28-Lead ROM-Based DSP Motor Controller

Preliminary Technical Data — ADMC328

TARGET APPLICATIONS

Washing Machines, Refrigerator Compressors, Fans, Pumps, Industrial Variable Speed Drives

FEATURES

20 MIPS Fixed-Point DSP Core
- Single Cycle Instruction Execution (50 ns)
- ADSP-2100 Family Code Compatible
- Independent Computational Units
  - ALU
  - Multiplier/Accumulator
  - Barrel Shifter
- Multifunction Instructions
- Single Cycle Context Switch
- Powerful Program Sequencer
  - Zero Overhead Looping
  - Conditional Instruction Execution
- Two Independent Data Address Generator

Memory Configuration
- 512 × 24-Bit Program Memory RAM
- 4K × 24-Bit Program Memory ROM
- 512 × 16-Bit Data Memory RAM

Three-Phase 16-Bit PWM Generator
- 16-Bit Center-Based PWM Generator
- Programmable Deadtime and Narrow Pulse Deletion
- Edge Resolution to 50 ns
- 150 Hz Minimum Switching Frequency
- Double/Single Duty Cycle Update Mode Control
- Programmable PWM Pulsewidth
- Suitable for AC Induction and Synchronous Motors
- Special Crossover Function for Brushless DC Motors
- Individual Enable and Disable for Each PWM Output
- High Frequency Chopping Mode for Transformer Coupled Gate Drives
- External $\overline{\text{PWMTRIP}}$ Pin

Six Analog Input Channels
- Acquisition Synchronized to PWM Switching Frequency

9-Pin Digital I/O Port
- Bit Configurable as Input or Output
- Change of State Interrupt Support

Two 8-Bit Auxiliary PWM Timers
- Synchronized Analog Output
- Programmable Frequency
- 0% to 100% Duty Cycle
- Two Programmable Operational Modes
  - Independent Mode
  - Offset Mode

16-Bit Watchdog Timer

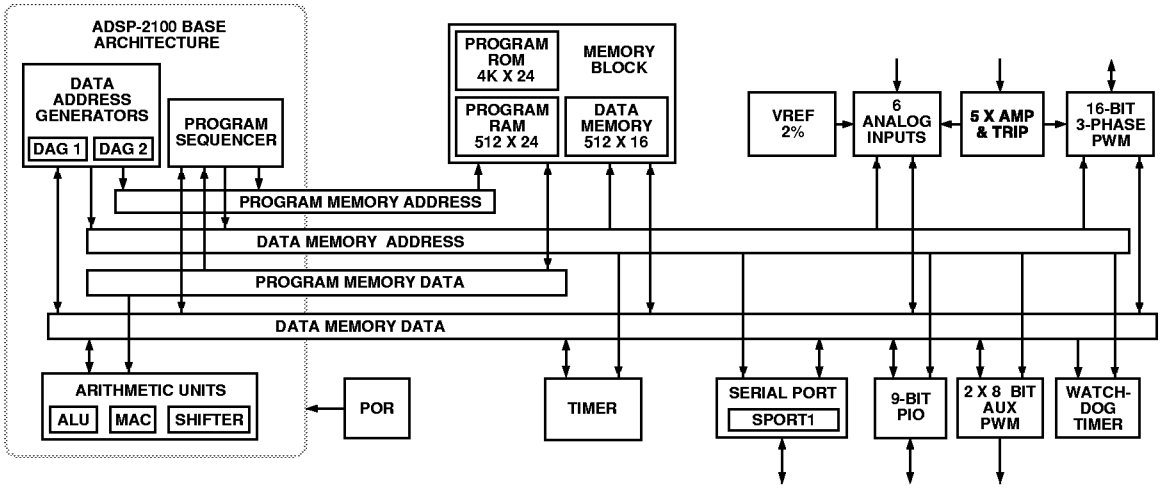
Programmable 16-Bit Internal Timer with Prescaler

Double Buffered Synchronous Serial Port

Hardware Support for UART Emulation

28-Lead SOIC Package

FUNCTIONAL BLOCK DIAGRAM



REV. PrA

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 World Wide Web Site: http://www.analog.com
Fax: 781/326-8703

ADMC328–SPECIFICATIONS (V_{DD} = +5 V ± 5%, GND = 0 V, T_A = –40°C to +85°C, CLKIN = 10 MHz, unless otherwise noted)

ANALOG-TO-DIGITAL CONVERTER

Parameter	Min	Typ	Max	Units	Conditions/Comments
Signal Input	0.3		3.5	V	V1, V2, VAUX0, VAUX1, VAUX2
Signal Input	–0.4		0	V	I_{SENSE}
Resolution[1]			12	Bits	No Missing Codes
Linearity Error[2]		2	4	Bits	
Zero Offset[2]	–5	0	50	mV	
Channel-to-Channel Comparator Match[2]			22	mV	
Comparator Delay		600		ns	
ADC Lo-Level Input Current[2]			10	µA	V_{IN} = 3.5 V
ADC Hi-Level Input Current[2]	–10			µA	V_{IN} = 0.0 V

NOTES
[1]Resolution varies with PWM switching frequency (double update mode) 78.1 kHz = 8 bits, 4.9 kHz = 12 bits.
[2]Charging Cap = 10 nF, 2.44 kHz sample frequency, V1, V2, VAUX0, VAUX1, VAUX2.

Specifications subject to change without notice.

ELECTRICAL CHARACTERISTICS

Parameter		Min	Typ	Max	Units	Conditions/Comments
V_{IL}	Lo-Level Input Voltage			0.8	V	
V_{IH}	Hi-Level Input Voltage	2			V	
V_{OL}	Low Level Output Voltage[1]			0.4	V	I_{OL} = 2 mA
V_{OL}	Low Level Output Voltage[2]			0.5	V	I_{OL} = –2 mA
V_{OH}	High Level Output Voltage	4			V	I_{OH} = 0.5 mA
I_{IL}	Low Level Input Current[3]	–50			µA	V_{IN} = 0 V
I_{IL}	Low Level Input Current[4]	–10			µA	V_{IN} = 0 V
I_{IH}	High Level Input Current[2]			50	µA	V_{IN} = V_{DD}
I_{IH}	High Level Input Current[3]			10	µA	V_{IN} = V_{DD}
I_{OZH}	Hi-Level Three-State Leakage Current[5]			50	µA	V_{IN} = 0 V
I_{OZL}	Lo-level Three-State Leakage Current[5]	–10			µA	V_{IN} = V_{DD}
I_{IL}	Lo-Level PWMTRIP Current			10	µA	@ V_{DD} = max, V_{IN} = 0 V
I_{DD}	Supply Current (Power-Down Mode)			6	mA	
I_{DD}	Supply Current (Idle)			60	mA	
I_{DD}	Supply Current (Dynamic)			100	mA	

NOTES
[1]Output pins PIO0–PIO8, AH, AL, BH, BL, CH, CL.
[2]XTAL Pin.
[3]Internal Pull-Up, $\overline{\text{RESET}}$.
[4]Internal Pull-Down, $\overline{\text{PWMTRIP}}$.
[5]Terminated low I/O's PIO0–PIO8.

Specifications subject to change without notice.

CURRENT SOURCE[1]

Parameter	Min	Typ	Max	Units	Conditions/Comments
Programming Resolution			3	Bits	
Default Current[2]	60	80	102	µA	ICONST_TRIM = 0x00
Tuned Current	95	100	105	µA	
Programmable Resolution	5		8	µA	

NOTES
[1]For ADC Calibration.
[2]0.3 V to 3.5 V I_{CONST} Voltage.

Specifications subject to change without notice.

VOLTAGE REFERENCE

Parameter	Min	Typ	Max	Units	Conditions/Comments
Voltage Level (V_{REF})	2.5	2.55	2.6	V	
Output Voltage Drift		20		ppm/°C	

Specifications subject to change without notice.

I_{SENSE} Amplifier—Trip

Parameter	Min	Typ	Max	Units	Conditions/Comments
I_{SENSE} Gain	–5.5	–5	–4.5		V_{IN} = –0.4 V to 0.0 V
I_{SENSE} Current	–20		0	µA	V_{IN} = –0.4 V to V_{DD} –1.0 V
I_{SENSE} Input Offset Voltage	–160		–50	mV	
Trip Voltage (V_{TRIP})	–0.610	–0.55	–0.49	mV	

Specifications subject to change without notice.

Power On Reset

Parameter	Min	Typ	Max	Units	Conditions/Comments
Reset Threshold (V_{RST})	3.7	3.9	4.1	V	
Hysteresis (V_{HYST})	200		300	mV	
Reset Active Timeout Period (t_{RST})		3.2[1]		ms	

NOTES

[1] 2^{16} CLKOUT Cycles.

Specifications subject to change without notice.

16-Bit PWM Timer

Parameter	Min	Typ	Max	Units	Conditions/Comments
Counter Resolution		16		Bits	
Edge Resolution (Single Update Mode)		100		ns	
Edge Resolution (Double Update Mode)		50		ns	
Programmable Deadtime Range	0		100	µs	
Programmable Deadtime Increments		100		ns	
Programmable Pulse Deletion Range	0		100	µs	
Programmable Pulse Deletion Increments		100		ns	
PWM Frequency Range	150			Hz	
PWMSYNC Pulsewidth (T_{CRST})	0.05		12.5	µs	
Gate Drive Chop Frequency Range	0.02		5	MHz	

Auxiliary PWM Timers

Parameter	Min	Typ	Max	Units	Conditions/Comments
Resolution		8		Bits	
PWM Frequency	0.039		5	MHz	10 MHz CLKIN

Specifications subject to change without notice.

TIMING PARAMETERS

Parameter		Min	Max	Unit
Clock Signals				
Signal t_{CK} is defined as 0.5 t_{CKIN}. The ADMC328 uses an input clock with a frequency equal to half the instruction rate; a 10 MHz input clock (which is equivalent to 100 ns) yields a 50 ns processor cycle (equivalent to 20 MHz). When t_{CK} values within the range of 0.5 t_{CKIN} period, they should be substituted for all relevant timing parameters to obtain specification value. Example: t_{CKH} = 0.5 t_{CK} – 10 ns = 0.5 (50 ns) – 10 ns = 15 ns.				
Timing Requirements:				
t_{CKIN}	CLKIN Period	100	150	ns
t_{CKIL}	CLKIN Width Low	20		ns
t_{CKIH}	CLKIN Width High	20		ns
Switching Characteristics:				
t_{CKL}	CLKOUT Width Low	0.5 t_{CK} – 10		ns
t_{CKH}	CLKOUT Width High	0.5 t_{CK} – 10		ns
t_{CKOH}	CLKIN High to CLKOUT High	0	20	ns
Control Signals				
Timing Requirement:				
t_{RSP}	RESET Width Low	5 t_{CK}[1]		ns

NOTES
[1]Applies after power-up sequence is complete.
Specifications subject to change without notice.

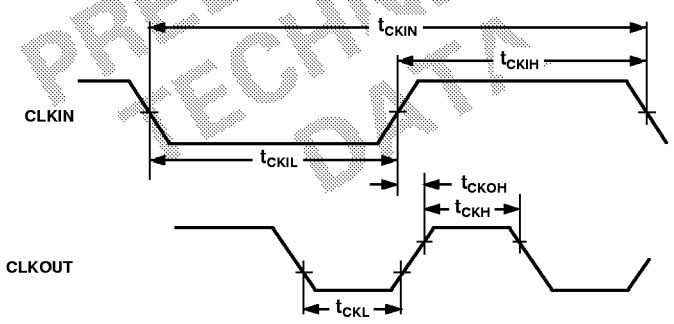


Figure 1. Clock Signals

Parameter		Min	Max	Unit
Serial Ports				
Timing Requirements				
t_{SCK}	SCLK Period	100		ns
t_{SCS}	DR/TFS/RFS Setup before SCLK Low	15		ns
t_{SCH}	DR/TFS/RFS Hold after SCLK Low	20		ns
t_{SCP}	$SCLK_{IN}$ Width	40		ns
Switching Characteristics:				
t_{CC}	CLKOUT High to $SCLK_{OUT}$	0.25 t_{CK}	0.25 t_{CK} + 20	ns
t_{SCDE}	SCLK High to DT Enable	0		ns
t_{SCDV}	SCLK High to DT Valid		30	ns
t_{RH}	TFS/RFS_{OUT} Hold after SCLK High	0		ns
t_{RD}	TFS/RFS_{OUT} Delay from SCLK High		30	ns
t_{SCDH}	DT Hold after SCLK High	0		ns
t_{SCDD}	SCLK High to DT Disable		30	ns
t_{TDE}	TFS (Alt) to DT Enable	0		ns
t_{TDV}	TFS (Alt) to DT Valid		25	ns
t_{RDV}	RFS (Multichannel, Frame Delay Zero) to DT Valid		30	ns

Specifications subject to change without notice.

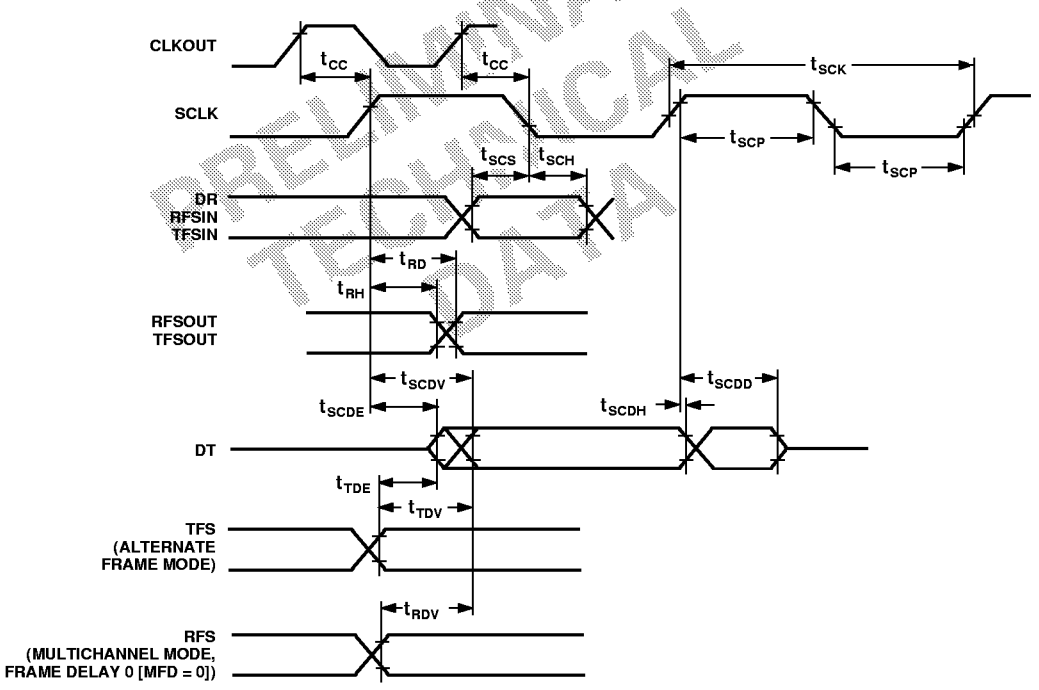


Figure 2. Serial Port Timing

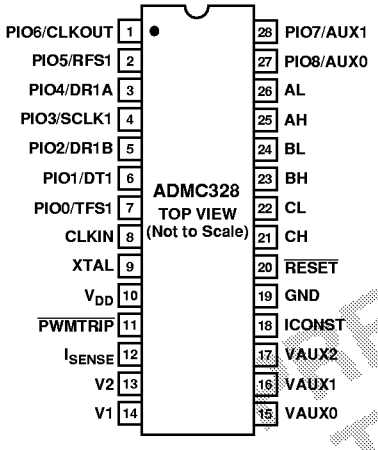
ABSOLUTE MAXIMUM RATINGS*

Supply Voltage (V_{DD}) . . . . . –0.3 V to +7.0 V
Input Voltage . . . . . –0.3 V to V_{DD} + 0.3 V
Output Voltage Swing . . . . . –0.3 V to V_{DD} + 0.3 V
Operating Temperature Range (Ambient) . . . –40°C to +85°C
Storage Temperature Range . . . . . –65°C to +150°C
Lead Temperature (5 sec) . . . . . +280°C

*Stresses greater than those listed may cause permanent damage to the device. These are stress ratings only; functional operation of the device at these or any other conditions greater than those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

PIN CONFIGURATION

28-Lead Wide-Body SOIC



PIN FUNCTION DESCRIPTIONS

Pin No.	Pin Type	Pin Name
1	I/O	PIO6/CLKOUT
2	I/O	PIO5/RFS1
3	I/O	PIO4/DR1A
4	I/O	PIO3/SCLK1
5	I/O	PIO2/DR1B
6	I/O	PIO1/DT1
7	I/O	PIO0/TFS1
8	I	CLKIN
9	O	XTAL
10	POWER	V_{DD}
11	I	PWMTRIP
12	I	I_{SENSE}
13	I	V2
14	I	V1
15	I	VAUX0
16	I	VAUX1
17	I	VAUX2
18	O	ICONST
19	GND	GND
20	I	$\overline{\text{RESET}}$
21	O	CH
22	O	CL
23	O	BH
24	O	BL
25	O	AH
26	O	AL
27	I/O	PIO8/AUX0
28	I/O	PIO7/AUX1

ORDERING GUIDE

Model	Temperature Range	Instruction Rate	Package Description	Package Option
ADMC328BR	–40°C to +85°C	20 MHz	28-Lead Wide Body (SOIC)	R-28

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADMC328 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

GENERAL DESCRIPTION

The ADMC328 is a low cost, single-chip DSP-based controller, suitable for ac induction motors, permanent magnet synchronous motors and brushless dc motors. The ADMC328 integrates a 20 MIPS, fixed-point DSP core with a complete set of motor control and system peripherals that permits fast, efficient development of motor controllers.

