MDS212

12-Port 10/100Mbps Ethernet Switch

Data Sheet

October 2003

Features

- 12 10/100Mbps Autosensing, Fast Ethernet ports with Reduced MII Interface
- 32-bit wide bi-directional pipe at 100Mhz provides 6.4Gbps pipe to connect two MDS212 chips
- Supports up to 3.572 Mpps system throughput using non-blocking architecture
- High performance Layer 2 packet forwarding and filtering at full wire speed.
- Very low latency through single store and forward at ingress port and cut-through switching at destination ports
- Port Trunking and Load Sharing for high bandwidth links between switches
- On-chip address lookup engine and memory for up to 2K MAC addresses
  - up to 64K using management CPU memory
  - Supports up to 4k MAC addresses with 24 ports (2-chip solution)
  - Up to 16K using external buffer memory
- Parallel Flash interface for fast self initialization
- Supports packet filtering and port security
  - System wide filtering
  - Static MAC destination and source address filtering
  - VLAN for multicast/broadcast filtering
  - Protocol filtering
  - Local port filtering
  - Aging control for secure MAC addresses
- Provides 256-port and ID Tagged Virtual LANs (VLANs) 802.1Q
  - ID Tagging Insertion/Extraction
- Supports IP Multicasting through IGMP Snooping
- XpressFlow Quality of Service (QoS), IEEE 802.1p, supports 4 Level transmission priorities, weighted fair queuing based packet scheduling, user mapping of priority levels and weights
- 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
- Standard software modules available:
  - Browser, GUI, and text menu
  - IEEE 802.1d Spanning Tree Algorithm
  - SNMP management
  - Telnet for remote control console
  - Automatic Booting via TFTP Protocols.
  - Remote Monitoring (RMON) and storage for a management agent
  - IGMP for IP multicast
  - GVRP, GMRP
- Packaged in 456-Pin Ball Grid Array

Ordering Information

MDS212CG 456 Pin HSBGA

0°C to 70°C

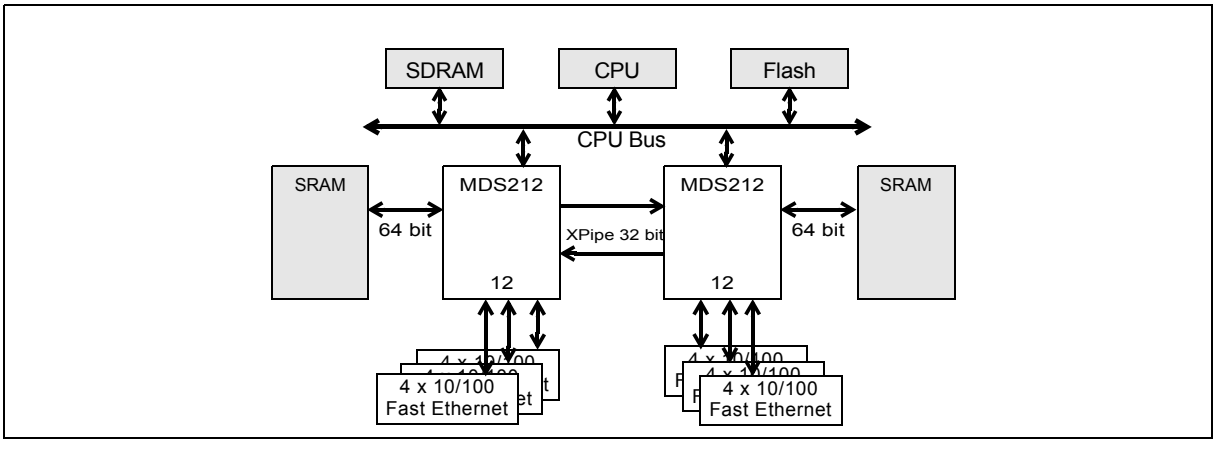


Figure 1 - 24 10/100Mbps Port System Configuration

Description

The MDS212 is a 12-port 10/100Mbps high-performance, non-blocking Ethernet switch with on-chip address memory and address lookup engine. A single chip provides 12 - 10/100Mbps ports. The MDS212 can be utilized in both managed and unmanaged switching applications.

The 3.2 Gbps XPipe allows a high-speed connection between two MDS212 chips, providing a optimal, low-cost, workgroup witch with 24 10/100 Fast Ethernet ports.

In half-duplex mode, all ports support back pressure 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 200Mbps aggregate bandwidth connections.

The MDS212 supports port trunking/load sharing on the 10/100Mbps ports. Port trunking/load sharing can be used to group ports between interlinked switches for increased system bandwidth. Ports within a trunk must reside within a single MDS212, 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 256 IEEE 802.1Q Virtual LANs (VLAN). Each port may be programmed to recognize VLANs, and will transmit frames along with their VLAN Tags, for interoperability, to systems that support VLAN Tagging.

Each port independently collects statistical information using SNMP and the Remote Monitoring Management Information Base (RMON – MIB). Access to these statistical counter/registers are provided via the CPU interface. SNMP Management frames may be received and/or transmitted via the CPU interface and thus creates a complete network management solution.

The MDS212 utilizes cost effective, high performance, pipelined SBRAM 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 MDS212 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 MDS212 is packaged in a 456-pin Ball Grid Array.

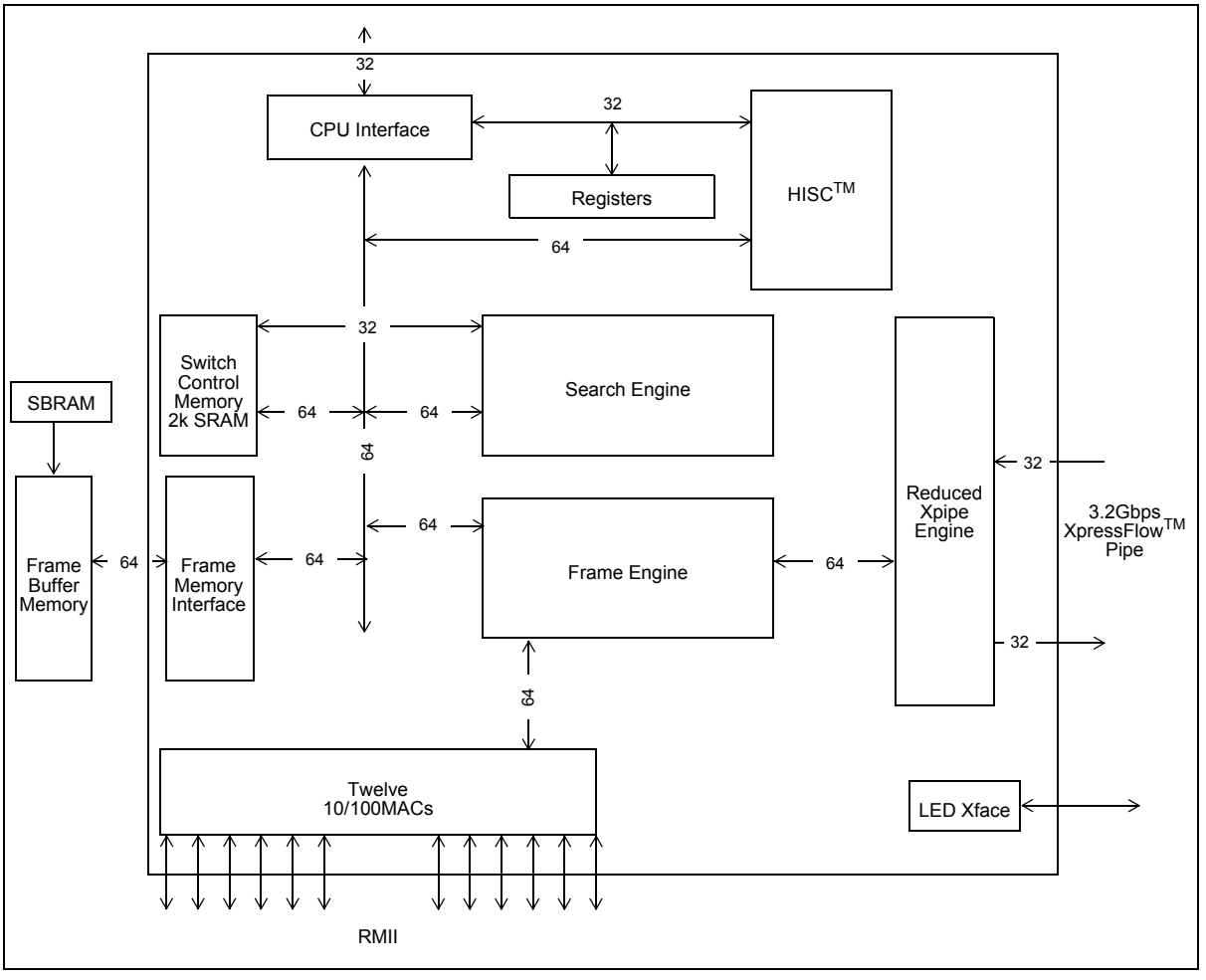


Figure 2 - System Block Diagram

Note:
All registers are 32-bit width.
The Control Bus is 32-bits wide and the Memory Bus is 64-bits wide.
The MDS212 contains 12 Fast Ethernet Ports.
The LED interface has 3 output signals (1 data and 2 control).
The XPipe is 32-bits wide.

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

List of Figures

List of Tables

1.0 Ball-signal Description and Assignments

	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_REQ1	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	Reserved	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	Reserved	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_GNT1	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_WEO#	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																	NC	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	NC	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	NC	NC	NC	NC	NC	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	NC	NC	NC	NC	NC	NC	NC
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	NC	NC	NC	NC	NC	NC	NC
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	NC	NC	NC	NC	NC	NC	NC
	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

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

1.1 Ball-Signal Assignments

Signal Name	Ball No.
NC	D4
RESERVED	C3
AGND	A1
RESERVED	B1
AVDD	C1
T_MODE#	F5
RESERVED	C2
L_CLK	D3
L_D0	E4
L_D1	D2
L_D2	E3
L_D3	F4
L_D4	D1
L_D5	E2
L_D6	E1
L_D7	G4
L_D8	F3
L_D9	H5
L_D10	F2
L_D11	F1
L_D12	H4
L_D13	G3
L_D14	G2
L_D15	G1
L_D16	H3
L_D17	J4
L_D18	H2
L_D19	K5
L_D20	H1
L_D21	J3
L_D22	K4
L_D23	J2
L_D24	J1
L_D25	M5
L_D26	K3
L_D27	K2
L_D28	L4
L_D29	K1
L_D30	L3
L_D31	L2
L_D32	R5
L_D33	R1
L_D34	R2
L_D35	R4
L_D36	R3
L_D37	T1
L_D38	T2
L_D39	T3
L_D40	U1
L_D41	T4
L_D42	U2
L_D43	U3
L_D44	V1
L_D45	V2
L_D46	U4
L_D47	U5
L_D48	V3
L_D49	W1
L_D50	W2
L_D51	V4
L_D52	W3
L_D53	Y1
L_D54	Y2
L_D55	Y3
L_D56	W4
L_D57	W5
L_D58	AA1
L_D59	AA2
L_D60	AA3
L_D61	Y4
L_D62	AB1
L_D63	AB2
M0_LNK	AC1
M_CLKI	AA4
M0_TXEN	AB3
M0_TXD1	AC2
M3_TXD1	AC7
M3_TXD0	AD6
M3_CRS_DV	AB8
M3_RXD1	AE6
M3_RXD0	AF6
M4_LNK	AC8
M4_TXEN	AD7
M4_TXD1	AE7
M4_TXD0	AF7
M4_CRS_DV	AD8
M4_RXD1	AC9
M4_RXD0	AE8
M5_LNK	AB10
M5_TXEN	AF8
M5_TXD1	AD9
M5_TXD0	AC10
M5_CRS_DV	AE9
M5_RXD1	AF9
M5_RXD0	AD10
M6_LNK	AE10
M6_TXEN	AC11
M6_TXD1	AB12
M6_TXD0	AF10
M6_CRS_DV	AD11

Signal Name	Ball No.
M6_RXD1	AE11
M6_RXD0	AF11
M7_LNK	AC12
M7_TXEN	AD12
M7_TXD1	AE12
M7_TXD0	AF12
M7_CRS_DV	AC13
M7_RXD1	AD13
M7_RXD0	AF13
M8_LNK	AE13
M11_TXD1	AF19
M11_TXD0	AB19
M11_CRS_DV	AE19
M11_RXD1	AC18
M11_RXD0	AD19
NC	AF20
NC	AE20
NC	AD20
NC	AC19
NC	AF21
NC	AE21
NC	AD21
NC	AC20
NC	AF22
NC	AE22
NC	AF23
NC	AC21
NC	AD22
NC	AE23
NC	AB21
NC	AC22
NC	AD23
NC	AE24
NC	AF24

Signal Name	Ball No.
NC	AF25
NC	AE25
NC	AA22
NC	AC23
NC	AD24
NC	AF26
NC	AE26
NC	AD26
NC	AD25
M_MDIO	AC24
L_WE0#	L1
L_OE0#	M2
L_WE1#	M3
L_OE1#	M4
L_BW0#	M1
L_BW1#	N4
S_CLK	N5
L_BW2#	N3
L_BW3#	N1
L_ADS#	N2
L_BW4#	P2
L_BW5#	P1
L_BW6#	P4
L_BW7#	P3
T_D15	U23
T_D14	V25
T_D13	V26
T_D12	U24
T_D11	U25
T_D10	R22
T_D9	T23
T_D8	U26
T_D7	T24
T_D6	T25

Signal Name	Ball No.
T_D5	T26
T_D4/BS_RDYOP	R23
T_D3/BS_PSD	R24
T_D2/BS_SWM	R25
T_D1/BS_RW	R26
T_D0/BS_BMOD	P23
P_D0	P22
P_D1	P24
P_D2	P26
P_D3	P25
P_D4	N25
P_D5	N26
P_D6	N24
P_D7	N23
P_D8	M26
P_D9	M25
P_D10	M23
P_D11	M24
P_D12	L26
P_D13	L25
P_D14	L24
P_D15	M22
P_D16	K26
P_D17	L23
P_D18	K25
P_D19	K24
P_D20	J26
P_D21	J25
P_D22	K23
P_D23	J24
P_D24	H26
P_D25	K22
P_D26	H25
P_D27	J23

Signal Name	Ball No.
P_D28	H24
P_D29	G26
P_D30	G25
P_D31	G24
P_A1	H23
P_A2	F26
P_A3	F25
P_A4	F24
P_A5	H22
P_A6	G23
P_A7	E26
P_CLK	E25
P_A8	D26
P_A9	F23
M0_TXD0	AA5
M0_CRS_DV	AB4
M0_RXD1	AC3
M0_RXD0	AD2
NC	AD1
NC	AE1
M1_LNK	AE2
M1_TXEN	AC4
M1_TXD1	AD3
M1_TXD0	AF1
M1_CRS_DV	AF2
M1_RXD1	AF3
M1_RXD0	AE3
M2_LNK	AB6
M2_TXEN	AD4
M2_TXD1	AC5
M2_TXD0	AE4
M2_CRS_DV	AD5
M2_RXD1	AC6
M2_RXD0	AF4

Signal Name	Ball No.
M3_LNK	AE5
M3_TXEN	AF5
P_A10	E24
P_RST#	D25
P_RWC#	F22
P_ADS#	E23
P_RDY#	D24
P_BRDY#	C25
P_BLAST#	C26
NC	B26
NC	B25
P_INT	D23
P_REQC	C24
P_GNTC	A26
P_REQ1	A25
P_GNT1	E21
P_CSI#	A24
RESERVED	B24
RESERVED	C23
X_FCI	D22
X_DCLKI	B23
X_DNI	C22
X_DI0	E19
X_DI1	D21
X_DI2	A23
X_DI3	B22
X_DI4	A22
X_DI5	D20
X_DI6	C21
X_DI7	B21
X_DI8	A21
X_DI9	D19
X_DI10	C20
X_DI11	B20

Signal Name	Ball No.
X_DI12	A20
X_DI13	E17
X_DI14	C19
X_DI15	D18
X_DI16	B19
X_DI17	A19
X_DI18	C18
X_DI19	D17
X_DI20	B18
X_DI21	A18
X_DI22	C17
X_DI23	B17
X_DI24	D16
X_DI25	A17
X_DI26	E15
X_DI27	C16
X_DI28	B16
X_DI29	A16
X_DI30	D15
X_DI31	C15
X_FCO	B15
X_DCLKO	A15
X_DENO	D14
X_DO0	E14
X_DO1	C14
X_DO2	A14
M8_TXEN	AE14
M8_TXD1	AF14
M8_TXD0	AB13
M8_CRS_DV	AD14
M8_RXD1	AC14
M8_RXD0	AF15
M9_LNK	AE15
M9_TXEN	AC15

Signal Name	Ball No.
M9_TXD1	AB15
M9_TXD0	AD15
M9_CRS_DV	AF16
M9_RXD1	AE16
M9_RXD0	AD16
M10_LNK	AF17
M10_TXEN	AC16
M10_TXD1	AE17
M10_TXD0	AD17
M10_CRS_DV	AF18
M10_RXD1	AB17
M10_RXD0	AC17
M11_LNK	AE18
M11_TXEN	AD18
X_DO3	B14
X_DO4	B13
X_DO5	A13
X_DO6	C13
X_DO7	D13
X_DO8	A12
X_DO9	E12
X_DO10	B12
X_DO11	D12
X_DO12	C12
X_DO13	A11
X_DO14	B11
X_DO15	C11
X_DO16	A10
X_DO17	D11
X_DO18	B10
X_DO19	C10
X_DO20	A9
X_DO21	B9
X_DO22	E10

Signal Name	Ball No.
X_DO23	D10
X_DO24	C9
X_DO25	A8
X_DO26	B8
X_DO27	D9
X_DO28	C8
X_DO29	A7
X_DO30	B7
X_DO31	C7
L_A3	D8
L_A4	A6
L_A5	B6
L_A6	C6
L_A7	D7
L_A8	A5
L_A9	E8
L_A10	B5
L_A11	A4
L_A12	D6
L_A13	C5
L_A14	B4
L_A15	E6
L_A16	D5
L_A17	C4
L_A18	B3
L_A19	A3
L_A20	A2
RESERVED	B2
VCC	E7
VCC	E11
VCC	E16
VCC	E20
VCC	G5
VCC	G22

Signal Name	Ball No.
VCC	L5
VCC	L22
VCC	T5
VCC	T22
M_MDC	AB23
LE_DI	AC25
LE_CLKO	AB24
LE_SYNCI	AA23
LE_DO	AC26
LE_SYNCO	AB25
T_D31/PM_D0[1]	W22
T_D30/PM_D0[0]	AB26
T_D29/PM_DENO	Y23
T_D28/PM_D1[1]	AA24
T_D27/PM_D1[0]	AA25
T_D26/PM_DENI	AA26
T_D25	W23
T_D24	Y24
T_D23	Y25
T_D22	Y26
T_D21	W24
T_D20	U22
T_D19	V23
T_D18	W25
T_D17	W26
T_D16	V24
VCC	Y5
VCC	Y22
VCC	AB7
VCC	AB11
VCC	AB16
VCC	AB20
VDD	E9
VDD	E18

Signal Name	Ball No.
VDD	J5
VDD	J22
VDD	N22
VDD	P5
VDD	V5
VDD	V22
VDD	AB9
VDD	AB18
GND	E5
GND	E13
GND	E22
GND	L11
GND	L12
GND	L13
GND	L14
GND	L15
GND	L16
GND	M11
GND	M12
GND	M13
GND	M14
GND	M15
GND	M16
GND	N11
GND	N12
GND	N13
GND	N14
GND	N15
GND	N16
GND	P11
GND	P12
GND	P13
GND	P14
GND	P15

Signal Name	Ball No.
GND	P16
GND	R11
GND	R12
GND	R13
GND	R14
GND	R15
GND	R16
GND	T11
GND	T12
GND	T13
GND	T14
GND	T15
GND	T16
GND	AB5
GND	AB14
GND	AB22

2.0 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.

CPU 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	Processor Data Bus- Data Bit [31:0]
E24, F23, D26, E26, G23, H22, F24, F25, F26, H23	P_A[10:1]	Input /Output –U	Process Address Bus-Address Bit [10:1]
D25	P_RST#	Input-ST	Process Bus – Master Reset
F22	P_RWC#	Input/Output-TS, U	Process Bus – Read/Write Control Programmable polarity
E23	P_ADS#	Input /Output-TS, U	Process Address Strobe
D24	P_RDY#	Output-OD - TS, U	Process Bus – Data Ready
C25	P_BRDY#	Input- TS, U	Process Bus – Burst Ready
C26	P_BLAST#	Input- TS, U	Process Bus – Burst Last
D23	P_INT	Output	Process Bus – Interrupt Request Programmable polarity
E25	P_CLK	Input	Process Bus – Bus Clock
A24	P_CSI#	Input- U	Chip Select
C24	P_REQC	Input	Bus Request from CPU - Only using in debug mode when system is unmanaged.
A26	P_GNTC	Output	Bus Grant to CPU - Only using in debug mode when system in unmanaged.
A25	P_REQ1	Input/Output	Bus Request from secondary MDS212 to primary MDS212. Only using in debug mode when system in unmanaged.
E21	P_GNT1	Input/Output	Bus Grant to secondary MDS212 from primary MDS212. Only using in debug mode when system is unmanaged.

Note:
# = 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

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, B4, 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 RMII Ports[11:0])
AC24	M_MDIO	IO-TS	MII Management Data I/O – (Common for all RMII Ports{11:0])
AA4	M_CLKI	Input	Reference Input Clock

Frame Buffer Interface (continued)

Ball No(s)	Symbol	I/O	Description
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]_TXE N	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
XPIPE INTERFACE	**I/O FUNCTION**		
B23	X_DCLKI	Input	XpipeFlow Data Clock Input
C22	X_DENI	Input	XpipeFlow Data Enable Input
D22	X_FCI	Input	XpipeFlow 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	XpipeFlow Data Input Bits [31:0]
A15	X_DCLKO	Output	XpipeFlow Data Clock Output
B15	X_FCO	Output	XpipeFlow Flow Control Output

Frame Buffer Interface (continued)

Ball No(s)	Symbol	I/O	Description
D14	X_DENO	Output	XpipeFlow 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	XpipeFlow 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 (sharing pins with other functions and for Testing purpose only)			
F15	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

Frame Buffer Interface (continued)

Ball No(s)	Symbol	I/O	Description
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	CPU Bus mode MUST BE SET TO 0
R26	BS_RW	Input	CPU Read/Write Control Polarity Selection Default=1 0=R/W#; 1=W/R#
R25	BS_SWM	Input	Switch mode: Default=1 0=Managed mode 1= Unmanaged
R24	BS_PSD	Input	Primary Device Enable Pin Default=1 0=Secondary 1=Primary
R23	BS_RDYOP	Input	Option of merge the P_RDY# and P_BRDY# as one pin Default=1 0=Merged pin 1=Separated pins

Note:
# = 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

3.0 The Media Access Control (MAC)

The MDS212 MAC contains twelve Fast Ethernet MACs, 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). The MAC 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 also manages half-duplex collisions, including collision avoidance and contention resolution (collision handling). The MDS212 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. In VLAN aware switches, the MAC inserts, replaces, or removes VLAN Tags from these frame formats based on instructions from the Search Engine. When necessary, the MAC regenerates the Frame Check Sequence and performs "padding" for frames of less than 64 bytes in size.

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 then signals the Frame Engine, using Receive Direct Memory Access (RxDMA), that data is available in the FIFO and is ready for storage.

3.1 MAC Configuration

MAC operations are configured through the Global Device Configuration Register (DCR2) and/or the MAC Control and Configuration Registers (ECR0, ECR1), defined in the Register Definition Section of the MDS212 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.

3.2 The Inter-Frame Gap

The Inter-frame Gap (IFG), defined as 96 bit times, is the interval between successive Ethernet frames for the MAC. 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 deassertion 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.

3.3 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 transmitting or forwarding frames whose data lengths do not meet the minimum requirements, the MAC 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. Although the MAC may be configured to forward oversized frames in the Device Configuration Register (DCR2), the frame buffers' maximum size of 1536 bytes cannot be exceeded. For VLAN Aware systems, the maximum frame size is increased from 1518 bytes to 1522 bytes to accommodate the 4-byte VLAN Tag.

3.4 Collission Handling and Avoidance

If multiple stations on the same network attempt to transmit at the same time, interference can occur causing a collision. The MAC 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 defers (delays) its own transmissions to decrease the load on the network. This is called collision avoidance.

If a collision occurs, the MAC ceases data transmission after the first 64 bytes of data 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 "backs off"

(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.

3.5 Auto-negotiation

The default value of the MDS212 MAC enables Auto-negotiation. The default value is overwritten if the PHY lacks the ability to support Auto-negotiation, which is ascertained through its respective management interface, RMII. The Auto-negotiation 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 Auto-negotiation. The MDS212 MAC, supporting Auto-negotiation, reads the results of the operation from the MAC Configuration Registers.

3.6 VLAN Support

Virtual Local Area Networks (VLANs) assemble a group of independent ports (and/or MAC addresses) to communicate as if they were on the same physical LAN segment, without being restricted by the physically connected hardware. The ports are logically grouped together by VLAN Identifiers (VLAN IDs). The MDS212 implements a MAC Address-based classification that associates each VLAN ID with its MAC address in the Switch Database Memory (SDM) for purposes of aging out, or replacing, old VLANs.

The MDS212 MAC recognizes VLAN-Tagged frame formats. During transmission the MAC inserts (or extracts) the 4-byte VLAN Tag and regenerates the Frame Check Sequence for the transmitted frame. VLAN support requires an increase in the maximum legal frame size, which is set in the Device Configuration Register (DCR2), from 1518 to 1522 bytes. During transmission, if the MAC is required to remove the VLAN Tag from a 64-67 byte Rx frame, the MAC will append extra bytes (pad) to form a 64 byte frame.

3.7 MAC Control Frames

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 MDS212 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.

3.8 Flow Control

Flow control reduces the risk of data loss during long bursts of activity, by saturating the buffer memory with backlogged frames. The MDS212 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 MDS212 recognizes congestion by constantly monitoring available frame buffer memory. When the amount of free buffer space has been depleted, the MDS212 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 back pressure flow control when necessary.

3.9 Collision-based Flow 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 MDS212 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.

3.9.1 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 MDS212 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.

4.0 Frame Engine Description

The Frame Engine is the heart of the MDS212. 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, the destination MAC address, and the VLAN tag. 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 and frame type (e.g. unicast or multicast).

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 queues (TxQ). There are four TxQs per port, one for each priority class. 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 signalling 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.

For multicast frames, the process is slightly different. The Frame Engine uses the VLAN index, which is part of the search result, to identify the destination ports. For local destination ports, the Frame Engine writes a job to each port's TxQ. When a transmission job is selected, the TxDMA moves data from the memory to the MAC TxFIFO. Multicast frame data is sent multiple times, until all local destination ports' requests are satisfied.

For a VLAN that includes remote destination ports, the multicast frame data is forwarded once through the XPipe and then stored in the remote device's memory. The remote Frame Engine processes this multicast frame as if it came from a local port.

A frame is stored in a Frame Data Buffer (FDB) until it is transmitted. FDBs are external, located in a MDS212'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.

As a frame lives through its lifecycle, its status is updated in the FCB. The FCB also contains vital frame information, such as destination port and length. There is a one-to-one correspondence between the FCB and the FDB: FCB#274 contains information about the frame stored in FDB#274. An FCB/FDB pair is called a "frame buffer," or simply a "buffer." The number 274 is called the handle or the buffer handle. The Frame Engine takes care of the distribution and the releasing of buffers. It also keeps buffer counters to ensure no port or single type of traffic occupies too many buffers.

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 two reasons. All buffers are currently occupied, or the received frame is a multicast frame and the multicast buffer quota is exhausted. In either case, the RxDMA will discard the frame, without getting a handle. If set, the register bit DCR2[26], IPMC, enables IP multicast privileges. If enabled, the RxDMA discards regular multicast frames if the multicast forwarding FIFO's occupancy exceeds the programmable threshold (see register MBCR[21:20], MCTH). An IP multicast frame is discarded only when the multicast frame's forwarding FIFO is full.

