

Epoch MultiLayer Switch Chipset

APPLICATIONS

- WAN edge routers
- MultiService platforms
- RAS platforms
- Group Switch/Router
- DSLAMs
- LAN PBX core

FEATURES AND BENEFITS

- MultiMedia-ready integrated switch on a chip
- Process Layer 3 and Layer 4 of the IP stack
- 1.4 Million packets/flow classifications per second; full Layer 4 flow recognition
- Up to 16 ports supported with powerful flexible built-in parsing function
- QoS Support for VoIP and other MultiMedia flows
- Differentiated Services per port (DS)
- Eight queues per output port enabling efficient MultiMedia integration of voice (VoIP), video, and data
- Two scheduling algorithms selectable on a per port basis.
- IPv6 and other protocols supported through processor interface
- No Head-of-line blocking
- Layer 2 switch-through support at wire speed
- Firewall assist on a per packet or per flow basis
- Flow aging support

DISTINCTIVE CHARACTERISTICS

- Wire speed Layer 3/Layer 4 switching for IPv4, IP Multicast, and IPX
- Header manipulation and checksum recalculation at wire speeds
- Support for Layer 3 CIDR (best prefix match)
- Per Flow and per IP or IPX address filtering options
- Behavior Aggregate Classification (BAC) and Microflow static/dynamic flow classification
- 64K default priority assignments with processor override for specific flows
- Eight levels of Weighted RR or eight levels of priority
- L3 to L2 support for IP to MAC address translation
- Destination and/or Source Port monitoring
- Generic 32 bit processor interface
- 66 MHz clock
- 3.3 Volt power with 5 Volt tolerant I/O pins
- IEEE 1149.1 (JTAG) boundary scan logic
- 456 PBGA Package

Related MUSIC Documentation:

Epoch Host Processor Software Development Manual

MUAC Routing CoProcessor (RCP) Family Data Sheet

AN-N25 Fast IPv4 and IPv4 CIDR Address Translation and Filtering Using the MUAC Routing CoProcessor (RCP) Application Note

AN-N27 Using MUSIC Devices and RCPs for IP Flow Recognition Application Note

OPERATIONAL OVERVIEW

The MUSIC Epoch MultiLayer Switch Chip for a Layer 3/4 switch performs all of the functions necessary to route IPv4, IPX and IP Multicast packets at wire-speed; to recognize and categorize traffic flows, optionally using IETF Differentiated Services (DS); and to queue each flow independently in an associated SDRAM. Upon transmission, DS information may be remarked. The Epoch handles up to 16 ports; one port is required for the processor to allow it to act as a packet source or destination.

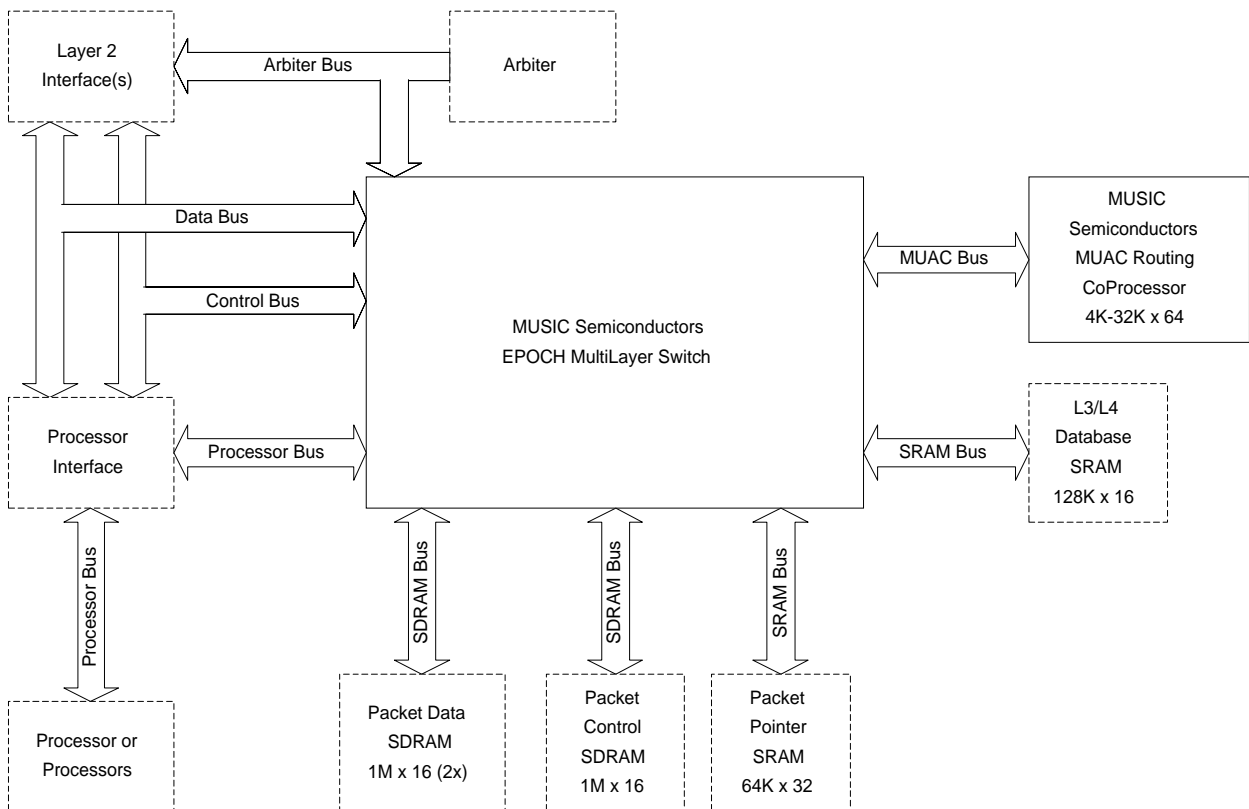
The Epoch chip fits into a system as shown in Figure 1. The Epoch chip itself is the heart of the system and forms the basis of a Layer 3/4 switch with up to 16 ports, one of which is used for the processor to send and receive packets. Each port has eight queues and a queue scheduler determines queue service order for each output port. Layer 3 and Layer 4 information are stored in a Routing CoProcessor (RCP) database. The RCP provides the

packet header processing performance necessary to do true wire-speed packet-by-packet routing and real-time flow recognition. The Epoch has a multicast switch fabric that also can be used for Layer 2 switches and xDSL multiplexers.

Various SRAM and SDRAM devices are required to store packet data and internal Epoch control information.

A processor provides non-real-time initialization and housekeeping functions. A processor also is used to handle packets destined to the switch and packets not supported by the Epoch. One processor may be used to handle both of these functions or separate processors may be used.

The Arbiter controls access to the bidirectional data bus among the Layer 2 ports, including the processor interface to the data bus. These components are detailed later.



Note: Solid boxes denote MUSIC standard products; dashed boxes denote either standard products; dashed boxes denote standard products from other manufacturers or customer ASICs/FPGAs/PLDs.

Figure 1: EPOCH MultiLayer Switch in a System

BALL DESCRIPTIONS

This section contains ball descriptions. Refer to Figure 2 below and Table 1, Ball Descriptions, on page 4.

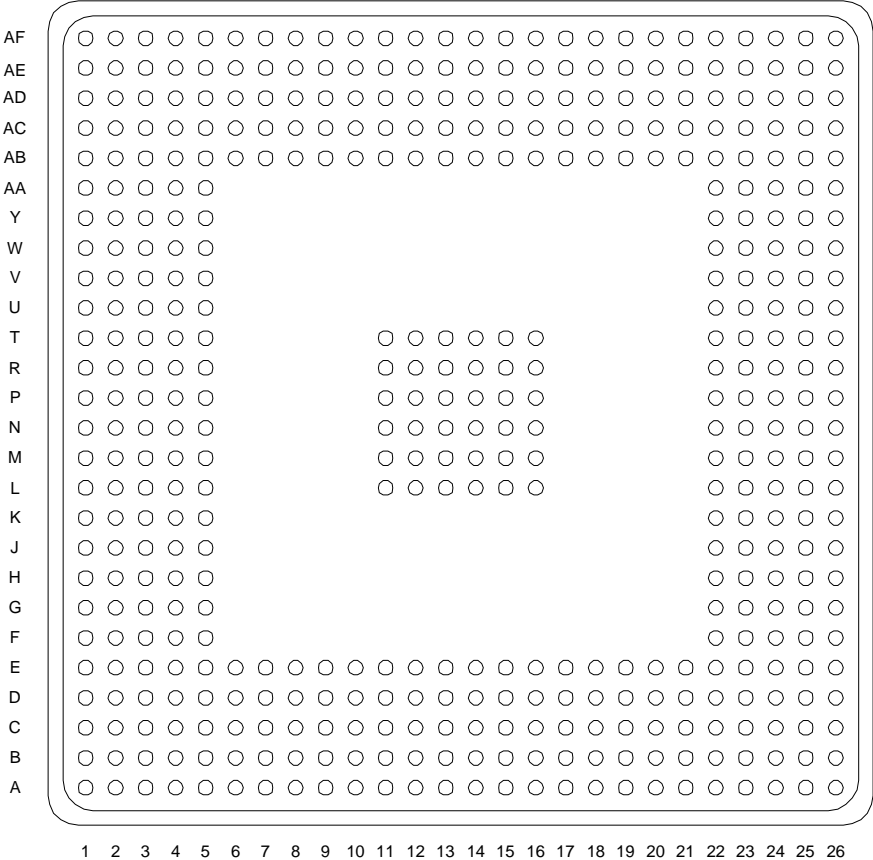


Figure 2: PBGA Ball Diagram (Underside View)

Table 1: Ball Descriptions

Functional Group	Ball Name(s) (Appended b indicates active low signal)	Function	Type	PBGA Ball(s)																																			
I/O TDM Bus	L2DATA[31:0] Interface to Layer 2 Devices.	TDM Data Bus. Transmit and receive data bursts in 64 byte blocks.	Bidir TTL 5V Tol.	b0:D14, b1:A13, b2:B13, b3:C13, b4:D13, b5:B12, b6:C12, b7:D12, b8:A11, b9:B11, b10:C11, b11:D11, b12:A10, b13:B10, b14:C10, b15:D10, b16:A9, b17:B9, b18:C9, b19:D9, b20:A8, b21:B8, b22:C8, b23:D8, b24:A7, b25:B7, b26:C7, b27:D7, b28:A6, b29:B6, b30:C6, b31:D6																																			
	L2CNTL[7:0]	TDM Control Bus. Transmit and receive control information.	Bidir TTL 5V Tol.	b0:D16, b1:A15, b2:B15, b3:C15, b4:D15, b5:A14, b6:B14, b7:C14																																			
	SYNC	TDM Timeslot. Synchronization Signal. Position is programmable.	Output TTL	B17																																			
	ABORT	Aborts the current receive packet indication. Assert for one CLK. May be asserted from the first word of a receive packet data burst, CLK 4 to CLK 47 of a bus cycle. Overrides or replaces LASTWORD and LASTBYTE[1:0].	Input TTL	D19																																			
	L2RXREADYIN	Asserted by L2 device when it has data next slot. Sampled by Epoch in CLK4.	Input TTL 5V Tol.	B16																																			
	L2TXREADYIN	Asserted by L2 device when it can accept data next slot. Sampled by Epoch in CLK4.	Input TTL 5V Tol.	A16																																			
	L2TXREADYOUT	Asserted to L2 device indicating data is available this slot. Asserted by Epoch in CLK11, de-asserted CLK30.	Output TTL	C16																																			
	L2LASTWORD	Asserted for one CLK at the CLK of the last word of a packet in the last buffer of a packet for both TX and RX packets.	Bidir TTL 5V Tol.	A17																																			
	L2LASTBYTE[1:0]	Indicates the last byte of the last word of a packet for both TX and RX packets. Encoding is as follows: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="2" style="text-align: center;">31</td> <td colspan="3" style="text-align: center;">L2DATA</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">Byte 0</td> <td style="text-align: center;">Byte 1</td> <td style="text-align: center;">Byte 2</td> <td style="text-align: center;">Byte 3</td> <td style="text-align: center;">L2LASTBYTE[1:0]</td> <td></td> </tr> <tr> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Invalid</td> <td style="text-align: center;">Invalid</td> <td style="text-align: center;">Invalid</td> <td style="text-align: center;">00</td> <td></td> </tr> <tr> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Invalid</td> <td style="text-align: center;">Invalid</td> <td style="text-align: center;">01</td> <td></td> </tr> <tr> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Invalid</td> <td style="text-align: center;">10</td> <td></td> </tr> <tr> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Valid</td> <td style="text-align: center;">Valid</td> <td style="text-align: center;">11</td> <td></td> </tr> </table>	31		L2DATA			0	Byte 0	Byte 1	Byte 2	Byte 3	L2LASTBYTE[1:0]		Valid	Invalid	Invalid	Invalid	00		Valid	Valid	Invalid	Invalid	01		Valid	Valid	Valid	Invalid	10		Valid	Valid	Valid	Valid	11		Bidir TTL 5V Tol.
31		L2DATA			0																																		
Byte 0	Byte 1	Byte 2	Byte 3	L2LASTBYTE[1:0]																																			
Valid	Invalid	Invalid	Invalid	00																																			
Valid	Valid	Invalid	Invalid	01																																			
Valid	Valid	Valid	Invalid	10																																			
Valid	Valid	Valid	Valid	11																																			
Arbiter Bus	L2NEXTPORT[3:0]	Inputs number of next active port. Sampled a TDM cycle ahead in CLK2.	Input TTL 5V Tol.	b0:A18, b1:B18, b2:C18, b3:D18																																			

Table 1: Ball Descriptions (continued)

Functional Group	Ball Name(s) (Appended b indicates active low signal)	Function	Type	PBGA Ball(s)
BFM Data SDRAM. Buffer Manager interface to data buffer RAM.	BSDRAMD[31:0]	Buffer Data SDRAM Data Bus	Bidir TTL 3.3V Only	b0:F24, b1:F23, b2:E26, b3:E25, b4:E24, b5:E23, b6:D26, b7:D25, b8:D24, b9:C26, b10:C25, b11:A25, b12:A24, b13:B24, b14:A23, b15:B23, b16:C23, b17:A22, b18:B22, b19:C22, b20:D22, b21:A21, b22:B21, b23:C21, b24:D21, b25:A20, b26:B20, b27:C20, b28:D20, b29:A19, b30:B19, b31:C19
	BSDRAMA[10:0]	Buffer Data SDRAM Address Bus	Output TTL	b0:J23, b1:H26, b2:H25, b3:H24, b4:H23, b5:G26, b6:G25, b7:G24, b8:G23, b9:F26, b10:F25
	BSDRAMBS	Buffer Data SDRAM Bank Select	Output TTL	J24
	BSDRAMRASb	Buffer Data SDRAM Row Address Strobe	Output TTL	J26
	BSDRAMCASb	Buffer Data SDRAM Column Address Strobe	Output TTL	J25
	BSDRAMWEb	Buffer Data SDRAM Write Enable Strobe	Output TTL	K23
	BSDRAMDQM	Buffer Data SDRAM Data Mask	Output TTL	K24
BFM Pointer SDRAM. Buffer Manager interface to pointer RAM.	PSDRAMD[15:0]	Control SDRAM Data Bus	Bidir TTL 3.3V Only	b0:P24, b1:P23, b2:N26, b3:N25, b4:N24, b5:N23, b6:M26, b7:M25, b8:M24, b9:M23, b10:L26, b11:L25, b12:L24, b13:L23, b14:K26, b15:K25
	PSDRAMA[10:0]	Control SDRAM Address Bus	Output TTL	b0:U23, b1:T26, b2:T25, b3:T24, b4:T23, b5:R26, b6:R25, b7:R24, b8:R23, b9:P26, b10:P25
	PSDRAMBS	Control SDRAM Bank Select	Output TTL	U26
	PSDRAMRASb	Control SDRAM Row Address Strobe	Output TTL	U24
	PSDRAMCASb	Control SDRAM Column Address Strobe	Output TTL	U25
	PSDRAMWEb	Control SDRAM Write Enable Strobe	Output TTL	V24
	PSDRAMDQMica	Control SDRAM Data Mask	Output TTL	V23

Table 1: Ball Descriptions (continued)

