK_EN	R/W	0x1	ssp0 clock gate enable, 1: clock is disable, 0: clock is enabled
7	I2C0_CLK_EN	R/W	0x1	i2c0 clock gate enable, 1: clock is disable, 0: clock is enabled
6	UART1_CLK_EN	R/W	0x0	uart1 clock gate enable, 1: clock is disable, 0: clock is enabled
5	UART0_CLK_EN	R/W	0x0	uart0 clock gate enable, 1: clock is disable, 0: clock is enabled
4	GPIO_CLK_EN	R/W	0x1	GPIO clock gate enable, 1: clock is disable, 0: clock is enabled
3	RTC_CLK_EN	R/W	0x1	rtc clock gate enable, 1: clock is disable, 0: clock is enabled
2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	QSPI0_CLK_EN	R/W	0x0	qspi0 clock gate enable, 1: clock is disable, 0: clock is enabled
0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.24 UART Fast Clock Div Register (UART_FAST_CLK_DIV)

Instance Name
UART_FAST_CLK_DIV

Offset
0x80

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								NOMINATOR													DENOMINATOR										
Default	?	?	?	?	?	?	?	?	1	0	0	0	0	0	1	1	1	0	0	1	0	1	0	0	1	1	1	0	0	0	1	1

Table 560: UART Fast Clock Div Register (UART_FAST_CLK_DIV)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23:11	NOMINATOR	R/W	0x1072	13bits nominator for fraction divider
10:0	DENOMINATOR	R/W	0x4E3	11bits denominator for fractional divider

A.19.25 UART Slow Clock Div Register (UART_SLOW_CLK_DIV)

Instance Name
UART_SLOW_CLK_DIV

Offset
0x84

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								NOMINATOR													DENOMINATOR										
Default	?	?	?	?	?	?	?	?	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0

Table 561: UART Slow Clock Div Register (UART_SLOW_CLK_DIV)

Bits	Name	Type	Reset	Description
31:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23:11	NOMINATOR	R/W	0x7A1	13bits nominator for fraction divider
10:0	DENOMINATOR	R/W	0x90	11bits denominator for fractional divider

A.19.26 UART Clock Select Register (UART_CLK_SEL)

Instance Name
UART_CLK_SEL

Offset
0x88

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																												UART3_CLK_SEL	UART2_CLK_SEL	UART1_CLK_SEL	UART0_CLK_SEL
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	

Table 562: UART Clock Select Register (UART_CLK_SEL)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	UART3_CLK_SEL	R/W	0x0	uart3 apb1 UART clk sel: 1-fast 0-slow
2	UART2_CLK_SEL	R/W	0x0	uart2 apb1 UART clk sel: 1-fast 0-slow
1	UART1_CLK_SEL	R/W	0x0	uart1 apb0 UART clk sel: 1-fast 0-slow
0	UART0_CLK_SEL	R/W	0x0	uart0 apb0 UART clk sel: 1-fast 0-slow

A.19.27 MCU CORE Clock Divider Ratio Register (MCU_CORE_CLK_DIV)

Instance Name

MCU_CORE_CLK_DIV

Offset

0x8C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										FCLK_DIV					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	1

Table 563: MCU CORE Clock Divider Ratio Register (MCU_CORE_CLK_DIV)

Bits	Name	Type	Reset	Description
31:6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
5:0	FCLK_DIV	R/W	0x1	divisor for hclk, fclk, AHB clocks values divisor 6'h00 divisor = 1 other divisor = fclk_div[5:0]

A.19.28 Peripheral0 Clock Divider Ratio Register (PERI0_CLK_DIV)

Instance Name

PERI0_CLK_DIV

Offset

0x90

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												SDIO_CLK_DIV			Reserved		SSP2_CLK_DIV			SSP1_CLK_DIV			SSP0_CLK_DIV								
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	1	?	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0

Table 564: Peripheral0 Clock Divider Ratio Register (PERI0_CLK_DIV)

Bits	Name	Type	Reset	Description
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:16	SDIO_CLK_DIV	R/W	0x1	sdio clk divisor values divisor 4'h0 divisor = 1 other divisor = sdio_clk_div[19:16]
15	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
14:10	SSP2_CLK_DIV	R/W	0x2	ssp2 abp1 clk divisor, values divisor 5'h00 divisor = 1 other divisor = ssp2_clk_div[14:10]
9:5	SSP1_CLK_DIV	R/W	0x2	ssp1 abp0 clk divisor, divisor =ssp1_clk_div values divisor 5'h0 0 divisor = 1 other divisor = ssp1_clk_div[9:5]
4:0	SSP0_CLK_DIV	R/W	0x2	ssp0 abp0 clk divisor, divisor =ssp0_clk_div values divisor 5'h00 divisor = 1 other divisor = ssp0_clk_div[4:0]

A.19.29 Peripheral1 Clock Divider Ratio Register (PERI1_CLK_DIV)

Instance Name
PERI1_CLK_DIV

Offset
0x94

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												APB1_CLK_DIV		APB0_CLK_DIV		Reserved	QSPI1_CLK_DIV			Reserved	QSPI0_CLK_DIV			Reserved				PMU_CLK_DIV			
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	?	0	0	1	?	0	0	1	?	?	?	?	0	0	0	1

Table 565: Peripheral1 Clock Divider Ratio Register (PERI1_CLK_DIV)

Bits	Name	Type	Reset	Description
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 565: Peripheral1 Clock Divider Ratio Register (PERI1_CLK_DIV)

Bits	Name	Type	Reset	Description
19:18	APB1_CLK_DIV	R/W	0x0	value apb1 clock divisor 2'b00 divisor=1 2'b01 divisor=2 2'b10 divisor=4 2'b11 divisor=8
17:16	APB0_CLK_DIV	R/W	0x0	apb0 clk divisor value apb0 clock divisor 2'b00 divisor=1 2'b01 divisor=2 2'b10 divisor=4 2'b11 divisor=8
15	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
14:12	QSPI1_CLK_DIV	R/W	0x1	qspi1 function clock divisor values divisor 3'b000 divisor = 1 other divisor = qspi1_clk_div[14:12]
11	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
10:8	QSPI0_CLK_DIV	R/W	0x1	qspi0 function clock divisor values divisor 3'b000 divisor = 1 other divisor = qspi0_clk_div[10:8]
7:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	PMU_CLK_DIV	R/W	0x1	pmu clk divisor, values divisor 4'b0000 divisor = 1 other divisor = pmu_clk_div[3:0]

A.19.30 Peripheral2 Clock Divider Ratio Register (PERI2_CLK_DIV)

Instance Name
PERI2_CLK_DIV

Offset
0x98

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Field	Reserved			WDT_CLK_DIV_2_2		Reserved		WDT_CLK_DIV_1_0		Reserved		I2C_CLK_DIV		Reserved				GPT3_CLK_DIV_5_3			Reserved		GPT3_CLK_DIV_2_0			Reserved		WDT_CLK_DIV_5_3			Reserved		GPT_SAMPLE_CLK_DIV	
Default	?	?	?	0	?	?	0	0	?	?	0	1	?	?	?	?	?	0	0	0	?	0	0	0	?	0	0	0	?	0	0	1		