4.1 Transmission Scheduling

There are four transmit scheduling queues (TxQ) per port, one for each priority. When a port is ready to transmit, when the previous frame finished transmitting, the port control module notifies the Frame Engine. The Frame Engine selects one TxQ out of the four priority queues, depending on the frame's arrival time and weighted round robin state (refer to the QoS chapter for more detail). It reads an entry from the selected transmission scheduling queue, and if the source port of the selected frame is local, a transmission request is issued to the local TxDMA module. If, on the other hand, the source port is remote, the data request message is forwarded across the XPIPE and subsequently arrives at the forwarding engine.

The four transmit scheduling queues per output port allows the Frame Engine to perform weighted round robin (WRR) to provide quality of service (QoS). The Search Engine classifies the frames into four internal priorities, Q0, Q1, Q2, and Q3, in decreasing priority. The 802.1p priority bits are mapped to the internal priorities by a programmable mapping, accessible via register AVTC. The user can program the queue weights via register AXSC, and thereby control the relative rates of the four internal-priority tagged frames.

The maximum TxQ lengths are programmable from 128 entries to 1024 entries per queue. 48 TxQs are located in the external memory. The maximum queue lengths and the base memory addresses are accessible by the register group {CPUIRCMD, CPUIRDAT, CPUIRRDY}, under type QCNT.

4.2 Buffer Management

The buffer manager is responsible for the free handle allocation, buffer usage monitoring, buffer release, and FCB access control. Free handles point to buffers that are not occupied by a frame. These free buffers can be allocated to a new frame received by the RxDMA or the MRP. When the Frame Engine is done processing a frame, its handle is released to the free handle pool.

The free handle pool must be initialized via the register group CPUIRCMD, CPUIRDAT, CPUIRRDY, type BMCT, before device operation. The Buffer Manager Control Table (BMCT) is the pool of free handles. At reset, the BMCT is empty. Prior to device operation, free handles must be written to the BMCT. The user must write the integers {0,1,2,3, … K-1} to the BMCT one-at-a-time, where K is the maximum number of buffers. The value of K depends on the external memory size and partition, and it can be 128, 256, 512, or 1024.

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 programmed 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.

The Frame Engine also keeps a buffer counter for multicast traffic types. The buffers occupied by incoming multicast frames are limited. This prevents multicast frames from blocking unicast ones from entering the system. The threshold for multicast traffic types is programmed via register MBCR.

5.0 Frame Buffer Memory

5.1 Frame Buffer Memory Configuration

The MDS212 system utilizes external SRAM for its Frame Buffer Memory configuration, where the size of memory supported is ½ MB, 1MB and 2MB configurations. The following table shows four memory configuration examples for the MDS212 system:

SRAM Type	One Bank		Two Bank	
	Address	Size	Address	Size
64Kx32	L_A[18:3]	½MB	L_A[19:3]	1M
128Kx32	L_A[19:3]	1MB	L_A[20:3]	2M

Table 1 - Type and Size of Memory Chips

The following figure shows the connections between the Frame Buffer Memory and the MDS212 for one-bank and two-bank memory configurations:

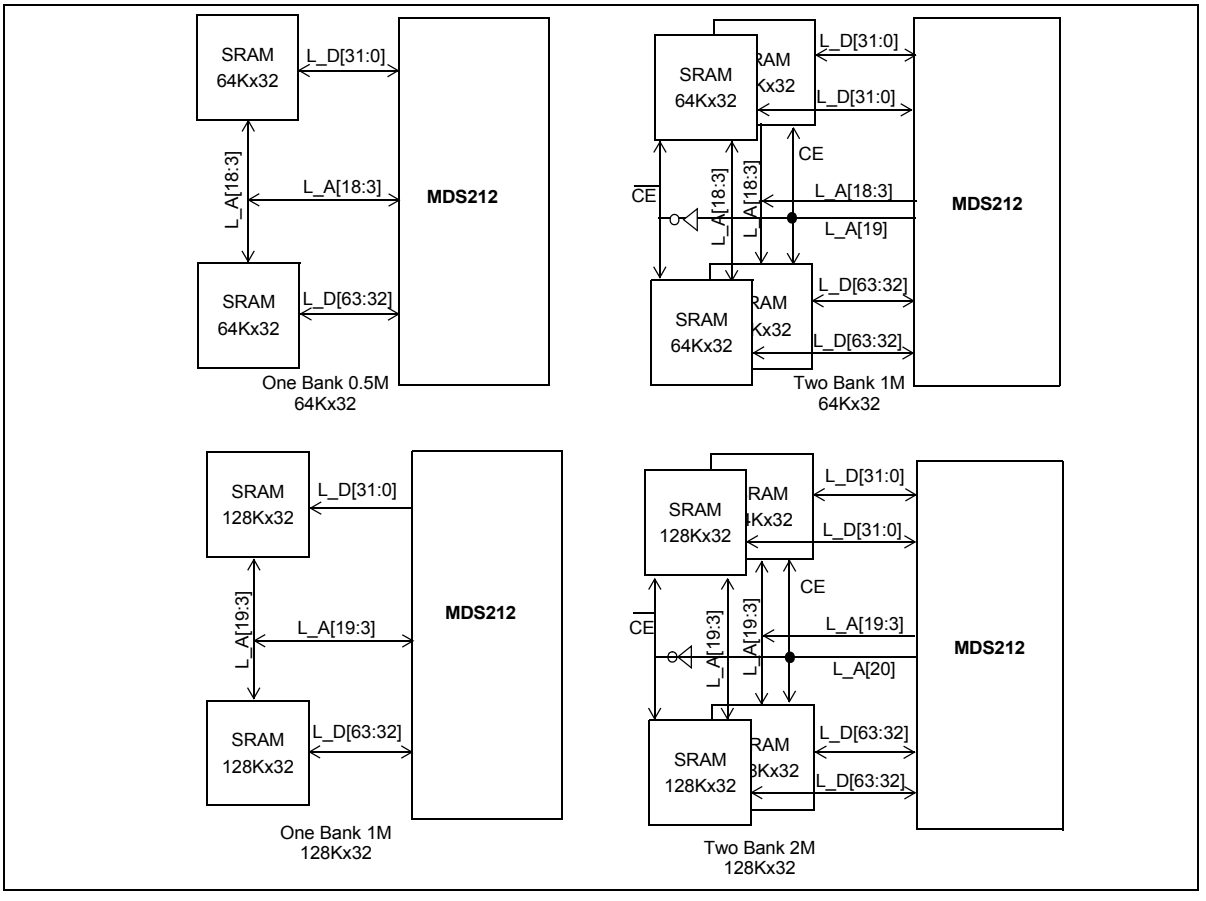


Figure 3 - Frame Buffer Memory Configuration

5.2 Frame Buffer Memory Usage

The MDS212 supports two switching modes: managed and unmanaged. The following tables describe Frame Buffer Memory usage for managed and unmanaged modes of operation, respectively:

Description	Unit Size	Unit Count	Total Size	Reference by
Frame Data Buffer (FDB)	1.5 Kbytes	256 to 1K	384 K bytes to 1.5M bytes	FE[1]
Transmission Queue	4 bytes x 128K to 4 bytes x 1K	48 (4 level priority)	24 Kbytes to 192Kbytes (at 4 level priority)	FE[1]
CPU/HISC Mailing List	32 Bytes to 64 Bytes (Programmable)	128 to 1K	4K bytes to 32 Kbytes (at 32 Bytes each)	CPU, HISC & SE [1]
VLAN Table	8 bytes x 4K	1	32 Kbytes	HISC & SE [1]
VLAN MAC Table	8 bytes to 32 bytes x 2K	1	16 Kbytes to 64 Kbytes	HISC & SE [1]

Note: FE = Frame Engine, SE = Search Engine

Table 2 - Frame Buffer Memory Usage for Managed Mode

Description	Unit Size	Unit Count	Total Size	Reference by
Frame Data Buffer (FDB)	1.5 Kbytes	256 to 1K	384 K bytes to 1.5M bytes	FE
Transmission Queue	4 bytes x 128K to 4 bytes x 1K	48 (4 level priority) 12 (1 level priority)	24 Kbytes to 192Kbytes (at 4 level priority)	FE
HISC Mailing List	32 Bytes to 64Bytes (Programmable)	128 to 1K	4K bytes to 32 Kbytes (at 32 Bytes each)	HISC & SE

Note: FE = Frame Engine, SE = Search Engine
In unmanaged mode, the system does not support VLAN features. Thus, VLAN related tables are not required.

Table 3 - Frame Buffer Memory Usage for Unmanaged Mode

5.2.1 Memory Allocation Of A Managed System

In a managed system, the Frame Buffer Memory is partitioned into five segments: Frame Data Buffers (FDBs), Transmission Queues, Mailing Lists, VLAN, and MCT VLAN Association Tables.

5.2.1.1 Frame Data Buffers

The Frame Data Buffers (FDBs) accommodate the incoming data frames and partition them into data blocks, where each block occupies 1.5K bytes. The number of data blocks in a FDB is configured by setting the value in the register FCBSL[9:0]. Since MDS212 supports up to 2M Bytes memory, the maximum number of data blocks is 1K.

Note: The FDB must start at location 0.

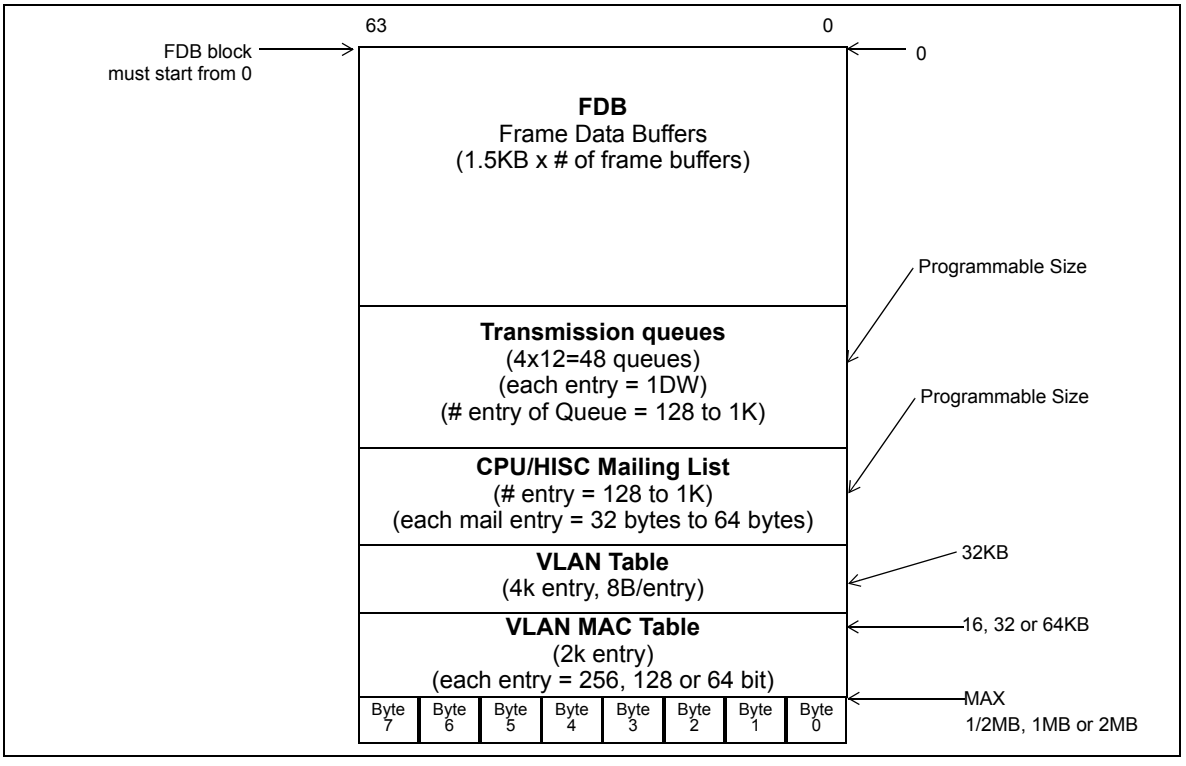


Figure 4 - Memory Map of Managed System

5.2.1.2 Transmission Queues

The Transmission Queues controls the scheduling of the transmission ports, where each of these ports can support up to 4 priorities. There are up to 48 individual Transmission Queues, which represents 4 priorities for each of the 12 ports of the MDS212. The number of priorities is programmable. Thus, the MDS212 may be configured for 12, 24, 36, or 48 Transmission Queues and may support 1, 2, 3, or 4 priority levels, respectively. The size per Transmission Queue is 128, 256, 512, or 1024 entries and may be setup during the initialization phase.

The Search Engine maintains the contents of each queue, where each queue consists of transmission priorities. Each double word (4-bytes) entry contains a FDB handle, which points to the corresponding frame in the buffer.

5.2.1.3 Mailing List

The Mailing List provides a communication channel between the HISC and CPU in managed mode. The size of a mail entry varies, ranging from 32 to 64 bytes, which is determined by the initialization setup. When the CPU or the HISC writes mail, the CPU/HISC can obtain a free mail by the register AFML that contains the addresses of free mail. Conversely, when the CPU or HISC reads its mail, the CPU/HISC accesses the mail by the register AMBX that contains the address of a CPU/HISC mail. All of the mail registers are maintained by the hardware.

5.2.1.4 VLAN Table

The VLAN Table associates the ports to their respective VLANs, using the VLAN ID. The table contains 4K VLAN entries, where each entry contains 8 bytes of information. The size of the VLAN Table is 32KB (4Kx8B). The base address of the VLAN Table is specified by the VIDB in the VTBP bit [5:0].

Note: The VLAN Table must be located at the 32K boundary.

5.2.1.5 VLAN MAC Association Table

The VLAN MAC Table (VLAN MCT) associates each port's MAC address with its respective VLAN. The Table comprises of 2048 entries, one entry per MAC address. Each VLAN MAC entry is mapped to each bit associated with a VLAN specified by the VLAN Index.

The size of the Table is defined by two bits in the VTBP register and depends on the system configuration (e.g. the number of VLANs supported in the system). Each entry may consist of 256, 128 or 64 bits (one bit per VLAN). The total size of the VLAN MAC Table may be 16, 32, or 64KB. The VMACB field in the register VTBP specifies the base address.

Note: The VLAN MAC Table must be located at the 16K boundary.

5.2.2 Unmanaged System Memory Allocation

Since an unmanaged system does not support VLAN operation, the VLAN and VLAN MAC tables are not required. Only the Frame Data Buffers, Transmission Queues, and HISC Mailing Lists are allocated in system memory.

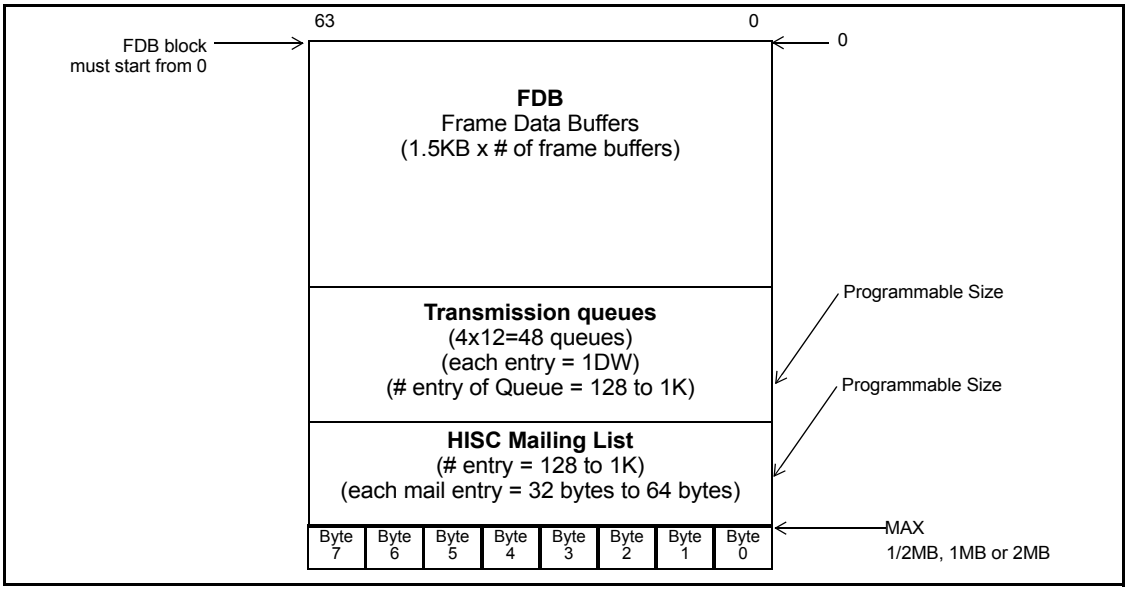


Figure 5 - Memory Map of an Unmanaged System

5.3 The Frame Memory Interface

5.3.1 Local Memory Interface

Each frame within the MDS212 is allocated its own buffer memory. The primary function of the Frame Buffer Memory is to provide a temporary buffering space for both received and transmitted frames, as well as frames waiting in the transmission queue. The actual usage depends on the frame type to be transmitted, either unicast or multicast and the relationship between the source and destination ports. The buffer memory also, contains other control structures including stacks, queues, other control tables. The buffer memory may be configured for 128K, 256K, 512K, 1024K Bytes depending on the application of the system designer. The MDS212 local memory interface supports up to 2M bytes of SBRAM.

The switch manager CPU initializes the local buffer memory during the switch initialization phase/process.

6.0 Search Engine

The Search Engine is responsible for determining the destination information for all packet traffic that enters the MDS212. The results from all address or VLAN searches are passed to the Frame Engine to be forwarded, or on to the HISC block for further processing. Either way, the result 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 one of four Hashing algorithms that can be selected. This provides the capability of changing the search hashing to optimize the hash tables based on the traffic patterns in a given network. For example, if a company gets all their Network Interface Cards (NIC) from one vendor, then the source and destination MAC addresses will have common fields. This can lead to inefficient search hashing. With 4 different hash selections that utilize different parts of the address fields, and can be 8, 9, or 10 bits in length, the hashing algorithm that works best for a user's network can be selected (by testing each hash algorithm).

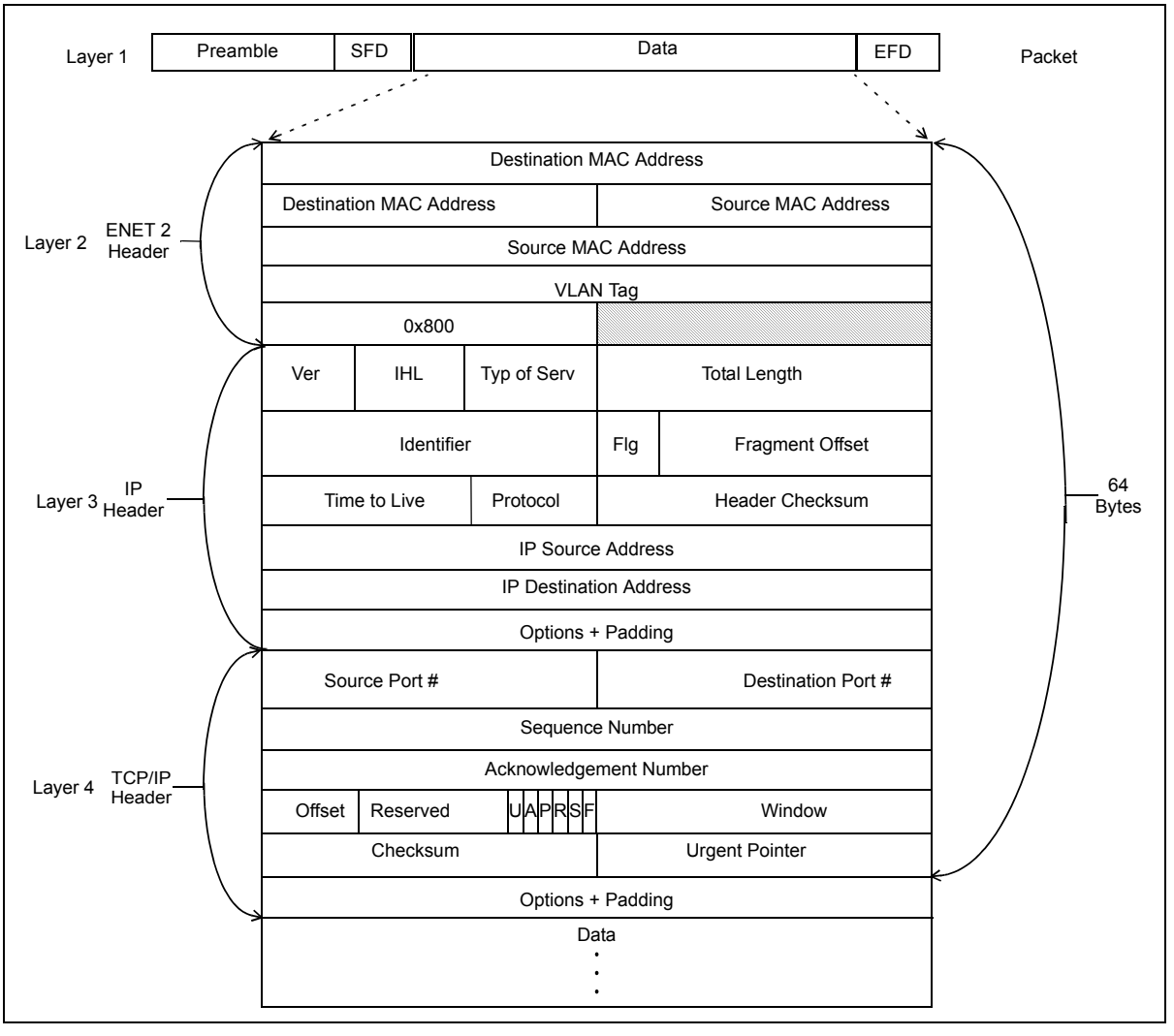


Figure 6 - Typical Packet Header Information

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 VLAN membership and other packet status information.

6.1 Layer 2 Search Process

When the MDS212 is in either a “forwarding” state (able to forward packets) or a “learning” state (able to learn new addresses), the Search Engine is capable of performing address searches. The search process begins when packet header information is transferred to the Search Engine from the Frame Engine.

The Search Engine first checks to determine if the MDS212 is configured to support Virtual Local Area Networks (VLAN). If VLANs are enabled, the Search Engine will search for both the destination MAC address, to get destination resolution information, and the source MAC address, to get the port’s VLAN membership and verify the validity of the port’s VLAN membership. If VLANs are disabled, the Search Engine will search for the destination and source MAC addresses but will not do a VLAN table check.

6.1.1 VLAN Unaware

When VLANs are not enabled or configured, the Search Engine will search the internal switch database memory for an MCT that matches the destination MAC address. When a match is found, the Search Engine will check to ensure that the destination address is not to be filtered before sending a search result message back to the Frame Engine to start the packet forwarding process. At the same time, a search for the MCT that matches the source MAC address is also performed. If no match is found for the source address, then the source MAC address needs to be learned.

6.1.2 VLAN Aware

When VLANs are enabled and configured, the Search Engine will begin searching for the destination MCT and the source MCT. If a matching MCT is found for the source address, then no learning is required, and the Search Engine will check the VLAN membership of the source port. If the source port is a member of the VLAN, and the destination port is also a member of the VLAN, then a normal response message will be passed to the Frame Engine. If the source port is not a valid member of the VLAN, or the destination port is not a member of the VLAN, then the Search Engine will decide to forward the packet or drop the packet depending upon a user defined configuration. Then it will send a message to the HISC to allow the HISC to resolve the issue.

6.2 Address and VLAN Learning

Address learning can be performed by either the HISC or the Search Engine and can be enabled or disabled. The global learning control is set in the Device Configuration Register (DCR2). The Global Learning Disable (GLN) bit controls whether learning is active or disabled, and can be set during initial power up configuration, or by an external CPU before it begins modifying the SDM. It is necessary for an external CPU to disable learning before updating or modifying MCT entries. This prevents the internal learning process from modifying MCT entries without the CPU’s knowledge.

When learning is globally enabled, by the Search Engine not finding a match 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 MDS212 chips are connected, and configured to operate 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 MDS212. 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 MDS212).

6.3 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 MDS212 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. Three separate flood control fields allow the user to specify flooding limits for:

- Unicast to Multicast (flooded) packets per source port
- Unicast to CPU packets per chip
- Multicast to CPU packets per chip

During the time base period, three separate counters at each port count the number of packets meeting the flood control types. Once a counter exceeds the allowed quantity, the Search Engine will then discard the packet and any other packets of that type that enter the port during the remainder of the time base period. When the time base period is completed, the three flood counters at each port are reset, and the counting process starts over.

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

6.4 Packet Filtering

Packet filtering occurs during the address search phase. For static source or destination MAC address filtering, there is a corresponding bit in the MCT entry that tells the Search Engine that the source or destination packet is to be filtered.

When a match is found to a destination MAC address search, the “Destination Filter” (D) field in the MCT is checked to determine if the destination address is to be filtered. If “D” is asserted, the Search Engine discards the packet by sending a message to the Frame Engine telling it to release the Frame Control Buffer (FCB) where the packet has been stored in the frame buffer memory. The packet thereby deleted from memory.

When a match is found to a source MAC address search, the “Source Filter” (F) field in the MCT is checked to determine if the source address is to be filtered. If “S” is asserted, then the Search Engine discards the packet by sending a message to the Frame Engine telling it to release the FCB for the packet.

6.5 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.

Some entries may be static and not subject to aging. These MCT entries have a status field that identifies them as being static, and will therefore always have their aging flag asserted. The network manager, using Zarlink software, establishes static entries during a switch configuration session.

6.6 IP Multicast

The Search Engine supports the ability of the MDS212 to provide IP Multicast by identifying Internet Group Multicast Protocol (IGMP) packets when parsing the packet header information provided by the Frame Engine.

IGMP packets are identified when the source MAC address is 01-00-5E-xx-xx-xx and the Protocol field has the value of 2, or when the source IP address is 224.0.0.x.

When an IGMP packet is identified, the Search Engine searches for the source address MCT entry, and then passes a message to the HISC to allow it to setup or tear down the IP Multicast session. IP Multicast sessions are treated as VLANs and use one of the 256 regular VLAN entries.

7.0 The High Density Instruction Set Computer (HISC)

7.1 Description

The High Density Instruction Set CPU (HISC) is specifically designed to implement highly efficient management functions for the MDS212 switching hardware, minimizing the management activity intervention during frame processing. The HISC services management requests based on an event- driven approach. Management requests can be generated from either the management CPU or the switching hardware. 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 CPU interface and the switching hardware.

7.2 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 MDS212 running at 100MHz, the HISC can produce up to 300MIPs processing power.

7.3 HISC Operations

With an event-driven operation model, upon the request from either the Search Engine or external management CPU, the HISC dynamically manages and maintains the Switch Database including MAC address entries, VLAN, and MAC-VLAN Association Tables. The HISC also provides an external management CPU a high-speed data communication interface, so management packets can be transmitted to or received from the network.

In general, the service request is received from one of four different sources:

- Messages from the management CPU
- Requests from the switching hardware (Search Engine)
- Real time clock
- Interrupts to the management CPU

The HISC performs the following major operations:

- Resource initialization
- Resource management
- Switching database management
- Send and receive frames for management CPU

7.3.1 Resource Initialization

The HISC initializes all internal data structures including the mail box and switching database data structures, which are used by the management CPU, HISC, and switching hardware.

7.3.2 Resource Management

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

7.3.3 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 or management CPU. Generally, the Search Engine performs the learning of new MAC addresses identified in the packet streams. For a single MDS212 system, the HISC simply informs the management CPU regarding the newly learned MAC addresses.

The HISC may also create, delete, or modify the MAC address entries based on the requests from the management CPU. For a multi-DS system, the HISC is response for synchronizing the switching databases. In addition to the MAC address entries, the HISC also maintains the following database information required for switching:

- Create, delete and modify VLAN table in the switching database.
- Create, delete and modify MAC VLAN table in the switching database.
- Create, delete and modify IP Multicast entries in the switching database.

7.3.4 Send And Receive Frames For Management CPU

The HISC delivers BDPU, SNMP, and other frames to and from the management CPU. In unmanaged mode, the HISC also responds to interrupts destined to the management CPU.

7.4 Communication Between HISC and Switching Hardware

High-speed communication channels are required to provide fast message deliveries between the HISC and switching hardware. Two high-performance FIFOs provide the required communication channels. They are between the HISC and the Frame Engine, and between the HISC and Search Engine.