Functional Group	Ball Name(s) (Appended b indicates active low signal)	Function	Type	PBGA Ball(s)
PKM SRAM. Packet manager SRAM interface.	PKTSRAMDATA [31:0]	Packet SRAM Data Bus	Bidir TTL 3.3V Only	b0:AE22, b1:AD22, b2:AC22, b3:AF23, b4:AE23, b5:AD23, b6:AF24, b7:AE24, b8:AE26, b9:AD26, b10:AD25, b11:AC26, b12:AC25, b13:AC24, b14:AB26, b15:AB25, b16:AB24, b17:AB23, b18:AA26, b19:AA25, b20:AA24, b21:AA23, b22:Y26, b23:Y25, b24:Y24, b25:Y23, b26:W26, b27:W25, b28:W24, b29:W23, b30:V26, b31:V25
	PKTSRAMADDRESS [15:0]	Packet SRAM Address Bus	Output TTL	b0:AE18, b1:AD18, b2:AC18, b3:AF19, b4:AE19, b5:AD19, b6:AC19, b7:AF20, b8:AE20, b9:AD20, b10:AC20, b11:AF21, b12:AE21, b13:AD21, b14:AC21, b15:AF22
	PKTSRAMWEB	Packet SRAM Write Enable	Output TTL	AF18
	PKTSRAMOEb	Packet SRAM Output TTL Enable	Output TTL	AF17
CRI RCP. Glue free Connection to the MUSIC MUAC RCP.	RCPAC[12:0]	RCP Address Control Inputs	Bidir TTL 3.3V Only	b0:AC13, b1:AF14, b2:AE14, b3:AD14, b4:AC14, b5:AF15, b6:AE15, b7:AD15, b8:AC15, b9:AF16, b10:AE16, b11:AD16, b12:AC16
	RCPFFb	RCP Full Flag	Input TTL 3.3V Only	AC17
	RCPMFb	RCP Match Flag	Input TTL 3.3V Only	AE17
	RCPMMb	RCP Multimatch Flag	Input TTL 3.3V Only	AD17
	RCPAVb Dual use pin, see PLL section.	RCP Address Valid PLL ENABLE Pull HIGH through 4K7R if CLK = 50-66MHz Pull LOW through 4K7R if CLK = 0-50MHz	Bidir TTL 3.3V Only	AD5
	RCPCS20b	Chip select2 for first RCP	Output TTL	AF5
	RCPCS21b	Chip select2 for second RCP	Output TTL	AE5
	RCPCS22b	Chip select2 for third RCP	Output TTL	AF4
	RCPCS23b	Chip select2 for fourth RCP	Output TTL	AE4

Table 1: Ball Descriptions (continued)

Functional Group	Ball Name(s) (Appended b indicates active low signal)	Function	Type	PBGA Ball(s)
CRI RCP. Glue free Connection to the MUSIC MUAC RCP. (continued)	RCPDSC	Data Segment Control Output	Output TTL	AF3
	RCPDQ[31:0]	RCP Data Bus	Bidir TTL 3.3V Only	b0:AC5, b1:AF6, b2:AE6, b3:AD6, b4:AC6, b5:AF7, b6:AE7, b7:AD7, b8:AC7, b9:AF8, b10:AE8, b11:AD8, b12:AC8, b13:AF9, b14:AE9, b15:AD9, b16:AC9, b17:AF10, b18:AE10, b19:AD10, b20:AC10, b21:AF11, b22:AE11, b23:AD11, b24:AC11, b25:AF12, b26:AE12, b27:AD12, b28:AC12, b29:AF13, b30:AE13, b31:AD13
	RCPEb	RCP Chip Enable	Output TTL	AD1
	RCPOEb	RCP Output Enable	Output TTL	AD2
	RCPRESETb	RCP Reset	Output TTL	AD4
	RCPVb	RCP Validity Bit	Bidir TTL 3.3V Only	AE3
	RCPWb	RCP Write Enable	Output TTL	AF2
	RCPADDR[14:0]	CRI Shared RCP and Associated Data SRAM Address Bus. Connect to AA12:0 on RCPs and to Address pins of CRI SRAM.	Bidir TTL 3.3V Only	b0:W3, b1:W4, b2:Y1, b3:Y2, b4:Y3, b5:Y4, b6:AA1, b7:AA2, b8:AA3, b9:AA4, b10:AB1, b11:AB2, b12:AB3, b13:AB4, b14:AC1
Interface to the CRI Associated Data SRAM	SRAMADDR [16:15]	CRI High SRAM Address Bits SRAM Address [16:15]	Output TTL	b15:AC2, b16:AC3
	SRAMDQ[15:0]	CRI SRAM Data Bus	Bidir TTL 3.3V Only	b0:R3, b1:R4, b2:T1, b3:T2, b4:T3, b5:T4, b6:U1, b7:U2, b8:U3, b9:U4, b10:V1, b11:V2, b12:V3, b13:V4, b14:W1, b15:W2
	SRAMOEb	CRI SRAM Output Enable	Output TTL	R2
	SRAMWEb	CRI SRAM Write Enable	Output TTL	R1
Processor Interface	PDATA[31:0]	Processor Data Bus	Bidir TTL 5V Tol.	b0:C1, b1:D1, b2:D2, b3:D3, b4:E1, b5:E2, b6:E3, b7:E4, b8:F1, b9:F2, b10:F3, b11:F4, b12:G2, b13:G3, b14:G4, b15:H1, b16:H2, b17:H3, b18:H4, b19:J1, b20:J2, b21:J3, b22:J4, b23:K1, b24:K2, b25:K3, b26:K4, b27:L1, b28:L2, b29:L3, b30:L4, b31:M1

Table 1: Ball Descriptions (continued)

Functional Group	Ball Name(s) (Appended b indicates active low signal)	Function	Type	PBGA Ball(s)
Processor Interface (continued)	PRWb	Processor Read/Write	Input TTL 5V Tol.	P4
	PCSb	Processor Chip Select. May be asynchronous to CLK.	Input TTL 5V Tol.	P2
	PREADY	Processor Ready Active present cycle may complete	Output TTL	P3
	PADDR[8:2]	Processor Address	Input TTL 5V Tol.	b2:M3, b3:M4, b4:N1, b5:N2, b6:N3, b7:N4, b8:P1
	INT	Interrupt to Processor	Output TTL	M2
JTAG	TDI	Test Data In	Input TTL 5V Tol.	B5
	TCLK	Test Clock	Input TTL 5V Tol.	C4
	TMS	Test Mode Select	Input TTL 5V Tol.	B3
	TDO	Test Data Out	Output TTL	C5
	TRST	Test Reset	Input TTL 5V Tol.	A4
PLL	ZLOOP	Filter Input	Input	A3
Miscellaneous	RESETb	Device Reset. Must be active for 2 CLKS min. Chip takes 5 CLKS to come out of reset from RESETb LOW to HIGH	Input TTL Schmidt 5V Tol.	B4
	CLK	Device Clock	Input CMOS TTL 5V Tol.	A5
Power	3.3V	Device Power 3.3 Volts	N/A	AB11, AB13, AB14, AB16, AB18, AB20, AB7, AB9, E11, E13, E14, E16, E18, E20, E7, E9, G22, G5, J22, J5, L22, L5, N22, N5, P22, P5, T22, T5, V22, V5, Y22, Y5

Table 1: Ball Descriptions (continued)

Functional Group	Ball Name(s) (Appended b indicates active low signal)	Function	Type	PBGA Ball(s)
Power (continued)	GND	Device Ground	N/A	A1, A2, A26, AA22, AA5, AB10, AB12, AB15, AB17, AB19, AB21, AB22, AB5, AB6, AB8, AC23, AC4, AD24, AD3, AE1, AE2, AE25, AF1, AF25, AF26, B2, B25, B26, C24,C3, D23, D4, E10, E12, E15, E17, E19, E21, E22, E5, E6, E8, F22, F5, H22, H5, K22, K5, M22, M5, R22, R5, U22, U5, W22, W5, L16-11, M16-11, N16-11, P16-11, R16-11, T16-11
	VDDWELL5V	Connect to 5 Volts for 5 Volt tolerant buffers. Connect to 3.3 Volts if no 5 Volt tolerance is required.	N/A	G1, A12
	AVDD	Analog VDD. Quiet VDD for PLL (3.3V)	N/A	B1
	AVSS	Analog VSS. Quiet VSS for PLL.	N/A	D5
	VDDWELLPLL	PLL well 3.3V. Connect to quiet VDD.	N/A	C2

FUNCTIONAL OVERVIEW

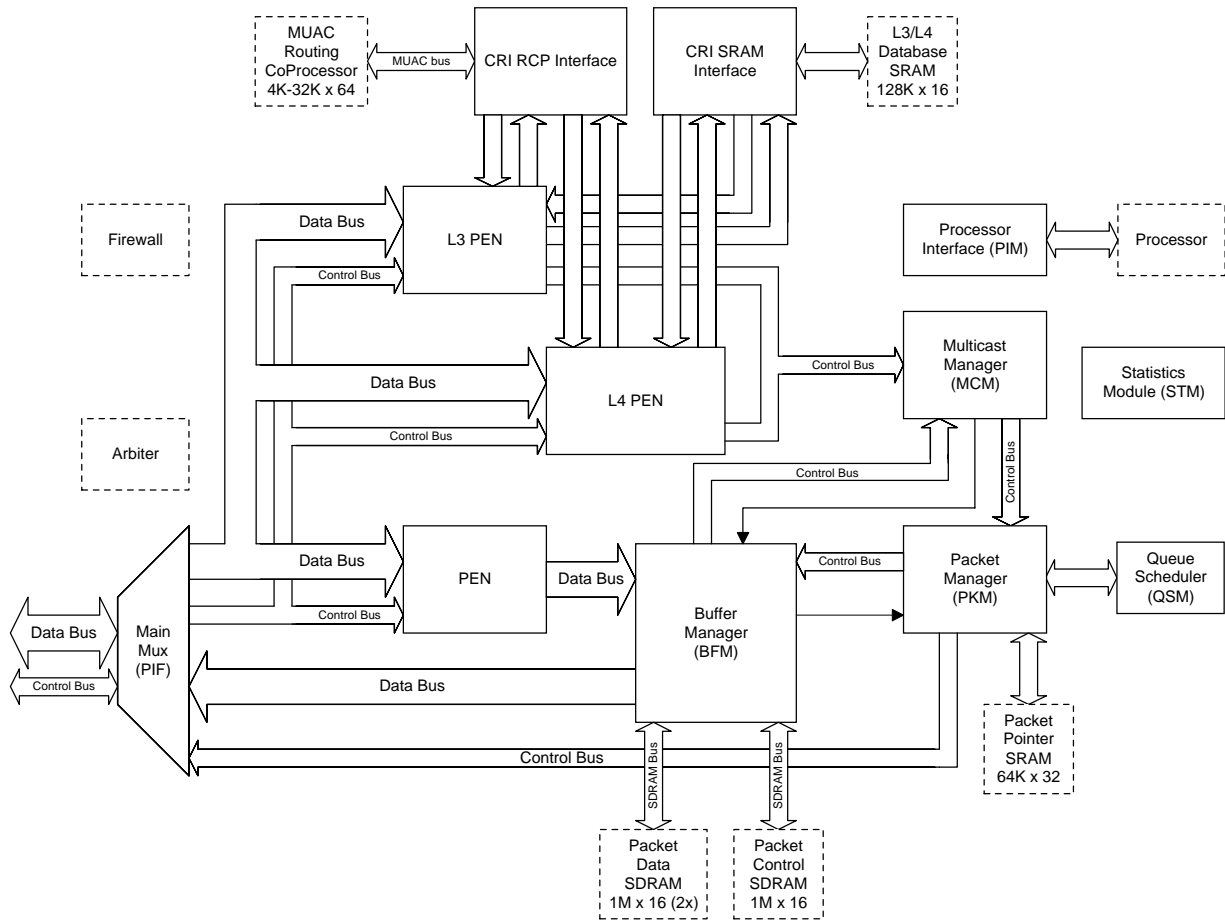


Figure 3: Epoch MultiLayer Switch Functional Block Diagram

Block Diagram Overview

The functional block diagram above (Figure 3) splits the Epoch chip into several blocks. Data moves to and from the Epoch via data and control buses with synchronized timeslots. The timeslots are long enough to move 64 bytes of data into the Epoch followed by 64 bytes of data from the Epoch. For longer packets, multiple timeslots are necessary. Timeslots for different Layer 2 ports interleave on the bus. Timeslots are 48 clock cycles and the minimum clock period is 15 ns; 16 clock cycles are for data movement into the Epoch and 16 clock cycles for data movement from the Epoch via a 32-bit bus. The remaining cycles are for system overhead.

There are eight queues for each of the 16 output ports. An on-chip scheduler with two modes of operation schedules among the queues. Queue zero has the highest priority and queue seven has the lowest priority.

- Mode 1- Weighted Round Robin
- Mode 2- Priority Order

Modes are selectable per port.

Main Multiplexer (PIF)

The Main Multiplexer (PIF) interfaces the bidirectional data and control buses used outside the Epoch chip to unidirectional data and control buses used inside the chip. The major function of the PIF is bus multiplexing. In addition, the PIF inserts Differentiated Service information to selected transmitted packets.

Buffer Manager (BFM)

The Buffer Manager (BFM) is responsible for storing and extracting packets from the SDRAM. All packet switching occurs in the SDRAM. The Packet Data SDRAM stores the packet data received by the Epoch. The Packet Control SDRAM stores linked list information as well as data associated with each packet. Separate SDRAMs are used for packet data and BFM control information. If a packet is larger than the size of a buffer, then the BFM is responsible for chaining together buffers to hold an entire packet.

The BFM also tracks the number of queues that point to each packet. This number decrements after packet transmission. When this number reaches zero, the buffers return to the free buffer pool.

L3/L4 Database

The L3/L4 Database RCP consists of up to four cascaded RCPs from the MUSIC MUAC Routing CoProcessor family. These RCPs provide the ability to perform 32-bit longest match address lookup or 64-bit exact address lookup. The RCPs contain the IP, IPX and IP Multicast routing tables, along with matched (see CRI section), authenticated flow information. See MUSIC Application Note AN-N25 for more detailed information on Layer 3 address lookup. See MUSIC Application Note AN-N27 for more detailed information on Layer 4 flow recognition.

RCP/RAM Interface (CRI)

The RCP/RAM Interface (CRI) interface contains the state machines for driving the RCP and associated data RAM as well as the logic to arbitrate among the sources of RCP operations (PEN and PIM).

Database SRAM

The external L3/L4 Database SRAM contains the Differentiated Services data associated with each word contained in the L3/L4 Database RCP as well as default queuing information based on TCP/UDP port number for unrecognized flows and the Differentiated Services BAC mode Table.

Parse Engine (PEN)

The Parse Engine (PEN) comprises the L3 data manipulation, L3 Lookup and L4 Lookup sections. The L3 Data Parse Engine makes the appropriate changes to the header of each IP or IPX packet that passes through the Epoch, such as decrementing the TTL field and recomputing the header checksum on IP or the Transport Control field on IPX. If the packet is not an IP or IPX packet, the header is unchanged. The PEN verifies various fields of the IPv4 or IPX header.

The L3 Engine in the PEN is responsible for extracting the IPv4, IP Multicast, or IPX address from the packet and using the RCP (via the CRI) to determine the output port(s) to which the packet is to be sent. The RCP returns the index of a matching entry and this index selects a 16-bit output port bitmap from a separate SRAM (via the CRI). If the packet is directly queued because it is not an IP or IPX packet, information passed to the Epoch on the control bus determines the port(s) for the packet.

The L4 Parse Engine in the PEN recognizes distinct packet flows travelling through the network. Traffic may be classified using the proposed IETF Differentiated Services. Filtering is also supported on a per flow basis. Traffic is classified in one of two modes:

- BAC (Behavior Aggregate Classification) Mode: The DS field is used to address a RAM that returns a Queue number for the destination port. The Mapping Table is processor maintained.
- Microflow Mode: The parse engine extracts the IP and L4 header including TCP or UDP port number, source and destination IP address fields and incoming L2 interface number. Packets are then prioritized to one of eight queues on the destination port from the default flow table or from the matched flow table if a match is found.