The DSP core of the ADMC328 is the ADSP-2171, which is completely code compatible with the ADSP-2100 DSP family and combines three computational units, data address generators, and a program sequencer. The computational units comprise an ALU, a multiplier/accumulator (MAC) and a barrel shifter. The ADSP-2171 adds new instructions for bit manipulation, multiplication (× squared), biased rounding, and global interrupt masking.

The system peripherals are the power on reset circuit (POR), the watchdog timer, and a synchronous serial port. The serial port is configurable, double buffered, with hardware support for UART and SCI port emulation.

The ADMC328 provides 512 × 24-bit program memory RAM, 4K × 24-bit program memory ROM and 512 × 16-bit data memory RAM. The program memory ROM contains the user specified program code and is defined using a single metal layer mask. The program and data memory RAM can be used for dynamic data storage or can be loaded through the serial port from an external device as in other ADMC3xx family parts.

The motor control peripherals of the ADMC328 comprise a 12-bit analog data acquisition system with 6 analog input channels and an internal 2% reference. In addition, a three-phase, 16-bit, center-based PWM generation unit can be used to produce high-accuracy PWM signals with minimal processor overhead. The ADMC328 also contains two auxiliary PWM outputs, 9 lines of digital I/O and a 16-bit watchdog timer.

Because the ADMC328 has a limited number of pins, a number of functions such as the auxiliary PWM and the serial communication port are multiplexed with the nine programmable input/output (PIO) pins. The pin functions can be independently selected to allow maximum flexibility for different applications.

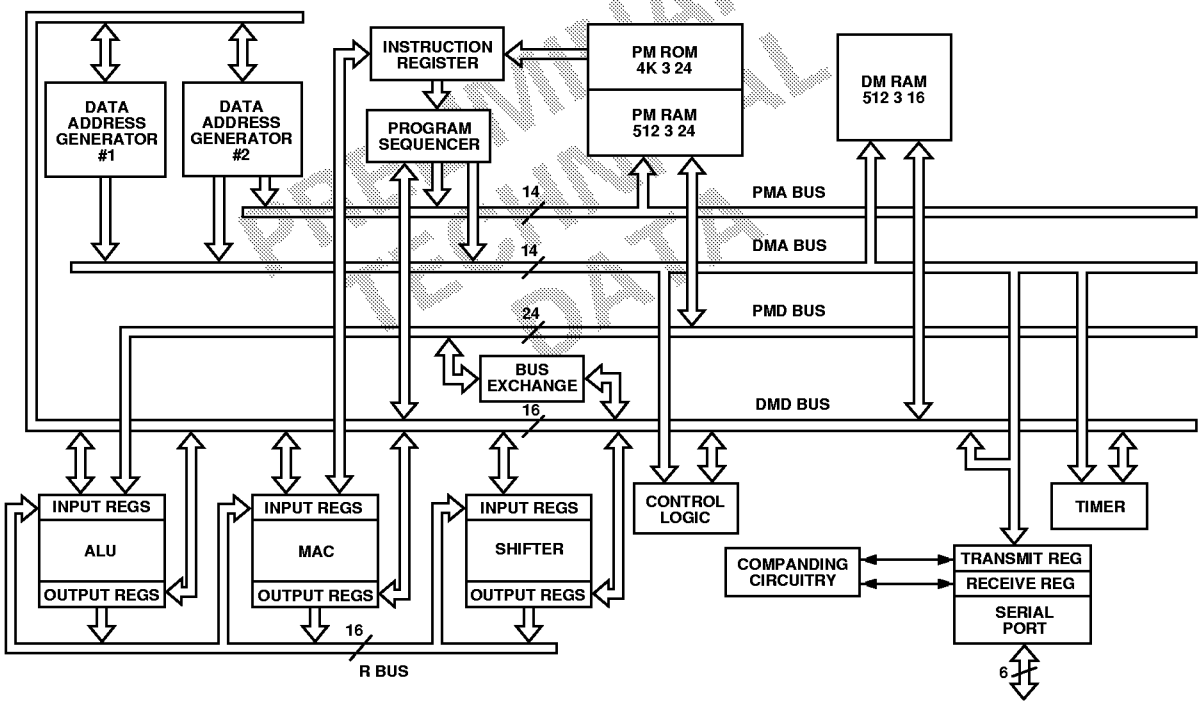


Figure 3. DSP Core Block Diagram

DSP CORE ARCHITECTURE OVERVIEW

Figure 3 is an overall block diagram of the DSP core of the ADMC328, which is based on the fixed-point ADSP-2171. The flexible architecture and comprehensive instruction set of the ADSP-2171 allow the processor to perform multiple operations in parallel. In one processor cycle (50 ns with a 10 MHz CLKIN) the DSP core can:

- Generate the next program address.
- Fetch the next instruction.
- Perform one or two data moves.
- Update one or two data address pointers.
- Perform a computational operation.

This all takes place while the processor continues to:

- Receive and transmit through the serial port.
- Decrement the interval timer.
- Generate three-phase PWM waveforms for a power inverter.
- Generate two signals using the 8-bit auxiliary PWM timers.
- Acquire four analog signals.
- Decrement the watchdog timer.

The processor contains three independent computational units: the arithmetic and logic unit (ALU), the multiplier/accumulator (MAC) and the shifter. The computational units process 16-bit data directly and have provisions to support multiprecision computations. The ALU performs a standard set of arithmetic and logic operations as well as providing support for division primitives. The MAC performs single-cycle multiply, multiply/add, and multiply/subtract operations with 40 bits of accumulation. The shifter performs logical and arithmetic shifts, normalization, denormalization and derives-exponent operations. The shifter can be used to efficiently implement numeric format control, including floating-point representations.

The internal result (R) bus directly connects the computational units so that the output of any unit may be the input of any unit on the next cycle.

A powerful program sequencer and two dedicated data address generators ensure efficient delivery of operands to these computational units. The sequencer supports conditional jumps and subroutine calls and returns in a single cycle. With internal loop counters and loop stacks, the ADMC328 executes looped code with zero overhead; no explicit jump instructions are required to maintain the loop.

Two data address generators (DAGs) provide addresses for simultaneous dual operand fetches from data memory and program memory. Each DAG maintains and updates four address pointers (I registers). Whenever the pointer is used to access data (indirect addressing), it is post-modified by the value in one of four modify (M) registers. A length value may be associated with each pointer (L registers) to implement automatic modulo addressing for circular buffers. The circular buffering feature is also used by the serial ports for automatic data transfers to and from on-chip memory. DAG1 generates only data memory address and provides an optional bit-reversal capability. DAG2 may generate either program or data memory addresses but has no bit-reversal capability.

Efficient data transfer is achieved with the use of five internal buses:

- Program memory address (PMA) bus.
- Program memory data (PMD) bus.
- Data memory address (DMA) bus.
- Data memory data (DMD) bus.
- Result (R) bus.

Program memory can store both instructions and data, permitting the ADMC328 to fetch two operands in a single cycle—one from program memory and one from data memory. The ADMC328 can fetch an operand from on-chip program memory and the next instruction in the same cycle.

The ADMC328 writes data from its 16-bit registers to the 24-bit program memory using the PX register to provide the lower eight bits. When it reads data (not instructions) from 24-bit program memory to a 16-bit data register, the lower eight bits are placed in the PX register.

The ADMC328 can respond to a number of distinct DSP core and peripheral interrupts. The DSP interrupts comprise a serial port receive interrupt, a serial port transmit interrupt, a timer interrupt, and two software interrupts. Additionally, the motor control peripherals include two PWM interrupts and a PIO interrupt.

The serial port (SPORT1) provides a complete synchronous serial interface with optional companding in hardware and a wide variety of framed and unframed data transmit and receive modes of operation. SPORT1 can generate an internal programmable serial clock or accept an external serial clock.

A programmable interval counter is also included in the DSP core and can be used to generate periodic interrupts. A 16-bit count register (TCOUNT) is decremented every *n* processor cycles, where n–1 is a scaling value stored in the 8-bit TSCALE register. When the value of the counter reaches zero, an interrupt is generated, and the count register is reloaded from a 16-bit period register (TPERIOD).

The ADMC328 instruction set provides flexible data moves and multifunction (one or two data moves with a computation) instructions. Each instruction is executed in a single 50 ns processor cycle (for a 10 MHz CLKIN). The ADMC328 assembly language uses an algebraic syntax for ease of coding and readability. A comprehensive set of development tools support program development. For further information on the DSP core, refer to the *ADSP-2100 Family User's Manual, Third Edition*, with particular reference to the ADSP-2171.

Serial Port

The ADMC328 incorporates a complete synchronous serial port (SPORT1) for serial communication and multiprocessor communication. The following is a brief list of capabilities of the ADMC328 SPORT1. Refer to the *ADSP-2100 Family User's Manual, Third Edition,* for further details.

- SPORT1 is bidirectional and has a separate, double-buffered transmit and receive section.
- SPORT1 can use an external serial clock or generate its own serial clock internally.
- SPORT1 has independent framing for the receive and transmit sections. Sections run in a frameless mode or with frame synchronization signals internally or externally generated. Frame synchronization signals are active-high or inverted, with either of two pulsewidths and timings.
- SPORT1 supports serial data word lengths from 3 bits to 16 bits and provides optional A-law and μ-law companding according to ITU (formerly CCITT) recommendation G.711.
- SPORT1 receive and transmit sections can generate unique interrupts on completing a data word transfer.
- SPORT1 can receive and transmit an entire circular buffer of data with only one overhead cycle per data word. An interrupt is generated after a data buffer transfer.
- SPORT1 can be configured to have two external interrupts (IRQ0 and IRQ1), and the Flag In and Flag Out signals. The internally generated serial clock may still be used in this configuration.
- SPORT1 has two data receive pins (DR1A and DR1B), which are internally multiplexed onto the one DR1 port of the SPORT1. The particular data receive pin selected is determined by a bit in the MODECTRL register.

PIN FUNCTION DESCRIPTION

The ADMC328 is available in a 28-lead SOIC package. Table I describes the pins.

Table I. Pin List

Pin Group Name	# of Pins	Input/ Output	Function
RESET	1	I	Processor Reset Input
SPORT1[1]	6	I/O	Serial Port 1 Pins (TFS1, RFS1, DT1, DR1A, DR1B, SCLK1)
CLKOUT[1]	1	O	Processor Clock Output
CLKIN, XTAL	2	I, O	External Clock or Quartz Crystal Connection Point
PIO0–PIO8[1]	9	I/O	Digital I/O Port Pins
AUX0–AUX1[1]	2	O	Auxiliary PWM Outputs
AH–CL	6	O	PWM Outputs
PWMTRIP	1	I	PWM Trip Signal
V1–V2	2	I	Analog Inputs
VAUX0–VAUX2	3	I	Auxiliary Analog Onput
I_{SENSE}	1	I	Current Sense Amplifier Input
ICONST	1	O	ADC Constant Current Source
V_{DD}	1		Power Supply
GND	1		Ground

NOTE
[1]Multiplexed pins, selectable individually through PIOSELECT and PIODATA1.

INTERRUPT OVERVIEW

The ADMC328 can respond to 16 different interrupt sources with minimal overhead, 5 of which are internal DSP core interrupts and 11 are from the motor control peripherals. The 5 DSP core interrupts are SPORT1 receive (or $\overline{IRQ0}$) and transmit (or $\overline{IRQ1}$), the internal timer, and two software interrupts. The motor control peripheral interrupts are the 9 programmable I/Os and two from the PWM (PWMSYNC pulse and $\overline{PWMTRIP}$). All motor control interrupts are multiplexed into the DSP core through the peripheral $\overline{IRQ2}$ interrupt. The interrupts are internally prioritized and individually maskable. A detailed description of the entire interrupt system of the ADMC328 is presented later, following a more detailed description of each peripheral block.

Memory Map

The ADMC328 has two distinct memory types: program memory and data memory. In general, program memory contains user code and coefficients, while the data memory is used to store variables and data during program execution. Both program memory RAM and ROM are provided on the ADMC328. Program memory RAM is arranged as one contiguous 512 3 24-bit block, starting at address 0x0000. Program memory ROM is a 4K × 24-bit block located at address 0x0800. Data memory is arranged as a 512 × 16-bit block starting at address 0x3800. The motor control peripherals are memory mapped into a region of the data memory space starting at 0x2000. The complete program and data memory maps are given in Tables II and III, respectively.

Table II. Program Memory Map

Address Range	Memory Type	Function
0x0000–0x002F	RAM	Interrupt Vector Table
0x0030–0x01FF	RAM	User Program RAM
0x0800–0x0919	ROM	Reserved Program ROM
0x091A–0x17EF	ROM	User Program ROM
0x017F0–0x17FF	ROM	Reserved Program ROM

Table III. Data Memory Map

Address Range	Memory Type	Function
0x2000–0x20FF		Memory Mapped Registers
0x2100–0x37FF		Reserved
0x3800–0x39FE	RAM	User Data Memory
0x39FF–0x39FF	RAM	Reserved for IRQFlag_Save
0x3A00–0x3BFF	RAM	Reserved
0x3C00–0x3FFF		DSP Memory Mapped Registers

SYSTEM INTERFACE

Figure 4 shows a basic system configuration for the ADMC328 with an external crystal.

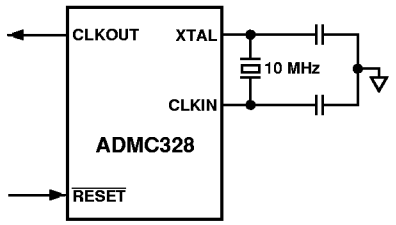


Figure 4. Basic System Configuration

Clock Signals

The ADMC328 can be clocked by either a crystal or a TTL-compatible clock signal. For normal operation, the CLKIN input cannot be halted, changed during operation or operated below the specified minimum frequency. If an external clock is used, it should be a TTL-compatible signal running at half the instruction rate. The signal is connected to the CLKIN pin of the ADMC328. In this mode, with an external clock signal, the XTAL pin must be left unconnected. The ADMC328 uses an input clock with a frequency equal to half the instruction rate; a 10 MHz input clock yields a 50 ns processor cycle (which is equivalent to 20 MHz). Normally, instructions are executed in a single processor cycle. All device timing is relative to the internal instruction rate, which is indicated by the CLKOUT signal, when enabled.

Because the ADMC328 includes an on-chip oscillator feedback circuit, an external crystal may be used instead of a clock source, as shown in Figure 4. The crystal should be connected across the CLKIN and XTAL pins, with two capacitors as shown in Figure 4. A parallel-resonant, fundamental frequency, microprocessor-grade crystal should be used. A clock output signal (CLKOUT) is generated by the processor at the processor's cycle rate of twice the input frequency.

Reset

The ADMC328 DSP core and peripherals must be correctly reset when the device is powered up to assure proper initialization. The ADMC328 contains an integrated power-on reset (POR) circuit which provides a complete system reset on power-up and power-down. The POR circuit monitors the voltage on the ADMC328 V_{DD} pin and holds the DSP core and peripherals in reset while V_{DD} is less than the threshold voltage level, V_{RST}. When this voltage is exceeded, the ADMC328 is held in reset for an additional 2^{16} DSP clock cycles (t_{RST} in Figure 5). On power-down, when the voltage on the V_{DD} pin falls below V_{RST}–V_{HYST}, the ADMC328 will be reset. Also, if the external $\overline{\text{RESET}}$ pin is actively pulled low at any time after power up, a complete hardware reset of the ADMC328 is initiated.