7.4.1 Communication Between Search Engine And HISC

The first 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 for 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.

7.4.2 Communication Between HISC and Frame Engine

The second high-speed FIFO is used by the HISC for sending data transfer requests to the Frame Engine. Whenever a packet-forwarding request is received from the management CPU, the HISC forwards the request to the Frame Engine via the FIFO. To alleviate the workload of the management CPU, certain management packets can be processed by the HISC, and then forwarded to the Frame Engine for transmission via the FIFO.

7.5 Communication Between Management CPU and HISC

The HISC serves as an intermediary communication channel between the switching hardware and the external management CPU. There are two communication mechanisms provided for messages exchanged between the management CPU and HISC.

7.5.1 CPU-HISC Communication Using Queues

The first communication mechanism is a pair of Input and Output Queues between the HISC and management CPU. The management CPU input/output queue is a very efficient mechanism for a single 32-bit data exchange between the HISC and management CPU. In general, a management frame, i.e., Bridged Data Protocol Units (BDPU), is forwarded directly from the HISC to the management CPU via the CPU Output Queue. Small management requests, less than 24 bits, are delivered to the HISC via the CPU Input Queue.

7.5.2 Mailbox

The second communication mechanism is a hardware mailbox that can support variable size messages, exchanged between the management CPU and the HISC. A major use of the mailbox is to exchange information required for updating the switching database.

7.5.2.1 CPU-HISC Mail

When the management CPU sends a mail message to the HISC, the CPU acquires an address of a free mail from the free mail list (via register AFML), it writes the mail content to the given memory address. Afterward, it sends the mail to the HISC via the Mailbox Access (AMBX) Register. Whenever a management mail message is received, an event is generated to inform the HISC to process the mail message.

7.5.2.2 HISC-CPU Mail

When a mail message arrives from the HISC, the mailbox hardware sends an interrupt, namely “Mail Arrive” (MAIL_ARR), to the CPU. The CPU can then access the mail via the Mailbox Access Register (AMBX). At this point, the CPU reads the mail handle and retrieves the contents of the mail from the AMBX Register.

8.0 The XPipe

The XPipe provides a high-speed link between systems utilizing two MDS212 devices. The XPipe incorporates a 32-bit-wide data pipe, with a high-speed point-to-point connection, and a full-duplex interface betweendevices. While operating at a 100MHz, this interface can provide 3.2G bits per second (Gbps) of bandwidth per pipe in both directions.

8.1 XPIPE Connection

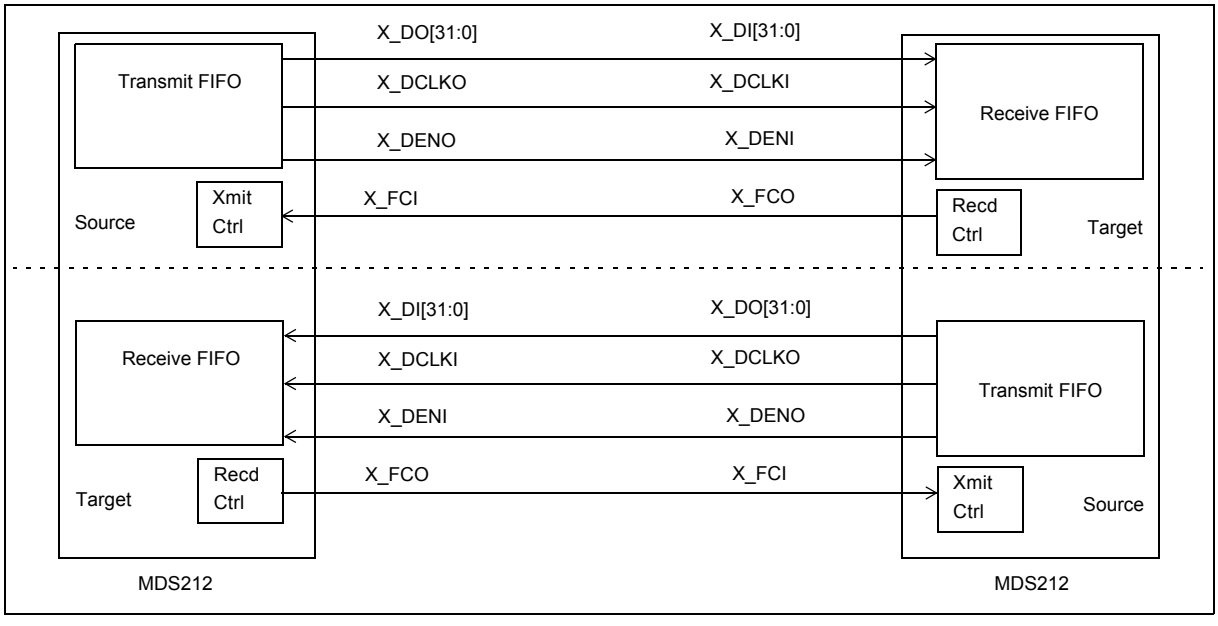


Figure 7 - XPipe System Block Diagram for the MDS212

The XPipe interface employs 32 data signals and three control signals for each direction. The pin connections between two MDS212 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_DCLKO), provides a synchronous clock to sample the data signals at the target device. The source device provides the Transmit Data Enable signal (X_DENO) 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 are described in the XPipe Timing (section Section 8.2 “XPipe Timing” ”).

Signal Name		Description
Source End	Target End	
X_DO[31:0]	X_DI[31:0]	32-bit-wide Transmit Data Bus - Includes an XPipe Message Header and followed 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 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_FCO signal to LOW. The source device will stop further transmission until the X_FCI signal asserts the XON state, which is an active HIGH (See Table 4).

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.

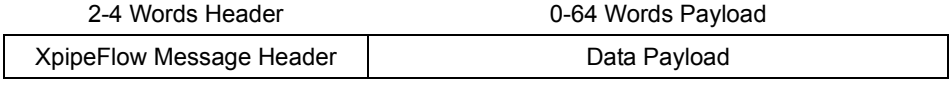


Figure 8 - XPipe Message Header

8.2 XPipe Timing

The source device generates the X_CLKO 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 9.

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).

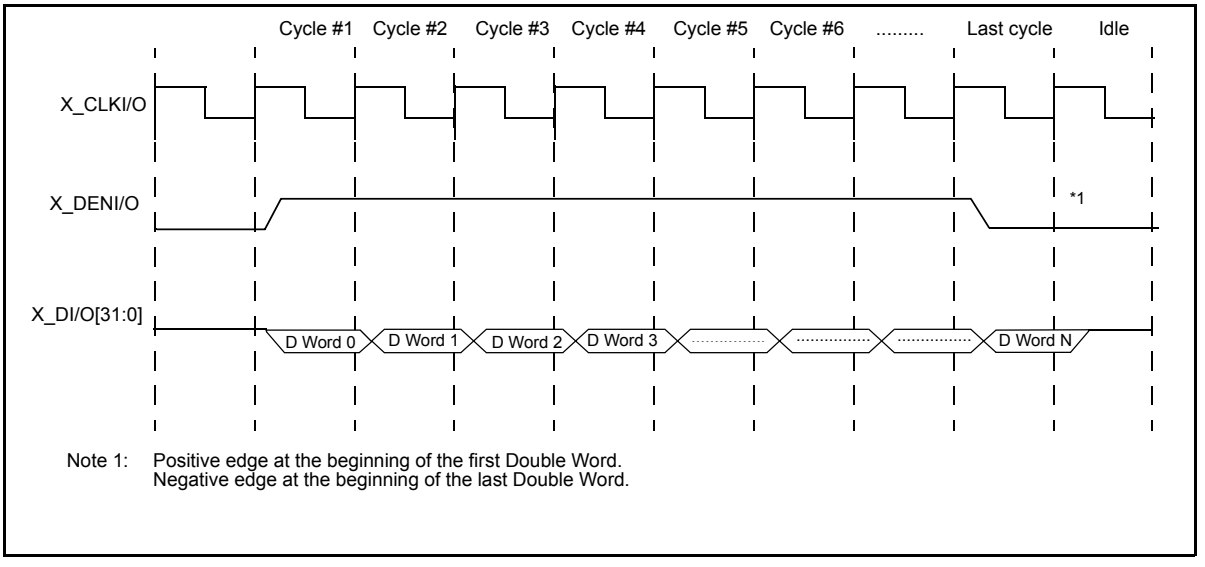


Figure 9 - Basic Timing Diagram of XPipe

9.0 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. 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.

9.1 Reduced MII (RMII)

The MDS212 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 [2] MII interface. Under IEEE 802.3u [2], 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. The MDS212 offer 12 or 24 ports, in one or two devices respectively. At 6 pins per port and 1 pin per switch ASIC, the RMII specification saves 119 pins plus the extra power and ground pins to support those additional pins for a 12 port switch ASIC. 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 to be identical to that defined in IEEE 802.3u [2].

The RMII supports both 10 and 100Mbps 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 10Mbps and 100Mbps 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	Direction (with respect to the PHY)	Direction (with respect to the MAC)
REF_CLK	Input or Output	Synchronous clock reference for receive, transmit and control interface
M[0:11]_CRS_DV	Input	Carrier Sense/Receive Data Valid
M[0:11]_RXD[1:0]	Input	Receive Data
M[0:11]_TX_EN	Output	Transmit Enable
M[0:11]_TXD[1:0]	Output	Transmit Data
M[0:11]_RX_ER	Input (Not required)	Receive Error

Table 5 - RMII Specification Signals

10.0 The Control Bus

The CPU Interface, or Control Bus, provides the communication path between the system CPU and all other key components within the switch (i.e. the HISC). It operates in two modes: managed mode, where it utilizes an external CPU, and unmanaged mode, where an external CPU does not exist.

In Managed mode, the CPU Interface provides the communication path between the systems' external CPU and the HISC, Frame Buffer Memory (SRAM) or another MDS212. See Figure 10.

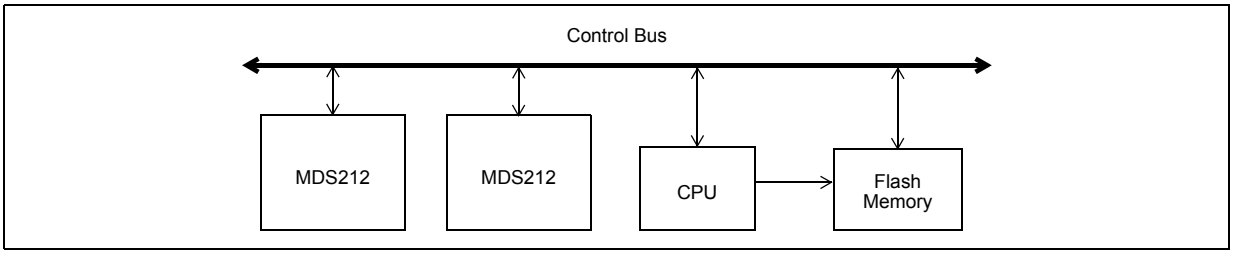


Figure 10 - CPU Interface Configuration in Managed Mode

In unmanaged mode, the CPU Interface provides the communication path between the Switch Devices

and Flash Memory, and between any two MDS212 Switches. See Figure 11.

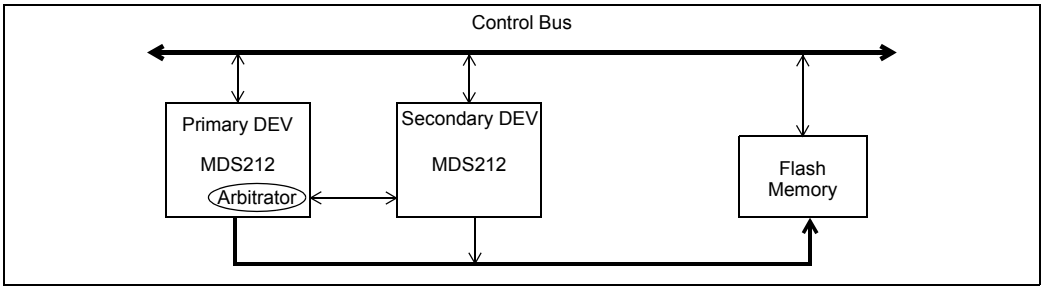


Figure 11 - Control bus Configuration in Unmanaged Mode

10.1 External CPU Support

The control bus comprises of a 32-bit wide CPU bus and supports Big and Little Endian CPU byte ordering. The standard microprocessors supported include:

- Intel 486 CPUs
- Motorola MPC860 and 801 CPUs.
- Intel i960Jx CPU
- MIPS processor with minimum conversion.

10.1.1 Power On/Reset Configuration

On power-up, the five Bootstrap bits, on Table 6, are used.

10.1.2 CPU Bus Clock Interface

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

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_SWM	1	Switch Mode (only in Managed Mode) 0=Managed Mode 1=Unmanaged Mode
BS_PSD	1	Primary Device Enable (only in Unmanaged Mode) 0=Secondary Mode 1=Primary Mode (The arbiter is activated in the chip with Primary Device.)
BS_RDYOP	1	Option of merger the P_RDY# and P_BRDY# 1=Seperated P_RDY# and P_BRDY# pins 0=merged P_RDY# and P_BRDY# pin

Table 6 - Bootstrapping Options

10.1.3 Address And Data Buses

The CPU Interface provides separate, non-multiplexed 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 for managed and unmanaged modes. The address bus supports 10 [10:1] address bits for managed and unmanaged modes. Each device occupies 2048 bytes of Input/Output space.

10.1.4 Bus Master

The nomenclatures “Master” and “Slave” refer to the device that possesses the CPU Interface, or Control Bus, 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.

In managed mode, the systems’ external CPU is the permanent master device. All other devices (e.g. the MDS212) are designated as slave devices only. In unmanaged mode, the arbiter (located within the primary device) selects one of the devices as the Master.

Note: In unmanaged mode, 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.

10.1.5 Input/output Mapped Interface

The systems' external CPU accesses the switch devices' local memory using single-read/write or burst- read/write I/O cycles. Burst I/O operations with auto address incrementing uses a 32-byte write data buffer and a 32-byte cache read data buffer.

10.1.6 Interrupt Request

The CPU Interface accepts an Interrupt Request (IRQ) from each device connected to the interface, and supports centralized interrupt arbitration and vector response. The interrupt output is an open-drain option with programmable polarity.

10.2 Control Bus Cycle Waveforms

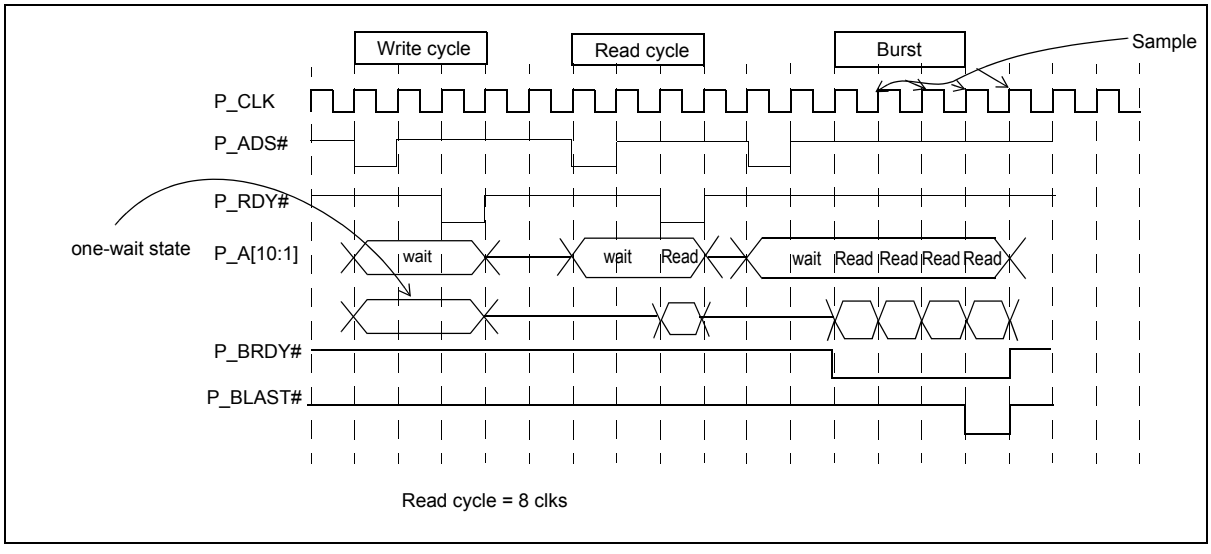


Figure 12 - Control Bus I/O and Flash Bus Access Operations

10.3 The CPU Interface in Unmanaged Mode

In unmanaged mode, the HISC processor of the Master device communicates with the slave device as a CPU function. Three registers and one flag are used to communicate between the HISC processor and the CPU Interface.

10.3.1 Arbiter

The arbiter of the XpressFlow MDS212 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 Figure 13).

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 MDS212 devices and one CPU, can operate on the CPU Interface at the same time.

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

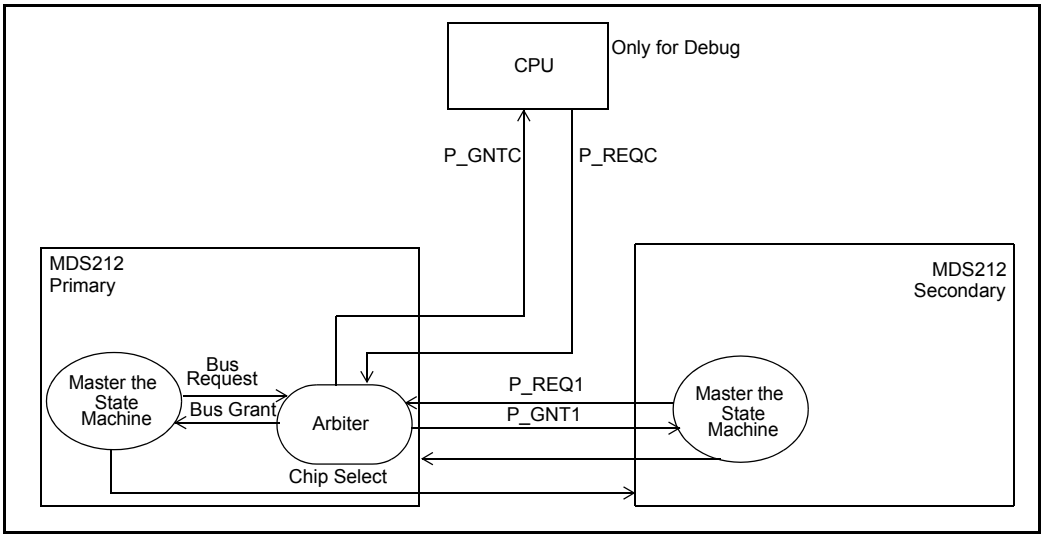


Figure 13 - Block Diagram of the Arbiter

Note: In unmanaged mode, the CPU is used only for debugging purposes and cannot be involved in switching decisions or management activities.

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.

10.4 CPU Interface In Managed Mode

The CPU Slave State Machine (SSM) accepts Address Strobe (P_ADS#), Chip Select (P_CS#), and Bus- Data Ready (P_RDY#) signals as ready state signals of a CPU cycle.

10.4.1 CPU Access

The 32-bit CPU bus interface supports both Big and Little Endian CPUs. The difference between Big and Little Endian is the byte swapping when CPU write data to external memory. Table 7 below summarizes the byte swapping operation and Figure 14 illustrates an example of bytes swapping.

If using Little Endian	Bit[1] must be ‘0’ for register of MWARS, MRARS, MWARB, MRARB	No byte swapping for CPU data write in or read out to/from MWDR, MRDR registers
If using Big Endian	Bit[1] must be ‘1’ for register of MWARS, MRARS, MWARB, MRARB	Automatic Byte swapping for CPU data write in or read out to/from MWDR, MRDR registers

Table 7 - Little and Big Endian Byte Swapping Operation

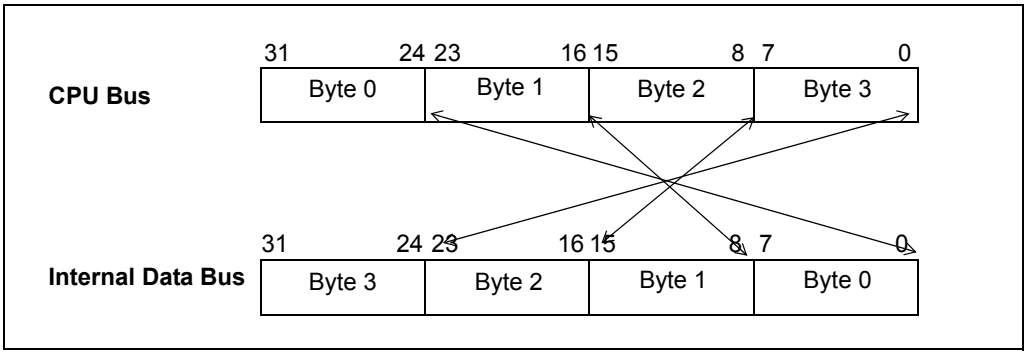


Figure 14 - An example of byte swapping

11.0 The LED Interface

11.1 LED Interface

The MDS212 LED interface supports the status per port in a serial stream that may be daisy-chained to connect two MDS212 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 in the MDS212, 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.

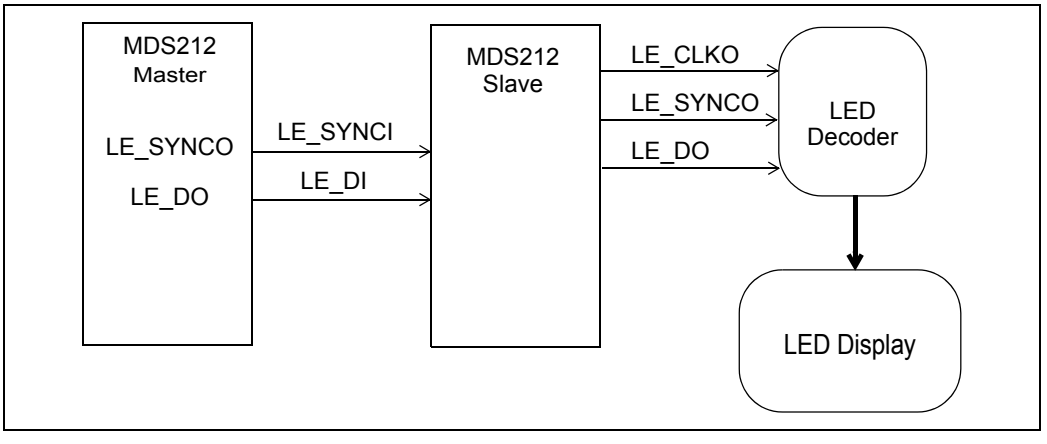


Figure 15 - LED Interface Connections

To provide the port status information from our MDS212 chips via a serial output channel, five additional

pins are required.

- LE_CLKO - at 12.5 MHz
- LE_SYNCI/O - a sync pulse -- defines the boundary between frames
- LE_DI/O - acontinuous 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 MDS212 is transmitted to an external decoder via a serial output channel. In the MDS212, we support cascading of this serial output channel between two devices. One MDS212 is configured as the master, this initiates the start of LED information frames, and serializes information bits. The MDS212 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 5. Figure 15 shows 2 LED interfaces are cascaded and the connections between the MDS212s, LED decoder and LED display.

11.1.1 Function Description

The LED interface employs the following signals:

Signal Name		Description
Master Device	**Slave Device**	
	LE_CLKO	LED Clock-Synchronous LED clock provided by the slave device to LED decoder at the system clock divided by 8(~12.5Mhz).
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 repeat once every frame time.

Table 8 - LED Signal Names and Descriptions

11.1.2 Port Status

In the MDS212, 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 12 ports of the MDS212, 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.

11.1.3 LED Interface Time Diagram

The Master needs to shift out (16)*8 status bits periodically((12 port status +4 reserved)*8). Thus, slave needs to shift out (16)*8 + (16)*8 status bits, which includes the status of the master device and itself.

The status of each port will be sampled by the LED State Machine every 20.5μs, the time period of the frame. That is, each LED data frame length equals (256)X 80nsec. Each frame is divided into two sub- frames: a master and a slave sub-frame. Furthermore, each sub-frame is partitioned into 16 slots (12 MAC ports and 4 reserved slots) and each slot will carry 8 status bits. Figure 16 shows the signal from the slave chip to LED decoder.

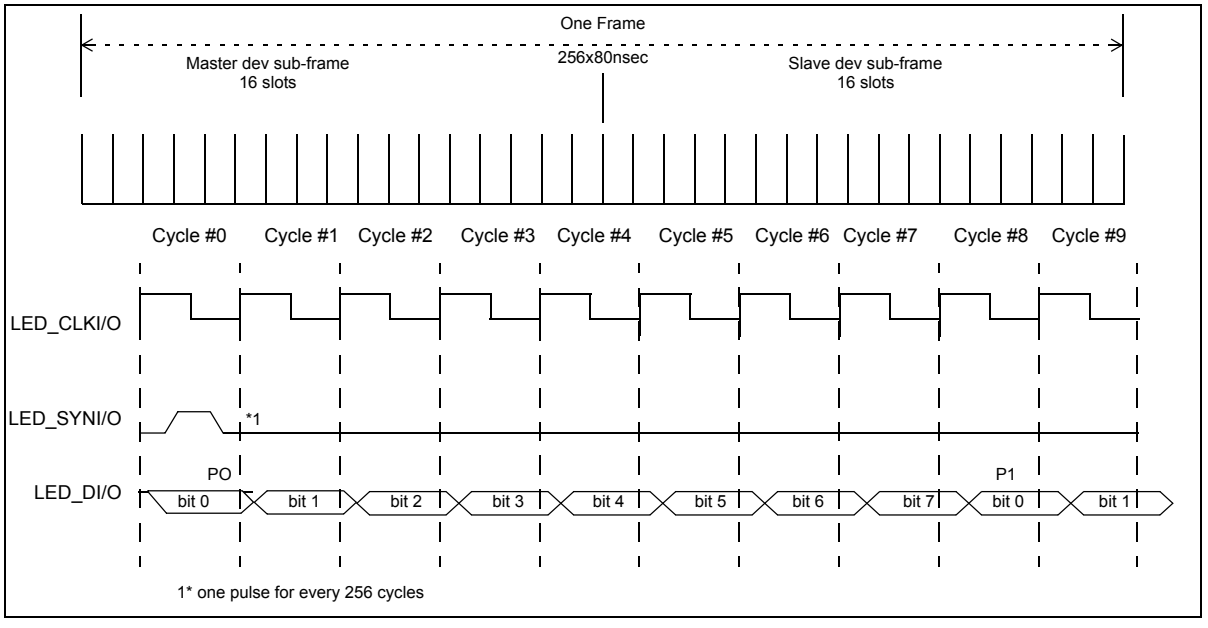


Figure 16 - Time Diagram of LED Interface

12.0 Data Forwarding Protocol and Data Flow

12.1 Data Forwarding Protocol

12.1.1 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 service discipline is round robin for the 100/10Mbps ports. 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)

12.1.2 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 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.

There are four transmission-scheduling queues for each port, one for each of the four classes of priority. 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.

12.1.3 Multicast Frame Forwarding

After the reception of the switching response, a job is sent to the Transmission Scheduling queues of the destination ports for local switching. However, for remote switching, one copy of the frame will be forwarded to the remote device in 128-byte granules via the XPipe. This copy of the frame will be sent to the frame data buffer. The Frame Engine, after the successful reception of this frame, will put jobs in the Transmission Scheduling queues of the destination ports of its device. When the TxFIFO is ready to receive the frame (same as the conditions stated in unicast frame forwarding section), the TxDMA will forward the frame from the FDB to the destination ports in granule form. The maximum size of a granule is 128 bytes.

12.2 Flow For Data Frame

The following subsections describe the flow of information during transfers of data frames, both unicast and multicast.

12.2.1 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 MAC address cannot be resolved by the Switch Engine, the HISC and/or the CPU are queried to resolve the address. For an unknown destination MAC, the frame will flood the frame into the source VLAN domain.

12.2.2 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 address resolution cannot be completed by the Switch Engine, the HISC and/or the CPU are 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. To reduce the latency, the frame is not stored in the FDB of the remote device again.

12.2.3 Multicast Data Frame

In this scenario, we assume that the multicast frame involves both local and remote ports. The received multicast frame is written to the local FDB by the Frame Engine. After resolving the destinations, the Switch Engine provides local destination port addresses and remote port addresses to the Frame Engine. If the address resolution cannot be completed by the Switch Engine, the HISC and/or the CPU are queried.

