MC92520 ATM Cell Processor User's Manual

MC92520UM/D
Rev. 0, 12/2000

Digital DNA from Motorola

MOTOROLA

DigitalDNA and Mfax are trademarks of Motorola, Inc.

Information in this document is provided solely to enable system and software implementers to use Motorola microprocessors. There are no express or implied copyright licenses granted hereunder to design or fabricate Motorola integrated circuits or integrated circuits based on the information in this document.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado 80217. 1–303–675–2140 or 1–800–441–2447

JAPAN: Motorola Japan Ltd.; SPS, Technical Information Center, 3–20–1, Minami–Azabu. Minato–ku, Tokyo 106–8573 Japan. 81–3–3440–3569

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong. 852–26668334

Technical Information Center: 1–800–521–6274

HOME PAGE: http://www.motorola.com/semiconductors

Document Comments: FAX (512) 895-2638, Attn: RISC Applications Engineering

World Wide Web Addresses: http://www.motorola.com/PowerPC
http://www.motorola.com/NetComm
http://www.motorola.com/ColdFire

CONTENTS

CONTENTS

Chapter 3
System Operation

CONTENTS

Chapter 4
External Interfaces

CONTENTS

Chapter 5
Data Path Operation

CONTENTS

Chapter 6
Protocol Support

CONTENTS

CONTENTS

CONTENTS

CONTENTS

CONTENTS

CONTENTS

CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS

ILLUSTRATIONS

ILLUSTRATIONS

ILLUSTRATIONS

ILLUSTRATIONS

ILLUSTRATIONS

ILLUSTRATIONS

Figure Number	Title	Page Number

TABLES

TABLES

TABLES

TABLES

About This Book

The primary objective of this user's manual is to define the functionality of the MC92520 asynchronous transfer mode cell processor (ATMC). The ATMC typically operates in ATM networks within switches and access multiplexers to process and route cells.

The information in this book is subject to change without notice, as described in the disclaimers on the title page of this book. As with any technical documentation, it is the readers' responsibility to use the most recent version of the documentation. For more information, contact your sales representative.

Audience

This manual is intended for system software and hardware developers and applications programmers who want to develop products using the MC92520 ATM cell processor. It is assumed that the reader understands ATM networks, cell processor system design, and datapath switching.

Organization

Following is a summary and a brief description of the major sections of this manual:

- Chapter 1, "Introduction," is useful for readers who want a general understanding of the features of the MC92520 and the differences between it and earlier ATMC processors.
- Chapter 2, "Functional Description," describes the ATMC system environment and reviews the basic functionality of the MC92520 components.
- Chapter 3, "System Operation," illustrates the operational aspects of an MC92520 application, i.e., how the ATMC is used to interact with other system components.
- Chapter 4, "External Interfaces," reviews the clock and reset functionality of the MC92520, and explains the operation of the ATMC PHY-side, switch-side, external memory, and microprocessor interfaces.
- Chapter 5, "Data Path Operation," describes the functions performed by the ATMC during data path ingress and egress operations.

- Chapter 6, “Protocol Support,” describes the industry protocols supported by the ATMC, including usage parameter control, cell loss priority bit management, guaranteed frame rate, available bit rate, and operations and maintenance functions.
- Chapter 7, “Programming Model,” describes the ATMC programming model including registers, external memory mapping, and a description of the data structures used in programming the chip.
- Chapter 8, “Test Operation,” discusses the MC92520’s JTAG boundary scan architecture and test access port (TAP) controller.
- Chapter 9, “Product Specifications,” discusses the MC92520 signals, physical and electrical characteristics, and mechanical and packaging specifications.
- Appendix A, “UPC/NPC Design,” explains traffic management of the MC92520 using usage and network parameter controls (UPC/NPC).
- Appendix B, “Maintenance Slot Calculations,” explains how maintenance slots are configured in the MC92520.
- Appendix C, “VC Bundling,” describes the use of the MC92520 for VC bundling.
- Appendix D, “MC92520 Applications,” gives examples of several kinds of applications that can use the MC92520.
- Appendix E, “BSDL Code,” contains the BSDL code.
- This manual also includes an index.

Additional Reading

Following is a list of additional reading that provides background for the information in this manual:

- ANSI Recommendation I.361, “B-ISDN ATM Layer Specification,” November, 1995
- ITU-T Recommendation I.610, “B-ISDN Operation and Maintenance Principles and Functions,” November, 1995
- ATM Forum, “ATM User-Network Interface Specification,” Version 3.1, September, 1994
- ATM Forum, “B-ISDN Inter Carrier Interface (B-ICI) Specification,” Version 1.1, September, 1994
- ATM Forum, “B-ISDN Inter Carrier Interface (B-ICI) Specification,” Version 2.0 Letter Ballot, November, 1995
- ATM Forum, “Traffic Management Specification,” Version 4.1
- ANSI T1S1.5/93-004R2, “Broadband ISDN - Operations and Maintenance Principles and Functions,” January, 1994

- Bellcore TA-NWT-01110, “Broadband ISDN Switching Systems Generic requirements,” Issue 1, August, 1992
- Bellcore GR-1113-CORE, “Asynchronous Transfer Mode (ATM) and ATM Adaptation Layer (AAL) Protocols,” Issue 1, July, 1994
- Bellcore GR-1248-CORE, “Generic Requirements for Operations of ATM Network Elements,” Issue 1, August, 1994
- Motorola, “MPC860 Integrated Communications Processor User’s Manual” for the MPC8*xx* family, and “MPC8260 PowerQUICCII User’s Manual” for the MPC82*xx* family of communications processors.
- “UTOPIA, An ATM-PHY Interface Specification, Level 1, Version 2.01,” March 21, 1994.
- “UTOPIA Level 2 Specification, Version 1.0,” June, 1995
- Bellcore TA-TSV-001408, “Generic Requirements for Exchange PVC Cell Relay Service,” Issue 1, August, 1993
- Motorola, “MCM69C432 16K x 64 CAM” and “MCM69C232 4K x 64 CAM” data sheets

Conventions

This document uses the following notational conventions:

- Abbreviations or acronyms for registers are shown in uppercase text, as in REG[FIELD]. Specific bits, fields, or ranges, as well as references to readings, appear in brackets. For example, MSR[LE] refers to the little-endian mode enable bit in the machine state register.

Signal representations:

- Active high signals (logic one) use the signal name (that is, A0).
- Active low signals (logic zero) use the signal name with an overbar (for example, $\overline{\text{WE}}$ or $\overline{\text{OE}}$).
- In certain contexts, such as a signal encoding, “x” indicates a don’t care.

Numeric representations:

- Hexadecimal values start with a “0x” sign (that is, 0x0FF0 or 0x80).
- Decimal values have no symbol attached to the number (for example, numbers such as 10 or 34 have no symbol attached).
- Binary values start with the letter “b” (such as b1010 or b0011).

Acronyms and Abbreviations

Table i contains acronyms and abbreviations that are used in this document.