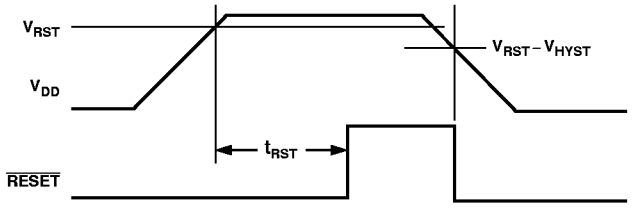


Figure 5. Power-On Reset Operation

The ADMC328 reset sets all internal stack pointers to the empty stack condition, masks all interrupts, clears the MSTAT register and performs a full reset of all of the motor control peripherals. Following a power-up, it is possible to initiate a DSP core and motor control peripheral reset by pulling the $\overline{\text{RESET}}$ pin low. The $\overline{\text{RESET}}$ signal must meet the minimum pulsewidth specification, t_{RSP}. Following the reset sequence, the DSP core starts executing code from the internal PM ROM located at 0x0800.

DSP Control Registers

The DSP core has a system control register, SYSCNTL, memory mapped at DM (0x3FFF). SPORT1 is configured as a serial port when Bit 10 is set, or as flags and interrupt lines when this bit is cleared. For proper operation of the ADMC328, all other bits in this register must be cleared (which is their default).

The DSP core has a wait state control register, MEMWAIT, memory mapped at DM (0x3FFE). For proper operation of the ADMC328, this register must always contain the value 0x8000 (which is the default).

The configuration of both the SYSCNTL and MEMWAIT registers of the ADMC328 is shown at the end of the data sheet.

THREE-PHASE PWM CONTROLLER

Overview

The PWM generator block of the ADMC328 is a flexible, programmable, three-phase PWM waveform generator that can be programmed to generate the required switching patterns to drive a three-phase voltage source inverter for ac induction motors (ACIM) or permanent magnet synchronous motors (PMSM). In addition, the PWM block contains special functions that considerably simplify the generation of the required PWM switching patterns for control of electronically commutated motors (ECM) or brushless dc motors (BDCM).

The PWM generator produces three pairs of active-high PWM signals on the six PWM output pins (AH, AL, BH, BL, CH and CL). The six PWM output signals consist of three high side drive signals (AH, BH and CH) and three low side drive signals (AL, BL and CL). The switching frequency, dead time and minimum pulsewidths of the generated PWM patterns are programmable using respectively the PWMTM, PWMDT and PWMPD registers. In addition, three registers (PWMCHA, PWMCHB and PWMCHC) control the duty cycles of the three pairs of PWM signals.

Each of the six PWM output signals can be enabled or disabled by separate output enable bits of the PWMSEG register. In addition, three control bits of the PWMSEG register permit crossover of the two signals of a PWM pair for easy control of ECM or BDCM. In crossover mode, the PWM signal destined for the high side switch is diverted to the complementary low side output, and the signal destined for the low side switch is diverted to the corresponding high side output signal.

In many applications, there is a need to provide an isolation barrier in the gate-drive circuits that turn on the power devices of the inverter. In general, there are two common isolation techniques: optical isolation using optocouplers, and transformer isolation using pulse transformers. The PWM controller of the ADMC328 permits mixing of the output PWM signals with a high frequency chopping signal to permit an easy interface to such pulse transformers. The features of this gate-drive chopping mode can be controlled by the PWMGATE register. There

is an 8-bit value within the PWMGATE register that directly controls the chopping frequency. In addition, high frequency chopping can be independently enabled for the high side and the low side outputs using separate control bits in the PWMGATE register.

The PWM generator is capable of operating in two distinct modes: single update mode or double update mode. In single update mode, the duty cycle values are programmable only once per PWM period, so that the resultant PWM patterns are symmetrical about the midpoint of the PWM period. In the double update mode, a second updating of the PWM duty cycle values is implemented at the midpoint of the PWM period. In this mode, it is possible to produce asymmetrical PWM patterns that produce lower harmonic distortion in three-phase PWM inverters. This technique also permits the closed loop controller to change the average voltage applied to the machine winding at a faster rate, allowing wider closed loop bandwidths to be achieved. The operating mode of the PWM block (single or double update mode) is selected by a control bit in MODECTRL register.

The PWM generator of the ADMC328 also provides an internal signal which synchronizes the PWM switching frequency to the A/D operation. In single update mode, a PWMSYNC pulse is produced at the start of each PWM period. In double update mode, an additional PWMSYNC pulse is produced at the midpoint of each PWM period. The width of the PWMSYNC pulse is programmable through the PWMSYNCWT register.

The PWM signals produced by the ADMC328 can be shut-off in a number of different ways. First, there is a dedicated asynchronous PWM shutdown pin, $\overline{\text{PWMTRIP}}$, which, when brought LO, instantaneously places all six PWM outputs in the OFF state. In addition, PWM shutdown is initiated when the voltage on the analog input pin (I_{SENSE}) is pulled 550 mV below ground, corresponding to an overcurrent fault. Because these two hardware shutdown mechanisms are asynchronous, and the associated PWM disable circuitry does not use clocked logic, the PWM will shutdown even if the DSP clock is not running. The PWM system may also be shutdown from software by writing to the PWMSWT register.

Status information about the PWM system of the ADMC328 is available to the user in the SYSSTAT register. In particular, the state of $\overline{\text{PWMTRIP}}$ is available, as well as a status bit that indicates whether operation is in the first half or the second half of the PWM period.

A functional block diagram of the PWM controller is shown in Figure 6. The generation of the six output PWM signals on pins AH to CL is controlled by four important blocks:

- The three-phase PWM timing unit, which is the core of the PWM controller, generates three pairs of complemented and dead time adjusted center based PWM signals.
- The output control unit allows the redirection of the outputs of the three-phase timing unit for each channel to either the high side or the low side output. In addition, the output control unit allows individual enabling/disabling of each of the six PWM output signals.
- The GATE drive unit provides the high chopping frequency and its subsequent mixing with the PWM signals.
- The PWM shutdown controller manages the three PWM shutdown modes (via the $\overline{\text{PWMTRIP}}$ pin, the analog block or the PWMSWT register) and generates the correct $\overline{\text{RESET}}$ signal for the Timing Unit
- The PWM controller is driven by a clock at the same frequency as the DSP instruction rate, CLKOUT, and is capable of generating two interrupts to the DSP core. One interrupt is generated on the occurrence of a PWMSYNC pulse, and the other is generated on the occurrence of any PWM shutdown action.

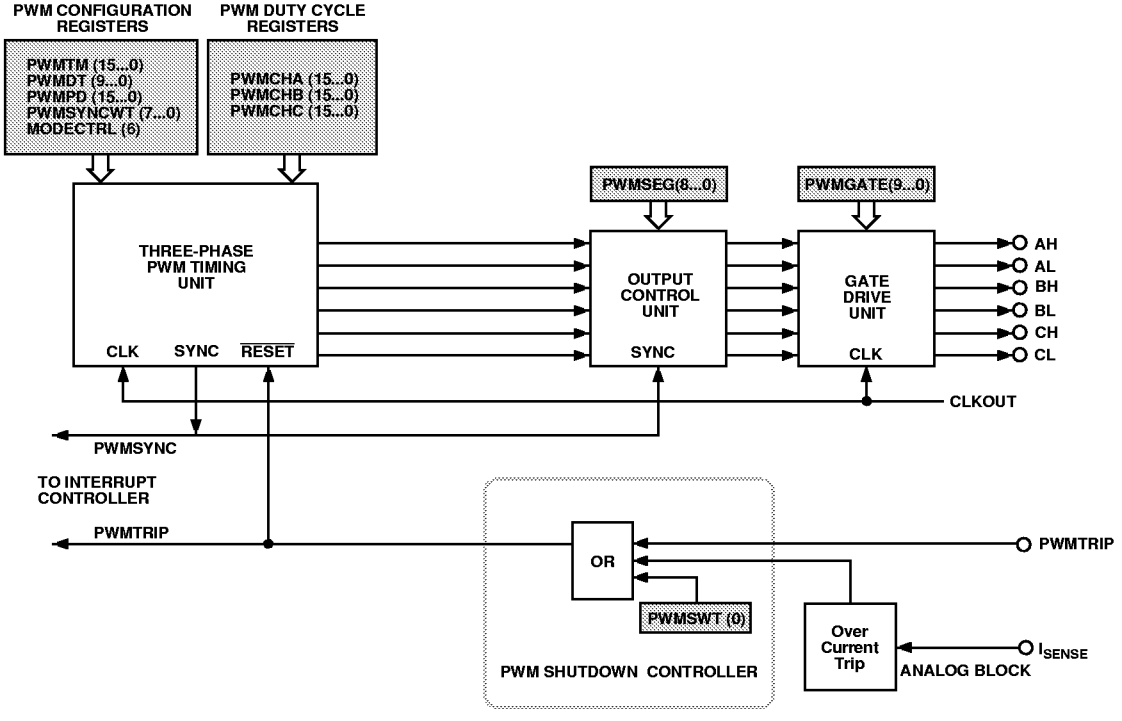


Figure 6. Overview of the PWM Controller of the ADMC328

Three-Phase Timing Unit

The 16-bit three-phase timing unit is the core of the PWM controller and produces three pairs of pulsewidth modulated signals with high resolution and minimal processor overhead. There are four main configuration registers (PWMTM, PWMDT, PWMPD and PWMSYNCWT) that determine the fundamental characteristics of the PWM outputs. In addition, the operating mode of the PWM (single or double update mode) is selected by Bit 6 of the MODECTRL register. These registers in conjunction with the three 16-bit duty cycle registers (PWMCHA, PWMCHB and PWMCHC) control the output of the three-phase timing unit.

PWM Switching Frequency: PWMTM Register

The PWM switching frequency is controlled by the PWM period register, PWMTM. The fundamental timing unit of the PWM controller is $t_{CK} = 1/f_{CLKOUT}$ where f_{CLKOUT} is the CLKOUT frequency (DSP instruction rate). Therefore, for a 20 MHz CLKOUT, the fundamental time increment is 50 ns. The value written to the PWMTM register is effectively the number of t_{CK} clock increments in half a PWM period. The required PWMTM value is a function of the desired PWM switching frequency (f_{PWM}) and is given by:

$$PWMTM = \frac{f_{CLKOUT}}{2 \times f_{PWM}} = \frac{f_{CLKIN}}{f_{PWM}}$$

Therefore, the PWM switching period, T_S, can be written as:

$$T_S = 2 \times PWMTM \times t_{CK}$$

For example, for a 20 MHz CLKOUT and a desired PWM switching frequency of 10 kHz (T_S = 100 µs), the correct value to load into the PWMTM register is:

$$PWMTM = \frac{20 \times 10^6}{2 \times 10 \times 10^3} 1000 = 0x3E8$$

The largest value that can be written to the 16-bit PWMTM register is 0xFFFF = 65,535 which corresponds to a minimum PWM switching frequency of:

$$f_{PWM,\min} = \frac{20 \times 10^6}{2 \times 65{,}535} = 153\ Hz$$

for a CLKOUT frequency of 20 MHz.

PWM Switching Dead Time: PWMDT Register

The second important PWM block parameter that must be initialized is the switching dead time. This is a short delay time introduced between turning off one PWM signal (for example AH) and turning on its complementary signal, AL. This short time delay is introduced to permit the power switch being turned off to completely recover its blocking capability before the complementary switch is turned on. This time delay prevents a potentially destructive short-circuit condition from developing across the dc link capacitor of a typical voltage source inverter.

Dead time is controlled by the PWMDT register. The dead time is inserted into the three pairs of PWM output signals. The dead time, T_D, is related to the value in the PWMDT register by:

$$T_D = PWMDT \times 2 \times t_{CK} = 2 \times \frac{PWMDT}{f_{CLKOUT}}$$

Therefore, a PWMDT value of 0x00A (= 10), introduces a 1 µs delay between the turn off of any PWM signal (for example AH) and the turn on of its complementary signal (AL). The amount of the dead time can therefore be programmed in increments of 2 t_{CK} (or 100 ns for a 20 MHz CLKOUT). The PWMDT register is a 10-bit register. For a CLKOUT rate of 20 MHz its maximum value of 0x3FF (= 1023) corresponds to a maximum programmed dead time of:

$$\begin{aligned} T_{D\,max} &= 1023 \times 2 \times t_{CK} \\ &= 1023 \times 2 \times 50 \times 10^{-9}\ sec \\ &= 102\ \mu s \end{aligned}$$

The dead time can be programmed to be zero by writing 0 to the PWMDT register.

PWM Operating Mode: MODECTRL and SYSSTAT Registers

The PWM controller of the ADMC328 can operate in two distinct modes: single update mode and double update mode. The operating mode of the PWM controller is determined by the state of Bit 6 of the MODECTRL register. If this bit is cleared the PWM operates in the single update mode. Setting Bit 6 places the PWM in the double update mode. By default, following either a peripheral reset or power on, Bit 6 of the MODECTRL register is cleared. This means the default operating mode is single update mode.

In single update mode, a single PWMSYNC pulse is produced in each PWM period. The rising edge of this signal marks the start of a new PWM cycle and is used to latch new values from the PWM configuration registers (PWMTM, PWMDT, PWMPD and PWMSYNCWT) and the PWM duty cycle registers (PWMCHA, PWMCHB and PWMCHC) into the three-phase timing unit. The PWMSEG register is also latched into the output control unit on the rising edge of the PWMSYNC pulse. In effect, this means that the parameters of the PWM signals can be updated only once per PWM period at the start of each cycle. Thus, the generated PWM patterns are symmetrical about the midpoint of the switching period.

In double update mode, there is an additional PWMSYNC pulse produced at the midpoint of each PWM period. The rising edge of this new PWMSYNC pulse is again used to latch new values of the PWM configuration registers, duty cycle registers and the PWMSEG register. As a result, it is possible to alter both the characteristics (switching frequency, dead time, minimum pulsewidth and PWMSYNC pulsewidth) and the output duty cycles at the midpoint of each PWM cycle. Consequently, it is possible to produce PWM switching patterns that are no longer symmetrical about the midpoint of the period (asymmetrical PWM patterns).

In the double update mode, operation in the first half or the second half of the PWM cycle is indicated by Bit 3 of the SYSSTAT register. In double update mode, this bit is cleared during operation in the first half of each PWM period (between the rising edge of the original PWMSYNC pulse and the rising edge of the new PWMSYNC pulse which is introduced in double update mode). Bit 3 of the SYSSTAT register is set during the second half of each PWM period. If required, a user may determine the status of this bit during a PWMSYNC interrupt service routine.

The advantages of the double update mode are that lower harmonic voltages can be produced by the PWM process and wider control bandwidths are possible. However, for a given PWM switching frequency, the PWMSYNC pulses occur at twice the rate in the double update mode. Because new duty cycle values

must be computed in each PWMSYNC interrupt service routine, there is a larger computational burden on the DSP in the double update mode.

Width of the PWMSYNC Pulse: PWMSYNCWT Register

The PWM controller of the ADMC328 produces an internal PWM synchronization pulse at a rate equal to the PWM switching frequency in single update mode and at twice the PWM frequency in the double update mode. This PWMSYNC synchronizes the operation of the PWM unit with the A/D converter system. The width of this PWMSYNC pulse is programmable by the PWMSYNCWT register. The width of the PWMSYNC pulse, $T_{PWMSYNC}$, is given by:

$$T_{PWMSYNC} = t_{CK} \times (PWMSYNCWT + 1)$$

which means the width of the pulse is programmable from t_{CK} to 256 t_{CK} (corresponding to 50 ns to 12.8 µs for a CLKOUT rate of 20 MHz). Following a reset, the PWMSYNCWT register contains 0x27 (= 39) so that the default PWMSYNC width is 2.0 µs.

PWM Duty Cycles: PWMCHA, PWMCHB, PWMCHC Registers

The duty cycles of the six PWM output signals are controlled by the three duty cycle registers, PWMCHA, PWMCHB and PWMCHC. The integer value in the register PWMCHA controls the duty cycle of the signals on AH and AL. PWMCHB controls the duty cycle of the signals on BH and BL, and PWMCHC controls the duty cycle of the signals on CH and CL. The duty cycle registers are programmed in integer counts of the fundamental time unit, t_{CK}, and define the desired on-time of the high-side PWM signal produced by the three-phase timing unit over half the PWM period. The switching signals produced by the three-phase timing unit are also adjusted to incorporate the programmed dead time value in the PWMDT register.