Packet Manager (PKM)

The Packet Manager (PKM) inserts and extracts packet pointers from queues. During packet reception, the PKM receives a queue number and packet pointer from the MCM, and adds the packet pointer at the tail of the specified queue. During packet transmission, the PKM receives a queue number from the Queue Scheduler Module (QSM), and it extracts the packet pointer at the head of the specified queue and passes it to the BFM.

The internal queue pointer SRAM contains the information that associates packets with queues. The QSM determines the service order for output port queues. The QSM returns the queue number (if any) for the next transmission.

PIM and STM Modules

The Processor Interface Module (PIM) interfaces an external processor to the Epoch. The processor has access to various blocks of RAM and RCP, as well as a number of registers for controlling and monitoring the Epoch.

The Statistics Module (STM) accumulates counts of dropped packets.

Arbiter

The Arbiter is a customer-designed system that tells the Epoch for each time slot, which port number is active. The Arbiter may allocate slots to an idle port. It is up to the individual L2 devices to inform the Epoch whether an operation is required for the cycle.

Processor

The Processor is for Epoch initialization and control. Control includes setting flow parameters, processing certain types of packets, updating the routing table, etc.

Packet Processing Methodology

Packets arrive at the Layer 2 interface device (not part of the Epoch). If the packet is an IP or IPX packet (as indicated by a field in the Layer 2 protocol), then the Layer 2 device simply removes the Layer 2 header and trailer and passes the packet to the Epoch, along with control information indicating that the packet is an IP, re-injected IP or IPX packet. If the packet is not IP or IPX, the Layer 2 device makes a switching decision based on the Layer 2 header, and then gives the whole packet to the Epoch along with information about the queue(s) and port(s) to which the packet should be sent. Packets destined for the processor are sent to whichever port is designated as the host port.

The Layer 2 device has signals indicating to the Epoch that it is not ready to transmit or receive (L2RXREADYIN and L2TXREADYIN), even if the Arbiter has assigned the timeslot to the given Layer 2 device. Similarly, the Epoch has a similar signal that indicates that it is unable to use a timeslot (L2TXREADYOUT). The Epoch asserts this signal only for data movement from the Epoch to the Layer 2 device when there are no packets available for the given Layer 2 device.

Packets enter the Epoch via a 32-bit bidirectional data bus (L2DATA[31:0]). The bus is shared among devices that connect to various ports, and packets move across the bus in blocks of 64 bytes. Blocks for packets of different Layer 2 ports interleave. Control information is provided with each block to indicate its position in the packet, the number of data bytes in the block, and information about how the packet is to be processed.

The data and control information are kept as separate as possible in the Epoch to ease implementation and test. We shall describe the two flows separately, starting with the data flow for received packets. Remember that for each timeslot, the Epoch performs a receive operation followed by a transmit operation.

Received Packet Data Flow

Data arrives at the PIF from the external bidirectional TDM data bus and it is sent to the PEN. If a packet arrives with explicit information as to the output ports and queues for which the packet is destined, the PEN passes the packet through unchanged.

If the packet is indicated to be IP, the IP header is checked for version number (IPv4) and IP options. If the IP version is not IPv4 or options are in use then the packet is sent to the processor port. Otherwise, the packet is transferred to the packet memory and the PEN locates and modifies the appropriate header fields. Specifically, the Time-to-Live (TTL) field is decremented and the checksum field is recomputed. If the new TTL value is zero, then the packet is forwarded to the processor port. Otherwise, the checksum field is recomputed incrementally as described in Network Working Group RFC 1624.

If the packet is IPX and the Transport control field is ≥ 15 , then the packet is sent to the host port. If the source or destination network number is zero it is replaced with the source port network number. The destination port is derived from the destination network number. The Transport Control field is incremented.

The packet then is passed to the BFM. The BFM is responsible for storing packets in SDRAM. Each block of a packet is stored in a different 64 byte memory block, and the memory blocks are chained together as a linked list. Note that packet data and BFM control information are kept in separate SDRAM devices; however, there is a one-to-one correspondence between data and control blocks.

Transmitted Packet Data Flow

The BFM receives the memory location of the packet to be transmitted from the PKM. The first block of the required packet is sent to the TDM bus via the PIF. On each following transmit cycle for that specific port, additional chained blocks that contain the packet are sent until the entire packet has been transmitted. When the last packet buffer has been reached, the BFM examines a count field that indicates the number of queues that still contain this packet (besides the current queue). If the number is zero, then the packet buffers are returned to the pool of unused packet buffers. Otherwise, the number is decremented and written back to memory. The PIF is also responsible for remarking the DS field and recalculating the check sum in selected transmitted IP packets from information passed to it from the BFM.

Received Packet Control Flow

Data arrive at the PIF from the external bidirectional TDM bus and are sent to the L3 PEN. As a packet is transferred to the packet memory, the PEN locates the destination IP or IPX address of the packet, and determines whether the packet is a unicast or a multicast packet. The IP or IPX address is compared against a database of IP or IPX addresses in the RCP database via the CRI. The value returned is an index that is used to search an SRAM of associated data via the CRI. The SRAM contains a port bitmap that indicates to which output port(s) the packet is to be sent. This information is sent to the MCM.

In parallel, the packet data is passed to the L4 PEN to perform flow recognition.

If the receive port is programmed to BAC mode the DS field is parsed and the destination Queue read from the DS to Queue mapping.

If the receive port is programmed to DS microflow mode, it then extracts the L4 information from the packet: the IP source and destination addresses, and the TCP/UDP source and destination port numbers. The extracted information, as well as the Layer 2 input port number, is compared with the L4 RCP database. If an exact match of the header fields is found in the RCP (via the CRI), then the queue number associated with that entry is returned from SRAM via the CRI using the index of the matching RCP address. Otherwise, a default queue number is obtained from a 64K entry SRAM table indexed by L4 port numbers via the CRI. If The DS field is to be changed, the replacement DS field is obtained from the L4 Associated Data. The default entry also indicates whether the packet is to be sent to the processor for flow recognition.

Each time a L4 RCP entry is matched by a packet, the activity flag is set. The processor walks through the activity table, noting which entries contain reset activity flags and resetting activity flags that have been set. If an entry is inactive for a sufficient time, it may be deleted from the table if required.

RCP entries are written by the processor. When a packet from an unknown flow arrives, it is sent to a default queue based on its TCP/UDP port number. The packet is diverted to the processor if the TCP/UDP port number is indicative of a flow that possibly should be learned.

The PEN passes the queue number for a packet to the MCM. The MCM uses the queue number and output port bitmap to create entries in one or more queues that point to the same packet. For a unicast packet, the queue number is matched with the packet pointer from the BFM, and the pair is passed to the PKM. The MCM passes a packet pointer and a queue number to the PKM for each packet in the unicast or multicast. The PKM places an entry with the packet information, including the packet pointer, into the appropriate queue. If the queue is empty before the packet is queued, the QSM is notified that the queue must be added to the queue-servicing schedule.

The QSM handles up to eight queues per port. Two algorithms are implemented on a per port basis.

Transmitted Packet Control Flow

The Arbiter informs the Epoch which port will use the next timeslot. If a new packet must be transmitted the PKM receives this information and sends a request to the QSM to provide the number of the queue that contains the packet to be sent to the given output port. The QSM passes back the queue number. The PKM finds the packet information at the head of the given queue, extracts the SDRAM memory location at which the first buffer of the packet is stored, and passes this information to the BFM. The BFM then transmits the packet, as described above in the section on Data Flow for Transmitted Packets. Control information is passed with the packet, including the port on which the packet arrived, and the queue number in which the packet was stored.

EPOCH OPERATION

*Note: This data sheet contains numerous references to Epoch clock cycles. The cycle number (e.g. CLK 12) refers to the cycle in which EPOCH asserts a signal (transmit) or samples a signal (receive). All signals are asserted or sampled on the rising edge of the referred to cycle. For example, Table 2 refers to the clock cycles that Epoch **samples** for the control information. Table 3 refers to the clock cycles that Epoch **asserts** for the control information.*

TDM I/O Data Bus

Data

Transmit and receive data is passed between Epoch, the L2 interfaces, and the host processor over this bus. Receive data is passed in the first half of the cycle and transmit in the second half. Data is burst in 64 byte blocks. The end of a packet is indicated to and from Epoch with the L2LASTWORD signal in the same CLK as the last word of the dataword. The last byte of the last word is indicated by the L2LASTBYTE[1:0] signals. See the Timing Diagrams section for detailed timing information.

Abort

In the event a receive packet needs to be aborted from the L2 interface for events such as CRC errors, underruns, etc., the ABORT signal may be asserted from CLK 4 up to CLK 47 of the TDM cycle. The packet being received on that port will be aborted. The buffers in use will be returned to the free list.

SYNC Pulse

The SYNC pulse indicates the start of each TDM cycle. The clock cycle that samples SYNC is referred to as CLK 0. Its position may be altered relative to the rest of the TDM bus to provide synchronization indication at a position useful to the user interface logic. The default position is CLK 0 of the TDM cycle which is three clocks

relative to the first cycle of the TDM data. See Table 5 for a description of the ProgSyncReg register.

FLOW Control

Epoch requires that an arbiter is connected to the L2NEXTPORT[3:0] input bus. This informs Epoch on CLK 2 of each TDM slot which physical port is the "next" port to be serviced. This information assures that Epoch and any interface hardware is always aware of the currently serviced port and the port to be serviced next. Epoch asserts L2TXREADYOUT at CLK 11 of the current TDM slot if TX data is available for the current slot.

Epoch samples both the L2TXREADYIN and L2RXREADYIN signals at CLK 4. Therefore, any interface hardware should assert these signals prior to CLK 4 and remain asserted until at least CLK 5 if the following conditions are satisfied:

- L2TXREADYIN—Asserted if the interface hardware can accept TX data in the next TDM slot.
- L2RXREADYIN—Asserted if the interface hardware is going to pass data to Epoch in the next TDM slot.

Table 2: Receive Control Bus Inputs

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Clock
MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0	3
MC15	MC14	MC13	MC12	MC11	MC10	MC9	MC8	4
0	Q2	Q1	Q0	0	0	Parse	Parse	5

Note: Clock 5, bits 7, 3, and 2 must be set to 0 to ensure proper Epoch operation.

Table 3: Transmit Control Bus Outputs

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Clock
L3Index7	L3Index6	L3Index5	L3Index4	L3Index3	L3Index2	L3Index1	L3Index0	21
N/A	L3Index14	L3Index13	L3Index12	L3Index11	L3Index10	L3Index9	L3Index8	22
N/A	Q2	Q1	Q0	N/A	N/A	Parse 1	Parse 0	23
L4Index7	L4Index6	L4Index5	L4Index4	L4Index3	L4Index2	L4Index1	L4Index0	24
N/A	L4Index14	L4Index13	L4Index12	L4Index11	L4Index10	L4Index9	L4Index8	25
Length7	Length6	Length5	Length4	Length3	Length2	Length1	Length0	26
Length15	Length14	Length13	Length12	Length11	Length10	Length9	Length8	27
N/A	N/A	Buffer Nearly Full	Buffer Full	Sport3	Sport2	Sport1	Sport0	28
Hbit	Intercept	Snoop1	Snoop0	PuntCode3	Puntcode2	Puntcode1	Puntcode0	29

Control Bus

The control bus is an 8-bit time-division multiplexed bus that provides control information to Epoch for received packets and control information from Epoch for transmitted packets. The signals carried during each clock cycle of the bus are shown in Table 2 and Table 3.

The receive control information is passed to Epoch along with the first buffer of a packet. The transmit control information is issued by Epoch with every packet buffer.

Receive Control Bus Inputs

MC[15:0]: Multicast (or unicast) Vector. Bit 0 represents Port 0, Bit 1 port 1, etc. This vector informs Epoch to which port(s) packets are routed with a parse option set to 00 or 01.

Q[2:0]: Queue number to queue packet, used only with the Bypass parse option.

Parse[1:0]: Parse Option

Indicates how a packet should be handled by the parser.

00 = Bypass. The Multicast (Unicast) Vector should be set accordingly

01 = Re-inject IP packet. The packet is treated as a Bypass packet with the exception that an L3 lookup is performed to provide an L3 index for the control out bus. Only IP packets may be reinjected.

10 = IPX Packet

11 = IP Packet

Transmit Control Bus Outputs

L3Index[14:0]: RCP address where L3 Match was obtained. May be used to maintain L3 to L2 Address Translation table (for example, ARP Cache).

Buffer Full: Indicates the buffer has reached the programmed full threshold.

Buffer Nearly Full: Indicates the buffer has reached the nearly full threshold. Becomes low again when the lower threshold is uncrossed.

Note: Buffer Full and Buffer Nearly Full apply to the Packet Pointer SRAM also. They are set when the Buffer SDRAM or the Packet Pointer SRAM full condition apply.

L4Index[14:0]: RCP address where L4 Match was obtained.

Length[15:0]: Length of packet transmitted by Epoch.

Parse[1:0]: Parse option used for transmitted packet.

Q[2:0]: Queue from which packet was transmitted.

Sport[3:0]: Source Port of transmitted packet.

Puntcode[3:0]: Reason the packet is diverted to port zero, the Host Process port, instead of being sent to its regular destination port.

0x0: None

0x1: TTL went to 0

0x2: IP Fragment. Packet exceeded port's MTU and is punted for fragmentation.

0x3: DROPSAM. IP multicast packet had a Multicast Source address. Not checked for re-injected packets.

0x4: Not IPv4. IP packet was not version 4.

0x5: IP Options. IP header was greater than 5 words.

0x6: No L3 match

0xD: IPX TC. IPX TC field ≥ 15 .

0xE: IPX Type 20

0xF: IPX Checksum was not 0xFFFF

Snoop[1:0]: Indicates whether and how packet is snooped.

0x0: No snooping occurred.

0x1: Packet destination port snooped.

0x2: Packet source port snooped.

0x3: Packet snooped by flow.

In the event a packet is snooped for multiple reasons, the smallest of all snoop codes is returned.

Intercept: Intercept bit set. Indicates this packet was intercepted.

Hbit: Hbit set. Indicates this packet was diverted to the host. See PEN details for description of these functions.

Buffer Manager (BFM)

The BFM is responsible for linking together SDRAM buffers to hold entire packets. Packet data and BFM control information are kept in separate SDRAM devices: Packet Data in two 1 M x 16 (4 MB) SDRAMs, and Packet Control data in a single 1 M x 16 SDRAM. Devices must be at least 512K x 16 x 2 banks with an access time of at most 12 ns. Larger devices can be used by tying the proper address and/or bank select inputs low. Different refresh cycles, such as 4K cycles in a 64 ms period or 2K cycles in a 32 ms period, are supported since one refresh is performed every 15.625 ms interval.

There is a one-to-one correspondence between packet data buffers and packet control buffers. Data buffers contain only data. Control buffers contain linked-list, multicast and transmit control bus information. There is enough memory for up to 64K 64-byte packets to reside in the buffer memory.

The processor has access to both data and control SDRAM for both diagnostic and initialization purposes. Memory is accessed through index registers and data written or read through data registers. Prior to any memory tests and initialization, the processor must program the refresh interval, start refresh running, and run precharge all

operations on the Buffer and Buffer Control SDRAMs.

At initialization the processor must walk through the buffer control memory and link the 16 bit `bfm_link_to_next_buffer` fields together to create the free buffer list. For example, `bfm_link_to_next_buffer` in buffer pointer 0x0000 points to Buffer pointer 0x0001, which in turn points to 0x0002 etc. The top of the free buffer list is buffer number 0x0000, and the bottom is buffer number 0xFFFF. Each buffer pointer is associated by address to each buffer. The `bfm_link_to_next_buffer` field for each buffer pointer resides at 0x10 intervals starting at address 0x00000 and ending at address 0xFFFFF0 in the buffer control SDRAM.

During operation, buffers are popped from the top of the free buffer list as frames are received. When frames are transmitted, the buffers are returned.

Once Epoch is running, the processor should not access BFM memories. Any data for the host is passed to the host port and the BFM is self-managing.

The BFM is responsible for refresh of the SDRAM. The refresh interval is programmable.