Table i. Acronyms and Abbreviated Terms

Term	Meaning
ABR	Available bit rate
ATM	Asynchrounous transfer mode
ATMC	Asynchronous transfer mode cell processor
BAT	Block address translation
BIST	Built-in self-test
BRC	Backward reporting cell
CAP	Cell arrival period
CDV	Cell delay variation
CLP	Cell loss priority
CR	Condition register
DMA	Direct memory access
EFCI	Egress flow connection indicator
EFI	Egress connection identifier
EPD	Early packet discard
EPHI	Egress PHY interface
EPU	Egress cell processing unit
ESWI	Egress switch interface
FCR	Fair cell rate
FIFO	First-in-first-out
FMC	Forward monitoring cell
GFR	Guaranteed frame rate
GPR	General-purpose register
HEC	Header error control
HOL	Head of line
ICI	Ingress connection identifier
IEEE	Institute for Electrical and Electronics Engineers
IPHI	Ingress PHY interface
IPU	Ingress cell processing unit
ISWI	Ingress switch interface
L2	Secondary cache
LR	Link register
LSB	Least-significant byte
MFS	Maximum frame size

Table i. Acronyms and Abbreviated Terms (Continued)

Term	Meaning
MMU	Memory management unit
MSB	Most-significant byte
MSR	Machine state register
NNI	Network–network interface
OAM	Operations and maintenance
PCR	Peak cell rate
PDU	Protocol data unit
PHY	Physical layer
PLL	Phase-locked loop
PPD	Partial packet discard
QOS	Quality of service
RFD	Random frame drop
RM	Resource management
SPR	Special-purpose register
SR	Segment register
TAP	Test access port
TTL	Transistor-to-transistor logic
UBR	Unspecified bit rate
UDF	Universal data format
UNI	User–network interface
UPC	User parameter control
VBR	Variable bit rate
VC	Virtual connection
VCC	Virtual channel connection
VCI	Virtual connection identifier
VP	Virtual path
VPC	Virtual path connection
VPI	Virtual path identifier

Chapter 1
Introduction

This chapter introduces ATM networks, describes the features of the MC92520 ATM cell processor, the difference between the MC92520 and the MC92510, and contrasts the MC92520 with its predecessor, the MC92501. With very few exceptions, which are noted, all information in this manual applies to both the MC92510 and the MC92520. A summary of these differences can be found in Section 1.6, “The MC92510 Versus the MC92520.”

The material in the chapter is conceptually arranged in the following manner:

- Section 1.1, “ATM Networks” is a general discussion of the ATM network.
- Section 1.2, “MC92520 Applications,” explains how the processor is used in line cards and in access multiplexers.
- Section 1.4, “MC92520 Features,” describe the functionality of the MC92520 ATM Cell Processor.
- Section 1.5, “The MC92520 Versus the MC92501,” discusses the differences between the MC92520 and the MC92501.
- Section 1.6, “The MC92510 Versus the MC92520” explains the difference between the MC92510 and the MC92520.

1.1 ATM Networks

The increasing demand for broadband data transmission has led to the development of international protocols issued by the American National Standards Institute (ANSI), International Telecommunications Union-Telecommunications (ITU-T), and the European Telecommunication Standards Institute (ETSI). These standards and the recommendations issued by the ATM Forum, Bellcore, and others are provided to ensure consistent, reliable, and uninterrupted data transmission world-wide.

The basic unit of transfer defined by these protocols is called a cell. Each cell is composed of 53 bytes that include a 5-byte header and 48 bytes of data. The header portion of the cell includes several sections of identifying information used by the cell processor to route and transmit the cell, including the type of data being transmitted and its origin and destination. As with many other means of electronic transfer, the cells of information must be combined with other cells (for efficiency of transfer) and transmitted. During transmission, the cells are evaluated and, if required, separated from the transmission stream and rerouted, similar

to commuters who share transportation routes during parts of their journeys and then transfer to alternate routes to complete their journeys.

Actual data transmission is performed asynchronously because the distance over which the transfer occurs prevents the use of the same clock signal at both ends and even within the transfer network. Therefore, the transfer method has been defined as asynchronous transfer mode (ATM). Each network consists of user end stations that transmit and receive the 53-byte data cells over virtual connections.

Each virtual connection is a specific combination of physical links connected by switches. The physical links that make up a virtual connection are chosen when the connection is established. Each connection is assigned a unique connection identifier that is placed in the header of each cell by the transmitting equipment and is used by the receiving equipment to route the cell to each physical link previously selected to be part of the connection path. All cells belonging to a specific virtual connection follow the identical path from the transmitting end station through the switches to the receiving end station. Figure 1-1 shows a typical ATM network.

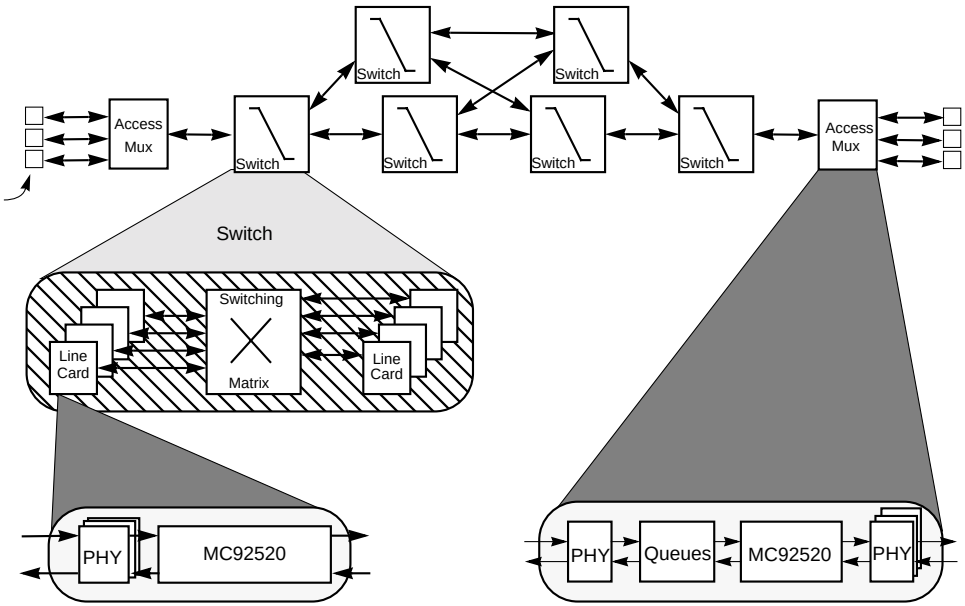


Figure 1-1. Typical ATM Network Structure

An ATM network is many physical links connected by switches, and many different virtual connections are created with these switches and links. Within the ATM network core, the switches route multiple links to multiple links. A typical ATM core switch consists of a switch matrix and some line cards. The switch matrix handles the routing of the ATM cells,

and the line cards interface between the physical signal lines and the matrix. The line cards perform their interface function by extracting incoming cells from the arriving bit stream and converting outgoing cells into a bit stream for transmission. Each physical link, or group of links, has a card. Line cards also perform ATM layer functions.

At the edges of the ATM network, access multiplexers route a single link to multiple links. An access multiplexer is typically connected to one physical (PHY) layer device on the network side and to multiple PHY layer devices on the subscriber side.

1.2 MC92520 Applications

As shown in Figure 1-1, the MC92520 ATM cell processor can be used both in line cards used by switching systems in ATM network cores and in access multiplexers. The primary function of the MC92520 in either application is to provide ATM-layer cell processing and routing functions.