The PWM is center-based. This means that in single update mode the resulting output waveforms are symmetrical and centered in the PWMSYNC period. Figure 7 presents a typical PWM timing diagram illustrating the PWM-related registers' (PWMCHA, PWMTM, PWMDT, and PWMSYNCWT) control over the waveform timing in both half cycles of the PWM period. The magnitude of each parameter in the timing diagram is determined by multiplying the integer value in each register by t_{CK} (typically 50 ns). It may be seen in the timing diagram how dead time is incorporated into the waveforms by moving the switching edges away from the instants set by the PWMCHA register.

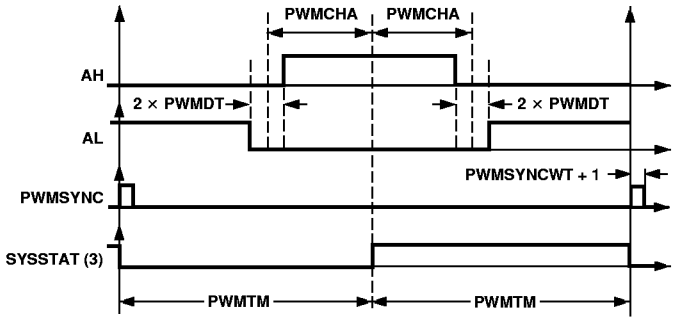


Figure 7. Typical PWM Outputs of Three-Phase Timing Unit in Single Update Mode

Each switching edge is moved by an equal amount (PWMDT × t_{CK}) to preserve the symmetrical output patterns. The PWMSYNC pulse, whose width is set by the PWMSYNCWT register, is also shown. Bit 3 of the SYSTAT register indicates which half cycle is active. This can be useful in double update mode, as will be discussed later.

The resultant on-times of the PWM signals shown in Figure 7 may be written as:

$$T_{AH} = 2 \times (PWMCHA - PWMDT) \times t_{CK}$$

$$T_{AL} = 2 \times (PWMTM - PWMCHA - PWMDT) \times t_{CK}$$

The corresponding duty cycles are:

$$d_{AH} = \frac{T_{AH}}{T_S} = \frac{PWMCHA - PWMDT}{PWMTM}$$

$$d_{AL} = \frac{T_{AL}}{T_S} = \frac{PWMTM - PWMCHA - PWMDT}{PWMTM}$$

Obviously, negative values of T_{AH} and T_{AL} are not permitted because the minimum permissible value is zero, corresponding to a 0% duty cycle. In a similar fashion, the maximum value is T_S, corresponding to a 100% duty cycle.

The output signals from the timing unit for operation in double update mode are shown in Figure 8. This illustrates a completely general case where the switching frequency, dead time and duty cycle are all changed in the second half of the PWM period. Of course, the same value for any or all of these quantities could be used in both halves of the PWM cycle. However, it can be seen that there is no guarantee that symmetrical PWM signals will be produced by the timing unit in this double update mode. Additionally, it is seen that the dead time is inserted into the PWM signals in the same way as in the single update mode.

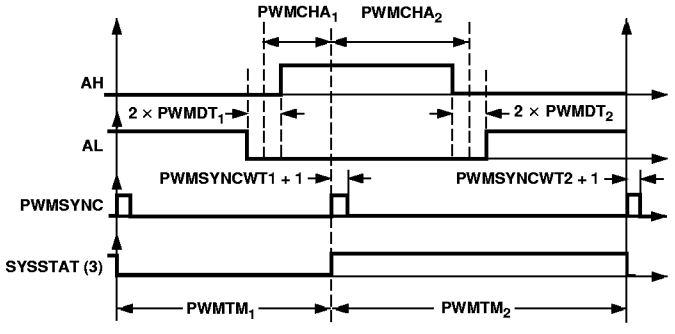


Figure 8. Typical PWM Outputs of Three-Phase Timing Unit in Double Update Mode

In general, the on-times of the PWM signals in double update mode are defined by:

$$T_{AH} = (PWMCHA_1 + PWMCHA_2 - PWMDT_1 - PWMDT_2)\ 3\ t_{CK}$$

$$T_{AL} = (PWMTM_1 + PWMTM_2 - PWMCHA_1 - PWMCHA_2 - PWMDT_1 - PWMDT_2)3\ t_{CK}$$

where the subscript 1 refers to the value of that register during the first half cycle and the subscript 2 refers to the value during the second half cycle. The corresponding duty cycles are:

$$d_{AH} = \frac{T_{AH}}{T_S}$$

$$= \frac{(PWMCHA_1 + PWMCHA_2)}{(PWMTM_1 + PWMTM_2)}$$

$$= \frac{(PWMDT_1 + PWMDT_2)}{(PWMTM_1 + PWMTM_2)}$$

$$d_{AL} = \frac{T_{AL}}{T_S}$$

$$= \frac{(PWMTM_1 + PWMTM_2 + PWMCHA_1)}{(PWMTM_1 + PWMTM_2)}$$

$$= \frac{(PWCHA_2 + PWMDT_1 + PWMDT_2)}{(PWMTM_1 + PWMTM_2)}$$

because for the completely general case in double update mode, the switching period is given by:

$$T_S = (PWMTM_1 + PWMTM_2)\ 3\ t_{CK}$$

Again, the values of T_{AH} and T_{AL} are constrained to lie between zero and T_S.

PWM signals similar to those illustrated in Figure 7 and Figure 8 can be produced on the BH, BL, CH and CL outputs by programming the PWMCHB and PWMCHC registers in a manner identical to that described for PWMCHA.

The PWM controller does not produce any PWM outputs until all of the PWMTM, PWMCHA, PWMCHB, and PWMCHC registers have been written to at least once. After these registers have been written, the counters in the three-phase timing unit are enabled. Writing to these registers also starts the main PWM timer. If during initialization, the PWMTM register is written after the PWMCHA, PWMCHB, and PWMCHC registers, then the first PWMSYNC pulse (and interrupt if enabled) will be generated (1.5 3 t_{CK} × PWMTM) seconds after the initial write to the PWMTM register in single update mode. In double update mode, the first PWMSYNC pulse will be generated (t_{CK} × PWMTM) seconds after the initial write to the PWMTM register in single update mode.

Effective PWM Resolution

In single update mode, the same values of PWMCHA, PWMCHB and PWMCHC are used to define the on-times in both half cycles of the PWM period. As a result the effective resolution of the PWM generation process is 2 t_{CK} (or 100 ns for a 20 MHz CLKOUT) since incrementing one of the duty cycle registers by 1 changes the resultant on-time of the associated PWM signals by t_{CK} in each half period (or 2 t_{CK} for the full period).

In double update mode, improved resolution is possible since different values of the duty cycles registers are used to define the on-times in both the first and second halves of the PWM period. As a result, it is possible to adjust the on-time over the whole period in increments of t_{CK}. This corresponds to an effective PWM resolution of t_{CK} in double update mode (or 50 ns for a 20 MHz CLKOUT).

The achievable PWM switching frequency at a given PWM resolution is tabulated in Table V.

Table V. Achievable PWM Resolution in Single and Double Update Modes

Resolution (Bit)	Single Update Mode PWM Frequency (kHz)	Double Update Mode PWM Frequency (kHz)
8	39.1	78.1
9	19.5	39.1
10	9.8	19.5
11	4.9	9.8
12	2.4	4.9

Minimum Pulsewidth: PWMPD Register

In many power converter switching applications, it is desirable to eliminate PWM switching pulses shorter than a certain width. It takes a finite time to both turn on and turn off modern power semiconductor devices. Therefore, if the width of any of the PWM pulses is shorter than some minimum value, it may be desirable to completely eliminate the PWM switching for that particular cycle.

The allowable minimum on-time for any of the six PWM outputs over half a PWM period that can be produced by the PWM controller may be programmed using the PWMPD register. The minimum on-time is programmed in increments of t_{CK} so that the minimum on-time that will be produced over any half PWM period, T_{MIN}, is related to the value in the PWMPD register by:

$$T_{MIN} = PWMPD \times t_{CK}$$

A *PWMPD* value of 0x002 defines a permissible minimum on-time of 100 ns for a 20 MHz CLKOUT.

In each half cycle of the PWM, the timing unit checks the on-time of each of the six PWM signals. If any of the times is found to be less than the value specified by the PWMPD register, the corresponding PWM signal is turned OFF for the entire half period, and its complementary signal is turned completely ON.

Consider the example where PWMTM = 200, PWMCHA = 5, PWMDT = 3, and PWMPD = 10 with a CLKOUT of 20 MHz while operating in single update mode. For this case, the PWM switching frequency is 50 kHz and the dead time is 300 ns. The minimum permissible on-time of any PWM signal over one-half of any period is 500 ns. Clearly, for this example, the dead-time adjusted on-time of the AH signal for one-half a PWM period is (5-3)350 ns = 100 ns. Because this is less than the minimum

permissible value, output AH of the timing unit will remain OFF (0% duty cycle). Additionally, the AL signal will be turned ON for the entire half period (100% duty cycle).

Output Control Unit: PWMSEG Register

The operation of the output control unit is managed by the 9-bit read/write PWMSEG register. This register sets two distinct features of the output control unit that are directly useful in the control of ECM or BDCM.

The PWMSEG register contains three crossover bits, one for each pair of PWM outputs. Setting Bit 8 of the PWMSEG register enables the crossover mode for the AH/AL pair of PWM signals; setting Bit 7 enables crossover on the BH/BL pair of PWM signals; and setting Bit 6 enables crossover on the CH/CL pair of PWM signals. If crossover mode is enabled for any pair of PWM signals, the high-side PWM signal from the timing unit (for example AH) is diverted to the associated low-side output of the output control unit so that the signal will ultimately appear at the AL pin. Of course, the corresponding low-side output of the timing unit is also diverted to the complementary high-side output of the output control unit so that the signal appears at Pin AH. Following a reset, the three crossover bits are cleared so that the crossover mode is disabled on all three pairs of PWM signals.

The PWMSEG register also contains six bits (Bits 0 to 5) that can be used to individually enable or disable each of the six PWM outputs. If the associated bit of the PWMSEG register is set, then the corresponding PWM output is disabled regardless of the value of the corresponding duty cycle register. This PWM output signal will remain in the OFF state as long as the corresponding enable/disable bit of the PWMSEG register is set. The PWM output enable function gates the crossover function. After a reset, all six enable bits of the PWMSEG register are cleared, thereby enabling all PWM outputs by default.

In a manner identical to the duty cycle registers, the PWMSEG is latched on the rising edge of the PWMSYNC signal so that changes to this register only become effective at the start of each PWM cycle in single update mode. In double update mode, the PWMSEG register can also be updated at the midpoint of the PWM cycle.

In the control of an ECM, only two inverter legs are switched at any time, and often the high-side device in one leg must be switched ON at the same time as the low-side driver in a second leg. Therefore, by programming identical duty cycles for two PWM channels (for example, let PWMCHA = PWMCHB) and setting Bit 7 of the PWMSEG register to crossover the BH/BL pair of PWM signals, it is possible to turn ON the high-side switch of Phase A and the low-side switch of Phase B at the same time. In the control of an ECM, one inverter leg (Phase C in this example) is disabled for a number of PWM cycles. This disable may be implemented by disabling both the CH and CL PWM outputs by setting Bits 0 and 1 of the PWMSEG register. This is illustrated in Figure 9 where it can be seen that both the AH and BL signals are identical, because PWMCHA = PWMCHB, and the crossover bit for Phase B is set. In addition, the other four signals (AL, BH, CH and CL) have been disabled by setting the appropriate enable/disable bits of the PWMSEG register. For the situation illustrated in Figure 9, the appropriate value for the PWMSEG register is 0x00A7. In ECM operation, because each inverter leg is disabled for certain periods of time, the PWMSEG register is changed based upon the position of the rotor shaft (motor commutation).

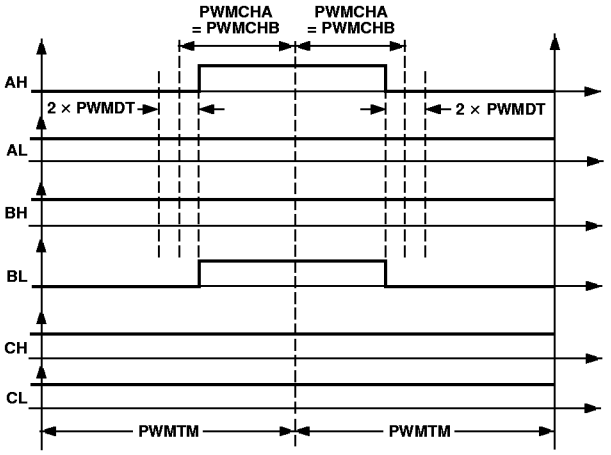


Figure 9. An example of PWM signals suitable for ECM control. PWMCHA = PWMCHB, BH/BL are a crossover pair. AL, BH, CH and CL outputs are disabled. Operation is in single update mode.

Gate Drive Unit: PWMGATE Register

The gate drive unit of the PWM controller adds features which simplify the design of isolated gate drive circuits for PWM inverters. If a transformer coupled power device gate drive amplifier is used, the active PWM signal must be chopped at a high frequency. The PWMGATE register allows the programming of this high frequency chopping mode. The chopped active PWM signals may be required for the high-side drivers only, for the low-side drivers only, or for both the high-side and low-side switches. Therefore, independent control of this mode for both high and low-side switches is included with two separate control bits in the PWMGATE register.

Typical PWM output signals with high-frequency chopping enabled on both high-side and low-side signals are shown in Figure 10. Chopping of the high-side PWM outputs (AH, BH and CH) is enabled by setting Bit 8 of the PWMGATE register. Chopping of the low-side PWM outputs (AL, BL and CL) is enabled by setting Bit 9 of the PWMGATE register. The high chopping frequency is controlled by the 8-bit word (GDCLK) written to Bits 0 to 7 of the PWMGATE register. The period and the frequency of this high frequency carrier are:

$$T_{CHOP} = \left[4 \times \left(GDCLK + 1\right) \times t_{CK}\right]$$

$$f_{CHOP} = \frac{f_{CLKOUT}}{\left[4 \times \left(GDCLK + 1\right)\right]}$$

The *GDCLK* value may range from 0 to 255, corresponding to a programmable chopping frequency rate from 19.5 kHz to 5 MHz for a 20 MHz CLKOUT rate. The gate drive features must be programmed before operation of the PWM controller and typically are not changed during normal operation of the PWM controller. Following a reset, by default, all bits of the PWMGATE register are cleared so that high frequency chopping is disabled.

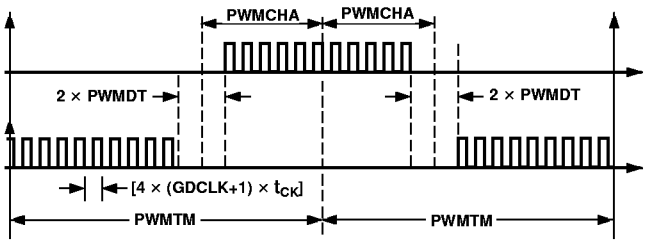


Figure 10. Typical PWM signals with high-frequency gate chopping enabled on both high-side and low-side switches (GDCLK is the integer equivalent of the value in Bits 0 to 7 of the PWMGATE register.)

PWM Shutdown

In the event of external fault conditions, it is essential that the PWM system be instantaneously shutdown. Two methods of sensing a fault condition are provided by the ADMC328. For the first method, a falling edge on the $\overline{\text{PWMTRIP}}$ pin initiates an instantaneous, asynchronous (independent of DSP clock) shutdown of he PWM controller. This places all six PWM outputs in the OFF state, disables the PWMSYNC pulse and associated interrupt signal and generates a $\overline{\text{PWMTRIP}}$ interrupt signal. The $\overline{\text{PWMTRIP}}$ pin has an internal pull-down resistor so that even if the pin becomes disconnected, the PWM outputs will be disabled. The state of the $\overline{\text{PWMTRIP}}$ pin can be read from bit 0 of the SYSTAT register.

The second method for detecting a fault condition is through the I_{SENSE} pin in the analog block of the ADMC328. The I_{SENSE} pin monitors the feedback signals from a dc bus current sensing resistor which represents the total current in the motor. When the voltage of I_{SENSE} goes below I_{SENSE} trip threshold, $\overline{\text{PWMTRIP}}$ will be internally pulled low. The negative edge of the internal $\overline{\text{PWMTRIP}}$ will generate a shutdown in the same manner as a negative edge on pin $\overline{\text{PWMTRIP}}$. This fault condition corresponds to a 5.5 ampere trip current in a 0.10 Ω sense resistor in the dc power bus.