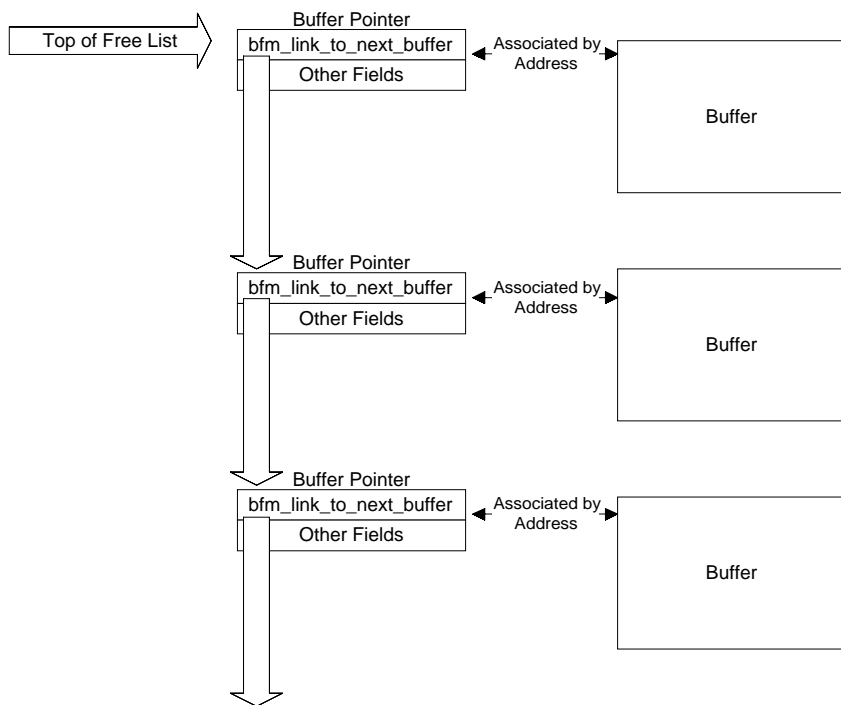


Figure 4: Free Buffer Pointer and Buffer Linked List

Table 4: BFM Memory

Function	Memory Type	Total Size	Component Size	Number	Suitable Parts
Buffers	SDRAM	1M x 32	1M x 16	2	Samsung KM416S1120DT-G/F Micron MT48LC1M16A1or equivalent, 12ns or faster.
Buffer Pointers	SDRAM	1M x 16	1M x 16	1	

Table 5: BFM Processor Registers

Address[8:2] Aligned [8:0]	Register Name	Access	Bits	Function
0x000	PsdramWrAddr	R/W	[19:0]	Processor buffer control write address
0x004	PsdramRdAddr	R/W	[19:0]	Processor buffer control read address. Access to this register starts the cycle.
0x008	PsdramWrData	R/W	[15:0]	Processor buffer control write data. Access to this register starts the cycle.
0x00c	PsdramRdData	R	[15:0]	Processor buffer control read data
0x010	BsdramWrAddr	R/W	[19:0]	Processor buffer write address
0x014	BsdramRdAddr	R/W	[19:0]	Processor buffer read address. Access to this register starts the cycle.
0x018	BsdramWrData	R/W	[31:0]	Processor buffer write data. Access to this register starts the cycle.
0x01c	BsdramRdData	R	[31:0]	Processor buffer read data
0x020	PortReady	R	[0]	Ready bit. Indicates when the BFM registers are ready. Note: For information only. It is not necessary to poll this bit as register accesses are controlled by the PREADY signal on the processor port.
0x024	PsdramMrsInitReg	R/W	[11:0]	Buffer Control SDRAM mode register settings Resets to 0x020 Set to 0x020 during initialization Set to 0x023 during operation
0x028	BsdramMrsInitReg	R/W	[11:0]	Buffer SDRAM mode register setting Resets to 0x020 Set to 0x020 during initialization Set to 0x027 during operation
0x02c	BufFullThreshReg	R/W	[15:0]	BUFFERFULL flag assertion threshold Resets to 0xFFFF De-asserts when not this value
0x030	BufUsedMaxThreshReg	R/W	[15:0]	BUFFERNEARLYFULL flag assertion threshold Resets to 0xEB85 (92% full)
0x034	BufUsedMinThreshReg	R/W	[15:0]	BUFFERNEARLYFULL flag de-assertion threshold Resets to 0xE666 (90% full)
0x038	RfshMaxCntReg	R/W	[15:0]	Refresh period. Number of CLK periods between Buffer and Buffer control Refresh cycles Resets to 0x380 13.58 uS with 66MHz CLK.
0x03c	SyncSuspendb	R/W	[0]	0 = Suspend SYNC pulse 1 = Enable SYNC pulse Resets to 0
0x040	RfshSuspendb	R/W	[0]	0 = Suspend Refresh 1 = Enable Refresh Resets to 0

Table 5: BFM Processor Registers (continued)

Address[8:2] Aligned [8:0]	Register Name	Access	Bits	Function
0x044	BsdramPcAllStart	R/W	[0]	1 = Initiate Precharge All operation on Buffer SDRAM
0x048	PsdramPcAllStart	R/W	[0]	1 = Initiate Precharge All operation on Buffer Control SDRAM
0x04c	BsdramRefreshStart	R/W	[0]	1 = Start Refresh to Buffer Memory
0x050	PsdramRefreshStart	R/W	[0]	1 = Start Refresh to Buffer Control Memory
0x054	ProgSyncReg	R/W	[5:0]	Number of clocks the SYNC pulse occurs before the first word of TDM RX data. This register resets to 3, placing the SYNC pulse as shown in Figure 10. Figure 11 shows some examples of which register values affect the SYNC pulse.
0x058–0x05c	Reserved	N/A	N/A	N/A

Note: Registers are all 32-bit aligned. 32-bit accesses only are supported.

Packet Manager (PKM)

The PKM inserts and extracts packet pointers from queues. During packet reception, the PKM receives a queue number and packet pointer from the MCM, and queues the packet pointer at the tail of the specified queue. During packet transmission, the PKM receives a queue number from the QSM, and it extracts the packet pointer at the head of the specified queue and passes it to the BFM. The queue information is stored in the queue pointer SRAM internal to Epoch. The packet pointer information is stored in a 64K x 32b SRAM external to Epoch.

The PKM memories are accessible by the processor for testing and initialization. Once Epoch is running, the processor should not access PKM memory.

There are 64K packet pointers available, hence up to 64K packets may reside in the buffer memory.

At initialization the processor must walk through the Packet Pointer memory and link the 16 bit `pkm_next_packet` fields together. This is the upper 16 bit field of each 32 bit packet pointer entry in the packet pointer SRAM. The top of the free packet pointer list is address 0xFFFF, and the bottom is pointer 0x0100. The `pkm_next_packet` field in packet pointer 0xFFFF should point to packet pointer 0xFFFFE which in turn should point to 0xFFFFD etc. The bottom 0x0FFF packet pointers are pre-allocated (see queue pointers below). The `pkm_head_of_packet` field does not require initialization.

As packets are received packet pointers are popped from the top of the packet pointer free queue. As packets are transmitted, they are returned.

There are 128 queue pointers in the internal 128 x 48 bit queue pointer memory. Address 0x000 is for port 0 queue 0, address 0x001, port 0 queue 1, address 0x008 for port 1 queue 0, etc.

At initialization the processor must set the `pkm_number_of_packets` field in each queue pointer to 0x0000. The `pkm_tail_of_queue` and `pkm_head_of_queue` fields for queue pointer 0x000 must both be set to 0x0000, for queue pointer 0x001 they must be set to 0x0001 etc. The head and tail of each queue are pointing to the same packet pointer.

The processor must not access the queue pointer memory while Epoch is running.

Each queue pointer is now initialized and pointing to the bottom 128 packet pointers, which is the packet pointer that will be used for the first packet in each queue. As packets are added to each queue more packet pointers are linked. Each packet pointer points to the first buffer of a packet and the buffers then link together to form packets. Figure 5 shows this relationship.

Notice that multicast packets are handled by multiplying the number of queue pointers and packet pointers, which point to the same packet buffers; thus, multicast packets are not duplicated in buffer memory.

Refer to Table 6 and Table 7 for the packet and queue pointer formats.

Table 6: Packet Pointer Format

pkm_next_packet 16 bits	pkm_head_of_packet 16 bits
----------------------------	-------------------------------

Table 7: Queue Pointer Format

pkm_number_of_packets 16 bits	pkm_tail_of_queue 16 bits	pkm_head_of_queue 16 bits
----------------------------------	------------------------------	------------------------------

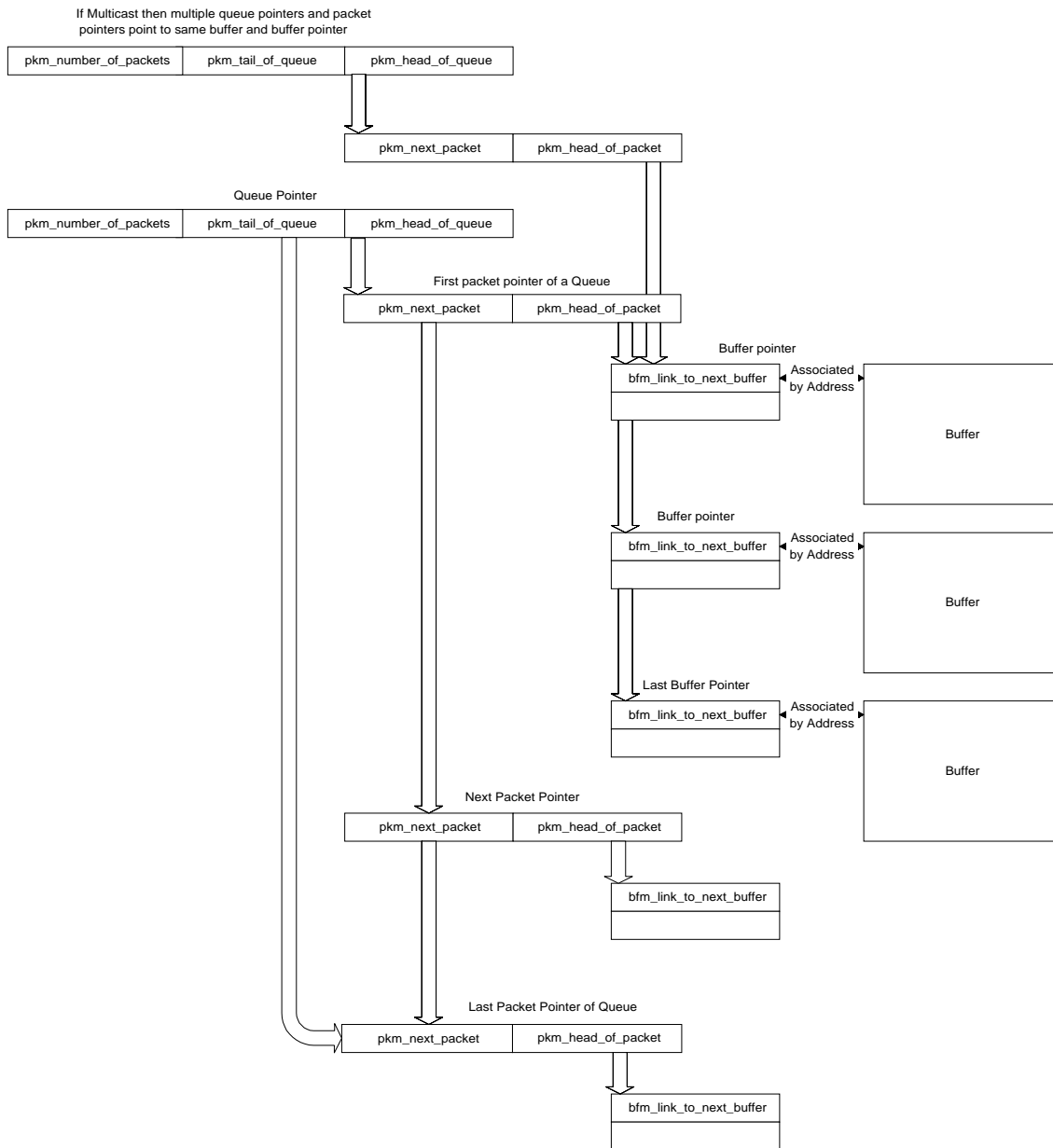


Figure 5: PKM Memory

Table 8: PKM Memory

Function	Memory Type	Total Size	Component Size	Number	Suitable Parts
Packet Pointers	SRAM	64K x 32	64K x 16	2	Samsung KM616V1002 Mitsubishi M5M564R16 or equivalent, 15 ns or faster
Queue Pointers	SRAM	128 x 48	128 x 48	1	Internal

Table 9: PKM Processor Registers

Address[8:2] Aligned [8:0]	Register Name	Access	Bits	Function
0x140	QueueSRAMWriteAddress	R/W	[6:0]	Address of Queue pointer SRAM to perform write
0x144	QueueSRAMWriteDataLow	R/W	[31:0]	LOW word of write data
0x148	QueueSRAMWriteDataHigh	R/W	[15:0]	HIGH word of write data. Access to this register starts cycle.
0x14c	QueueSRAMReadAddress	R/W	[6:0]	Address of Queue pointer SRAM to perform read. Access to this register starts cycle.
0x150	QueueSRAMReadDataLow	R	[31:0]	LOW word of read data
0x154	QueueSRAMReadDataHigh	R	[15:0]	HIGH word of read data
0x158	PacketSRAMWriteAddress	R/W	[15:0]	Address of Packet pointer SRAM to perform write
0x15c	PacketSRAMWriteData	R/W	[31:0]	Packet pointer Write data. Access to this register starts cycle.
0x160	PacketSRAMReadAddress	R/W	[15:0]	Address of Packet pointer SRAM to perform read. Access to this register starts cycle.
0x164	PacketSRAMReadData	R	[31:0]	Packet pointer read data
0x168	ProcessorPortEnable	R/W	[0]	1 = Enable processor access to BFM and PKM memories 0 = Disable Resets to 0 Must be set to 1 for processor access to BFM and PKM memories.
0x16c	ProcessorPortReady	R	[0]	Indicates the processor interface to the PKM is ready. Note: For info only. The PREADY signal controls access. The processor does not need to poll this bit.
0x170	NextFreePointer	R	[15:0]	Info. Next Free Packet pointer address
0x174	NextPacket	R	[15:0]	Info. Next packet to operate on.
0x178	HeadofPacket	R	[15:0]	Info. Current pkm_head_of_packet field
0x17c	HeadofQueue	R	[15:0]	Info. Current pkm_head_of_queue field
0x180	TailofQueue	R	[15:0]	Info. Current pkm_tail_of_queue field.
0x184	NumberofPacket	R	[15:0]	Info. Current pkm_number_of_packets field
0x188	TXPacket	R	[15:0]	1 = A packet is being transmitted b0 = port0, bit1 = port1, etc.
0x18c	FullThreshold	R/W	[15:0]	Used packet pointer Threshold at which BUFFERFULL should be asserted
0x190	UpperThreshold	R/W	[15:0]	Used packet pointer threshold at which BUFFERNEARLYFULL should be asserted.
0x194	LowerThreshold	R/W	[15:0]	Used packet pointer threshold at which BUFFERNEARLYFULL should be de-asserted.

Note: Registers are all 32-bit aligned. 32-bit accesses only are supported.

Multicast Manager (MCM)

The MCM takes a queue number and a port bitmap from the PEN and the first buffer of a packet pointer from the BFM and produces N queue/packet pointer pairs that each point to the same packet in the BFM. Each of the 16 ports has the same eight queue numbers. The packet is queued in the same queue number for each of the output ports selected by the output port bitmap. Note that the MCM treats unicast packets as a special case of multicast, in which the number of destination ports is one. Thus, only one copy of the multicast packet is maintained in the BFM.

The MCM contains a FIFO of packet pointers, queues, and output port bitmaps for multicast packets. The MCM generates a queue/packet pointer pair for each output port in the output port bitmap, which is passed to the PKM. The PKM assigns queue pointers and packet pointers for each multicast packet.

If the MCM FIFO overflows, an abort signal is sent to the BFM to drop the packet. The multicast FIFO holds 256 entries.

If the PKM packet pointer SRAM fills while multicast packet pointers are being expanded, the MCM waits for some packet pointers to be freed as packets are transmitted to complete the multicast expansion.

No MCM processor access is required.