Table 566: Peripheral2 Clock Divider Ratio Register (PERI2_CLK_DIV)

Bits	Name	Type	Reset	Description
31:29	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
28	WDT_CLK_DIV_2_2	R/W	0x0	please see detail in bit[25:24]
27:26	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
25:24	WDT_CLK_DIV_1_0	R/W	0x0	WDT clk divisor bit[6:4], bit[28], bit[25:24] combine to become a 6-bit WDT clock divisor. values divisor 6'b00 divisor = 1 other divisor = 2^value
23:22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21:20	I2C_CLK_DIV	R/W	0x1	i2c function clk divisor, divisor = i2c_clk_div values divisor 2'b00 divisor = 1 other divisor = i2c_clk_div[21:20]
19:15	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
14:12	GPT3_CLK_DIV_5_3	R/W	0x0	please see detail in bit[10:8]
11	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
10:8	GPT3_CLK_DIV_2_0	R/W	0x0	gpt3 clock divisor[2:0] bit[14:12], bit[10:8] combine to become a 6-bit GPT3 clock divisor. values divisor 6'b00 divisor = 1 other divisor = gpt3 clock divisor[5:0]

Table 566: Peripheral2 Clock Divider Ratio Register (PERI2_CLK_DIV)

Bits	Name	Type	Reset	Description
7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
6:4	WDT_CLK_DIV_5_3	R/W	0x0	please see detail in bit[25:24]
3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2:0	GPT_SAMPLE_CLK_DIV	R/W	0x1	gpt sample clk divisor

A.19.31 Select Signal for cau mclk (CAU_CLK_SEL)

Instance Name
CAU_CLK_SEL

Offset
0x9C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														CAU_CLK_SEL	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 567: Select Signal for cau mclk (CAU_CLK_SEL)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1:0	CAU_CLK_SEL	R/W	0x0	select signal for cau mclk values Clock Source 2'b00 RC32MHz clock 2'b01 MAINXTAL clock 2'b10 Aupll 30M clock 2'b11 Aupll 30M clock

A.19.32 WAKE_UP PAD Pull_up/down Control in PM3/PM4 (WAKEUP_PUPD_CTRL)

Instance Name
WAKEUP_PUPD_CTRL

Offset
0xA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																													WAKEUP1_PUPD_CTRL	WAKEUP0_PUPD_CTRL	Reserved
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	?

Table 568: WAKE_UP PAD Pull_up/down Control in PM3/PM4 (WAKEUP_PUPD_CTRL)

Bits	Name	Type	Reset	Description
31:3	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
2	WAKEUP1_PUPD_CTRL	R/W	0x1	WAKE_UP1 PAD pull_up/down control:0-pull up 1-pull down
1	WAKEUP0_PUPD_CTRL	R/W	0x1	WAKE_UP0 PAD pull_up/down control:0-pull up 1-pull down
0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.33 Power Config Register (IO_PAD_PWR_CFG)

Instance Name
IO_PAD_PWR_CFG

Offset
0xA4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Field	Reserved						POR_LVL_FL_LOW_Vddb_CORE	Reserved			VDDO_FL_REG_PDB_CORE	POR_LVL_GPIO0_LOW_Vddb_CORE	V18EN_LVL_GPIO0_V18EN_CORE	Reserved		VDD_IO1_REG_PDB_CORE	Reserved		V18EN_LVL_AON_V18EN_CORE	Reserved		VDD_IO2_REG_PDB_CORE	POR_LVL_GPIO1_LOW_Vddb_CORE	V18EN_LVL_GPIO1_V18EN_CORE	Reserved		VDD_IO4_REG_PDB_CORE	Reserved		VDD_IO6_REG_PDB_CORE	POR_LVL_SDIO_LOW_Vddb_CORE	V18EN_LVL_SDIO_V18EN_CORE	Reserved		VDD_IO7_REG_PDB_CORE	POR_LVL_GPIO2_LOW_Vddb_CORE	V18EN_LVL_GPIO2_V18EN_CORE	Reserved		VDD_IO9_REG_PDB_CORE
Default	?	?	?	?	?	?	0	?	?	1	0	0	?	1	?	0	?	1	0	0	?	1	?	1	0	0	?	?	1	0	0	?	1	0	0	?	1			

Table 569: Power Config Register (IO_PAD_PWR_CFG)

Bits	Name	Type	Reset	Description
31:26	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
25	POR_LVL_FL_LOW_VDDB_CORE	R/W	0x0	0: FLASH domain Power is off. 1: Flash domain power is on.
24:23	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
22	VDDO_FL_REG_PD_B_CORE	R/W	0x1	1: FLASH Pad in Normal mode. 0: Flash Pad in Power down mode.
21	POR_LVL_GPIO0_LOW_VDDB_CORE	R/W	0x0	0: GPIO_D0 domain Power is off. 1: GPIO_D0 domain Power is on.
20	V18EN_LVL_GPIO0_V18EN_CORE	R/W	0x0	0: 3.3v, 1:1.8v. The bit configuration must be set to match the GPIO_D0 Domain power supply.
19	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
18	VDD_IO1_REG_PD_B_CORE	R/W	0x1	1: Pad of GPIO_D0 domain is Normal mode. 0: Pad of GPIO_D0 domain is Power down mode.
17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16	V18EN_LVL_AON_V18EN_CORE	R/W	0x0	0: 3.3v, 1:1.8v. The bit configuration must be set to match the AON Domain power supply.
15	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
14	VDD_IO2_REG_PD_B_CORE	R/W	0x1	1: Pad of AON domain is Normal mode. 0: Pad of AON domain is Power down mode.
13	POR_LVL_GPIO1_LOW_VDDB_CORE	R/W	0x0	0: GPIO_D1 domain Power is off. 1: GPIO_D1 domain Power is on.
12	V18EN_LVL_GPIO1_V18EN_CORE	R/W	0x0	0: 3.3v, 1:1.8v. The bit configuration must be set to match the GPIO_D1 Domain power supply.
11	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
10	VDD_IO4_REG_PD_B_CORE	R/W	0x1	1: Pad of GPIO_D1 domain (GPIO_28~GPIO_39) is Normal mode. 0: Pad of GPIO_D1 domain (GPIO_28~GPIO_39) is Power down mode.
9	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
8	VDD_IO6_REG_PD_B_CORE	R/W	0x1	1: Pad of GPIO_D1 domain (GPIO40~GPIO50) is Normal mode. 0: Pad of GPIO_D1 domain (GPIO40~GPIO50) is Power down mode.
7	POR_LVL_SDIO_LOW_VDDB_CORE	R/W	0x0	0: SDIO domain Power is off. 1: SDIO domain Power is on.
6	V18EN_LVL_SDIO_V18EN_CORE	R/W	0x0	0: 3.3v, 1:1.8v. The bit configuration must be set to match the SDIO Domain power supply.
5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4	VDD_IO7_REG_PD_B_CORE	R/W	0x1	1: Pad of SDIO domain is Normal mode. 0: Pad of SDIO domain is Power down mode.