1.2.1 ATM Network Line Card

In a typical line card, the MC92520 serves as an interface between a physical layer device and an ATM switch matrix, as shown in Figure 1-2. The MC92520 uses a fast external memory for storing the ATM virtual connection information for the cells it processes. The microprocessor is used for configuration, control, and status monitoring of the MC92520 and is responsible for initializing and maintaining the external memory. The MC92520 is the master of the external memory bus. At regular intervals the MC92520 allows the microprocessor to access the external memory for updating and maintenance.

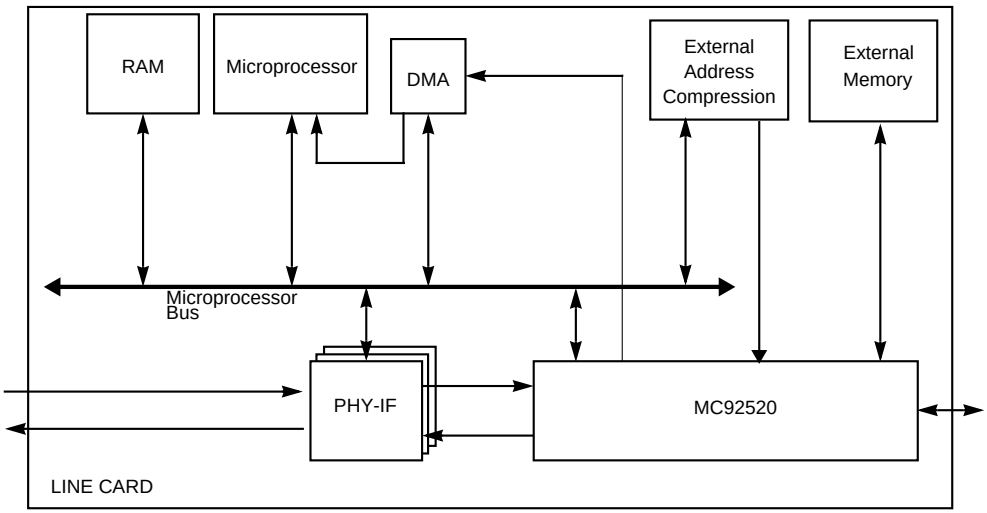


Figure 1-2. Typical ATM Line Card Application

The physical interface (PHY-IF) implements the physical layer functions of the broadband ISDN (B-ISDN) protocol reference model. This includes the physical medium dependent functions required to transport ATM cells between the ATM user and the ATM switch (i.e., a user-network interface or UNI) or between two ATM switches (i.e., a network-node interface or NNI). The cells are transferred between the physical interface and the MC92520 using the UTOPIA standard. The MC92520 implements ATM layer functions required to transfer cells to and from the switch over virtual connections. These functions include usage enforcement, address translation, and operation and maintenance (OAM) processing.

1.2.2 ATM Network Access Multiplexer

Figure 1-3 shows two example configurations for ATM network access multiplexers using an MC92520 ATM cell processor. In both designs, the MC92520 interfaces the PHY layer devices and the queuing point with the MC92520 switch interface connected to the queues.

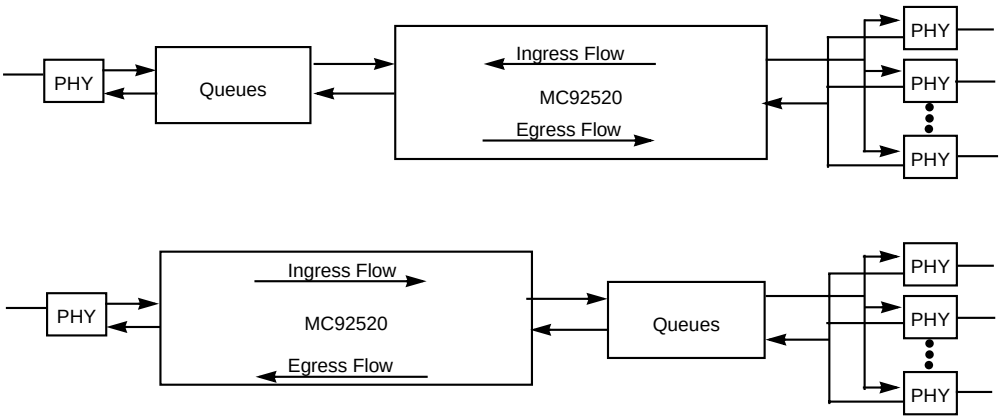


Figure 1-3. MC92520 Network Access Multiplexer Design Examples

1.3 MC92520 Block Diagram

Figure 1-4 is a diagram illustrating the MC92520.

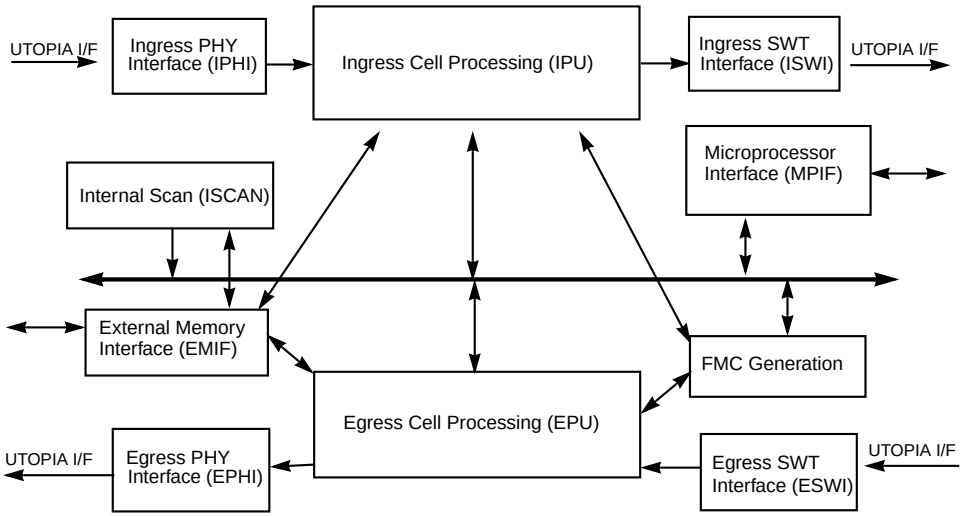


Figure 1-4. MC92520 Block Diagram

1.4 MC92520 Features

The MC92520 ATM cell processor contains these features and performs the following functions:

- Implements ATM layer functions for broadband ISDN according to ITU and ANSI recommendations, ATM Forum specifications, and Bellcore recommendations
- Provides a cell throughput bandwidth of up to 600 Mbps in each direction and is therefore capable of processing ATM cells from 1 SONET STM-4 or SONET STS-12c link, as well as several SONET STS-3/3c, SONET STS-1, DS3 PLCP, or any other physical links not exceeding an aggregate cell rate of up to 600 Mbps
- Optionally supports up to 16 physical links
- Optionally configured as a user-network interface (UNI) or network-node interface (NNI) on a per-link basis
- Operates in conjunction with an external memory (up to 16 Mbytes) to provide context management for up to 64K virtual connections
- Provides per-connection cell counters with the ability to maintain multiple copies of the counter tables and dynamically switch between them
- Provides per-link cell counters in both directions