It is possible through software to initiate a PWM shutdown by writing to the 1-bit read/write PWMSWT register (0x2061). Writing to this bit generates a PWM shutdown in a manner identical to the $\overline{\text{PWMTRIP}}$ or I_{SENSE} pins. Following a PWM shutdown, it is possible to determine if the shutdown was generated from hardware or software by reading the same PWMSWT register. Reading this register also clears it.

Restarting the PWM after a fault condition is detected requires clearing the fault and reinitializing the PWM. Clearing the fault requires that $\overline{\text{PWMTRIP}}$ returns to a HI state and I_{SENSE} returns to a voltage greater than the I_{SENSE} trip threshold. After the fault has been cleared, the PWM can be restarted by writing to registers PWMTM, PWMCHA, PWMCHB, and PWMCHC. After the fault is cleared and the PWM registers are initialized, internal timing of the three phase timing unit will resume, and the new duty-cycle values will be latched on the next rising edge of PWMSYNC.

PWM Registers

The configuration of the PWM registers is described at the end of the data sheet.

ADC OVERVIEW

Analog Block

The ADC of the ADMC328 is based upon the single slope conversion technique. While such an approach does not offer the self correction of a dual slope converter, it is an inherently monotonic conversion process and thus to within the noise and stability of its components, there will be no missing codes. Coupled with the temperature compensation techniques used on the ADMC328, this provides a repeatable conversion which is near the resolution of the 12-bit ADC counter. The single slope technique has been adapted so that 4 input channels are converted simultaneously.

The functional schematic diagram, Figure 11, shows a multiplexer in series with one of the ADC inputs. Table VI shows the multiplexer input selection codes. One of these auxiliary multiplexed channels is used to calibrate the ramp against the internal 2% reference (VREF). Note that two of the main inputs are directly connected as high impedance voltage inputs. The third main input channel has a special design to monitor the voltage on a current sensing resistor whose voltage is always below (more negative than) the power supply ground rail.

Table VI. ADC Auxiliary Channel Selection

Select	MODECTRL (1) ADCMUX1	MODECTRL (0) ADCMUX0
VAUX0	0	0
VAUX1	0	1
VAUX2	1	0
Calibration (V_{REF})	1	1

The ramp voltage used to perform the conversion is generated by driving a fixed current into an off-chip capacitor, where the capacitor voltage is

$$V_C = (I/C) \times t$$

Following reset, $V_C = 0$ at $t = 0$. This reset and the start of the conversion process are initiated by the PWMSYNC pulse, as shown in Figure 12. In order to compensate for IC process manufacturing tolerances (and to adjust for capacitor tolerances), the current source of the ADMC328 is software programmable. The software setting of the magnitude of the ICONST current generator is accomplished by selecting one of 8 steps over a 20% current range. This is discussed in more detail below.

An inherent limitation of the single slope technique is that it is slow because the conversion resolution depends upon the ratio of the ramp time to reach full scale, to the clock period. Attempting to increase conversion speed by steepening the ramp (shortening the time) results in fewer counts for a full scale reading, and thus more coarse resolution. In addition, the voltage inputs to the ADMC328 have no sample and hold function. Therefore, the actual conversion point is the time point intersection of the input voltage and the ramp voltage (V_C) which is also shown in Figure 12.

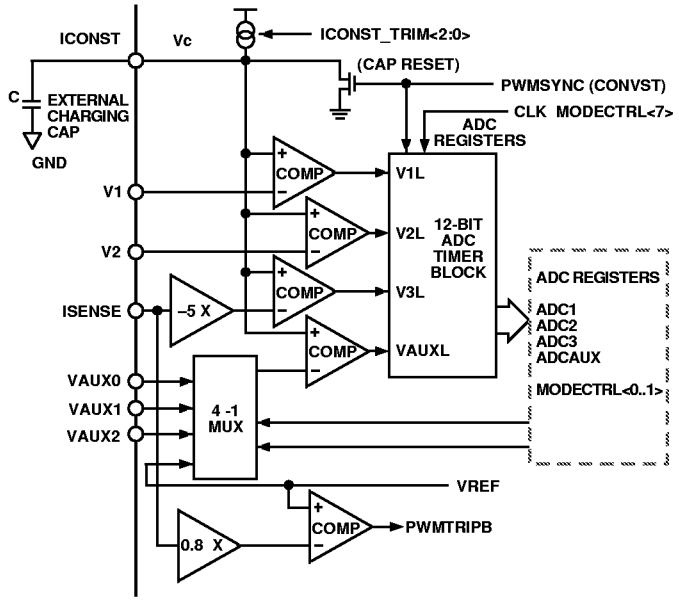


Figure 11. ADC Overview

There are some special characteristics of the I_{SENSE} input that must be considered. The I_{SENSE} input is connected to an inverting amplifier with a gain of (nominally) 5 and it accepts an input voltage within the range of –400 mV to 0 V with respect to ground. The I_{SENSE} input amplifier also has a sample-an-hold input. In addition, the I_{SENSE} input can generate a PWM trip signal which is ORed with other PWM on-chip trip signals. The I_{SENSE} trip is generated by asynchronous circuits which do not require a system clock. The trip voltage is set to a nominal value of –550 mV at the I_{SENSE} terminal. Selection of the I_{SENSE} resistor value determines both the conversion (transimpedance) gain and the trip current.

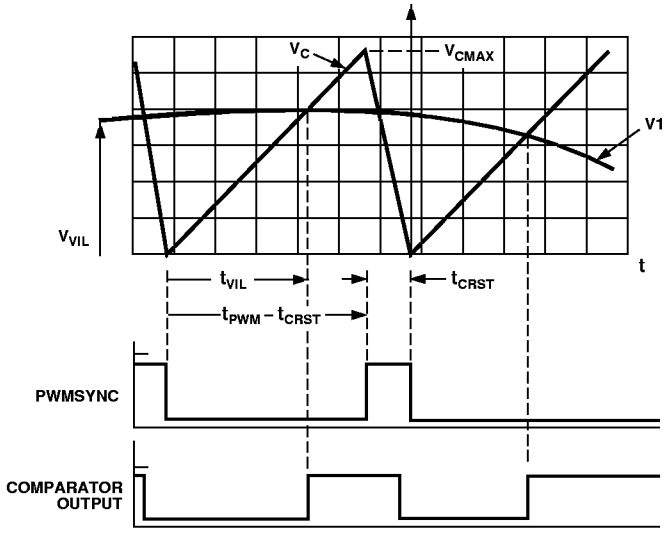


Figure 12. Analog Input Block Operation

There are four ADC timers, one for each channel. These timers are reset to zero at the start of each conversion by the PWMSYNC pulse. Each timer counts until its comparator's input equals the ramp voltage which stops the counter thus capturing the time (and therefore the voltage) when its input voltage equals the ramp voltage. Each counter is a 12-bit register clocked at a rate controlled by the ADDCNT bit, Bit (7) in the MODECTROL register. When this bit is cleared, the timers count at the slower rate of CLKIN. When this bit is set, they count at CLKOUT or twice the rate of CLKIN. ADC1, ADC2, ADC3 and ADCAUX are the registers which capture the conversion times.

ADC Resolution

Because the operation of the ADC is intrinsically linked to the PWM block, the effective resolution of the ADC is a function of the PWM switching frequency. The effective ADC resolution is determined by the rate at which the counter timer is clocked. The counter clock rate, as stated above, is controlled by the ADCCNT bit (Bit 7) of the MODECTRL register. For a CLKOUT period of t_{CK} and a PWM period of T_{PWM}, the maximum count of the ADC is given by:

$$Max\ Count = min\ (4095, (T_{PWM} - T_{CRST})/2\ t_{CK})$$

for MODECTRL Bit 7 = 0

$$Max\ Count = min\ (4095, (T_{PWM} - T_{CRST})/t_{CK})$$

for MODECTRL Bit 7 = 1

Where T_{PWM} is equal to the PWM period if operating in single update mode, or it is equal to half that period if operating in double update mode. For an assumed CLKOUT frequency of 20 MHz and PWMSYNC pulsewidth of 2.0 µs, the effective resolution of the ADC block is tabulated for various PWM switching frequencies in Table VII.

Table VII. ADC Resolution Examples

PWM Freq. (kHz)	MODECTRL[7] = 0 Max Count	MODECTRL[7] = 0 Effective Resolution	MODECTRL[7] = 1 Max Count	MODECTRL[7] = 1 Effective Resolution
2.5	4095	12	4095	12
4	3230	>11	4095	12
8	1605	>10	3210	>11
18	702	>9	1404	>10
24	521	>9	1043	>10

External Timing Capacitor

In order to maximize the useful input voltage range and effective resolution of the ADC, it is necessary to select the value of the external timing capacitor. For a given capacitance value, C_{nom}, the peak ramp voltage is given by:

$$V_{C\,MAX} = \frac{ICONST_{NON} \times (T_{PWM} - T_{CRST})}{C_{NON}}$$

where $ICONST_{NOM}$ is the nominal current source value of 100 µA and T_{CRST} is the PWMSYNC pulsewidth. When the appropriate capacitor is used, it will be possible to program the current source to compensate for its variation in the nominal value and the capacitor tolerance.

Table VIII. Timing Capacitor Selection

PWM Frequency (kHz) MODECTRL[7] = 0	PWM Frequency (kHz) MODECTRL[7] = 1	Timing Capacitor (nF)
2.1–2.7	4.2–5.2	15
2.7–3.2	5.2–6.3	12
3.2–3.9	6.3–7.7	10
3.9–4.7	7.7–9.2	8.2
4.7–5.6	9.2–11.2	6.8
5.6–6.7	11.2–13.3	5.6
6.7–8.0	13.3–16.0	4.7
8.0–9.5	16.0–18.8	3.9
9.5–11.5	18.8–23.0	3.3
11.5–14.1	23.0–28.1	2.7
14.1–17.1	28.1–34.1	2.2
17.1–20.4	34.1–40.8	1.8
20.4–25.3	40.8–50.6	1.5
25.3–30.1	50.6–60.2	1.2

Programmable Current Source

The ADMC328 has an internal current source which is used to charge an external capacitor, generating the voltage ramp used for conversion. The magnitude of the output of the current source circuit is subject to manufacturing variations and can vary from one device to the next. Therefore, the ADMC328 incudes a programmable current source whose output can always be tuned to within 5% of the target 100 µA. A 3-bit register, ICONST_TRIM, allows the user to make this adjustment. The output current is proportional to the value written to the register: 0x0 produces the minimum output, and 0x7 produces the maximum output. The default value of ICONST_TRIM after reset is 0x0. Because of manufacturing variations, the minimum current source output for any particular part can range from 64 µA to 100 µA. Similarly, the maximum output can range from 100 µA to 155 µA.

ADC Reference Ramp Calibration

The programmable current source can be used to compensate the reference ramp slope for both the current source manufacturing variations and the external charging capacitor tolerance. The current source calibration sets the reference ramp to within 5% of its target slope. The following is a brief description of a simple calibration algorithm which can be implemented in the user's code. The target reference ramp, shown in Figure 13, is matched to the 0.3 V–3.5 V signal range of the ADC comparators. Since the default output current after reset is a minimum, the default slope is also a minimum. This initial slope is determined by converting the reference voltage, VREF, on the ADC auxiliary channel(refer to Table VI). A single conversion is used to determine the value written to the ICONST_TRIM register which will produce the target reference ramp slope. A detailed description of the calibration process is available in an application note.

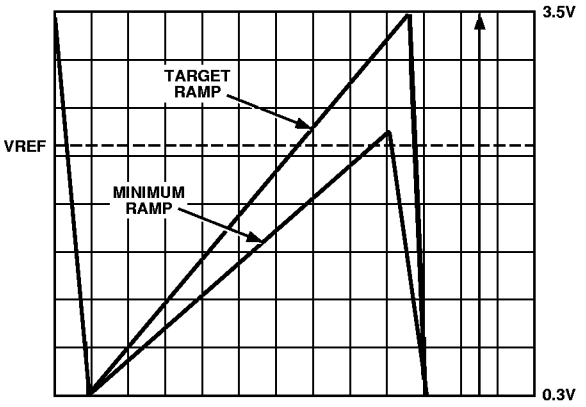


Figure 13. Current Ramp

Current Sense Amplifier Application

The ADMC328 current sense amplifier system has been provided to simplify the measurement of the motor winding currents in brushless dc motor control systems. The assumed power circuit configuration, illustrated in Figure 14, is one in which a current sense resistor is placed between the circuit common and the return path to the negative power bus. The normal PWM modulation scheme keeps one upper device fully conducting while the duty cycle of one of the lower power switches is varied. In this case, there is a negative going voltage across the resistive shunt when the complementary upper diode conducts. The shunt signal, I_{bus} shown in Figure 15, is sampled at the midpoint of the lower device on period using the PWMSYNC pulse. The captured value represents the current in the motor winding $I_{WINDING}$.

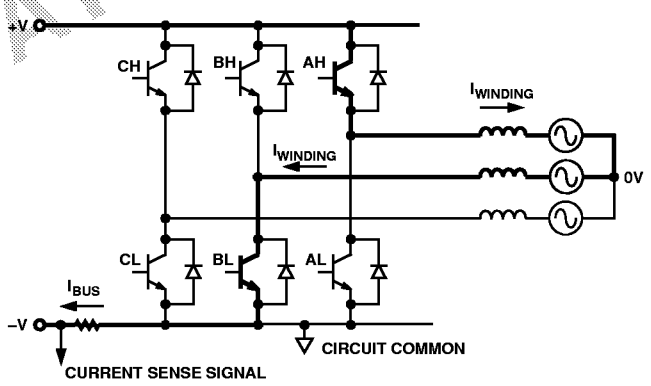


Figure 14. Typical Power Inverter Switching Topology for DC Brushless Motor Control

Current Sense Amplifier

The ADMC328 analog circuit block also integrates a gain of five inverting amplifier, a sample-and-hold amplifier, and an overcurrent-trip comparator. The current sense amplifier input signal range from 0 mV to –600 mV is matched to the requirements of medium to low power motor control applications. There is a 300 mV output offset which matches the amplifier output signal range to the 0.3 V to 3.5 V input signal range of the A/D converter. This amplifier is followed by a sample-and-hold amplifier which samples the current sense signal on the falling edge of the PWMSYNC pulse. This sampling amplifier system can be used to capture the winding current signal in a brushless dc motor.

The overcurrent-trip-comparator trips when the 2.5 V reference voltage equals 80% of the amplifier output. Thus, this circuit will generate a $\overline{\text{PWMTRIP}}$ signal when the voltage across the resistor is larger than 570 mV.

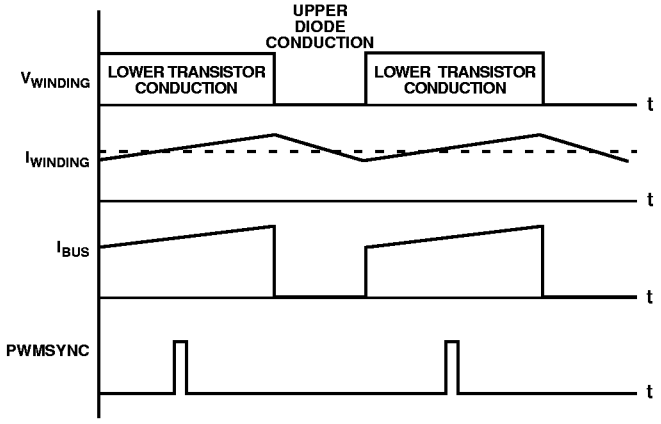


Figure 15. Bus Current Signals

ADC Registers

The configuration of all registers of the ADC System is shown at the end of the data sheet.

AUXILIARY PWM TIMERS

Overview

The ADMC328 provides two variable-frequency, variable duty cycle, 8-bit, auxiliary PWM outputs that are available at the AUX1 and AUX0 pins when enabled. These auxiliary PWM outputs can be used to provide switching signals to other circuits in a typical motor control system such as power factor corrected front-end converters or other switching power converters. Alternatively, by addition of a suitable filter network, the auxiliary PWM output signals can be used as simple single-bit digital-to-analog converters.

