MDS113CG

13-Port 10/100/1000 Mbps Ethernet Distributed Switch

DS5440 ISSUE 1.0 February 2001

1 Features

- 12 10/100 Mbps Autosensing, Fast Ethernet ports with reduced MII interface and a single Gigabit Ethernet port with GMII or TBI interface
- Full wire speed Layer 2 Switching
- Internal Switch Database Memory supports up to 2k MAC addresses, up to 16K MAC addresses, using external memory
- Automatic Source learning and age-out
- Port Trunking and Load Sharing for high bandwidth links between switches
- Full duplex Ethernet IEEE 802.3x flow control
- Supports back-pressure flow control for half-duplex mode
- Flooding and Broadcasting control
- Parallel Flash interface in fast self-initialization
- Port Mirroring Support
  12 10/100 Mbps Autosensing, Fast Ethernet ports with Reduced MII Interface

Ordering Information

- Single Gigabit Ethernet port
- Supports both GMII and integrated Physical Coding Sublayer (TBI) logic to interface directly with Gigabit transceivers
- Two-chip solution for 24+2 configuration
- 32-bit wide bi-directional pipe at 100Mhz provides 6.4Gbps pipe to connect two DS113 chips
- Supports up to 6.548 Mpps system throughput using non-blocking architecture
- High performance Layer 2 packet forwarding and filtering at full wire speed.

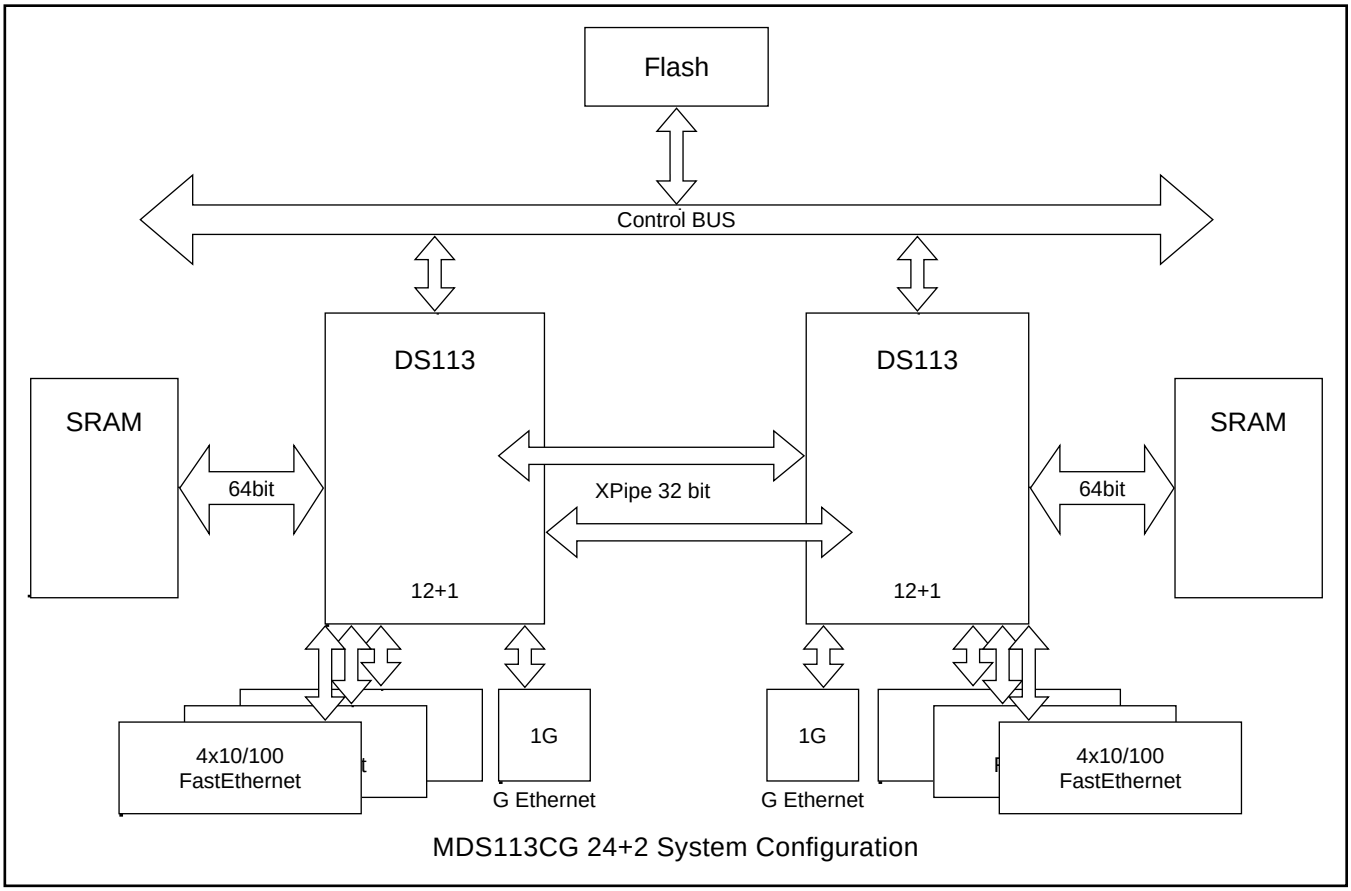


Figure 1 - MDS113CG System Diagram

- Port Trunking and Load Sharing for high bandwidth links between switches
  Very low latency through single store and forward at ingress port and cut-through switching at destination ports
- On-chip address lookup engine and memory for up to 2K MAC addresses
- Up to 16K using external memory via HISC
- Parallel Flash interface for fast self initialization
- Full duplex Ethernet IEEE 803.2x flow control minimizes traffic congestion
- Supports back-pressure flow control for half duplex mode
- Flooding and Broadcasting control
- Link status and TX/RX activity through serial LED interface
- Port Mirroring
- Packaged in 456-Pin Ball Grid Array

2 Description

The Zarlink MDS113CG is a 13-port 10/100/1000 Mbps high-performance, non-blocking Ethernet switch with on-chip address memory and address lookup engine. A single chip provides 12 - 10/100 Mbps ports and 1 - 1000 Mbps port. The MDS113CG is utilized in unmanaged switching applications.

The 3.2 Gbps XPipe allows a high-speed connection between two MDS113CG chips, providing an optimal, low-cost, workgroup switch with 24 10/100 Fast Ethernet ports and 2 Gigabit Ethernet ports.

In half-duplex mode, all ports support backpressure flow control to minimize the risk of losing data for long activity bursts. In full-duplex mode, IEEE 802.3x frame based flow control is used. With full-duplex capabilities, the Fast Ethernet ports supports 200 Mbps aggregate bandwidth connections, while the Gigabit Ethernet port supports 2 Gbps to desktops, servers, or other high-performance switches. The Physical Coding Sublayer (TBI) is integrated on-chip to provide a direct 10-bit Gigabit Media Independent Interface (GMII). The on-chip TBI may be bypassed to provide a Ten Bit Interface (TBI) to an existing fiber-based Gigabit Transceiver.

The MDS113CG supports port trunking/load sharing on the 10/100 Mbps ports. Port trunking/load sharing can be used to group ports between inter-linked switches for increased system bandwidth. Ports within a trunk must reside within a single MDS113CG, such that trunks may not be configured across two switches.

The on-chip address lookup engine supports up to 2K MAC addresses and up to 16K MAC address using the external memory.

The MDS113CG utilizes cost effective, high performance, pipelined synchronous burst RAM to achieve full wire speed on all ports simultaneously. Data is buffered into memory, using 0-128 byte bursts, from the ingress ports, and transferred to an internal transmit FIFO, before being sent from the frame memory to the egress output ports. Extremely high memory bandwidth is therefore achieved, which allows each of the ports to be active without creating a memory bottleneck.

The MDS113CG is fabricated with 2.5 V technology, where the inputs are 3.3V tolerant and the outputs are capable of directly interfacing to Low-Voltage TTL levels. The Zarlink Semiconductor MDS113CG is packaged in a 456-pin Ball Grid Array.

3 MDS113CG Block Diagram

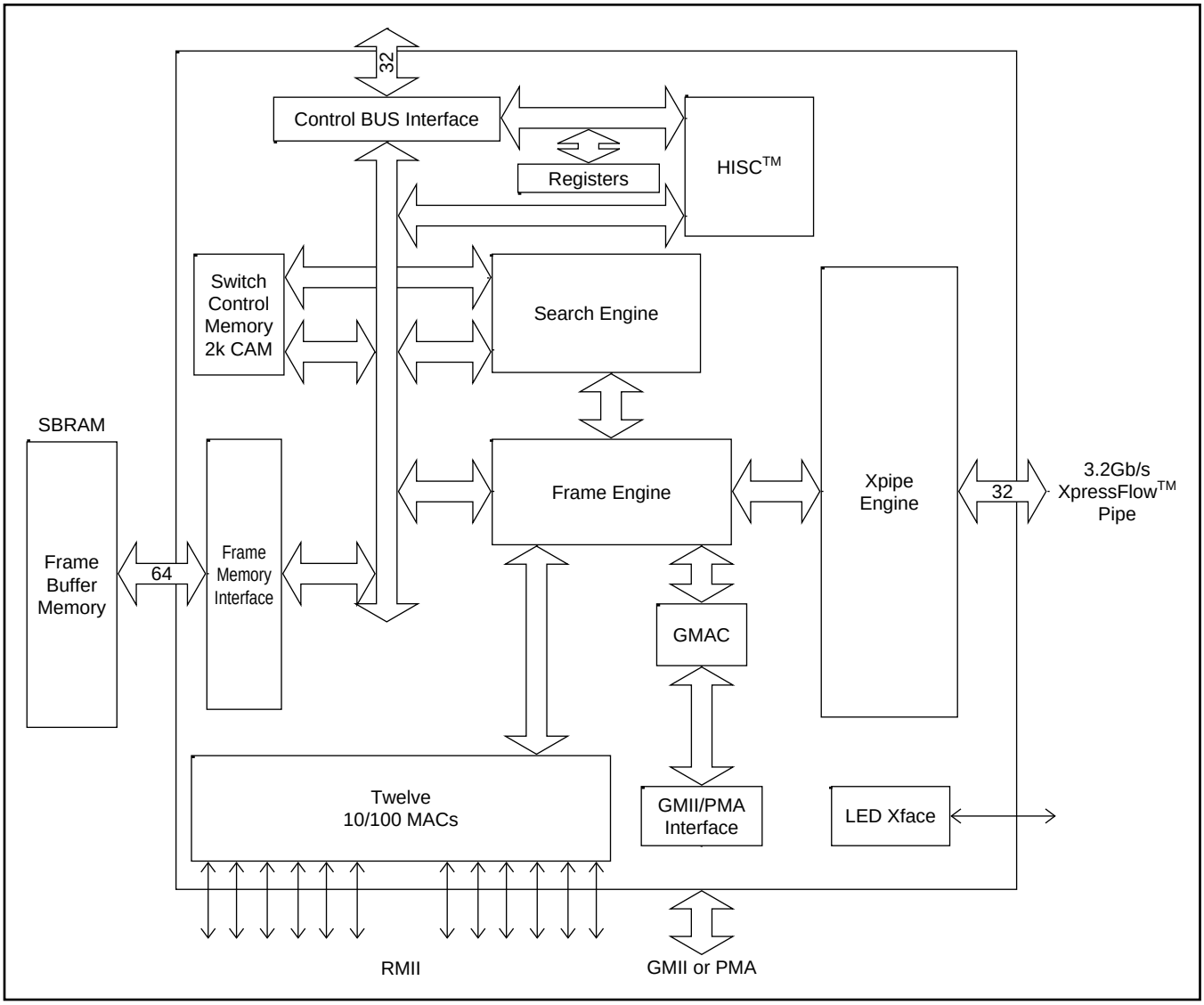
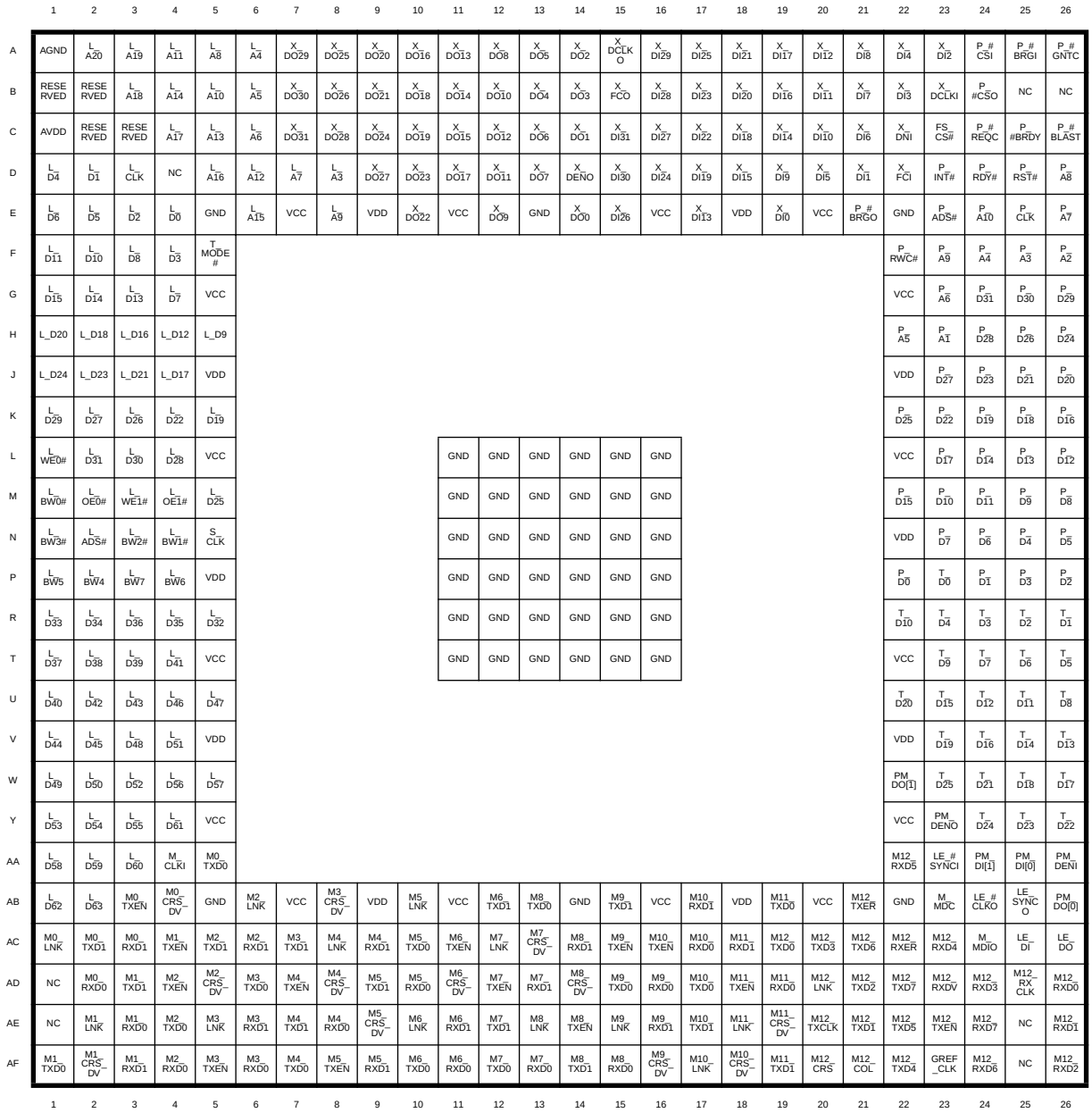


Figure 2 - System Block Diagram

Notes: All registers are 32-bit width
The Control Bus is 32-bits wide and the Memory Bus is 64-bits wide
The MDS113CG contains 12 Fast Ethernet Ports and 1 Gigabit Ethernet Port
The LED interface has 3 output signals (1 data and 2 control)
The XPipe is 32-bits wide

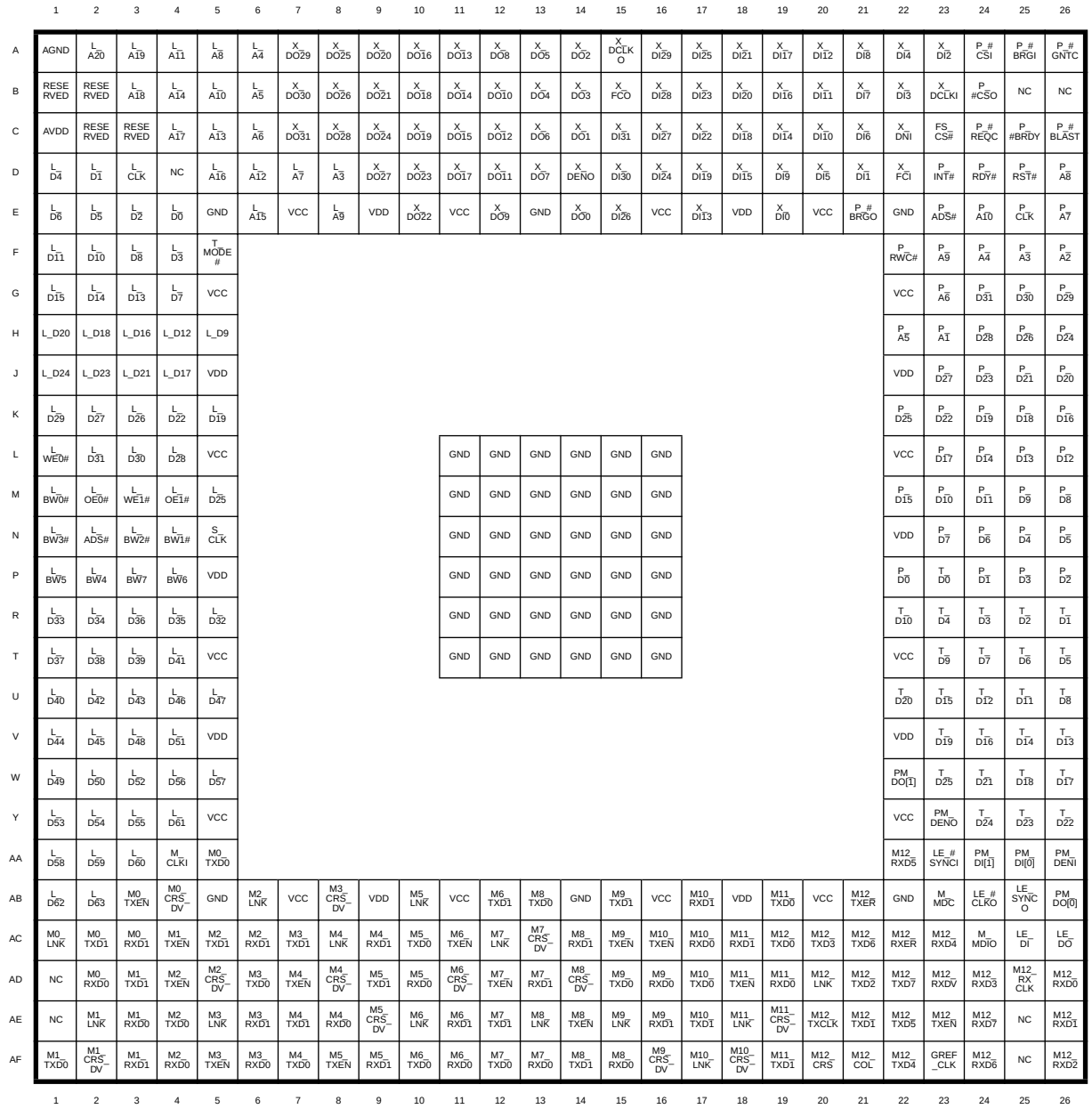
4 Ball - Signal Description and Assignments



A	AGND	L_A20	L_A19	L_A11	L_A8	L_A4	X_DO29	X_DO25	X_DO20	X_DO16	X_DO13	X_DO8	X_DO5	X_DO2	X_DCLKO	X_DI29	X_DI25	X_DI21	X_DI17	X_DI12	X_DI8	X_DI4	X_DI2	P_#CSI	P_#BRGI	P_#GNTC
B	RESERVED	RESERVED	L_A18	L_A14	L_A10	L_A5	X_DO30	X_DO26	X_DO21	X_DO18	X_DO14	X_DO10	X_DO4	X_DO3	X_FCO	X_DI28	X_DI23	X_DI20	X_DI16	X_DI11	X_DI7	X_DI3	X_DCLKI	P_#CSO	NC	NC
C	AVDD	RESERVED	RESERVED	L_A17	L_A13	L_A6	X_DO31	X_DO28	X_DO24	X_DO19	X_DO15	X_DO12	X_DO6	X_DO1	X_DI31	X_DI27	X_DI22	X_DI18	X_DI14	X_DI10	X_DI6	X_DNI	FS_CS#	P_#REQC	P_#BRDY	P_#BLAST
D	L_D4	L_D1	L_CLK	NC	L_A16	L_A12	L_A7	L_A3	X_DO27	X_DO23	X_DO17	X_DO11	X_DO7	X_DENO	X_DI30	X_DI24	X_DI19	X_DI15	X_DI9	X_DI5	X_DI1	X_FCI	P_INT#	P_RDY#	P_RST#	P_A8
E	L_D6	L_D5	L_D2	L_D0	GND	L_A15	VCC	L_A9	VDD	X_DO22	VCC	X_DO9	GND	X_DO0	X_DI26	VCC	X_DI13	VDD	X_DI0	VCC	P_#BRGO	GND	P_ADS#	P_A10	P_CLK	P_A7
F	L_D11	L_D10	L_D8	L_D3	T_MODE#																	P_RWC#	P_A9	P_A4	P_A3	P_A2
G	L_D15	L_D14	L_D13	L_D7	VCC																	VCC	P_A6	P_D31	P_D30	P_D29
H	L_D20	L_D18	L_D16	L_D12	L_D9																	P_A5	P_A1	P_D28	P_D26	P_D24
J	L_D24	L_D23	L_D21	L_D17	VDD																	VDD	P_D27	P_D23	P_D21	P_D20
K	L_D29	L_D27	L_D26	L_D22	L_D19																	P_D25	P_D22	P_D19	P_D18	P_D16
L	L_WE0#	L_D31	L_D30	L_D28	VCC						GND	GND	GND	GND	GND	GND						VCC	P_D17	P_D14	P_D13	P_D12
M	L_BW0#	L_OE0#	L_WE1#	L_OE1#	L_D25						GND	GND	GND	GND	GND	GND						P_D15	P_D10	P_D11	P_D9	P_D8
N	L_BW3#	L_ADS#	L_BW2#	L_BW1#	S_CLK						GND	GND	GND	GND	GND	GND						VCC	P_D7	P_D6	P_D4	P_D5
P	L_BW5	L_BW4	L_BW7	L_BW6	VDD						GND	GND	GND	GND	GND	GND						P_D0	T_D0	P_D1	P_D3	P_D2
R	L_D33	L_D34	L_D36	L_D35	L_D32						GND	GND	GND	GND	GND	GND						T_D10	T_D4	T_D3	T_D2	T_D1
T	L_D37	L_D38	L_D39	L_D41	VCC						GND	GND	GND	GND	GND	GND						VCC	T_D9	T_D7	T_D6	T_D5
U	L_D40	L_D42	L_D43	L_D46	L_D47																	T_D20	T_D15	T_D12	T_D11	T_D8
V	L_D44	L_D45	L_D48	L_D51	VDD																	VDD	T_D19	T_D16	T_D14	T_D13
W	L_D49	L_D50	L_D52	L_D56	L_D57																	PM_DO[1]	T_D25	T_D21	T_D18	T_D17
Y	L_D53	L_D54	L_D55	L_D61	VCC																	VCC	PM_DENO	T_D24	T_D23	T_D22
AA	L_D58	L_D59	L_D60	M_CLKI	M0_TXD0																	M12_RXD5	LE_#SYNCI	PM_DI[1]	PM_DI[0]	PM_DENI
AB	L_D62	L_D63	M0_TXEN	M0_CRS_DV	GND	M2_LNK	VCC	M3_CRS_DV	VDD	M5_LNK	VCC	M6_TXD1	M8_TXD0	GND	M9_TXD1	VCC	M10_RXD1	VDD	M11_TXD0	VCC	M12_TXER	GND	M_MDC	LE_#CLKO	LE_SYNCO	PM_DO[0]
AC	M0_LNK	M0_TXD1	M0_RXD1	M1_TXEN	M2_TXD1	M2_RXD1	M3_TXD1	M4_LNK	M4_RXD1	M5_TXD0	M6_TXEN	M7_LNK	M7_CRS_DV	M8_RXD1	M9_TXEN	M10_TXEN	M10_RXD0	M11_RXD1	M12_TXD0	M12_TXD3	M12_TXD6	M12_RXER	M12_RXD4	M_MDIO	LE_DI	LE_DO
AD	NC	M0_RXD0	M1_TXD1	M2_TXEN	M2_CRS_DV	M3_TXD0	M4_TXEN	M4_CRS_DV	M5_TXD1	M5_RXD0	M6_CRS_DV	M7_TXEN	M7_RXD1	M8_CRS_DV	M9_TXD0	M9_RXD0	M10_TXD0	M11_TXEN	M11_RXD0	M12_LNK	M12_TXD2	M12_TXD7	M12_RXDV	M12_RXD3	M12_RX_CLK	M12_RXD0
AE	NC	M1_LNK	M1_RXD0	M2_TXD0	M3_LNK	M3_RXD1	M4_TXD1	M4_RXD0	M5_CRS_DV	M6_LNK	M6_RXD1	M7_TXD1	M8_LNK	M8_TXEN	M9_LNK	M9_RXD1	M10_TXD1	M11_LNK	M11_CRS_DV	M12_TXCLK	M12_TXD1	M12_TXD5	M12_TXEN	M12_RXD7	NC	M12_RXD1
AF	M1_TXD0	M1_CRS_DV	M1_RXD1	M2_RXD0	M3_TXEN	M3_RXD0	M4_TXD0	M5_TXEN	M5_RXD1	M6_TXD0	M6_RXD0	M7_TXD0	M7_RXD0	M8_TXD1	M8_RXD0	M9_CRS_DV	M10_LNK	M10_CRS_DV	M11_TXD1	M12_CRS	M12_COL	M12_TXD4	GREF_CLK	M12_RXD6	NC	M12_RXD2