Table 569: Power Config Register (IO_PAD_PWR_CFG)

Bits	Name	Type	Reset	Description
3	POR_LVL_GPIO2_L OW_VDDDB_CORE	R/W	0x0	0: GPIO_D2 domain Power is off. 1: GPIO_D2 domain Power is on.
2	V18EN_LVL_GPIO2_ V18EN_CORE	R/W	0x0	0: 3.3v, 1:1.8v. The bit configuration must be set to match the GPIO_D2 Domain power supply.
1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	VDD_IO9_REG_PD B_CORE	R/W	0x1	1: Pad of GPIO_D2 domain is Normal mode. 0: Pad of GPIO_D2 domain is Power down mode.

A.19.34 Extra Interrupt Select Register0 (EXT_SEL_REG0)

Instance Name
EXT_SEL_REG0

Offset
0xA8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SEL_49		SEL_48		SEL_47		SEL_46		Reserved		SEL_44		SEL_43		SEL_42		SEL_41		SEL_40		SEL_39		Reserved		SEL_37		SEL_36		SEL_35		SEL_34	
Default	0	0	0	0	0	0	0	0	?	?	0	0	0	0	0	0	0	0	0	0	0	0	?	?	0	0	0	0	0	0	0	0

Table 570: Extra Interrupt Select Register0 (EXT_SEL_REG0)

Bits	Name	Type	Reset	Description
31:30	SEL_49	R/W	0x0	values GPIO number 2'b00 RESERVED 2'b01 RESERVED 2'b10 GPIO_50 2'b11 GPIO_50
29:28	SEL_48	R/W	0x0	values GPIO number 2'b00 GPIO_45 2'b01 RESERVED 2'b10 RESERVED 2'b11 RESERVED
27:26	SEL_47	R/W	0x0	values GPIO number 2'b00 GPIO_42 2'b01 GPIO_43 2'b10 GPIO_44 2'b11 GPIO_44

Table 570: Extra Interrupt Select Register0 (EXT_SEL_REG0)

Bits	Name	Type	Reset	Description
25:24	SEL_46	R/W	0x0	values GPIO number 2'b00 RESERVED 2'b01 GPIO_40 2'b10 GPIO_41 2'b11 GPIO_41
23:22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21:20	SEL_44	R/W	0x0	values GPIO number 2'b00 GPIO_33 2'b01 GPIO_34 2'b10 GPIO_35 2'b11 GPIO_35
19:18	SEL_43	R/W	0x0	values GPIO number 2'b00 GPIO_30 2'b01 RESERVED 2'b10 GPIO_32 2'b11 GPIO_32
17:16	SEL_42	R/W	0x0	values GPIO number 2'b00 GPIO_24 2'b01 GPIO_28 2'b10 GPIO_29 2'b11 GPIO_29
15:14	SEL_41	R/W	0x0	values GPIO number 2'b00 GPIO_21 2'b01 GPIO_22 2'b10 GPIO_23 2'b11 GPIO_23
13:12	SEL_40	R/W	0x0	values GPIO number 2'b00 GPIO_18 2'b01 GPIO_19 2'b10 GPIO_20 2'b11 GPIO_20

Table 570: Extra Interrupt Select Register0 (EXT_SEL_REG0)

Bits	Name	Type	Reset	Description
11:10	SEL_39	R/W	0x0	values GPIO number 2'b00 RESERVED 2'b01 GPIO_16 2'b10 GPIO_17 2'b11 GPIO_17
9:8	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
7:6	SEL_37	R/W	0x0	values GPIO number 2'b00 GPIO_9 2'b01 GPIO_10 2'b10 GPIO_11 2'b11 GPIO_11
5:4	SEL_36	R/W	0x0	values GPIO number 2'b00 GPIO_6 2'b01 GPIO_7 2'b10 GPIO_8 2'b11 GPIO_8
3:2	SEL_35	R/W	0x0	values GPIO number 2'b00 GPIO_3 2'b01 GPIO_4 2'b10 GPIO_5 2'b11 GPIO_5
1:0	SEL_34	R/W	0x0	values GPIO number 2'b00 GPIO_0 2'b01 GPIO_1 2'b10 GPIO_2 2'b11 GPIO_2

A.19.35 Extra Interrupt Select Register1 (EXT_SEL_REG1)

Instance Name Offset
EXT_SEL_REG1 0xAC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												SEL_59		SEL_58		SEL_57		Reserved		SEL_55		SEL_54		SEL_53		SEL_52		SEL_51		SEL_50	
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	?	?	0	0	0	0	0	0	0	0	0	0	0	0

Table 571: Extra Interrupt Select Register1 (EXT_SEL_REG1)

Bits	Name	Type	Reset	Description
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	SEL_59	R/W	0x0	select signal for extra interrupt 59,3:1 values GPIO number 2'b00 GPIO_78 2'b01 GPIO_79 2'b10 GPIO_79 2'b11 GPIO_79
17:16	SEL_58	R/W	0x0	values GPIO number 2'b00 GPIO_75 2'b01 GPIO_76 2'b10 GPIO_77 2'b11 GPIO_77
15:14	SEL_57	R/W	0x0	values GPIO number 2'b00 GPIO_72 2'b01 GPIO_73 2'b10 GPIO_74 2'b11 GPIO_74
13:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11:10	SEL_55	R/W	0x0	values GPIO number 2'b00 GPIO_66 2'b01 RESERVED 2'b10 GPIO_68 2'b11 GPIO_68

Table 571: Extra Interrupt Select Register1 (EXT_SEL_REG1)

Bits	Name	Type	Reset	Description
9:8	SEL_54	R/W	0x0	values GPIO number 2'b00 GPIO_63 2'b01 GPIO_64 2'b10 GPIO_65 2'b11 GPIO_65
7:6	SEL_53	R/W	0x0	values GPIO number 2'b00 GPIO_60 2'b01 GPIO_61 2'b10 GPIO_62 2'b11 GPIO_62
5:4	SEL_52	R/W	0x0	values GPIO number 2'b00 GPIO_57 2'b01 GPIO_58 2'b10 GPIO_59 2'b11 GPIO_59
3:2	SEL_51	R/W	0x0	values GPIO number 2'b00 GPIO_54 2'b01 GPIO_55 2'b10 GPIO_56 2'b11 GPIO_56
1:0	SEL_50	R/W	0x0	values GPIO number 2'b00 GPIO_51 2'b01 GPIO_52 2'b10 GPIO_53 2'b11 GPIO_53

A.19.36 USB and Audio PLL Control Register (AUPLL_CTRL1)

Instance Name
AUPLL_CTRL1

Offset
0xB0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved		PI_EN	Reserved					UPDATE_SEL	Reserved				CLK_DET_EN	RESET_INTP_EXT	FREQ_OFFSET																
Default	?	?	0	?	?	?	?	?	1	?	?	?	?	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 572: USB and Audio PLL Control Register (AUPLL_CTRL1)