The auxiliary PWM system of the ADMC328 can operate in two different modes: independent mode, or offset mode. The operating mode of the auxiliary PWM system is controlled by Bit 8 of the MODECTRL register. Setting Bit 8 of the MODECTRL register places the auxiliary PWM system in the independent mode. In this mode, the two auxiliary PWM generators are completely independent and separate switching frequencies and duty cycles may be programmed for each auxiliary PWM output. In this mode, the 8-bit AUXTM0 register sets the switching frequency of the signal at the AUX0 output pin. Similarly, the 8-bit AUXTM1 register sets the switching frequency of the signal at the AUX1 pin. The fundamental time increment for the auxiliary PWM outputs is twice the DSP instruction rate (or 2 t_{CK}) and the corresponding switching periods are given by:

$$T_{AUX0} = 2 \times (AUXTM0 + 1) \times t_{CK}$$

$$T_{AUX1} = 2 \times (AUXTM1 + 1) \times t_{CK}$$

Since the values in both AUXTM0 and AUXTM1 can range from 0 to 0xFF, the achievable switching frequency of the auxiliary PWM signals may range from 39.1 kHz to 10 MHz for a CLKOUT frequency of 20 MHz.

The on-time of the two auxiliary PWM signals is programmed by the two 8-bit AUXCH0 and AUXCH1 registers, according to:

$$T_{ON,\ AUX0} = 2 \times (\text{AUXCH0}) \times t_{CK}$$

$$T_{ON,\ AUX1} = 2 \times (\text{AUXCH1}) \times t_{CK}$$

so that output duty cycles from 0% to 100% are possible. Duty cycles of 100% are produced if the on-time value exceeds the period value. Typical auxiliary PWM waveforms in independent mode are shown in Figure 16(a).

When Bit 8 of the MODECTRL register is cleared, the auxiliary PWM channels are placed in offset mode. In offset mode, the switching frequency of the two signals on the AUX0 and AUX1 pins are identical and controlled by AUXTM0 in a manner similar to that previously described for independent mode. In addition, the on times of both the AUX0 and AUX1 signals are controlled by the AUXCH0 and AUXCH1 registers as before. However, in this mode the AUXTM1 register defines the offset time from the rising edge of the signal on the AUX0 pin to that on the AUX1 pin according to:

$$T_{OFFSET} = 2 \times (AUXTM1 + 1) \times t_{CK}$$

For correct operation in this mode, the value written to the *AUXTM*1 register must be less than the value written to the *AUXTM*0 register. Typical auxiliary PWM waveforms in offset mode are shown in Figure 16(b). Again, duty cycles from 0% to 100% are possible in this mode.

In both operating modes, the resolution of the auxiliary PWM system is 8-bits only at the minimum switching frequency (AUXTM0 = AUXTM1 = 255 in independent mode, AUXTM0 = 255 in offset mode). Obviously, as the switching frequency is increased, the resolution is reduced.

Values can be written to the auxiliary PWM registers at any time. However, new duty cycle values written to the AUXCH0 and AUXCH1 registers only become effective at the start of the next cycle. Writing to the AUXTM0 or AUXTM1 registers causes the internal timers to be reset to 0 and new PWM cycles to begin.

By default following a reset, Bit 8 of the MODECTRL register is cleared thus enabling offset mode. In addition, the registers AUXTM0 and AUXTM1 default to 0xFF, corresponding to the minimum switching frequency and zero offset. The on-time registers AUXCH0 and AUXCH1 default to 0x00.

Auxiliary PWM Interface, Registers and Pins
The registers of the auxiliary PWM system are summarized at the end of the data sheet.

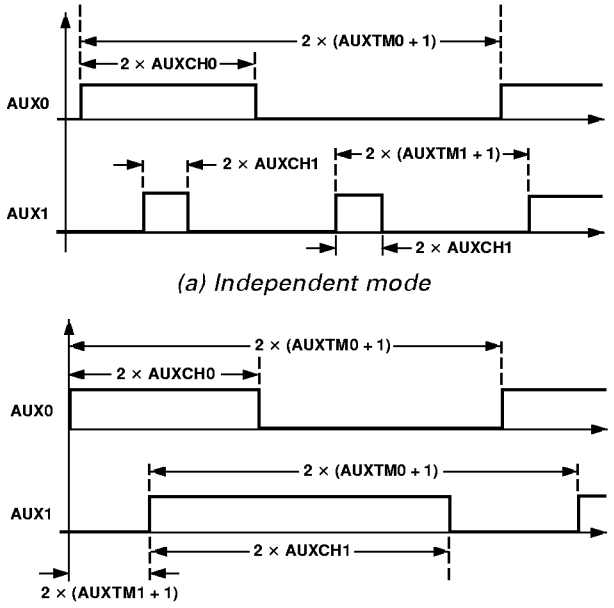


Figure 16. Typical auxiliary PWM signals. (All times in increments of t_{CK}.)

PWM DAC Equation
The auxiliary PWM output can be filtered in order to produce a low frequency analog signal between 0 V to V_{DD}. For example, a 2-pole filter with a 1.2 kHz cutoff frequency will sufficiently attenuate the PWM carrier. Figure 17 shows how the filter would be applied.

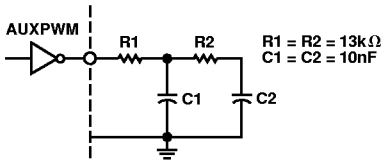


Figure 17. Auxiliary PWM Output Filter

WATCHDOG TIMER
The ADMC328 incorporates a watchdog timer that can perform a full reset of the DSP and motor control peripherals in the event of software error. The watchdog timer is enabled by writing a timeout value to the 16-bit WDTIMER register. The timeout value represents the number of CLKIN cycles required for the watchdog timer to count down to zero. When the watchdog timer reaches zero, a full DSP core and motor control peripheral reset is performed. In addition, Bit 1 of the SYSSTAT register is set so that after a watchdog reset, the ADMC328 can determine that the reset was due to the timeout of the watchdog timer and not an external reset. Following a watchdog reset, Bit 1 of the SYSSTAT register may be cleared by writing zero to the WDTIMER register. This clears the status bit but does not enable the watchdog timer.

On reset, the watchdog timer is disabled and is only enabled when the first timeout value is written to the WDTIMER register. To prevent the watchdog timer from timing out, the user must write to the WDTIMER register at regular intervals (shorter than the programmed WDTIMER period value). On all but the first write to WDTIMER, the particular value written to the register is unimportant since writing to WDTIMER simply reloads the first value written to this register.

PROGRAMMABLE DIGITAL INPUT/OUTPUT
The ADMC328 has 9 programmable digital input/output (PIO) pins that are all multiplexed with other functions. The 9 PIO lines PIO0–PIO8 are multiplexed with the serial port (Pins PIO0/TFS1 to PIO5/RFS1), the CLKOUT (pin PIO6/CLKOUT) and the auxiliary PWM outputs (Pins PIO7/AUX1 and PIO8/AUX0). When configured as a PIO, each of these 9 pins can act as an input, output, or an interrupt source.

The operating mode of pins PIO0/TFS1 to PIO7/AUX1 is controlled by the PIOSELECT register. This 8-bit register has a bit for each input so that the mode of each pin may be selected individually. Bit 0 of PIOSELECT controls the operation of the PIO0/TFS1 pin. Bit 1 controls the PIO1/DT1 pin, etc. Setting the appropriate bit in the PIOSELECT register causes the corresponding pin to be configured for PIO functionality. Clearing the bit selects the alternate (SPORT, CLKOUT, or AUXPWM) mode of the corresponding pin. Following power-on reset, all bits of PIOSELECT are set such that PIO functionality is selected. The operating mode of the PIO8/AUX0 pin is selected by bit 1 of the PIODATA1 register. In a manner identical to the PIOSELECT register, setting this bit enables PIO functionality (PIO8) while clearing the bit enables auxiliary PWM functionality (AUX0).

Once PIO functionality has been selected for any or all of these 9 pins, the direction may be set by the 8-bit PIODIR0 register (for PIO0-PIO7) and the 1-bit PIODIR1 register (for PIO8). Clearing any bit configures the corresponding PIO line as an input while setting the bit configures it as an output. By default, following a reset all bits of PIODIR0 and PIODIR1 are cleared configuring the PIO lines as inputs.

The data of the PIO0 to PIO8 lines is controlled by the PIODATA0 register (for PIO0 to PIO7) and Bit 0 of the PIODATA1 register (for PIO8). These registers can be used to read data from those PIO lines configured as inputs and write data to those configured as outputs. Any of the 9 pins that have been configured for PIO functionality can be made to act as an interrupt source by setting the appropriate bit of the PIOINTEN0 register (for PIO0 to PIO7) or the PIOINTEN1 register (for PIO8). In order to act as an interrupt source the pin must also be configured as an input. An interrupt is generated upon a change of state (low-to-high transition or high-to-low transition) on any input that has been configured as an interrupt source. Following a change of state event on any such input, the corresponding bit is set in the PIOFLAG0 register (for PIO0 to PIO7) and PIOFLAG1 (for PIO8) and a common PIO interrupt is generated. Reading the PIOFLAG0 and PIOFLAG1 registers permits determining the interrupt source. Reading the PIOFLAG0 and PIOFLAG1 registers automatically clears all bits of the registers. Following power-on or reset, all bits of PIOINTEN0 and PIOINTEN1 are cleared so that no interrupts are enabled.

Each PIO line has an internal pull-down resistor so that following power-on or reset all 9 lines are configured as input PIOs and will be read as logic lows if left unconnected.

PIO Registers
The configuration of all registers of the PIO system is shown at the end of the data sheet.

INTERRUPT CONTROL

The ADMC328 can respond to 8 different interrupt sources, some of which are generated by internal DSP core interrupts and others from the motor control peripherals. The DSP core interrupts include the following:

- Reserved.
- A Peripheral (or IRQ2) Interrupt.
- A SPORT1 Receive (or IRQ0) and a SPORT1 Transmit (or IRQ1) Interrupt.
- Two Software Interrupts
- An Interval Timer Time-Out Interrupt

The interrupts generated by the motor control peripherals include:

- A PWMSYNC Interrupt.
- A Programmable Input/Output (PIO) Interrupt.
- A PWM Trip Interrupt.

The core interrupts are internally prioritized and individually maskable. All peripheral interrupts are multiplexed into the DSP core through the peripheral ($\overline{IRQ2}$) interrupt.

The PWMSYNC interrupt is triggered by a low-to-high transition on the PWMSYNC pulse. The $\overline{PWMTRIP}$ interrupt is triggered on a high-to-low transition on the $\overline{PWMTRIP}$ pin, an overcurrent on the I_{SENSE} pin, or by writing to the PWMSWT register. A PIO interrupt is detected on any change of state (high-to-low or low-to-high) on the PIO lines.

The ADMC328 interrupt control system is configured and controlled by the IFC, IMASK and ICNTL registers of the DSP core and by the IRQFLAG register for the PWMSYNC and $\overline{PWMTRIP}$ interrupts. PIO interrupts are enabled and disabled by the PIOINTEN0 and PIOINTEN1 registers.

Table IX. Interrupt Vector Addresses

Interrupt Source	Interrupt Vector Address
$\overline{PWMTRIP}$	0x002C (Highest Priority)
Peripheral Interrupt ($\overline{IRQ2}$)	0x0004
PWMSYNC	0x000C
PIO	0x0008
Software Interrupt 1	0x0018
Software Interrupt 0	0x001C
SPORT1 Transmit Interrupt or $\overline{IRQ1}$	0x0020
SPORT1 Receive Interrupt or $\overline{IRQ0}$	0x0024
Timer	0x0028 (Lowest Priority)

Interrupt Masking

Interrupt masking (or disabling) is controlled by the IMASK register of the DSP core. This register contains individual bits that must be set to enable the various interrupt sources. If any peripheral interrupt is to be enabled, the $\overline{IRQ2}$ interrupt enable bit (Bit 9) of the IMASK register must be set. The configuration of the IMASK register of the ADMC328 is shown at the end of the data sheet.

Interrupt Configuration

The IFC and ICNTL registers of the DSP core control and configure the interrupt controller of the DSP core. The IFC register is a 16-bit register that may be used to force and/or clear any of the eight DSP interrupts. Bits 0 to 7 of the IFC register may be used to clear the DSP interrupts while Bits 8 to 15 can be used to force a corresponding interrupt. Writing to Bits 11 and 12 in IFC is the only way to create the two software interrupts.

The ICNTL register is used to configure the sensitivity (edge or level) of the $\overline{IRQ0}$, $\overline{IRQ1}$ and $\overline{IRQ2}$ interrupts and to enable/disable interrupt nesting. Setting Bit 0 of ICNTL configures the $\overline{IRQ0}$ as edge-sensitive, while clearing the bit configures it for level-sensitive. Bit 1 is used to configure the $\overline{IRQ1}$ interrupt. Bit 2 is used to configure the $\overline{IRQ2}$ interrupt. It is recommended that the $\overline{IRQ2}$ interrupt always be configured as level-sensitive to ensure that no peripheral interrupts are lost. Setting Bit 4 of the ICNTL register enables interrupt nesting. The configuration of both the IFC and ICNTL registers is shown at the end of the data sheet.

Interrupt Operation

Following a reset, the ROM code on the ADMC328 must copy a default interrupt vector table into program memory RAM from address 0x0000 to 0x002F. Since each interrupt source has a dedicated four-word space in this vector table, it is possible to code short interrupt service routines (ISR) in place. Alternatively, it may be necessary to insert a JUMP instruction to the appropriate start address of the interrupt service routine if more memory is required for the ISR.

When an interrupt occurs, the program sequencer ensures that there is no latency (beyond synchronization delay) when processing unmasked interrupts. In the case of the timer, SPORT1, and software interrupts, the interrupt controller automatically jumps to the appropriate location in the interrupt vector table. At this point, a JUMP instruction to the appropriate ISR is required.

Motor control peripheral interrupts are slightly different. When a peripheral interrupt is detected, a bit is set in the IRQFLAG register for PWMSYNC and $\overline{PWMTRIP}$ or in the PIOFLAG0, or PIOFLAG1 registers for a PIO interrupt, and the $\overline{IRQ2}$ line is pulled low until all pending interrupts are acknowledged. For any of the peripheral interrupts, the interrupt controller automatically jumps to location 0x0004 in the interrupt vector table. Code which is automatically loaded into location 0x0004 on reset, subsequently reads the IRQFLAG register to determine if the source of the interrupt was a PWM trip, a PWMSYNC or a PIO and jumps to the appropriate interrupt vector location.

The code automatically loaded at location 0x0004 by the monitor on reset is:

```
0x0004:ASTAT = DM(IRQFLAG);
  DM(IRQFLAG_SAVE) = ASTAT;
  IF EQ JUMP 0x002C
  IF LT JUMP 0x000C;
```

At this point, a JUMP instruction to the appropriate ISR, at the interrupt vector location shown in Table IX, is required. If more than one interrupt occurs simultaneously, the higher priority interrupt service routine is executed. Reading the IRQFLAG register clears the $\overline{PWMTRIP}$ and PWMSYNC bits and acknowledges the interrupt, thus allowing further interrupts when

the ISR exits. When the IRQFLAG register is read, it is saved in a data memory variable so the user ISR can check to see if there are simultaneous PWMSYNC and $\overline{\text{PWMTRIP}}$ interrupts.

A user's PIO interrupt service routine must read the PIOFLAG0 and PIOFLAG1 registers to determine which PIO port is the source of the interrupt. Reading registers PIOFLAG0 and PIOFLAG1 clears all bits in the registers and acknowledges the interrupt, thus allowing further interrupts after the ISR exits.

The configuration of all these registers is shown at the end of the data sheet.

SYSTEM CONTROLLER

The system controller block of the ADMC328 performs the following functions:

1. Manages the interface and data transfer between the DSP core and the motor control peripherals.
2. Handles interrupts generated by the motor control peripherals and generates a DSP core interrupt signal $\overline{\text{IRQ2}}$.
3. Controls the ADC multiplexer select lines.
4. Enables $\overline{\text{PWMTRIP}}$ and PWMSYNC interrupts.
5. Controls the multiplexing of the SPORT1 pins to select either DR1A or DR1B data receive pins. It also allows configuration of SPORT1 as a UART interface.
6. Controls the PWM single/double update mode.
7. Controls the ADC conversion time modes.
8. Controls the auxiliary PWM operation mode.
9. Contains a status register (SYSSTAT) that indicates the state of the $\overline{\text{PWMTRIP}}$ pin, the watchdog timer, and the PWM timer.
10. Performs a reset of the motor control peripherals and control registers following a hardware, software or watchdog initiated reset.