- Provides per-connection usage parameter control (UPC) or network parameter control (NPC) using a leaky bucket design with up to four buckets per connection
- Supports packet-based UPC using three methods: partial packet discard (PPD), early packet discard (EPD), and limited EPD
- Supports selective discard on CLP = 1 and CLP = 0 + 1 on selected connections
- Provides support for the following operation and maintenance (OAM) functions:
  - Continuity check function for all connections
  - Fault management loop-back test on all connections
  - Bidirectional OAM performance monitoring on all connections
- Supports virtual path (VP) and virtual channel (VC) level alarm surveillance on all connections using an internal scan process to generate and insert OAM cells
- Supports available bit rate (ABR) on all connections
- Supports guaranteed frame rate (GFR) on all connections
- Supports changing a resource management (RM) cell's priority by marking the ingress switch parameters
- Supports CLP transparency
- Provides a slave microprocessor interface including a 32-bit data bus
- Provides indirect access to its external memory
- Provides byte-swapping on cell payloads to and from the microprocessor bus to support both big-endian and little-endian buses
- Supports cell insertion into the cell streams using direct access registers that may be written by the microprocessor or by a DMA device
- Supports copying cells from the cell streams using direct access registers that may be read by the microprocessor or by a DMA device
- Glueless connection to the MPC860 (PowerQUICC) and MPC8260 (PowerQUICC II)
- Glueless connection to external memory ZBT RAMs
- Glueless connection to external address compression CAMs
- Enables masking of overhead ECI and MTTS fields before use
- Supports multicast operation
- 1.8-V core power supply
- 3.3-V I/O
- Package configuration: 352-pin PBGA

1.4.1 Ingress Features

The MC92520 assists the processor's ingress function of transferring cells from the physical interface to the switch with the following functions:

- Receives ATM cells from one or more PHY devices via a 16-bit or 8-bit wide, parity-protected data bus supporting UTOPIA level 1 or UTOPIA level 2 master interfaces
- Decouples PHY timing from switch timing using independent clocks and a FIFO in the physical interface
- Performs ingress cell discrimination based on pre-assigned ATM cell header values
- Provides either a restricted address table lookup scheme for ingress address compression or support for an external address compression mechanism
- Reads virtual connection related UPC/NPC, OAM, and switch context parameters through a 32-bit wide interface to an external memory
- For each connection, the MC92520 provides the following:
  - A usage count
  - An option to copy cells to the microprocessor
  - UPC/NPC policing including detection and counting of violating cells
- Supports packet-based UPC using three methods: PPD, EPD, and limited EPD
- Supports OAM continuity check, alarm surveillance, loop-back test, and performance monitoring on all connections
- Supports available bit rate (ABR) on all connections
- Supports guaranteed frame rate (GFR) on all connections
- Supports changing an RM cell's priority by marking the ingress switch parameters
- Supports CLP transparency
- Supports insertion of cells into the ingress cell flow
- Optionally performs VPI/VCI translation
- Transfers ATM cells to the switch using a UTOPIA-style, 8- or 16-bit wide, parity-protected slave interface, optionally adding associated switch context parameters
- Delay of 3–5 cell times from the PHY to the switch

1.4.2 Egress Features

The MC92520 assists the processor's egress function of transferring cells from the switch to the physical interface with the following functions:

- Receives ATM cells and associated switch context parameters from the switch using a UTOPIA-style, 8- or 16-bit wide, parity-protected slave interface
- Provides per-PHY FIFOs and optional switch-side multi-PHY operation in combination with the use of a PHY-side multi-PHY interface

- Provides optional multicast identifier to connection identifier translation
- Reads virtual connection related UPC/NPC, OAM and address translation parameters through a 32-bit wide interface to an external memory
- For each connection, the MC92520 provides the following:
  - A usage count
  - An option to copy cells to the microprocessor
  - UPC/NPC policing including detection and counting of violating cells
- Supports packet-based UPC using three methods: PPD, EPD, and limited EPD
- Supports OAM continuity check, alarm surveillance, loop-back, and performance monitoring test on all connections
- Supports available bit rate (ABR) on all connections
- Supports CLP transparency
- Supports insertion of cells into the egress cell flow
- Performs VPI/VCI translation
- Provides head-of-line (HOL) blocking detection, avoidance, and reporting
- Transfers ATM cells to one or more PHY devices using an 8- or 16-bit wide, parity-protected data bus supporting UTOPIA level 1 or UTOPIA level 2 master interfaces
- Decouples PHY timing from switch timing using independent clocks and a FIFO in the physical interface
- Delay of 3–5 cell times from the switch to the PHY

1.5 The MC92520 Versus the MC92501

The MC92520 is an updated version of the MC92501. Motorola's ATM strategy is to update products in accordance with the latest technology standards while maintaining backward compatibility. Due to UTOPIA and external memory interface changes required to achieve the higher cell processing bandwidth, the MC92520 is *not* pin-compatible with the MC92501. All changes and additions to the programming model, however, have been made to be backward-compatible with the MC92501/MC92500 when possible. With the exception of maintenance slot accesses, most new features are generally enabled through previously reserved bit positions in existing configuration registers. MC92520 features that are not included on the MC92501 can be summarized as follows:

- Full-duplex cell rates up to 600 Mbps
- An 8- or 16-bit wide data path on both PHY-side and switch-side UTOPIA interfaces with clock rates up to 50 MHz
- Egress switch-side multi-PHY interface with per-PHY FIFOs
- Egress head-of-line (HOL) blocking avoidance and reporting

- Guaranteed frame rate (GFR) support
- Improved host interface supporting 8x bandwidth to external memory in operate mode
- Support of VPI = 0 as egress connection identifier (ECI)
- Glueless PowerQUICC or PowerQUICC II processor interfaces
- Glueless MCM69C232 or MCM69C432 CAM interface

The following MC92501 features are not supported:

- Motorola multi-PHY extension to UTOPIA level 1 mode of operation
- Support for third DMA request signal
- Support for TXCLR (replaced by HOL blocking avoidance)

1.6 The MC92510 Versus the MC92520

All of the differences between the MC92510 and the MC92520 derive from the fact that the MC92510 is a pin-compatible version of the MC92520 without an internal PLL.

Software can differentiate the MC92510 and MC92520 through unique identifcation numbers in the revision register RR. Testers can differentiate the MC92510 and MC92520 through unique JTAG IDs.

All functional differences are due to the lack of an internal PLL in the MC92510. As a consequence, writing the MC92510 PLL configuration and control registers has no effect. Reading the MC92510 PLL status register returns results consistent with an MC92520 configuration that has the internal PLL turned off.

Due to the missing PLL function, the MC92510's maximum operating frequency is limited to 50 MHz and thus the maximum usable data path bandwidth is 300 Mbps.

Chapter 2
Functional Description

The MC92520 implements Broadband ISDN (B-ISDN) user network interface/network node interface (UNI/NNI) asynchronous transfer mode (ATM) layer functions and provides context management for up to 64K virtual connections (VCs): either virtual channel connections (VCCs), or virtual path connections (VPCs). ATM cells belonging to a particular VCC on a logical link have the same unique VPI/VCI value in the cell header. Similarly, cells with the same VPC on the same logical link share a unique VPI.