Queue Scheduler Module (QSM)

The QSM determines the order in which flows are serviced at each output queue. The queue scheduler is a work-conserving scheduler that guarantees that no packet slot is idle if the Epoch has a packet to send.

The QSM supplies the next queue to be transmitted on a given port when the queue manager requests this information. The QSM must know what queues have information in them for each port. To acquire this information, every time a packet is received, the PKM informs the QSM of the port and queue into which the received packet will go.

There are eight prioritized queues for each port. The queues are numbered q[0:7], with 0 designating the highest priority queue and 7 the lowest priority queue. When the PKM desires the next queue to be serviced for a given port, the QSM algorithmically computes the next queue and informs the PKM which queue to service for that port.

Algorithm 1. Weighted Round Robin Mode

A binary weighted priority algorithm is used which increases the priority of each ascending queue by a factor of 2. Normalizing the priority of the lowest priority queue, q[7], to 1, the relative priority of each queue is given below:

Table 10: Queue Relative Priority

Queue	Weighting
0	128
1	64
2	32
3	16
4	8
5	4
6	2
7	1

If the queue to be serviced is empty, then the next lowest queue is checked to see if it is empty. If all lower priority queues are empty, the search wraps to Queue 0. This continues until a queue which is not empty is found. This circular fashion of finding the next available queue prevents starvation of the lower priority queues while still guaranteeing a specified bandwidth on the higher priority queues.

Algorithm 2. Priority Mode

Queue 0 always gets priority over Queue 1 over Queue 2 over Queue 3 etc. All of Queue 0 is emptied before Queue 1 is serviced, then all of Queue 1 is emptied before Queue 2 etc. If another packet arrives in a higher priority queue it will always be transmitted next after the current packet has finished transmitting.

Table 11: PKM Processor Registers

Address[8:2] Aligned [8:0]	Register Name	Access	Bits	Function
0x198	ProgramAlgorithmSelect	R/W		0 = Weighted Round Robin Mode 1 = Priority Mode
			[31:16]	Reserved. Set to 0x0000.
			[15]	Port 15
			[14]	Port 14
			[13]	Port 13
			[12]	Port 12
			[11]	Port 11
			[10]	Port 10
			[9]	Port 9
			[8]	Port 8
			[7]	Port 7
			[6]	Port 6
			[5]	Port 5
			[4]	Port 4
			[3]	Port 3
[2]	Port 2			
[1]	Port 1			
[0]	Port 0			

Note: Registers are all 32 bit-aligned. 32-bit accesses only are supported.

L3 Data Parse Engine (PEN)

The PEN makes the appropriate changes to the header of each IP or IPX packet that passes through the Epoch, such as decrementing the TTL field and re-computing the header checksum. If the packet is not an IP or IPX packet, then the header is passed through unchanged. The IP or IPX header is examined, and if the specific header type is not supported by the Epoch, then the PEN informs the PKM to queue the packet to the processor. The PEN also checks the TTL of the IP header and compares it against the per port TTL threshold for multicast packets to determine which multicast packets should be discarded.

L3 Address Parse Engine (L3PEN)

The L3PEN parses headers for Layer 3 protocols IPv4, IP Multicast and IPX. The PEN includes an interface to the RCP database via the CRI. The packet address is compared with the RCP database. The RCP database returns the index of the first matching entry (if any). The index is used to find a 16 bit output port bitmap contained in external SRAM via the CRI database.

IP Packets

Before processing the header, the PEN must assure that the packet is of the correct type: the IP version number is four, the IP header length indicates that no IP options are in use, the IP flags field indicates that no additional fragments exist, and the fragment offset is zero. This information is given by the PEN. The destination port is then determined from the L3 RCP match and associated data.

IP Multicast Packets

If a multicast address is matched in the RCP, the ports belonging to a multicast group are obtained from the associated data RAM and the packet forwarded to all the ports in the group.

The PEN implements TTL thresholding for multicast packets. For each output port of a multicast, the packet TTL is compared with the TTL threshold. If the TTL is too small, the packet is not multicast to that output port.

IPX Packets

The Transport Control field is checked to be less than 15. If the source or destination network number is 0 they are replaced with the source port network number. The destination port is derived from the destination network number. The Transport Control field is incremented.

All Forwarded Packets

The PEN checks the packet size against the Maximum Transmission Unit (MTU) for each port. If a packet is too large for a given port, it is sent to the host to be fragmented if required.

The PEN contains a mechanism to force a packet to be multicast for network management purposes (For example, “snooping”). Bitmaps are used to determine the sources and/or destinations for which packets are snooped, and snoop port registers determine to which port the duplicates of these packets are sent. There is also an intercept mechanism such that intercepted packets do not continue towards their usual destination port and are redirected to a different port.

L4 PEN

The L4PEN examines the packet header and returns the queue number to which the packet is to be queued. The L4 RCP is searched for a matching flow entry via the CRI. If a matching entry exists, the index returned by the RCP is used to find a 3-bit queue number in a separate block of SRAM via the CRI.

Traffic may be classified using the proposed IETF Differentiated Services. Per Flow Filtering also is provided.

Traffic Classification BAC (Behavior Aggregate Classification) Mode

The DS field is used to address a RAM, the BAC Table, of 256 entries of 16 bits which returns a flow handle for the destination port. Flow handles are described in the Per Flow Filtering section. The Mapping Table is processor maintained.

Microflow Mode

The parse engine extracts the IP and L4 header, TCP or UDP port number, source and destination IP address fields and incoming L2 interface number. The RCP is searched (via the CRI). After a match is found, the index of the matching entry is used to find the flow handle of the flow in the L3/L4 SRAM (via the CRI). The packet optionally can be diverted to the Host port for flow recognition or authentication. The processor updates the L4 RCP with the recognized flow. Each time an entry is found for a flow,

the activity flag (T bit) associated with the flow is updated. The processor uses the activity flag to determine which flows are old and can be purged from the RCP.

If no L4 flow is found in the RCP, both the source and destination L4 (TCP/UDP) port numbers are used to index the default queue. Of the two default queue values found, the higher priority queue is chosen. The default queue also can indicate that the packet should be redirected to the Host for flow recognition.

DS Re-marking Differentiated Services Packet Re-marking by BAC

If BAC mode is enabled, the PEN uses the packet’s original DS field to index into the BAC Table. The results of the indexed lookup are a replacement DS field and a flow handle (Refer to Figure 7). The flow handle contains a replacement bit (R bit) which indicates whether the DS field in the outgoing packet should be replaced. It contains a replacement width bit (T8 bit) which controls whether to replace either all eight bits or just six bits of the DS field of the outgoing packet.

Differentiated Services Packet Re-marking by Microflow

If the IP packet was received on a port in microflow mode, the PEN utilizes the IP Layer 4 lookup sequence to obtain a flow handle. The flow handle will be obtained from either a Layer 4 CAM match or a default flow match. The PEN obtains three fields from the matching flow handle for use in DS field marking: the replacement DS value, the T8 bit, and the R bit. These values are used in the same manner as in BAC mode. If the R bit is not set, the packet’s DS field is left unmodified. If the R bit is set and the T8 bit also is set, all eight bits of the packet DS field are changed to the replacement DS field value. If the R bit is set while the T8 bit is not set, only the most significant six bits in the packet DS field are replaced.

Per Flow Filtering

Each flow handle entry contains a D bit which indicates that if a matched or default flow is found the packet should be dropped. There is also a H bit which indicates the packets in a flow should be diverted to the host.

Table 12: PEN Processor Registers

Address[8:2] Aligned [8:0]	Register Name	Access	Bits	Function
0x060	TTLThreshold-Port0	R/W	[7:0]	TTL Threshold per port Multicast TTL Thresholding.
	TTLThreshold-Port1	R/W	[15:8]	
	TTLThreshold-Port2	R/W	[23:16]	
	TTLThreshold-Port3	R/W	[31:24]	
0x064	TTLThreshold-Port4	R/W	[7:0]	
	TTLThreshold-Port5	R/W	[15:8]	
	TTLThreshold-Port6	R/W	[23:16]	
	TTLThreshold-Port7	R/W	[31:24]	
0x068	TTLThreshold-Port8	R/W	[7:0]	
	TTLThreshold-Port9	R/W	[15:8]	
	TTLThreshold-Port10	R/W	[23:16]	
	TTLThreshold-Port11	R/W	[31:24]	
0x06c	TTLThreshold-Port12	R/W	[7:0]	
	TTLThreshold-Port13	R/W	[15:8]	
	TTLThreshold-Port14	R/W	[23:16]	
	TTLThreshold-Port15	R/W	[31:24]	
0x070	MTU-Port0	R/W	[15:0]	MTU per port for Punting packets to host.
	MTU-Port1	R/W	[31:16]	
0x074	MTU-Port2	R/W	[15:0]	
	MTU-Port3	R/W	[31:16]	
0x078	MTU-Port4	R/W	[15:0]	
	MTU-Port5	R/W	[31:16]	
0x07c	MTU-Port6	R/W	[15:0]	
	MTU-Port7	R/W	[31:16]	
0x080	MTU-Port8	R/W	[15:0]	
	MTU-Port9	R/W	[31:16]	
0x084	MTU-Port10	R/W	[15:0]	
	MTU-Port11	R/W	[31:16]	
0x088	MTU-Port12	R/W	[15:0]	
	MTU-Port13	R/W	[31:16]	
0x08c	MTU-Port14	R/W	[15:0]	
	MTU-Port15	R/W	[31:16]	
0x090	IPXattach-Port0	R/W	[31:0]	IPX Network number per port for IPX forwarding.
0x094	IPXattach-Port1	R/W	[31:0]	
0x098	IPXattach-Port2	R/W	[31:0]	
0x09c	IPXattach-Port3	R/W	[31:0]	
0x0a0	IPXattach-Port4	R/W	[31:0]	
0x0a4	IPXattach-Port5	R/W	[31:0]	
0x0a8	IPXattach-Port6	R/W	[31:0]	
0x0ac	IPXattach-Port7	R/W	[31:0]	
0x0b0	IPXattach-Port8	R/W	[31:0]	
0x0b4	IPXattach-Port9	R/W	[31:0]	
0x0b8	IPXattach-Port10	R/W	[31:0]	
0x0bc	IPXattach-Port11	R/W	[31:0]	
0x0c0	IPXattach-Port12	R/W	[31:0]	
0x0c4	IPXattach-Port13	R/W	[31:0]	
0x0c8	IPXattach-Port14	R/W	[31:0]	
0x0cc	IPXattach-Port15	R/W	[31:0]	

Table 12: PEN Processor Registers (continued)

Address[8:2] Aligned [8:0]	Register Name	Access	Bits	Function	
0x0d0	DefaultIPXFlowHandle	R/W	[15:0]	Default Flow handle for IPX packets.	
			[15:5]	Not implemented	
			[4]	Drop	
			[3]	Divert to Host	
				[2:0]	Queue Number
	DefaultNotVersion4FlowHandle	R/W	[31:16]	Default flow handle for Not IPv4 Packets.	
			[31:21]	Not implemented	
			[20]	Drop	
[19]			Not implemented		
			[18:16]	Queue Number	
0x0d4	DefaultTTLFlowHandle	R/W	[15:0]	Default flow handle for TTL=0 Packets.	
			[15:5]	Not implemented	
			[4]	Drop	
			[3]	Not Implemented	
				[2:0]	Queue Number
	DefaultIPOptionsFlowHandle	R/W	[31:16]	Default flow handle for IP Options packets	
			[31:21]	Not implemented	
			[20]	Drop	
[19]			Not Implemented		
			[18:16]	Queue Number	
0x0d8	DefaultFragPackFlowHandle	R/W	[15:0]	Default Flow Handle for Fragmented packets	
			[15:8]	Replacement DS field	
			[7]	T8 (Replace 8 bit or 6 bit) 1 = Replace 8 bits of DS 0 = Replace top 6 bits of DS	
			[6]	R (Replace bit) 1 = Replace DS field 0 = Do not replace DS field	
			[5]	Not Implemented	
			[4]	Drop	
			[3]	Divert to host	
			[2:0]	Queue Number	
	DefaultUnknownL4ProtoFlowHandle	R/W	[31:16]	Default Flow Handle for Unknown L4 Protocol packets	
			[31:24]	Replacement DS field	
			[23]	T8 (Replace 8 bit or 6 bit) 1 = Replace 8 bits of DS 0 = Replace top 6 bits of DS	
			[22]	R (Replace bit) 1 = Replace DS field 0 = Do not replace DS field	
			[21]	Not Implemented	
			[20]	Drop	
			[19]	Divert to Host	
			[18:16]	Queue Number	
0x0dc	DSMAP	W	[15:0]	Destination Snoop map. Each bit represents the destination port to snoop, for example bit 0 = port 0.	
	DSIsel	W	[19:16]	Destination Snoop Interface select for augmenting.	
0x0dc	SSMAP	R	[15:0]	Source Snoop map. Each bit represents the source port to snoop, for example bit 0 = port 0.	
	SSIsel	R	[19:16]	Source Snoop Interface select for augmenting.	

Table 12: PEN Processor Registers (continued)

Address[8:2] Aligned [8:0]	Register Name	Access	Bits	Function
0x0e0	SSMAP	W	[15:0]	Source Snoop map. Each bit represents the source port to snoop, for example bit 0 = port 0.
	SSIsel	W	[19:16]	Source Snoop Interface select for augmenting.
0x0e0	DSMAP	R	[15:0]	Destination Snoop map. Each bit represents the destination port to snoop, for example bit 0 = port 0.
	DSIsel	R	[19:16]	Destination Snoop Interface select for augmenting.
0x0e4	SIMAP	R/W	[15:0]	Source Intercept map. Each bit represents the source port to intercept, for example bit 0 = port 0.
	SIIsel	R/W	[19:16]	Source Intercept Interface select for diverting.
0x0e8	DoNotProcess	R/W	[15:0]	Disable TTL decrement per port. Bit 0 is port 0 etc. 0 = Enable
	NoL3MatchHostForward	R/W	[27]	Forward Packets to host port if no L3 match is obtained. 1 = Forward 0 = Drop
	HostForwardPortSelect	R/W	[31:28]	Host Port number for all packets forwarded to the host when the H bit is set in their flow handle and packet snooping by flow is off for that destination port.
0x0ec	EnableDSclassifier	R/W	[15:0]	Enable DS classification. Each bit represents the port to enable. Bit 0 = port 0 etc. If DS classification is off, flows are classified by microflow but no DS re-marking can occur whether the R bit is set in the flow handle or not.
	BACnMicroflow	R/W	[31:16]	DS mode. DS classification must be enabled. Bit 16 represents port 0 etc. 0 = Microflow mode 1 = BAC Mode
0x0f0	L3DefaultIndexPtr	R/W	[14:0]	When no L3 match is obtained, the L3 Index pointer is set to this value.
	L4DefaultIndexPtr	R/W	[30:16]	When no L4 match is obtained the L4 Index pointer is set to this value.
0x0f4	FSMAP	R/W	[15:0]	Snoop by Flow Map. Each bit represents the destination port to snoop by flow. b0 = port 0.
	FSIsel	R/W	[19:16]	Snoop by Flow Interface select for augmenting.

Notes:

1. Registers are all 32-bit aligned. 32-bit accesses only are supported.
2. The DSMAP, DSIsel, SSMAP, and SSIsel registers have a write address that is different from the read address.

RCP/RAM Interface (CRI)

The CRI contains the state machine for accessing the L3/L4 RCP database and also arbitrates among the sources of RCP operations (PEN and PIM). Data is written to or read from CRI registers and when the RCP op code register is written, this initiates a RCP operation.

The RCP contains the IPv4 routing table, including IP Multicasts, the IPX routing table and Matched flows. Up to four MUSIC MUAC 4K or 8K RCPs may be connected to build routing tables and matched flows up to 32K in depth.

RCP Entry Structure

Figure 6 illustrates the structure of the five types of entries in the RCP. The types are:

- Unicast IP
- Multicast IP
- Matched flow L4 Parent
- Matched flow L4 Child
- IPX Network address

Entries may be stored anywhere in the RCP but there are restrictions as to the ordering of IP unicast ternary structures to ensure correct operation of hierarchical lookups. See Application Note AN-N25 for details.