Bits	Name	Type	Reset	Description
31:30	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
29	PI_EN	R/W	0x0	Enable signal for INTP block. 0: INTP is off, 1: INTP is on
28:24	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
23	UPDATE_SEL	R/W	0x1	1:lower update_rate 0: higher update_rate
22:19	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
18	CLK_DET_EN	R/W	0x1	Enables PI output clock for internal reset circuit
17	RESET_INTP_EXT	R/W	0x0	External interpolator reset signal
16:0	FREQ_OFFSET	R/W	0x0	Frequency offset setting

A.19.37 USB and Audio PLL Control Register (AUPLL_CTRL2)

Instance Name
AUPLL_CTRL2

Offset
0xB4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved					CLKOUT_30M_EN	FREQ_OFFSET_VALID	RESET_OFFSET_EXT	POSTDIV_USB	POSTDIV_USB_EN	POSTDIV_AUDIO_EN	POSTDIV_AUDIO				Reserved																
Default	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	1	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Table 573: USB and Audio PLL Control Register (AUPLL_CTRL2)

Bits	Name	Type	Reset	Description
31:27	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 573: USB and Audio PLL Control Register (AUPLL_CTRL2)

Bits	Name	Type	Reset	Description
26	CLKOUT_30M_EN	R/W	0x0	0: disable 30MHz output clock, 1: enable 30MHz output clock
25	FREQ_OFFSET_VALID	R/W	0x0	Frequency offset value readiness indicator for both
24	RESET_OFFSET_EXT	R/W	0x0	External offset logic reset signal
23	POSTDIV_USB	R/W	0x0	Control signal for the USB divider to get 60MHz USB-CLKOUT
22	POSTDIV_USB_EN	R/W	0x0	USB post divider enable
21	POSTDIV_AUDIO_EN	R/W	0x0	Control signal for the post divider to get audio output clock
20:14	POSTDIV_AUDIO	R/W	0x2	Divisor for audio clock post divider. Divisor=3 when POSTDIV_AUDIO[6:0]=3. Divisor=2*POSTDIV_AUDIO[6:1], start from POSTDIV_AUDIO[6:0]=2 except 3. Invalid setting for POSTDIV_AUDIO[5:0]=0,1 and other odd number.
13:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.38 CAU Clock Control Register (CAU_CTRL)

Instance Name
CAU_CTRL

Offset
0xB8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											CAU_GPADC0_MCLK_EN	CAU_GPADC1_MCLK_EN	CAU_GPDAC_MCLK_EN	CAU_ACOMP_MCLK_EN	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	1	1

Table 574: CAU Clock Control Register (CAU_CTRL)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	CAU_GPADC0_MCLK_EN	R/W	0x1	cau_gpda0c module main clock enable signal

Table 574: CAU Clock Control Register (CAU_CTRL)

Bits	Name	Type	Reset	Description
2	CAU_GPADAC1_MCLK_EN	R/W	0x1	cau_gpdac1 module main clock enable signal
1	CAU_GPDAC_MCLK_EN	R/W	0x1	cau_gpdac module main clock enable signal
0	CAU_ACOMP_MCLK_EN	R/W	0x1	cau_acomp module main clock enable signal

A.19.39 RC32K Control Register (RC32K_CTRL)

Instance Name

RC32K_CTRL

Offset

0xBC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Field	Reserved							RC32K_PD	RC32K_CAL_EN	RC32K_CODE_FR_EXT						RC32K_EXT_CODE_EN	Reserved	RC32K_CAL_DIV		RC32K_ALLOW_CAL	RC32K_CODE_FR_CAL						RC32K_CAL_INPROGRESS	Reserved	RC32K_RDY	RC32K_CAL_DONE				
Default	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	?	1	1	0	0	0	0	0	0	0	0	0	0	?	?	0	0

Table 575: RC32K Control Register (RC32K_CTRL)

Bits	Name	Type	Reset	Description
31:25	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
24	RC32K_PD	R/W	0x0	Power down 32K oscillator
23	RC32K_CAL_EN	R/W	0x0	Enable calibration of 32K oscillator
22:16	RC32K_CODE_FR_EXT	R/W	0x0	External code In for frequency setting
15	RC32K_EXT_CODE_EN	R/W	0x0	Allow external code In to go into the ckt.
14	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
13:12	RC32K_CAL_DIV	R/W	0x3	Divider for the clock step during calibration
11	RC32K_ALLOW_CAL	R/W	0x0	Allow calibration to be performed (monitor system clk)
10:4	RC32K_CODE_FR_CAL	R	0x0	After calibration hold calibrated code

Table 575: RC32K Control Register (RC32K_CTRL)

Bits	Name	Type	Reset	Description
3	RC32K_CAL_INPROGRESS	R	0x0	Asserts hi when calibration is in progress
2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	RC32K_RDY	R	0x0	Asserts hi when 32K clock is ready upon pwrup
0	RC32K_CAL_DONE	R	0x0	Asserts hi when calibration is done

A.19.40 XTAL32K Control Register (XTAL32K_CTRL)

Instance Name

XTAL32K_CTRL

Offset

0xC0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																			X32K_EN	X32K_EXT_OSC_EN	Reserved										X32K_RDY
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	?	?	?	?	?	?	?	0

Table 576: XTAL32K Control Register (XTAL32K_CTRL)

Bits	Name	Type	Reset	Description
31:13	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
12	X32K_EN	R/W	0x0	Enable 32K oscillator
11	X32K_EXT_OSC_EN	R/W	0x0	Enable external oscillator mode for outside clock
10:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	X32K_RDY	R	0x0	Assert hi when ready

A.19.41 Comparator Control Register (PMIP_CMP_CTRL)

Instance Name Offset
PMIP_CMP_CTRL 0xC4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Field	Reserved																						CAU_REF_EN	COMP_HYST			COMP_EN	COMP_DIFF_EN		COMP_REF_SEL				COMP_RDY		COMP_OUT	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	0	0	0	0	0	0	0	0					

Table 577: Comparator Control Register (PMIP_CMP_CTRL)

Bits	Name	Type	Reset	Description
31:10	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
9	CAU_REF_EN	R/W	0x0	Enable CAU-current-generated reference in PMIP
8:7	COMP_HYST	R/W	0x2	Control of comparator hysteresis. 0x0: 0mV (single-ended mode) 0mV (differential mode) 0x1: 4mV (single-ended mode) 8mV (differential mode) 0x2: 8mV (single-ended mode) 17mV (differential mode) 0x3: 22mV (single-ended mode) 52mV (differential mode)
6	COMP_EN	R/W	0x0	Enable AON domain comparator
5	COMP_DIFF_EN	R/W	0x0	Enable Differential mode for AON comparator
4:2	COMP_REF_SEL	R/W	0x0	Select comparator reference for single-ended mode. 0x0: 0.2V reference voltage 0x1: 0.4V reference voltage 0x2: 0.6V reference voltage 0x3: 0.8V reference voltage 0x4: 1.0V reference voltage 0x5: 1.2V reference voltage 0x6: 1.4V reference voltage 0x7: 1.6V reference voltage
1	COMP_RDY	R	0x0	Ready to use AON domain comparator
0	COMP_OUT	R	0x0	Output of AON domain comparator

A.19.42 PMIP Config Register 0. (PMIP_CONFIG0)

Instance Name
PMIP_CONFIG0

Offset
0xC8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Field	Reserved												DEL_AV18_SEL		Reserved																LDO_AV18_RAMP_RATE		Reserved	
Default	?	?	?	?	?	?	?	?	?	?	?	?	1	1	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1	?	?		