VCC	=	3.3VDC for I/O	(16 balls)
VDD	=	2.5VDC for core logic	(10 balls)
GND	=	Digital Ground for both VCC and VDD	(42 balls)
AVDD	=	2.5VDC for Analog PLL	(1 ball)
AGND	=	Isolated Analog Ground for AVDD	(1 ball)
NC	=	No connection	
Reserved	=	DO NOT CONNECT	

Power and Ground Distribution



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	AGND	L_A20	L_A19	L_A11	L_A8	L_A4	X_DO29	X_DO25	X_DO20	X_DO16	X_DO13	X_DO8	X_DO5	X_DO2	X_DCLKO	X_DI29	X_DI25	X_DI21	X_DI17	X_DI12	X_DI8	X_DI4	X_DI2	P_#CSI	P_#BRGI	P_#GNTC
B	RESERVED	RESERVED	L_A18	L_A14	L_A10	L_A5	X_DO30	X_DO26	X_DO21	X_DO18	X_DO14	X_DO10	X_DO4	X_DO3	X_FCO	X_DI28	X_DI23	X_DI20	X_DI16	X_DI11	X_DI7	X_DI3	X_DCLKI	P_#CSO	NC	NC
C	AVDD	RESERVED	RESERVED	L_A17	L_A13	L_A6	X_DO31	X_DO28	X_DO24	X_DO19	X_DO15	X_DO12	X_DO6	X_DO1	X_DI31	X_DI27	X_DI22	X_DI18	X_DI14	X_DI10	X_DI6	X_DNI	FS_CS#	P_#REQC	P_#BRDY	P_#BLAST
D	L_D4	L_D1	L_CLK	NC	L_A16	L_A12	L_A7	L_A3	X_DO27	X_DO23	X_DO17	X_DO11	X_DO7	X_DENO	X_DI30	X_DI24	X_DI19	X_DI15	X_DI9	X_DI5	X_DI1	X_FCI	P_INT#	P_RDY#	P_RST#	P_A8
E	L_D6	L_D5	L_D2	L_D0	GND	L_A15	VCC	L_A9	VDD	X_DO22	VCC	X_DO9	GND	X_DO0	X_DI26	VCC	X_DI13	VDD	X_DI0	VCC	P_#BRGO	GND	P_ADS#	P_A10	P_CLK	P_A7
F	L_D11	L_D10	L_D8	L_D3	T_MODE#																	P_RWC#	P_A9	P_A4	P_A3	P_A2
G	L_D15	L_D14	L_D13	L_D7	VCC																	VCC	P_A6	P_D31	P_D30	P_D29
H	L_D20	L_D18	L_D16	L_D12	L_D9																	P_A5	P_A1	P_D28	P_D26	P_D24
J	L_D24	L_D23	L_D21	L_D17	VDD																	VDD	P_D27	P_D23	P_D21	P_D20
K	L_D29	L_D27	L_D26	L_D22	L_D19																	P_D25	P_D22	P_D19	P_D18	P_D16
L	L_WE0#	L_D31	L_D30	L_D28	VCC						GND	GND	GND	GND	GND	GND						VCC	P_D17	P_D14	P_D13	P_D12
M	L_BW0#	L_OE0#	L_WE1#	L_OE1#	L_D25						GND	GND	GND	GND	GND	GND						P_D15	P_D10	P_D11	P_D9	P_D8
N	L_BW3#	L_ADS#	L_BW2#	L_BW1#	S_CLK						GND	GND	GND	GND	GND	GND						VDD	P_D7	P_D6	P_D4	P_D5
P	L_BW5	L_BW4	L_BW7	L_BW6	VDD						GND	GND	GND	GND	GND	GND						P_D0	T_D0	P_D1	P_D3	P_D2
R	L_D33	L_D34	L_D36	L_D35	L_D32						GND	GND	GND	GND	GND	GND						T_D10	T_D4	T_D3	T_D2	T_D1
T	L_D37	L_D38	L_D39	L_D41	VCC						GND	GND	GND	GND	GND	GND						VCC	T_D9	T_D7	T_D6	T_D5
U	L_D40	L_D42	L_D43	L_D46	L_D47																	T_D20	T_D15	T_D12	T_D11	T_D8
V	L_D44	L_D45	L_D48	L_D51	VDD																	VDD	T_D19	T_D16	T_D14	T_D13
W	L_D49	L_D50	L_D52	L_D56	L_D57																	PM_DO[1]	T_D25	T_D21	T_D18	T_D17
Y	L_D53	L_D54	L_D55	L_D61	VCC																	VCC	PM_DENO	T_D24	T_D23	T_D22
AA	L_D58	L_D59	L_D60	M_CLKI	M0_TXD0																	M12_RXD5	LE_#SYNCI	PM_DI[1]	PM_DI[0]	PM_DENI
AB	L_D62	L_D63	M0_TXEN	M0_CRS_DV	GND	M2_LNK	VCC	M3_CRS_DV	VDD	M5_LNK	VCC	M6_TXD1	M8_TXD0	GND	M9_TXD1	VCC	M10_RXD1	VDD	M11_TXD0	VCC	M12_TXER	GND	M_MDC	LE_#CLKO	LE_SYNCO	PM_DO[0]
AC	M0_LNK	M0_TXD1	M0_RXD1	M1_TXEN	M2_TXD1	M2_RXD1	M3_TXD1	M4_LNK	M4_RXD1	M5_TXD0	M6_TXEN	M7_LNK	M7_CRS_DV	M8_RXD1	M9_TXEN	M10_TXEN	M10_RXD0	M11_RXD1	M12_TXD0	M12_TXD3	M12_TXD6	M12_RXER	M12_RXD4	M_MDIO	LE_DI	LE_DO
AD	NC	M0_RXD0	M1_TXD1	M2_TXEN	M2_CRS_DV	M3_TXD0	M4_TXEN	M4_CRS_DV	M5_TXD1	M5_RXD0	M6_CRS_DV	M7_TXEN	M7_RXD1	M8_CRS_DV	M9_TXD0	M9_RXD0	M10_TXD0	M11_TXEN	M11_RXD0	M12_LNK	M12_TXD2	M12_TXD7	M12_RXDV	M12_RXD3	M12_RX_CLK	M12_RXD0
AE	NC	M1_LNK	M1_RXD0	M2_TXD0	M3_LNK	M3_RXD1	M4_TXD1	M4_RXD0	M5_CRS_DV	M6_LNK	M6_RXD1	M7_TXD1	M8_LNK	M8_TXEN	M9_LNK	M9_RXD1	M10_TXD1	M11_LNK	M11_CRS_DV	M12_TXCLK	M12_TXD1	M12_TXD5	M12_TXEN	M12_RXD7	NC	M12_RXD1
AF	M1_TXD0	M1_CRS_DV	M1_RXD1	M2_RXD0	M3_TXEN	M3_RXD0	M4_TXD0	M5_TXEN	M5_RXD1	M6_TXD0	M6_RXD0	M7_TXD0	M7_RXD0	M8_TXD1	M8_RXD0	M9_CRS_DV	M10_LNK	M10_CRS_DV	M11_TXD1	M12_CRS	M12_COL	M12_TXD4	GREF_CLK	M12_RXD6	NC	M12_RXD2

VCC	=	3.3VDC for I/O	(16 balls)
VDD	=	2.5VDC for core logic	(10 balls)
GND	=	Digital Ground for both VCC and VDD	(42 balls)
AVDD	=	2.5VDC for Analog PLL	(1 ball)
AGND	=	Isolated Analog Ground for AVDD	(1 ball)
NC	=	No connection	
Reserved	=	DO NOT CONNECT	

Ball - Signal Assignments

Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name
D4	NC	R2	L_D34	AC7	M3_TXD1	AF19	M11_TXD1
C3	RESERVED	R4	L_D35	AD6	M3_TXD0	AB19	M11_TXD0
A1	AGND	R3	L_D36	AB8	M3_CRS_DV	AE19	M11_CRS_DV
B1	RESERVED	T1	L_D37	AE6	M3_RXD1	AC18	M11_RXD1
C1	AVDD	T2	L_D38	AF6	M3_RXD0	AD19	M11_RXD0
F5	T_MODE#	T3	L_D39	AC8	M4_LNK	AF20	M12_CRS
C2	RESERVED	U1	L_D40	AD7	M4_TXEN	AE20	M12_TXCLK / GP_TXCLK
D3	L_CLK	T4	L_D41	AE7	M4_TXD1	AD20	M12_LNK / GP_LNK
E4	L_D0	U2	L_D42	AF7	M4_TXD0	AC19	M12_TXD0 / GP_TXD9
D2	L_D1	U3	L_D43	AD8	M4_CRS_DV	AF21	M12_COL / GP_RXCLK1
E3	L_D2	V1	L_D44	AC9	M4_RXD1	AE21	M12_TXD1 / GP_TXD8
F4	L_D3	V2	L_D45	AE8	M4_RXD0	AD21	M12_TXD2 / GP_TXD7
D1	L_D4	U4	L_D46	AB10	M5_LNK	AC20	M12_TXD3 / GP_TXD6
E2	L_D5	U5	L_D47	AF8	M5_TXEN	AF22	M12_TXD4 / GP_TXD5
E1	L_D6	V3	L_D48	AD9	M5_TXD1	AE22	M12_TXD5 / GP_TXD4
G4	L_D7	W1	L_D49	AC10	M5_TXD0	AF23	GREF_CLK
F3	L_D8	W2	L_D50	AE9	M5_CRS_DV	AC21	M12_TXD6 / GP_TXD3
H5	L_D9	V4	L_D51	AF9	M5_RXD1	AD22	M12_TXD7 / GP_TXD2
F2	L_D10	W3	L_D52	AD10	M5_RXD0	AE23	M12_TX_EN / GP_TXD1
F1	L_D11	Y1	L_D53	AE10	M6_LNK	AB21	M12_TX_ER / GP_TXD0
H4	L_D12	Y2	L_D54	AC11	M6_TXEN	AC22	M12_RX_ER / GP_RXD0
G3	L_D13	Y3	L_D55	AB12	M6_TXD1	AD23	M12_RX_DV / GP_RXD1
G2	L_D14	W4	L_D56	AF10	M6_TXD0	AE24	M12_RXD7 / GP_RXD2
G1	L_D15	W5	L_D57	AD11	M6_CRS_DV	AF24	M12_RXD6 / GP_RXD3
H3	L_D16	AA1	L_D58	AE11	M6_RXD1	AF25	NC
J4	L_D17	AA2	L_D59	AF11	M6_RXD0	AE25	NC
H2	L_D18	AA3	L_D60	AC12	M7_LNK	AA22	M12_RXD5 / GP_RXD4
K5	L_D19	Y4	L_D61	AD12	M7_TXEN	AC23	M12_RXD4 / GP_RXD5
H1	L_D20	AB1	L_D62	AE12	M7_TXD1	AD24	M12_RXD3 / GP_RXD6
J3	L_D21	AB2	L_D63	AF12	M7_TXD0	AF26	M12_RXD2 / GP_RXD7
K4	L_D22	AC1	M0_LNK	AC13	M7_CRS_DV	AE26	M12_RXD1 / GP_RXD8
J2	L_D23	AA4	M_CLKI	AD13	M7_RXD1	AD26	M12_RXD0 / GP_RXD9
J1	L_D24	AB3	M0_TXEN	AF13	M7_RXD0	AD25	M12_RXCLK / GP_RXCLK0
M5	L_D25	AC2	M0_TXD1	AE13	M8_LNK	AC24	M_MDIO
K3	L_D26	AA5	M0_TXD0	AE14	M8_TXEN	AB23	M_MDC
K2	L_D27	AB4	M0_CRS_DV	AF14	M8_TXD1	AC25	LE_DI
L4	L_D28	AC3	M0_RXD1	AB13	M8_TXD0	AB24	LE_CLKO
K1	L_D29	AD2	M0_RXD0	AD14	M8_CRS_DV	AA23	LE_SYNCI
L3	L_D30	AD1	NC	AC14	M8_RXD1	AC26	LE_DO
L2	L_D31	AE1	NC	AF15	M8_RXD0	AB25	LE_SYNCO
L1	L_WE0#	AE2	M1_LNK	AE15	M9_LNK	W22	T_D31/PM_DO1
M2	L_OE0#	AC4	M1_TXEN	AC15	M9_TXEN	AB26	T_D30/PM_DO0
M3	L_WE1#	AD3	M1_TXD1	AB15	M9_TXD1	Y23	T_D29/PM_DENO
M4	L_OE1#	AF1	M1_TXD0	AD15	M9_TXD0	AA24	T_D28/PM_DI1
M1	L_BW0#	AF2	M1_CRS_DV	AF16	M9_CRS_DV	AA25	T_D27/PM_DI0
N4	L_BW1#	AF3	M1_RXD1	AE16	M9_RXD1	AA26	T_D26/PM_DENI
N5	S_CLK	AE3	M1_RXD0	AD16	M9_RXD0	W23	T_D25
N3	L_BW2#	AB6	M2_LNK	AF17	M10_LNK	Y24	T_D24
N1	L_BW3#	AD4	M2_TXEN	AC16	M10_TXEN	Y25	T_D23
N2	L_ADS#	AC5	M2_TXD1	AE17	M10_TXD1	Y26	T_D22
P2	L_BW4#	AE4	M2_TXD0	AD17	M10_TXD0	W24	T_D21
P1	L_BW5#	AD5	M2_CRS_DV	AF18	M10_CRS_DV	U22	T_D20
P4	L_BW6#	AC6	M2_RXD1	AB17	M10_RXD1	V23	T_D19
P3	L_BW7#	AF4	M2_RXD0	AC17	M10_RXD0	W25	T_D18
R5	L_D32	AE5	M3_LNK	AE18	M11_LNK	W26	T_D17
R1	L_D33	AF5	M3_TXEN	AD18	M11_TXEN	V24	T_D16

Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name
U23	T_D15	E24	P_A10 / FS_A10	B14	X_DO3	Y5	VCC
V25	T_D14	D25	P_RST#	B13	X_DO4	Y22	VCC
V26	T_D13	F22	P_RWC#	A13	X_DO5	AB7	VCC
U24	T_D12	E23	P_ADS#	C13	X_DO6	AB11	VCC
U25	T_D11	D24	P_RDY#	D13	X_DO7	AB16	VCC
R22	T_D10	C25	P_BRDY#	A12	X_DO8	AB20	VCC
T23	T_D9	C26	P_BLAST#	E12	X_DO9	E9	VDD
U26	T_D8	B26	NC	B12	X_DO10	E18	VDD
T24	T_D7	B25	NC	D12	X_DO11	J5	VDD
T25	T_D6	D23	P_INT	C12	X_DO12	J22	VDD
T26	T_D5	C24	P_REQC#	A11	X_DO13	N22	VDD
R23	T_D4/BS_RDYOP	A26	P_GNTC#	B11	X_DO14	P5	VDD
R24	T_D3/BS_PSD	A25	P_BRQI#	C11	X_DO15	V5	VDD
R25	T_D2/BS_SWM	E21	P_BRQO#	A10	X_DO16	V22	VDD
R26	T_D1/BS_RW	A24	P_CSI#	D11	X_DO17	AB9	VDD
P23	T_D0/BS_BMOD	B24	P_CSO#	B10	X_DO18	AB18	VDD
P22	P_D0 / FS_D0	C23	FLS_CS#	C10	X_DO19	E5	GND
P24	P_D1 / FS_D1	D22	X_FCI	A9	X_DO20	E13	GND
P26	P_D2 / FS_D2	B23	X_DCLKI	B9	X_DO21	E22	GND
P25	P_D3 / FS_D3	C22	X_DNI	E10	X_DO22	L11	GND
N25	P_D4 / FS_D4	E19	X_DI0	D10	X_DO23	L12	GND
N26	P_D5 / FS_D5	D21	X_DI1	C9	X_DO24	L13	GND
N24	P_D6 / FS_D6	A23	X_DI2	A8	X_DO25	L14	GND
N23	P_D7 / FS_D7	B22	X_DI3	B8	X_DO26	L15	GND
M26	P_D8 / FS_D8	A22	X_DI4	D9	X_DO27	L16	GND
M25	P_D9 / FS_D9	D20	X_DI5	C8	X_DO28	M11	GND
M23	P_D10 / FS_D10	C21	X_DI6	A7	X_DO29	M12	GND
M24	P_D11 / FS_D11	B21	X_DI7	B7	X_DO30	M13	GND
L26	P_D12 / FS_D12	A21	X_DI8	C7	X_DO31	M14	GND
L25	P_D13 / FS_D13	D19	X_DI9	D8	L_A3	M15	GND
L24	P_D14 / FS_D14	C20	X_DI10	A6	L_A4	M16	GND
M22	P_D15 / FS_D15	B20	X_DI11	B6	L_A5	N11	GND
K26	P_D16	A20	X_DI12	C6	L_A6	N12	GND
L23	P_D17	E17	X_DI13	D7	L_A7	N13	GND
K25	P_D18	C19	X_DI14	A5	L_A8	N14	GND
K24	P_D19	D18	X_DI15	E8	L_A9	N15	GND
J26	P_D20	B19	X_DI16	B5	L_A10	N16	GND
J25	P_D21	A19	X_DI17	A4	L_A11	P11	GND
K23	P_D22	C18	X_DI18	D6	L_A12	P12	GND
J24	P_D23	D17	X_DI19	C5	L_A13	P13	GND
H26	P_D24 / FS_A11	B18	X_DI20	B4	L_A14	P14	GND
K22	P_D25 / FS_A12	A18	X_DI21	E6	L_A15	P15	GND
H25	P_D26 / FS_A13	C17	X_DI22	D5	L_A16	P16	GND
J23	P_D27 / FS_A14	B17	X_DI23	C4	L_A17	R11	GND
H24	P_D28 / FS_A15	D16	X_DI24	B3	L_A18	R12	GND
G26	P_D29 / FS_A16	A17	X_DI25	A3	L_A19	R13	GND
G25	P_D30 / FS_A17	E15	X_DI26	A2	L_A20	R14	GND
G24	P_D31 / FS_A18	C16	X_DI27	B2	RESERVED	R15	GND
H23	P_A1 / FS_A1	B16	X_DI28	E7	VCC	R16	GND
F26	P_A2 / FS_A2	A16	X_DI29	E11	VCC	T11	GND
F25	P_A3 / FS_A3	D15	X_DI30	E16	VCC	T12	GND
F24	P_A4 / FS_A4	C15	X_DI31	E20	VCC	T13	GND
H22	P_A5 / FS_A5	B15	X_FCO	G5	VCC	T14	GND
G23	P_A6 / FS_A6	A15	X_DCLKO	G22	VCC	T15	GND
E26	P_A7 / FS_A7	D14	X_DENO	L5	VCC	T16	GND
E25	P_CLK	E14	X_DO0	L22	VCC	AB5	GND
D26	P_A8 / FS_A8	C14	X_DO1	T5	VCC	AB14	GND
F23	P_A9 / FS_A9	A14	X_DO2	T22	VCC	AB22	GND

Ball - Signal Descriptions

The Type of All pins is CMOS.
All Input pins are 5-Volt Tolerance.
All Output Pins are 3.3 CMOS Drive

CONTROL BUS INTERFACE			
Ball No(s)	**Symbol**	**I/O**	**Description**
G24, G25, G26, H24, J23, H25, K22, H26, J24, k23, J25, J26, K24, K25, L23, K26, M22, L24, L25, L26, M24, M23, M25, M26, N23, N24, N26, N25, P25, P26, P24, P22	P_D[31:0]	I/O-TS, U	Control Data Bus- Data Bit[31:0]
E24, F23, D26, E26, G23, H22, F24, F25, F26, H23	P_A[10:1]	Input/ Output-U	Control Address Bus-Address Bit[14: 1]
D25	P_RST#	Input-ST	Control Bus – Master Reset
F22	P_RWC#	Input/Output-TS, U	Control Bus – Read/Write Control Programmable polarity
E23	P_ADS#	Input/Output-TS, U	Control Address Strobe
D24	P_RDY#	Output-OD-TS, U	Control Bus – Data Ready
E23	P_BRDY#	Input- TS, U	Control Bus – Burst Ready
C26	P_BLAST#	Input- TS, U	Control Bus – Burst Last
D23	P_INT	Output	Control Bus – Interrupt Request Programmable polarity
E25	P_CLK	Input	Control Bus – Bus Clock
C24	P_REQC#	Input	Control Bus Request from CPU: Used for debug purpose
A26	P_GNTC#	Output	Control BUS Grant to CPU: Used for debug purpose
A25	P_BRGI#	Input	Control Bus Request/Grant Input: At Primary Device, it receives process bus Request signal At Secondary Device, it receives process bus Grant signal
E21	P_BRGO#	Output	Control Bus Request/Grant Output: At Primary Device, it sends process bus Grant signal At Secondary Device, it sends process bus Request signal
A24	P_CSI#	Input- U	Chip Select: Input
B24	P_CSO#	Output	Chip Select: Output

Note that the primary or secondary device is determined by bootstrap pin, BS_PSD.

NOTES:

#	=	Active low signal
Input	=	Input signal
In-ST	=	Input signal with Schmitt-Trigger
Output	=	Output signal (Tri-State driver)
Out-OD	=	Output signal with Open-Drain driver
I/O-TS	=	Input & Output signal with Tri-State driver
I/O-OD	=	Input & Output signal with Open-Drain driver
U	=	Internal weak pull-up
TS	=	Tri-state
ST	=	Schmitt-Trigger

Flash BUS INTERFACE: used for initialization			
Ball No(s)	**Symbol**	**I/O**	**Description**
M22, L24, L25, L26, M24, M23, M25, M26, N23, N24, N26, N25, P25, P26, P24, P22	FS_D[15:0]	I/O-TS	Flash Data Bit[15:0] Share with P_D[15:0]
G24, G25, G26, H24, J23, H25, K22, H26,E24, F23, D26, E26, G23, H22, F24, F25, F26, H23	FS_A[18:1]	Input /Output	Flash Address Bit[18:1] share with P_A[10:1] and P_D[31:24]
C23	FS_CS	Output	Flash Memory Chip Select
Frame Buffer Interface			
Ball No(s)	**Symbol**	**I/O**	**Description**
AB2, AB1, Y4, AA3, AA2, AA1, W5, W4, Y3, Y2, Y1, W3, V4, W2, W1, V3, U5, U4, V2, V1, U3, U2, T4, U1, T3, T2, T1, R4, R3, R2, R1, R5, L2, L3, K1, L4, K2, K3, M5, J1, J2, K4, J3, H1, K5, H2, J4, H3, G1, G2, G3, H4, F1, F2, H5, F3, G4, E1, E2, D1, F4, E3, D2, E4.	L_D[63:0]	I/O-TS, U	Frame Buffer – Data Bit[63:0]
A2, A3, B3, C4, D5, E6, C5, D6, A4, B5, E8, A5, D7, C6, B6, A6, D8.	L_A[20:3]	Output	Frame Buffer – Address Bit[20:3]
D3	L_CLK	Output	Frame Buffer Clock
N2	L_ADS#	Output	Frame Buffer Address Status Control
P3, P4, P1, P2, N1, N3, N4, M1	L_BW[7:0]#	Output	Frame Buffer Individual Byte Write Enable[7:0]
M3, L1	L_WE[1:0]#	Output	Frame Buffer Write Chip Select[1:0]
M4, M2	L_OE[1:0]#	Output	Frame Buffer Read Chip Select[1:0]
BALL NO(S)	**RMII Ethernet Access Ports[11:0]**		
AB23	M_MDC	Output	MII Management Data Clock – (Common for all MII Ports[12:0])
AB24	M_MDIO	IO-TS	MII Management Data I/O – (Common for all MII Ports[12:0])
AA4	M_CLKI	Input	Reference Input Clock
AC18, AB17, AE16, AC14, AD13, AE11, AF9, AC9, AE6, AC6, AF3, AC3.	M[11:0]_RXD[1]	Input- U	Ports[11:0] – Receive Data Bit[1]
AD19, AC17, AD16, AF15, AF13, AF11, AD10, AE8, AF6, AF4, AE3, AD2.	M[11:0]_RXD[0]	Input- U	Ports[11:0] – Receive Data Bit[0]
AE19, AF18, AF16, AD14, AC13, AD11, AE9, AD8, AB8, AD5, AF2, AB4.	M[11:0]_CRS_DV	Input- U	Ports[11:0] – Carrier Sense and Receive Data Valid
AD18, AC16, AC15, AE14, AD12, AC11, AF8, AD7, AF5, AD4, AC4, AB3.	M[11:0]_TXEN	Output	Ports[11:0] – Transmit Enable
AF19, AE17, AB15, AF14, AE12, AB12, AD9, AE7, AC7, AC5, AD3, AC2.	M[11:0]_TXD[1]	Output	Ports[11:0] – Transmit Data Bit[1]
AB19, AD17, AD15, AB13, AF12, AF10, AC10, AF7, AD6, AE4, AF1, AA5.	M[11:0]_TXD[0]	Output	Ports[11:0] – Transmit Data Bit[0]
AE18, AF17, AE15, AE13, AC12, AE10, AB10, AC8, AE5, AB6, AE2, AC1.	M[11:0]_LNK	Input- ST, U	Ports[11:0] -- Link Status

BALL NO(S)	GMII Gigabit Ethernet Access Port[12]		
AE24, AF24, AA22, AC23, AD24, AF26, AE26, AD26	M[12]_RXD[7:0]	Input- U	Port[12] -- Receive Data Bit[7:0]
AD23	M[12]_RX_DV	Input- U	Port[12] -- Receive Data Valid
AC22	M[12]_RX_ER	Input- U	Port[12] -- Receive Error
AF20	M[12]_CRS	Input- U	Port[12] – Carrier Sense
AF21	M[12]_COL	Input- U	Port[12] – Collision Detected
AD25	M[12]_RXCLK	Input- U	Port[12] – Receive Clock
AD22, AC21, AE22, AF22, AC20, AD21, AE21, AC19.	M[12]_TXD[7:0]	Output	Port[12] -- Transmit Data Bit[7:0]
AE23	M[12]_TX_EN	Output	Port[12] -- Transmit Data Enable
AB21	M[12]_TX_ER	Output	Port[12] -- Transmit Error
AE20	M[12]_ TXCLK	Output	Port[12] – Gigabit Transmit Clock
AD20	M[12]_LNK	Input-ST, U	Port[12]-- :Link Status
AF23	GREF_CLK	Input- U	Port[12] – Gigabit Reference Clock
BALL NO(S)	**PCS (TBI) Interface Gigabit Ethernet Access Port[12]**		
AD26, AE26, AF26, AD24, AC23, AA22,AF24, AE24, AD23, AC22,	GP_RXD[9:0]	Input- U	Port[12] – PCS Receive Data Bit[9:0]
AF21	GP_RXCLK1	Input- U	Port[12] – PCS Receive Clock 1
AD25	GP_RXCLK0	Input- U	Port[12] – PCS Receive Clock 0
AC19, AE21, AD21, AC20, AF22, AE22, AC21, AD22, AE23, AB21	GP_TXD[9:0]	Output	Port[12] – PCS Transmit Data Bit[9:0]
AE20	GP_TXCLK	Output	Port[12] – PCS Gigabit Transmit Clock
AD20	GP_LNK	Input- ST, U	Port[12] – PCS Link Status
AF23	GREF_CLK	Input – U	Port[12] – PCS Gigabit Reference Clock