Unicast IP is entered in the ternary fashion shown to implement hierarchical address searches, allowing longestable match searches. Refer to MUSIC Application Note AN-N25 (Fast IPv4 and IPv4CIDR Address Translation and Filtering Using the MUAC Routing CoProcessor (RCP)) for details on how this works.

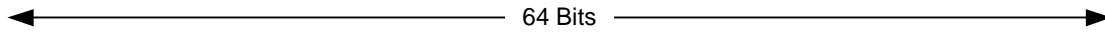
IP Multicasts are entered as a source and destination IP binary address.

IP matched flows are entered as a parent source/destination binary IP pair with a child L4 entry for each TCP/UDP source-destination port pair and source port.

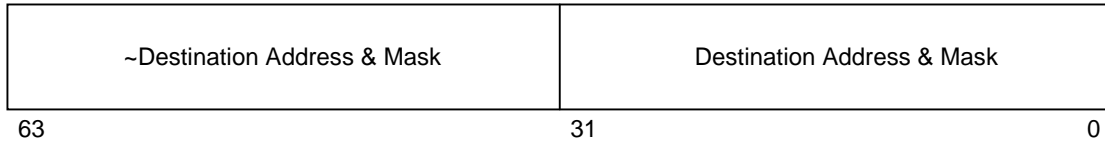
- PP[1:0] = Physical Port bits 1:0. The flow source port.
- PP[3:2] = Physical Port bits 3:2.
- Pind[7:0] = Parent index 7:0. The RCP Index of the Parent entry for the flow
- Pind[15:0] = Parent Index 15:8. Note bit 15 is always 0.

IPX network addresses are entered as a binary entry as shown.

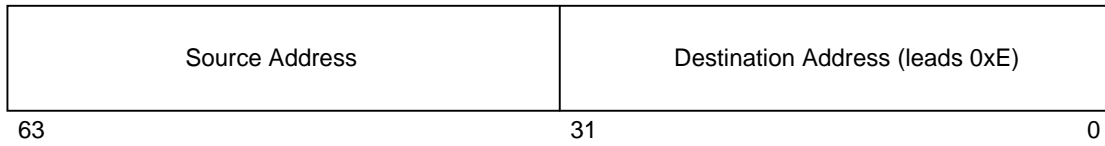
Refer to AN-N27 (Using MUSIC Devices and RPCs for IP Flow Recognition) for more details on the technique.



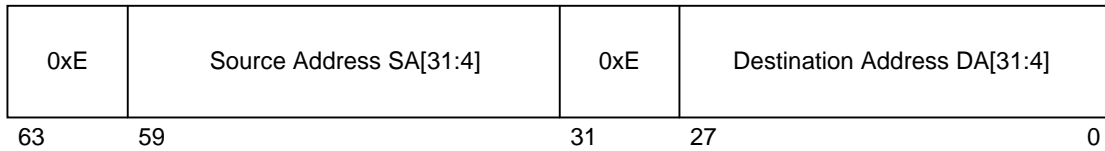
(1) IP Unicast ternary structure



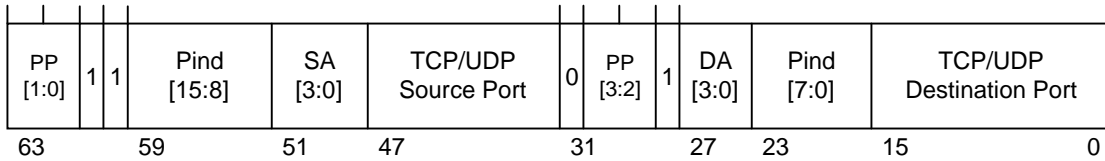
(2) IP Multicast binary structure



(3) IP Layer-4 parent binary structure



(4) IP Layer-4 child binary structure



(5) IPX Network Address binary structure

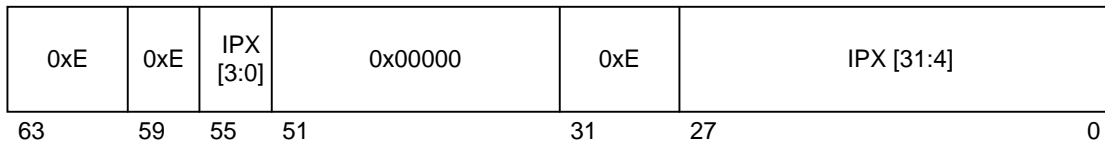


Figure 6: Binary Structure

RCP Mask Registers

Each entry type (IPX, Layer 3 Multicast and Layer 4) is associated with a dedicated MUAC RCP Mask register so that user-specified packet-header bit fields can be used in lookup operations. The same mask is used for all entries of a given type and for all ports. Mask registers are assigned to entry types as follows:

Table 13: Mask Register Assignments

Type	RCP Mask Register Number
IPX	4
IPv4 Layer 3 MultiCast	5
IPv4 Layer 4 Parent	6
IPv4 Layer 4 Child	7

RCP Mask registers are described in the MUAC data sheet and are accessed through Epoch registers RCPRdOp, RCPRdData, RCPWrOp and RCPWrData. By default, no masking is used since the MUAC Mask registers default to all zeros upon reset. Mask registers must be set if their use is desired.

Note: Each IPv4 Layer 3 Unicast entry incorporates its own mask, so no RCP Mask register is used for this type of entry, whereas other entry types, by using RCP Mask registers, are subject to the same mask. For example, all IPX lookups from all ports are masked by mask 4.

SRAM

Figure 7 illustrates CRI SRAM information. The SRAM contains associated data for the L3 information and matched flow information flow handles contained in the RCP as well as the DS BAC Table and flow handles and the default flow and flow handles and the multicast MIAN Table. As entries are made in the RCP the associated data for each entry must be programmed before the validity bit for each RCP entry is enabled.

RCP Associated Data

The bottom 32K words of CRI SRAM contains the associated data for the RCP. The entry types are:

Layer 3. Associated data which provides the port bitmap for the L3 address matched. Bit 0 represents port 0, etc. For multicast entries, multiple bits are set.

Layer 4. Associated data which contains the flow handle for the matched flow RCP data.

BAC and L3 MIAN Tables

The next 32K words contains the BAC Table and L3 MIAN Table (Multicast Interface Authentication Number).

The BAC Table contains flow handles for DS BAC mode packets. The received packets DS field is used to address the BAC Table to return the appropriate flow handle for that DS field. Note the flow handles are split into two eight bit words in the BAC Table. The first eight bits are the bottom half of the handle and the next eight bits the top.

The MIAN Table is composed of four bit fields for authenticating the source port a multicast is accepted on when received. The MIAN is programmed with the authorized source port for the matched multicast.

The top 64K words contain the flow handles for default flows. When the flow is not matched in the RCP, the CRI uses the source and destination TCP or UDP port values as indices into the Port Default Queue table to retrieve two flow handles. It then uses the flow handle with the lowest queue number, indicating the higher priority.

Flow Handles

The flow handle is the same format for entries in the BAC Table, the Layer 4 Associated data for matched flows for the default queue table and the default flow registers.

Bits 15:8 are a replacement DS field for flows where the DS field in packets is to be re-marked.

Bit 7 is the T8 bit. If set, all eight bits if a DS field are replaced. If clear, only the top six bits are replaced.

Bit 6 is the DS replace bit. If set the DS for this flows packets is to be replaced.

Bit 5 is the Touch bit. As entries are matched this bit is set. The processor may periodically read the T bit. If it finds the T bit is not set, the flow may be stale. If the processor does not want to delete the entry it should clear this bit for next time.

Note: This bit is only relevant for matched L4 child handles. In all other flow handles this bit is ignored.

Bits 4:0 are collectively known as the queue handle.

Bit 4 is the D bit. When set, packets in this flow are dropped.

Bit 3 is the Host bit. The function of this bit depends on the setting of the Snoop by Flow registers, FSMAP and FSIsel (Refer to Table 14).

Bits 2:0 are the queue number. Packets in this flow are queued for transmission in the specified queue. See Queue Scheduler Module (QSM) section for details on queuing algorithms.

Table 14: Host Bit Function

Flow Handle H Bit	Snoop by Flow Bit for Destination Port (FSMAP Register)	Function
0	X	Packet forwarded according to Layer 3 portmap
1	0	Packet diverted to port specified by HostForwardPortSelect register
1	1	Packet forwarded according to Layer 3 portmap and copy sent to snoop port specified by FSIsel

Notes: If a multicast packet is diverted to the host, a copy is sent to the host and not to any of the ports specified by the port bitmap. If Snoop by Flow is enabled for a port, then host diversion for that port does not occur.

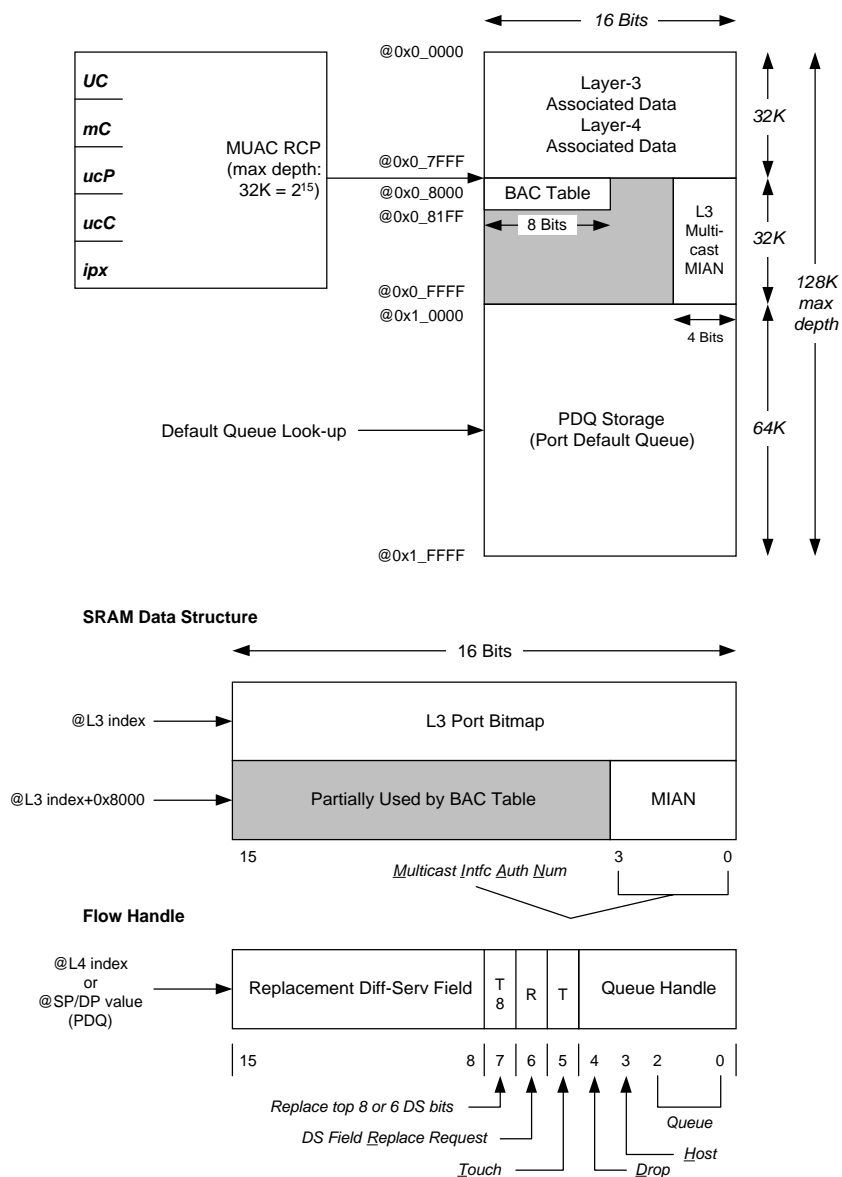


Figure 7: CRI SRAM

Table 15: CRI Processor Registers

Address[8:2] Aligned [8:0]	Processor Register	Access	Bits	Function
0x108	SRAMRdAddr	R/W	[16:0]	CRI SRAM read address, launches the read
0x10c	SRAMRdData	R	[15:0]	CRI SRAM read Data
0x110	SRAMWrData	R/W	[15:0]	CRI SRAM Write Data
0x114	SRAMWrAddr	R/W	[16:0]	CRI SRAM Write Address, launches the write
0x118	RCPRdOp	R/W	[18:0]	CRI RCP Read Operand, launches the read
			[12:0]	RCP Op-Code (See MUAC RCP Data Sheet)
			[13]	RCP Chip 0 Select. RCPCS20b pin 0 = Chip Selected
			[14]	RCP Chip 1 Select. RCPCS21b pin
			[15]	RCP Chip 2 Select. RCPCS22b pin
			[16]	RCP Chip 3 Select. RCPCS23b pin
			[17]	RCP Data Segment Control. RCPDSC pin
[18]	RCP Address Valid. RCPAVb pin			
0x11c	RCPRdData	R	[31:0]	CRI RCP Read Data
0x120	RCPWrData	R/W	[31:0]	CRI RCP Write Data
0x124	RCPWrOp	R/W	[19:0]	CRI RCP Write Operand, launches the write
			[12:0]	RCP Op-Code
			[13]	RCP Chip 0 Select. RCPCS20b pin 0 = Chip Selected
			[14]	RCP Chip 1 Select. RCPCS21b pin
			[15]	RCP Chip 2 Select. RCPCS22b pin
			[16]	RCP Chip 3 Select. RCPCS23b pin
			[17]	RCP Data Segment Control. RCPDSC pin
[18]	RCP Address Valid. RCPAVb pin			
[19]	RCP Validity bit. RCPVBb pin			
0x128	RCPStatus	R/W	[0]	RCP Validity bit. Reverse sense RCPVBb pin. 1 = Valid RCP data 0 = Invalid RCP data. Valid after write to RCPRdOp
			[1]	RCP Full Flag. Reverse sense RCPFFb pin. 1 = RCP full 0 = RCP not full
0x12c	RCPSearchData	R/W	[31:0]	CRI RCP Search Data

Table 15: CRI Processor Registers (continued)

Address[8:2] Aligned [8:0]	Processor Register	Access	Bits	Function
0x130	RCPSearchOp	R/W	[18:0]	CRI RCP Search Operand, launches the search.
			[12:0]	RCP Op-Code
			[13]	RCP Chip 0 Select. RCPCS20b pin 0 = Chip Selected
			[14]	RCP Chip 1 Select. RCPCS21b pin
			[15]	RCP Chip 2 Select. RCPCS22b pin
			[16]	RCP Chip 3 Select. RCPCS23b pin
			[17]	RCP Data Segment Control. RCPDSC pin
			[18]	Fast Write Compare 0 = Search using RCPSearchData and RCPSearchOp registers 1 = Search using RCPWrData, RCPSearchData, and RCPSearchOP registers
0x134	RCPSearchResult	R	[16:0]	CRI RCP Search Index, Match Flag and Multiple Match Flag
			[14:0]	CRI RCP Search Index returned from search operation. RCPADDR[14:0] pins.
			[15]	CRI RCP Search Match Flag. Reverse sense RCPMFb pin. 1 = Match 0 = No Match
			[16]	CRI RCP Search Multiple Match Flag. Reverse sense RCPMMb pin. 1 = Multiple Match 0 = One or no Match
0x138	MIAN_CREG	R/W	[15:0]	1 = Enable MIAN. Each bit represents a source port, bit 0 is port 0 etc.
0x13c	Reserved	N/A	N/A	N/A

Note: Registers are all 32-bit aligned. 32-bit accesses only are supported.