The Frame Engine pushes the jobs to the corresponding transmission queues (per job per local port). When a local port is ready for this multicast frame, the Frame Engine moves the frame to the corresponding TXFIFO. There is a counter to track of the number of copies to be sent. The number is provided by the Search Engine and the frame engine keeps track of this counter. When a frame is sent, the counter is decreased by one. The FDB will be released when the counter becomes zero.

When the destination ports include remote ports, the frame is transferred over the XPipe to the remote Frame Engine, which writes a single copy of it into the remote FDB. That is, we use double store-and- forward for remote multicast. After receiving the whole frame, the remote Frame Engine utilizes the control information in the internal header, which indicates the associated destination ports in the remote device to push the jobs into the corresponding transmission queues. When a port is ready for this multicast frame, the Frame Engine moves the frame to the corresponding TXFIFO. Similarly, the Frame Engine also keeps track of the number of copies of a frame to be sent and releases the frame when the counter is reduced to be zero.

12.3 Flow For CPU Control Frame

In a managed system, the CPU may transmit or receive CPU control frames, e.g., Protocols, SNMP frames to/from a MAC port via a CPU unicast frame. On the other hand, the CPU may receive a multicast frame from a MAC port. Moreover, the CPU can transmit a multicast frame to multiple ports. The following four scenarios illustrate the four possible forwarding flows.

12.3.1 CPU Transmitting Unicast CPU Frame

The CPU initiates Unicast control messages, by first writing the frame into the FDB, and then sending a message to the HISC. The HISC forwards a switch response to the Frame Engine, which transmits the frame to the destination MAC port. After receiving switch response, Frame Engine performs the same unicast forwarding as for unicast data frame. Refer to previous subsection for unicast data frame mechanism.

12.3.2 CPU Transmitting Multicast CPU Frame

When the CPU sends a multicast control message to the ports, the CPU first writes the frame to the local FDB. The CPU then sends a message to the HISC, which provides a switch response message to the local Frame Engine. After receiving the switch response, the Frame Engine performs the same multicast forwarding as for the multicast data frame. Refer to previous subsection for multicast data frame mechanism.

12.3.3 CPU Receiving Unicast Frame

The receiving CPU frame is moved to the FDB and the Frame Engine forwards a switch request, including the frame header, to the Search Engine. After the Search Engine decodes the header and determines to forward it to the HISC to process, HISC informs the CPU via a mail, which indicates the handle of the FDB. The CPU then obtains the frame through the MDS212. After reading the frame from the FDB, the CPU will inform the HISC to release the FDB. Finally, the HISC passes the release command to the Frame Engine to release the FDB accommodated CPU frame.

12.3.4 CPU Receiving Multicast Frame

The MDS212 is capable of receiving a multicast packet for a combination of local or remote ports and the CPU. In this case, the received frame includes multicast destination ports on a remote device, and also the CPU. The Search Engine moves the multicast frame to the FDB and then forms a switch request including the frame header, which it sends to the Search Engine. Since the frame involves the CPU, the Search Engine passes the request to the HISC for further processing. The HISC informs the CPU via a mail, which indicates the handle of the FDB. In parallel, the Search Engine sends back a switch response and asks the Frame Engine to forward the frame to the destinations ports. The Frame Engine will perform the same multicast forwarding as mentioned above.

The CPU reads the frame from the FDB via the MDS212. The CPU will then inform the HISC to release the FDB. Finally, the HISC passes the release command to the Frame Engine to release the FDB accommodated CPU frame.

Note: The Search Engine will not release the FDB until it receives the release signal from the HISC and the counter is reduced to zero. This occurs when the CPU all of the ports have read the frame.

13.0 Port Mirroring

13.1 Features

The received or transmitted data of any 10/100 port in any MDS212 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 MDS212 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.

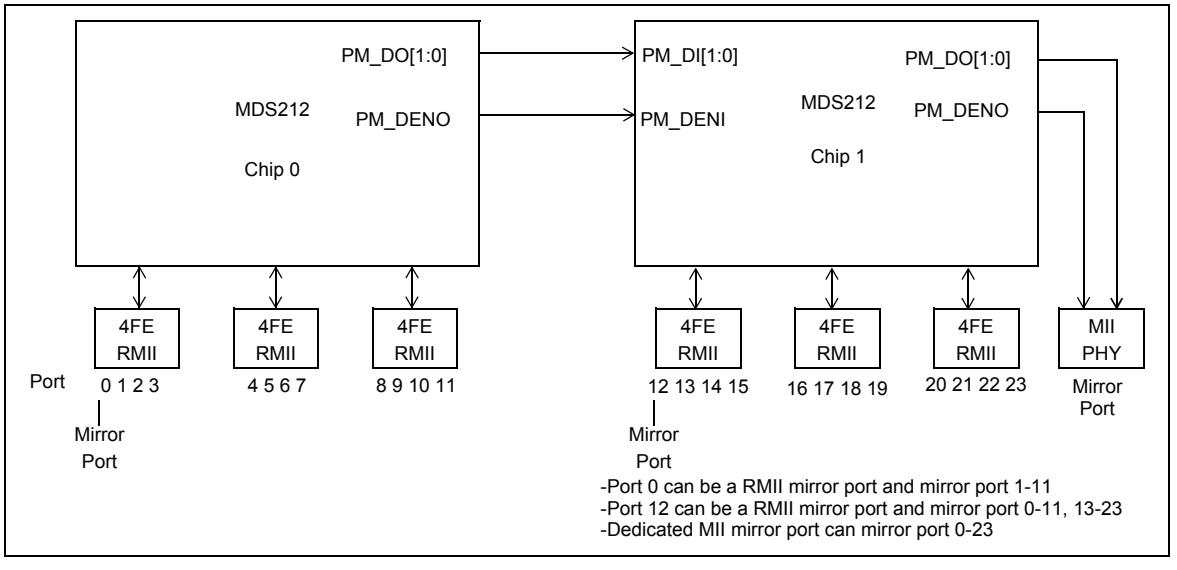


Figure 17 - Configuration of Mirror Port for MDS212

13.1.1 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 MDS212.
- PM_DENI Port Mirroring Data Enable signal for PM_DI Input Provide Data Enable signal for PM_DI signals

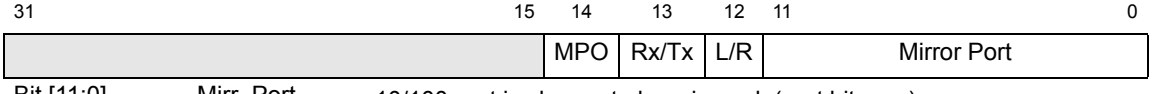
PM_DENI	Port Mirroring Data Enable signal for PM_DI Input Provide Data Enable signal for PM_DI signals
PM_DO[1:0]	Port Mirroring Output Data Bit [1:0] Transmit the mirrored data signal to remote MDS212.
PM_DENO	Port Mirroring Data Enable Output. Provide Data Enable signal for PM_DO signals

Refer to Figure 17 for connecting above pins.

13.1.2 Setting Register For Port Mirroring

APMR register controls the mirrored port and the designated mirroring port. The definition of the register is shown in section 13.1.2.1 “APMR- Port Mirroring Register” .

13.1.2.1 APMR- Port Mirroring Register



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 that at most one of bits in Bit[11:0] can be set to 1.

Bit [13]	Rx/Tx	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 [31:15]	Reserve	

The following examples, based on the configuration of Figure 17, illustrate how to set the APMR 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]=0Local mirrored port

Set APMR[13]=0Transmission mirroring

Set APMR[14]=1Port 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]= 0x002Mirrored port = 1

Set APMR[12]=0 Local mirrored port

Set APMR[13]=1Receiving mirroring

Set APMR[14]=0Port 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 careBit[13] has meaning only in the chip of mirrored port

Set APMR[14]=1Port 13 is the mirroring port

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

For Chip 0

Set APMR[11:0]= 0x002Mirrored port = 1

Set APMR[12]=0 Local mirrored port

Set APMR[13]=1Receiving mirroring

Set APMR[14]=0Port 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 careBit[13] has meaning only in the chip of mirrored port

Set APMR[14]=0Port 13 is not the mirroring port

Note: The CPU needs to find out the speed of the mirrored port and configures the mirroring port to the same speed.

14.0 Virtual Local Area Networks (VLAN)

14.1 Introduction

A Virtual LAN (VLAN) is a logical, independent workgroup within a network. The members in this workgroup communicate as if they are sharing the same physical LAN segment. VLANs are not limited by the hardware constraints that physically connect traditional LAN segments to a network. As such, VLANs can define a network into multiple logical configurations.

14.2 VLAN Implementation

The MDS212 based VLAN implementation allows up to 256 VLANs in one switch. By using explicit or implicit VLAN tagging and the GARP/GVRP protocol (defined in IEEE 802.1p and 802.1Q), VLANs may span across multiple switches. A MAC address can belong to multiple VLANs, and a switch port may be associated with multiple VLANs.

14.2.1 Static Definitions Of VLAN Membership

The MDS212 defines VLAN membership based on ports. Port based VLANs are organized by physical port numbers. For example, switch ports 1, 2, 4, and 6 can be one VLAN, while ports 3, 5, 7, and 8 can be another VLAN. Broadcasts from servers within each group would only go to the members of its own VLAN. This ensures that broadcast storms cannot cause a network melt-down due to traffic volume.

14.2.2 Dynamic Learning Of VLAN Membership

While port based VLAN only defines static binding between a VLAN and its port members, the MDS212's forwarding decision needs to be based on the following:

- A destination MAC address and its associated port ID for a unicast frame, or
- The associated VLAN of a source MAC address, if the destination MAC address is unknown or it is a multicast/broadcast frame. To make valid forwarding and flooding decisions, the MDS212 learns the relationship of the MAC address to its associated port number and VLAN ID and builds up the internal Switching Database at run-time for further use.

14.2.3 Dynamic Learning Of Remote VLAN

In addition to adding and deleting VLAN member ports through network management tools statically, a MDS212 based switch can also support GVRP (GARP VLAN Registration Protocol). GVRP allows for dynamic registration of VLAN port members within a switch and across multiple switches. In addition to supporting the dynamic update of registration entries in a switch, GVRP is also used to communicate VLAN registration information to other VLAN-aware switches, so that a VLAN member can be covered by a wide range of switches in a network. GVRP allows both VLAN-aware workstations and switches to issue and revoke VLAN memberships. VLAN-aware switches register and propagate VLAN membership to all ports belonging to the active topology of the VLAN.

14.2.4 MDS212 Data Structures For VLAN Implementation

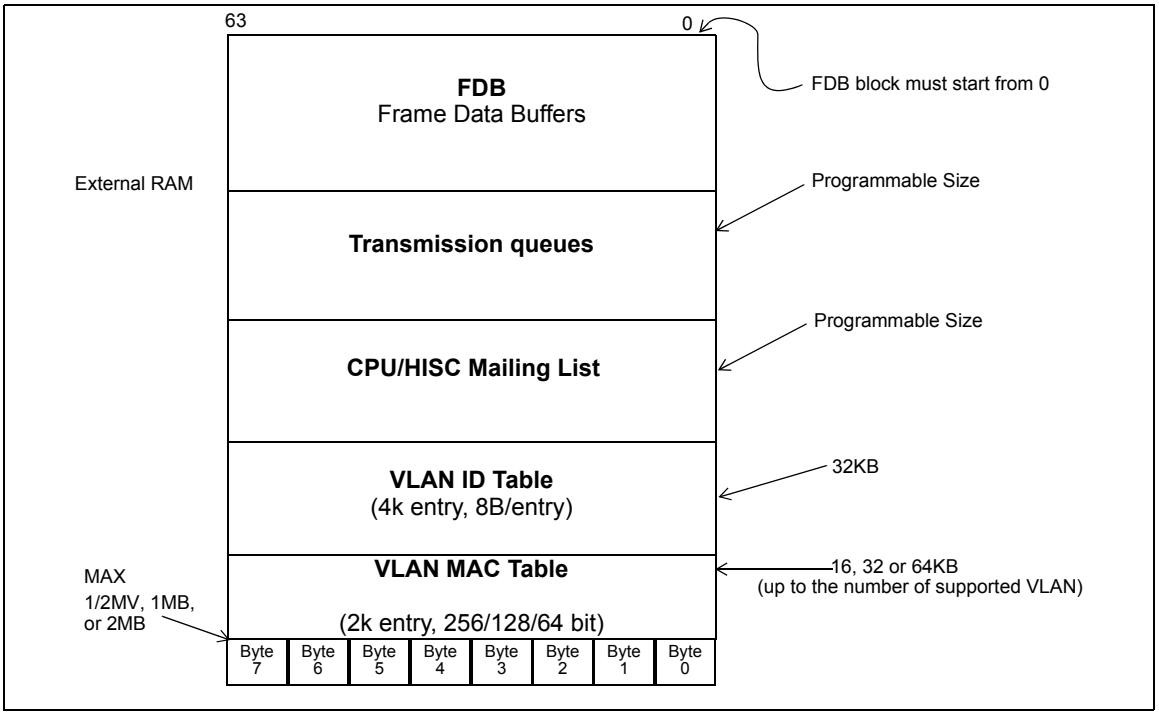


Figure 18 - Data Structure Diagram

14.2.4.1 VLAN ID Table

The VLAN ID Table is used by Search Engine for unicast frames. The base address of this table is specified by VIDB subfield in BIT[5:0] of VTBP register. The contents of this table are set up by the MDS212's microcode through the command of CPU software at the time of VLAN creation and deletion. The VLAN ID Table covers the entire 4K VLAN ID space, and is used by the Search Engine to map the VLAN ID into an internal VLAN Index. It also includes port membership and port tagging information for each VLAN. Each VLAN ID entry is 8 bytes long, and the total size of the VLAN ID Table is 32KB. The VLAN ID table must be located at the 32K boundary.

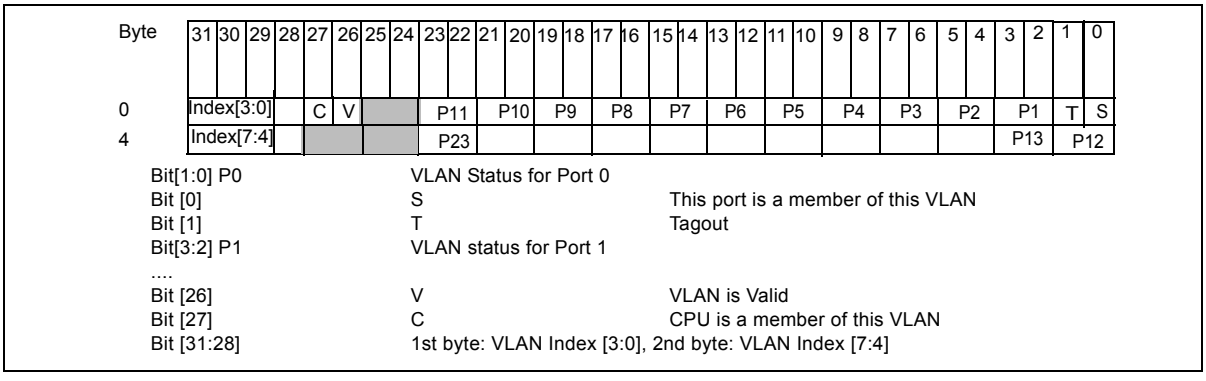


Figure 19 - VLAN ID Table

Note: P0 to P11 are used to identify the ports on the first chip, while P12 to P23 are used to identify the ports on the second chip.

14.2.4.2 VLAN MAC Table

The size of this table is defined by VLMS subfield in BIT[8:7] of VTBP register. The base address of this table is specified by VMACB subfield in BIT[15:9] of VTBP register. The VLAN MAC Table contains all associated VLANs for each MAC Address learned by MDS212, and is used by the software to keep track of every MAC and its associated VLANs. The contents of this table are set up by the Search Engine at the reception of incoming frames, if the Search Engine is not fully occupied. When the Search Engine is too busy handling frame forwarding decisions, microcode in the HISC engine will be assigned the setup new MAC to VLAN associations. Rows in this table can be cleared up by microcode through a CPU software command during VLAN deletion or port link down. A row in this table will be cleared and a new bit set up by the MDS212's microcode, when the port change of a MAC address is detected.

There are a total of 2K entries in this table, one entry per MAC. Each entry may consist of 256, 128 or 64 bits, one bit per VLAN. The total size of the VLAN Table may be 64, 32, or 16KB. This table must be located at the boundary of its own table size.

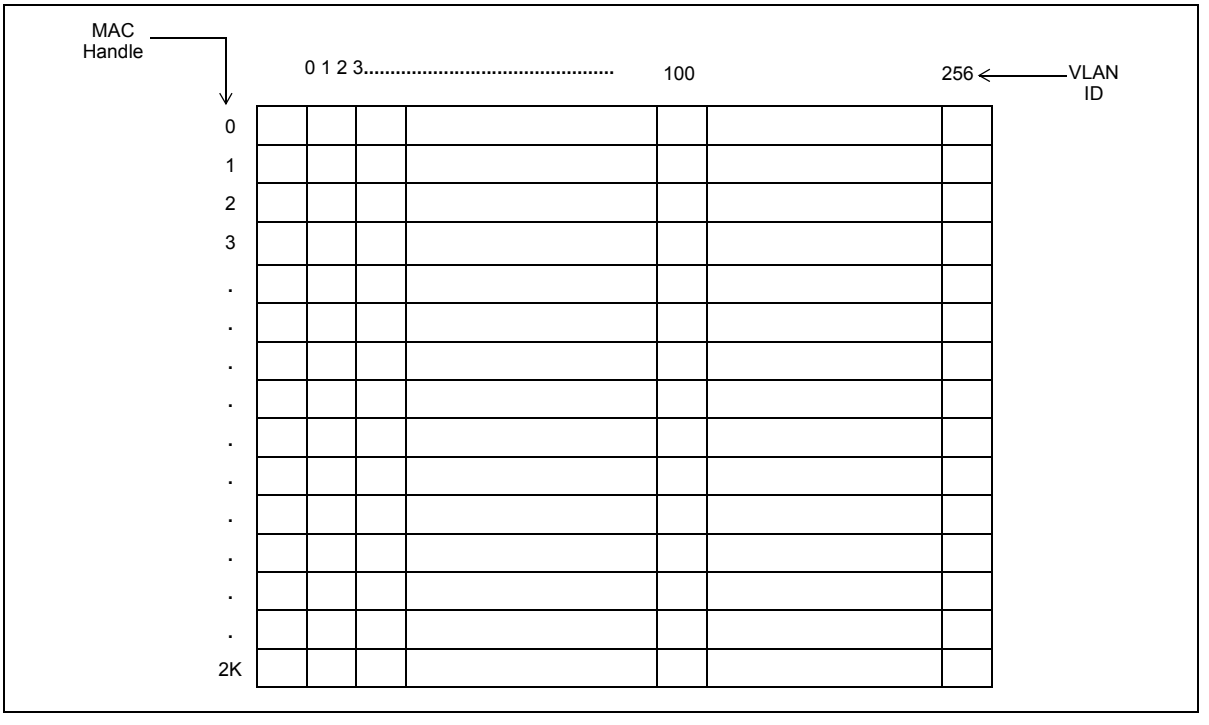
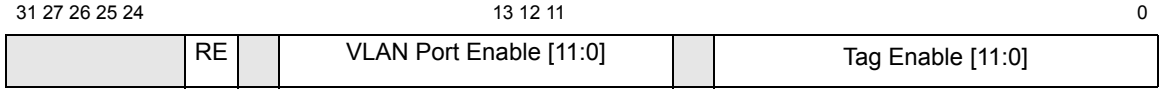


Figure 20 - VLAN MAC Table

14.2.4.3 VLAN Port Mapping Table (VMAP)

The VLAN Port Mapping Table (VMAP) is an internal table within the MDS212. It contains 256 entries, one for each VLAN, identified by an internal VLAN Index. This table can be accessed by CPU software through CPUIRCMD and CPUIRDAT registers. The contents of this table are set up and maintained by CPU software during VLAN creation,

deletion and VLAN port membership modification. VMAP is used by Frame Engine to forward multicast or destination unknown unicast frames to multiple ports simultaneously.



Bit [11:0]VLAN Tag Enable [11:0]One bit for each Ethernet MAC Port

0 = disable, 1 = enable

Bit [12]Reserve

Bit [24:13]VLAN Port Enable [11:0]One bit for each Ethernet MAC Port, identifying the ports

associated with each VLAN. 0 = disable, 1 = enable

Bit [25]Reserve

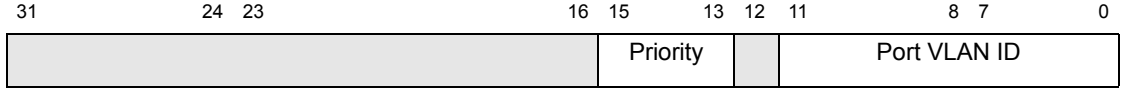
Bit [26]REREmote Ports Enable: Indicate some members in the

remote device. 0=disable, 1=enable

Bit [29:31]Reserve

14.2.5 Port VLAN ID (PVID) Register

This register defines the Port VLAN ID (PVID) and priority for each port. PVID needs to be set up by CPU software, and is used by MDS212 to decide the port's VLAN ID and priority if the incoming packet is VLAN untagged.



Bit [11:0]:Port VLAN ID (PVID),

Bit [12]:Reserved

Bit [15:13]:Priority

Bit [31:16]:Reserved

15.0 IP Multicast

15.1 Introduction

IP Multicast permits an IP host (source) to transmit a single IP packet to multiple IP hosts (receivers). IP Multicasting allows a source to send only one copy and the network ensures delivery to each member of the specified multicast group. Network bandwidth is allocated more efficiently, as multiple copies of the same frame are not transmitted between common ports.

The packet destined to an IP multicast group address determines the set of recipients. Hosts may choose to be members of a number of multicasts, and hence select the multicast packets they wish to receive. They may subscribe or unsubscribe to these multicast groups dynamically, using the Internet Group Management Protocol (IGMP) that support automatic multicast group membership. IGMP is configured to create, update, and/or remove

dynamic multicast group entries between switches and multicast clients and servers. RFC 1112 specifies the protocols and behaviour for IP Multicasting.

The MDS212 supports up to 255 IP Multicast Groups and treats them as extensions of the VLAN operation. No additional hardware is needed, since the IGMP operates on the hardware already provided for VLAN functionality. IGMP packets are identified by the Frame Engine and are passed to the external CPU for processing, when the destination MAC address is 01-00-5E-xx-xx-xx and the Protocol field value equals 2, or when the destination IP address is 224.0.0.x. The external CPU then instructs the HISC to setup IP Multicast entries for the MAC Addresses in the Switch Database Memory, the VLAN Table, and the MCT-VLAN Table. The HISC builds and maintains an MCT-VLAN and a VLAN Table for IP Multicast Groups in the Frame Buffer Memory.

When an IP Multicast packet is received, it is identified by a specific class of Multicast Destination MAC addresses, where the high-order bits indicate use of IGMP, and the low-order bits indicate the specific IGMP Group Identifier. The MDS212 searches the MCT VLAN Association Table for destination MAC addresses, using the IGMP or the IGMP Group Identifier stored in the MCT, to obtain port membership for the IP Multicast Group. The Search Engine forwards the packet to each port associated with the IP Multicast Group. Where no address is found, the HISC firmware updates the MCT-VLAN to include this address.

The Multicast Buffer Control Register (MBCR) allows the configuration of multicast frames to be forwarded, the number of buffers reserved for receiving remote multicast frames, the number of multicast frames allowed, and the multicast forwarding threshold.

15.2 IGMP AND IP Multicast Filtering

IP multicast filtering optimizes switched network performance by limiting multicast packets to only be forwarded to ports containing multicast group membership instead of flooding all ports in a subnet (VLAN).

The Internet Group Management Protocol (IGMP) runs between hosts and their immediate neighbouring multicast routers. The mechanism of the protocol allows a host to inform its local router that it wishes to receive transmissions addressed to a specific multicast group. Routers, also, periodically query the LAN to determine if known group members are still active.

Based on the group membership information, learned from the IGMP, a router is able to determine which (if any) multicast traffic needs to be forwarded to each of its “leaf” subnetwork. Multicast routers use this information, in conjunction with a multicast routing protocol, to support IP multicasting across the Internet.

The MDS212 based switch supports IP Multicast Filtering by passively snooping on the IGMP Query. The IGMP Report packets are transferred between IP Multicast Routers and IP Multicast host groups to learn the IP Multicast group members within each VLAN actively sending out IGMP Query messages soliciting IP Multicast group members. They thus learn the location of multicast routers and member hosts in multicast groups within each VLAN. Since IGMP is not concerned with the delivery of IP multicast packets across subnetworks, an external IP multicast router will be needed if the IP multicast packets have to be routed across different IP subnetworks.

15.3 Implementation In MDS212

The MDS212 supports up to 255 IP Multicast Groups and treats them as an extension of the VLAN operation. No additional hardware is needed, since IP Multicast Switching/Filtering already operates in hardware provided for VLAN functionality.

IGMP packets are identified by the Search Engine and are passed to the external CPU for processing, when the destination MAC address is 01-00-5E-xx-xx-xx and the Protocol field value equals 2, or when the destination IP address is 224.0.0.x. The external CPU then instructs the HISC to setup an MCT entry for this IP Multicast Address in the Switch Database Memory. If this is a new IP Multicast group, it sets up an entry in the VLAN Port Mapping Table by itself,

Whenever an IP Multicast data packet (destination MAC = 01-00-5e-xx-xx-xx, and destination IP address is within the range of 224.0.1.0 and 239.255.255.255) is received, the Search Engine will use the MCT table to look up the

IP Multicast address of the incoming packet. Frame Engine then will use the result from the search (VLAN Index) to forward this IP Multicast packet to its member ports according to the VLAN Port Mapping Table.

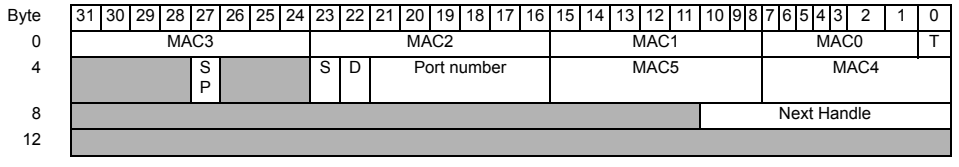
15.3.1 MCT Table

The MCT table is an internal table within the MDS212 chip that has a total of 2K entries. The CPU setups and read the table one entry at a time through microcode in the HISC. There are two types of overlapped MCT entries, one used for layer-2 MAC address based unicast switching, and the other for IP Multicasting.

15.3.1.1 MCT Structure For Unicast Frame

The MCT table is used by the Search Engine to forward unicast frames. By looking up a destination MAC address from this table, the associated port number is found and used for packet forwarding decisions. The content of the table is set up by the Search Engine at the reception of an incoming frame, if the Search Engine is not fully occupied. When the Search Engine is too busy handling frame forwarding decisions, microcode in the HISC engine will be assigned to do the learning job by setting up new MAC to Port associations.

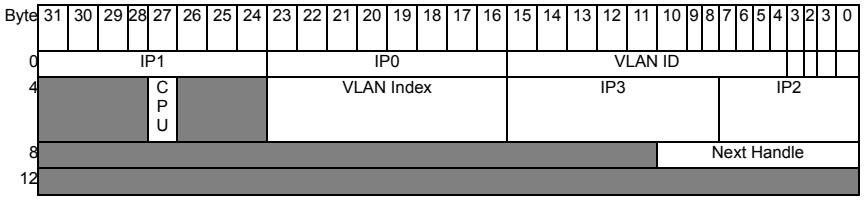
An entry in this table can be setup by microcode in HISC through a CPU software command for static layer-2 packet filtering based on either the source or destination address. An entry can be cleared by microcode in the HISC through a CPU software command, during VLAN deletion, port link down, or when it is aged out. It will also be cleared and a new one set up when a port change of a MAC address is detected.



T: Time stamp, used for aging. Set to 1 after MAC is found, and cleared to 0 when aged.
MAC[5:0]: MAC Address
S: Source MAC address filtering
D: Destination MAC address filtering
SP: Transmit Speed. 0-100Mbps
Next Handle: Pointer to the next entry in a hashed link list.