XPipe Interface		I/O	Function
B23	X_DCLKI	Input	Xpipe Data Clock Input
C22	X_DENI	Input	Xpipe Data Enable Input
D22	X_FCI	Input	Xpipe Flow Control Input
C15, D15, A16, B16, C16, E15, A17, D16, B17, C17, A18, B18, D17, C18, A19, B19, D18, C19, E17, A20, B20, C20, D19, A21, B21, C21, D20, A22, B22, A23, D21, E19.	X_DI[31:0]	Input	Xpipe Data Input Bits[31:0]
A15	X_DCLKO	Output	Xpipe Data Clock Output
B15	X_FCO	Output	Xpipe Flow Control Output
D14	X_DENO	Output	Xpipe Data Enable Output
C7, B7, A7, C8, D9, B8, A8, C9, D10, E10, B9, A9, C10, B10, D11, A10, C11, B11, A11, C12, D12, B12, E12, A12, D13, C13, A13, B13, B14, A14, C14, E14.	X_DO[31:0]	Output	Xpipe Data Output Bit[31:0]
Port Mirroring			
AA26	PM_DENI	Input- TS, U	Port Mirroring Data Enable Input
AA25, AA24	PM_DI[1:0]	Input- TS, U	Port Mirroring Input Data Bit[1:0]
Y23	PM_DENO	Output	Port Mirroring Data Enable Output
AB26, W22	PM_DO[1:0]	Output	Port Mirroring Output Data Bit[1:0]
TEST FACILITY	**Use for debug purpose**		
F5	T_MODE#	IO-TS, U	Test Pin – Set Mode upon Reset, and provides test status output
W22, AB26, Y23, AA24, AA25, AA26, W23, Y24, Y25, Y26, W24, U22, V23, W25, W26, V24, U23, V25, V26, U24, U25, R22, T23, U26, T24, T25, T26, R23, R24, R25, R26, P23	T_D[31:0]	Output	Test Output
LED Interface			
AC25	LE_DI	Input- U	LED Serial Data Input Stream
AA23	LE_SYNCI#	Input- U	LED Input Data Stream Envelop
AB24	LE_CLKO	Output	LED Serial Interface Output Clock
AC26	LE_DO	Output	LED Serial Data Output Stream
AB25	LE_SYNCO#	Output	LED Output Data Stream Envelop

System Clock, Power, and Ground Pins			
N5	S_CLK	Input	System Clock at 100 MHz
E9, E18, J5, J22, N22, P5, V5, V22, AB9, AB18	VDD	Power	+2.5 Volt DC Supply
E7, E11, E16, E20, G22, L22, T22, Y22, AB20, AB16, AB11, AB7, Y5, T5, L5, G5	VCC	Power	+3.3 Volt DC Supply
E5, E13, E22, L11, L12, L13, L14, L15, L16, M11, M12, M13, M14, M15, M16, N11, N12, N13, N14, N15, N16, P11, P12, P13, P14, P15, P16, R11, R12, R13, R14, R15, R16, T11, T12, T13, T14, T15, T16, AB5, AB14, AB22	VSS	Power Ground	Ground
C1, C1	AVDD[1:0]	Analog Power	Used for the PLL
A1, A1	AVSS[1:0]	Analog Ground	Used for the PLL
Bootstrap Pins			
P23	BS_BMOD	Input	Control Bus mode MUST BE SET TO 0
R26	BS_RW	Input	Control Bus Read/Write Control Polarity Selection Default=1 0=R/W# ; 1=W/R#
R24	BS_PSD	Input	Primary Device Enable Pin Default=1 0=Secondary 1=Primary
R23	BS_RDYOP	Input	Option of merge the RDY_ and B_RDY as one pin Default=1 0=Merged pin 1=Separated pins

NOTES:

#	=	Active low signal
Input	=	Input signal
In-ST	=	Input signal with Schmitt-Trigger
Output	=	Output signal (Tri-State driver)
Out-OD	=	Output signal with Open-Drain driver
I/O-TS	=	Input & Output signal with Tri-State driver
I/O-OD	=	Input & Output signal with Open-Drain driver
U	=	Internal weak pull-up
TS	=	Tri-state
ST	=	Schmitt-Trigger

5 The Media Access Control (MAC)

The MDS113CG MAC/GMAC contains twelve Fast Ethernet MACs and one Gigabit Ethernet MAC, defined by the IEEE Standard 802.3 CSMA/CD. Each Fast Ethernet MAC is connected to a Physical Layer (PHY) via the Reduced Media Independent Interface (RMII), and the Gigabit Ethernet MAC is connected to a PHY via the Gigabit Media Independent Interface (GMII) or the Ten Bit Interface (TBI). The MAC/GMAC sublayer consists of a Transmit and Receive section and is responsible for data encapsulation/decapsulation.

Data encapsulation/decapsulation involves framing (frame alignment and frame synchronization), handling source and destination addresses, and detecting physical medium transmission errors. The MAC/GMAC also manages half-duplex collisions, including collision avoidance and contention resolution (collision handling). The MDS113CG includes an optional MAC Control sublayer ("MAC Control") used for IEEE Flow Control functions.

During frame transmission, the MAC transmit section encapsulates the data by prepending a preamble and a Start of Frame Delimiter (SFD), inserts a destination and source address, and appends the Frame Check Sequence (FCS) for error detection.

During frame reception, the MAC receive section verifies that the CRC is valid, de-serializes the data, and buffers the frame into the Receive FIFO. The MAC/GMAC then signals the Frame Engine, using Receive Direct Memory Access (RxDMA), that data is available in the FIFO and is ready for storage. When necessary, the MAC/GMAC regenerates the Frame Check Sequence and performs "padding" for frames less than 64 bytes.

MAC/GMAC Configuration

MAC/GMAC operations are configured through the global Device Configuration Register (DCR2) and/or the MAC/GMAC Control and Configuration Registers (ECR1), defined in the Register Definition Section of the MDS113CG Datasheet. The default settings for autonegotiation, flow control, frame length, and duplex mode may be changed and configured by the user on a per-port basis, either in hardware or software.

The Inter-frame Gap

The Inter-Frame Gap (IFG), defined as 96 bit times, is the interval between successive Ethernet frames for the MAC/GMAC. Depending on traffic conditions, the measurement reference for the IFG changes. If a frame is successfully transmitted without a collision, the IFG measurement starts from the assertion of the Transmit Enable (TXEN) signal. However, if a frame suffers a collision, the IFG measurement starts from the deassertion of the Carrier Sense (CRS) signal.

Ethernet Frame Limits

A legal Ethernet frame size, defined by the IEEE specification, must be between 64 and 1518 bytes, referring to the packet length on the wire. For frames whose data lengths do not meet the minimum requirements, the MAC/GMAC appends extra bytes (padding) from the PAD field. Frames, longer than the maximum length may either be forwarded or discarded, depending on the register configuration. The maximum frame size is 1518 bytes without VLAN tag and 1522 bytes with VLAN tag.

Collision and Avoidance

If multiple stations on the same network attempt to transmit at the same time, interference could occur causing a collision. The MAC/GMAC monitors the Carrier Sense (CRS) signal to determine if the medium is available before attempting to transmit data. If the transmission medium is busy, the MAC/GMAC defers (delays) its own transmissions to decrease the load on the network. This is called collision avoidance.

If a collision occurs, after the first 64 bytes of data, the MAC/GMAC ceases data transmission and sends the jam sequence to notify all connected nodes of a collision. This jam sequence will persist for 32 bit times. The jam sequence is a 32 bit predetermined pattern used to notify others of a collision on the network.

If a collision occurs during preamble generation, or within the first 64 bytes, the transmitter waits until the preamble is completed and then "backsoff" (that is, stops transmitting) for a specific period (defined by the IEEE 802.3 Binary Exponential Backoff Algorithm) before sending the jam sequence and rescheduling transmission. A frame with a size of no

less than 96 bits (64 bits of preamble and 32 bits of jam pattern), is sent to guarantee that the duration of the collision is long enough to be detected by the transmitting ports involved.

Autonegotiation

The default value of the MDS113CG MAC/GMAC enables Autonegotiation. The default value is overwritten if the PHY lacks the ability to support Autonegotiation, which is ascertained through its respective management interface, RMII/GMII. The Autonegotiation process detects the different modes of operation (i.e. speed selection, duplex mode) supported by the system at the other end of the link segment. Upon power on/reset, the PHY generates a special sequence of fast link pulses (FLPs) to begin Autonegotiation. The MDS113CG MAC/ GMAC, supporting autonegotiation, reads the results of the operation from the MAC/GMAC configuration registers.

MAC Control Frame

MAC Control Frames, as defined by the IEEE, are used for specific control functions within the MAC Control sublayer "MAC Control." Similar to data frames, control frames are also encapsulated by the CSMA/CD MAC, meaning that they are prepended by a Preamble and Start of Frame delimiter and appended by a Frame Check Sequence. These frames may be distinguished from other MAC frames by their length/type field identifier (88.08h). The control functions are distinguished by an opcode contained in the first two bytes of the frame. Upon receipt, MAC control parses the incoming frame and determines, by looking at the opcode and the MAC address, whether it is destined for the MAC (a data frame) or for a specific function within MAC Control. After performing the specified functions, the MDS113CG discards all MAC control frames it receives, regardless of the port configuration. These control frames are not forwarded to any other port and are not used to learn source addresses.

Flow Control

Flow control reduces the risk of data loss during long bursts of activity, by saturating the buffer memory with backlogged frames. The MDS113CG supports two types of Flow Control: Collision-based for half-duplex mode and IEEE 802.3x Flow Control for full duplex mode. In both cases, the MDS113CG recognizes congestion by constantly monitoring available frame buffer memory. When the amount of free buffer space has been depleted, the MDS113CG initiates the flow control mechanism appropriate to the current mode of operation. Setting the Flow Control (FC_Enable) bit in the MAC Port Configuration Register (ERC1) turns this operation on, thereby initiating PAUSE frames or applying backpressure flow control when necessary.

Collision-BasedFlow Control

Collision-based Flow Control, also referred to as Backpressure Flow Control, inhibits frame reception for ports operating in half-duplex mode by "jamming" the link. When the free buffer space drops below a user-defined buffer memory threshold, the MDS113CG sends a jam sequence to all non transmitting ports, after approximately eight bytes of payload data has been received, to generate a collision. The jam sequence is a predefined serial data stream sent to all ports to indicate that there has been a collision on the network. These ports will delay (defer) the transmission of data onto the network until the sequence has been completed.

IEEE 802.3x Flow Control

IEEE 802.3x Flow Control reduces network congestion on ports that are operating in full duplex mode using MAC Control PAUSE frames and is managed by the Flow Control Management Registers. The full-duplex PAUSE operation instructs the MAC to enable the reception of frames with a destination address equal to a globally assigned 48-bit reserved multicast address of 01-80-C2-00-00-01. These PAUSE frames are subsets of MAC Control frames with an opcode field of 0x0001 and are used by the MAC Control to request that the recipient stops transmitting non-control frames for a specific period. The PAUSE Timer is loaded from the PAUSE frame and is started upon the reception of a PAUSE frame. It will request a length of time for which it wishes to inhibit data frame transmission.

In general, the IEEE standard allows pause frames longer than 64 bytes to be discarded or interpreted as valid. The MDS113CG recognizes all MAC Control frames (PAUSE frames) between 64 and 1518 bytes long. Any PAUSE frames presented to the MAC outside of these parameters are discarded.

Frame Carrier Extension

At speeds faster than 100 Mbps and operating in half-duplex mode, it is possible for the receiving port to begin frame transmission before the source frame from the sending port reaches the receiving end, if

the frame is less than 512 bytes. The slot time is defined as the round trip delay between the sending port and the receiving port. The frame must be long enough to reach the destination port before it decides to transmit. This is eliminated when using Carrier Extension.

Carrier extension allows the slot time to be increased to a sufficient value for accommodating the faster speeds of Gigabit operation without increasing the standard 64-byte minimum frame size. Extension bits, recognized and extracted by the receiving MAC, are appended to frames that are less than 4096 bits (i.e., the Gigabit Ethernet slot time). This ensures that the transmitted frame is received before the receiving port begins transmission, thereby avoiding a collision result. To utilize Carrier Extension, the Physical Layer (PHY) must implement an encoding/ decoding scheme (i.e., Block Encoding) that distinguishes between data and non-data bits. Once the first frame has been extended, the MAC/GMAC sends frames continuously using a method called Frame Bursting.

Frame Bursting

At speeds faster than 100 Mbps and operating in half-duplex mode, the MAC/GMAC can transmit a series of frames without relinquishing control of the transmission medium. This is called Frame Bursting. Frame Bursting is utilized when a frame must be extended to the length of the slot time. With Frame bursting, only the first transmitted frame requires extension. Once a frame has been successfully transmitted, the Transmit section may submit consecutive frames onto the medium without contention, provided that no idle conditions exist between frames (e.g., no inter-frame Gap). The transmitting MAC/GMAC inserts extension bits to be detected and extracted by the receiving MAC/GMAC, into the inter-frame space interval. The MAC/GMAC may continuously initiate burst frame transmission up to the burst limit of 65,536 bits.

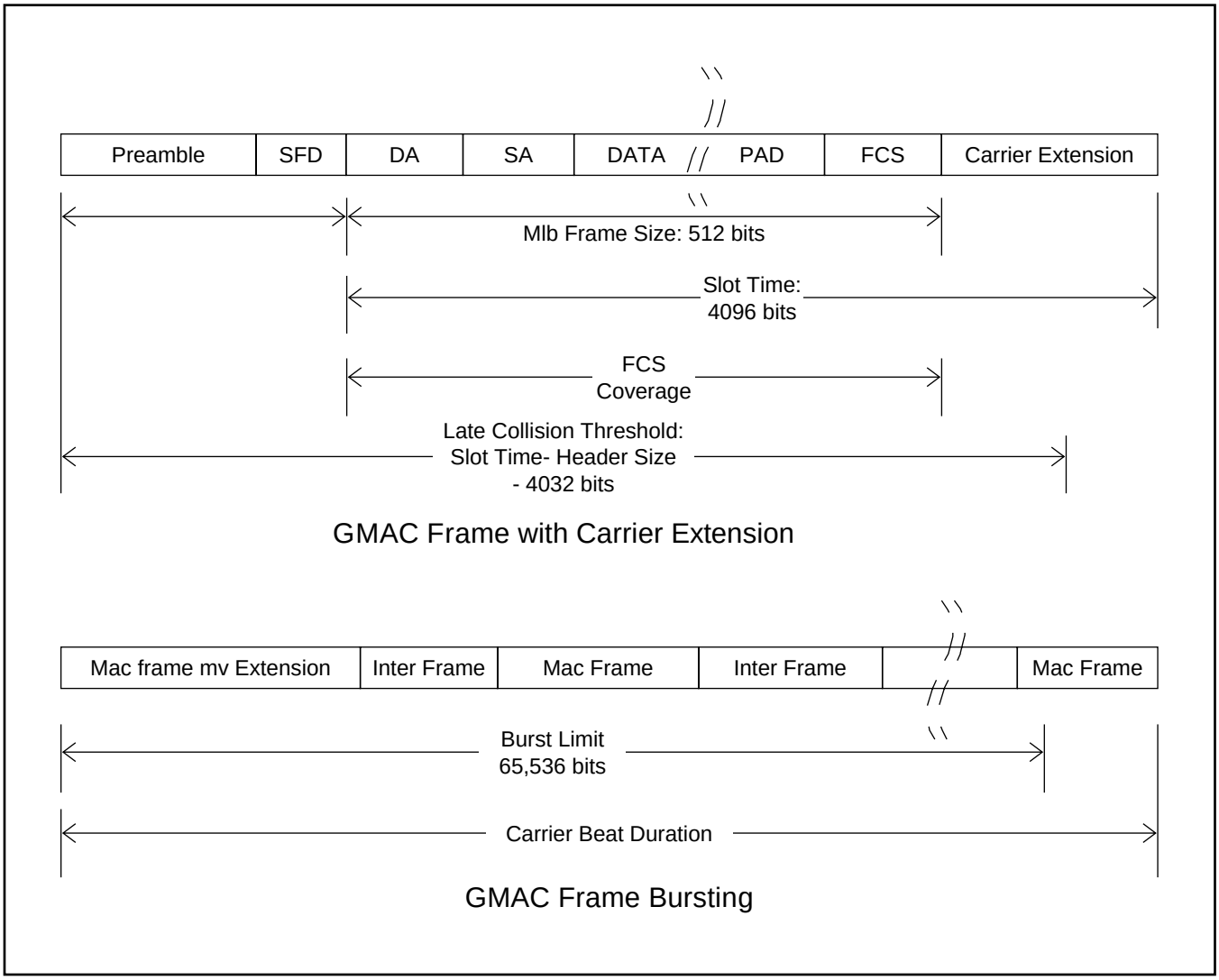


Figure 3 - Frame with Carrier Extension and Frame Bursting

6 Frame Engine Description

The Frame Engine is the heart of the MDS113CG. It coordinates all data movements, ensuring fair allocation of the memory bandwidth and the XPipe bandwidth.

When frame data is received from a MAC port, it is temporarily stored in the MAC RxFIFO until the Frame Engine moves it to the chip's external memory one granule (128-byte-or-less fragment of frame data) at a time. The Frame Engine then issues the Search Engine a switching request that includes the source MAC address and the destination MAC address. After the Search Engine has resolved the address, it transfers the information back to the Frame Engine via a switching response that includes the destination port.

When the destination port is idle, the frame data is fetched from the memory and is written to the destination port's MAC TxFIFO. However, when the destination port is busy transmitting another frame, the Frame Engine writes a transmission job that includes a frame handle for future identification. These transmission jobs are stored in the destination port's transmission scheduling queue (TxQ). When the destination port is ready, the Frame Engine selects the head-of-line job from a TxQ. The frame, specified by the job, will be fetched from the memory and will be written to the MAC TxFIFO.

For unicast frames, if the destination device is local (i.e., the destination port is located in the same device), the Frame Engine writes a job into the destination port's transmission scheduling queue (TxQ). The Transmit DMA (TxDMA) moves the frame data to the MAC TxFIFO once the frame's transmission job is selected for transmission.

If the destination device is remote (i.e., the destination port is located on another device, and can only be reached through the XPipe), all signaling between the two devices are sent as XPipe messages. The Frame Engine sends a scheduling request message via the XPipe to the destination port. This message asks the remote Frame Engine to write a job into the destination port's TxQ. When that job is selected, the remote Frame Engine sends a data request message via the XPipe to the local Frame Engine. Reception of a data request message triggers the forwarding engine module to forward the frame data to the destination port, one granule at a time through the XPipe until the end of file (EOF) safely arrives at the remote port's MAC TxFIFO.

A frame is stored in a Frame Data Buffer (FDB) until it is transmitted. FDBs are external, located in a MDS113CG's frame buffer memory. To keep track of per-frame control information, the Frame Engine maintains one Frame Control Buffer (FCB) per frame. FCBs are internal. Since the Frame Engine does not access the external memory for frame control information, this conserves memory bandwidth for better performance.

The receiving DMA (RxDMA) moves frame data from the MAC RxFIFO to the FDB. Before the RxDMA writes frame data into the FDB, it must obtain a free buffer handle from the buffer manager. A free buffer handle points to an empty or released frame buffer, ensuring that no stored frame data will get overwritten. After the EOF has been safely stored in the FDB, it writes the frame information to the FCB and issues a switching request to the Search Engine. If the frame is found to be bad (e.g., bad CRC), the buffer handle will be released and nothing will be written to the Search Engine or the FCB. This returns the buffer back to circulation and the frame is discarded.

The RxDMA can fail to obtain a free buffer handle for one reason. All buffers are currently occupied. In this case, the RxDMA will discard the frame, without getting a handle. The maximum TxQ lengths are configurable from 128 entries to 1024 entries per queue per queue. 13 TxQs are located in the external memory.

If all buffers are used, no more frames can enter the device. The Frame Engine keeps buffer counters that limit the number of buffers occupied by frames destined for each output port. If a buffer counter exceeds a programmable threshold, its associated output port is "blacklisted." Entering frames destined to this output port are discarded, until the counter goes below the threshold. This threshold is configurable via registers BCT and BCHL. These counters prevent complete depletion of buffers due to an overloaded port, thus allow frames destined for non-congested ports to enter the system. This effectively avoids head-of-line blocking.

7 Frame Buffer Memory

Frame Buffer Memory Configuration

The MDS113CG system utilizes external SRAM for its Frame Buffer Memory configuration, where the size of memory supported is ∫ MB, 1MB and 2MB configurations. The following table shows four memory configuration examples for the MDS113CG system.

SRAM Type	One Bank		Two Bank	
	Address # pin	Size	Address #pin	Size
64Kx32	19	∫ MB	20	1M
128Kx32	20	1MB	21	2M

Table 1 - Type and Size of Memory Chips

The following figure shows the connections between the Frame Buffer Memory and the MDS113CG for one-bank and two-bank memory configurations:

Note: LA[1:0] = 00

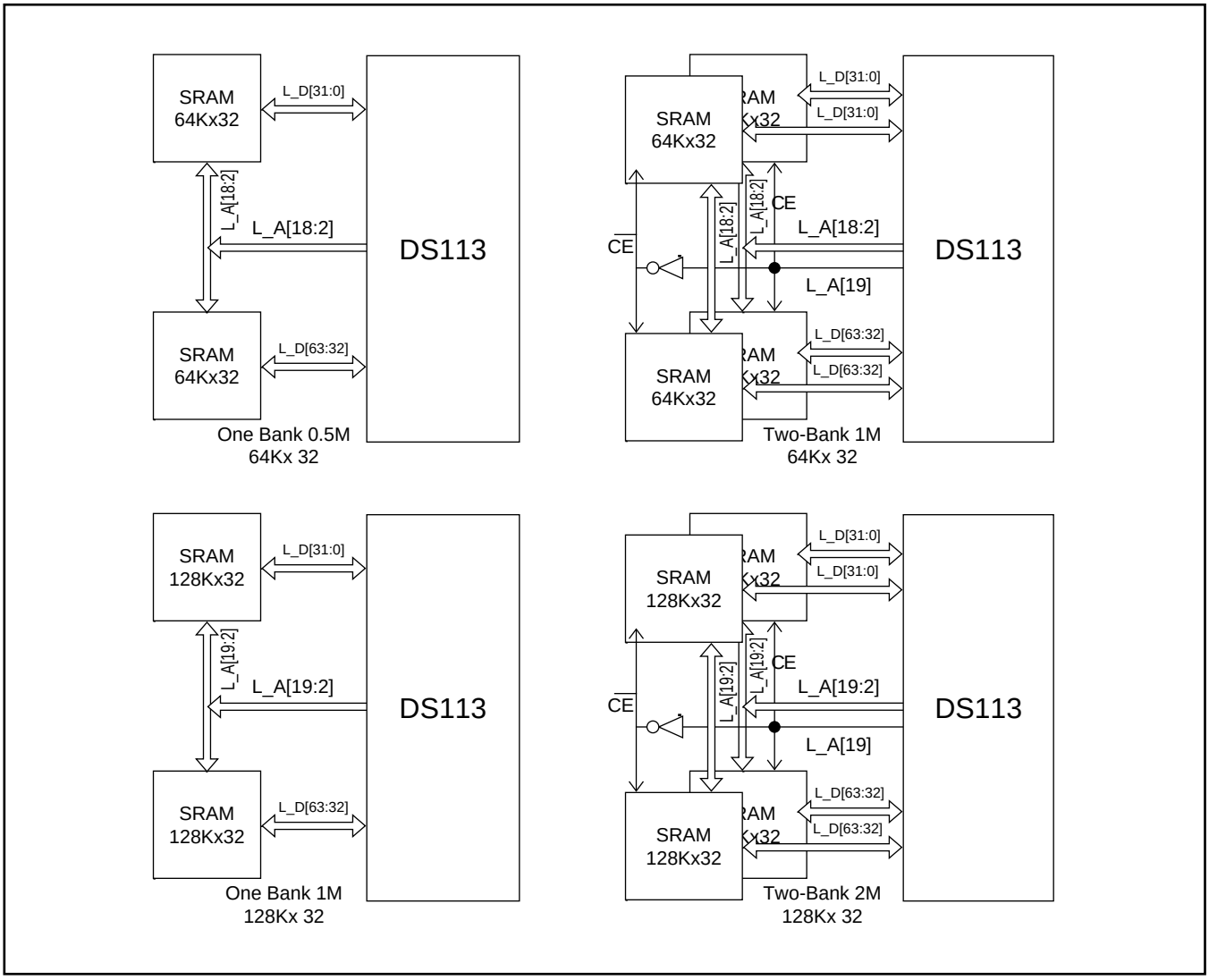


Figure 4 - Frame Buffer Memory Configuration

Frame Buffer Memory Usage

Description:	Unit Size:	Unit Count:	Total Size
Frame Data Buffer (FDB)	1.5 Kbytes	256 to 1K	384 K bytes to 1.5M bytes
Transmission Queue	4 bytes x128 to 4 bytes x 1K	13	6.5 Kbytes to 52 Kbytes
HISC Mailing List	32 Bytes to 64Bytes (Configurable)	128 to 1K	4K bytes to 32 Kbytes (at 32 Bytes each)

Table 2 - Frame Buffer Memory Usage

System Memory Allocation

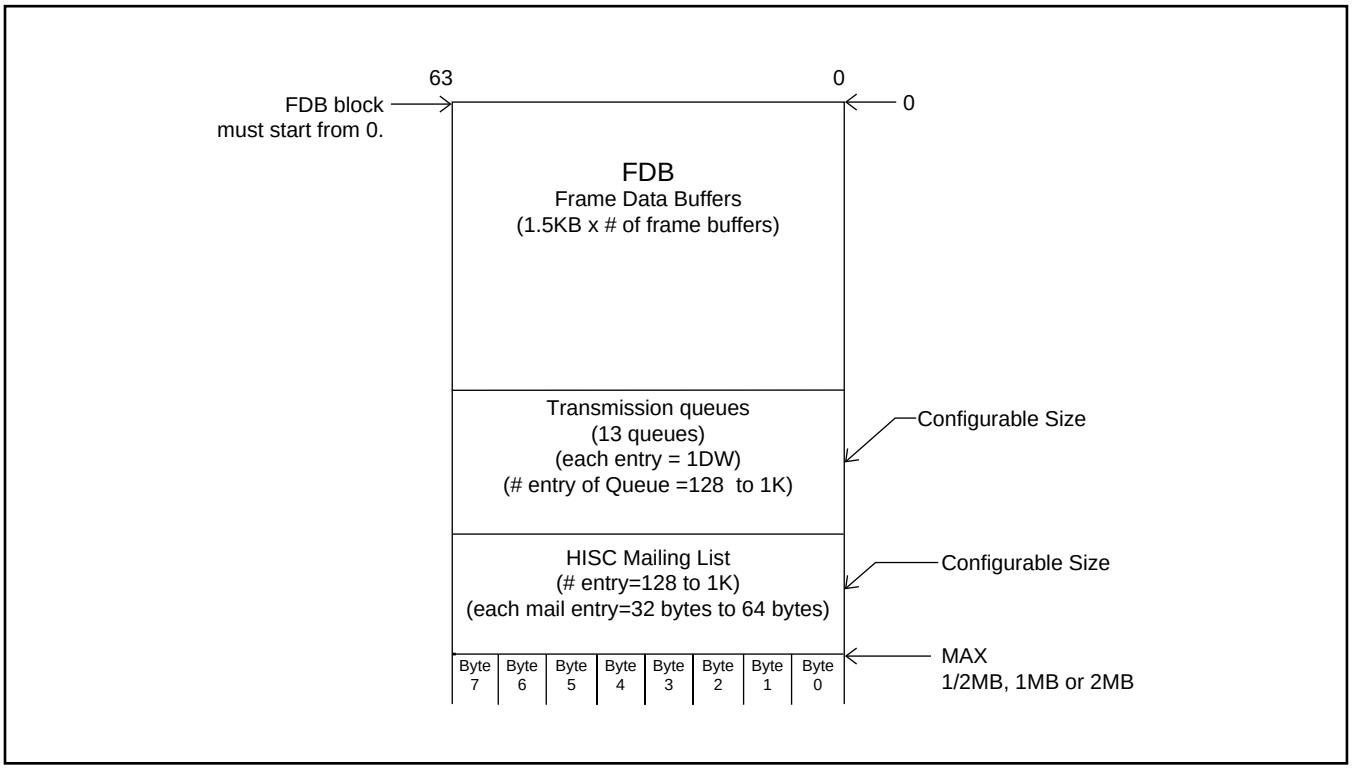


Figure 5 - Memory Map of a System

Frame Data Buffers

Frame Data Buffers (FDBs) accommodate incoming data frames and partition them into data blocks, consisting of 1.5K bytes. The number of data blocks in FDBs is configured by setting the value in the register FCBSL[9:0]. Since the MDS113CG can support up to 2M Bytes memory, the maximum number of data blocks is 1K.