2.1 System Environment

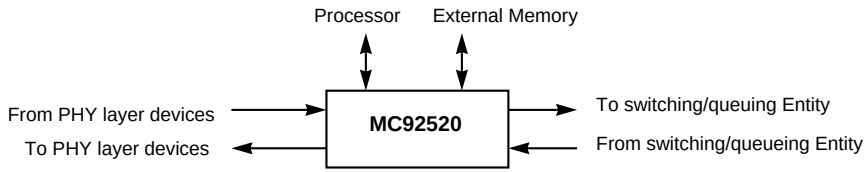


Figure 2-1. ATMC Schematic Interface

The asynchronous transmission mode cell processor (ATMC) is always placed between PHY layer devices and the switching/queuing entity. The PHY devices are connected to the MC92520 using 8- or 16-bit wide UTOPIA data paths operating at up to 50 MHz. The maximum cell processing bandwidth of the MC92520 is 662.5 Mbps (1,562,500 cells/sec) in both directions, when operated at its maximum frequency of 100 MHz. In practice, approximately 10% of the cell processing bandwidth is reserved for microprocessor-initiated context memory and table maintenance. The microprocessor initializes and provides real-time control of the MC92520 using slave accesses. The MC92520 operates with external memory that provides one context entry for each active connection. The entry consists of two types of context parameters:

- Static—These parameters are loaded into the context memory when the VC is established and are valid for that connection duration. Static parameters include traffic descriptors, OAM flags, and parameters used by the ATM switch.
- Dynamic—These context parameters (which include cell counters, UPC/NPC fields and OAM parameters) can be modified while cells belonging to that particular connection are processed by the MC92520.

The microprocessor accesses the external memory from time to time through the MC92520 to collect traffic statistics and update the OAM parameters.

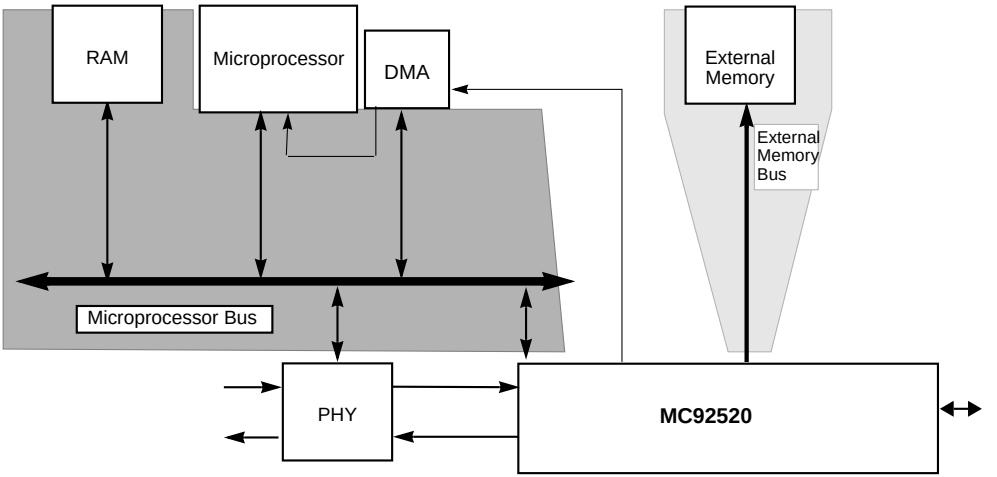


Figure 2-2. MC92520 in Operate Mode

In operate mode, the MC92520 has exclusive access to external memory, as shown in Figure 2-2. As cells are processed, the associated context entries are read and any updated dynamic parameters are written back. While each cell is processed by the MC92520, the microprocessor can issue one indirect external memory access as described in Section 3.5.1 “Indirect Memory Access.” Because this access is limited to one per cell time and asynchronous to the cell processing, it cannot be used for updates of active contexts or for updates that require more than one access to preserve table coherence. Therefore, the MC92520 provides means to reserve so-called maintenance slots at user-programmable cell intervals to perform up to 64 queued external memory accesses. One maintenance slot consumes one MC92520 cell time, but the break in cell processing has no negative effect as long as the MC92520 cell processing rate is at least the cell rate produced by the ingress PHY and egress switch interfaces.

To use a maintenance slot, the microprocessor queues a sequence of access requests that will be executed by the MC92520 during the next available maintenance slot. Any read access results are in turn queued by the MC92520 to be fetched by the microprocessor. The mechanism provides access grouping for an atomic access sequence (a sequence of requests that is guaranteed to complete in one maintenance slot) and microprocessor interrupt upon access completion on a per-access basis. For a more detailed description of maintenance slot configuration and use, refer to Section 3.2.4, “Configuring External Memory Maintenance Slots.”

Maintenance slot accesses can be used by the microprocessor for one or more of the following tasks:

- Connection setup and tear down
- Statistics collection
- Updating OAM parameters of active connections

The microprocessor is responsible for guaranteeing the coherence of the external memory at the end of each maintenance slot.

2.2 MC92520 Functional Description

The primary functions performed by the MC92520 can be divided in functions performed on the ingress (PHY-to-switch) cell flow and functions performed on the egress (switch-to-PHY) cell flow. In terms of bus accesses, the MC92520 implements a slave interface to the system microprocessor and any in-circuit test equipment.

2.2.1 Ingress Cell Flow

In the ingress direction, the MC92520 receives cells from the FIFO in the PHY. Cell discrimination is performed and uses predefined header field values to recognize unassigned and invalid cells. Unassigned and invalid cell slots can be used to insert OAM and messaging cells into the ingress cell flow. Cell rate decoupling is accomplished by discarding unassigned cells.

For VCCs, the 28-bit VPI/VCI address space (32-bit link/VPI/VCI if multiple physical links are supported) needs to be compressed into a 16-bit ingress connection identifier (ICI). The MC92520 provides two methods for performing VCC address compression to generate the ICI:

- Table lookup based on reduced addressing
- External address compression

For VPCs, the VPI field is used for a lookup into the VP table to obtain the ICI. The ICI is a pointer used to access the context parameters for the current ingress cell from the external context memory. Included in these parameters are cell counters, UPC/NPC traffic descriptor, OAM parameters and switch parameters.

The UPC/NPC mechanism counts the arriving cells and, using a flexible arrangement of traffic enforcement algorithms, admits cells that do not violate the traffic characteristics established for that connection. Violating cells are tallied and can be tagged or discarded (removed from the cell flow).

The OAM parameters are used to control when and how OAM cells are processed and to indicate if the current user cell belongs to a connection selected for a performance monitoring test. If the ingress cell belongs to such a connection, the OAM table in external memory contains the relevant parameters.

Subsequent to the context processing, the ingress cells are transferred to the switch. Optionally, the associated switch context parameters can be added to the cell before the header or placed in the VPI/VCI fields of the header.

2.2.2 Egress Cell Flow

In the egress direction, the MC92520 receives cells from the switch along with any associated parameters. One parameter is the egress connection identifier (ECI), which is used for direct lookup into the context table to obtain the VPI/VCI, cell counters, and OAM flags. If multicast translation is enabled, the multicast identifier (MI) is received from the switch instead of the ECI, and the ECI is obtained from a lookup in the multicast translation table. If enabled, the UPC function is executed. Cells are subject to processing as indicated by the OAM flags. If the egress cell belongs to a connection that has been selected for a performance monitoring test, the OAM Table in external memory contains the relevant parameters.