15.3.2 MCT Structure For IP Multicast Packet

An IP Multicast entry in the MCT table can be setup or torn down by microcode in HISC through a CPU software command for IP Multicasting. Whenever an IP multicast data packet is received, the Search Engine will use this table to look up the IP Multicast address and VLAN ID of the incoming packet. If the IP Multicast address is found, an internal VLAN Index from the MCT entry will be used by the Search Engine and Frame Engine to forward the IP Multicast packet to the specific IP Multicast group members in a VLAN. If not, the packet will be forwarded to the VLAN it belongs to.



VLAN ID: The VLAN ID this IP Multicast group is located in.
IP[3:0]: IP Multicast Address
VLAN Index: Internal VLAN Index used to identify this IP Multicast group
CPU: 1: Switch CPU is part of this IP Multicast group
Next Handle: Pointer to the next entry in a hashed link list.

16.0 Quality Of Service (QOS)

Quality of Service (QoS) provides the capability to reserve bandwidth throughout the network. This is particularly useful for sending voice or video over the switched network. In a switched Ethernet environment, this is only possible with Resource Reservation Protocol (RSVP), a Layer 3 protocol. In a Layer 2 switch, QoS, referred as Class of Service (CoS) by the IEEE 802.1Q standard, provides the capability to prioritize certain tasks on the network. This is done at the application level, where applications can set the priority when the frame is created. The MDS212 classifying Ethernet frames according to their IEEE 802.1p/Q VLAN priorities. There are three bits in the VLAN ID reserved to designate the priority of a packet.

Each port stores its transmission jobs into four transmission scheduling queues, one for each internal priority. Before transmitting, a port selects a queue from which a transmission job is read. The transmission job points to a frame stored in memory that is fetched and transmitted. The four queues, representing four classes of traffic, are selected using a weighted round robin (WRR) strategy. The relative service rates among these queues are programmable such that bandwidth can be allocated according to classes. This ensures that critical applications get a fair share of bandwidth, even when the network is overloaded.

The Search Engine recognizes the IEEE 802.1p priority tag and classifies each incoming frame into four internal priority classes: P0, P1, P2, and P3, in decreasing priority. Since the IEEE 802.1p/Q allows up to eight priorities, a programmable mapping allows the user to map the 802.1p priority to the internal priority tag via register AVTC.

16.1 Weighted Round Robin Transmission Strategy

Frames of four different priorities are transmitted according to a weighed round robin (WRR) strategy. The WRR is a modified form of the fair round-robin strategy, in which the server visits the queues in turn. In a fair round-robin strategy, the server treats all queues equally and visits them with identical frequency. In a WRR, the queues are weighted, i.e., one queue may be visited more frequently than another. These weighs are programmable via register AXSC, in which the service rate ratio between two adjacent classes of traffic is set.

In register AXSC, setting QSW0=2, QSW1=QSW2=1 gives the service ratio 8:2:1:1, which is a good start for most LAN switches. This ratio allocates 67% = 8/12 of bandwidth to P0, 16% = 2/12 of bandwidth to P1, and P2 and P3 each receives 8.3% = 1/12 of bandwidth, assuming all frames have identical frame length.

16.2 Buffer Management Functions

The MDS212 stores frame data in frame buffers. The number of frame buffers in a system is the maximum number of frames a device can store. When all frame buffers are used, incoming frames cannot enter the memory and are discarded. Without buffer management, a congested port causes a backlog of frames that eventually occupy all frame buffers. The MDS212 features buffer management functions that prevent a single type of traffic from depleting all frame buffers. The buffer manager limits the number of frames each destination port can store, thereby preventing congested ports from occupying all the buffers and blocking incoming frames.

The buffer manager examines the destination port of every frame stored, and increments a counter associated with this destination port. These buffer counters keep track of the number of buffers occupied by frames destined to each port. If the counter reaches a threshold, incoming frames destined for the associated port will be dropped. This threshold is programmable via register BCT and BCHL. Register BCT allows the user to program two thresholds, one high and one low. The user specifies a threshold, high or low, for each port in register BCHL.

The buffer manager also prevents multicast frames from occupying all frame buffers. A programmable threshold, register MBCR, limits the number of multicast frames stored in memory. In another word, buffers are reserved for unicast frames.

A multicast forwarding job points to a multicast frame in memory fetched and forwarded by the Frame Engine across the Xpipe to the remote device. The Frame Engine can only forward a handful of multicast frames simultaneously across the Xpipe. Excess multicast forwarding jobs are stored in an internal FIFO, called the MC forwarding FIFO. If the MC forwarding FIFO is full, incoming multicast frames can no longer be forwarded to the remote device. For these blocked multicast frames, their remote destination ports are discarded.

The MDS212 has a programmable option to recognize IP multicast (IPMC) frames. By default, IPMC frames are treated equally with Layer 2 multicast frames. This option gives IPMC privilege, in terms of buffer allocation, over regular Layer 2 multicast frames. In a broadcast storm, Layer 2 multicast frames are discarded before IPMC ones. The system has the flexibility to recognize a programmable IPMC MAC address signature, set by registers IPMCAS0, IPMCAS1, IPMCMSK0, and IPCMMSK1. If a programmable option, DCR2, bit 26, is turned on, the system reserves space in the MC forwarding FIFO for IPMC frames. This ensures that Layer 2 multicast frames do not block IPMC ones

17.0 Port Trunking

Port trunking groups a set of 8 MDS212 10/100Mbps 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 belong to the same VLAN and share the same MAC Address. Each system can support up to 4 groups.

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

17.1 Unicast Packet Forwarding

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.

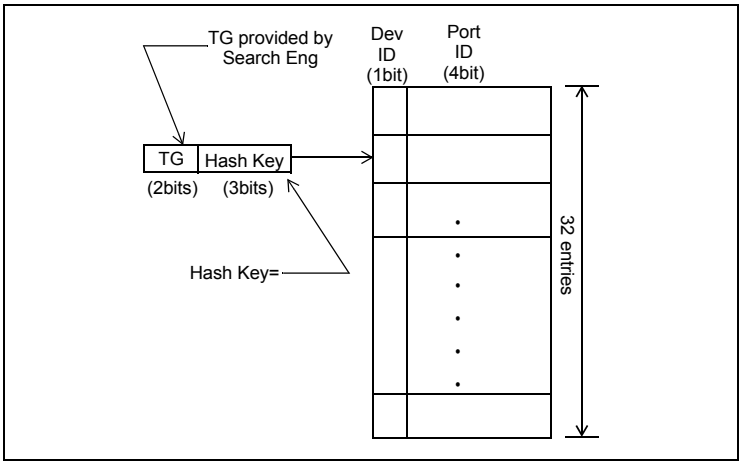


Figure 21 - Port Mapping Table

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 Figure 21.

17.2 Multicast Packet Forwarding

For multicast packet forwarding, the destination device must determine the proper set of ports to transmit the packet based on the VLAN Index and Hash Key, generated by the source Search Engine. Two functions are needed to distribute multicast packets to the appropriate destination ports in a Trunk Group:

1. Selecting a Forwarding Port per Trunk Group

Only one port per Trunk Group will be used to forward multicast packets. This can be done with a VLAN INDEX Table and a Forwarding Port MASK Table set up by CPU.

2. Blocking Multicast Packet Back to the Source Trunk

For multicast forwarding that includes ports in Trunk Groups in the same device as source port, all ports in the same Trunk Group at the receiving port must be excluded. Otherwise, this multicast packet will be looped back to the same source Trunk Group. This is achieved through a Trunk Group ID Register that contains 36 bits (36=12x3).

17.2.1 Select One Forwarding Port Per Trunk Group

To forward multicast frames, the Frame Engine retrieves the VLAN member ports from one of the 256 entries in the VLAN Port Mapping Table (VMAP) as described in the VLAN section. By using the Hash Key and the Forwarding Port Mask table, the Frame Engine can obtain the corresponding FP Mask. The final forwarding ports can then be determined by the logical AND of the FP Mask and the VLAN Member Port bit map. The Forwarding Port- Mask Table must be set by the CPU to THKM[0:7] registers beforehand. The format of this table and the method of setting it up are shown below.

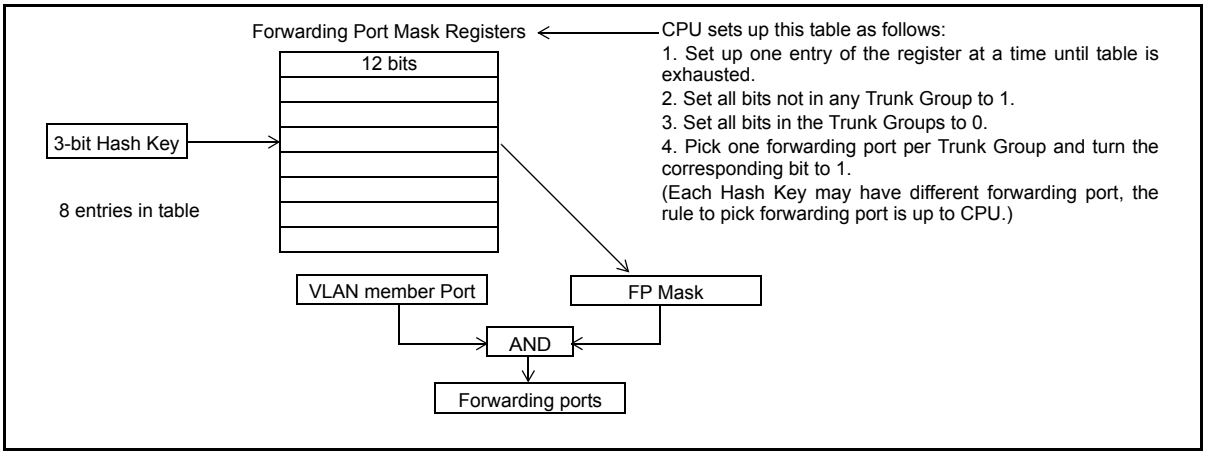


Figure 22 - Forwarding Port Mask Table

Two restrictions exist in setting up tables:

1. When setting up the VLAN Port Mapping Table, all the ports in the Trunk Group must be set to 1, if the VLAN has ports in any Trunk Group.
2. When setting up the Forwarding Port Mask Table, the CPU software picks only one forwarding port per Trunk Group.

17.2.2 Blocking Multicast Packets Back To The Source Trunk

For local multicast packets, the Frame Engine needs to block the multicast packets from being sent to the same Trunk Group as the receiving (source) port. To do it, the Search Engine utilizes the Trunk Group ID (TGID) in ECR1 Register.

The Frame Engine compares the TGID of the source and forwarding ports. If the two TGIDs are the same, the Frame Engine blocks the forwarding port for this multicast packet. The Switch Engine provides the TGID of the source port.

Example:

The following demonstrates the port trunking scheme for multicast packet forwarding.

4 Trunk Group in a switch:

Group 0: port 0,1,2 in device 0
Group 1: port 4, 5,6 in device 0
Group 2: port 1, 2,3 in device 1
Group 3: port 4, 5,6 in device 1

A multicast packet with VLAN INDEX=5 is received at port 0 of device 0.

The membership of this VLAN:

Device 0: port 0, 1, 2, 4, 5, 6, 7
Device 1: port 1, 2, 3, 4, 5, 6, 8

Hash Key = 3

Forwarding Port for each group with Hash Key=3,

Port 2 for Group 0
Port 4 for Group 1
Port 3 for Group 2
Port 6 for Group 3

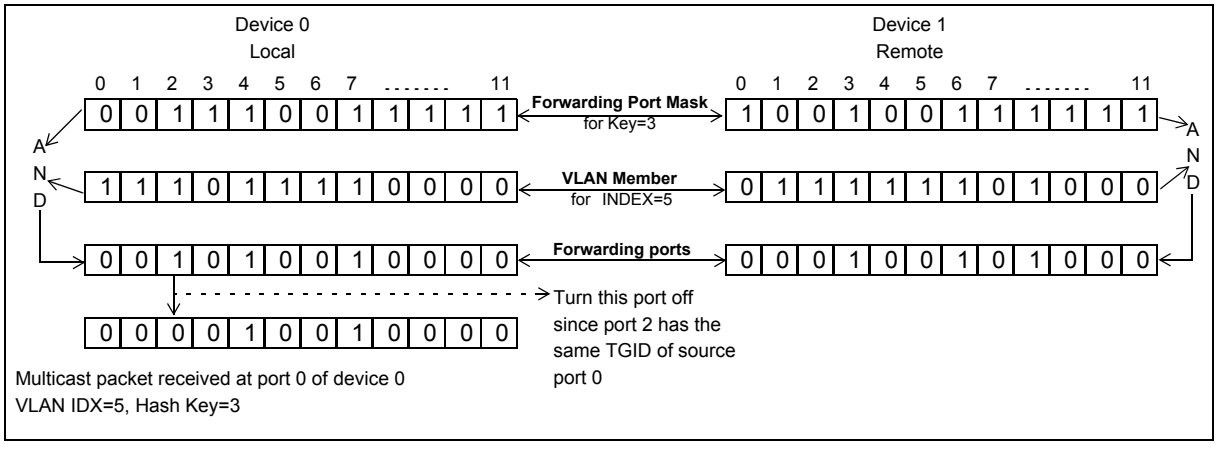


Figure 23 - Multicast Packet Forwarding Example

17.3 MAC Address Assignment

In MDS212, 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 ADAOR0 and ADAOR1 registers, while the 4 LSBs, MAC[3:0] are specified in ADAOR0 and ADAOR1 registers for port 0-port 7 and port 8-port 11, respectively. The 4 LSBs MAC[3:0] can be assigned as follows:

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

18.0 Register Definitions

18.1 Register Map

All registers are grouped into sets:

- Device Configuration
- Buffer Memory Interface
- Frame Control Buffer
- Queue Management
- Switching Control
- Link List Management
- 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

Table 9 - MDS212 Register Map

Tag	Description	Address	W/R
DCR3	Interface Status Register	7CC	--/R
MEMP	Memory packed Register	7DC	W/R
2. Interrupt Controls			
ISR	Interrupt Status Register-Unmasked	7E0	--/R
ISRM	Interrupt Status Register-Masked	7E4	--/R
IMSK	Interrupt Mask Register	7E8	W/R
IAR	Interrupt Acknowledgment Register	7EC	W/--
3. Buffer Memory Interface			
MWARS	Memory Write Addr. Reg. - Single Cycle	780	W/--
MRARS	Memory Read Addr. Reg. - Single Cycle	784	W/--
MWARB	Memory Burst Write Address Register	788	W/--
MRARB	Memory Burst Read Address Register	78C	W/--
MWDR	Memory Write Data Register	790	W/--
MRDR	Memory Read Data Register	794	--/R
VTBP	VLAN ID & MAC member Table Base Pointer	798	W/R
MBCR	Multicast Buffer Control Register	79C	W/R
RAMA	RAM block access Register	7A0	W/R
Reserve	Must Set to “0x0001 0008”	7B8	W/R
Reserve	Must Set to “0x0001 0000”	7BC	W/R
4. 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
5. Queue Management			
CINQ	CPU Input Queue	708	W/--
COTQ	CPU Output Queue	70C	--/R
6. Switching Control			
HPCR	HISC Processor Control Register	6C0	W/R
HMCL0	HISC Micro-Code Loading Port-Low	6C4	W/R
HMCL1	HISC Micro-Code Loading Port-High	6C8	W/R
HPRC	HISC Priority Control Register	6D0	W/R
MCS0R	Micro Sequence 0 Register	6D4	W/R
MCS1R	Micro Sequence 1 Register	6D8	W/R

Table 9 - MDS212 Register Map (continued)

Tag	Description	Address	W/R
FCR	Flooding Control Register	6DC	W/R
MCAT	MCT Aging Timer	6E0	W/R
TPMXR	Trunk Port Mapping Table Index Register	6E4	W/--
TPMTD	Trunk Port Mapping Table Data Register	6E8	W/R
PTR	Pacing Time Regulation	6EC	W/R
MTCR	MCT Threshold & Counter Register	6F0	W/R
7. Link List Management			
LKS	Link List Status Register	680	W/R
AMBX	Mail Box Access Port	684	W/R
AFML	Free Mail Box List Access Port	688	W/R
8. Access Control Function Group 1 (Chip Level controls)			
AVTC	VLAN Type Code	648	W/R
AXSC	Transmission Scheduling Control Register	64C	W/R
ATTL	Transmission Timing & Threshold Control Register	650	W/R
AMIIC	MII Command Register	654	W/--
AMIIS	MII Status Register	658	--/R
AFCRIA	Flow Control Ram Input Address	65C	W/--
AFCRID0	Flow Control Ram Input Data	660	W/R
AFCRID1	Flow Control Ram Input Data	664	W/R
AFCR	Flow Control Register	670	W/R
AMAR0	Multicast Addr. For MAC Control Frames Byte [3,2,1,0]	674	W/R
AMAR1	Multicast Addr. For MAC Control Frames Byte [5,4]	678	W/R
AMCT	MAC Control Frame Type Code Register	67C	W/R
ADAR0	Base MAC Address Register - Byte [3,2,1,0]	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 10Mbps port	614	W/R
AFCOFT100	Flow Control Off Time for 100Mbps port	618	W/R
AFCHT10	Flow Control Holding Time for 10Mbps port	620	W/R
AFCHT100	Flow Control Holding Time for 100Mbps port	624	W/R
9. Access Control Function Group 2 (Chip Level controls)			
APMR	Port Mirroring Register	5C0	W/R

Table 9 - MDS212 Register Map (continued)

Tag	Description	Address	W/R
PFR	Protocol Filtering Register	5C4	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
IPMCAS0	IP multicast MAC address signature Low Register - Byte [3:0]	5E8	W/R
IPMCAS1	IP multicast MAC address signature High Register - Byte [5:4]	5EC	W/R
IPMCMSK0	IP multicast MAC address Mask Low Register - Byte [3:0]	5F0	W/R
IPMCMSK1	IP multicast MAC address Mask High Register - Byte [5:4]	5F4	W/R
CFCBHDL	FCB Handle Register for CPU	580	--/R
CPUIRCMD	CPU Internal RAM Command Register	584	W/R
CPUIRDAT0	CPU Internal RAM Data Register - 0	588	W/R
CPUIRDAT1	CPU Internal RAM Data Register - 1	58C	W/R
CPUIRDAT2	CPU Internal RAM Data Register - 2	590	W/R
CPUIRRDY	Internal RAM Read Ready for CPU	594	--/R
LEDR	LED Register	598	W/R
10. Ethernet MAC Port Control Registers - (substitute [N] with Port Number, N={0.11}			
ECR0	MAC Port Control Register	[N*4]0	W/R
ECR1	MAC Port Configuration Register	[N*4]4	W/R
ECR2	MAC Port Interrupt Mask Register	[N*4]8	W/R
ECR3	MAC Port Interrupt Status Register	[N*4]C	--/R
ECR4	Status Counter Wrap Signal	[N*4+1]0	W/R
PVIDR	PVID Register	[N*4+2]4	W/R

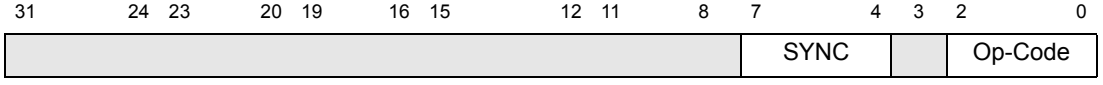
Table 9 - MDS212 Register Map (continued)

18.2 Register Definitions

18.2.1 Device Configuration Register

18.2.1.1 GCR - Global Control Register

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



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

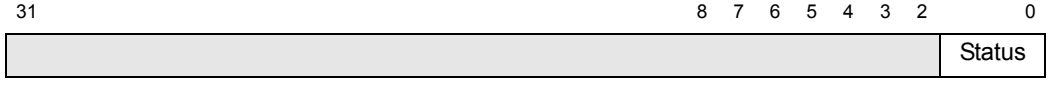
Op-Code	Command	Description
000	Clr RST	**Clear Device Reset:** Allows state machines to exit from RESET state and to initialize their internal control parameters if necessary.
001	RESET	**Device Reset:** Resets all internal state machines of each device and stays in RESET state (except the Processor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

Table 10 - Global Control Register

Bit [7:4] SYN bits, reserved for HISC Usage.

18.2.1.2 DCR0 - Device Status Register

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



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

- Power-up default = 00

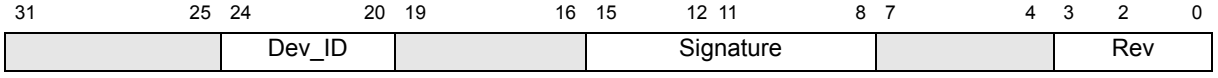
Status	State	Description
00	INIT	**Initialization:** Device is in idle state pending for system software initialization.
01	RESET	**Device Reset:** Device is in RESET state.
10	EXEC	**Execution:** Device is under normal operation.

Table 11 - Device Status Register

Bit [31:2] Reserved

18.2.1.3 DCR1 - Signature, Revision & ID Register

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



Bit [3:0] Device Revision Code

Bit [6:4] Reserved

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

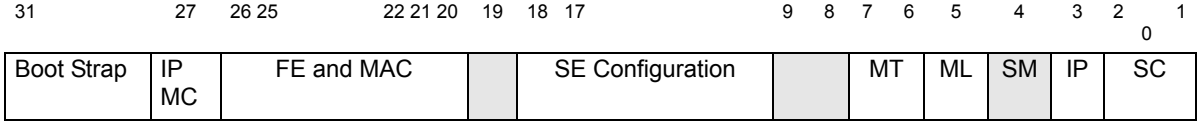
Bit [19:16] Reserved

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

Bit [31:25] Reserved

18.2.1.4 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 10=90Mhz

01 = 120Mhz 11=80Mhz

Bit [2] IP IRQ output polarity control Power-up default =0

0 = active low output 1 = active high output

Bit [3] SM System Configuration mode

0=Nonblocking (For MDS212, 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 Default = 0

1 = 4 memory chips

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
Bit [11:10]	HM	Hashing Mode, each of which uses different bits of MAC address to come up with each bit of hashing key. Default = 00 00=mode 0 01=mode 1 10=mode 2 11=mode 3
Bit [13:12]	HS	Hashing Size Default = 01 00= 8 01= 9 10= 10 11= TDB
Bit [14]	VSW	LAN Aware Switch Default = 0 0 = VLAN Unaware 1 = VLAN Aware
Bit [15]	NoIPM	No IP Multicast Default = 1 1 = IP Multicast Disable 0 =IP Multicast Enable
Bit [16]	GLN	Global Learning Disable, where CPU shall disable global learning before look into it as a whole piece. Default = 0 1 = Learning Disable 0 = Learning Enable
Bit [17]	Partial Syn enable	Partial Synchronization enable for MAC Table Default=0 0= Fully Synchronization for MCT table 1= Partial Synchronization for MCT table
Bit [18]	Reserved	

Frame Engine and MAC Configuration

Bit [19]	FE_AGEN	Aging enable. If true, the memory resources, occupied by the old message, will free up. Default = 1 0 = disable aging 1 = enable aging
Bit [20]	FOF	Forward Oversize FramesPower-up default =0 0 = Discard oversize frames 1 = Forward oversize frames
Bit [21]	Dec_Buffer_C NT	Decrements buffer counter. When the software writes “1” to this bit, the Frame Engine decreases buffer counter by one.
Bit [22]	BC_EN	Buffer counter enable 0 = Disable (no head of line control) 1 = enable
Bit [23]	STA_EN	Status counter enable 0 = collect status in counter disable1 = collect status in counter enable
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 [26]	IPMCIP	Multicast privileges enable: IP multicast traffic has a privilege over regular multicast traffic. Default=0 0= disable 1= enable

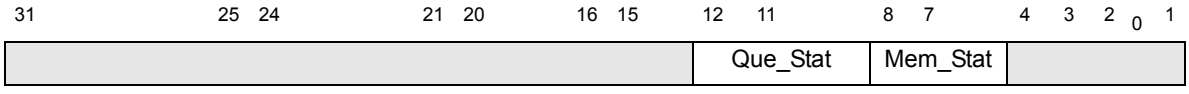
Boot Strap Determine by the bootstrap value.

Bit [27]	BMOD	Control Bus Mode **(Read only bit) Must be 0**
Bit [28]	RW	CPU Read/Write Control Polarity Selection **(Read only bit)** 0 = R/W# 1 = W/R#
Bit [29]	SWM	Switching Mode **(Read only bit)** Default=1 0 = Managed mode 1 = Unmanaged mode

Bit [30]	PSD	Master Device Enable **(Read only bit)** Default=1 1 =Primary 0 = Secondary
Bit [31]	MRDY	Option of merge the RDY and B_RDY as one pin **(Read only bit)** Default=1 0 = merged pin 1 = separated pins

18.2.1.5 DCR3 - Interfaces Status Register

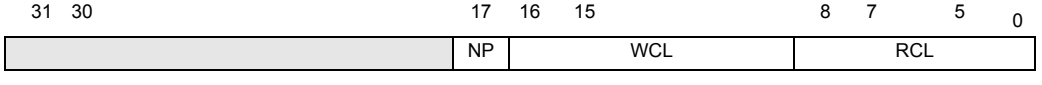
- Access: Zero-Wait-State, Direct Access, Read only
- Address: h7CC



Bit [3:0]	Reserved	
Bit [7:4]	Mem_Stat	Buffer Memory Interface Status
Bit [4]	BB	Buffer Memory Busy, CPU interface is busy accessing Memory
Bit [5]	RE	Read FIFO Empty, the FIFO that CPU interface reads is empty
Bit [6]	WE	Write FIFO Empty, the FIFO that CPU writes is empty
Bit [7]	Res.	Reserved
Bit [11:8]	Que_Stat	Queue Manager Interface Status
Bit [8]	IQ_Rdy	CPU Input Queue is ready for CPU to write into queue
Bit [9]	IQ_Full	CPU Input Queue is full
Bit [31:12]	Reserved	

18.2.1.6 MEMP - Memory Packed Register

- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h7DC

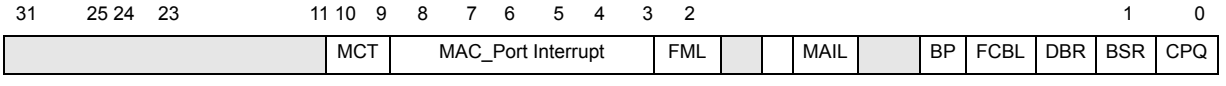


Bit [7:0]	RCL	Read Cycle Limit (Unit is system Clock). Threshold of reads cycle time.	Default=16
Bit [15:8]	WCL	Write Cycle Limit (Unit is system Clock). Threshold of writes cycle time.	Default=16
Bit [16]	NP	Not Packed NP=0 Enable the feature of memory read/write packed.	Default=0 NP=1 Disable, memory access will be a pure round-robin scheme.
Bit [31:17]	Reserved		

18.2.2 Interrupt Control Registers

- Four 32-bit Control Registers.
- **ISR** Interrupt Status Register Identify the unmasked interrupt request sources
- Access: Zero-Wait-State, Direct Access, Read only
- Address: h7E0
  - **ISRM** Masked Interrupt Status Reg. Identify the sources of interrupt with masking
  - Access: Zero-Wait-State, Direct Access,Read only
  - Address: h7E4
  - **IMSK** Interrupt Mask Register Defines the interrupt sources to be masked
  - Access: Non-Zero-Wait-State, Direct Access,Write/Read
  - Set bits to 1 to mask the corresponding interrupt sources
  - Address: h7E8
  - **IAR** Interrupt Acknowledgment Reg. Clear the interrupt request bits
  - Access: Non-Zero-Wait-State, Direct Access,Write only
  - Set bits to 1 to clear the corresponding interrupt sources
  - Address: h7EC

All 4 registers have a common register format and bit assignment



Interrupt MAC port mapping bit/port Interrupt Source

Interrupt Sources (The following bits need to be redefined.)