Table 16: CRI Memory

Function	Memory Type	Total Size	Component Size	Number	Suitable Parts
L3 Output Port Bitmap, L4 Matched flow handles, BAC Table, MIAN Table, Default flow handles	SRAM	128K x 16	128K x 8	2	Samsung - KM68V1002B-15 Mosel Vitelic-V61C3181024-15 IDT - IDT71V124-15 or equivalent, 15 ns or faster
			64K x 16	2	
L3, L4 search table	MUSIC RCP	4K-32K x 64	4K x 64K or 8K x 64K	1-4	MUAC4K64-90TDC or faster MUAC8K64-90TDC or faster

Table 17: PIM Processor Registers

Address[8:2] Aligned [8:0]	Processor Register	Access	Bits	Function
0x19c	Reset/Status	R/W	[5:0]	Software Reset and PLL control
		R/W	[0]	Software Reset Write a one to reset the part. Read zero
		R	[1]	PLL Enabled 0 = Disabled 1 = Enabled
			[2]	PLL Locked 0 = Not locked 1 = Locked
		R/W	[4:3]	PLL Multiplier. PLL must run between 100 and 200 MHz. External clock frequency range is indicated in parenthesis. 00: Multiply by 2 (50 to 66 MHz) 01: Multiply by 4 (25 to 50 MHz) 10: Multiply by 8 (12.5 to 25 MHz) 11: Multiply by 16 (6.25 to 12.5 MHz)
			[5]	Clock on INT. Enables system clock to be output on INT pin for test purposes. 0 = INT normal (Processor Interrupt) 1 = INT is internal clock
0x1a0	Ready Status	R	[6:0]	Subsystems ready status. Information only
			[0]	PEN
			[1]	QSM
			[2]	BFM
			[3]	PKM
			[4]	INT
			[5]	STM
			[6]	CRI
0x1a4	Version/ID	R	[7:0]	Revision
			[23:8]	Model Number: 0xec01

Statistics and Interrupts Module (STM)

Statistics

The statistics module maintains counts of dropped packets. There are ten counters, one holding a total count of packets dropped and nine others, each associated with a specific reason for dropping a packet. In the event a packet is dropped for multiple reasons, only the counter with the highest priority is incremented. The counters and their priorities are shown in Table 19.

All of the counters are 16 bit. When a counter crosses its threshold register an interrupt is generated if it is enabled by the interrupt mask. The register continues to count. When a counter is read by the processor it clears back to 0, or 1 if it is read at the same time as it is incremented. The processor is at liberty to poll these registers instead of being interrupt driven.

Interrupts

The interrupts provided are shown in Table 20.

Exception Processing

There are several situations, in which packets are routed to a different destination port than intended either through punting, diverting or intercepting. Packets can also be replicated to another port in addition to being routed to their intended destination port; this is referred to as snooping.

Punting

A packet is punted when it can not be processed by Epoch, in which case it is forwarded to port zero. A punt code corresponding to the reason for punting is issued on the control bus. Punt codes are listed in the control bus section.

Diverting

Also called Host Forwarding, occurs when the H bit is set in a flow handle and Snoop by Flow is disabled for the packet normal destination port. The packet is diverted to the port indicated by the HostForwardPortSelect register (0x0e8).

Intercepting

A packet is intercepted at the source port it arrives on and is forwarded to the port indicated by SIIsel (0x0e4) when the corresponding port bit is set in the SIMAP register (0x0e4).

Snooping

A packet can be snooped by destination port, source port, or by flow on a destination port by setting the corresponding port bit in DSMAP (0x0dc), SSMAP (0x0e0) or FSMAP (0x0f4) respectively. The packet is forwarded to the normal port and a copy is sent to the snoop port defined by the corresponding register: DSIsel (0x0dc), SSIsel (0x0e0), FSIsel (0xf4). For a flow to be snooped, the H bit in its flow handle must also be set. See Table 14. Multiple exceptions can occur, Table 18 shows these exceptions in order of decreasing priority.

PLL

There is a PLL in Epoch in order to improve the clock insertion delay through the chip. It must be enabled when the CLK frequency is 50MHz – 66MHz. It must be disabled when the clock frequency is lower than 50MHz. The RCPVBb signal should be pulled HIGH with 4K7R to enable the PLL or LOW to bypass it.

The PLL requires the filter network shown in Figure 8 to be connected to the ZLOOP signal.

The AVDD and AVSS power balls should be connected to a quiet Power and Ground.

JTAG

Device must be reset by asserting RESETb and TRST before JTAG functions can be used.

Please refer to IEEE Standard 1149.1 for information on using the JTAG functions. Please refer to Table 21 and Table 22 for function and ID information. A BSDL file is available; contact MUSIC technical support for more information.

Table 18: Exceptions Priority

Exception	Priority	Function
MTU Exceeded	1	Packet punted to port zero
Snoop by Source, Destination, or Flow	2	Packet duplicated to user selected snoop port SSIsel, DSIsel, or FSIsel
Drop bit set in flow handle	3	Packet dropped
Source Intercept	4	Packet diverted to user-selected intercept port
Host bit set in flow handle	5	Packet diverted to HostForwardPortSelect port or copied to FSIsel port if Snoop by Flow is on
TTL Threshold	6	Multicast packet dropped for destination port(s) where threshold is exceeded

Table 19: Statistics Counters

Address[8:2] Aligned [8:0]	Processor Register	Access	Bits	Priority	Function
0x1a8	CntOutOfBuffers	R	[15:0]	1	Number of buffer full aborts
0x1ac	CntOutOfBuffersThresh	R/W	[15:0]		Interrupt Threshold for above
0x1b0	CntOutOfPacketPointers	R	[15:0]	2	Number of packet pointer exhausted aborts
0x1b4	CntOutOfPacketPointersThresh	R/W	[15:0]		Interrupt Threshold for above
0x1b8	CntFifoOverflow	R	[15:0]	3	Number of multicast FIFO overflow aborts
0x1bc	CntFifoOverflowThresh	R/W	[15:0]		Interrupt Threshold for above.
0x1c0	CntPacketTooBig	R	[15:0]	4	Number of oversize (>64K-1) packet aborts
0x1c4	CntPacketTooBigThresh	R/W	[15:0]		Interrupt Threshold for above
0x1c8	CntPacketTooSmall	R	[15:0]	5	Number of undersize packet aborts. IP and IP re-inject packets < 20 bytes, IPX packets < 30 bytes.
0x1cc	CntPacketTooSmallThresh	R/W	[15:0]		Interrupt Threshold for above
0x1d0	CntL2Abort	R	[15:0]	6	Number of packets aborted by L2ABORT pin being asserted
0x1d4	CntL2AbortThresh	R/W	[15:0]		Interrupt Threshold for above
0x1d8	CntTTLThreshold	R	[15:0]	7	Number of multicast packets aborted because TTL threshold exceeded on ALL destination ports. Note: Multicast packet only counted once, not once for each destination port.
0x1dc	CntTTLThresholdThresh	R/W	[15:0]		Interrupt Threshold for above
0x1e0	CntL3FilterAbort	R	[15:0]	8	Number of packets dropped because port bitmap set to 0 in associated data RAM. Note: Packet is still counted as dropped even if it is snooped or diverted to host.
0x1e4	CntL3FilterAbortThresh	R/W	[15:0]		Interrupt Threshold for above
0x1e8	CntL4FilterAbort	R	[15:0]	9	Number of packets dropped because Drop bit set in L4 flow handle. Note: Packet is still counted as dropped if it is snooped or diverted to host.
0x1ec	CntL4FilterAbortThresh	R/W	[15:0]		Interrupt Threshold for above
0x1f0	CntTotalAborts	R	[15:0]	All	Total number of packets aborted
0x1f4	CntTotalAbortsThresh	R/W	[15:0]		Interrupt Threshold for above

Notes: Registers are all 32-bit aligned. 32-bit accesses only are supported. Priorities are numbered in decreasing priority order: 1 is highest priority.

Table 20: Interrupt Registers

Address[8:2] Aligned [8:0]	Processor Register	Access	Bits	Function
0x1f8	Interrupt bits	R	[13:0]	Each bit is set accordingly when the interrupt condition occurs. The bits are sticky, that is they remain set until the processor reads this register. They are automatically cleared at the end of the read operation. The INT pin is asserted when at least one enabled interrupt condition occurs. The INT pin is cleared when this register is read.
			[0]	Buffer full. The main packet buffer is completely full
			[1]	Buffer nearly full. The main packet buffer's number of used buffers has crossed the upper nearly full threshold.
			[2]	Packet Pointer RAM full. The packet pointer RAM is completely full.
			[3]	Packet Pointer RAM nearly full. The packet pointer RAM has crossed the upper nearly full threshold.
			[4]	Read 0
			[5]	CntOutOfBuffers counter crossed threshold
			[6]	CntOutOfPacketPointers counter crossed threshold
			[7]	CntFifoOverflow counter crossed threshold
			[8]	CntPacketTooBig counter crossed threshold
			[9]	CntPacketTooSmall counter crossed threshold
			[10]	CntL2Abort counter crossed threshold
			[11]	CntTTLThreshold counter crossed threshold
			[12]	CntL3FilterAbort counter crossed threshold
			[13]	CntL4FilterAbort counter crossed threshold
[15:14]	Read 0			
0x1fc	Interrupt Mask and INT pin polarity	R/W	[13:0]	Mask for the Interrupt register. If a bit is set the corresponding interrupt is enabled. Resets to 0.
			[14]	Read 0
			[15]	Polarity of INT pin 0 = Active LOW 1 = Active HIGH

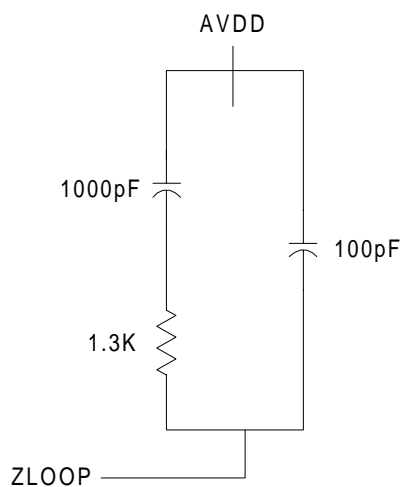


Figure 8: PLL ZLOOP Circuit

Table 21: JTAG Functions

Function	Code
EXT TEST	0000
BYPASS	1111
SAMPLE	0001
ID CODE	0010
CLAMP	0100
HIGH-Z	0011
TEST MODE	0111

Table 22: JTAG ID

Version	ID	Manufacturers ID
0xX	0xEC01	0x133

Note: 4-field changes with device revisions.

ELECTRICAL

Absolute Maximum Ratings

Supply Voltage	-0.5 to 4.6 Volts
Temperature	-40°C to 85°C
Storage Temperature	-55°C to 125°C
DC Output Current	20 mA (per output, one at a time, one second duration)

Stresses exceeding those listed under Absolute Maximum Ratings may induce failure. Exposure to absolute maximum ratings for extended periods may reduce reliability. Functionality at or above these conditions is not implied.

All voltages referenced to GND.

Operating Conditions

Symbol	Parameter	Min.	Typical	Max.	Units	Notes
V _{CC}	Operating supply voltage	3.0	3.3	3.6	Volts	
V _{IH}	Input voltage logic 1	2.0		5.5	Volts	5V tolerant pins
		2.0		V _{CC} + 0.3	Volts	3.3V only pins
V _{IL}	Input voltage logic 0	-0.3		0.8	Volts	
T _A	Ambient operating temperature	0		70	°C	Still air

Electrical Characteristics

Symbol	Parameter	Min.	Typical	Max.	Units	Notes
I _{CC}	Average power supply current		TBD	TBD	mA	
I _{CC(SB)}	Stand-by power supply current		TBD	TBD	mA	
V _{OH}	Output voltage logic 1	2.4			Volts	I _{OH} = -4.0 mA
V _{OL}	Output voltage logic 0			0.4	Volts	I _{OL} = 4.0 mA
I _{Iz}	Input leakage current	-2		2	μA	V _{SS} ≤ V _{IN} ≤ V _{CC}
I _{Oz}	Output leakage current	-10		10	μA	V _{SS} ≤ V _{OUT} ≤ V _{CC}

Capacitance

Symbol	Parameter	Max.	Units	Notes
C _{IN}	Input capacitance	6	pF	f = 1 MHz, V _{IN} = 0V
C _{OUT}	Output capacitance	7	pF	f = 1 MHz, V _{OUT} = 0V

AC Test Conditions

Parameter	Units
Input signal transitions	0.0 Volts to 3.3 Volts
Input signal rise time	< 3 ns
Input signal fall time	< 3 ns
Input timing reference level	1.5 Volts
Output timing reference level	1.5 Volts