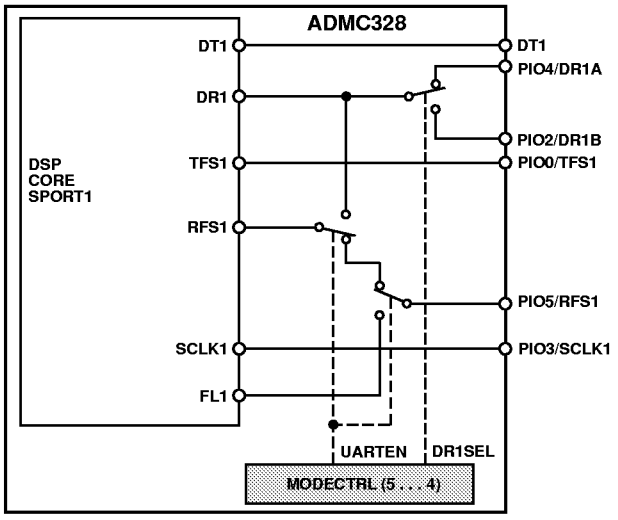


Figure 18. Internal Multiplexing of SPORT1 Pins

SPORT1 Control

Both data receive pins are multiplexed internally into the single data receive input of SPORT1 as shown in Figure 18. Two control bits in the MODECTRL register control the state of the SPORT1 pins by manipulating internal multiplexers in the ADMC328.

Bit 4 of the MODECTRL register (DR1SEL) selects between the two data receive pins. Setting Bit 4 of MODECTRL connects pin DR1B to the internal data receive port DR1 of SPORT1. Clearing Bit 4 connects DR1A to DR1.

Setting Bit 5 of the MODECTRL register (UARTEN) configures the serial port for UART mode. In this mode, the DR1 and RFS1 pins of the internal serial port are connected together. Additionally, setting the UARTEN bit connects the FL1 flag of the DSP to the external PIO5/RFS1 pin.

Flag Pins

The ADMC328 provides flag pins. The alternate configuration of SPORT1 includes a Flag In (FI) and Flag Out (FO) pin. This alternate configuration of SPORT1 is selected by Bit 10 of the DSP system control register, SYSCNTL at data memory address, 0x3FFF. In the alternate configuration, the DR1 pin (either DR1A or DR1B depending upon the state of the DR1SEL bit) becomes the FI pin and the DT1 pin becomes the FO pin. Additionally, RFS1 is configured as the $\overline{\text{IRQ0}}$ interrupt input and TFS1 is configured as the $\overline{\text{IRQ1}}$ interrupt. The serial port clock, SCLK1, is still available in the alternate configuration.

Development Kit

To facilitate device evaluation and programming, an emulation kit (ADMC32x-EMULKIT) is available from Analog Devices. The emulation kit consists of an emulation board and the motion control debugger software. The emulation kit contains the latest programming and device information. This kit is recommended for initial program development.

Table X. Peripheral Register Map

Address (HEX)	Name	Bits Used	Function
0x2000	ADC1	[15 . . . 4]	ADC Results for V1
0x2001	ADC2	[15 . . . 4]	ADC Results for V2
0x2002	ADC3	[15 . . . 4]	ADC Results for I_{SENSE}
0x2003	ADCAUX	[15 . . . 4]	ADC Results for VAUX
0x2004	PIODIR0	[7 . . . 0]	PIO0 . . . 7 Pins Direction Setting
0x2005	PIODATA0	[7 . . . 0]	PIO0 . . . 7 Pins Input/Output Data
0x2006	PIOINTEN0	[7 . . . 0]	PIO0 . . . 7 Pins Interrupt Enable
0x2007	PIOFLAG0	[7 . . . 0]	PIO0 . . . 7 Pins Interrupt Status
0x2008	PWMTM	[15 . . . 0]	PWM Period
0x2009	PWMDT	[9 . . . 0]	PWM Deadtime
0x200A	PWMPD	[9 . . . 0]	PWM Pulse Deletion Time
0x200B	PWMGATE	[9 . . . 0]	PWM Gate Drive Configuration
0x200C	PWMCHA	[15 . . . 0]	PWM Channel A Pulsewidth
0x200D	PWMCHB	[15 . . . 0]	PWM Channel B Pulsewidth
0x200E	PWMCHC	[15 . . . 0]	PWM Channel C Pulsewidth
0x200F	PWMSEG	[8 . . . 0]	PWM Segment Select
0x2010	AUXCH0	[7 . . . 0]	AUX PWM Output 0
0x2011	AUXCH1	[7 . . . 0]	AUX PWM Output 1
0x2012	AUXTM0	[7 . . . 0]	Auxiliary PWM Frequency Value
0x2013	AUXTM1	[7 . . . 0]	Auxiliary PWM Frequency Value/Offset
			Reserved
0x2015	MODECTRL	[8 . . . 0]	Mode Control Register
0x2016	SYSSTAT	[3 . . . 0]	System Status
0x2017	IRQFLAG	[1 . . . 0]	Interrupt Status
0x2018	WDTIMER	[15 . . . 0]	Watchdog Timer
0x2019 . . . 43			Reserved
0x2044	PIODIR1	[0]	PIO8 Pin Direction Setting
0x2045	PIODATA1	[1 . . . 0]	PIO8 Data and Mode Control
0x2046	PIOINTEN1	[0]	PIO8 Pin Interrupt Enable
0x2047	PIOFLAG1	[0]	PIO8 Pin Interrupt Status
0x2048			Reserved
0x2049	PIOSELECT	[7 . . . 0]	PIO0 to PIO7 Mode Select
0x204A . . . 5F			Reserved
0x2060	PWMSYNCWT	[7 . . . 0]	PWMSYNC Pulsewidth
0x2061	PWMSWT	[0]	PWM S/W Trip Bit
0x2062 . . . 67			Reserved
0x2068	ICONST_TRIM	[2. . .0]	ICONST Trim
0x2069 . . . FF			Reserved

Table XI. DSP Core Registers

Address	Name	Bits	Function
0x3FFF	SYSCNTL	[15 . . . 0]	System Control Register
0x3FFE	MEMWAIT	[15 . . . 0]	Memory Wait State Control Register
0x3FFD	TPERIOD	[15 . . . 0]	Interval Timer Period Register
0x3FFC	TCOUNT	[15 . . . 0]	Interval Timer Count Register
0x3FFB	TSCALE	[7 . . . 0]	Interval Timer Scale Register
0x3FFA . . . F3			Reserved
0x3FF2	SPORT1_CTRL_REG	[15 . . . 0]	SPORT1 Control Register
0x3FF1	SPORT1_SCLKDIV	[15 . . . 0]	SPORT1 Clock Divide Register
0x3FF0	SPORT1_RFSDIV	[15 . . . 0]	SPORT1 Receive Frame Sync Divide
0x3FEF	SPORT1_AUTOBUF_CTRL	[15 . . . 0]	SPORT1 Autobuffer Control Register

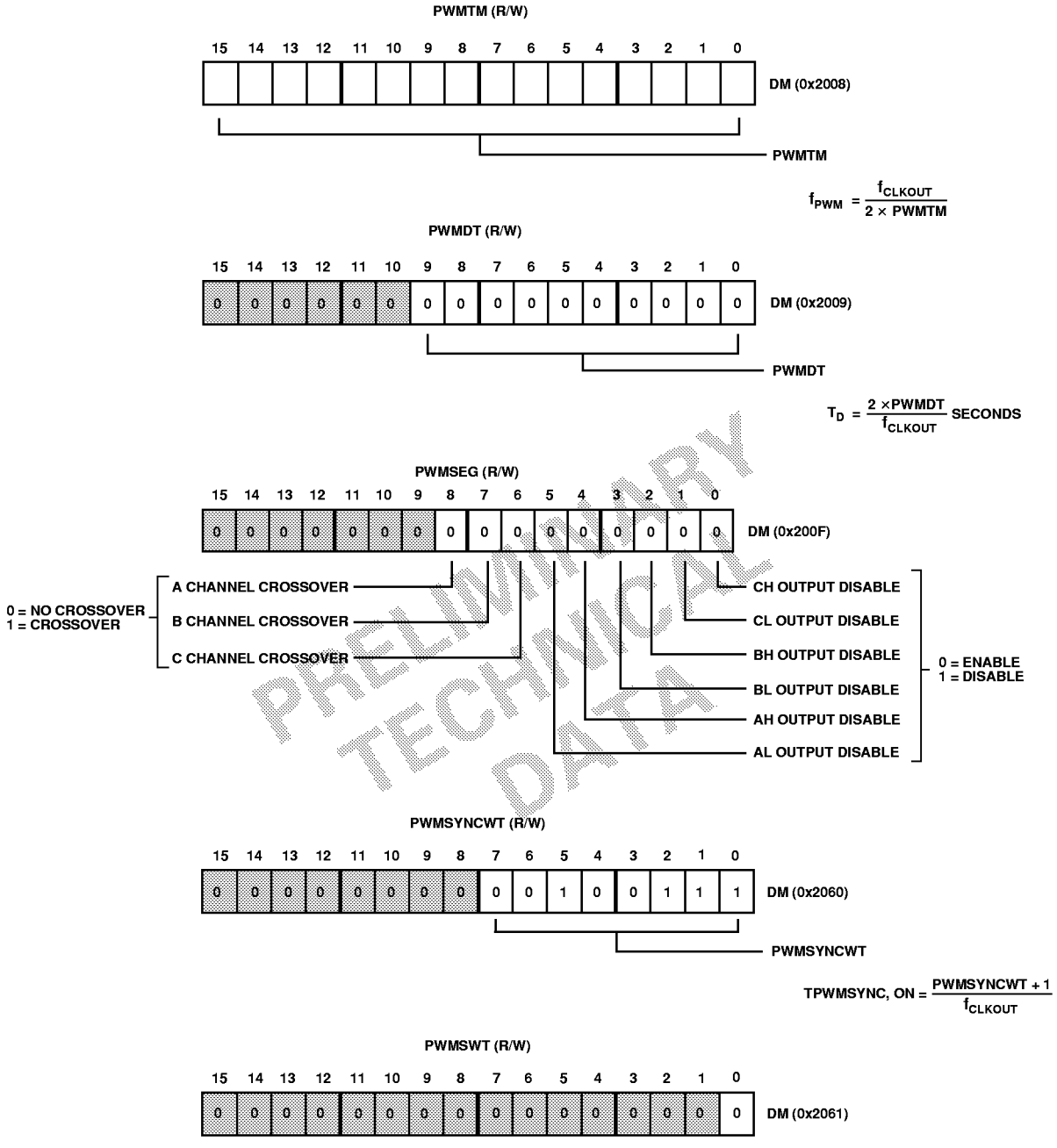


Figure 19. Configuration of PWM Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field—these bits should always be written as shown.

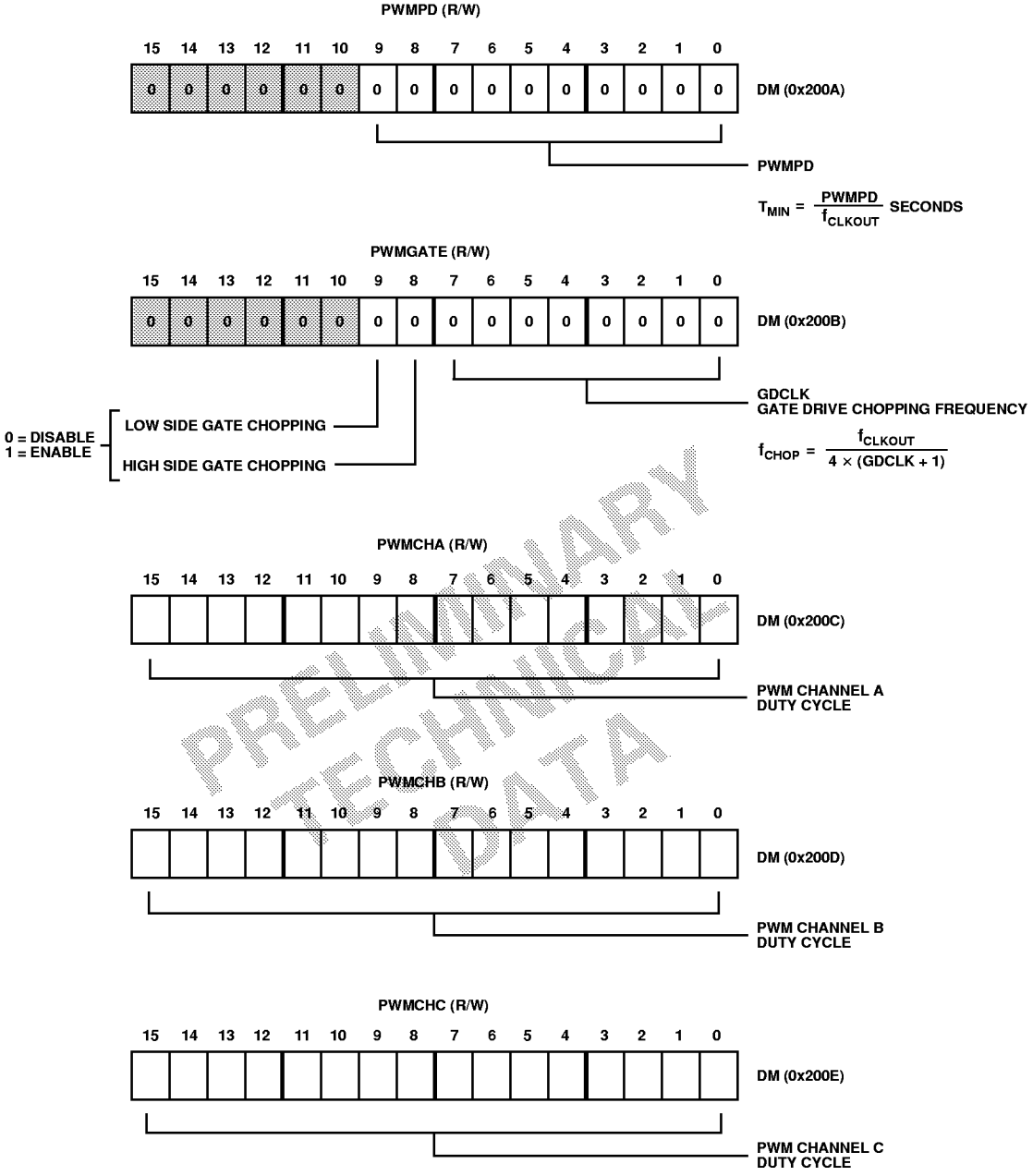


Figure 20. Configuration of Additional PWM Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field—these bits should always be written as shown.

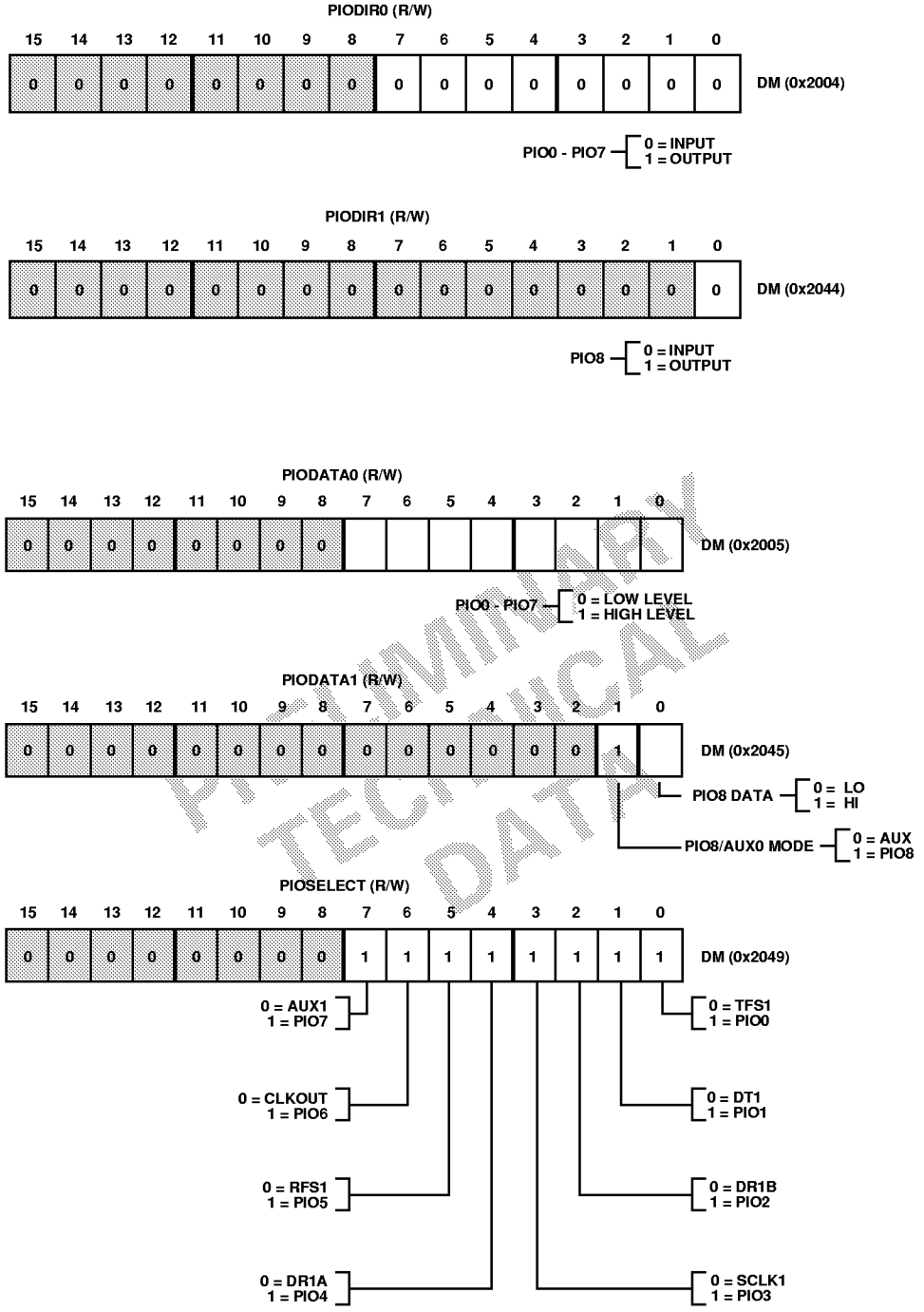


Figure 21. Configuration of PIO Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field—these bits should always be written as shown.