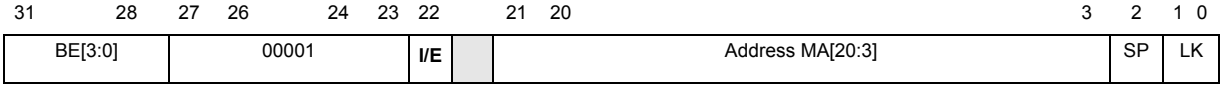
Bit [0]	CPU_Q_Out	CPU output queue level interrupt
Bit [1]	BSR	Bad switch response
Bit [2]	Double R	Double Release
Bit [3]	FCB_Low	FCB Low
Bit [4]	HISC_BP	HISC instruction pointer matched with Breakpoint Register
Bit [5]	Reserved	
Bit [6]	Reserved	
Bit [7]	MAIL_ARR	Mail arrived from HISC
Bit [8]	HISC_TO	HISC Timeout Interrupt
Bit [9]	Reserved	
Bit [10]	FML_Av	Link manager informs CPU that at least 16 Free Mail entry available after CPU encounters empty Free Mail list situation.
Bit [23:11]	MAC_PORT	Interrupt from MAC ports Bit [11] for Port 0, Bit [12] for Port 1 … Bit [23] for port 12.
Bit [24]	MCT	Search Engine found looped MCT Chain.
Bit [31:25]	Reserved	

Note: MAIL_ARR, CPU_Q_Out, and interrupts cannot be cleared by the CPU. They will be cleared whenever their queues are emptied.

18.2.3 Buffer Memory Interface Register

18.2.3.1 MWARS - Memory Write Address Register - Single Cycle

- Access: Zero-Wait-State, via FIFO, Write
- Address: h780



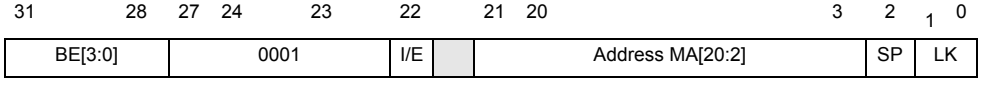
Bit [0]	LK	Lock Flag (for internal memory only)
		LK=0 UnlockLK=1 Lock
Bit [1]	SP	Swap Byte Order
Bit [20:2]	MA [20:2]	Buffer memory address Bit [20:2] – (Bit [1:0] = 00)
Bit [21]	Reserved	
Bit [22]	I/E	Indicates the Address is Internal or External memory
		I/E=0 Internal memory
		I/E=1 External memory
Bit [27:23]	Count	Must be 00001
Bit [31:28]	BE [3:0]	Byte lane enables

CPU Bus Type	Bit [31]	Bit [30]	Bit [29]	Bit [28]
Little Endian	BE [3]	BE [2]	BE [1]	BE [0]
Big Endian	BE [0]	BE [1]	BE [2]	BE [3]

18.2.3.2 MRARS - Memory Read Address Register - Single Cycle

Access: Zero-Wait-State,via FIFO,Write

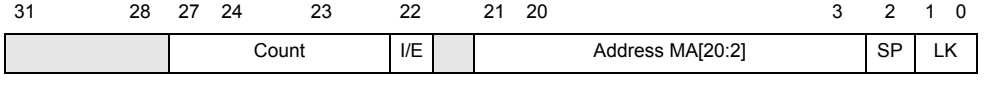
Address:h784



Bit [0]	LK	Lock Flag memory LK=0 Unlock LK=1 Lock
Bit [1]	SP	Swap Byte Order
Bit [20:2]	MA [20:2]	Buffer memory address Bit [20:2] – (Bit [1:0] = 00)
Bit [21]	Reserved	
Bit [22]	I/E	Indicate the Address is Internal or External memory I/E=0 Internal memory I/E=1 External memory
Bit [27:23]	Count	Must be 00001
Bit [31:28]	BE [3:0]	Byte lane enables.

18.2.3.3 Address Registers For Burst Cycle

- Two 32-bit burst size registers share a common format
  - **MWARB** Memory Address Register – Burst Write (in D-words) – Maximum 8 D-words
    - Address: h788
  - **MRARB** Memory Address Register – Burst Read (in D-words) – Maximum 8 D-words
    - Address: h78C
  - Access: Zero-Wait-State, via FIFO, Write



Bit [0]	LK	Lock Flag LK=0 Unlock LK=1 Lock
Bit [1]	SP	Swap Byte Order
Bit [2]	Reserved	
Bit [20:2]	MA [20:2]	Buffer memory address Bit [20:2] – (Bit [1:0] = 00)
Bit [21]	Reserved	
Bit [22]	I/E	Indicate the Address is Internal or External memory I/E=0 Internal memory I/E=1 External memory

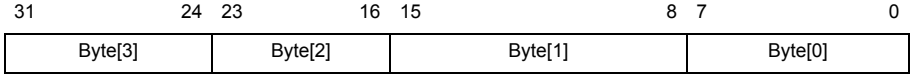
Bit [27:23]	Count	Count = Burst Size in double words Burst size for internal memory is up to 8 D-words. Burst size for external memory is up to 16 D-words 00001 = 1 D-word, ……01000 = 8 D-word 01111 = 15 D-word 10000 = 16 D-word Valid value range for internal memory is {1 to 8} Valid value range for external memory is {1 to 16}

Caution: When setting Count = 16, the Starting address has to be in the Q-word boundary. That is MA[2]=0.

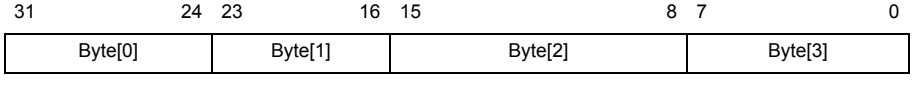
Bit [31:28]Reserved

18.2.3.4 Memory Read/Write Data Registers

- Four 32-bit data registers share a common format
  - MWDR Memory Write Data Register
  - Access: Zero-Wait-State, via FIFO, Write only
  - Address: h790
  - MRDR Memory Read Data Register
  - Access: Zero-Wait-State, Direct Access, Read only
  - Address: h794
- Byte Order depends on CPU types
  - Little Endian CPUs

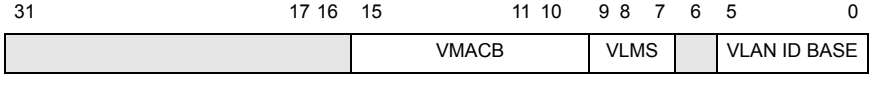


  - Big Endian CPUs



18.2.3.5 VTB - VLAN ID Table Base Pointer

- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h798

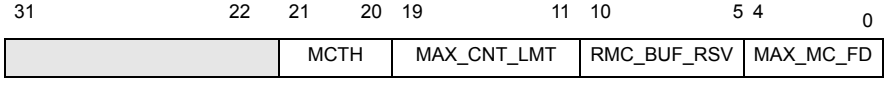


Bit [5:0]	VIDB	VLAN ID Table Base, serves as [20:15] bits of address. (VLAN ID Table is 32KB)
Bit [6]	Reserved	
Bit [8:7]	VLMS	The size of VLAN MAC Table Default=11 00= reserved 10=32K (for 128 VLANs) 01=16K (for 64 VLANs) 11=64K (for 256 VLANs)

Bit [15:9]	VMACB	VLAN Mac Table Base, serves as [20:14] bit of address. This table indicates the association of MAC address and VLAN.
Bit [31:16]	Reserved	

18.2.3.6 MBCR - Multicast Buffer Control Register

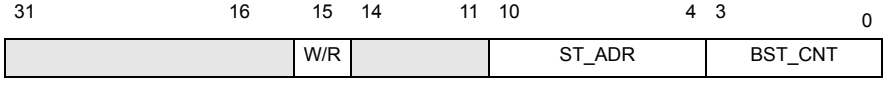
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h79C



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 [21:20]	MCFTH	Multicast Forwarding Threshold: Watermark for forwarding FF to drop regular multicast packet if IPMC bit in DCR2[26] is ON. CPU can set four level watermarks, which are programmable. 00=25% 10=75% 01=50% 11=100%
Bit [31:22]	Reserved	

18.2.3.7 RAMA - RAM Counter Block Access Register

- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h7A0
- RAM counter block contains 12 counter blocks (one for each port) Port 0 counter block starts at address 0.) The size of each block is 16 double words, which holds, in total, 30 statistic counters. The size and type of each counter is referred to the register ECR4.
- CPU uses this register to access the specified statistic counter by setting the start address of RAM counter block and the length.

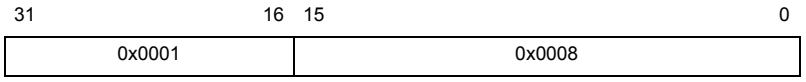


Bit [3:0]	BST_CNT	Read/Write burst (length) of RAM Block. (Unit = 1double words)
Bit [10:4]	ST_ADR	Read/Write Start Address.
Bit [14:11]	Reserved	
Bit [15]	W/R	RAM Block Access Write/Read indicator 1 = Write 0 = Read
Bit [16:31]	Reserved	

Note: The access range is equal to from ST_ADR to END_ADR= S_ADR+ BST_CNT.The END_ADR cannot cross the boundary of each port block, i.e., 8 double words.

18.2.3.8 Reserve Register 1

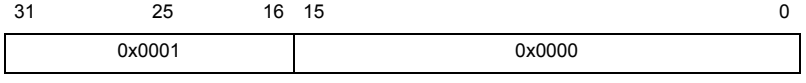
- Access Non-Zero-Wait-State Direct-Access Write/Read
- Address: h7B8



Must be set to "0X00010008"

18.2.3.9 Reserve Register 2

- Access Non-Zero-Wait-State Direct-Access Write/Read
- Address: h7BC

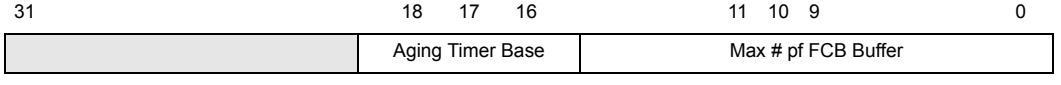


Must be set to "0X00010000"

18.2.4 Frame Control Buffers Management Register

18.2.4.1 FCBSL - FCB Queue

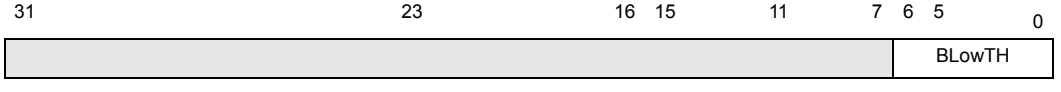
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h740



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

18.2.4.2 FCBST - FCB QUEUE - Buffer Low Threshold

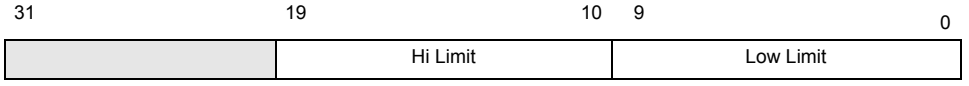
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h744



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 CPU.
Bit [31:6]	Reserved	

18.2.4.3 BCT - (FCB) Buffer Counter Threshold

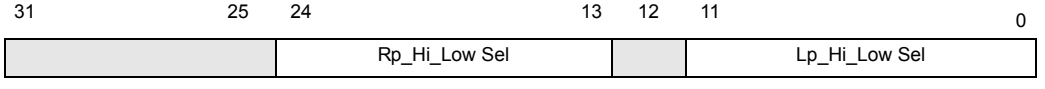
- Access Non-Zero-Wait-State Direct-Access Write/Read
- Address: h74C



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

18.2.4.4 BCHL - Buffer Counter Hi-low Selection

- Access Non-Zero-Wait-State Direct-Access Write/Read
- Address: h750

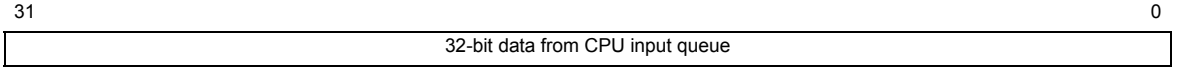


Bit[11:0]	Lp_Hi_Low Sel	Selection for Low or High Limit of Buffer Counter for Local device 12 bits maps to 12 ports in Local Device 1 = select hi limit 0 = select low limit
Bit[12]	Reserved	
Bit[24:13]	Rp_Hi_Low Sel	Selection for Low or High Limit of Buffer Counter for Remote device 13 bits maps to 13 ports in Remote Device 1 = select hi limit 0 = select low limit
Bit[31:25]	Reserved	

18.2.5 Queue Management Register

18.2.5.1 CINQ - CPU Input Queue

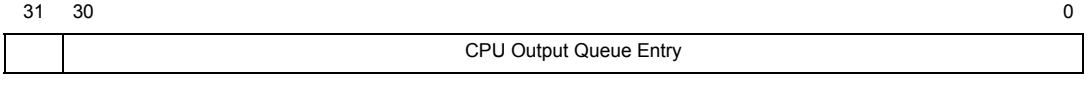
- Access: Non-Zero-Wait-State, Direct Access, Write only
- Address: h708



Note: Check IQ_RDY=1 in DCR3 (Interface Status Register) before writing into CPU Input Queue.

18.2.5.2 COTQ - CPU Output Queue

- Access: Non-Zero-Wait-State, Direct Access, Read only
- Address: h70C

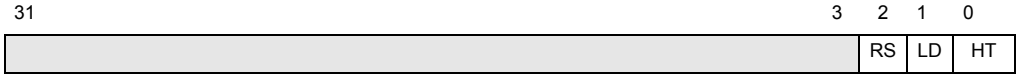


Bit [30:0]	31-bit CPU Output Queue Entry
Bit [31]	Status queue is ready

18.2.6 Switching Control Register

18.2.6.1 HPCR - HISC Processor Control Register

- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h6C0



Bit [0] HT — Halt the HISC processor from execution Not Apply for non-managed mode (It can be fixed in next cut.)
Power-up default = 1

Bit [1] LD — Switch the Micro-Code Memory from instruction fetch mode to down-loading mode

Bit [2] RS — Reset IP to 0 – **(Write only bit)** (This bit is auto reset to 0 after IP is reset to 0)

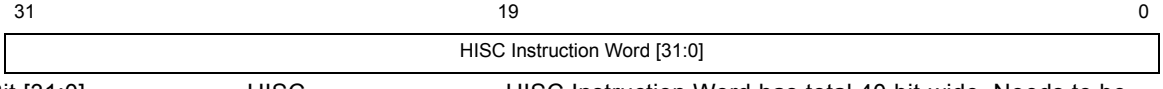
Bit [31:3] Reserved

RS	LD	HT	State	Description
1	0	1	**INIT**	**Initialization State:** Stopped HISC execution, reset IP to 0.
0	0	1	**HALT**	**Halt State:** Stopped HISC execution, waiting for HT=0
0	1	X	**LOAD**	**Micro-Code Loading State:** Stopped HISC execution, increment IP for every Wr/RD to HMPC.
1	0	0	**START**	**Start State:** Reset IP=0, and start HISC execution.
0	0	0	**EXEC**	**Execution State:** Continue HISC execution without reset IP.
1	1	X	**--**	**Illegal State**

Table 12 - HPCR - HISC Processor Control Register

18.2.6.2 HMCL0 - HISC Micro-code Loading Port – Low

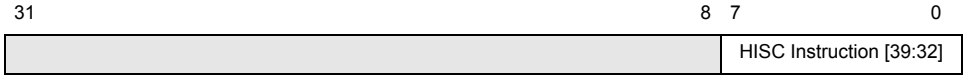
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h6C4
- Loading micro code into HISC.



Bit [31:0] HISC Instruction Word [31:0] — HISC Instruction Word has total 40 bit-wide. Needs to be broken into two registers.

18.2.6.3 HMCL1 - HISC Micro-code Loading Port – High

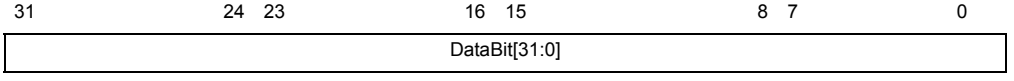
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h6C8



Bit [7:0] HISC Instruction Word [39:32]
Bit [31:8] Reserved

18.2.6.4 MS0R Micro Sequence 0 Register

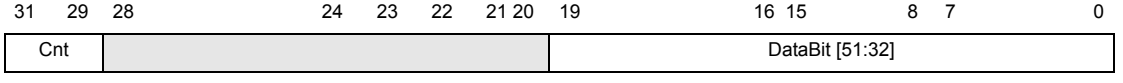
- Access: Zero-Wait-State, Direct Access, Write/Read
- Address: h6D4



Bit[31:0] DataBit [31:0] Data Bit [31:0] to the sequencer RAM (The length of Micro Sequence Data is 54-bit, Need to be broken into tow registers)

18.2.6.5 MS1R Micro Sequence 1 Register

- Access: Zero-Wait-State, Direct Access, Write/Read
- Address: h6D8



Bit [19:0] Data Bit [51:32] to the sequencer RAM

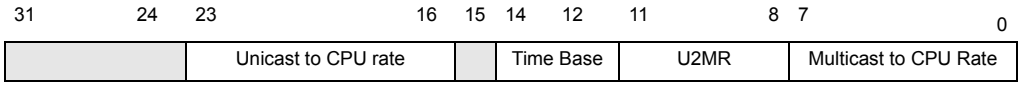
Bit [31:29] CNT Control bits (Write only bits)

000	NOP
001	Load
010	Restart Ptr
011	IncAdr
100	Halt
101	UnLoad
110	UnHalt

Bit [28:20] Reserved

18.2.6.6 Flooding Control Register

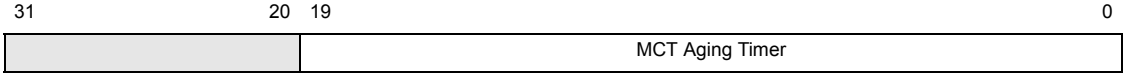
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h6DC



Bit [7:0]	M2CR	Multicast to CPU Rate Restricts the number of frames within the Time window defined in bit[15:12]
Bit [11:8]	U2MR	Unicast to Multicast Rate Restricts the number of flooding unicast frames within the Time window
Bit [14:12]	Time Base	Defines the time window used by M2CR and U2MR 000 = 100us 001 = 200us 010 = 400us 011 = 800us 100 = 1.6ms 101 = 3.2ms 110 = 6.4ms 111 = 100us
Bit [23:15]	U2CR	Unicast to CPU Rate Restricts the number of frames within the Time window defined in bit[15:12]
Bit [31:24]	Reserved	

18.2.6.7 MCAT - MCT Aging Timer

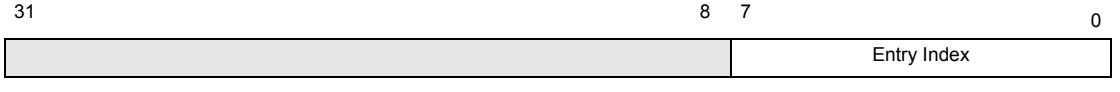
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h6E0



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.

18.2.6.8 TPMXR - Trunk Port Mapping Table Index Register

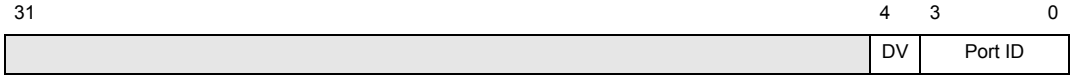
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h6E4
- For Trunk port mapping Table pointer



Bit [7:0]		8-bit Table entry Index
Bit [8:31]	Reserved	Value set to 0

18.2.6.9 TPMTD - Trunking Port Mapping Table Data Register

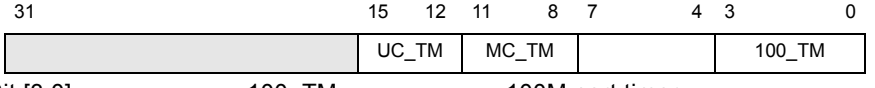
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h6E8



Bit [0:3]	Port ID	Trunking port
Bit [4]	DV	Device ID

18.2.6.10 PTR - Pacing Time Regulation

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

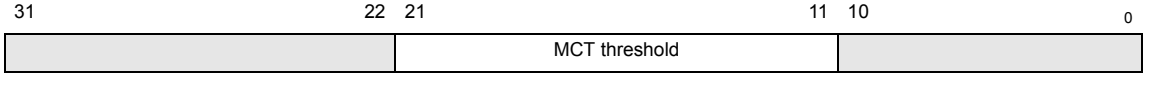


Bit [3:0]	100_TM	100M port timer	Default =5
Bit [7:4]	RSVD		
Bit [11:8]	mc_TM	Multicast timer	Default =5
bit[15:12]	uc_TM	Unicast timer	Default =5
		Unit time = 80 nsec.(for 64Bytes Frame.)	

Note that the Frame Engine determines 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.

18.2.6.11 MTCR - MCT Threshold & Counter Register

- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h6F0

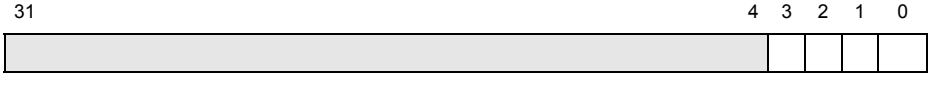


Bit [10:0]	Reserved	
Bit [21:11]	MCT Threshold	Alert system when free MCT entries are below this threshold

18.2.7 Link List Management

18.2.7.1 LKS - Link List Status Register

- Access: Zero-Wait-State, Direct Access, Read only
- Address: h680

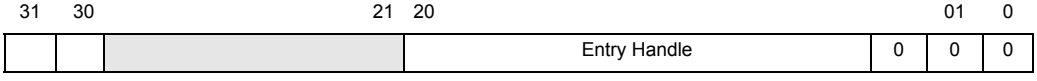


Bit [0]	Mail Box is not ready for CPU to send entry to HISC	
	1=Not Ready	0=Ready

Bit [1] Free Mail Box is not ready for CPU to put entry into
1= Not Ready 0=Ready
Bit [2] CPU gets Mail from HISC
1= Ready 0=Not Ready
Bit [3] Free Mail Box has entry for CPU to get
1=Ready 0=Not Ready
Bit [31:4] Reserved

18.2.7.2 AMBX - Mail Box Access Port

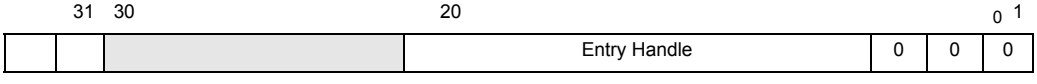
- Access: Zero-Wait-State, Direct Access, Write/Read
- Address: h684
- In write mode, CPU sends Mail to HISC
- In Read mode, CPU receives Mail from HISC



Bit [20:0] Entry handle, the bit [2:0] always 2'b000
Bit [29:21] Reserved
Bit [30] Link List is Empty. (**Read only**)
Bit [31] Link List is Ready. (Same as bit [0] of LKS register) (**Read only**)

18.2.7.3 AFML - Free Mail Box List Access Port

- Access: Zero-Wait-State, Direct Access, Write/Read
- Address: h688

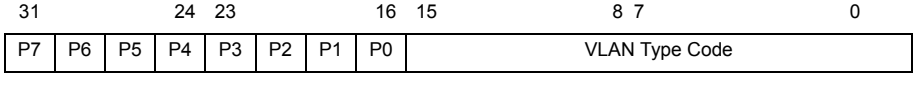


Bit [20:0] Entry handle, the bit [2:0] always 2'b000
Bit [29:21] Reserved
Bit [30] Link List is Empty. (**Read only**)
Bit [31] Link List is Ready.
(Same as bit [1] of LKS register) (**Read only**)

18.2.8 Access Control Function

18.2.8.1 AVTC - VLAN Type Code Register

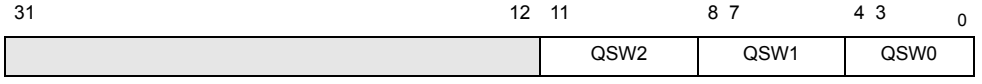
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h648



Bit [0:15] 2-byte VLAN Type Code defined by IEEE 802.1Q VLAN Standard
Bit [31:16] Priority 4 level priority denoting by 2-bit for each
Mapping 8 level VLAN priorities to 4 level internal priorities.

18.2.8.2 AXSC - Transmission Scheduling Control Register

- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h64C



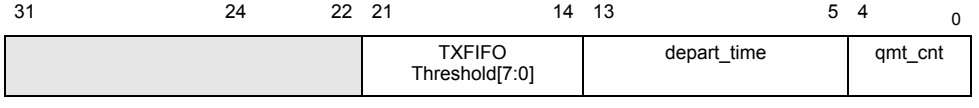
Bit [11:0] QSW[2:0] Transmission Queue Service Weight for queue 2, 1, & 0.
(4 bit each) Defines the service rate for each queue QR0-QR3

QR0 : QR1 : QR2 : QR3
QR0 = QSW0(QSW1+1)* (QSW2+1)*
QR1 = QSW1(QSW2+1)*
QR2 = QSW2
QR3 = 1

Note: Queue 0 has the highest priority. Queue Size is defined in the Queue Control Table.

18.2.8.3 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)
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.

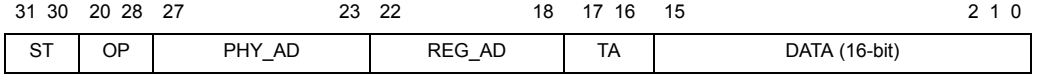
18.2.9 MII Serial Management Channel

These registers are part of the Management Module. They allow the upper layer services to communicate with any one of the PHYs that are connected to the Management Module through the serial interface.

18.2.9.1 AMIIC - MII Command Register

This is a write-only register. The upper layer services write the management frame to be sent to the PHYs into this register. The MSB (bit 31) is the first bit sent over the serial interface.

- Access: Non-Zero-Wait-State, Direct Access, Write only
- Address: h654

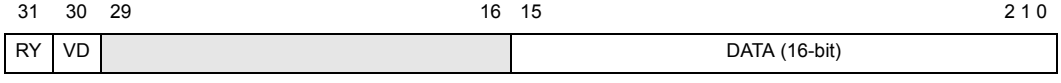


Bit [31:30]	ST	Start of frame – always = “01”
Bit [29:28]	OP	Operation code – “10” for read command and “01” for write command
Bit [27:23]	PHY_AD	5-bit PHY Address
Bit [22:18]	REG_AD	5-bit Register Address in PHY
Bit [17:16]	TA	Turnaround – “10” for write
Bit [15:0]	DATA	16-bit Write Data to PHY

18.2.9.2 AMIIS - MII Status Register

The upper layer services should read this register for data sent by the PHYs. The lower 16 bits contain data received by the Management Module

- Access: Non-Zero-Wait-State, Direct Access, Read only
- Address: h658



Bit [31]	RDY	Data Ready
Bit [30]	VALID	Data Valid
Bit [15:0]	DATA	16-bit Read Data from PHY

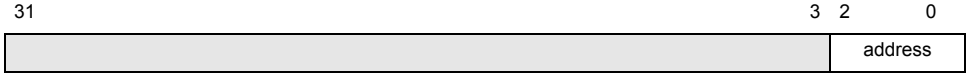
Bit [31] RDY	Bit [30] VALID	Description
1	1	Data field contains valid data from the PHYs.
1	0	Data field contains invalid data from the PHYs.
0	X	Data field is not ready to be read by Switch Manager CPU.

Table 13 - AMIIS - MII Status Register

18.2.10 Flow Control Management

18.2.10.1 AFCRIA - Flow Control RAM Input Address

Access: Non-Zero-Wait-State, Direct Access, Write only
Address: h65C

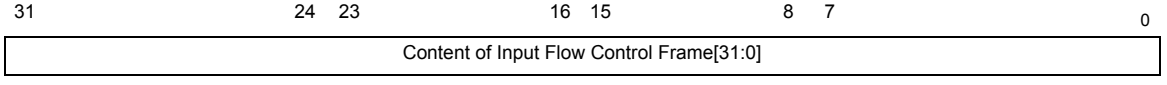


Bit [2:0] 3-bit address for the RAM in MAC storing flow control frame

Usage: Flow Control Frame consists of 64 Bytes. Using AFCRIA and AFCRID0-1, the CPU loads 8 bytes each time. The CPU specifies the address in AFCRIA and writes the content of 4 bytes in AFCRID0 and 4 Bytes in AFCRID1. Then repeats the above procedure 8 times to load a whole flow control frame into the Chip.