Table 578: PMIP Config Register 0. (PMIP_CONFIG0)

Bits	Name	Type	Reset	Description
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	DEL_AV18_SEL	R/W	0x3	AV18 power delay control; 00: short delay; 11: long delay
17:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:2	LDO_AV18_RAMP_RATE	R/W	0x3	AV18 LDO ramp rate control.10: fast ramp; 11: slow ramp
1:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.43 PMIP Config Register 1. (PMIP_CONFIG1)

Instance Name
PMIP_CONFIG1

Offset
0xCC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved												DEL_VFL_SEL		Reserved												LDO_V12_OUT_PM2			Reserved			
Default	?	?	?	?	?	?	?	?	?	?	?	?	1	1	?	?	?	?	?	?	?	?	?	?	?	1	0	0	?	?	?	?	

Table 579: PMIP Config Register 1. (PMIP_CONFIG1)

Bits	Name	Type	Reset	Description
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19:18	DEL_VFL_SEL	R/W	0x3	VFL power delay control; 01: short delay; 11: long delay

Table 579: PMIP Config Register 1. (PMIP_CONFIG1)

Bits	Name	Type	Reset	Description
17:7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
6:4	LDO_V12_OUT_PM 2	R/W	0x4	Select output voltage of v12 at PM2 mode.
3:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.44 Vbat Brownout Detection Control Register (PMIP_BRNDET_VBAT)

Instance Name

PMIP_BRNDET_VBAT

Offset

0xD0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												BRNDET_VBAT_EN	BRNTRIG_VBAT_CN TL		BRNHYST_VBAT_CN TL		BRNDET_VBAT_FILT		BRNDET_VBAT_RDY		BRNDET_VBAT_OUT	Reserved									
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	1	0	0	1	0	1	0	0	0	?	?	?	?	?	?	?	?	?	?

Table 580: Vbat Brownout Detection Control Register (PMIP_BRNDET_VBAT)

Bits	Name	Type	Reset	Description
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19	BRNDET_VBAT_EN	R/W	0x0	Enable Vbat brownout detection
18:16	BRNTRIG_VBAT_CN TL	R/W	0x4	Control trigger voltage of Vbat brownout detection. 0x0: 1.70V 0x1: 1.80V 0x2: 1.90V 0x3: 2.00V 0x4: 2.10V 0x5: 2.20V 0x6: 2.30V 0x7: 2.40V
15:14	BRNHYST_VBAT_CN TL	R/W	0x2	Control of Vbat brownout detection hysteresis. 0x0: 0mV 0x1: 40mV 0x2: 66mV 0x3: 85mV

Table 580: Vbat Brownout Detection Control Register (PMIP_BRNDET_VBAT)

Bits	Name	Type	Reset	Description
13:12	BRNDET_VBAT_FILTER	R/W	0x2	Select filtering level for Vbat pulse to Vbat Brownout Detection. 0x0: <0.2ms 0x1: <0.4ms 0x2: <0.8ms 0x3: <1.6ms
11	BRNDET_VBAT_READY	R	0x0	Assert hi if Vbat brownout is ready--> out can be taken
10	BRNDET_VBAT_OUT	R	0x0	Assert hi if Vbat brownout happened
9:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.45 PMIP Config Register 2. (PMIP_CONFIG2)

Instance Name
PMIP_CONFIG2

Offset
0xD4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														LDO_V12_RAMP_RATE	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1

Table 581: PMIP Config Register 2. (PMIP_CONFIG2)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1:0	LDO_V12_RAMP_RATE	R/W	0x3	V12 LDO ramp rate control.10:fast ramp;11: slow ramp

A.19.46 LDO Control Register (PMIP_LDO_CTRL)

Instance Name
PMIP_LDO_CTRL

Offset
0xD8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																				LDO_AV18_EN	Reserved				LDO_AV18_PWRSW_EN	LDO_V12_EN	Reserved				
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	?	?	?	?	0	1	?	?	?	?	?

Table 582: LDO Control Register (PMIP_LDO_CTRL)

Bits	Name	Type	Reset	Description
31:12	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
11	LDO_AV18_EN	R/W	0x1	Enable ldo_av18
10:7	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
6	LDO_AV18_PWRSW_EN	R/W	0x0	Enable bypass (power switch mode) for ldo_av18
5	LDO_V12_EN	R/W	0x1	Enable ldo_v12
4:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.47 Peripheral Clock Source Register (PERI_CLK_SRC)

Instance Name
PERI_CLK_SRC

Offset
0xDC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																													RTC_INT_SEL	SSP2_AUDIO_SEL	SSP1_AUDIO_SEL	SSP0_AUDIO_SEL
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	

Table 583: Peripheral Clock Source Register (PERI_CLK_SRC)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3	RTC_INT_SEL	R/W	0x0	RTC function clock select. 1: XTAL32K, 0: RC32K

Table 583: Peripheral Clock Source Register (PERI_CLK_SRC)

Bits	Name	Type	Reset	Description
2	SSP2_AUDIO_SEL	R/W	0x0	SSP2 function clock select. 1: output of audio PLL, 0: PMU generated clock
1	SSP1_AUDIO_SEL	R/W	0x0	SSP1 function clock select. 1: output of audio PLL, 0: PMU generated clock
0	SSP0_AUDIO_SEL	R/W	0x0	SSP0 function clock select. 1: output of audio PLL, 0: PMU generated clock

A.19.48 GPT0 Clock Control Register (GPT0_CTRL)

Instance Name

GPT0_CTRL

Offset

0xE4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																					GPT0_CLK_SEL0		GPT0_CLK_SEL1		Reserved	GPT0_CLK_DIV					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	?	0	0	0	0	0	1

Table 584: GPT0 Clock Control Register (GPT0_CTRL)

Bits	Name	Type	Reset	Description
31:11	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
10:9	GPT0_CLK_SEL0	R/W	0x0	select for first level MUX values Clock Source 2'b00 system source clock 2'b01 system source clock 2'b10 RC32M clock 2'b11 MAINXTAL clock
8:7	GPT0_CLK_SEL1	R/W	0x0	select for second level MUX values Clock Source 2'b00 output of first level MUX 2'b01 output of first level MUX 2'b10 RC32K clock 2'b11 XTAL32K clock
6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 584: GPT0 Clock Control Register (GPT0_CTRL)