The egress cell header is generated by inserting the VPI/VCI-field obtained from the address translation table in the (GFC)/VPI/VCI position and modifying the PTI-field, if indicated by the switch or in case of an OAM cell. The cell is then forwarded to the PHY queue. Cell rate decoupling is performed in the egress direction. Optionally, unassigned cells are generated if no cells are available from the switch.

2.2.3 Other Functions

A general 32-bit slave system interface is provided for configuration, control, status monitoring, and insertion and extraction of cells. This interface provides access to the MC92520 registers. The MC92520 also includes a standard JTAG boundary scan architecture test access port (TAP).

2.3 MC92520 Block Diagram

Figure 2-3 is a block diagram of the MC92520. The individual blocks are described in this section.

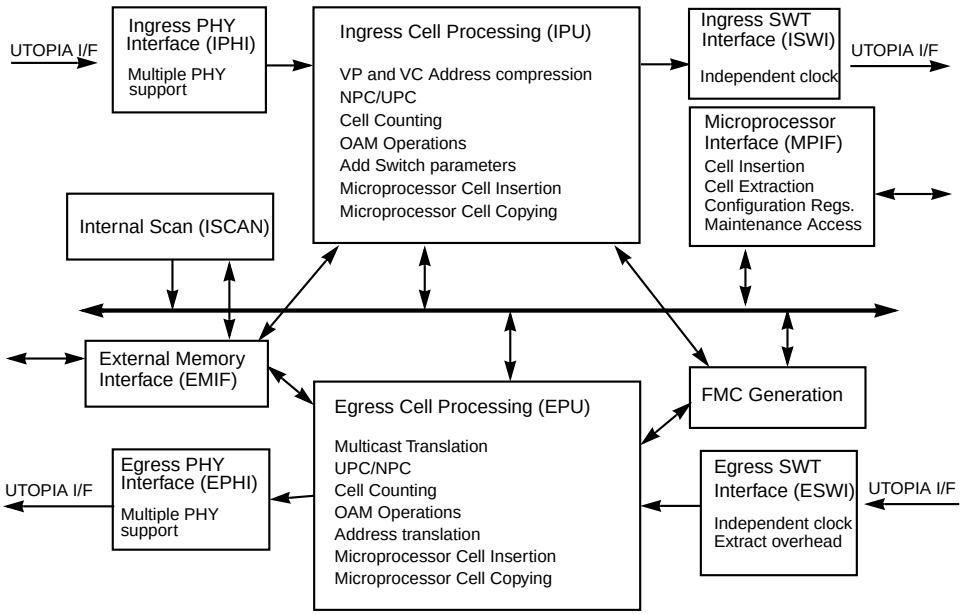


Figure 2-3. MC92520 Block Diagram

2.3.1 Ingress PHY Interface (IPHI)

The ingress PHY interface (IPHI) block contains a cell FIFO. It receives cells on a byte or word basis from the ATM PHY layer using the UTOPIA standard interface and assembles the cells and synchronizes their arrival to the MC92520 cell processing slots. Unassigned and invalid cells (see Table 5-1 and Table 5-2) are removed to provide cell rate decoupling. The MC92520 processes cells at a higher rate than the PHY provides them, so there are always some holes in the cell flow. These can be used either for cell insertion or for maintenance accesses that are initiated by the microprocessor to maintain the external memory content.

2.3.2 Ingress Cell Processing Unit (IPU)

The ingress cell processing unit (IPU) processes cells from the following sources at the rate of one cell per cell processing slot:

- Ingress PHY interface (IPHI)
- Microprocessor interface block (MPIF)

- Internal scan block (ISCAN)
- Forward monitoring cell (FMC) generation block

The IPU performs the following tasks:

- The IPU performs address compression on cells that arrived from the IPHI block in order to associate the cell with context table records in the MC92520 external memory. The address compression function detects inactive cells (cells with no corresponding records in the context table).
- UPC/NPC is performed on a connection basis, or optionally on arbitrary groups of connections. The UPC/NPC function can detect violating cells as dictated by the selected UPC/NPC algorithm. Violating cells are normally tagged or removed from the cell flow, but an option exists to perform the UPC/NPC algorithm for statistical purposes only without modifying or removing the cells.
- Cell insertion is controlled through a leaky bucket mechanism and prioritized by cell type. The highest insertion priority is for PM forward monitoring cells generated by the MC92520, medium insertion priority is for cells from the microprocessor, and the lowest priority is for AIS, RDI, and CC cells generated by the MC92520's internal scan process.
- OAM processing is performed where appropriate. The ingress OAM function records OAM alarm cells (AIS/RDI). OAM processing for user cells involved in a performance monitoring block test involves computing the bit-interleaved parity (BIP) and updating the total user cells (TUC) count. For OAM cells, the processing can include overwriting the values of specific fields and checking or generating the CRC-10 field.
- Switch-specific overhead information is read from the context entry and added to the cell before it is sent on to the switch interface block. Address translation can optionally be performed at this point.
- The IPU removes any OAM cell that has reached its end-point from the cell flow.
- Certain cells can be copied to the microprocessor interface (MPIF) for transfer to the microprocessor.

2.3.3 Ingress Switch Interface (ISWI)

The ISWI block contains a cell FIFO. Cells are received from the IPU. When a full cell has been transferred, the overhead information needed by the switch (as programmed by the user) is extracted from the internal data structure along with the ATM header and payload of the cell. This information is transferred to the switch at the rate of one byte or word per clock cycle.

2.3.4 Egress Switch Interface (ESWI)

The ESWI block contains a cell FIFO. In single-PHY mode of operation, the cell available signal (CLAV) is reported based on the ESWI cell FIFO. In multi-PHY mode of operation, CLAV is reported based on the combination of the ESWI cell FIFO state and the state of the per-PHY FIFOs in the egress PHY interface (EPHI) block.

Data is received from the switch at the rate of one byte or one word per clock cycle. The data structure received from the switch includes overhead routing information in addition to the ATM cell. When a full cell has been transferred, it is transformed into an internal data structure and presented to the EPU for processing.

2.3.5 Egress Cell Processing Unit (EPU)

The egress cell processing unit (EPU) processes cells from the following sources at the rate of one cell per cell processing slot:

- ESWI switch interface block (ESWI)
- Microprocessor interface block (MPIF)
- Internal scan block (ISCAN)
- Forward monitoring cell (FMC) generation block

EPU processing may include the following tasks:

- Multicast translation, if needed. This is the first stage of egress cell processing.
- Optionally, the EPU performs UPC/NPC.
- Cell insertion is controlled through a leaky bucket mechanism and prioritized based on cell type. The highest insertion priority is for PM forward monitoring cells generated by the MC92520, medium insertion priority is for cells from the microprocessor, and the lowest priority is for AIS, RDI, and CC cells generated by the MC92520's internal scan process.
- Where appropriate, the EPU performs OAM processing. The egress OAM function records OAM alarm cells. OAM processing for user cells involved in a performance monitoring block test is limited to computing the bit-interleaved parity (BIP) and updating the total user cells (TUC) count. For OAM cells the processing may include overwriting the values of specific fields and checking or generating the CRC-10 field.
- Address translation is performed to replace the address fields of the ATM cell header with the address of the outgoing link.
- The EPU removes any OAM cell that has reached its end point from the cell flow.
- Certain cells may be copied to the microprocessor interface (MPIF) for transfer to the microprocessor.