18.2.10.2 AFCRID0 - Flow Control RAM Input Data 0

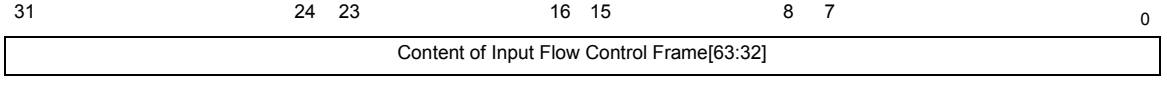
- Access: Non-Zero-Wait-State, Direct Access, Write only
- Address: h660



Bit [31:0] Content of flow control frame [31:0],
Flow Control Frame has 64 bytes and is defined by IEEE

18.2.10.3 AFCRID1 - Flow Control RAM Input Data 1

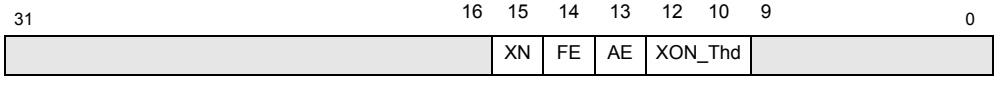
- Access: Non-Zero-Wait-State, Direct Access, Write only
- Address: h664



Bit [31:0] Content of flow control frame [63:32]

18.2.10.4 AFCR - Flow Control Register

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



Bit [9:0] Reserved

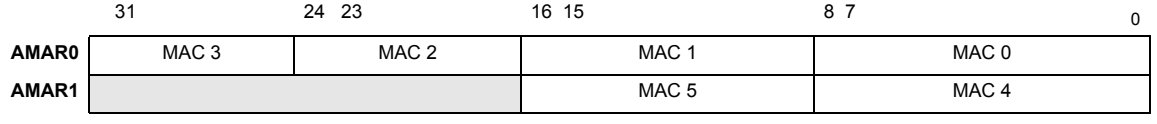
Bit [12:10] XON_Thd Defines the minimum # of free Frame Buffers before transmitting XON flow control frame.

$$XON\_Thd = \left\lfloor \frac{\min.\#offreeFCB}{8} \right\rfloor$$

Bit [13]	Queue Aging Enable	TX queue aging function enable
Bit [14]	Flush Enable	When stack is full, enable flush procession 0 = disable 1 = enable
Bit [15]	XON Enable	Full Duplex XON enable 0 = disable 1 = enable
Bit [31:16]	Reserved	

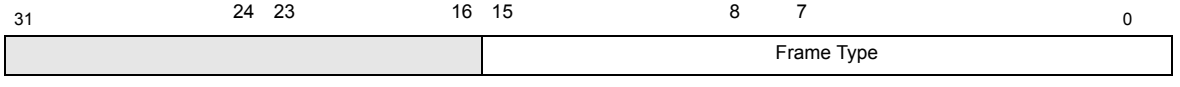
18.2.10.5 AMAR[1:0] - Multicast Address Reg. For MAC Control Frames

- This 6-byte MAC Address is stored in two 32-bit registers
  - AMAR0 MAC Address Byte [3:0]
  - Address: h674
  - AMAR1 MAC Address Byte [5:4]
  - Address: h678
- Access: Non-Zero-Wait-State, Direct Access, Write/Read



18.2.10.6 AMCT - MAC Control Frame Type Code Register

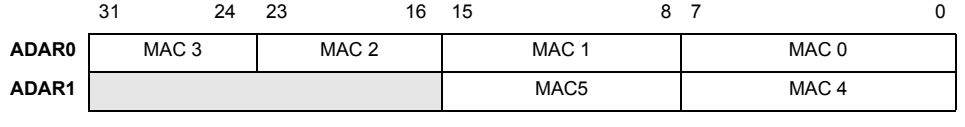
Access: Non-Zero-Wait-State, Direct Access, Write/Read
Address: h67C



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

18.2.10.7 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

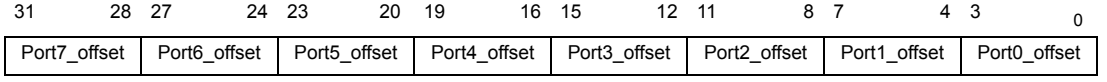


- These two registers define the base MAC address of the device.
- Bit [3:0] of Byte 0 (MAC5) is always set to 0.
- MAC address for each port is defined by

- MAC Address for Port n = Base MAC Address + MAC Offset [n] where n = {0..12}
- MAC Offset[n] is defined by the following registers

18.2.10.8 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



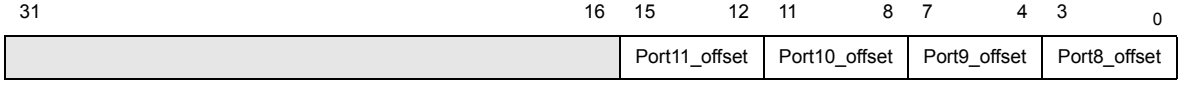
Bit [3:0] MAC Offset address for Port 0
Bit [7:4] MAC Offset address for Port 1
Bit [11:8] MAC Offset address for Port 2
Bit [15:12] MAC Offset address for Port 3
Bit [19:16] MAC Offset address for Port 4
Bit [23:20] MAC Offset address for Port 5
Bit [27:24] MAC Offset address for Port 6
Bit [31:28]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

Usage: There are three ways to assign the MAC address to each port. 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:

1. In a managed system, if the device does not support port trunking, MAC_Offset[3:0]= the port number.
2. In a managed system where device supports port trunking, the ports in the same trunk group shares the same MAC[3:0]. The value of MAC[3:0] is assigned by the smallest port number in the Trunk Group.
3. In a managed system, if BIT [18] of DCR2, SMAC=0, all ports are assigned to a single MAC.
4. In an unmanaged system, MAC[3:0] is fixed for all devices (i.e., only one MAC[3:0] address for the whole system).

18.2.10.9 ADAOR1 - MAC Offset Address Register 1

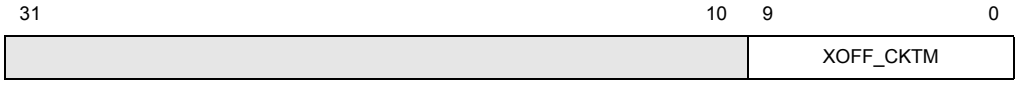
- MAC Offset Address for Port [12:8], 4-bit per port
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h60C



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 [31:16] Reserved

18.2.10.10 ACKTM - Timer For SOF Checking

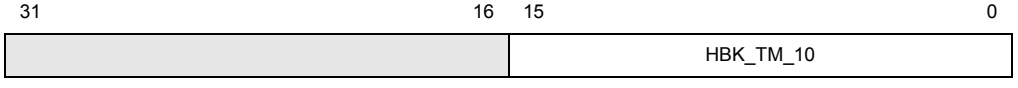
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h610



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

18.2.10.11 AFCHT10 - Flow Control Hold Time Of 10Mbps Port

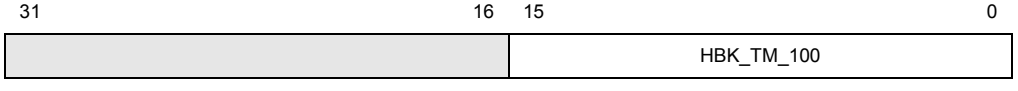
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h620



Bit [15:0]	HBK_TM_10	Holding time to remote station for head of line blocking control for 10M port.
Bit [31:16]	Reserved	

18.2.10.12 AFCHT 100 - Flow Control Hold Time Of 100Mbps Port

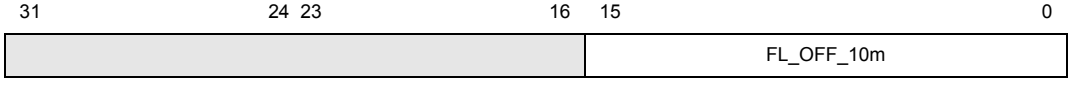
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h624



Bit [15:0]	HBK_TM_100	Holding time to remote station for head of line blocking control for 100M port.
Bit [31:16]	Reserved	

18.2.10.13 AFCOFT10 - Flow Control Off Time Of 10Mbps Port

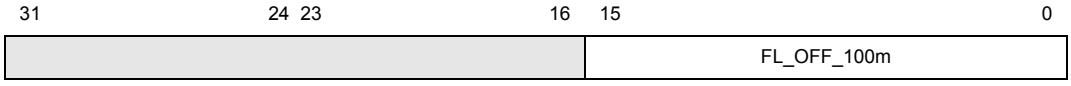
Access: Non-Zero-Wait-State, Direct Access Write/Read
Address: h614



Bit [15:0]	FL_OFF_10M	Off time to remote station for 10M Port.
Bit [31:16]	Reserved	

18.2.10.14 AFCOFT100 - Flow Control Off Time Of 100Mbps Port

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

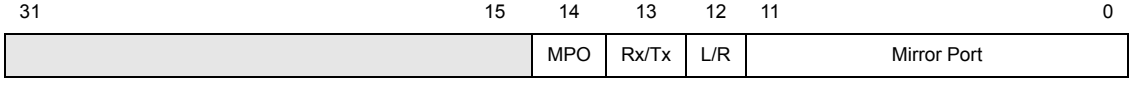


Bit [15:0]	FL_OFF_100M	Off time to remote station for 100M Port.
Bit [31:16]	Reserved	

18.2.11 Access Control Function Group 2 (Chip Level)

18.2.11.1 APMR - Port Mirroring Register

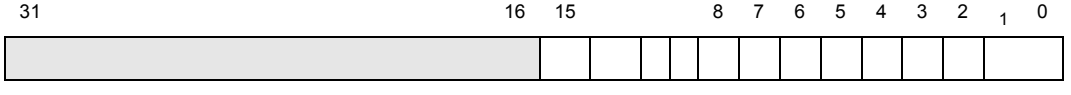
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h5C0



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=local 1=remote
Bit [13]	Rx/Tx	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	

18.2.11.2 PFR - Protocol Filtering Register

- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h5C4



The Search Engine will provide ingress filtering on a per-port basis. Each bit of PF register (default value = 0) will cause packets matching that category to be dropped.

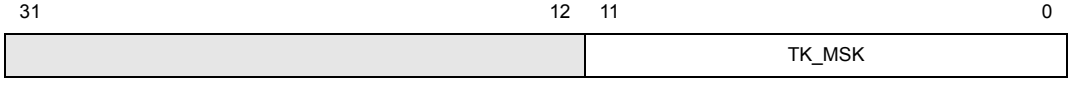
Bit [7:0]	Protocol Filter for Unicast Frames
Bit [0]	IP – Ethernet II encapsulation
Bit [1]	IP – 802_SNAP encapsulation
Bit [2]	IPX – Ethernet II encapsulation
Bit [3]	IPX – 802_SNAP encapsulation
Bit [4]	IPX – 802.2 encapsulation
Bit [5]	IPX – 802.3_RAW encapsulation
Bit [6]	Other (Packets with unknown encapsulation, or non-IP, non-IPX packets)

Bit [7]	Untagged Frames
Bit [15:8]	Protocol Filter for Multicast Frames
Bit [8]	Multicast IP – Ethernet II encapsulation
Bit [9]	Multicast IP – 802_SNAP encapsulation
Bit [10]	Multicast IPX – Ethernet II encapsulation
Bit [11]	Multicast IPX – 802_SNAP encapsulation
Bit [12]	Multicast IPX – 802.2 encapsulation
Bit [13]	Multicast IPX – 802.3_RAW encapsulation
Bit [14]	Multicast Other (Packets with unknown encapsulation, or non-IP, non-IPX packets)
Bit [15]	Multicast Untagged Frames
Bit [31:16]	Reserved

Usage: There is only one PFR register. For each port there is an enable bit (ECR1 bit 6: IFE- Ingress filter Enable) which determines whether the settings in PFR are applied to that port.

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

- Eight Trunking Hash Key Mask Registers shared the same format.
  - **THKM0** Trunking Forwarding Port Mask for hash key 0
- Address: h5C8
  - **THKM1** Trunking Forwarding Port Mask for hash key 1
- Address: h5CC
  - **THKM2** Trunking Forwarding Port Mask for hash key 2
- Address: h5D0
  - **THKM3** Trunking Forwarding Port Mask for hash key 3
- Address: h5D4
  - **THKM4** Trunking Forwarding Port Mask for hash key 4
- Address: h5D8
  - **THKM5** Trunking Forwarding Port Mask for hash key 5
- Address: h5DC
  - **THKM6** Trunking Forwarding Port Mask for hash key 6
- Address: h5E0
  - **THKM7** Trunking 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.

18.2.11.4 IPMCAS - IP Multicast MAC Address Signature

Usage: For following four registers IPMCAS0, IPMCAS1, IPMCMSK0 and IPMCMSK1, are used to distinguish between IP multicast traffic and regular multicast. The MAC for IP multicast are h"01:00:5e:00:00:00" to h" 01:00:5e:7f:ff:ff" And the MASK for IPMC is: h"ff:ff:ff:80:00:00".

- The 6-byte of IP multicast MAC Address is stored in two 32-bit registers
  - IPMCAS0 IP Multicast MAC Address Byte [3:0]
- Address: h5E8
  - IPMCAS1 IP Multicast MAC Address Byte [5:4]
- Address: h5EC
- Access: Non-Zero-Wait-State, Direct Access, Write/Read

	31 24	23 16	15 8	7 0
IPMCAS0	MAC 3	MAC 2	MAC 1	MAC 0
IPMCAS1			MAC 5	MAC 4

  - These two registers define the MAC address signature of IP multicast.
  - Default = h" 01:00:5e:7f:ff:ff"

18.2.11.5 IPMCMSK - IP Multicast MAC Address Mask

- The 6-byte of IP multicast MAC Mask is stored in two 32-bit registers
  - IPMCAS0 IP Multicast MAC Mask Byte [3:0]
- Address: h5F0
  - IPMCAS1 IP Multicast MAC Mask Byte [5:4]
- Address: h5F4
- Access: Non-Zero-Wait-State, Direct Access, Write/Read

	31 24	23 16	15 8	7 0
IPMCMSK0	MASK 3	MASK 2	MASK 1	MASK 0
IPMCMSK1			MASK 5	MASK 4

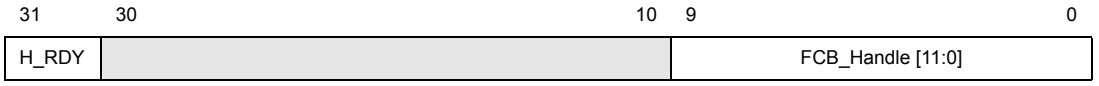
- These two registers define the MAC Mask of IP multicast.
- Default = h"ff:ff:ff:80:00:00".

18.2.11.6 CFCBHDL - FCB Handle Register For CPU Read

- Access: Non-Zero-Wait-State,Direct Access,Read only
- Address: h580

Usage: When CPU requests a free FDB to write a frame, it must request a free FCB via this register. The register contains a free handle of FCB, which also pointer to a free FDB.

- CPU reads FCB Handle: (When the CPU write FDB, it requires a FDB handle first).
- CPU checks CFCBHDL[31],H_RDY ready or not. If so, CPU gets the FCB Handle from CFCBHDL[9:0]



Bit [9:0]	FCB_HANDLE	FCB Handle Address
Bit [30:10]	Reserved	
Bit [31]	H_RDY	FCB Handle Ready 0=Not Ready 1=Ready

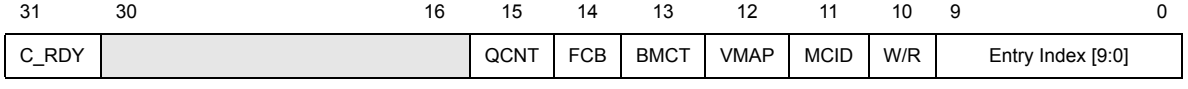
18.2.11.7 CPU Access Internal RAMs (Tables)

Usage: (refer to section 9.0 “The High Density Instruction Set Computer (HISC)” on page 37 for details)

- The CPU uses the following methods to access the five internal RAMs, including MCID, VLAN port mapping (VMAP), BM control Table (BMCT), FCB and Transmission Queue control (QCNT).
- Registers:
  - **CPUIRCMD**: Command register
  - **CPUIRDAT0**: Data Register for specific entry of content Bit[31:0]
  - **CPUIRDAT1**: Data Register for specific entry of content Bit[63:32]
  - **CPUIRDAT2**: Data Register for specific entry of content Bit[95 64]
  - **CPUIRRDY**: Data Read Ready.
- CPU Reads FCB
  - CPU write the read command into CPUIRCMD with FCB Handle, W/R=0. And set C_RDY. Also, set the table type = FCB, (CPUIRCMD[14]=1)
  - Frame Engine puts the specified FCB content into CPUIRDATL and CPUIRDATM
  - Frame Engine Clear C_RDY
  - Frame Engine set CPUIRRDY[0] to notify CPU that the FCB data is ready to be read.
- CPU writes FCB
  - CPU writes the content of FCB into CPUIRDATL and CPUIRDATM
  - CPU writes the handle of FCB into CPUIRCMD [9:0], set CPUIRCMD [10] = 1,(write CMD), set CPUIRCMD[31]=1, CMD_RDY and set the Table Index to FCB, (CPUIRCMD[14]=1).
  - Frame Engine clears CPUIRCMD [31], C_RDY, when Frame Engine reads FCB done
  - Apply the similar method to access the other four tables.

18.2.11.8 CPUIRCMD - CPU Internal RAM Command Register

- Access: Non-Zero-Wait-State,Direct Access,Write/Read
- Address: h584
- Command for CPU accesses five internal Tables

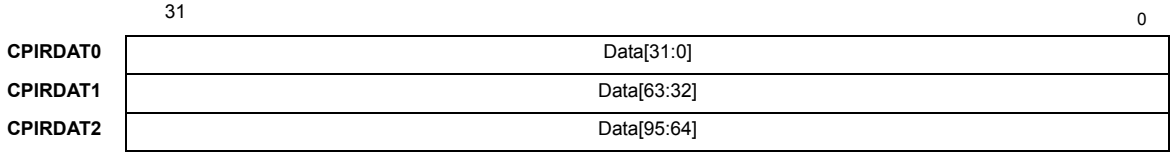


Bit [9:0]	Entry Index	The index of specified entry
	Type = MCID(16)	Entry index[3:0]
	Type = VMAP(256)	Entry index[7:0]
	Type = BMCT(1K)	Entry index[9:0]
	Type = FCB(1K)	Entry index[9:0]
	Type = QCNT (64)	Entry index[5:0]
Bit [10]	W/R	Write or Read the table entry 0=Read 1=Write

Bit[15:11]	Table bit map	Bit maps of five tables.
Bit[11]	MCID	MCID=1 Use MC ID l Table
Bit[12]	VMAP	VMAP=1 Use VLAN port mapping Table (VMAP)
Bit[13]	BMCT	BMCT=1 Use Buffer Manager Control Table (BM control)
Bit[14]	FCB	FCB=1 Use FCB Table
Bit[15]	QCNT	QCNT=1 Use Transmission Queue control Table (QM control)
Bit [30:16]	Reserve	
Bit [31]	C_RDY	Command Ready 0=Not Ready 1=Ready

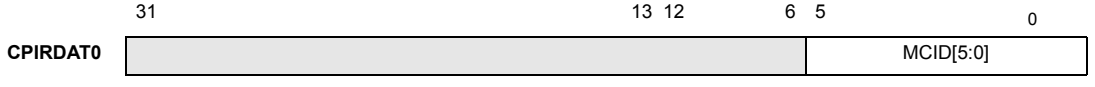
18.2.11.9 CPUIRDAT - CPU Internal RAM Data Register

- The 3 data registers are used when CPU reads or writes the content of the specified entry table.
  - CPUIRDAT0 CPU Internal RAM Data register for Data[31:0]
- Address: h588
  - CPUIRDAT1 CPU Internal RAM Data register for Data[63:32]
- Address: h58C
  - CPUIRDAT2 CPU Internal RAM Data register for Data[95:64]
- Address: h590
- Access: Non-Zero-Wait-State, Direct Access, Write/Read



The content is dependent on the type of table, as described below:

Type = MC ID (6bits)



Bit [5:0]	MCID	multicast ID FIFO data output (Note that up to 16 for this version.)
Bit [31:6]	Reserved	

Type = VMAP Table (27 bits)

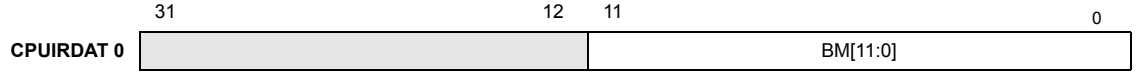
Bit [11:0]	VLAN Port Enable [12:0]one bit for each Ethernet MAC Port	Identify the ports associated with each VLAN 0 = disable 1 = enable
Bit [12]	Reserved	
Bit [24:13]	VLAN Tag Enable [12:0]one bit for each Ethernet MAC Port	0 = disable 1 = enable

Bit [25] Reserved

Bit [26] RE Remote Ports Enable: Indicate some members in the remote device.
0=disable 1=enable

Bit [31:27] Reserved

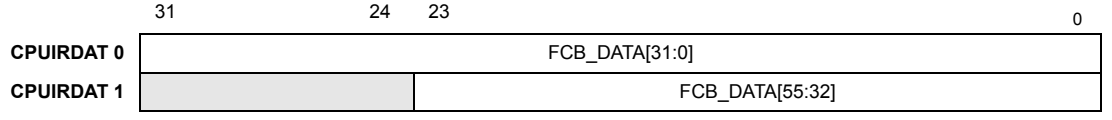
Type = BMCT (12bits)



Bit [11:0] BM Buffer Management control FIFO Output
BM stores free FCB handles. (FCB handle=0 cannot be used.)

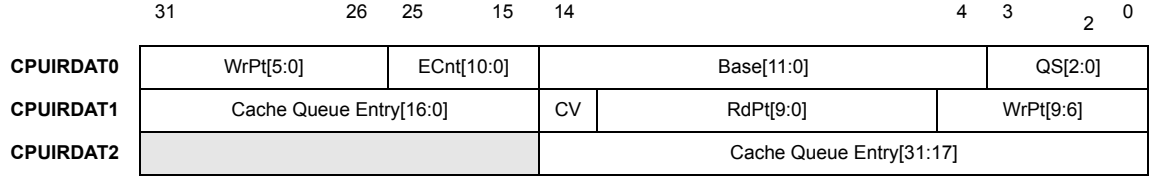
Bit [12:31] Reserved

Type = FCB (56 bits)



Bit [55:0] FCB Frame Control Block.
Refer to Section 9.0 “The High Density Instruction Set Computer (HISC)” on page 37 for detailed data structure.

Type = QCNT (79 bits)



Bit [2:0] Que_S [2:0] Queue size000=128 entries 001=128*2 entries
111=128*8=1K entries
Each entry contains 4 bytes

Bit [14:3] Base [11:0]Base pointer to its Transmission Queue

Bit [25:15] ECnt [10:0]Entry Count: Total entries in its queue.

Bit [35:26] WrPt [9:0]Write Pointer
Address_Write_Entry[20:9]=Base[11:0]+WrPt[9:7]
Address_Write_Entry[9:3]= WrPt[6:0]
Address_Write_Entry[2:0]= 0
(The address [2:0] is always equal to 0.)

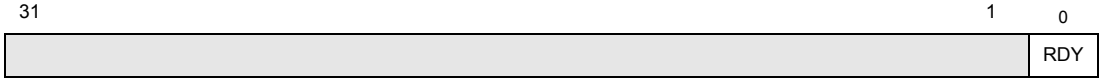
Bit [45:36] RdPt [9:0]Read Pointer
Address_Read_Entry[20:9]=Base[11:0]+RdPt[9:7]
Address_Read_Entry[9:3]= RdPt[6:0]
Address_Read_Entry[2:0]= 0 (The address [2:0] is always equal to 0.)

Bit[46] CV Cache Valid
CV=1, Cache of Queue Entry QE[31:0] is valid.

Bit[78:47] QE[31:0]Cache a queue entry

18.2.11.10 CPUIRRDY - Internal Ram Read Ready For CPU

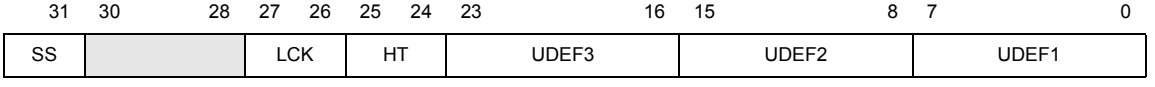
- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h594
- The Frame Engine sets this ready bit to notify the CPU that the requested data is ready to read.



Bit [0] R_RDY Data in Data registers is ready for CPU Read
Bit [31:1] Reserved

18.2.11.11 LEDR - LED Register

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



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] 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/8=12.5Mhz 01=100M/16=6.25Mhz
10=100M/32=3.125Mhz 11=100M/64=1.5625Mhz
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: 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.

18.2.12 Ethernet MAC Port Control Registers

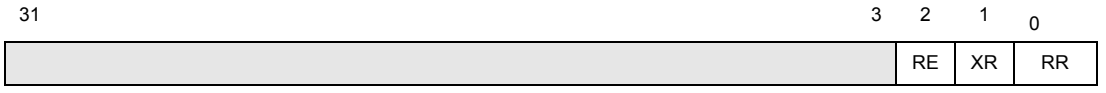
- One set for each Ethernet MAC Port [11:0]

18.2.12.1 ECR0 - MAC Port Control Register

- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h0x0*4 x: port n (n=0 - 11)

h000	ECR0_p0
h040	ECR0_p1
h080	ECR0_p2
h0c0	ECR0_p3
h100	ECR0_p4
h140	ECR0_p5
h180	ECR0_p6

h1c0 ECR0_p7
h200 ECR0_p8
h240 ECR0_p9
h280 ECR0_p10
h2c0 ECR0_p11



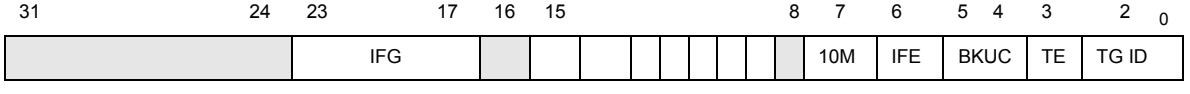
Bit [0]	RR	Reset Receiver
Bit [1]	XR	Reset Transmitter
Bit [2]	RE	RX Enable
Bit [31:3]	Reserved	

- Port is disabled when both RR & XR bits are set.

18.2.12.2 ECR1 - MAC Port Configuration Register

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



Configuration Bits Trunking

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 [5:4]	Block_UC_Frame	Instructs the Rx MAC to discard incoming Unicast Frames. This feature is used by Spanning Tree. 0X Blocking, all frames (Default state) 10 Learning but not forwarding 11 Forwarding all frames
Bit [6]	IFE	Ingress Filter Enable Default = 0

		Used to enable protocol filtering on a port by port basis. There is only one Protocol Filtering Register (PFR), but it can be used on any combination of ports. 0= disable ingress filter 1= enable ingress filter

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 cause 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:

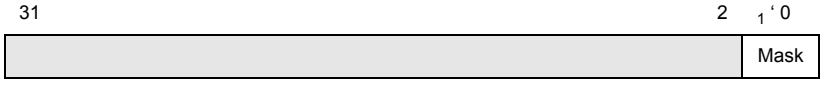
- The MAC Transmitter asserts flow control neither by sending Flow Control frames nor by jamming collision.
- The MAC Receiver still interprets and processes the Flow-Control frames. The MAC Receiver marks all Flow Control frames. Receive DMA discards the received Flow Control frames and send a status 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 [16]	Reserved	
Bit [23:17]	IFG	Inter-frame Gap (Default=7'd24) Use to adjust the inter-frame gap. (Unit =transmit Clock.) The default is 7'd24, stands for 24 transmit clock (each clock transmit 4 bits).
Bit [31:24]	Reserved	

18.2.12.3 ECR2 - MAC Port Interrupt Mask Register

• Access:	Non-Zero-Wait-State,	Direct Access,	Write/Read
• Address:	h0x2*4	x: port number	
	h008	ECR2_p0	
	h048	ECR2_p1	
	h088	ECR2_p2	

h0c8	ECR2_p3
h108	ECR2_p4
h148	ECR2_p5
h188	ECR2_p6
h1c8	ECR2_p7
h208	ECR2_p8
h248	ECR2_p9
h288	ECR2_p10
h2c8	ECR2_p11



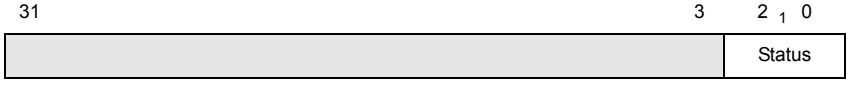
Bit [0]	WAS	If set, the status counter wrap around signal is masked.
Bit [1]	Link_Change	If set, the Link_Up and Link_Down Interrupts are masked.
Bit [31:2]	Reserved	

- Link Change interrupts are automatically disabled whenever both MAC Transmitter & Receiver are in Reset state – i.e. both XR & RR bits are set.