Note: FDBs must start at location 0.

Transmission Queues

Transmission Queues control the scheduling of transmitting ports. The Search Engine maintains the contents of these queues, consisting of up to 13 Transmission Queues and representing each of the 13 ports in the MDS113CG.

Each queue consists of one double word (4-bytes) transmission entry, containing a FDB handle pointing to the corresponding frame in the buffer. The size of the Transmission Queue is 128, 256, 512, or 1K entries, while the location is setup during Power On/ Reset in an internal table called the Queue Control Table (located inside the MDS113CG).

Mailing List

The Mailing List provides a communication channel between two HISC in two-chip configuration. The size of a mail entry is 32 bytes. When the HISC writes mail, the HISC can obtain a free mail by the hardware. Conversely, when the HISC reads its mail, the HISC accesses the mail by the hardware, as well.

Local SBRAM Memory Interface

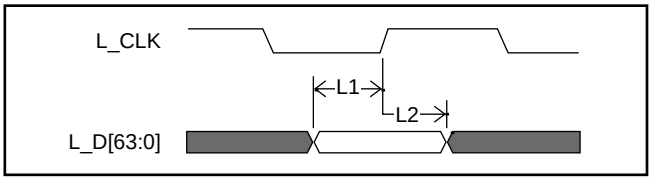


Figure 6 - Local Memory Interface - Input Setup and Hold Timing

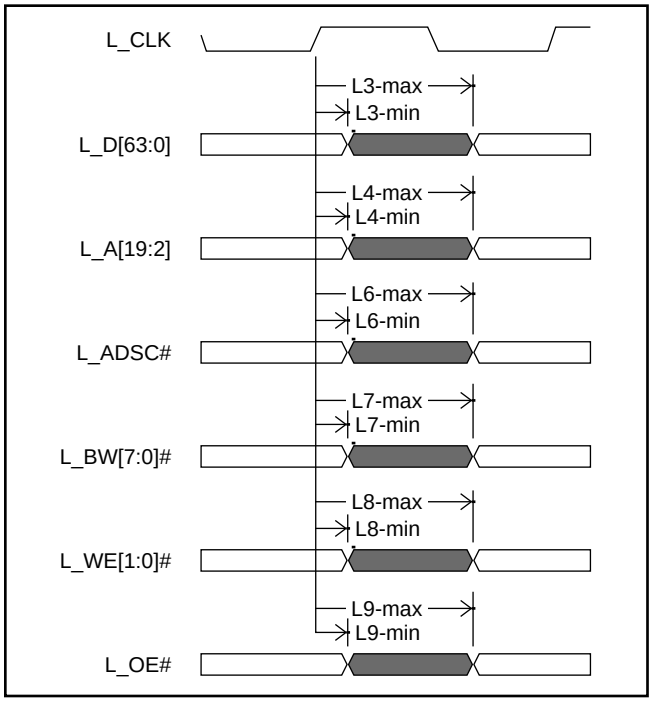


Figure 7 - Local Memory Interface - Output Valid Delay Timing

8 Search Engine

The Search Engine is responsible for determining the destination information for all packet traffic that enters the MDS113CG. The results from all address searches are passed to the Frame Engine to be forwarded, or on to the HISC block for further processing. Either way, the resulting messages provide all the needed information to allow the destination block to process the packet.

The Search Engine has been optimized for high throughput searching, utilizing the integrated Switch Database Memory (SDM). The internal SDM contains up to 2k MAC Control Table (MCT) entries. These MCT entries are searched utilizing a Hashing algorithm.

The search process begins when the Frame Engine transfers the first 64 bytes of a packet header to the Search Engine. These bytes are parsed to extract the information needed to perform the search for the MCT entries that match the source and destination MAC address, generate the search hash keys, and lookup other packet status information.

Layer 2 Search Process

The search process begins when packet header information is transferred to the Search Engine from the Frame Engine.

The Search Engine will search for the destination and source MAC addresses.

Address Learning

Address learning can be performed by either the HISC or the Search Engine.

When the Search Engine is learning and a match is not found to a source address search, it can create a new MCT with the necessary information, and then notify the HISC that a new address has been learned. If the Search Engine request queue becomes 3/4th full, the Search Engine will ignore address learning until the request queue is less full. In that case, packets are forwarded as usual, and a message is sent to the HISC requesting that the HISC learn the new address. If the Search Engine request queue is too full, and the HISC request queue is full, then no learning will take place.

When two MDS113CG chips are connected and are operating with synchronized MCT entries, the HISC processor has the ability to send a request to the Search Engine, instructing it to learn a new address received from the other MDS113CG. The HISC processor can also use this method to make simple edits to the MCT entries for port changes (i.e. source MAC address is now connected to a different port on the MDS113CG).

Flooding and Packet Control

Packets, for which there are no matching destination MCT entries, are by default flooded to all output ports. This can result in broadcast storms and cause the number of flooded packets to increase rapidly. The MDS113CG provides the user a means for setting a level of flooding, by providing a Flooding Control Register (FCR). The FCR allows the user to define a time base (100us to 12.8ms) during which packet flooding at each output port will be counted. The flood control field allows the user to specify limits for the number of flooded packets per source port.

During the time base period, the counter at each port counts the number of flooding packets. Once a counter exceeds the allowed quantity, the Search Engine discards any flooding packets that enter the port during the remainder of the time base flooding period. When the time base period is completed, the flood counter at each port is reset, and the counting process starts over.

The flooding control register is global for setting the limits on all ports, but the individual ports have separate counters to keep track of the number of flooded.

Address Aging

Entries in the MCT database are removed if they have not been used within a user selectable timeframe. This aging process is handled by inspecting a single MCT entry during each clock period. If the entry is valid and subject to aging, an aging flag in the MCT entry is cleared. If the aging flag is already set to zero during the inspection, an aging message is sent to the HISC processor to delete and free up the aged MCT entry. Each time an MCT entry is matched by way of a Search Engine, source search process, the aging flag is asserted to restart the aging process for that entry.

9 The High-Density Instruction Set Computer (HISC)

Description

The High Density Instruction Set CPU (HISC) is specifically designed to implement highly efficient management functions for the MDS113CG switching hardware, minimizing the management activity intervention during frame processing. The HISC is also designed with a powerful instruction set and dedicated hardware interfaces for packet processing and transmission to provide high performance packet transfers between the switching hardware components.

HISC Architecture

The HISC is designed with an advanced pipeline architecture that combines the advantages of both RISC and VLIW architectures. The HISC core combines a rich instruction set with 88 general-purpose registers and support for multiple-way jump. The 88 registers are divided into three parts, eight common general-purpose registers and two banks of 40 registers for two different task contexts. All registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction execution. Each HISC instruction may have up to three sub-instructions, which can be executed in one clock cycle. The resulting architecture is more code efficient, while achieving throughputs up to ten times faster than a CISC processor or up to three times faster than a RISC processor. For a MDS113CG running at 100MHz, the HISC can produce up to 300MIPs processing power.

HISC Operations

With an event-driven operation model, upon the request from the Search Engine, the HISC dynamically manages and maintains the Switch Database including MAC address entries.

The HISC performs the following major operations:

- Resource initialization
- Resource management
- Switching database management

Resource Initialization

The HISC initializes all internal data structures, including the mailbox and switching database data structures, which are used by the HISC and switching hardware.

Resource Management

The HISC can enforce a replacement policy when the number of free data structure for new MAC address entries is lower than the predefined threshold.

Switching Database Management

One of the major management tasks required of the HISC is to create, delete, and modify MAC address entries upon requests from the Search Engine. Generally, the Search Engine performs the learning of new MAC addresses identified in the packet streams.

Communicaton Between HISC and Search Engine

High-speed communication channels are required to provide fast message deliveries between the HISC and switching hardware. One high-performance FIFO provides the required communication channels between the HISC and the Frame Engine.

The high-speed FIFO is used by the Search Engine to send messages, management requests or received packets, to the HISC. Whenever a message is sent to the FIFO, the HISC is notified of the new event. Each message may contain up to two command codes, processed by the HISC sequentially. The HISC can also request the Search Engine to do operations such as search or learn via a HISC I/O interface. After processing the requests, the Search Engine sends the response back to the HISC via the FIFO.

Mailbox

The second communication mechanism is a hardware mailbox that can support variable size messages, exchanged between two HISCs in 2-chip configuration. A major use of the mailbox is to exchange information required for updating the switching database.

HISC-HISC Mail

When one HISC sends a mail message to the other HISC, the first HISC acquires an address of a free mail from the free mail list (maintained by hardware), it writes the mail content to the given memory address. Whenever a management mail message is received from the remote HISC, an event is generated to inform the HISC to process the mail message.

10 The XPIPE

The XPipe provides a high-speed link between systems utilizing two MDS113CG devices. The XPipe incorporates a 32-bit-wide data pipe, with a high-speed point-to-point connection, and a full-duplex interface between devices. While operating at a 100MHz, this interface can provide 3.2G bits per second (Gbps) of bandwidth per pipe in both directions.

XPIPE Connection

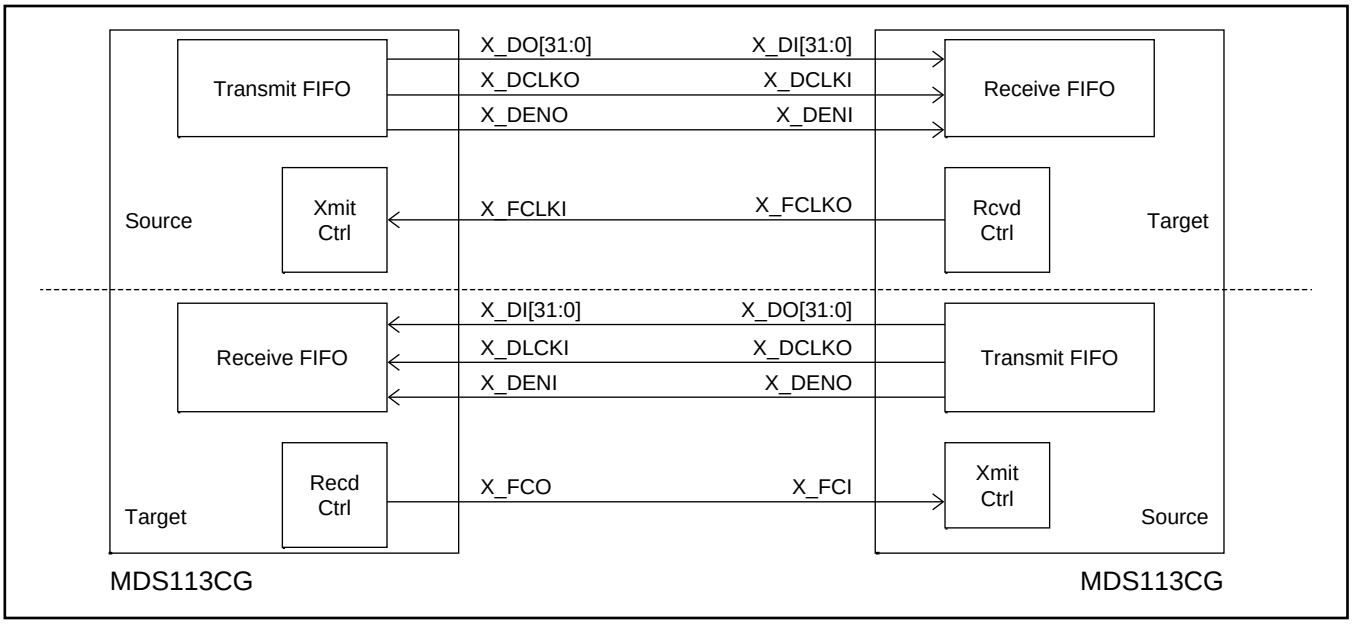


Figure 9 - XPipe System Block Diagram for the MMDS113CGCG

The XPipe interface employs 32 data signals and three control signals for each direction. The pin connections between two MDS113CG devices are depicted in Figure 7. These 32 data signals form a 32-bit-wide transmission data pipe that carries XpressFlow messages to and from the devices. The direction of all signals are from the source to the target device, except for the flow control signal, which sends messages in the opposite direction; from the target to the source. The three control signals consist of a Transmit Clock signal, a Transmit Data Enable signal, and a Flow Control signal.

The Transmit Clock signal (X_CLK) provides a synchronous clock to sample the data signals at the target device. The source device provides the Transmit Data Enable signal (X_DEN) that envelops an entire XPipe message (including the Header and the Payload) and is used to identify the message boundary from the received data stream. The timing relationship between the data clock and data enable signals is described in the XPipe Timing (see XPIPE Timing).

The Flow Control signal (X_FC) monitors the state of the receiving queue at the target end to prevent XPipe message loss. When the target end does not have enough space to accommodate an entire XPipe message, the target device sends a XOFF signal by driving the X_FC signal to LOW. The source device will stop further transmission until the X_FC signal asserts the XON state, which is an active HIGH (refer to the following table).

Signal Name		
Source End	**Target End**	**Description**
X_DO[31:0]	X_DI[31:0]	32-bit-wide Transmit Data Bus – Includes an XPipe Message Header and follows by the data payload.
X_DCLKO	X_DCLKI	Transmit Clock – Synchronous data clock provided by the source end.
X_DENO	X_DENI	Transmit Data Enable – Provided by the source end to envelop the entire XPipe message.
X_FCI	X_FCO	Flow Control Signal– A flow control pin from the target end to signal the source end to active XON/XOFF.

Table 4 - Summary Description of the Source and Target End Signals

The XPipe Message Header provides the payload size, type of message, routing information, and control information for the XPipe incoming message. The routing information includes the device ID and port ID. The header size is dependent upon the message types and may be 2 to 4 words in length.

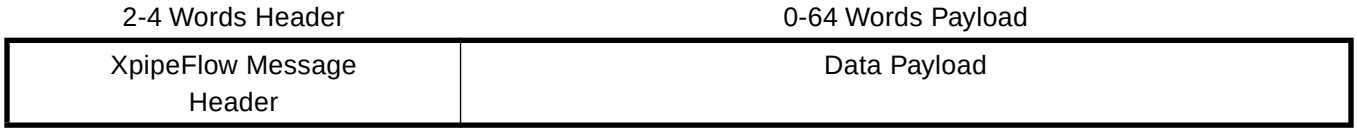


Figure 10 - XPipe Message Header

XPIPE Timing

The source device generates the X_CLK signal to provide a synchronous transmit data clock. The Receiver will then sample the data on the falling (negative) edge of the clock, as shown in Figure 9.

To identify the boundary between the XPipe messages and the data stream, the source device uses the X_DEN signal to envelop the entire XPipe message. That is, a rising (positive) edge at the beginning of the first double word (4 bytes) and a falling (negative) edge at the beginning of the last double word of an XPipe message as shown in Figure 10.

Note: The negative edge does not occur at the end of the last double word, but instead, at the beginning of the last double word. This allows XPipe messages to be sent consecutively (back-to-back).

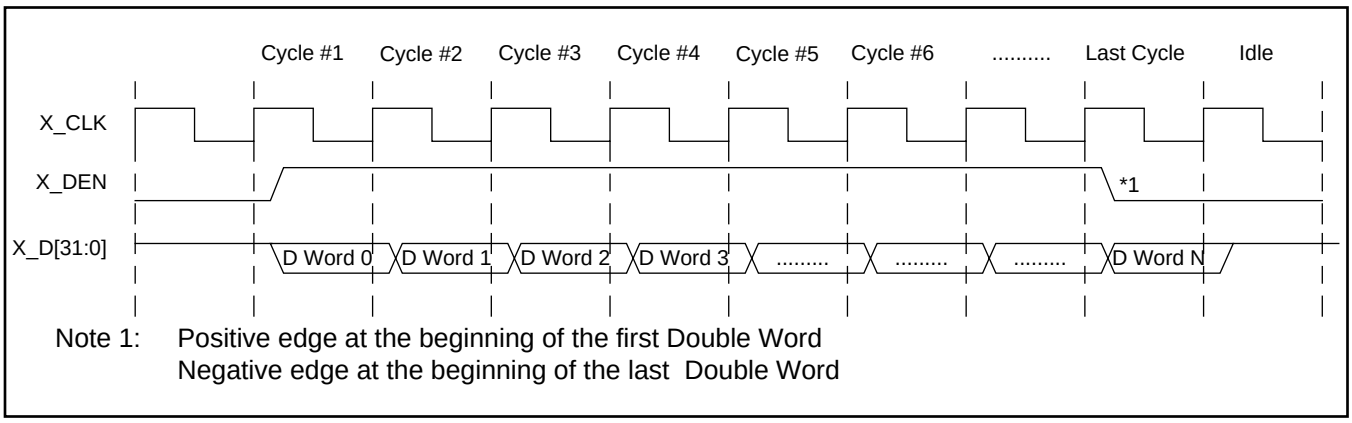


Figure 11 - Basic Timing Diagram of XPipe

XPIPE Interface

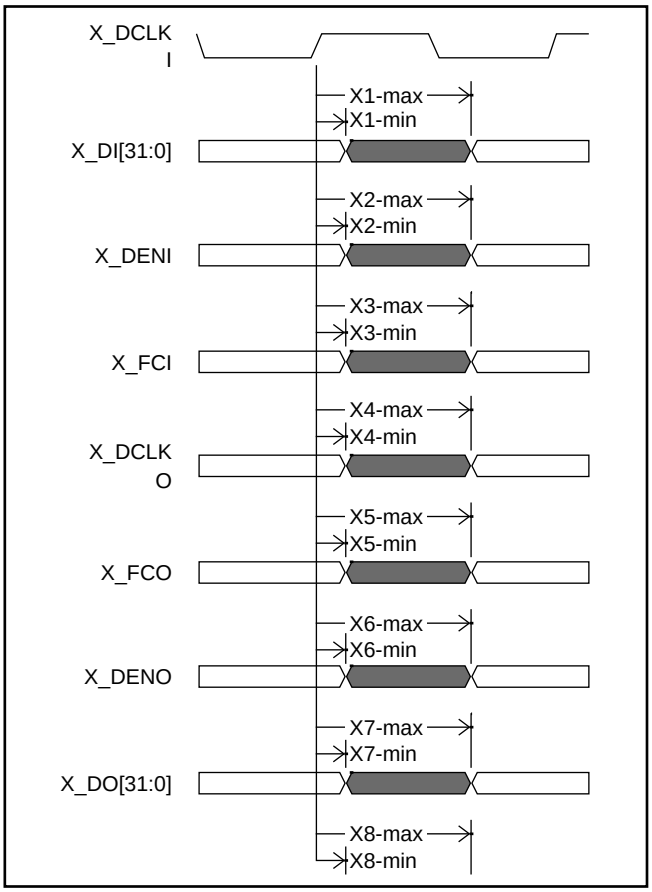


Figure 12 - LXPipe Interface - Output Valid Delay Timing

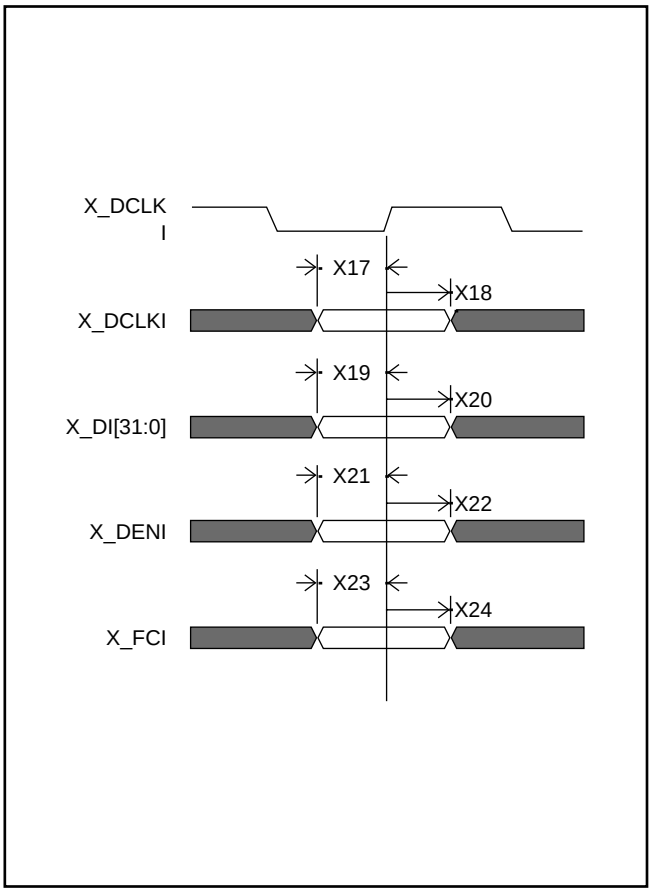


Figure 13 - XPipe Interface - Input Set and Holding Time

11 Physical Layer (PHY) Interface

The Physical Layer Interface is designed to interface Zarlink Semiconductor chipsets to a variety of Physical Layer devices. Reduced Media Independent Interface (RMII) is used for 10/100 interfaces, while Gigabit connections can use either Gigabit Media Independent Interfaces (GMII) or Physical Coding Sublayer (TBI).

The chip ball names for the MAC use M as the first letter of the name, followed by their pin number, and then their function. M1_RXD0 refers to Mac port 1, receive data 0 of the receive data pair.

Reduced MII (RMII))

The MDS113CG implements the Reduced Media Independent Interface (RMII) signals, REF_CLK, CRS_DV, RXD[1:0], TX_EN, and TXD[1:0], defined in Section 5 of the RMII Consortium Specification. The purpose of this interface is to provide a low cost alternative to the IEEE 802.3u MII interface. Under IEEE 802.3u, an MII comprised of 16 pins for data and control is defined. In devices incorporating many MACs or PHY interfaces such as switches, the number of pins can add significant cost as the port counts increase. Architecturally, the RMII specification provides for an additional reconciliation layer on either side of the MII but can be implemented in the absence of an MII. The management interface (MDIO/MDC) is assumed identical to that defined in IEEE 802.3u.

The RMII supports both 10 and 100 Mbps data rates across a two-bit Transmit Data (TXD) path and a two-bit Receive Data (RXD) path.

The RMII uses a single synchronous clock reference sourced from the Media Access Controller (MAC), or an external clock source, to the Physical Layer (PHY). Doubling the clock frequency to 50 MHz allows a reduction of required data and control signals, thereby providing a low cost alternative to the IEEE Std 802.3u Media Independent Interface (MII). The RMII functions to make the differences between copper and optical PHYs transparent to the MAC sublayer.

The RMII specification has the following characteristics:

- It is capable of supporting 10Mb/s and 100Mb/s data rates
- A single clock reference is sourced from the MAC to PHY (or from an external source)
- It provides independent 2 bit wide (di-bit) transmit and receive data paths
- It uses TTL signal levels, compatible with common digital CMOS ASIC processes.