2.3.6 Egress PHY Interface (EPHI)

The egress PHY interface (EPHI) block temporarily stores cells processed by the EPU. The block contains cell FIFOs for each multi-PHY port. Based on configuration information, either one per-port FIFO is shared or several per-port FIFOs are used to store cells.

When a sufficient amount of data is stored and the destination PHY has room to accept a cell, the stored data is segmented into bytes or words, and transferred to the physical layer using the UTOPIA standard interface. In single-PHY mode of operation, unassigned or idle cells may be inserted to provide cell rate decoupling.

2.3.7 External Memory Interface (EMIF)

The external memory interface (EMIF) block performs address generation so the MC92520 can access the external memory. It also drives the memory control signals.

2.3.8 Microprocessor Interface (MPIF)

The microprocessor interface (MPIF) block provides for configuration of the MC92520, the transfer of cells between the microprocessor and the MC92520, and the maintenance of the external memory. A standard 32-bit slave interface is provided for glueless connection to the MPC860 (PowerQUICC) and MPC8260 (PowerQUICC II) microprocessors, as well as allowing connection to other processors. Output signals are provided that can serve as request signals for up to two DMA devices to improve system performance.

The cell extraction queue is used to store cells that are directed to the processor. Cells in this queue are transferred first to an internal cell buffer, then they may be read by the processor. Cells to be inserted in the ingress or egress flows are transferred from the processor memory to an internal cell buffer.

2.3.9 Internal Scan (ISCAN)

The internal scan (ISCAN) block scans the external memory for connections on which alarm indication signal (AIS), remote defect indicator (RDI), or continuity check (CC) OAM cells must be inserted. When such a connection is found, the cell is generated and added to the insertion queue for the cell flow in the appropriate direction.

2.3.10 FMC Generation

The forward monitoring cell (FMC) generation block monitors and prioritizes the connections on which FMCs are pending during the course of a performance monitoring block test. When a hole in the cell flow is available, this block requests the insertion of an FMC on the highest-priority connection.

Chapter 3
System Operation

This chapter illustrates the operational aspects of an MC92520 application. Specifically, the chapter outlines the tasks a control processor must perform to set up and operate the MC92520. Operational aspects of attached PHY- and switch-side devices and how they relate to MC92520 interface configuration options and function are described in Section 3.3, “PHY-Side Interface Operation,” and Section 3.4, “Switch-Side Interface Operation.”

3.1 Setup Mode

The MC92520 enters setup mode after reset. This mode is used to configure the MC92520 and to initialize the attached external memory (EM). Because the MC92520 does not use the external memory in setup mode, any bus master on the microprocessor bus has unrestricted access to external memory. In addition, those registers identified as configuration registers (see Section 7.1.5, “Control Registers”) can be written only when the MC92520 is in setup mode. For a detailed description of the reset process that returns the MC92520 to setup mode, see Section 4.1.2, “Reset.”

The following tasks are performed during setup mode:

- Configuring the phase-locked loop (PLL)
- Programming the configuration registers
- Initializing external memory
- Configuring external memory maintenance slots
- Writing the enter operate mode register (EOMR)

3.1.1 Configuring the Phase-Locked Loop (PLL)

Configuration of the MC92520 must start with PLL configuration, unless it is disabled. If the PLL is used, three registers configure and control its function:

- PLL control register (PLLCR) enables PLL operation
- PLL range register (PLLRR) configures the PLL input clock frequency range
- PLL status register (PLLSR) provides current PLL locking status

In addition, any loss of PLL lock is detected and reported through the PLL lost lock (IR[PLL]) interrupt status bit. The possible combinations of PLL register settings and the associated core cell processing bandwidths are shown in Table 3-1.

Table 3-1. PLL Register Configuration Options

PLL Mode	PLLCR [PLLE]	PLLRR [PICR]	ACLK [MHz]	ZCLKs [MHz]	Core Cell Processing Bandwidth [Mbps]
Standard MC92510	0	Don't care	DC - 50	DC - 50	0 - 331.328
Standard MC92520	1	0	25 - 50	50 - 100	331.328 - 662.656
Low Power MC92520	1	1	12.5 - 25	25 - 50	165.664 - 331.328

The PLL setup must be the first configuration activity after the MC92520 comes out of reset. The following steps configure the PLL:

1. After hardware reset the MC92520 PLL is disabled. If the MC92520's target clocking configuration does not require a PLL function, no PLL configuration is necessary and steps 2 – 4 can be skipped. In any case, the current PLL locking status is available by reading the PLLSR.
2. If the MC92520 PLL is required, the PLL input clock range bit (PLLRR[PICR]) must be set to match the provided ACLK frequency range.
3. PLL operation is started by setting the PLL enable bit (PLLCR[PLLE]). This enables the PLL to generate a phase-locked ZCLKO at the required frequency.
4. The PLL will first attempt to lock using an internal feedback path. Once internal lock is achieved, it will then attempt to lock using the external feedback path. The status of the PLL can be seen as it goes through this process. The PLLSR will indicate "not locked, attempting internal feedback lock" (PLLSR[NLI] = 1); then "not locked, attempting external feedback lock" (PLLSR[NLE] = 1); and then, finally "locked" (PLLSR[PLLS] = 1). If a problem occurs, "PLL Lock Time-Out" (PLLSR[LTO] = 1) may occur. This may be due to improper connection of the external feedback path in the user's application.
5. When the PLL lock status bit(PLLSR[PLLS]) is set, lock has been acquired and ZCLKO is validated. Note that it can take up to 30uS + 4096 ACLK cycles for the PLL to lock if PLLRR[PICR] = 0, or 30uS + 2048 ACLK cycles if PLLRR[PICR] = 1.
6. After PLL lock is validated, the processor must issue a software reset by writing to the MC92520 software reset register (SRR). The software reset eliminates any transitory register states that might have been latched while the PLL was changing clock frequencies and phase during the locking.

The PLL can be reconfigured at any time. If the MC92520 is in operate mode and external memory consistency must be preserved, the PLL reconfiguration must be preceded by a software reset. The following describes the reconfiguration process for all possible transitions:

1. No-PLL mode to any-PLL mode: follow the full procedure described above.
2. Any-PPL mode to different-PLL mode: follow steps 2, 4, and 5 of the procedure described above.
3. Any-PLL mode to no-PLL mode: clear the PLL enable bit (PLLCR[PLLE]) to disable the PLL and issue a software reset by writing SRR (Section 4.1.2, “Reset”).

3.1.2 Programming the Configuration Registers

The configuration registers may be modified in setup mode only. (Writing to a configuration register in operate mode may produce unpredictable results.) Therefore, all configuration registers, as described in Section 7.1, “Registers Description” must be written before leaving setup mode.

In addition, the various control registers described in Section 7.1, “Registers Description” may be programmed at this time.