18.2.12.4 ECR3 - MAC Port Interrupt Status Register

- Access: Non-Zero-Wait-State, Direct Access, Read only
- Address: h0x3*4 x: port number

h00c	ECR3_p0
h04c	ECR3_p1
h08c	ECR3_p2
h0cc	ECR3_p3
h10c	ECR3_p4
h14c	ECR3_p5
h18c	ECR3_p6
h1cc	ECR3_p7
h20c	ECR3_p8
h24c	ECR3_p9
h28c	ECR3_p10
h2cc	ECR3_p11

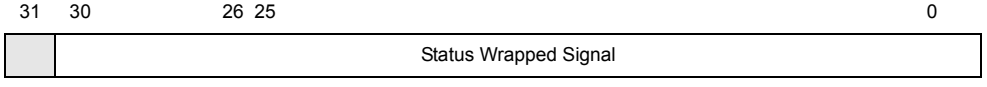


Bit [0]	WAS	Wrapped around signal.
Bit [1]	Link_Change	This bit is set when the MAC determines that the status of physical link has been changed
Bit [2]	LK_UP	0=Link Down, 1=Link UP This bit is reset whenever the PHY has identified the lost of physical link integrity.
Bit [31:3]	Reserved	

18.2.12.5 ECR4 - Port Status Counter Wrapped Signal

- Access: Non-Zero-Wait-State, Direct Access, Read only
- Address: h0x4*4 x: port number

h010	ECR4_p0
h050	ECR4_p1
h090	ECR4_p2
h0d0	ECR4_p3
h110	ECR4_p4
h150	ECR4_p5
h190	ECR4_p6
h1d0	ECR4_p7
h210	ECR4_p8
h250	ECR4_p9
h290	ECR4_p10
h2d0	ECR4_p11



B[0]. 0-d	Bytes Sent(D)
B[1]. 1-L	Unicast Frames Sent
B[2]. 1-U	Flow Control Sent
B[3]. 2-l	Non-unicast frame sent
B[4]. 2-U1	frame send fail
B[5]. 2-U2	Alignment Error
B[6]. 3-d	Bytes Received (Good or Bad) (D)
B[7]. 4-d	Frames Received (Good or Bad) (D)
B[8]. 5-d	Total Bytes Received (Good) (D)
B[9]. 6-L	Total Frames Received (Good)
B[10]. 6-U	Flow Control Frames Received
B[11]. 7-l	Multicast Frames Received
B[12]. 7-u	Broadcast Frames Received
B[13]. 8-L	Frames with length of 64 bytes
B[14]. 8-U	Jabber Frames
B[15]. 9-L	Frames with length between 65-127 bytes
B[16]. 9-U	Oversize Frames
B[17]. A-l	Frames with length between 128-255 bytes
B[18]. A-u	Frames with length between 256-511 bytes
B[19]. B-l	Frames with length between 512-1023 bytes
B[20]. B-u	Frames with length between 1024-1528 bytes
B[21]. C-l	Undersize Frames
B[22]. C-u	Fragment
B[23]. D-l	CRC
B[24]. D-u	Short Event
B[25]. E-l	Collision
B[26]. E-u	Drop

B[27]. F-l Filtering Counter
B[28]. F-U1 Delay exceed discard counter
B[29]. F-U2 Late Collision

Note: Each port owns a counter block, containing 16 double words. The 29 bits indicate that each corresponding counter is wrapping around the signal. The type and location of each counter is specified by the following format.

The format description:

X-Y: X means the relative Physical Address in its counter blocks.
: Y indicates the type of counter it is (Notation “C”= double word read from RAM block)

D: C[31:0] double word counter

L: C[23:0] 24 bits counter

U: C[31:24] 8 bits counter

U1: C[23:16] 8 bits counter

U2: C[31:24] bits counter (the same as notation “U”)

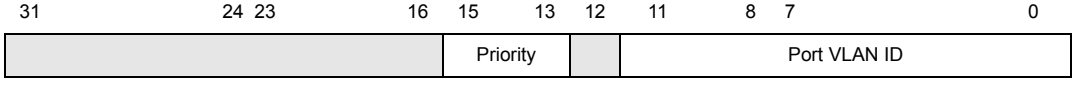
l: C[15:0] 16 bits counter

u C[31:16] 16 bits counter

18.2.12.6 PVID Register

- Access: Non-Zero-Wait-State, Direct Access, Write/Read
- Address: h0x9*4 x: port number
- For Default VLAN ID

h024	PVIDR_p0
h064	PVIDR _p1
h0a4	PVIDR _p2
h0e4	PVIDR _p3
h124	PVIDR _p4
h164	PVIDR _p5
h1a4	PVIDR _p6
h1e4	PVIDR _p7
h224	PVIDR _p8
h264	PVIDR _p9
h2a4	PVIDR _p10
h2e4	PVIDR _p11



Bit [0:11] Port VLAN ID (PVID)
Bit [12] Reserved
Bit [15:13] Priority
Bit [31:16] Reserved

19.0 DC Electrical Characteristics

19.1 Absolute Maximum Ratings

Package: 456 HBGA (Heatslug BGA)

Storage Temperature: -65C to +150C

Operating Temperature: 0C to +70C

Maximum Junction Temperature: 125C

Supply Voltage VCC with Respect to VSS +3.0 V to +3.6 V

Supply Voltage VDD with Respect to VSS +2.38 V to +2.75 V

Voltage on 5V Tolerant Input Pins -0.5 V to (VCC + 3.3 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.

19.2 DC Electrical Characteristics

VCC = 3.0 V to 3.6 V (3.3v +/- 10%) $T_{AMBIENT}$ = 0 C to +70 C
VDD = 2.5V +10% - 5%

Recommended Operating Conditions

Symbol	Parameter Description	Min	Type	Max	Unit
f_{osc}	Frequency of Operation		100		MHz
I_{CC}	Supply Current – @ 100 MHz (VCC =3.3 V)		220	286	mA
I_{DD}	Supply Current – @ 100 MHz (VDD =2.5 V)		720	936	
V_{OH}	Output High Voltage (CMOS)	2.4			V
V_{OL}	Output Low Voltage (CMOS)			0.4	V
V_{IH-TTL}	Input High Voltage (TTL 5V tolerant)	2.0		VCC + 2.0	V
V_{IL-TTL}	Input Low Voltage (TTL 5V tolerant)			0.8	V
I_{IL}	Input Leakage Current (all pins except those with internal pull-up/pull-down resistors)			10	μA
I_{OL}	Output Leakage Current			10	μA
C_{IN}	Input Capacitance			5	pF
C_{OUT}	Output Capacitance			5	pF
$C_{I/O}$	I/O Capacitance			7	pF
θ_{ja}	Thermal resistance with 0 air flow			12[1]	C/W
θ_{ja}	Thermal resistance with 1 m/s air flow			11	C/W
θ_{ja}	Thermal resistance with 2 m/s air flow			9.6	C/W
θ_{jc}	Thermal resistance between junction and case			3.3	C/W

Note 1: When external heat sink is attached, θ_{JA} is reduced by about 8-12% in still air.

20.0 AC Specification

20.1 XPIPE Interface

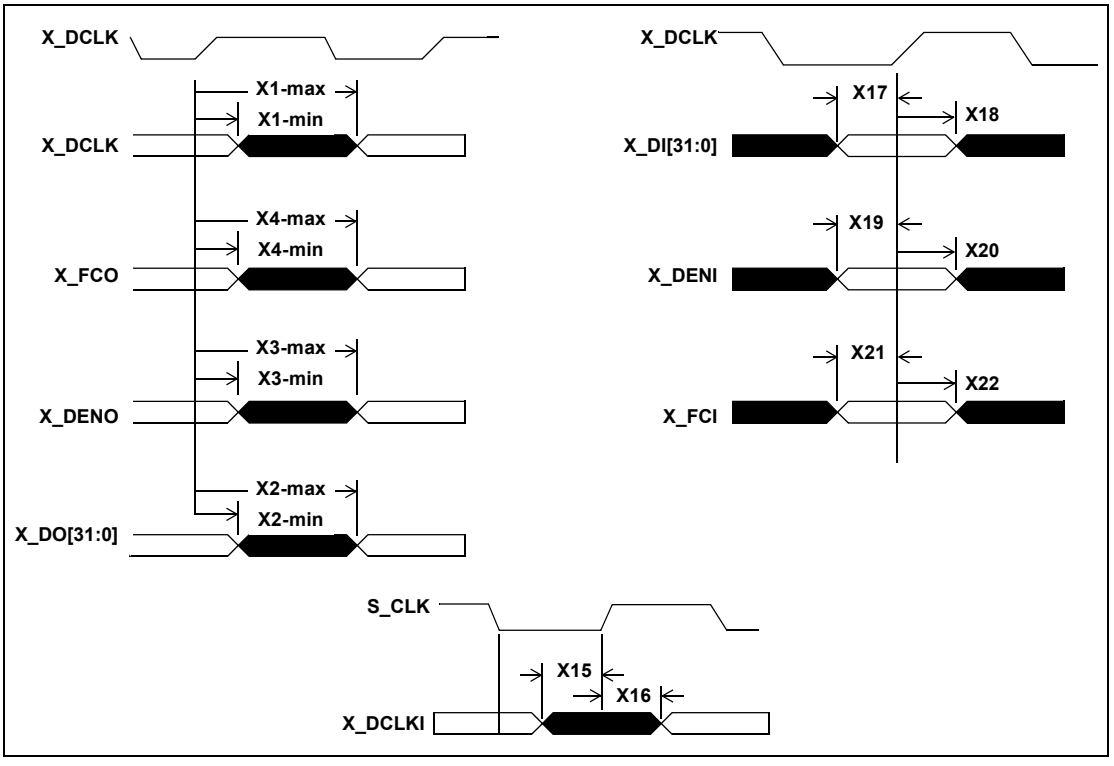


Figure 24 - XPIPE Interface - Output Valid Delay Timing

Symbol	Parameter	-100MHZ		Note
		Min (ns)	Max (ns)	
X1	X_DCLKO output valid delay	1	5	CL = 30pf
X2	X_DO[31:0] output valid delay	1	5	CL = 30pf
X3	X_DENO output valid delay	1	5	CL = 30pf
X4	X_FCO output valid delay	1	5	CL = 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 14 - AC Characteristics - XPipe Interface

20.2 CPU BUS Interface

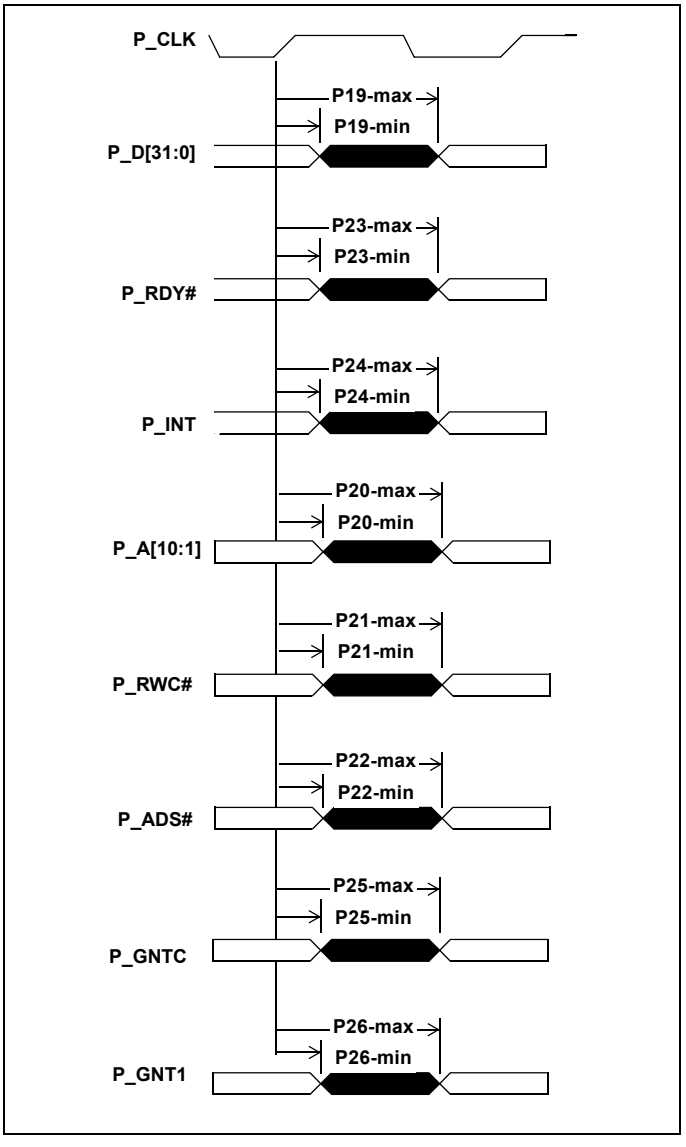


Figure 25 - CPU Bus Interface - Output Valid Delay Timing

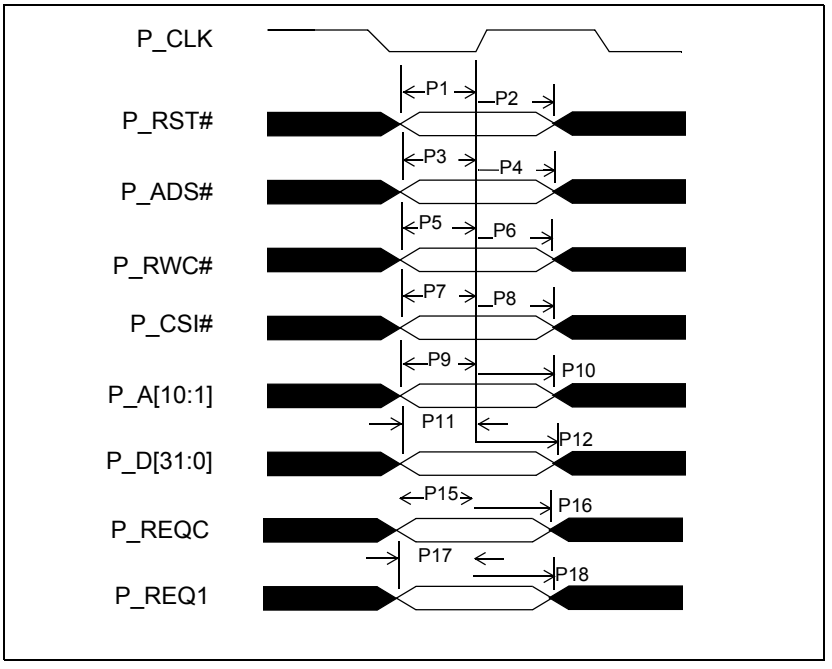


Figure 26 - CPU Bus Interface - Input Setup and Hold Timing

Symbol	Parameter	-66MHZ		Note
		Min (ns)	Max (ns)	
	P_CLK			
P1	P_RST# input setup time	6		
P2	P_RST# input hold time	2		
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		

Table 15 - AC Characteristics - CPU Bus Interface

Symbol	Parameter	-66MHZ		Note
		Min (ns)	Max (ns)	
P17	P_REQ1 input setup time	6		
P18	P_REQ1 input hold time	2		
P19	P_D[31:0] output valid delay	2	12	CL = 65pf
P20	P_A[10:1] output valid delay	2	9	CL = 50pf
P21	P_RWC# output valid delay	2	9	CL = 50pf
P22	P_ADS# output valid delay	2	9	CL = 50pf
P23	P_RDY# output valid delay	2	9	CL = 50pf
P24	P_INT output valid delay	2	9	CL = 30pf
P25	P_GNTC output valid delay	2	9	CL =20pF
P26	P_GNT1 output valid delay	2	9	CL =20pF

Table 15 - AC Characteristics - CPU Bus Interface (continued)

20.3 Local SBRAM Memory Interface

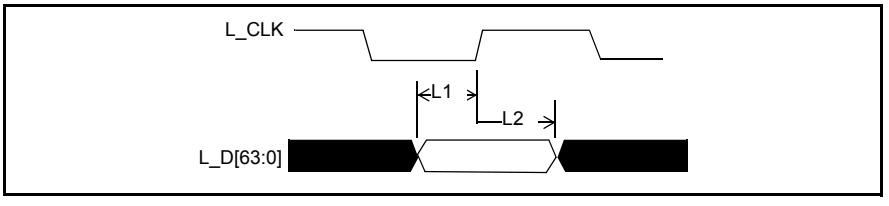


Figure 27 - Local Memory Interface - Input Setup and Hold Timing

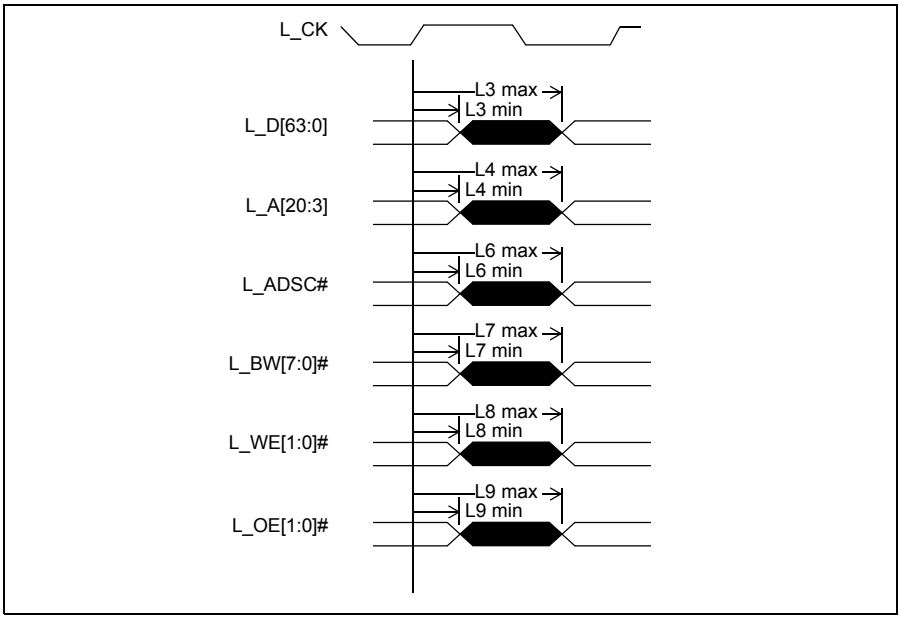


Figure 28 - Local Memory Interface - Output Valid Delay Timing

Symbol	Parameter	-100MHz		Notes
		Min (ns)	Max (ns)	
	L_CLK			C_L=50pf
L1	L_D[60:0] 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 16 - AC Characteristics - Local SBRAM Memory Interface

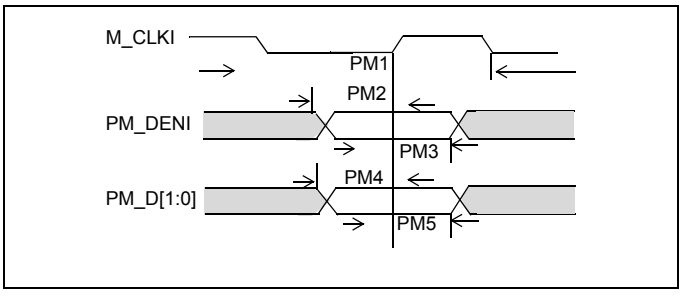


Figure 29 - Port Mirroring Interface - Input Setup and Hold Timing

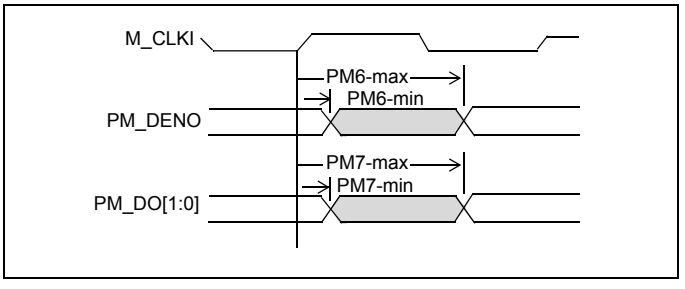


Figure 30 - Port Mirroring Interface - Output Delay Timing

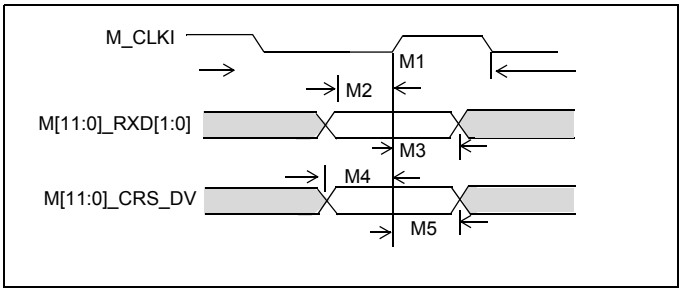


Figure 31 - Reduce Media Independent Interface - Input Setup and Hold Timing

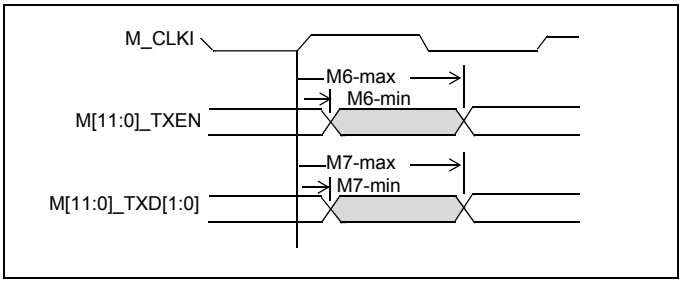


Figure 32 - Reduce Media Independent Interface - Output Delay Timing

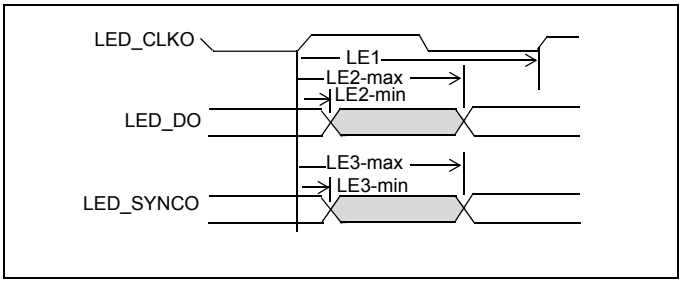


Figure 33 - LED Interface - Output Delay Timing

Symbol	Parameter	-50Mz		Notes
		Min (ns)	Max (ns)	
PM1	M_CLK			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	CL = 30 pF
PM7	PM_DO[1:0] Output Delay Time	2	11	CL = 30 pF

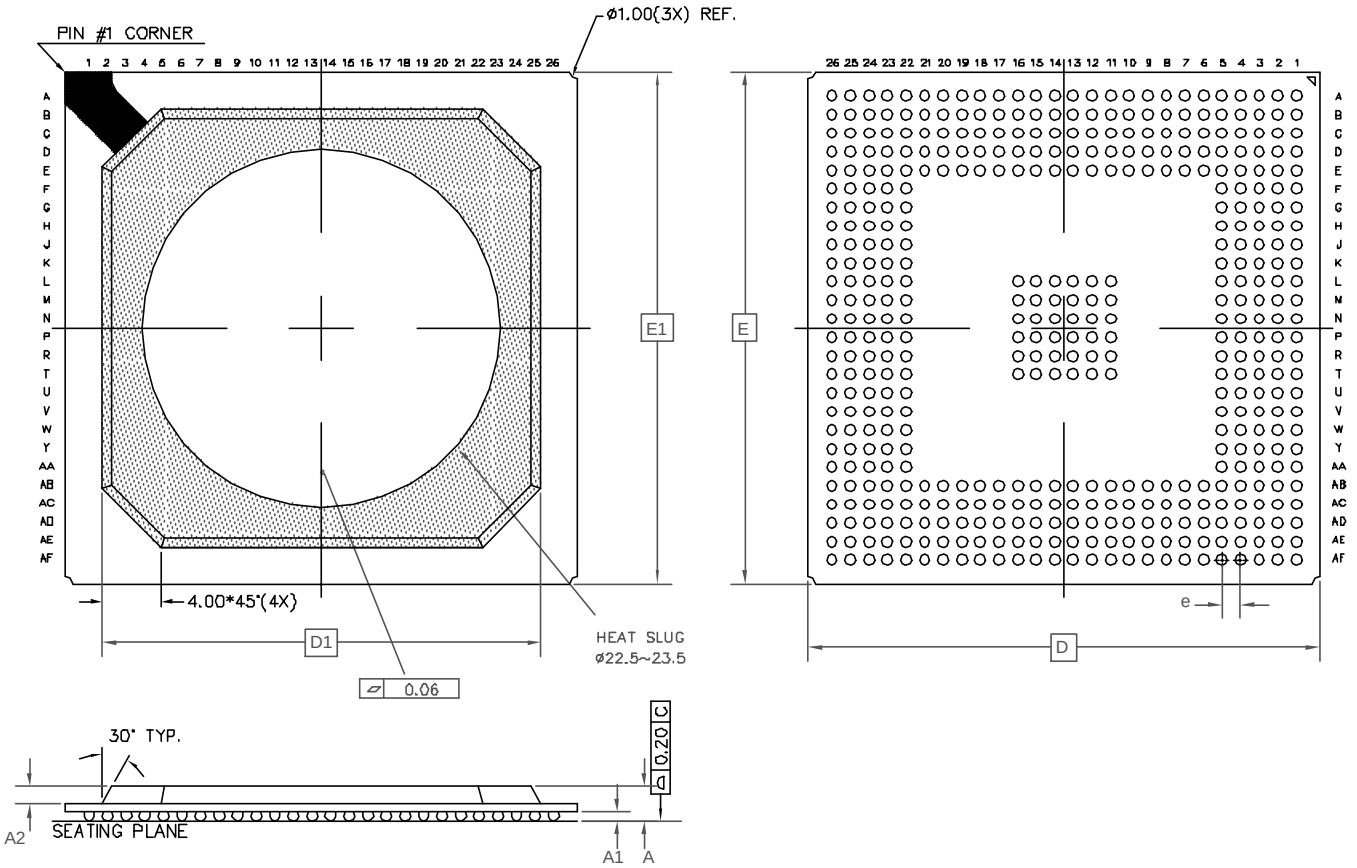
Table 17 - AC Characteristics - Port Mirroring Interface

Symbol	Parameter	-50Mz		Notes
		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 Setup Time	2		
M5	M[11:0]_CRS_DV Input Hold Time	2		
M6	M[11:0]_TXEN Output Delay Time	2	11	CL = 30 pF
M7	M[11:0]_TXD[1:0] Output Delay Time	2	11	CL = 30 pF

Table 18 - AC Characteristics - Reduced Media Independent Interfac

Symbol	Parameter	Variable Freq.		Notes
		Min (ns)	Max (ns)	
LE1	LE_DI Input LE_CLKO Times			Reference Output Clock
LE2	LE_DO Output Valid Delay	-1	7	CL = 30 pF
LE3	LE_SYNCO Output Valid Delay	-1	7	CL = 30 pF

Table 19 - AC Characteristics - LED Interface



DIMENSION	MIN	MAX
A	2.20	2.46
A1	0.50	0.70
A2	1.17 REF	
D	34.80	35.20
D1	30.00 REF	
E	34.80	35.20
E1	30.00 REF	
b	0.60	0.90
e	1.27	
N	456	
Conforms to JEDEC MS - 034		

1. CONTROLLING DIMENSIONS ARE IN MM
2. DIMENSION "b" IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER
3. PRIMARY DATUM -C- AND SEATING PLANE
   ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
4. N IS THE NUMBER OF SOLDER BALLS
5. NOT TO SCALE.
6. SUBSTRATE THICKNESS IS 0.56 MM

ISSUE	1			
ACN	213982			
DATE	3Feb03			
APPRD.				

ZARLINK SEMICONDUCTOR

Previous package codes: BH / G

Package Code GK

Package Outline for 456 Ball HSBGA (35x35x2.33mm)

GPD00826

For more information about all Zarlink products
visit our Web Site at

www.zarlink.com

TECHNICAL DOCUMENTATION - NOT FOR RESALE