Signal Name	Use	Direction (with respect to the PHY)	Direction (with respect to the MAC)
REF_CLK	Input	Input or Output	Synchronous clock reference for receive, transmit and control interface
M[0:11]_CRS_DV	Output	Input	Carrier Sense/Receive Data Valid
M[0:11]_RXD[1:0]	Output	Input	Receive Data
M[0:11]_TX_EN	Input	Output	Transmit Enable
M[0:11]_TXD[1:0]	Input	Output	Transmit Data
M[0:11]_RX_ER	Output	Input (Not required)	Receive Error

Table 5 - RMII Specification Signals

THE Gigabit Media Independent Interface (GMII)

The GMII supports the 1000 Mbps full-duplex operations of the MDS113CG, based on the Media Independent Interface (MII) defined by IEEE Std 802.3 (Clause 22). The GMII retains the names and functions of most of the MII signals, but defines valid signal combinations for 1000 Mbps operations. The GMII transfers data in each direction for the Data[7:0], Delimiter, Error, and Clock signals. The GMII implementation extends the Transmit Data (TXD) and Receive Data (RXD) signals of the MII from four bits wide to eight bits wide and synchronizes the data and the delimiters using a Gigabit Transmit Clock (GTX_CLK) instead of the MIIs' Transmit Clock (TX_CLK). The GMII is designed to make the differences between various media transparent to the MAC.

The MII Management Interface

The GMII uses the MII Management Interface to control and gather status information from the Gigabit Physical Layer (PHY) to configure MDS113CG operations using Autonegotiation. The management interface consists of a pair of signals, called the M_MDIO and M_MDC management pins, that will physically transport the management information across the GMII, format frames, exchange management frames, and read from or write to specific MDS113CG management registers. Both of these pins have internal pull-up resistors and may be placed in a high impedance state to allow another master to manage the devices on the interface.

MII Command and Status Registers

The GMII utilizes the MII Command and Status registers defined in the 10/100 Mbps Specification and additional extended registers to support Autonegotiation (IEEE Std 802.3, Clause 37). The commonality of the MII management registers will allow the MDS113CG to determine the capabilities supported by the PHY and to implement such functions as "Start of Frame" and "Determine PHY Address."

The Physical Coding Sublayer (TBI) Interface

For the Zarlink Semiconductor MDS113CG, the 1000BASE-X TBI Interface is designed internally and may be utilized in absence of a GMII. The TBI interface incorporates all the functions required by the GMII, to include encoding/decoding serial data to/from 8B/10B for PHY communication and generating Collision Detect (COL) signals for half-duplex mode. In addition, it manages the Autonegotiation process by informing the management entity via the GMII that the PHY is ready for communications. The on-chip TBI may be disabled, using the Device Configuration Register (DCR2), if TBI exists within the Gigabit PHY. The TBI interface provides a uniform interface for all 1000 Mbps PHY implementations.

The TBI Interface comprises of the TBI Transmit, Synchronization, TBI Receive, and Autonegotiation processes for 1000BASE-X. The TBI communicates with the GMII using a byte-wide synchronous data path, where packet delimiting is provided by separate Transmit (TX_EN and TX_ER) and Receive signals (RX_EN and RX_ER). This interface also provides the functions necessary to map packets between the GMII and the physical medium.

The TBI Transmit process sends the GMII signals TXD[7:0], TX_EN, and TX_ER to the physical medium and generates the GMII Collision Detect (COL) signal based on whether a reception is occurring simultaneously with transmission. Additionally, the Transmit process generates an internal "transmitting" flag and monitors Autonegotiation to determine whether to transmit data or to reconfigure the link.

The TBI Synchronization process determines whether or not the receive channel is operational.

The TBI Receive process generates RXD[7:0], RX_DV, and RX_ER on the GMII, and the internal "receiving" flag for use by the Transmit processes.

The TBI Autonegotiation process provides the means to exchange configuration information between two devices that share a link segment and to automatically configure the link for the appropriate speed of operation for both devices.

12 The Control Bus

The Control bus provides the communication path between the Switch Devices and Flash Memory, and between any two MDS113CG Switches (see the following figure).

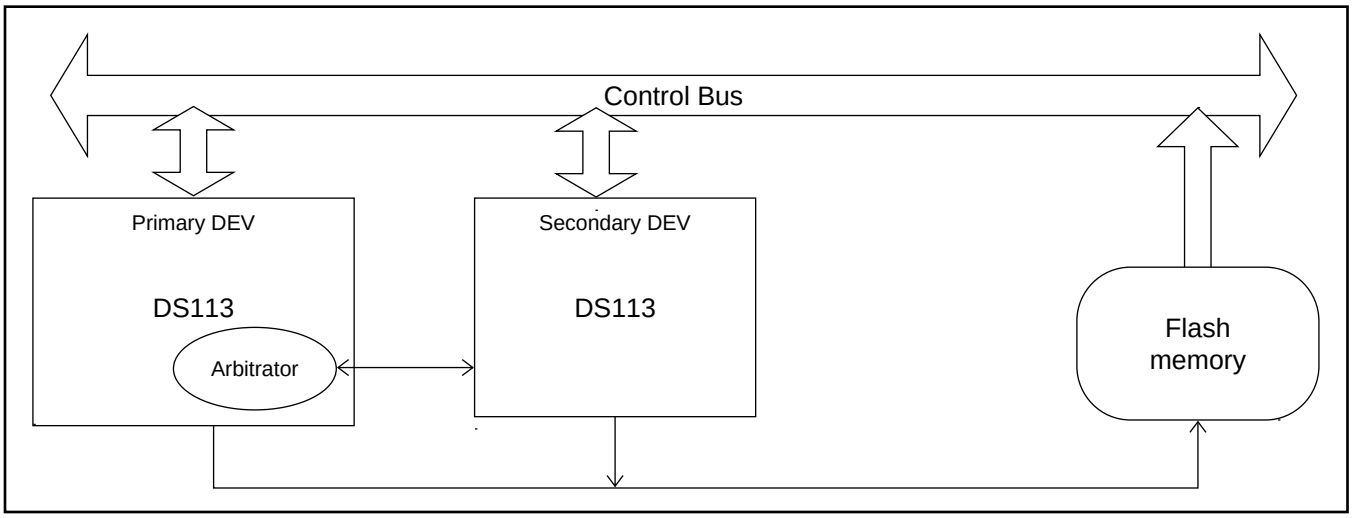


Figure 14 - Control Bus Configuration

Power On/Reset Configuration

On power-up, the following four Bootstrap bits of the following table are used:

Name	Default	Functional Description
BS_BMOD	1	Bus Mode Must be 0
BS_RW	1	Selects R/W Control polarity 0=R/W# 1=W/R#
BS_PSD	1	Primary Device Enable 0=Secondary Mode 1=Primary Mode (The arbiter is activated in the chip with Primary Device.)
BS_RDYOP	1	Option of merger the RDY and B_RDY 0=merged RDY and B_RDY pin 1=Separated RDY and B_RDY pins

Table 6 - Bootstrapping Options

Control Bus Clock Interface

The Control BUS Interface allows the Control BUS clock to operate at clock rates different from the system clock rate. The Control BUS Clock rate is always less than or equal to the System Clock rate.

Address and Data Buses

- The data bus is a synchronous, 32-bit bus that can receive 16 or 32-bit wide data. The Flash memory uses a 16-bit data bus. The data bus supports 32 bit wide data.
- The address buses supports 10 address bits ([10:1]).
- Each device occupies 2048 bytes of Input/ Output space.

Bus Master

The nomenclatures "Master" and "Slave" refer to the device that possesses the Control BUS Interface, while the designations of "Primary" and "Secondary" refer to the device that possesses the Bus Arbiter. The primary or secondary device is determined during Power On/Reset, bootstrap options, while the master or slave device changes dynamically, and will be determined by the Arbiter. The arbiter (located within the primary device) selects one of the devices as the Master.

Note: The primary device may be the Master or the Slave. The master device is the bus master (controls the bus), while the other device is a slave device.

Control Bus Cycle Waveforms

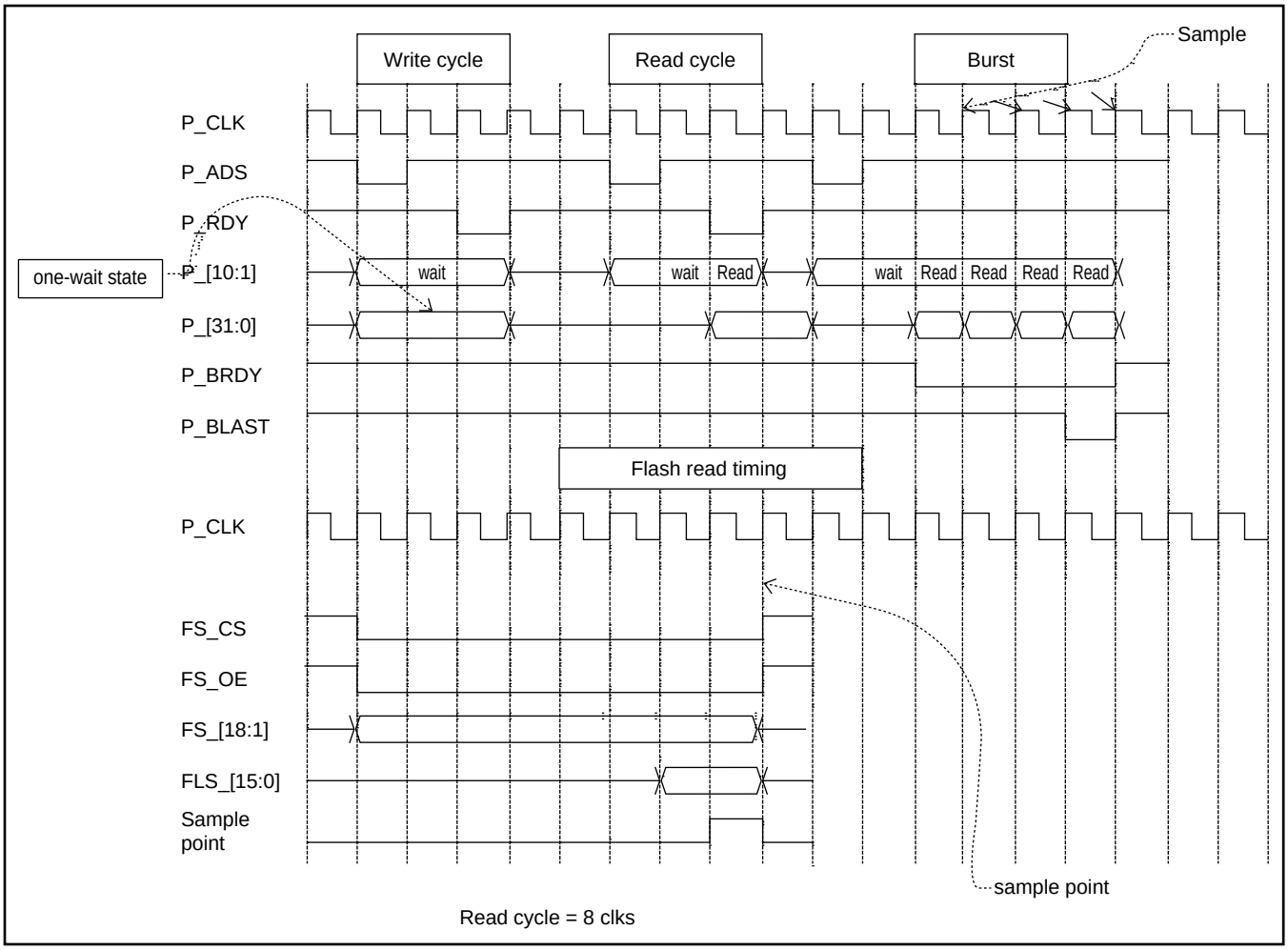


Figure 15 - Control Bus I/O and Flash Bus Access Operations

The Control Bus Interface

TThe HISC processor of the Master device communicates with the slave device as a CPU function.

Arbiter

The arbiter of the MDS113CG is an internal logic device used to determine which device will function as the master device. The connections between the master device, slave device, and the CPU are used for debugging purposes only (see the following figure).

Note: A CPU is used only for debugging purposes and cannot be involved in switching decisions or management activities.

During Power On/Reset, the bootstrap pin, BS_PSD, determines which device will be the primary and activates the arbiter of that device. At most, three devices, two MDS113CG devices and one CPU (in debug mode), can operate on the Control BUS Interface at the same time.

Each device may request access to the Control BUS Interface by sending a Request signal to the arbiter. The arbiter then sends a Grant signal acknowledging which device has been chosen.

An arbitrate scheduler, located within the arbiter, decides which device functions as the Master device. If the Master is the secondary device, the arbiter will send a Grant signal and a Chip Select (P_CS) signal to the device. If the Master is the primary device, the Grant signal is sent directly to the Master State Machine (MSM) by an internal signal. The scheduler then performs a round robin configuration and allows each device to be the Master device.

Note: During Power On/Reset, the arbiter always selects the primary device to be master device.

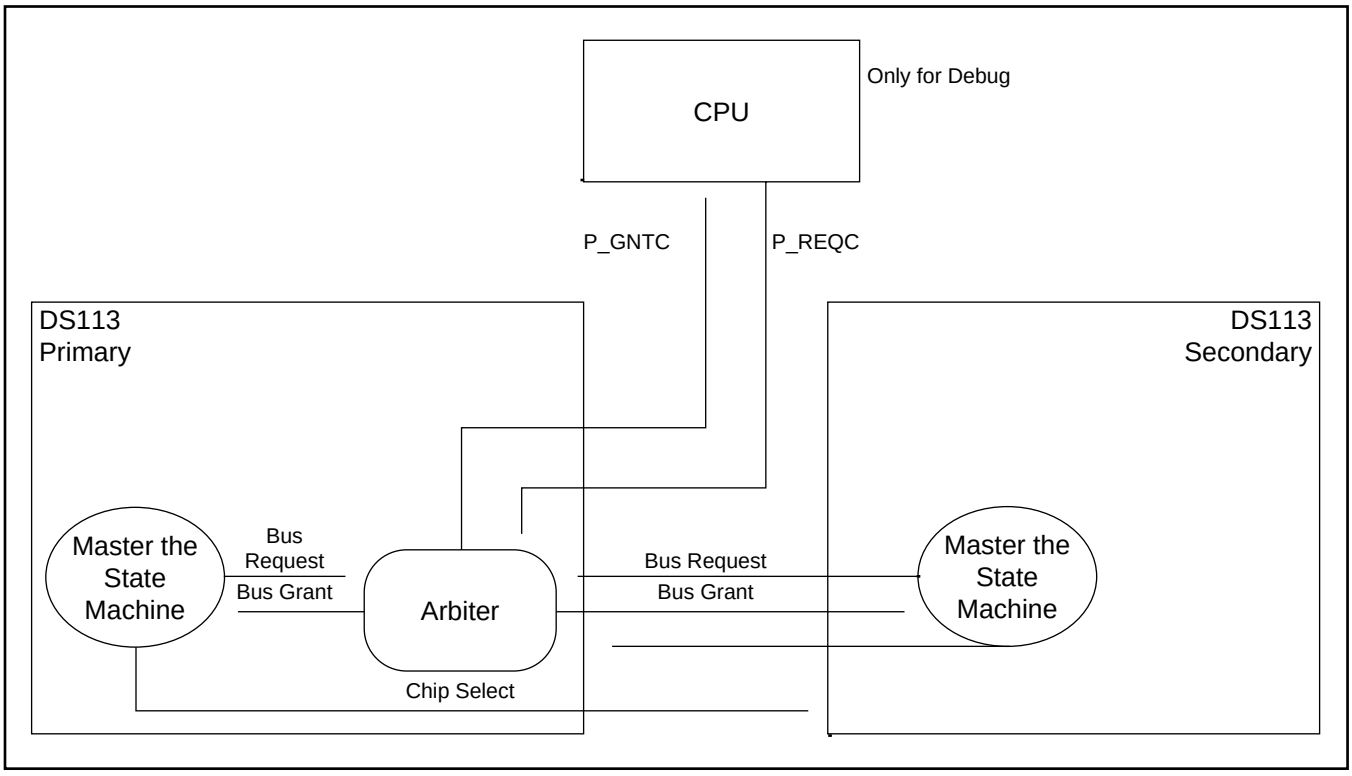


Figure 16 - Block Diagram of the Arbiter

Control Bus Timing

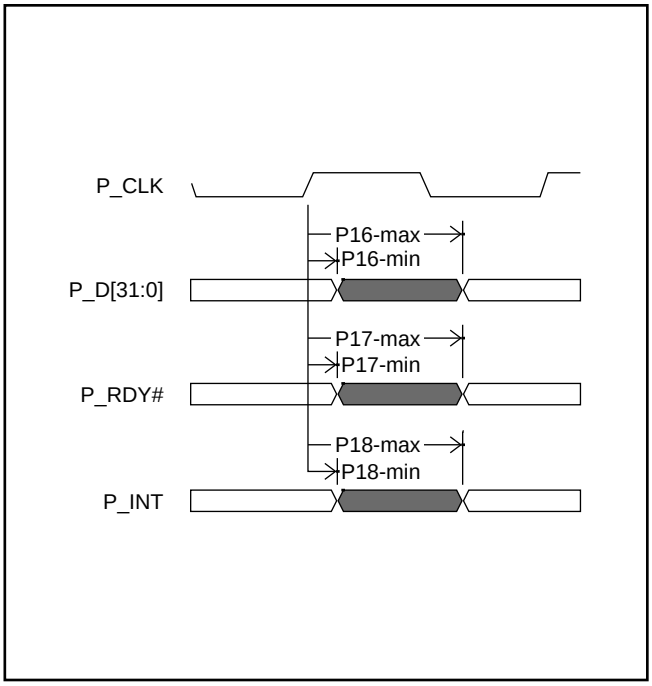


Figure 17 - Control Bus - Output Valid Delay Timing

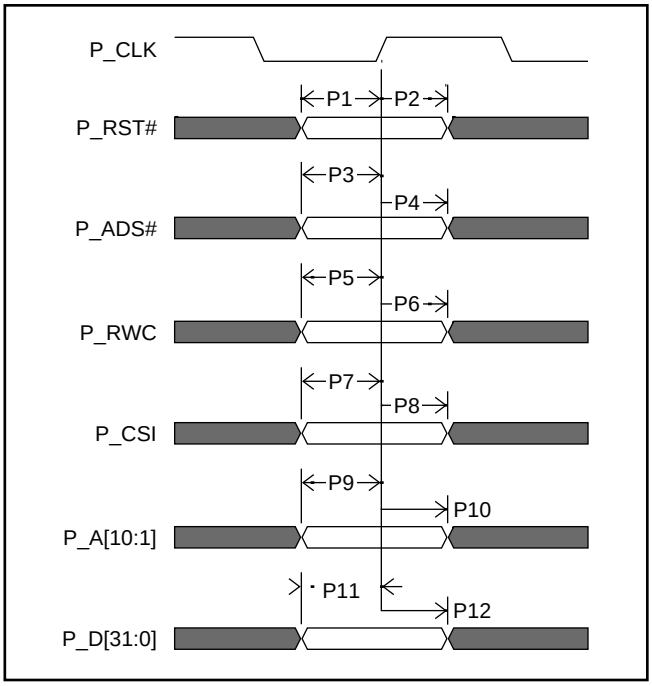


Figure 18 - Control Bus - Input Setup and Hold Timing

13 The LED Interface

LED Interface

The MDS113CG LED interface supports the status per port in a serial stream that may be daisy-chained to connect two MDS113CG chips. Daisy-chaining greatly reduces the pin count and number of board traces routed from the Physical Layer to the LEDs, thus simplifying system design and reducing overall system cost. For a large port configuration such as the 24+2 in the MDS113CG, a large number of LED signals is needed, which may induce noise and layout issues in the system. The LED information is transmitted in a frame-structured format with a synchronization pulse at the start of each frame.

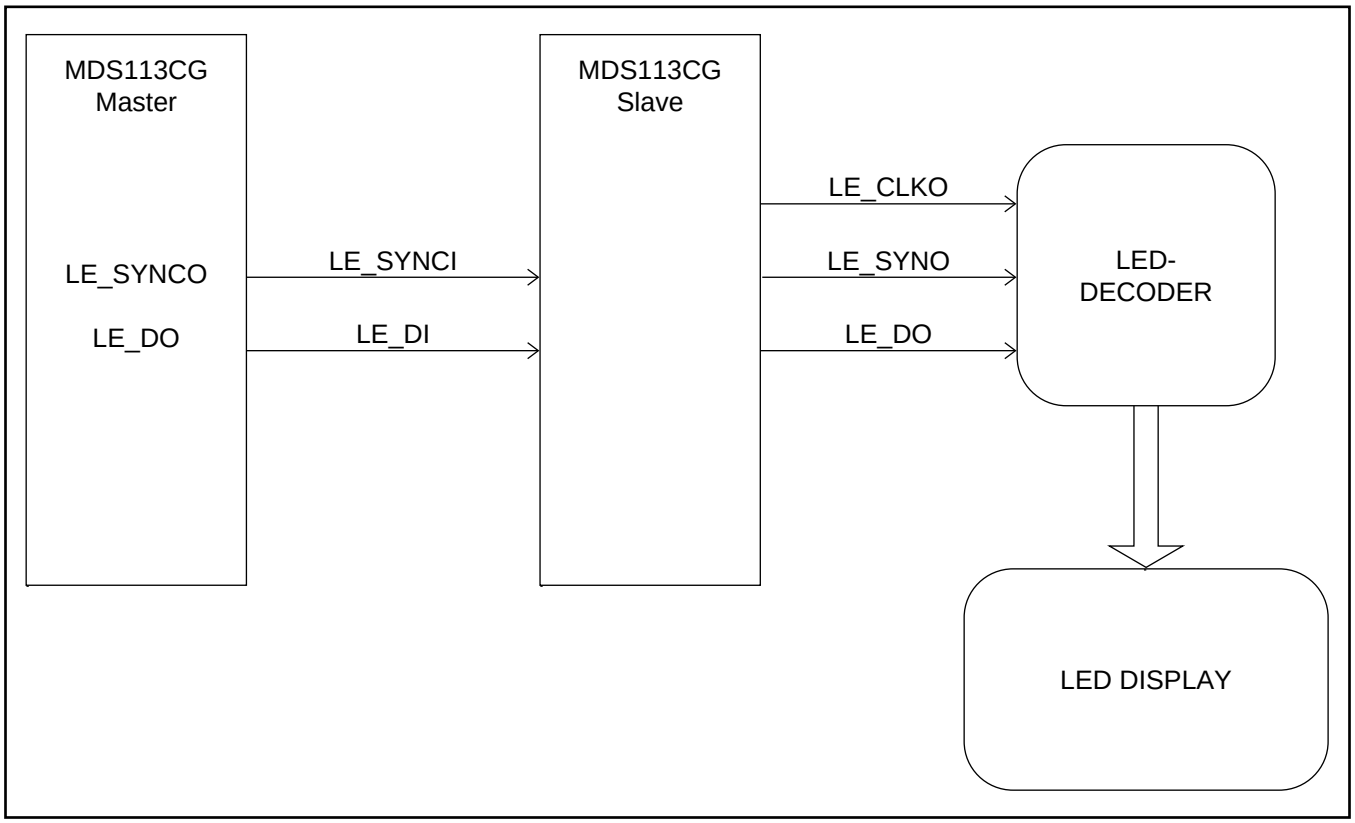


Figure 19 - LED Interface Connections

To provide the port status information from our MDS113CG chips via a serial output channel, six additional pins are required.

-
- LE_CLKI/O at 25 MHz
- LE_SYNI/O a sync pulse — defines the boundary between frames
- LE_DI/O a continuous serial stream of data for all status LEDs which repeat once every frame time

A low cost external device (i.e. a 44-pin FPGA-like device) decodes the LED framed data and drives the LED array for display. This device may be customized for different system configurations.

The port status of the MDS113CG is transmitted to an external decoder via a serial output channel. In the MDS113CG, we support cascading of this serial output channel between two devices. One MDS113CG is configured as the master; this initiates the start of LED information frames, and serializes information bits. The MDS113CG slave repeats the information sent from the master and appends its own information bits. To cascade these two devices, we will need to extend the number of LED pins from 3 to 6. The following table shows two cascaded LED interfaces and the connections between the MDS113CGs, the LED decoder, and the LED display.

Function Description

Signal Name		Description
Master Device	**Slave Device**	
LE_CLKI	LE_CLKO	LED Clock-Synchronous LED clock provided by the slave device to LED decoder at the system clock divided by 8 (~97.5Khz)
LE_SYNI	LE_SYNO	A synchronous pulse — defines the boundary between frames The length of each LED data frame is about 256 bits that shift out by LED_CLK per bit
LE_DI	LE_DO	A continuous serial stream of data for all status LEDs, which repeats once every frame time

Table 7 - LED Signal Names and Descriptions

Port Status

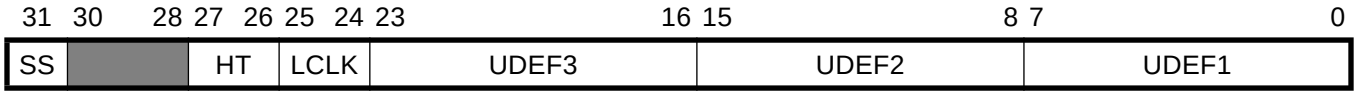
In the MDS113CG, each port consists of 8 different LED status, represented by separate bits:

1. Flow Control
2. Transmitting Data
3. Receiving Data
4. Action (TxD or RxD)
5. Link UP/DOWN
6. Speed
7. Full-Duplex/Half-Duplex
8. Collision.

In addition to the 13 ports of the MDS113CG, three extra user-defined status sets may be sent through the LED serial channel for debugging or other applications, where each user-defined status set is also represented by 8 bits.

LED Interface Time Diagram

The Master needs to shift out (13+3)*8 status bits periodically. Thus, slave needs to shift out (13+3)*8 + (13+3)*8 status bits, which includes the status of the master device and itself.



Bit [7:0]	UDEF1	User defined information status 1 for debug purpose
Bit [15:8]	UDEF2	User defined information status 2 for debug purpose
Bit [23:16]	UDEF3	User defined information status 3 for debug purpose
Bit [25:24]	LCLK	LED Clock frequency (Default=00) 00=100M/8=12.5Mhz 01=100M/16=6.25Mhz 10=100M/32=3.125Mhz 11=100M/64=1.5625Mhz
Bit [27:26]	HT	Holding time for LED signal (Default=00) 00=8msec 01=16msec 10=32msec11=64msec
Bit [30:28]	Reserve	
Bit [31]	SS	Start Shift out the status bits out from the master device. This bit has no effect on slave chip.