Figure 22. Configuration of Additional PIO Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field—these bits should always be written as shown.

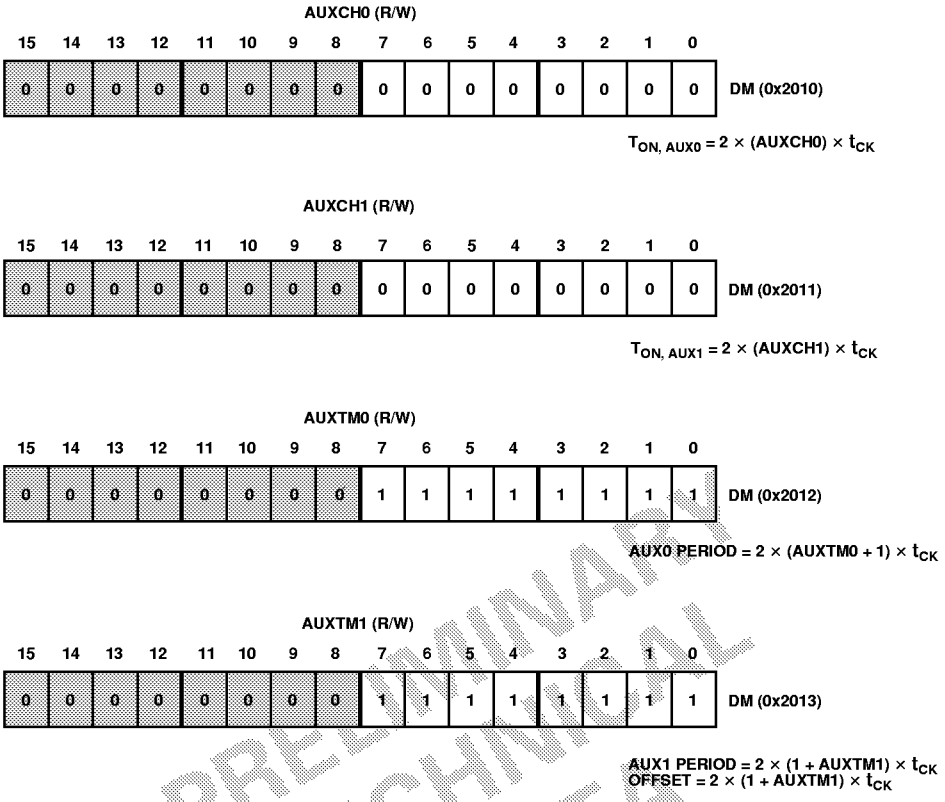


Figure 23. Configuration of AUX Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field—these bits should always be written as shown.

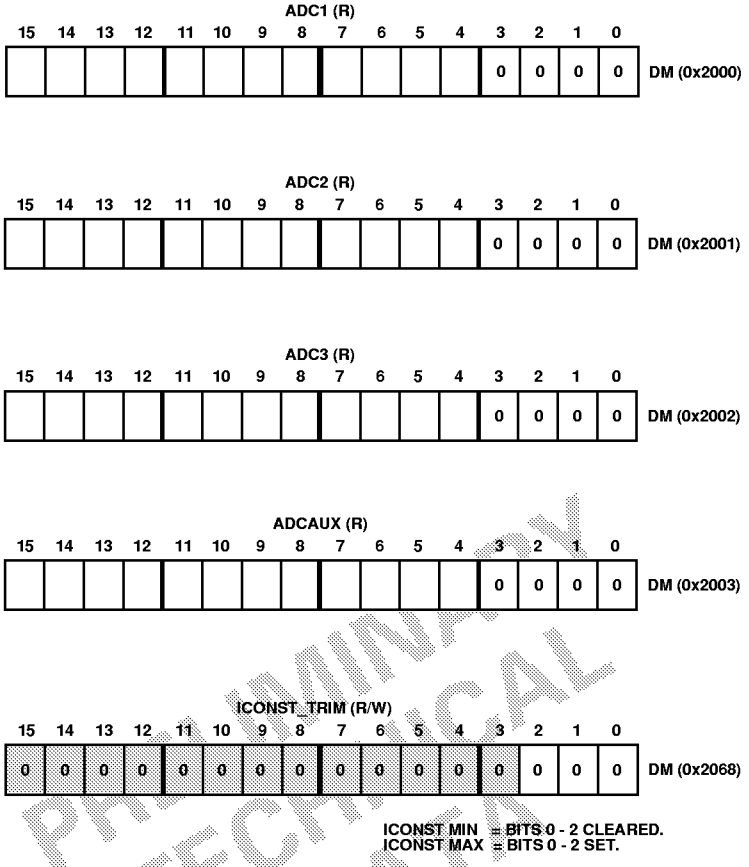


Figure 24. Configuration of Additional AUX Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field—these bits should always be written as shown.

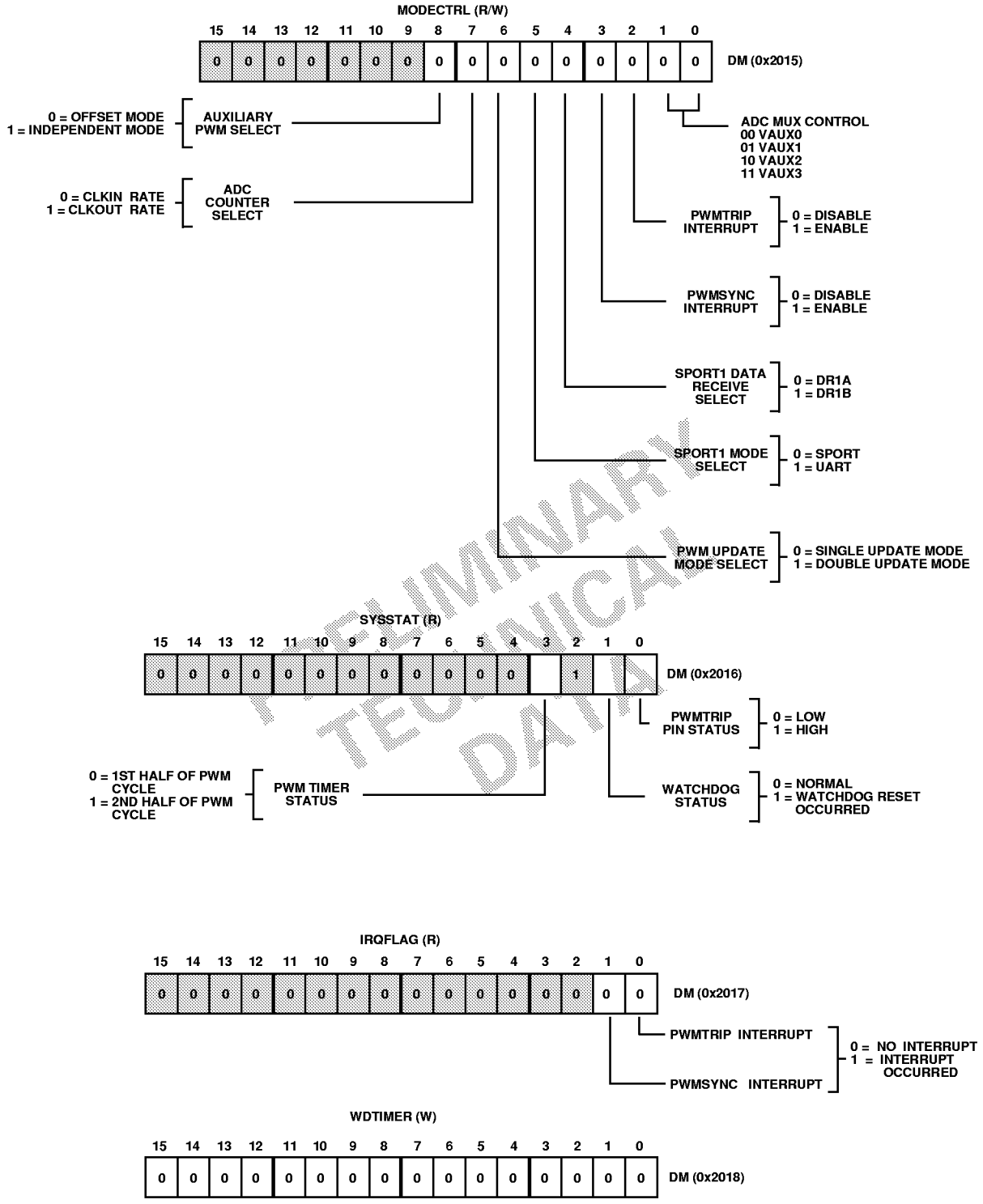


Figure 25. Configuration of Status Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field—these bits should always be written as shown.

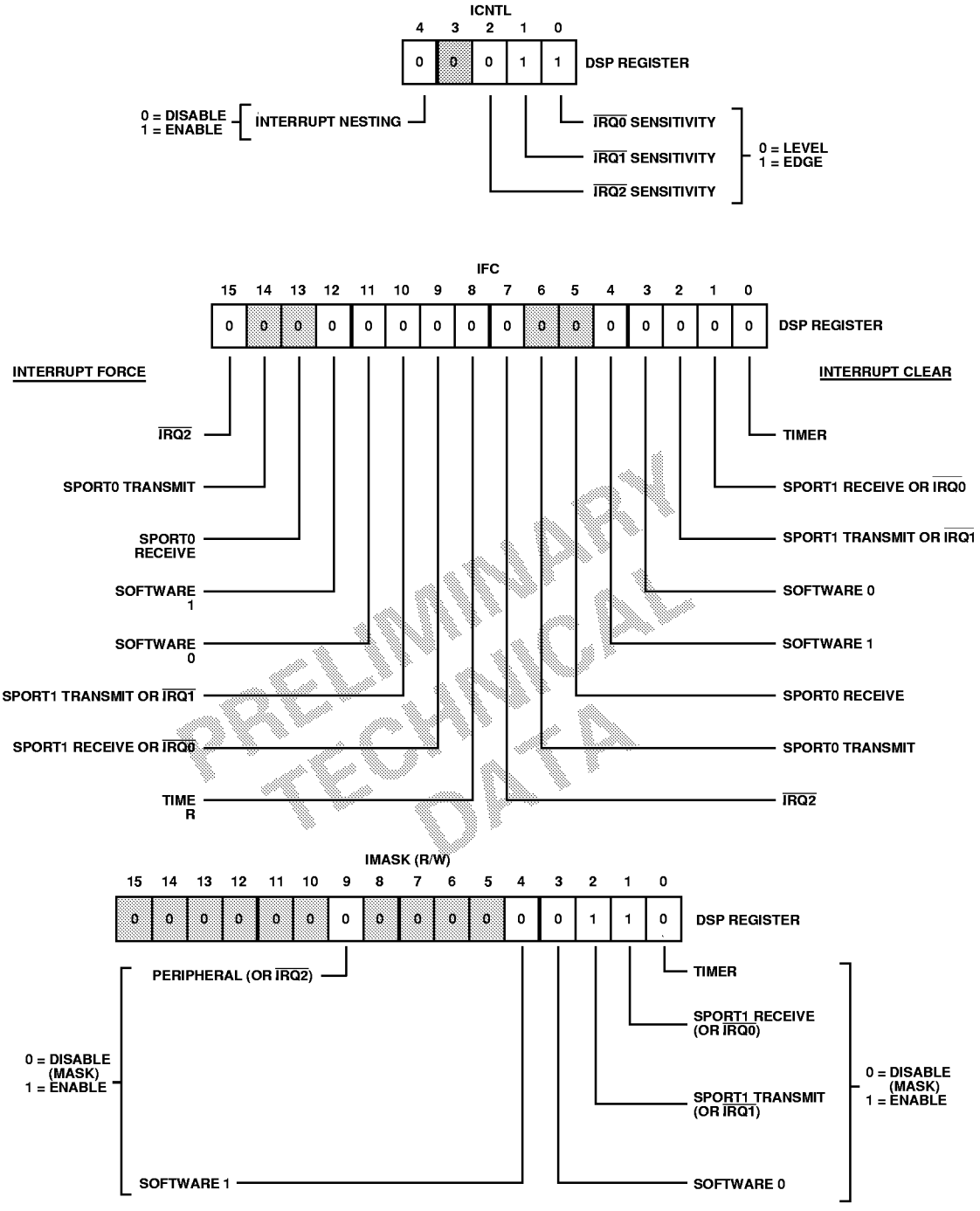


Figure 26. Configuration of Interrupt Control Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field—these bits should always be written as shown.

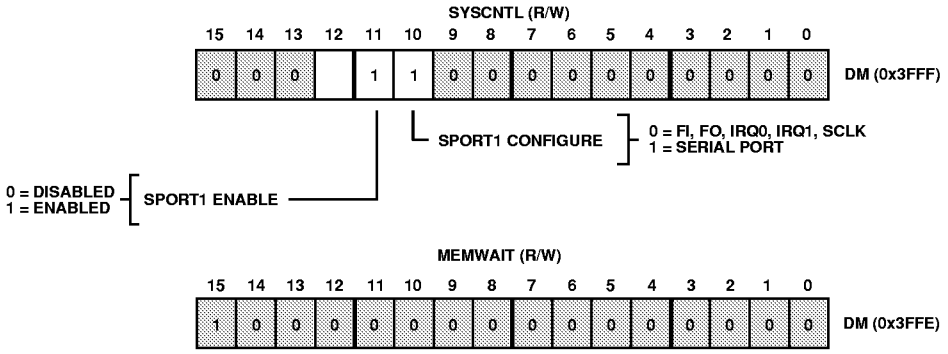


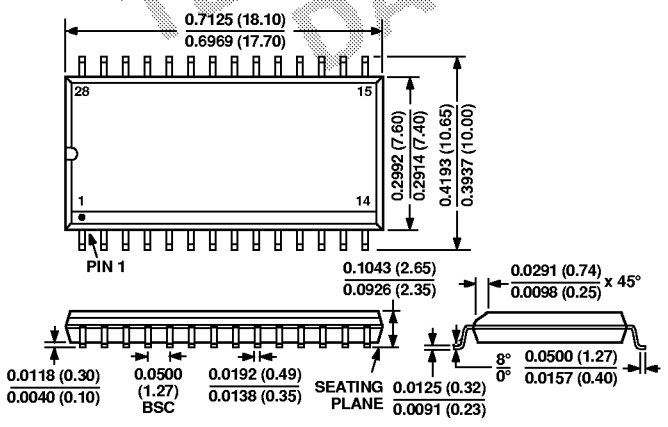
Figure 27. Configuration of Registers

Default bit values are shown; if no value is shown, the bit field is undefined at reset. Reserved bits are shown on a gray field—these bits should always be written as shown.

OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

28-Lead Wide-Body SOIC
(R-28)



C3429–8–10/98

PRINTED IN U.S.A.