TIMING DIAGRAMS

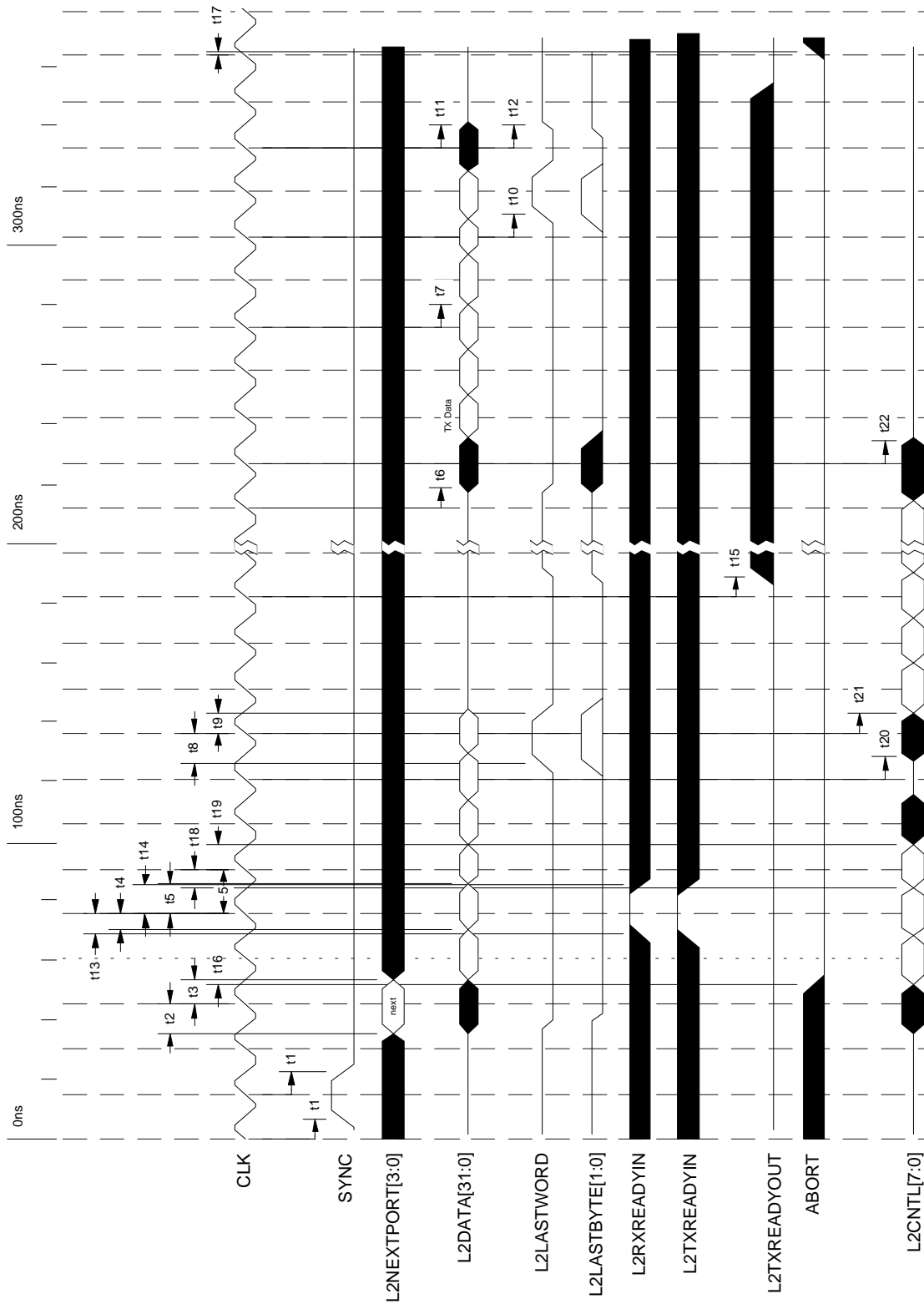


Figure 9: L2 TDM Bus Timing

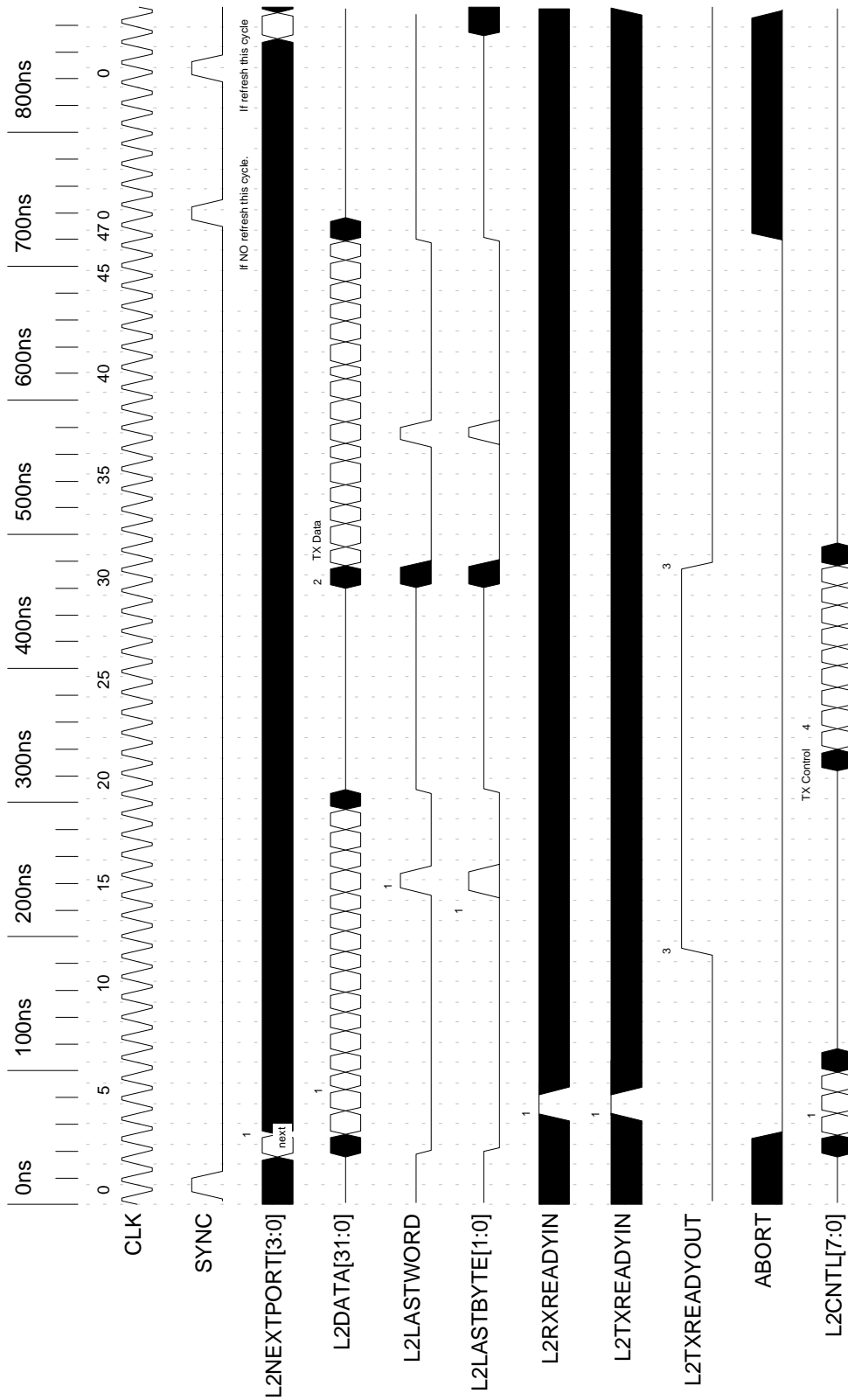


Figure 10: L2 TDM Bus Cycle

Notes:

1. These signals are sampled by Epoch on the clock cycles shown.
2. TX L2DATA is asserted by Epoch from CLK 30 and should be sampled by interface hardware from CLK 31.
3. L2TXREADYOUT is asserted by Epoch at CLK 11 and de-asserted at CLK 30.
4. TX L2CNTL is asserted by Epoch from CLK 21 and should be sampled by interface hardware from CLK 22.

Table 23: L2 TDM Bus Timing

Name	PLL Disabled Min. ns	PLL Disabled Max. ns	PLL Enabled Min. ns	PLL Enabled Max. ns	Comment
t1		13.5		9	SYNC delay from CLK
t2	0		2		L2NEXTPORT setup to CLK
t3	5		0.5		L2NEXTPORT hold from CLK
t4	0		2		RX L2DATA[31:0] setup to CLK
t5	5		0.5		RX L2DATA[31:0] hold from CLK
t6		13.5		9	TX L2DATA[31:0] driven from CLK
t7		13.5		9	TX L2DATA[31:0] delay from CLK
t8	0		2		RX L2LASTWORD/L2LASTBYTE[1:0] setup to CLK
t9	5		0.5		RX L2LASTWORD/L2LASTBYTE[1:0] hold from CLK
t10		13.5		9	TX LASTWORD/LASTBYTE[1:0] delay from CLK
t11		13.5		9	TX L2DATA[31:0] float delay from CLK
t12		13.5		9	TX L2LASTWORD/L2LASTBYTE[1:0] float delay from CLK
t13	0		2		L2RXREADYIN/L2RXREADYOUT setup to CLK
t14	5		0.5		L2RXREADYIN/L2RXREADYOUT hold from CLK
t15		13.5		9	L2TXREADYOUT delay from CLK
t16	0		2		ABORT setup to CLK
t17	5		0.5		ABORT hold from CLK
t18	0		2		RX L2CNTL[7:0] setup to CLK
t19	5		0.5		RX L2CNTL[7:0] hold from CLK
t20		13.5		9	TX L2CNTL[7:0] driven from CLK
t21		13.5		9	TX L2CNTL[7:0] delay from CLK
t22		13.5		9	TX L2CNTL[7:0] float delay from CLK

Note: Timings are specified for 50pF load.

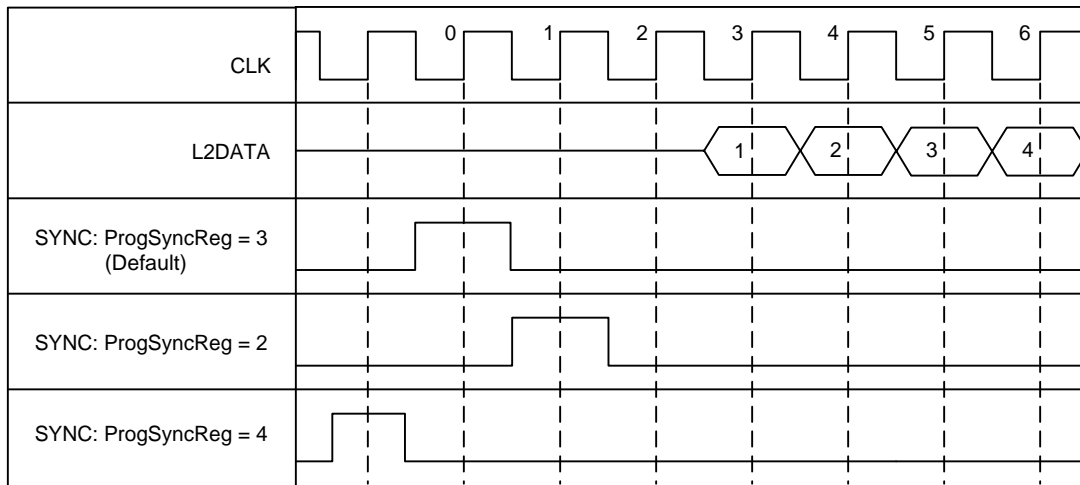


Figure 11: ProgSyncReg Register Examples

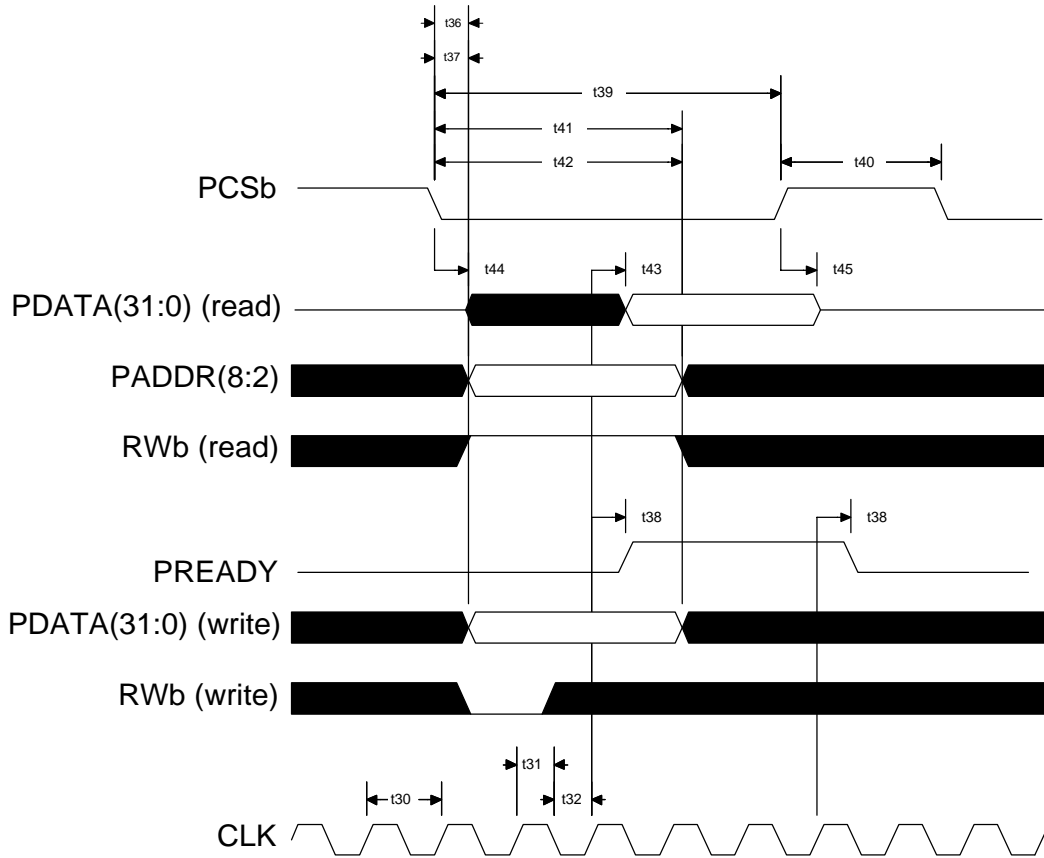


Figure 12: Processor Interface Timing

Table 24: Processor Interface Timing

Name	PLL Disabled Min. ns	PLL Disabled Max. ns	PLL Enabled Min. ns	PLL Enabled Max. ns	Comment
t30	20	DC	15	20	CLK period
t31	8	DC	6	8	CLK high
t32	8	DC	6	8	CLK low
t36	-1 CLK		-1 CLK		PADDR[8:2]/RW/b setup to PCSb low
t37	-1 CLK		-1 CLK		Write PDATA[31:0] setup to PCSb low
t38		13.5		8.5	PREADY delay from CLK
t39	5 CLKs		5 CLKs		PCSB low time
t40	2 CLKs		2 CLKs		PCSB high time
t41			Hold until PREADY HIGH		PADDR[8:2]/RWb hold from PCSb low.
t42			Hold until PREADY HIGH		Write PDATA[31:0] hold from PCSb low.
t43		13.5		8.5	Read PDATA[31:0] delay from CLK
t44		4 CLKs + 13.5		4 CLKs + 13.5	PDATA[31:0] driven from PCSb low
t45		1 CLKs + 13.5		1 CLKs + 13.5	PDATA[31:0] float from PCSb high

PACKAGE INFORMATION

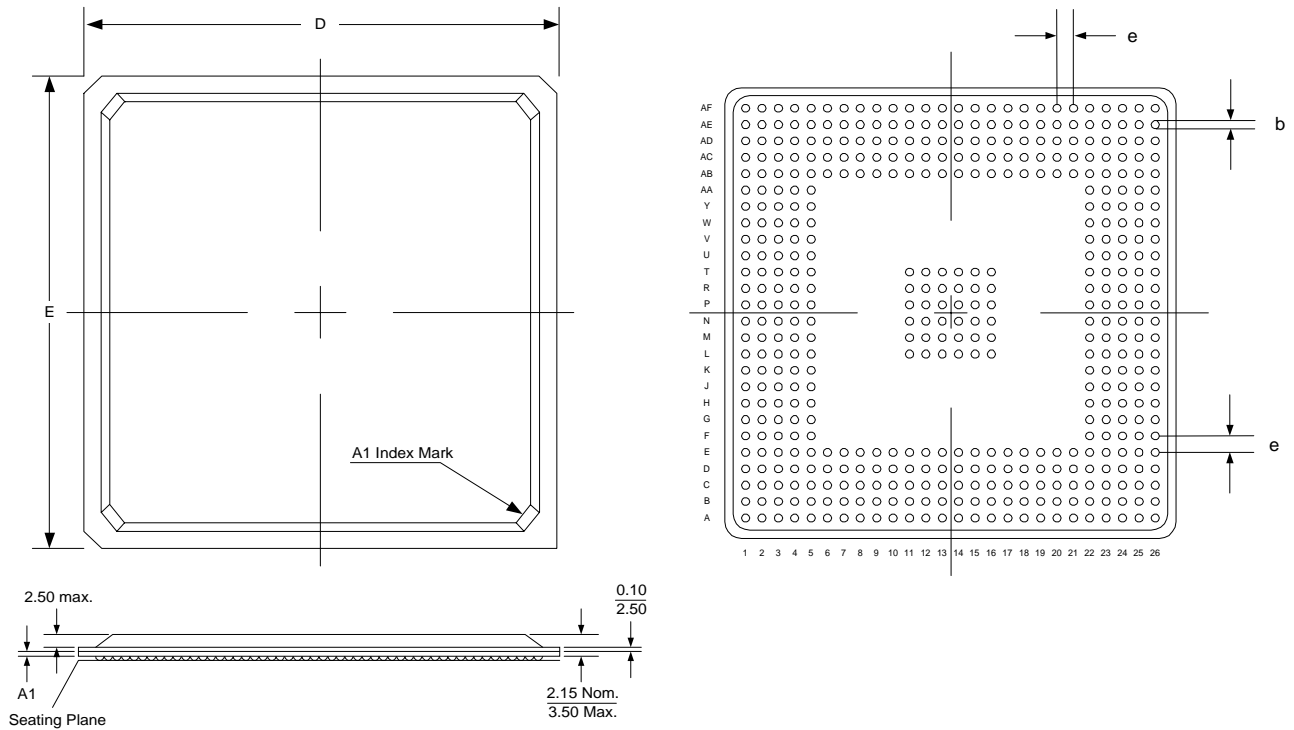


Figure 13: Package Detail

Table 25: Package Dimensions

Symbol	Millimeters			Inches		
	Min.	Min.	Max.	Min.	Nom.	Max.
A1	0.50	0.60	0.70	0.020	0.024	0.028
b	0.60	0.75	0.90	0.024	0.030	0.035
D		35.00			1.378	
E		35.00			1.378	
e		1.27			0.050	

ORDERING INFORMATION

Part Number	Cycle Time	Package	Temperature	Voltage
MUSA16P14-B456C	15ns	456-Pin BGA	0 - 70° C	3.3

MUSIC Semiconductors' agent or distributor:

MUSIC Semiconductors reserves the right to make changes to its products and specifications at any time in order to improve on performance, manufacturability or reliability. Information furnished by MUSIC is believed to be accurate, but no responsibility is assumed by MUSIC Semiconductors for the use of said information, nor for any infringements of patents or of other third-party rights which may result from said use. No license is granted by implication or otherwise under any patent or patent rights of any MUSIC company.

© Copyright 2000, MUSIC Semiconductors



<http://www.music-ic.com>
email: info@music-ic.com

Worldwide Headquarters

MUSIC Semiconductors
2290 N. First St., Suite 201
San Jose, CA 95131
USA

Tel: 408 232-9060
Fax: 408 232-9201

USA Only: 800 933-1550 Tech Support
888 226-6874 Product Info

Asian Headquarters

MUSIC Semiconductors
Special Export Processing Zone
Carmelray Industrial Park
Canlubang, Calamba, Laguna
Philippines

Tel: +63 49 549-1480

Fax: +63 49 549-1024
Sales Tel/Fax: +632 723-6215

European Headquarters

MUSIC Semiconductors
P. O. Box 184
6470 ED Eygelshoven
The Netherlands

Tel: +31 43 455-2675

Fax: +31 43 455-1573