Note that UDEF1-UDEF3 are used for debug purpose. The contents of UDEF1-3 are loaded by CPU and the usage of these are up to software.

The status of each port will be sampled by the LED State Machine every 20.5 ms, the period of the frame. That is, each LED data frame length equals 256 * 80 nsec. Each frame is divided into two sub-frames: a master and a slave sub-frame. Furthermore, each sub-frame is partitioned into 16 slots (13 MAC ports plus 3 user-defined sets) and each slot will carry 8 status bits. The following figure shows the signal from the slave chip to LED decoder.

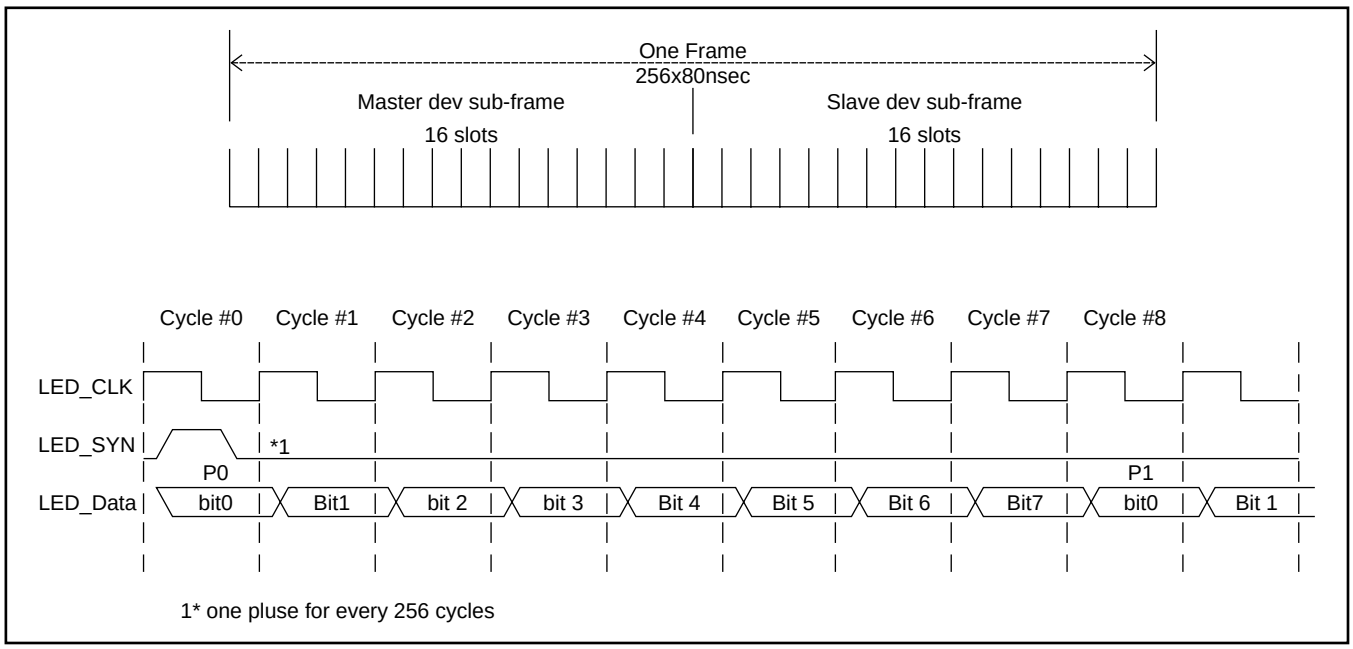


Figure 20 - Time Diagram of LED Interface

14 Data Forwarding Protocol and Data Flow

Frame Reception

For normal frame reception, a 128-byte block of frame data is stored in the RxFIFO. This block may be shorter if an End of Frame (EOF) arrives. At that point, the RxDMA will request the use of the internal memory bus. When this memory request is granted, the RxDMA will move the block from the RxFIFO to the Frame Data Buffer (FDB).

The MAC ports are partitioned into two groups, one for the Gbps Port and one for all 12 of the 100Mbps Ports. The service discipline is round robin for both the Gbps Port and 100/10Mbps group. After the entire frame is moved to the frame data buffer (FDB), a switch request will be sent to the Search Engine (Reference Search Engine Section)

Unicast Frame Forwarding

For forwarding of the unicast frame, the Search Engine first resolves the destination device and the destination port, and sends a switch response back to the Frame Engine. The Frame Engine will obtain the type (unicast or multicast), the destination port, and the destination device from the search response. After processing the search response, the Frame Engine will notify the destination port that it has a frame to forward to the destination port's TxFIFO.

For local forwarding (e.g. the destination port is in the local device), the Frame Engine will send the job to the Transmission Scheduling queue of the destination port.

For remote forwarding (i.e. the destination port is in the remote device), the Frame Engine will create a data forwarding request command message (DATA_FWD_REQ), which is sent via the XPipe to the remote device. The remote Frame Engine, after receiving this DATA_FWD_REQ message, will place a job in the Transmission Scheduling queue of the destination port.

The port will serve the next job from the Transmission Scheduling queue when the following two conditions are met:

- There is enough room for a 1.5Kbyte frame (a maximum-sized frame) within the TxFIFO.
- The end-of-frame (EOF) of the current frame has arrived at the TxFIFO.

The port will send the jobs to the transmission scheduling queues according to a first in first out (FIFO) order.

To start data transmission, the port obtains a job from the transmission scheduling queue and notifies the Transmit DMA (TxDMA) to move the data from the FDB to the MAC Transmit FIFO (TxFIFO) in 128-byte granules (for local forwarding). Otherwise, the device sends a DATA_REQ command message via the XPipe to the source device to request remote forwarding. The data forwarding engine module in the Frame Engine of the source device will then forward the frame in 128-byte granules via the XPipe.

Flow for Data Frame

The following subsections describe the flow of information during transfers of unicast data frames.

Unicast Data Frame to Local Device

In the simplest case, the data frame is destined for a port on the local device. The Frame Engine moves the received frame to the local FDB. The Search Engine forms a switch request with the frame header (includes source MAC and Destination MAC) and passes it to the Switch Engine to resolve the destination. The Switch Engine then provides a destination port address to the Frame Engine via a switch response message. The Frame Engine transmits to put a transmission job in transmission scheduling. After the port is ready to send the frame, the Frame Engine starts to move the frame to the TxFIFO. If the Switch Engine cannot resolve the MAC address, the HISC is queried to resolve the address.

Unicast Data Frame to Remote Device

In another case, the data frame is destined for a port on a remote device. First, the Frame Engine moves the received frame to the local FDB. A switch request with a frame header (includes source MAC and Destination MAC) is passed to the Switch Engine to resolve the destination. The Switch Engine then provides a destination port address to the Frame Engine. If the Switch Engine cannot resolve the address, the HISC is queried. Once the address is resolved, the two Frame Engines perform the following interactive handshaking procedures via the XPipe:

- Source Frame Engine sends a Data Forwarding Request message to Destination, where the destination Frame Engine puts a job in the associated transmission scheduling queue.
- When the destination port is ready to send the frame, the destination Frame Engine send a Data Request message to the source Frame Engine.
- After the source Frame Engine receives the Data Request Message, it starts to move the frame in granule form, which is directly written in the destination TxFIFO.

Note that, at the remote device, the frame is written into the transmit FIFO of the remote destination port. In order to reduce the latency, the frame is not stored in the FDB of the remote device again.

15 Port Trunking

Port trunking groups a set of 8 MDS113CG 10/100 Mbps physical ports into one logical link; however, all ports in the trunk group must be within the same access device, and each port can only belong to one trunk group. All ports in the Trunk group must and share the same MAC Address. Each system can support up to 4 groups.

Load distribution for unicast traffic is done based on a hash key, a hash function of the Source Address and the Destination Address.

Note: Refer to the "MDS113CG/DS113 Port Trunking and Port Mirroring Application Note." In this document, we describe how to specify the trunk groups on line via DIP switches.

Unicast Packet Forwarding

ECR1 - MAC Port Configuration Register

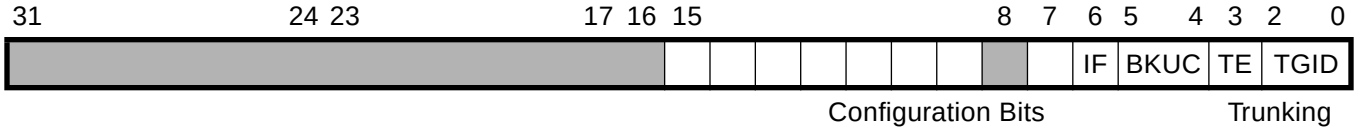


Figure 21 - ECR1 - MAC Port Configuration Register

Port Trunking ID Bits

Bit [0:2] TGID Group ID

Bit [3] TE Trunk Enable 0= Trunk disable 1= Trunk Enable

A trunked port will need to have its ECR1 MAC Port Configuration Register set by CPU software to contain its associated Trunk Group ID. Later on, when a new source MAC Address is learned through that port, the Trunk Group ID will be recorded in the MCT entry by either the Search Engine or the microcode in the HISC. The Trunk Group ID will be used for forwarding decision when the destination MCT entry of a received packet is found by the Search Engine, if the status field indicates that the address found is on a Trunk Group.

The Trunk Group ID is used by the Search Engine, along with the "hash key" (3 bits result of a hash operation between source address and destination MAC address), to access a Trunk Port Mapping Table entry in the internal RAM. Each entry in this table contains the device and port IDs for the physical port used to transmit this packet. Software needs to set these entries, using TPMXR and TPMTD registers, to distribute the traffic load across the ports in the Trunk Group.

If the source MAC Address of an incoming packet is on a Trunk Group (based on the MCT information), the receiving port's TGID will be compared against the Trunk Group ID in the source MCT to decide whether the source MAC address has moved to another Trunk Group or not.

The Trunk Port Mapping Table is 32 entries deep (4 groups * 8 hash entries), and each entry is 5 bits wide (1-bit device ID, 4-bit port ID), as show in the following format.

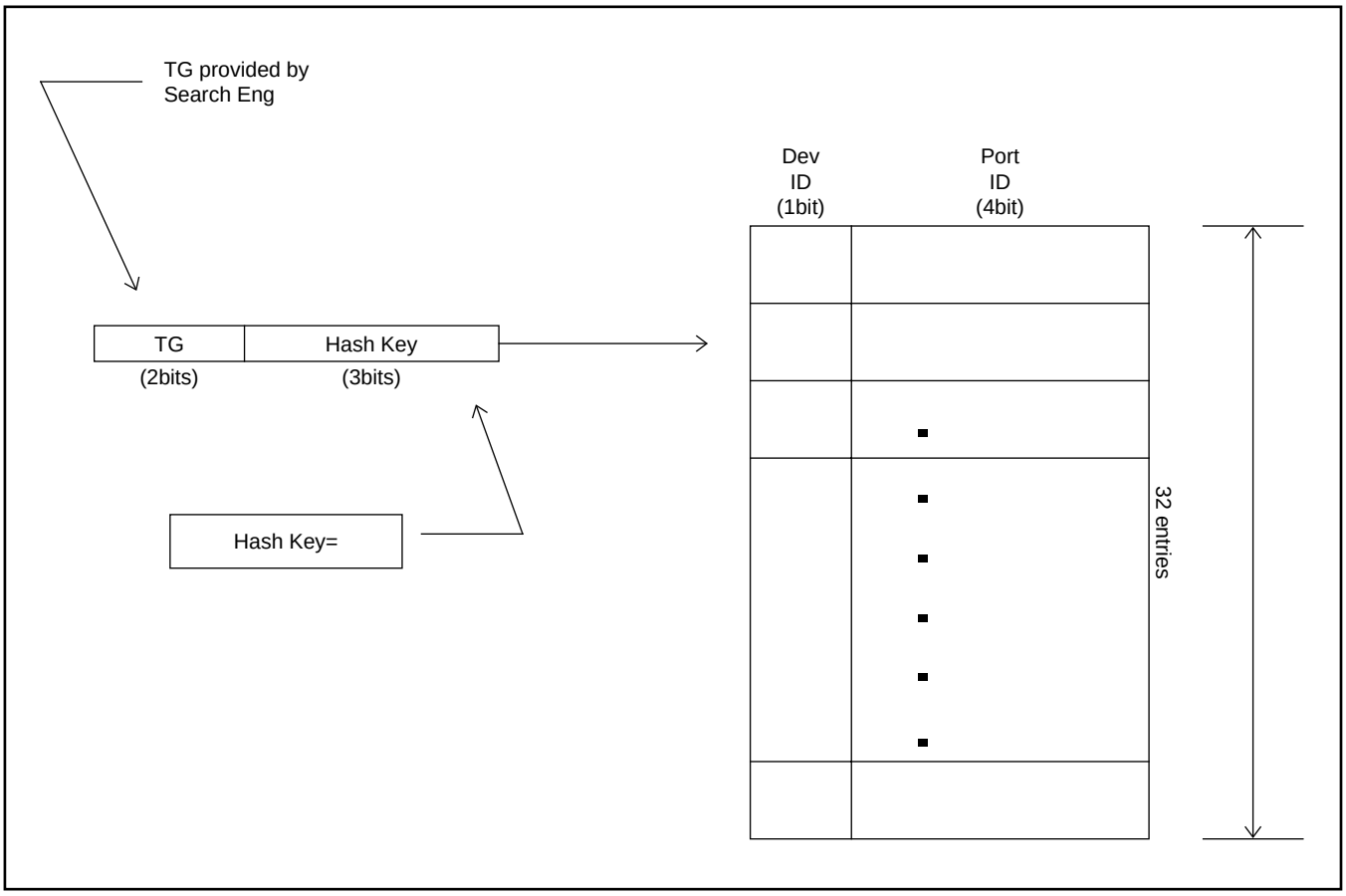


Figure 22 - Port Mapping Table

MAC Address Assignment

In MDS113CG, there are three ways to assign the MAC address to each port. All the ports in the same device share the 44 MSBs, MAC[47:4], which are shown in ADAR0 and ADAR1 registers, while the 4 LSBs, MAC[3:0] are specified in ADOR0 and ADOR1 registers for port 0-port 6 and port 8-port 11, respectively. The 4 LSBs MAC[3:0] can be assigned as follows:

1. If the switch does not support Port Trunking, MAC[3:0]= port number

2. If the switch supports multiple MAC addresses and Port Trunking, the ports in the same Trunk Group share the same MAC[3:0]. The value of MAC[3:0] is assigned by the Trunk Group (TG) Table.

3. If the switch supports only a single MAC address, all the 4 LSBs of MAC will be set the same value in ADOR0 and ADOR1 register.

16 Port Mirroring

Features

The received or transmitted data of any 10/100 port in any MDS113CG chip, connected by Port Mirror signal pins, PM_DO and PM_DI, can be chosen to be mirrored to the "Mirror Port." The mirror port can be the first port in a MDS113CG with RMII or a dedicated mirror port with MII, driven by the pin, PM_DO[0:1]. Once the first RMII port of a chip is selected to be the mirror port, it cannot be used to serve as a data port. The configuration of port mirroring is shown in the following diagram, based on the current evaluation board design.

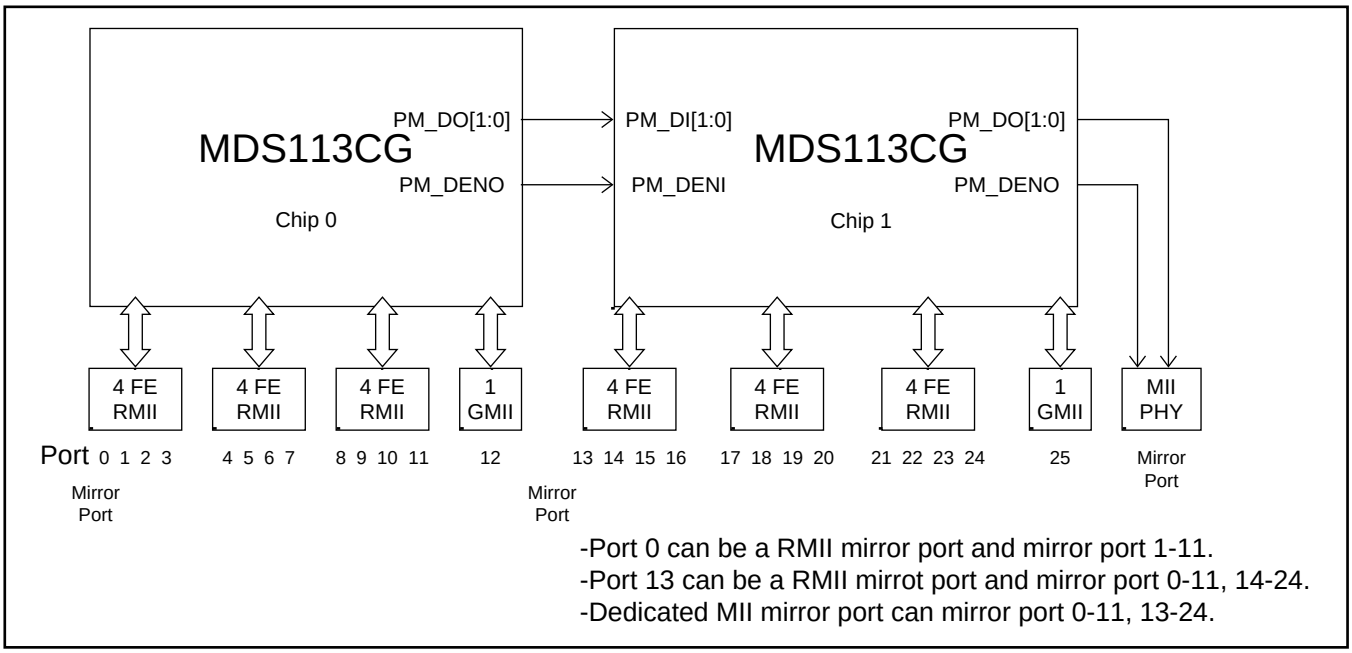


Figure 23 - Configuration of Mirror Port for MDS113CG

Physical Pins

There are 6 related pins to Port Mirroring functions:

PM_DI [1:0] Port Mirroring Input Data Bit [1:0]
Receive the mirrored data signal from the remote MDS113CG.

PM_DENI Port Mirroring Data Enable signal for PM_DI Input
Provide Data Enable signal for PM_DI signals

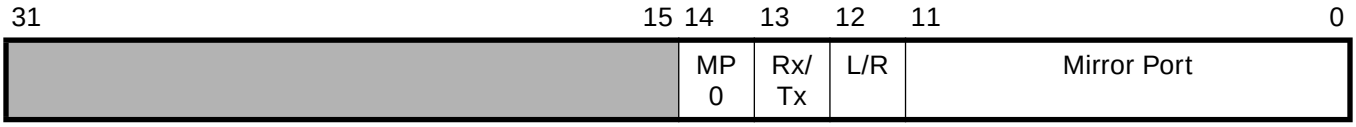
PM_DO[1:0] Port Mirroring Output Data Bit [1:0]
Transmit the mirrored data signal to remote MDS113CG.

PM_DENO Port Mirroring Data Enable Output.
Provide Data Enable signal for PM_DO signals

Refer to the figure on this page for connecting above pins.

Setting Register for Port Mirroring

APMR register controls the mirrored port and the designated mirroring port. The definition of the register is shown as follows:



Bit [11:0]	Mirr_Port	10/100 port is chosen to be mirrored, (port bit map)
Bit [12]	Local/Remote	Indicate the mirrored port from local or remote device. 0=local 1=remote (Note: Not support 1G port Mirroring.)

Note that at most one of bits in Bit[12:0] can be set to 1.

Bit [13]	Chose_rx	Whether mirror receiving data or transmitting data 0= Transmission Mirroring, 1=Receiving Mirroring
Bit [14]	MP0	Mirror to Port 0 (Default=0) MP0=1 Mirror to port 0 MP0=0 Mirror not go to port 0. I.e., to PM_DO pins.
Bit [31:14]	Reserve	

The following examples, based on the configuration of the figure on the previous page, illustrate how to set the register:

Example 1: Mirroring port 1 to port 0 and Mirror transmission direction.

For Chip 0

Set APMR[11:0]=0x002	Mirrored port=1
Set APMR[12]=0	Local mirrored port
Set APMR[13]=0	Transmission mirroring
Set APMR[14]=1	Port 0 is the mirroring port

For Chip 1:

Don't Care

Example 2: Mirroring port 1 to port 12 and Mirror receiving direction.

For Chip 0

Set APMR[11:0]= 0x002	Mirrored port= 1
Set APMR[12]=0	Local mirrored port
Set APMR[13]=1	Receiving mirroring
Set APMR[14]=0	Port 0 is not the mirroring port

For Chip 1:

Set APMR[11:0]=0x000	
Set APMR[12]=1	Remote mirrored port
Set APMR[13]=Don't care	Bit[13] has meaning only in the chip of mirrored port
Set APMR[14]=1	Port 13 is the mirroring port

Example 3: Mirroring port 1 to MII Mirroring port Mirror receiving direction.

For Chip 0

Set APMR[11:0]= 0x002	Mirrored port= 1
Set APMR[12]=0	Local mirrored port
Set APMR[13]=1	Receiving mirroring
Set APMR[14]=0	Port 0 is not the mirroring port

For Chip 1:

Set APMR[11:0]= 0x000	
Set APMR[12]=1	Remote mirrored port
Set APMR[13]= Don't care	Bit[13] has meaning only in the chip of mirrored port
Set APMR[14]=0	Port 13 is not the mirroring port

Note: Refer to “MDS113CG/MDS113CG Port Trunking and Port Mirroring Application Note”. This document describes how to programming the port mirroring register on line via DIP switches.

17 Register Definitions

Register Map

All registers are grouped into sets.

- DEVICE CONFIGURATION
- BUFFER MEMORY INTERFACE
- FRAME CONTROL BUFFER
- SWITCHING CONTROL
- ACCESS CONTROL FUNCTIONS
- MAC PORT CONTROL

Access Control:

- W/R = These register bits may be read from and written to by software
- W/-- = These register bits may be written to by software, but not read. Write Only
- (--/R) = These register bits may be read but not written to by software. Read Only
- Latched and held bits
- Clear bits
- Permanently set bits

All registers are 32-bit wide. They are classified in the following tables:

Tag	Description	ADDRESS	W/R
1. Device Configuration Registers (DCR)			
GCR	Global Control Register	7C0	W/--
DCR0	Device Status Register	7C0	--/R
DCR1	Signature & Revision & ID Register	7C4	W/R
DCR2	Device Configuration Register	7C8	W/R
2. Buffer Memory Interface			
MBCR	Multicast Buffer Control Register	79C	W/R
Reserve	Must Set to “0x0001 0008”	7B8	W/R
Reserve	Must Set to “0x0001 0000”	7BC	W/R
3. Frame Control Buffers Management			
FCBSL	FCB Stack Size Limit	740	W/R
FCBST	Frame Ctrl Buffer Stack – Buffer Low Threshold	744	W/R
BCT	Buffer Counter Threshold	74C	W/R
BCHL	Buffer Counter Hi-Low Selection	750	W/R
4. Switching Control			
FCR	Flooding Control Register	6DC	W/R
MCAT	MCT Aging Timer	6E0	W/R
PTR	Pacing Time Regulation	6EC	W/R

Table 8 - MDS113CG Register Map (1 of 2)

Tag	Description	ADDRESS	W/R
5. Access Control Function Group 1 (Chip Level controls)			
ATTL	Transmission Timing & Threshold Control Register	650	W/R
AFCR	Flow Control Register	670	W/R
AMCT	MAC Control Frame Type Code Register	67C	W/R
ADAR0	Base MAC Address Register – Byte[3,0]	600	W/R
ADAR1	Base MAC Address Register – Byte[5,4]	604	W/R
ADAOR0	MAC Offset Address Register Port[0:7]	608	W/R
ADAOR1	MAC Offset Address Register Port[12:8]	60C	W/R
ACKTM	Timer For SOF Check	610	W/R
AFCOFT10	Flow Control Off Time for 10 port	614	W/R
AFCOFT100	Flow Control Off Time for 100 port	618	W/R
AFCOFT1000	Flow Control Off Time for Gig port	61C	W/R
AFCHT10	Flow Control Holding Time for 10 port	620	W/R
AFCHT100	Flow Control Holding Time for 100 port	624	W/R
AFCHT1000	Flow Control Holding Time for Gig port	628	W/R
6. Access Control Function Group 2 (Chip Level controls)			
APMR	Port Mirroring Register	5C0	W/R
THKM0	Trunking Forward Port Mask 0 (hash key=0)	5C8	W/R
THKM1	Trunking Forward Port Mask 1 (hash key=1)	5CC	W/R
THKM2	Trunking Forward Port Mask 2 (hash key=2)	5D0	W/R
THKM3	Trunking Forward Port Mask 3 (hash key=3)	5D4	W/R
THKM4	Trunking Forward Port Mask 4 (hash key=4)	5D8	W/R
THKM5	Trunking Forward Port Mask 5 (hash key=5)	5DC	W/R
THKM6	Trunking Forward Port Mask 6 (hash key=6)	5E0	W/R
THKM7	Trunking Forward Port Mask 7 (hash key=7)	5E4	W/R
LEDR	LED Register	598	W/R
7. Ethernet MAC Port Control Registers – (substitute [N] with Port Number, N = {0..12})			
ECR1	MAC Port Configuration Register	[N*4]4	W/R

Register Definitions

Device Configuration Register

GCR - Global Control Register

- Access: Zero-Wait-State,Direct Access,Write only
- Address: h7C0

Bit[2:0] Op-Code 3-bit Operation Control Code

Op-Code	Command	Description
000	Clr RST	**Clear Device Reset:** — Allows state machines to exit from RESET state and to initialize their internal control parameters if necessary.
001	RESET	**Device Reset:** — Resets all internal state machines of each device and stays in RESET state (except the Processor Bus Interface logic)
010	EXEC	**Execution:** — Allows state machines to start their normal operations.
011	--	No-Op
1XX	--	No-Op

Bit[31:4] Reserved

DCR0 - Device Status Register

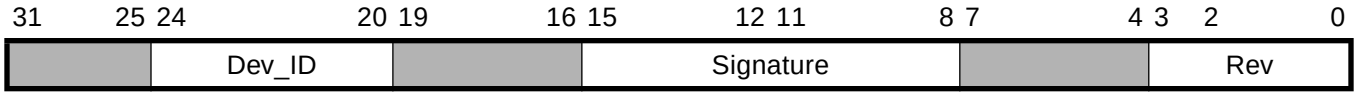
- Access: Zero-Wait-State,Direct Access,Read only
- Address: h7C0

Bit[1:0] Status 2-bit Device Operation Status Code

Status	State	Description
01	RESET	Device Reset: — Device is in RESET state
10	EXEC	Execution: — Device is under normal operation

DCR1 - Signature, Revision, & ID Register

- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h7C4



Bit[3:0] Device Revision Code

Bit[7:4] Reserved

Bit[15:8] Signature 8-bit Device Signature

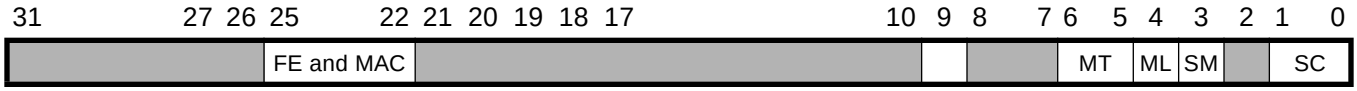
Bit[19:16] Reserved

Bit[24:20] DEV_ID 5-bit Device ID (Read/Write)

Bit[31:25] Reserved

DCR2 - Device Configuration Register

- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h7C8



Bit[1:0] SC System Clock Rate Default = 00

00= 100Mhz 01 = 120Mhz

10=90Mhz 11= 80Mhz

Bit[2] Reserved

Bit[3] SM System Configuration mode

0=Nonblocking (For MDS113CG, Always equal to 0)

1=Blocking

SRAM Memory Characteristics

Bit[4] ML Buffer Memory Level, which can be either 2 chips or 4 chips.