Bits	Name	Type	Reset	Description
5:0	GPT0_CLK_DIV	R/W	0x1	gpt0 function clock divisor values divisor 2'b00 divisor = 1 other divisor = gpt0_clk_div[5:0]

A.19.49 GPT1 Clock Control Register (GPT1_CTRL)

Instance Name
GPT1_CTRL

Offset
0xE8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Field	Reserved																					GPT1_CLK_SEL0		GPT1_CLK_SEL1		Reserved		GPT1_CLK_DIV						
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	?	0	0	0	0	0	1		

Table 585: GPT1 Clock Control Register (GPT1_CTRL)

Bits	Name	Type	Reset	Description
31:11	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
10:9	GPT1_CLK_SEL0	R/W	0x0	sel signal for MUX before frequency divisor values Clock Source 2'b00 system source clock 2'b01 system source clock 2'b10 RC32M clock 2'b11 MAINXTAL clock
8:7	GPT1_CLK_SEL1	R/W	0x0	sel signal for MUX after frequency divisor values Clock Source 2'b00 output of first level MUX 2'b01 output of first level MUX 2'b10 RC32K clock 2'b11 XTAL32K clock
6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 585: GPT1 Clock Control Register (GPT1_CTRL)

Bits	Name	Type	Reset	Description
5:0	GPT1_CLK_DIV	R/W	0x1	gpt1 function clock divisor values divisor 2'b00 divisor = 1 other divisor = gpt1_clk_div[5:0]

A.19.50 GPT2 Clock Control Register (GPT2_CTRL)

Instance Name
GPT2_CTRL

Offset
0xEC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																				GPT2_CLK_SEL0		GPT2_CLK_SEL1		Reserved	GPT2_CLK_DIV						
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	?	0	0	0	0	0	1

Table 586: GPT2 Clock Control Register (GPT2_CTRL)

Bits	Name	Type	Reset	Description
31:11	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
10:9	GPT2_CLK_SEL0	R/W	0x0	sel signal for MUX before frequency divisor values Clock Source 2'b00 system source clock 2'b01 system source clock 2'b10 RC32M clock 2'b11 MAINXTALclock
8:7	GPT2_CLK_SEL1	R/W	0x0	sel signal for MUX after frequency divisor values Clock Source 2'b00 output of first level MUX 2'b01 output of first level MUX 2'b10 RC32K clock 2'b11 XTAL32K clock
6	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

Table 586: GPT2 Clock Control Register (GPT2_CTRL)

Bits	Name	Type	Reset	Description
5:0	GPT2_CLK_DIV	R/W	0x1	gpt2 function clock divisor values divisor 2'b00 divisor = 1 other divisor = gpt2_clk_div[5:0]

A.19.51 GPT3 Clock Control Register (GPT3_CTRL)

Instance Name
GPT3_CTRL

Offset
0xF0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field	Reserved																					GPT3_CLK_SEL0		GPT3_CLK_SEL1		Reserved							
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	?	?	?	?	?	?	?	

Table 587: GPT3 Clock Control Register (GPT3_CTRL)

Bits	Name	Type	Reset	Description
31:11	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
10:9	GPT3_CLK_SEL0	R/W	0x0	sel signal for MUX before frequency divisor values Clock Source 2'b00 system source clock 2'b01 system source clock 2'b10 RC32M clock 2'b11 MAINXTAL clock
8:7	GPT3_CLK_SEL1	R/W	0x0	sel signal for MUX after frequency divisor values Clock Source 2'b00 output of first level MUX 2'b01 output of first level MUX 2'b10 RC32K clock 2'b11 XTAL32K clock
6:0	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.

A.19.52 Wakeup Edge Detect Register (WAKEUP_EDGE_DETECT)

Instance Name
WAKEUP_EDGE_DETECT

Offset
0xF4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														WAKEUP1	WAKEUP0
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1

Table 588: Wakeup Edge Detect Register (WAKEUP_EDGE_DETECT)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	WAKEUP1	R/W	0x1	external pin1 wakeup edge detect register. 1: active-high, 0: active-low.
0	WAKEUP0	R/W	0x1	external pin0 wakeup edge detect register.1: active-high, 0: active-low.

A.20 SYS_CTRL Register Information

Table 589: SYS_CTRL Register Summary

Offset	Name	Description	Details
0x00	REV_ID	Chip Revision Register	Page: 431
0x04	MEM	Memory Space Configuration	Page: 432
0x08	RESERVED	Reserved	Page: 432
0x0C	RESERVED	Reserved	Page: 433
0x10	RESERVED	Reserved	Page: 433
0x14	RESERVED	Reserved	Page: 433
0x18	RESERVED	Reserved	Page: 434
0x1C	RESERVED	Reserved	Page: 434
0x20	RESERVED	Reserved	Page: 435
0x24	RESERVED	Reserved	Page: 435
0x28	RESERVED	Reserved	Page: 435
0x2C	RESERVED	Reserved	Page: 436
0x30	DMA_HS	DMA Handshaking Mapping Register	Page: 436
0x34	RESERVED	Reserved	Page: 438
0x38	RESERVED	Reserved	Page: 439
0x3C	PERI_SW_RST	Peripheral SW reset	Page: 439
0x40	USB_CTRL	USB Control Register	Page: 441
0x44	USB_PHY_CTRL	USB PHY Control Register	Page: 443
0x48	RESERVED	Reserved	Page: 443
0x4C	RESERVED	Reserved	Page: 444

A.20.1 Chip Revision Register (REV_ID)

Instance Name

REV_ID

Offset

0x00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	REV_ID																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 590: Chip Revision Register (REV_ID)

Bits	Name	Type	Reset	Description
31:0	REV_ID	R	0x1	Chip revision id

A.20.2 Memory Space Configuration (MEM)

Instance Name

MEM

Offset

0x04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														CFG	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Table 591: Memory Space Configuration (MEM)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1:0	CFG	R/W	0x0	Memory Space Configuration Note: Please refer to Chapter 6 Memory map, bus fabric and interrupts for more information. 0x0: RAM0, RAM1 mapped to CODE, RAM2, RAM3 mapped to SRAM 0x1: RAM0 mapped to CODE, RAM1, RAM2, RAM3 mapped to SRAM 0x2: RAM0, RAM1 and RAM2 mapped to CODE, RAM3 mapped to SRAM 0x3: invalid combination, defaults to CFG=00 memory map

A.20.3 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																												Reserved			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	1	0

Table 592: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x6	Reserved. Do not change the reset value.

A.20.4 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	1	0	

Table 593: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x6	Reserved. Do not change the reset value.

A.20.5 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	1	0	

Table 594: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x6	Reserved. Do not change the reset value.

A.20.6 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																										Reserved					
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	1	0	

Table 595: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x6	Reserved. Do not change the reset value.

A.20.7 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

Offset

RESERVED

0x18

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 596: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.20.8 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

Offset

RESERVED

0x1C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 597: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.20.9 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x20

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 598: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.20.10 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x24

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 599: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:0	Reserved	R	0x0	Reserved. Do not change the reset value.