0 = 2 memory chips (one bank) Default = 0

1 = 4 memory chips (two banks)

Bit[6:5] MT Memory Chip Type Default = 01

00 = 64K x 32-bit 01 = 128K x 32-bit

10 = 256K x 32-bit 11 = 512K x 32-bit

Bit[8:7] Reserved

Search Engine Configuration

Bit[9]	SE_AGEN	Aging enable, if which is true, the old MCT can be aged out. Default = 1 0 = disable aging 1 = enable aging

Frame Engine and MAC Configuration

Bit[21:10]	Reserved	
Bit[22]	BC_EN	Buffer counter enable 0 = Disable (no head of line control 1 = enable
Bit[23]	Reserved	
Bit[24]	SEL_PCS	Default =0 0 = Use external PCS 1 = Use internal PCS in the chip
Bit[25]	Link_GT	TX LED will be off when the link is down and this bit is 0 Default =0 0 = Gate 0ff TX_En when Link down 1 = Not Gate off TX_En when Link down
Bit[31:26]	Reserved	

Buffer Memory Interface Register

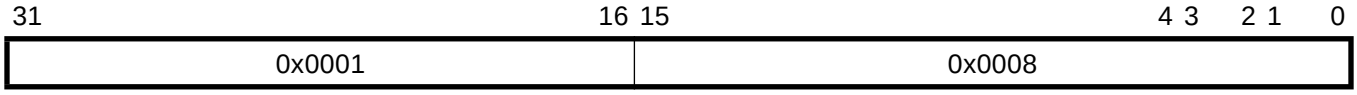
MBCR- Multicast Buffer Control Register

- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h79C

Bit[4:0]	MAX_MC_FD	Maximum Number of Multicast Frames allowed for forwarding.
Bit[10:5]	RMC_BUF_RSV	Number of buffers reserved for receiving remote Multicast Frames.
Bit[19:11]	MAX_CNT_LMT	Maximum Number of Multicast Frames allowed per device
Bit[31:20]	Reserved	

Reserve Register 1

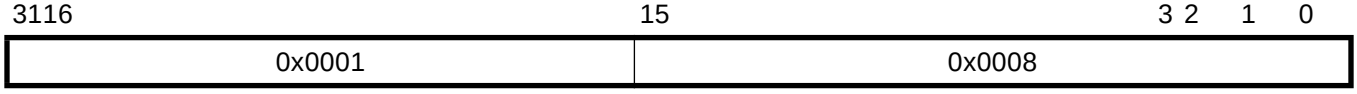
- Access Non-Zero-Wait-StateDirect-AccessWrite/Read
- Address: h7B8



MUST BE SET TO “0X00010008”

Reserve Register 2

- Access Non-Zero-Wait-StateDirect-AccessWrite/Read
- Address: h7BC
-

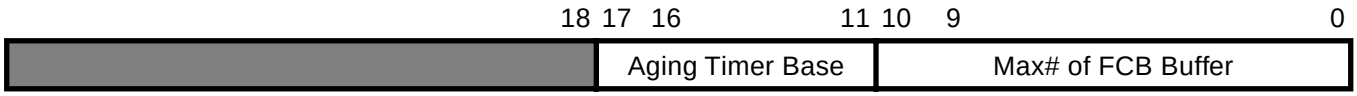


MUST BE SET TO “0X00010000”

Frame Control Buffers Management Register

FCBSL - FCB QUEUE

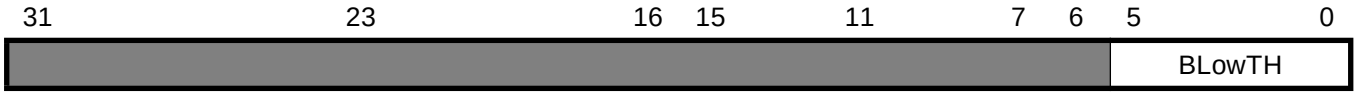
- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h740



Bit[10:0]		Defines Max # of FCB Buffers Size Range: 1 entry, to 1024 entries
Bit[17:11]	Aging Timer Base	Defines the time interval between scanning of FCB Buffers for aged buffers Aging Time = (Number of valid FCB Buffers* Aging Timer Base) msec

FCBST - FCB Queue - Buffer Low Threshold

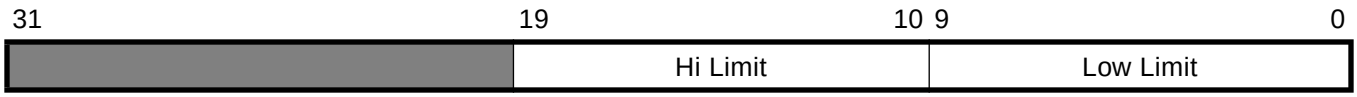
- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h744



Bit[5:0]	Buf_Low_Th	Buffer Low Threshold – The number of frame control buffer handles left in the Queue to be considered as running low and trigger the interrupt to the HISC.
Bit[31:6]	Reserved	

BCT - (FCB) Buffer Counter Threshold

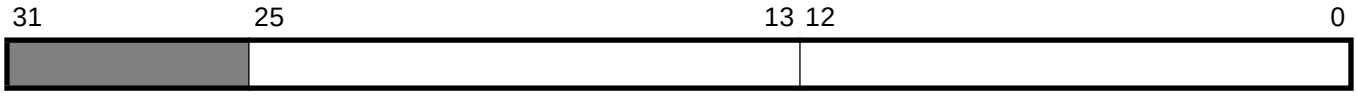
- Access Non-Zero-Wait-StateDirect-AccessWrite/Read
- Address: h74C



Bit[9:0]	Low_Limit	Low limit number of frames to each destination port (i.e., Source port limits the # of FCB used by each destination port)
Bit[10:19]	HI_Limit	High limit number of frames to each destination port (i.e., Source port limits the # of FCB used by each destination port)

BCHL - Buffer Control Hi-Low Selection

- Access Non-Zero-Wait-StateDirect-AccessWrite/Read
- Address: h750

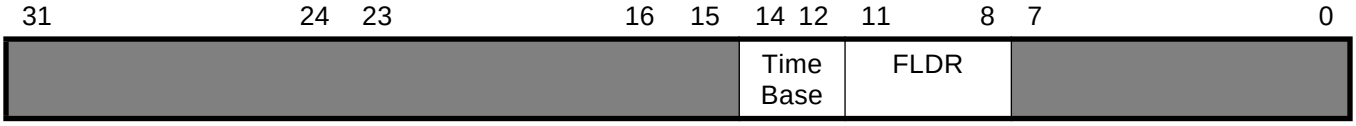


Bit[12:0]	Lp_Hi_Low Sel	Selection for Low or High Limit of Buffer Counter for Local device 13 bits maps to 13 ports in Local Device 1 = select hi limit 0 = select low limit
Bit[25:13]	Rp_Hi_Low Sel	Selection for Low or High Limit of Buffer Counter for Remote device 13 bits maps to 13 ports in Remote Device 1 = select hi limit 0 = select low limit

Switching Control Register

Flooding Control Register

- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h6DC



Bit[0:7] Reserved

Bit[11:8] FLDR Flooding Rate
Restricts the number of flooding unicast frames within the Time window

Bit[14:12]Time Base Defines the time window used by FLDR

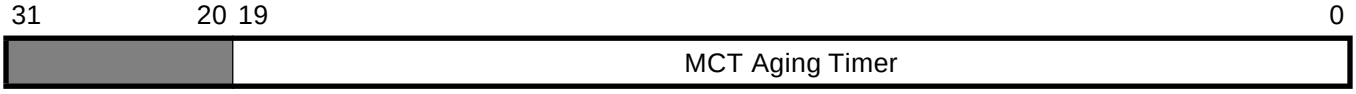
000 = 100us 001 = 200us 010 = 400us 011 = 800us

100 = 1.6ms 101 = 3.2ms 110 = 6.4ms 111 = 100us

Bit[31:15] Reserved

MCAT- MCT Aging Timer

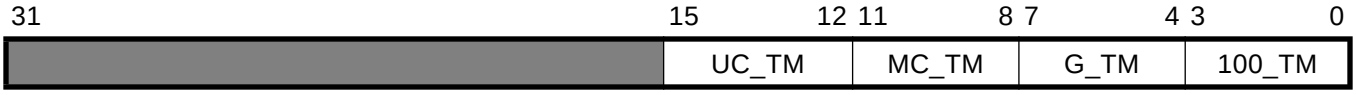
- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h6E0



Bit[19:0] When the value is reached, it ages out
Default=0 msec (unit=msec) Must be configured to not zero value.
Suggestion value: 5msec.

PTR - Pacing Time Regulation

- Access Non-Zero-Wait-StateDirect-AccessWrite/Read
- Address: h6EC
- Use for Pacing traffic to Remote Ports via XpressFlow Pipe or Local transmission



Bit[3:0] 100_TM 100M port timer Default =5

Bit[7:4] g_TM Gigabit port timer Default =6

Bit[11:8] mc_TM Multicast timer Default =5

Bit[15:12] uc_TM Unicast timer Default =5

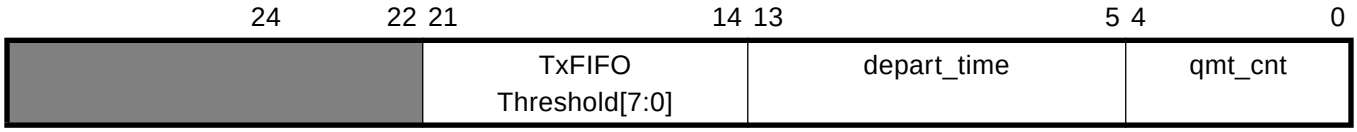
Unit time = 80 nsec (for 64Bytes Frame).

Note that Frame Engine determine the tic value dependent upon the frame. If short frame, it takes above value. For long frame (> 64 frame), it will double the above value as the reference.

Access Control Function

ATTL - Transmission Timing Control

- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h650

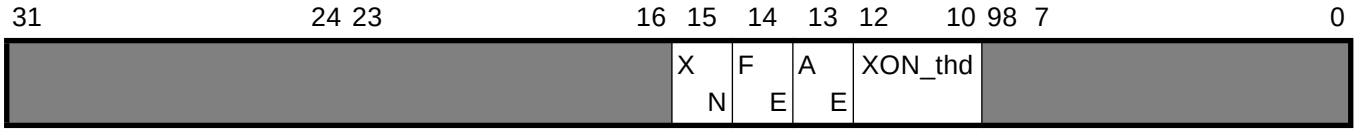


Bit[4:0]		Transmission queue aging time out counter
Bit[13:5]		frame latest departure time
Bit[21:14]	TXFIFOT	Transmission FIFO Threshold in Bytes (Default =0) Only for 100M ports Unit=8Bytes 0= Cut Through at the destination 100M port

When the value does not equal zero, it indicates the port cannot start sending frames out, until the TxFIFO reaches the threshold or EOF.

AFCR - Flow Control Register

- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h670



Bit[9:0] Reserved

Bit[12:10] XON_Thd Defines the minimum # of free Frame Buffers before transmitting XON flow control frame.

$$XON\_Thd = \left\lceil \frac{min.\#offreeFCB}{8} \right\rceil$$

Bit[13] Queue Aging EnableTX queue aging function enable

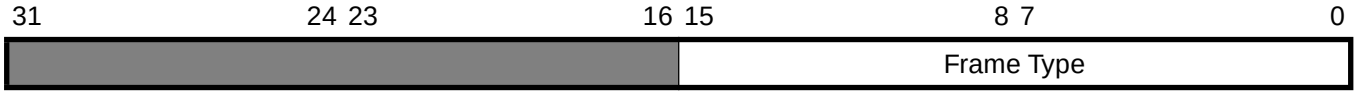
Bit[14] Flush EnableWhen stack is full, enable flush procession
0 = disable 1 = enable

Bit[15] XON EnableFull Duplex XON enable
0 = disable 1 = enable

Bit[31:16] Reserved

AMCT - MAC Control Frame Type Code Register

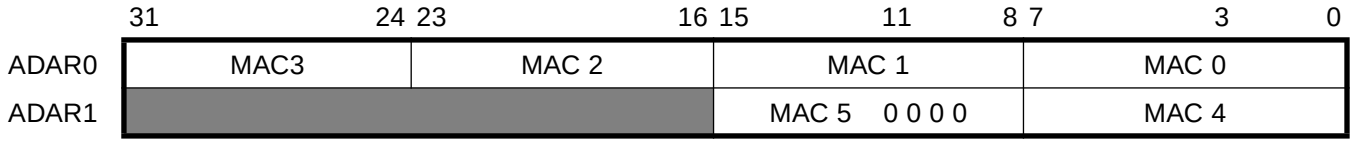
- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h67C



- 2-byte MAC Control Frame Type Code defined by IEEE 802.3X Full Duplex Flow Control Standard

ADAR[1:0] - Base MAC Address Registers

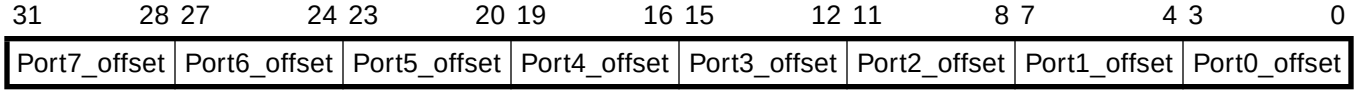
- The 6-byte MAC Address is stored in two 32-bit registers
  - ADAR0 MAC Address Byte[3:0]
  - Address: h600
  - ADAR1 MAC Address Byte[5:4]
  - Address: h604
- Access: Non-Zero-Wait-State,Direct Access,Write/Read



- These two registers define the base MAC address of the device.
- Bit[3:0] of Byte 0 is always set to 0.
- MAC address for each port is defined by
  - MAC Address for Port n = Base MAC Address + MAC Offset[n] where n = {0..12}
  - MAC Offset[n] is defined by the following registers

ADAOR0 - MAC Offset Address Register 0

- MAC Offset Address for Port[7:0], 4-bit per port
- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h608

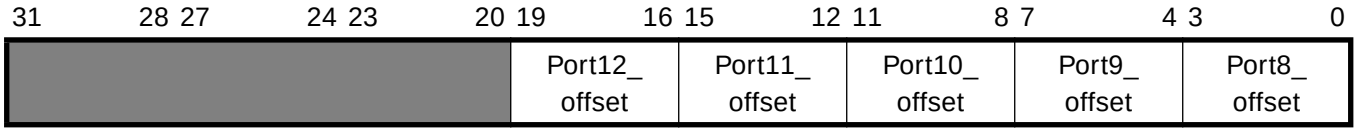


Bit[3:0] MAC Offset address for Port 0
Bit[7:4] MAC Offset address for Port 1
Bit[11:8] MAC Offset address for Port 2
Bit[15:12] MAC Offset address for Port 3
Bit[19:16] MAC Offset address for Port 4
Bit[23:20] MAC Offset address for Port 5
Bit[27:24] MAC Offset address for Port 6
Bit[31:28] MAC Offset address for Port 7

Usage: All ports in the same device share the 44 MSBs, MAC[47:4] in ADAR[0:1], while the 4 LSBs, MAC Offset[3:0] can be assigned as follows: If the device supports port-trunking, the ports in the same trunk group share the same MAC[3:0]. The value of MAC[3:0] is assigned by the smallest port nubmer in the Trunk Group. Otherwise, MAC[3:0] is fixed for all devices (i.e. only one MAC[3:0] address for the whole system).

ADAOR1- MAC Offset Address Register 1

- MAC Offset Address for Port[12:8], 4-bit per port
- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h60C



Bit[3:0] MAC Offset address for Port 8
Bit[7:4] MAC Offset address for Port 9
Bit[11:8] MAC Offset address for Port 10
Bit[15:12] MAC Offset address for Port 11
Bit[19:16] MAC Offset address for Port 12
Bit[31:20] Reserved

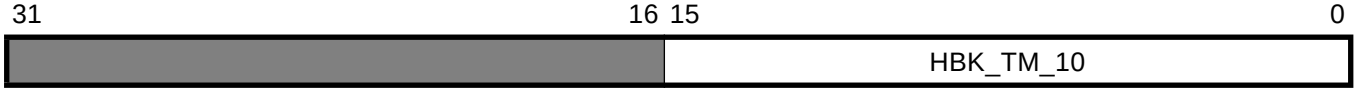
ACKTM - Timer for SOF Checking

- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h610

Bit[9:0] XOFF_CKTMThe time out value to check SOF after XOFF
Bit[31:10] Reserved

AFCHT10 - Flow Control Hold Time of 10MBS Port

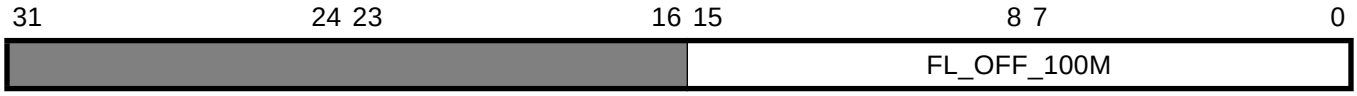
- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h620



Bit[15:0] HBK_TM_10Holding time to remote station for head of line blocking control for 10M port.
Bit[31:16] Reserved

AFCHT 100 - Flow Control Hold Time of 100MBS Port

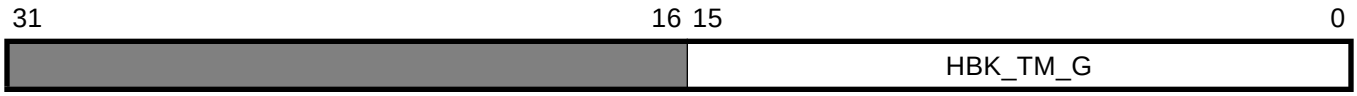
- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h624



Bit[15:0] HBK_TM_100Holding time to remote station for head of line blocking control for 100M port.
Bit[31:16] Reserved

AFCHT1000 - Flow Control Hold Time of GIGBS Port

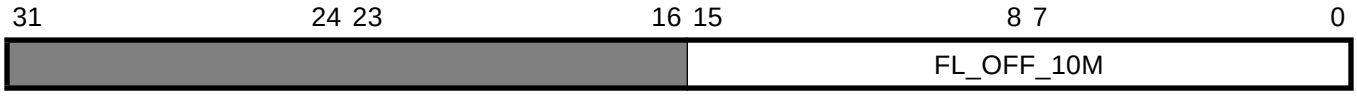
- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h628



Bit[15:0] HBK_TM_GHolding time to remote station for head of line blocking control for 1G port.
Bit[31:16] Reserved

Flow Control Off Time of 10MBS Port

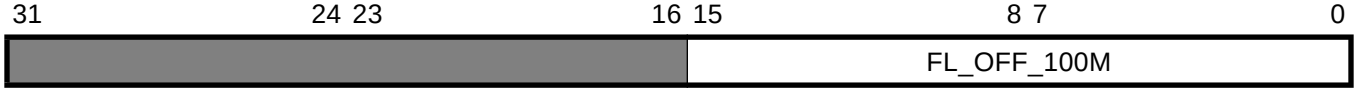
- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h614



Bit[15:0] FL_OFF_10MOff time to remote station for 10M Port.
Bit[31:16] Reserved

AFCOFT100 - Flow Control Off Time of 100MBS Port

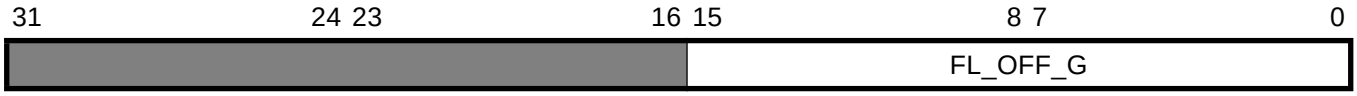
- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h618



Bit[15:0] FL_OFF_100MOff time to remote station for 100M Port.
Bit[31:16] Reserved

AFCOFT100 - Flow Control Off Time of GIGBS Port

- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h61c



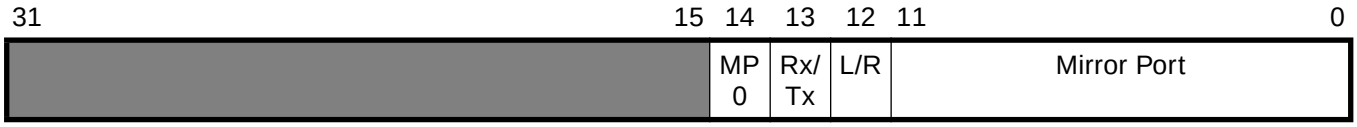
Bit[15:0] FL_OFF_G Off time to remote station for 1G Port.

Bit[31:16] Reserved

Access Control Function Group 2 (Chip Level)

APMR- Port Mirroring Register

- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h5C0



Bit[11:0]	Mirr_Port	The 10/100 port chosen to be mirrored
Bit[12]	Local/Remote	Indicates the mirrored port is from a local or remote device. 0=local1=remote (Note: Does not support 1G port Mirroring.)
Bit[13]	Chose_rx	Indicates whether the mirror is receiving data or transmitting data
Bit[14]	MP0	Mirror to Port 0 (Default=0) MP0=1 Mirror to port 0 MP0=0 Mirror not go to port 0
Bit[31:15]	Reserved	

THKM[0:7] - Trunking Forwarding Port Mask 0-7

- Eight Trunking Hash Key Mask Registers shared the same format.
  - THKM0 Forwarding Port mask for hash key 0
- Address: h5C8
  - THKM1 Forwarding Port mask for hash key 1
- Address: h5CC
  - THKM2 Forwarding Port mask for hash key 2
- Address: h5D0
  - THKM3 Forwarding Port mask for hash key 3
- Address: h5D4
  - THKM4 Forwarding Port mask for hash key 4
- Address: h5D8
  - THKM5 Forwarding Port mask for hash key 5
- Address: h5DC
  - THKM6 Forwarding Port mask for hash key 6
- Address: h5E0
  - THKM7 Forwarding Port mask for hash key 7
- Address: h5E4
- Access: Non-Zero-Wait-State,Direct Access,Write/Read

Bit[11:0] TK_MSK Port trunk mask for trunking hash key

Bit[31:12] Reserved

- CPU sets up this table as follows:

1. Set all bits not in Trunk Groups to 1

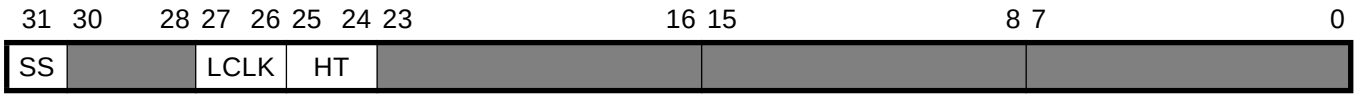
2. Set all bits in the Trunk Group to 0

3. Pick one forwarding port per trunk group and turn the corresponding bit to 1 (each Hash Key may have different forwarding ports, the rule to pick forwarding ports is up to the CPU).

Usage: These masks are used to prevent flooded or multicast packets from being transmitted out with more than one port on a trunk. The Trunking Hash Key is used to select the proper mask(for load distribution). The mask value will be set up to mask off all but one port within each trunk group.

LEDR - LED Register

- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address:h598



Bit[23:0] Reserved

Bit[25:24] HT Holding time for LED signal (Default=00)
00=8msec 01=16msec
10=32msec 11=64msec

Bit[27:26] LCLK LED Clock frequency (Default=00)
00=100M/64=1.56Mhz 01=100M/128=0.78Mhz
10=100M/512=0.195Mhz 11=100M/1024=0.0976Mhz

Bit[30:28] Reserved

Bit[31] SS Start Shift the status bits out from the master device.
This bit has no effect on the slave chip.

Refer to LED design document for detail.

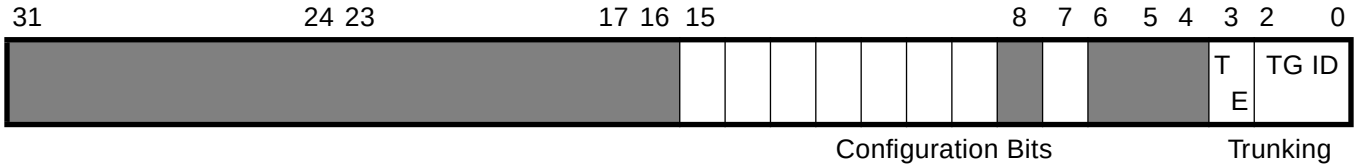
Ethernet MAC Port Control Registers

- One set for each Ethernet MAC Port[12:0]
- MII related controls applies to Port[1:0] only
- Port 12 is always dedicated to GMAC
- Port is disabled when both RR & XR bits are set.

ECR1 - MAC Port Configuration Register

- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address:h0x1*4x: port number

h004	ECR1_p0
h044	ECR1_p1
h084	ECR1_p2
h0c4	ECR1_p3
h104	ECR1_p4
h144	ECR1_p5
h184	ECR1_p6
h1c4	ECR1_p7
h204	ECR1_p8
h244	ECR1_p9
h284	ECR1_p10
h2c4	ECR1_p11
h304	ECR1_p12



Port Trunking ID Bits

Bit[0:2]	TGID	Group ID
Bit[3]	TE	Trunk Enable
		0= Trunk disable 1 = Trunk Enable

Unicast Blocking Control Bits

Bit[6:4]	Reserved

Physical Layer Control Bits

Bit[7] 10M 10M or 100M
1 = 10Mbps 0 = 100Mbps

Bit[8] Reserved

Bit[9] Full_Duplex Enables full duplex mode Default = 0 - Half Duplex

Bit[10] FDX_Polarity Selects the output polarity of Full_Duplex control signal
0 = Low true (Default) 1 = High true

Bit[11] Int_Lpback Setting this bit causes internal connect
TXCLK, TXD, TXD[0:3] to RXCLK, RXD, RXD[0:3]
Default =0 - Disable

Bit[12] Ext_Lpback Setting this bit indicate an external loop-back
(Connection of TXCLK, TXD[0:3] to RXCLK, RXD[0:3] are required)
Default =0 - Disable

Bit[13] FC_Enable Flow Control Enable
Default =0 - Disable
When enabled:

- In Half Duplex mode, the MAC Transmitter applies backpressure for flow control.
  - In Full Duplex mode, the MAC Transmitter sends Flow-Control frames when necessary. The MAC Receiver interprets and processes incoming Flow Control frames. The MAC Receiver marks all Flow Control Frames. Receive DMA discards the received Flow Control Frame and send status reports to the Switch Manager for statistic collection.

  When Disabled:
  - The MAC Transmitter asserts flow control neither by sending Flow Control frames nor by jamming collision.
  - The MAC Receiver still interprets and processes the Flow-Control frames. The MAC Receiver marks all Flow Control frames. Receive DMA discards the received Flow Control frames and send a status report to the Switch Manager for statistic collection.

Bit[14] Link_Polarity Selects the input polarity of Link Status signal
0 = Low true (Default)1 = High true

Bit[15] Tx_Enable Enables MAC Transmitter for transmission
Default =0 - Disable

Bit[31:16] Reserved

18 DC Electrical Characteristics

Absolute Maximum Ratings

Heat Resistance

Package 456 HBGA (Heatslug BGA)
Storage Temperature -65C to +150C
Operating Temperature 0C to +70C
qJC: 3.3 C/W
Maximum Junction Temperature125C

Air Velocity	θ_{JA}(C/W)
0 m/s	12.0
1 m/s	11.0
2 m/s	9.6

Table 1 - Thermal Data for Cooled Chip

Note: When external heat sink is attached, qJA is reduced by about 8-12% in still air.

Voltage

Supply Voltage $VDD2_1$ with Respect to VSS +3.0 V to +3.6 V
Supply Voltage $VDD2_2$ with Respect to VSS +2.38 V to +2.75 V
Voltage on 5V Tolerant Input Pins -0.5 V to ($VDD2_1$ + 3.3 V)
Voltage on 5V Tolerant Input Pins -0.5 V to ($VDD2_2$ + 2.5 V)
Voltage on Other Pins -0.5 V to (VDD2 + 0.3 V)

Caution: Stresses above those listed may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to the Absolute Maximum Ratings for extended periods may affect device reliability.

DC Electrical Characteristics

$VDD2_1$ = 3.0 V to 3.6 V (3.3v +/- 10%)TAMBIENT = 0 C to +70 C. $VDD2_2$ = 2.5V +10% - 5%

Symbol	Parameter Description	Preliminary			Unit
		Min	TypE	Max	
f_{osc}	Frequency of Operation (-50)		100		MHz
I_{DD1}	Supply Current – @ 100 MHz (VDD2 =3.3 V)			TBD	mA
I_{DD2}	Supply Current – @ 100 MHz (VDD2 =2.5 V)				
V_{OH}	Output High Voltage (CMOS)	VDD2 - 0.5			V
V_{OL}	Output Low Voltage (CMOS)			0.5	V
V_{IH-TTL}	Input High Voltage (TTL 5V tolerant)	$VDD2_1$ x 70%		VDD2 + 2.0	V
V_{IL-TTL}	Input Low Voltage (TTL 5V tolerant)			VDD2 x 30%	V
I_{IH-5VT}	Input Leakage Current (0.1 V < V_{IN} < VDD2) (all pins except those with internal pull-up/pull-down resistors)			TBD	µA
I_{IL-5VT}	Output Leakage Current (0.1 V < V_{OUT} < VDD2)			TBD	µA
I_{LI}	Input Leakage Current V_{IH} = VDD2 - 0.1 V (pins with internal pull-down resistors)			TBD	µA
I_{LO}	Input Leakage Current V_{IL} = 0.1 V (pins with internal pull-up resistors)			TBD	µA
C_{IN}	Input Capacitance			5	pF
C_{OUT}	Output Capacitance			5	pF
$C_{I/O}$	I/O Capacitance			7	pF

Table 9 - Recommended Operation Conditions

19 AC Specifications

XPipe Interface

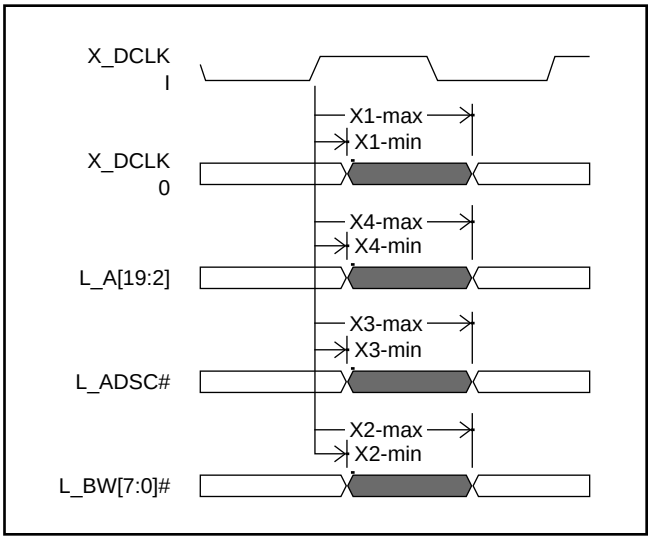


Figure 24 - XPIPE Interface - Output Valid Delay Timing

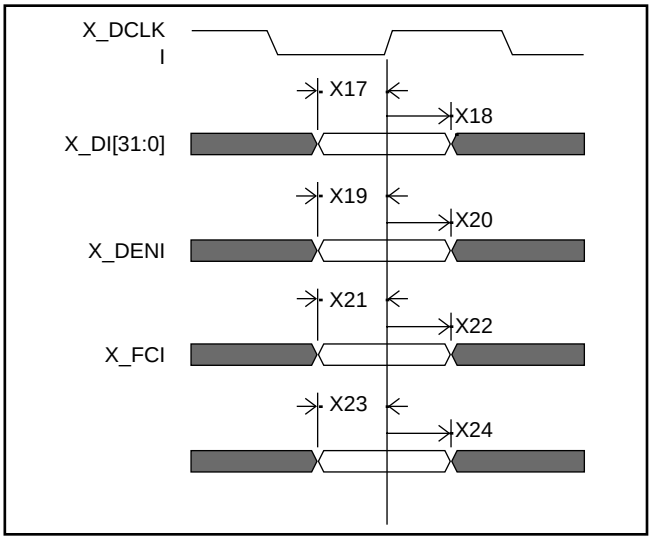
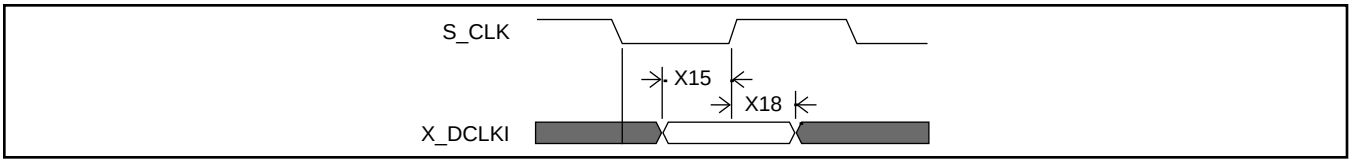


Figure 25 - XPIPE Interface - Output Valid Delay Timing



Symbol	Parameter	-100MHz		Note:
		MIN (ns)	MAX (ns)	
X1	X_DCLKO output valid delay	1	5	C_L = 30pf
X2	X_DO[31:0] output valid delay	1	5	C_L = 30pf
X3	X_DENO output valid delay	1	5	C_L = 30pf
X4	X_FCO output valid delay	1	5	C_L = 30pf
X15	X_DCLKI input set-up time	3		Reference S-CLK
X16	X_DCLKI input hold time	0		Reference S-CLK
X17	X_DI[31:0] input set-up time	3		
X18	X_DI[31:0] input hold time	0		
X19	X_DENI input set-up time	3		
X20	X_DENI input hold time	0		
X21	X_FCI input set-up time	3		
X22	X_FCI input hold time	0		

Table 10 - AC Characteristics - XPIPE Interface

Control Bus Interface

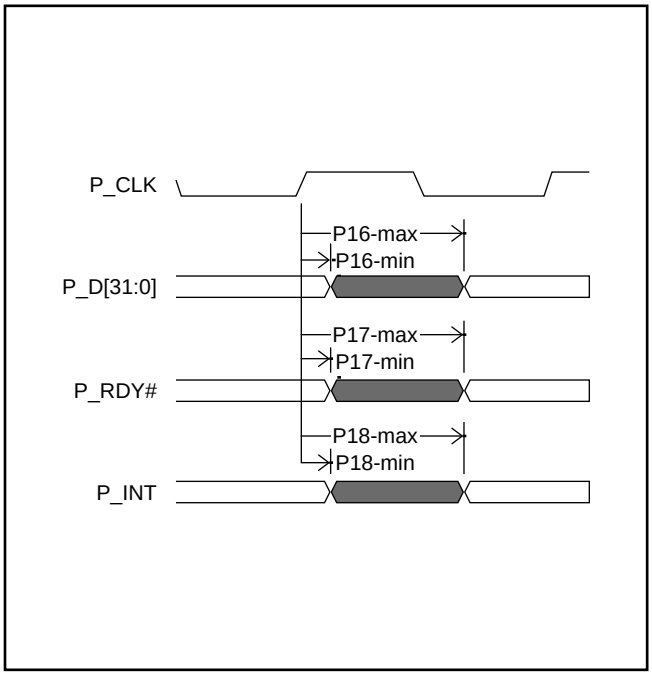


Figure 26 - Control Bus Interface - Output Valid Delay Timing

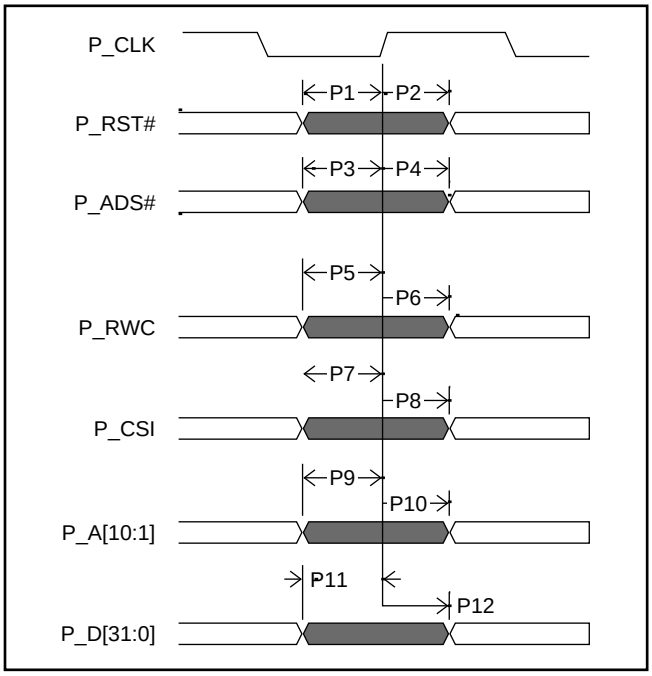


Figure 27 - Control Bus Interface - Input Setup and Hold Timing

Symbol	Parameter	-66MHz		Note:
		Min (ns)	Max (ns)	
	P_CLK			
P1	P_RST# input setup time			
P2	P_RST# input hold time			
P3	P_ADS# input setup time	6		
P4	P_ADS# input hold time	2		
P5	P_RWC# input setup time	6		
P6	P_RWC# input hold time	2		
P7	P_CSI# input setup time	6		
P8	P_CSI# input hold time	2		
P9	P_A[10:1] input setup time	6		
P10	P_A[10:1] input hold time	2		
P11	P_D[31:0]# input setup time	6		
P12	P_D[31:0]# input hold time	2		
P15	P_REQC# input setup time	6		
P16	P_REQC# input hold time	2		
P17	P_BRGI# input setup time	6		
P18	P_BRGI# input hold time	2		
P19	P_D[31:0] output valid delay	2	12	C_L = 65pf
P20	P_A[10:1] output valid delay	2	9	C_L = 50pf
P21	P_RWC# output valid delay	2	9	C_L = 50pf
P22	P_ADS# output valid delay	2	9	C_L = 50pf
P23	P_RDY# output valid delay	2	9	C_L = 50pf
P24	P_INT output valid delay	2	9	C_L = 30pf
P25	P_GNTC output valid delay	2	9	C_L =20pF
P26	P_BRGO# output valid delay	2	9	C_L =20pF
P27	P_CSO# output valid delay	2	9	C_L =20Pf
P28	P_RDY#			
P29	P_RDY#			
P30	FS_CS			
P31	P_BRDY#			
P32	P_BLAST#			

Local SBRAM Memory Interface

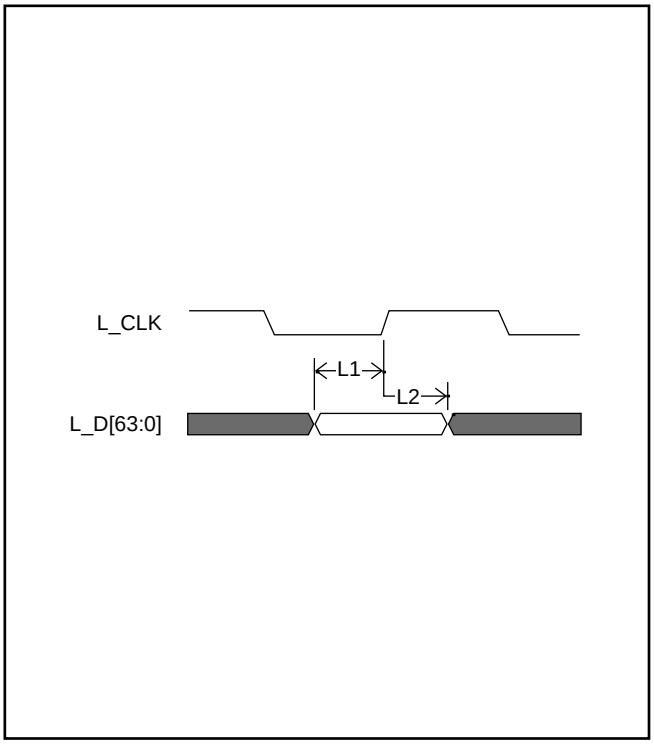


Figure 28 - Local Memory Interface - Input Setup and Hold Timing

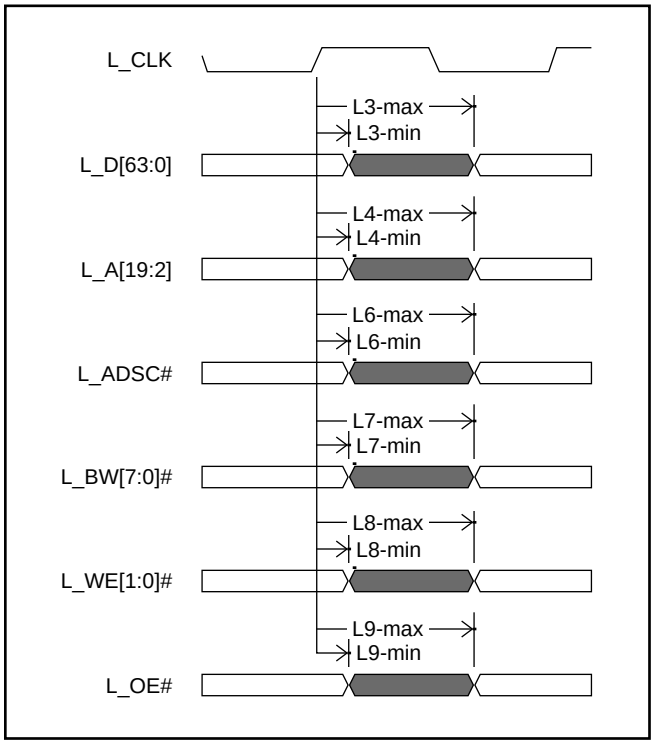


Figure 29 - Local Memory Interface - Output Valid Delay Timing

Symbol	Parameter	-100MHz		Note:
		Min (ns)	Max (ns)	
	L_CLK			C_L = 50pf
L1	L_D[63:0] input set-up time	3		
L2	L_D[63:0] input hold time	1.5		
L3	L_D[63:0] output valid delay	2	7	C_L = 30pf
L4	L_A[20:3] output valid delay	2	7	C_L = 50pf
L6	L_ADSC# output valid delay	2	7	C_L = 50pf
L7	L_BW[7:0]# output valid delay	2	7	C_L = 30pf
L8	L_WE[1:0]#output valid delay	2	7	C_L = 30pf
L9	L_OE[1:0]# output valid delay	0	1	C_L = 30pf

Table 11 - AC Characteristics - Local Memory Interface

Symbol	Parameter	-50MHz		Note:
		MIN (ns)	MAX (ns)	
PM1	M_CLKI			Reference Input Clock
PM2	PM_DENI Input Setup Time	1.5		
PM3	PM_DENI Input Hold Time	2		
PM4	PM_DI[1:0] Input Setup Time	1.5		
PM5	PM_DI[1:0] Input Hold Time	2		
PM6	PM_DENO Output Delay Time	2	11	C_L = 30 pF
PM7	PM_DO[1:0] Output Delay Time	2	11	C_L = 30 pF

Table 12 - AC Characteristics - Port Mirroring Interface

Symbol	Parameter	-50MHz		Note:
		MIN (ns)	MAX (ns)	
M1	M_CLKI			Reference Input Clock
M2	M[11:0]_RXD[1:0] Input Setup Time	1.5		
M3	M[11:0]_RXD[1:0] Input Hold Time	2		
M4	M[11:0]_CRS_DV Input Hold Time	1.5		
M5	M[11:0]_TXEN Output Delay Time	2	11	C_L = 30 pF
M6	M[11:0]_TXD[1:0] Output Delay Time	2	11	C_L = 30 pF
M7	M[11:0]_LINK Input Setup Time			

Table 13 - AC Characteristics - Reduced Media Independent Interface

Symbol	Parameter	-125Mhz		Note:
		MIN (ns)	MAX (ns)	
GREF_CLK	GREF_CLK			Gigabit Ref Clock
G1	M[12]_RXD[7:0] Input Setup Times	2		
	M[12]_RXD[7:0] Input Hold Times	0		
G2	M[12]_RX_DV Input Setup Times	2		
G3	M[12]_RX_DV Input Hold Times	0		
G4	M[12]_RX_ER Input Setup Times	2		
G5	M[12]_RX_ER Input Hold Times	0		
G6	M[12]_CRS Input Setup Times	2		
G7	M[12]_CRS Input Hold Times	0		
G8	M[12]_COL Input Setup Times	2		
G9	M[12]_COL Input Hold Times	0		
G12	M[12]_TXD[7:0] Output Delay Times	1	5	C_L = 20pf
G13	M[12]_TX_EN Output Delay Times	1	5	C_L = 20pf
G14	M[12]_TX_ER Output Delay Times	1	5	C_L = 20pf
G16	M[12]_LNK Input Setup Times			

Table 14 - AC Characteristics - Giabit Media Independent Interface

Symbol	Parameter	-125Mhz		Note:
		MIN (ns)	MAX (ns)	
GREF_CLK	GREF_CLK			Gigabit Reference Clock
GP1	GP_RXD[7:0] Input Setup Times	2		
GP2	GP_RXD[7:0] Input Hold Times	0		
GP3	GP_RXD[8] Input Setup Times	2		
GP4	GP_RXD[8] Input Hold Times	0		
GP5	GP_RXD[9] Input Setup Times	2		
GP6	GP_RXD[8] Input Hold Times	0		
GP7	GP_RXCLK1 Input Setup Times	2		
GP8	GP_RXCLK1 Input Hold Times	0		
GP9	GP_RXCLK0 Input Setup Times	2		
GP10	GP_RXCLK0 Input Hold Times	0		
GP11	GP_TXD[7:0] Output Delay Times	1	5	C_L = 20pf
GP12	GP_TXD[8] Output Delay Times	1	5	C_L = 20pf
GP13	GP_TXD[9] Output Delay Times	1	5	C_L = 20pf
GP14	GP_TXCLK Output Delay Times			C_L = 20pf
GP15	GP_LNK Input Setup Times			

Table 15 - AC Characteristics - Physical Media Attachment Interface

Symbol	Parameter	Variable FREQ.		Note:
		MIN (ns)	MAX (ns)	
LE1	LE_DI Input Setup Times			
LE2	LE_DI Input Hold Times			
LE3	LE_SYNCI Input Setup Times			
LE4	LE_SYNCI Input Hold Times			
LE5	LE_CLKO Output Valid Delay			C_L = 30pf
LE6	LE_DO Output Valid Delay			C_L = 30pf
LE7	LE_SYNCO Output Valid Delay			C_L = 30pf

Table 16 - AC Characteristics - LED Interface

20 Mechanical Data

Packaging Information

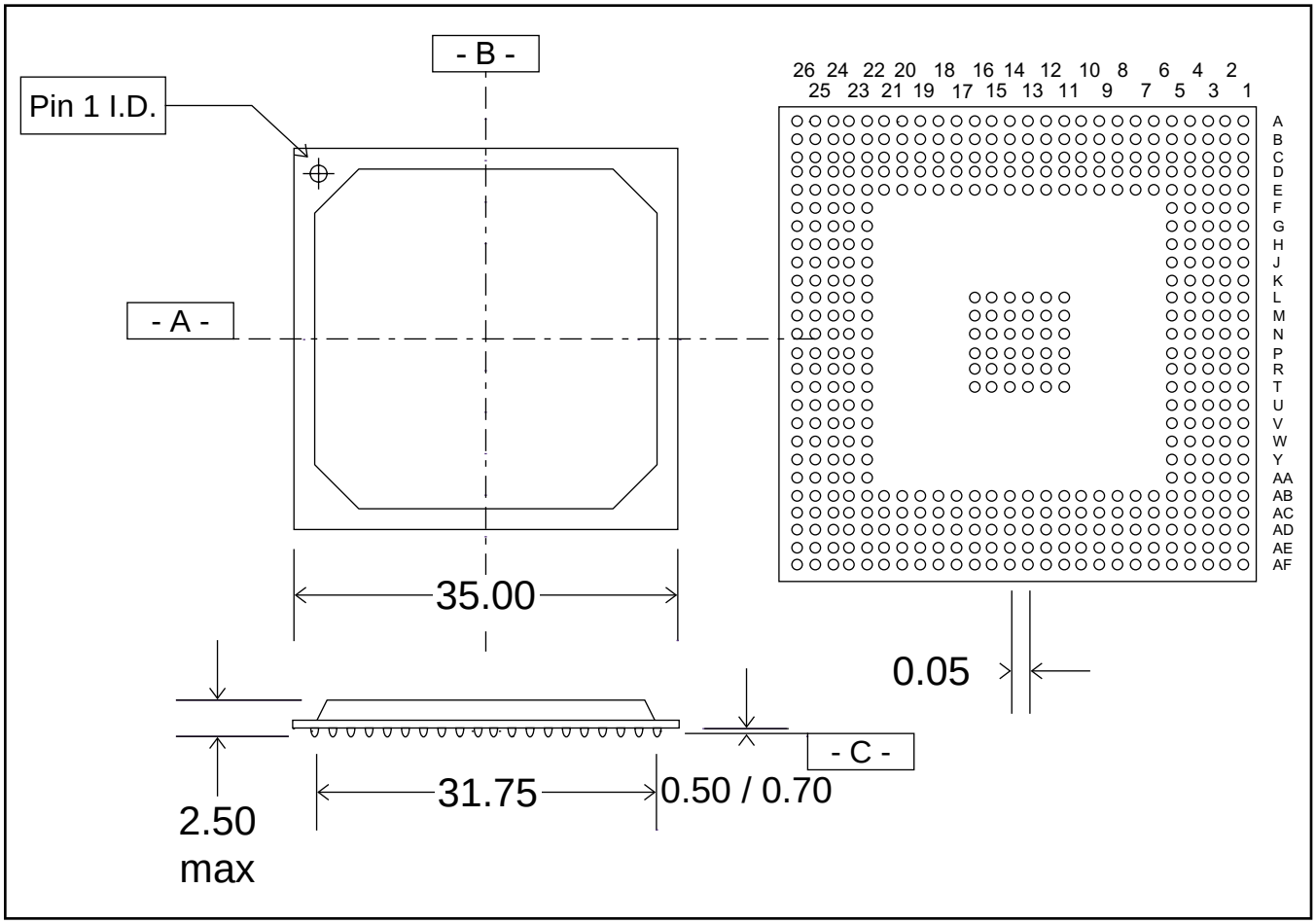


Figure 30 - 456-PIN BGA Packaging Diagram

ZARLINK SEMICONDUCTOR

http://www.zarlink.com

World Headquarters - Canada
Tel: +1 (613) 592 0200
Fax: +1 (613) 592 1010

North America - West Coast
Tel: (858) 675-3400
Fax: (858) 675-3450

North America - East Coast
Tel: (978) 322-4800
Fax: (978) 322-4888

Asia/Pacific
Tel: +65 333 6193
Fax: +65 333 6192

Europe, Middle East, and Africa (EMEA)
Tel: +44 (0) 1793 518528
Fax: +44 (0) 1793 518581