A.20.11 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x28

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																											Reserved				
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	0	1	1	0

Table 600: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:5	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
4:0	Reserved	R/W	0x16	Reserved. Do not change the reset value.

A.20.12 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x2C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																												Reserved			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	1	0

Table 601: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x6	Reserved. Do not change the reset value.

A.20.13 DMA Handshaking Mapping Register (DMA_HS)

Instance Name

DMA_HS

Offset

0x30

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Field	Reserved										MAPPING_15	MAPPING_14	MAPPING_13		MAPPING_12		MAPPING_11		MAPPING_10		MAPPING_9		MAPPING_8		MAPPING_7		MAPPING_6		MAPPING_5		MAPPING_4		MAPPING_3		MAPPING_2		MAPPING_1		MAPPING_0	
Default	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

Table 602: DMA Handshaking Mapping Register (DMA_HS)

Bits	Name	Type	Reset	Description
31:22	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
21	MAPPING_15	R/W	0x0	Request Interface 15 configuration. dma_hs_sel Peripheral 0x0: GPT3_1 0x1: I2C0 TX
20	MAPPING_14	R/W	0x0	Request Interface 14 configuration. dma_hs_sel Peripheral 0x0: GPT3_0 0x1: I2C0 RX
19:18	MAPPING_13	R/W	0x0	Request Interface 13 configuration. dma_hs_sel Peripheral 0x0: GPT2_1 0x1: I2C2 TX 0x2: UART1 RX 0x3: QSPI1 TX

Table 602: DMA Handshaking Mapping Register (DMA_HS)

Bits	Name	Type	Reset	Description
17:16	MAPPING_12	R/W	0x0	Request Interface 12 configuration. dma_hs_sel Peripheral 0x0: GPT2_0 0x1: I2C1 TX 0x2: UART0 RX 0x3: QSPI1 RX
15	MAPPING_11	R/W	0x0	Request Interface 11 configuration. dma_hs_sel Peripheral 0x0: GPT1_1 0x1: SSP2 TX
14	MAPPING_10	R/W	0x0	Request Interface 10 configuration. dma_hs_sel Peripheral 0x0: GPT1_0 0x1: SSP0 TX
13	MAPPING_9	R/W	0x0	Request Interface 9 configuration. dma_hs_sel Peripheral 0x0: ADC1 0x1: SSP0 RX
12	MAPPING_8	R/W	0x0	Request Interface 8 configuration. dma_hs_sel Peripheral 0x0: ADC0 0x1: I2C2 RX
11:10	MAPPING_7	R/W	0x0	Request Interface 7 configuration. dma_hs_sel Peripheral 0x0: DAC1 0x1: UART3 TX others: SSP1 TX
9:8	MAPPING_6	R/W	0x0	Request Interface 6 configuration. dma_hs_sel Peripheral 0x0: DAC0 0x1: UART2 TX others: I2C1 RX
7:6	MAPPING_5	R/W	0x0	Request Interface 5 configuration. dma_hs_sel Peripheral 0x0: GPT0_1 0x1: UART1 TX others: QSPI0 RX
5	MAPPING_4	R/W	0x0	Request Interface 4 configuration. dma_hs_sel Peripheral 0x0: UART0 TX 0x1: SSP2 RX

Table 602: DMA Handshaking Mapping Register (DMA_HS)

Bits	Name	Type	Reset	Description
4:3	MAPPING_3	R/W	0x0	Request Interface 3 configuration. dma_hs_sel: Peripheral 0x0: GPT0_0 0x1: UART3 RX others: SSP1 RX
2	MAPPING_2	R/W	0x0	Request Interface 2 configuration. dma_hs_sel: Peripheral 0x0: UART2 RX 0x1: QSPI0_TX
1	MAPPING_1	R/W	0x0	Request Interface 2 configuration. dma_hs_sel: Peripheral 0x0: AES_CRC_OUT 0x1: UART1 RX
0	MAPPING_0	R/W	0x0	Request Interface 2 configuration. dma_hs_sel: Peripheral 0x0: AES_CRC_IN 0x1: UART0 RX

A.20.14 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x34

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														Reserved	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0

Table 603: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:1	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.20.15 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x38

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																														Reserved	
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?		0

Table 604: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1:0	Reserved	R/W	0x1	Reserved. Do not change the reset value.

A.20.16 Peripheral SW reset (PERI_SW_RST)

Instance Name

PERI_SW_RST

Offset

0x3C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved											QSPI0_RSTN_EN	QSPI1_RSTN_EN	Reserved	UART0_RSTN_EN	UART1_RSTN_EN	UART2_RSTN_EN	UART3_RSTN_EN	I2C0_RSTN_EN	I2C1_RSTN_EN	I2C2_RSTN_EN	SSP0_RSTN_EN	SSP1_RSTN_EN	SSP2_RSTN_EN	GPT0_RSTN_EN	GPT1_RSTN_EN	GPT2_RSTN_EN	GPT3_RSTN_EN	SDIO_RSTN_EN	Reserved	USB_RSTN_EN	WDT_RSTN_EN
Default	?	?	?	?	?	?	?	?	?	?	?	1	1	?	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	?	1	1

Table 605: Peripheral SW reset (PERI_SW_RST)

Bits	Name	Type	Reset	Description
31:21	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
20	QSPI0_RSTN_EN	R/W	0x1	QSPI0 software reset enable. Write 0 to reset. It only resets QSPI0 function clock domain.
19	QSPI1_RSTN_EN	R/W	0x1	QSPI1 software reset enable. Write 0 to reset. It only resets QSPI1 function clock domain.
18	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
17	UART0_RSTN_EN	R/W	0x1	UART0 software reset enable. Write 0 to reset. It only resets UART0 function clock domain.
16	UART1_RSTN_EN	R/W	0x1	UART1 software reset enable. Write 0 to reset. It only resets UART1 function clock domain.

Table 605: Peripheral SW reset (PERI_SW_RST)

Bits	Name	Type	Reset	Description
15	UART2_RSTN_EN	R/W	0x1	UART2 software reset enable. Write 0 to reset. It only resets UART2 function clock domain.
14	UART3_RSTN_EN	R/W	0x1	UART3 software reset enable. Write 0 to reset. It only resets UART3 function clock domain.
13	I2C0_RSTN_EN	R/W	0x1	I2C0 software reset enable. Write 0 to reset. It only resets I2C0 function clock domain.
12	I2C1_RSTN_EN	R/W	0x1	I2C1 software reset enable. Write 0 to reset. It only resets I2C1 function clock domain.
11	I2C2_RSTN_EN	R/W	0x1	I2C2 software reset enable. Write 0 to reset. It only resets I2C2 function clock domain.
10	SSP0_RSTN_EN	R/W	0x1	SSP0 software reset enable. Write 0 to reset. It only resets SSP0 function clock domain.
9	SSP1_RSTN_EN	R/W	0x1	SSP1 software reset enable. Write 0 to reset. It only resets SSP1 function clock domain.
8	SSP2_RSTN_EN	R/W	0x1	SSP2 software reset enable. Write 0 to reset. It only resets SSP2 function clock domain.
7	GPT0_RSTN_EN	R/W	0x1	GPT0 software reset enable. Write 0 to reset. It only resets GPT0 function clock domain.
6	GPT1_RSTN_EN	R/W	0x1	GPT1 software reset enable. Write 0 to reset. It only resets GPT1 function clock domain.
5	GPT2_RSTN_EN	R/W	0x1	GPT2 software reset enable. Write 0 to reset. It only resets GPT2 function clock domain.
4	GPT3_RSTN_EN	R/W	0x1	GPT3 software reset enable. Write 0 to reset. It only resets GPT3 function clock domain.
3	SDIO_RSTN_EN	R/W	0x1	SDIO software reset enable. Write 0 to reset. It only resets SDIO function clock domain.
2	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
1	USB_RSTN_EN	R/W	0x1	USB software reset enable. Write 0 to reset. It only resets USB function clock domain.
0	WDT_RSTN_EN	R/W	0x1	WDT software reset enable. Write 0 to reset. It only resets WDT function clock domain.

A.20.17 USB Control Register (USB_CTRL)

Instance Name

USB_CTRL

Offset

0x40

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												DISABLE_EL16	USBBUF_PDWN_EN	USBBUF_PDWN	TX_BUF_WTC		TX_BUF_RTC		RX_BUF_WTC		RX_BUF_RTC		EXT_FS_RCAL			FSDRV_EN			PLL_LOCK_BYPASS		
Default	?	?	?	?	?	?	?	?	?	?	?	?	0	1	0	0	1	1	0	0	1	1	0	1	0	0	0	1	1	1	1	0

Table 606: USB Control Register (USB_CTRL)

Bits	Name	Type	Reset	Description
31:20	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
19	DISABLE_EL16	R/W	0x0	1: Disable the EL16 patch for device mode.0: Don't disable it. Default=0
18	USBBUF_PDWN_EN	R/W	0x1	1: To enable HW to control USB buffer power down.0: SW will control USB buffer power down default=1'b1
17	USBBUF_PDWN	R/W	0x0	When reg_usbbuf_pdown_en=0, SW will control USB buffer SRAM power down. The power down mode is controlled with this register bit. Note: for SRAM power down control: To enter Leakage Reduction Mode, NCEPA and NCEPB must be de-asserted HIGH at least one clock cycle before asserting PDWN HIGH. To exit Leakage Reduction Mode, PDWN must be de-asserted LOW at least 20ns before memory can be re-enabled (NCEPA and NCEPB asserted LOW). While in Leakage Reduction Mode, CLKPA and CLKPB can be stopped LOW. While CLKPA and CLKPB are stopped LOW, NCEPA and NCEPB 0x1: Power down 0x0: Normal mode.
16:15	TX_BUF_WTC	R/W	0x1	USB TX buffer write timing control. Please see AC Characteristics table for specific timing information. It is REQUIRED to have these two values registered and modifiable through firmware or software rather than hardwired. At this time, the default setting is the recommended maximum setting. Please use a lower setting if it does not impact system performance. Default=2'b01.
14:13	TX_BUF_RTC	R/W	0x2	USB TX buffer read timing control. Please see AC Characteristics table for specific timing information. It is REQUIRED to have these two values registered and modifiable through firmware or software rather than hardwired. At this time, the default setting is the recommended maximum setting. Please use a lower setting if it does not impact system performance. Default=2'b10

Table 606: USB Control Register (USB_CTRL)

Bits	Name	Type	Reset	Description
12:11	RX_BUF_WTC	R/W	0x1	USB RX buffer write timing control. Please see AC Characteristics table for specific timing information. It is REQUIRED to have these two values registered and modifiable through firmware or software rather than hardwired. At this time, the default setting is the recommended maximum setting. Please use a lower setting if it does not impact system performance. Default=2'b01.
10:9	RX_BUF_RTC	R/W	0x2	USB RX buffer read timing control. Please see AC Characteristics table for specific timing information. It is REQUIRED to have these two values registered and modifiable through firmware or software rather than hardwired. At this time, the default setting is the recommended maximum setting. Please use a lower setting if it does not impact system performance. Default=2'b10
8:5	EXT_FS_RCAL	R/W	0x8	TX FS driver impedance adjusting for HS loading Which is 45 ohms Imp_cal[8:5] Impedance(SE) 4'b0000 900/12 = 75 ohms 4'b0001 900/13 = 69 ohms 4'b0010 900/14 = 64 ohms ... 4'b1000 900/20 = 45 ohms ... 4'b1111 900/27 = 33 ohms Default=4'b1000
4:1	FSDRV_EN	R/W	0xF	The whole FS driver include 12 programmable Driver cell and 17 calibration driver cell. Default=4'b1111 0x1: 3 programmable cell on 0x3: 6 programmable cell on 0x7: 9 programmable cell on 0xF: 12 programmable cell on
0	PLL_LOCK_BYPASS	R/W	0x0	1: Bypass pll_lock signal0: Don't bypass pll_lock. Pll_lock will gate xcvr_clk inside USB2 module. Default=0

A.20.18 USB PHY Control Register (USB_PHY_CTRL)

Instance Name
USB_PHY_CTRL

Offset
0x44

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved						Reserved				REG_PU_USB	LS_EN				TX_LS				Reserved	Reserved											
Default	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 607: USB PHY Control Register (USB_PHY_CTRL)

Bits	Name	Type	Reset	Description
31:26	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
25:22	Reserved	R/W	0x0	Reserved. Do not change the reset value.
21	REG_PU_USB	R/W	0x0	1: Turn on USB2 PHY analog and OTG part. Default=0
20:17	LS_EN	R/W	0x0	[20:17]: LS drive enable, default is 4'b0000
16:13	TX_LS	R/W	0x0	[16:13]: TX LS driver impedance adjustment, default is 4'b0000.
12:11	Reserved	R	0x0	Reserved. Do not change the reset value.
10:0	Reserved	R/W	0x0	Reserved. Do not change the reset value.

A.20.19 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name
RESERVED

Offset
0x48

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															Reserved																
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Table 608: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:17	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
16:0	Reserved	R/W	0x60	Reserved. Do not change the reset value.

A.20.20 Reserved (RESERVED)

Reserved. Do not change the reset value.

Instance Name

RESERVED

Offset

0x4C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved																												Reserved			
Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	1	0

Table 609: Reserved (RESERVED)

Bits	Name	Type	Reset	Description
31:4	Reserved	RSVD	--	Reserved. Always write 0. Ignore read value.
3:0	Reserved	R/W	0x6	Reserved. Do not change the reset value.



Marvell Semiconductor, Inc.
5488 Marvell Lane
Santa Clara, CA 95054, USA

Tel: 1.408.222.2500

Fax: 1.408.988.8279

www.marvell.com

Marvell. Moving Forward Faster