NOTE:

In setup mode, the UTOPIA PHY and switch side interfaces do not accept or transfer ATM cells. This remains true while the MC92520 is in setup mode for all but the egress transmit PHY interface. If the egress PHY configuration register (EPHCR) is configured to generate invalid (idle) cells or unassigned cells, the generation and transmission of these cells starts as soon as the configuration bits are written (and while the MC92520 is in setup mode). For more information, see Section 7.1.6.4, “Egress PHY Configuration Register (EPHCR).”

3.1.3 Initializing External Memory

For most MC92520 applications, a significant amount of MC92520 register and external memory initialization is required to prepare the MC92520 address compression and VCC/VPC context table entries. For a detailed discussion of the configuration options, refer to Chapter 5, “Data Path Operation,” and Chapter 6, “Protocol Support.”

When the MC92520 is in setup mode, the external memory can be accessed directly from the processor interface. If MADD[25] = 1, EMADD[23:2] are driven with the value of MADD[23:2], and the EM interface control pins are driven as necessary. The data is transferred between MDATA and EMDATA. If MADD[24] is set to 1, the MC92520 automatically writes back zero to each word of EM that is read. (That is, a “destructive read” is performed.)

3.1.4 Configuring External Memory Maintenance Slots

One more important aspect of the MC92520 configuration is the allocation of external memory bandwidth for external memory maintenance. These accesses are initiated by the microprocessor while the MC92520 is in operate mode. The specific bandwidth requirement is tightly coupled with each MC92520 application, but a 5 to 10% allocation is sufficient for most cases. As a result, the maintenance period length (MPL) field of the maintenance control register (MACTLR) should be defined with a non-zero value. For more information on maintenance accesses, refer to Section 3.2.2.1, “Maintenance Slot Access.”

3.1.5 Writing the Enter Operate Mode Register (EOMR)

Setup mode is concluded by writing the enter operate mode register (EOMR). Within 1 to 3 cell times, the MC92520 enters operate mode and starts to process ATM cells. The operation mode bit (OM) of the interrupt register (IR) can be used confirm the transition.

NOTE:

Once the MC92520 has left setup mode, any write access to configuration registers can produce indeterminate results and direct access to external memory is no longer possible. Setup mode can be re-entered through a hardware or software reset.

3.2 Operate Mode

After the enter operate mode register (EOMR) has been written by the microprocessor, the MC92520 enters operate mode within 1 to 3 cell times and begins to receive and transmit ATM cells on its PHY and switch interfaces. While in operate mode, the microprocessor must not write the MC92520 configuration registers and has no direct access to the MC92520 external memory.

In operate mode, the MC92520 processes ATM cells, and the microprocessor is concerned with the maintenance aspects of MC92520 external memory, ATM cell insertion and extraction, and the handling of other asynchronous events (errors) reported through the MC92520 microprocessor interrupt mechanism. A detailed description of these activities is provided in the following sections:

- Section 3.2.1, “External Memory Maintenance”
- Section 3.2.2, “External Memory Access”
- Section 3.2.3, “Cell Insertion And Extraction”

The MC92520 stays in operate mode until either a hardware or a software reset occurs. In case of a microprocessor-initiated software reset (by writing the software reset register. SRR), the MC92520 completes any ongoing cell processing to ensure external memory coherence before the actual reset process starts. Upon completion of the reset, the

MC92520 is in setup mode. For a detailed description of the reset process, see Section 4.1.2, “Reset.”

3.2.1 External Memory Maintenance

The process of updating and adding information to, and removing information from, external memory while the MC92520 is in operate mode is external memory maintenance. While this maintenance is initiated by the microprocessor, external memory accesses are made through the MC92520 for the following reasons:

- Sharing access bandwidth and support procedures is efficient, and
- External memory coherence while processing ATM cells is assured.

For a detailed description on how to access external memory through the MC92520 see Section 3.2.2, “External Memory Access.”

External memory maintenance is required to achieve the following MC92520-supported tasks and ATM layer activities:

- PHY link activation and deactivation
  - — On-demand addition and removal of link-specific address compression tables
- VCC/VPC connection setup and removal
  - — On-demand addition and removal of VCC/VPC context and address compression entries
- OAM functions
  - — On-demand reading, resetting, and processing of the flags table
  - — On-demand updating of OAM related VCC/VPC context parameters
- Collection of VCC/VPC related accounting and statistics
  - — On-demand collection of accounting information
  - — On-demand or periodic collection of statistics
- Collection of PHY interface and MC92520 device statistics
  - — On-demand or periodic collection of MC92520 and PHY interface statistics

3.2.1.1 Periodic External Memory Maintenance

The flags table in the MC92520 external memory is used to communicate state information of operation and maintenance (OAM) tasks between the MC92520 and the microprocessor. If an MC92520 application requires support for an alarm indication signal (AIS), a remote defect indicator (RDI), a continuity check, or other OAM functions, the microprocessor regularly reads and resets the flags table. Current ATM OAM standards require failure response times to be on the order of 1 second or less. Therefore, the entire flags table should be covered at least once each second.

All MC92520-maintained cell counters are 32 bits wide and wrap around to zero after their maximum value is reached. Assuming a maximum cell rate of 1,562,500 cps, a full bandwidth cell counter wraps around after 45 minutes. Therefore, depending on the counted information, cell counters should be read at appropriate periodic intervals to prevent loss of information. Note that most counter tables are accessed through pointers in MC92520 control registers rather than in configuration registers. Thus, assuming sufficient external memory is available, most counters can be sampled (by reprogramming a control register) at a given moment and read as system bandwidth allows.

3.2.2 External Memory Access

In operate mode, the external memory is accessed in two ways:

- By using a maintenance slot at programmable intervals to perform up to 64 queued accesses. This method is described in Section 3.2.2.1, “Maintenance Slot Access.”
- By using a single indirect access during each cell processing period. This method is described in detail in Section 3.2.2.2, “Indirect Memory Access.”

3.2.2.1 Maintenance Slot Access

The MC92520 can be configured to periodically provide maintenance slots, during which cells are not processed. Interrupting cell processing is permissible as long as the cell processing rate of the MC92520 is at least as high as the cell rate delivered by the ingress PHY and the egress switch interface. The maximum cell processing rate of the MC92520 is 662.5 Mbps, for example, while the net bandwidth of an STM-4 link is 599.04 Mbps. The MC92520 can reserve about 10% of its bandwidth for maintenance slots without impacting its ability to process cells at the STM-4 link rate. Such breaks in the cell processing are not visible at the PHY and switch interfaces because the delay is absorbed by the MC92520 interface FIFOs.

During a maintenance slot, the MC92520 can process up to 64 queued access requests. Access requests are writes, reads, or destructive reads (reads that clear the read location) previously queued by the microprocessor. The microprocessor does not need to know when a maintenance slot is occurring to access the external memory. Instead, the microprocessor requests access to the external memory maintenance slot access request (EMREQ) FIFO, describing the desired external memory accesses. In addition, selected access requests can be marked such that they start and complete within one maintenance slot, or that their completion is reflected by the setting of both (IR[ASCO]) and the ASCOE bit in the interrupt mask register (IMR). When the next maintenance slot occurs, the requested accesses are performed, the results of any read requests are stored in the external memory maintenance slot access result (EMRSLT) FIFO, and access complete status can be reported to the microprocessor as requested. Later, the microprocessor reads the ASCO bit to see if the access sequence is complete, checks for errors, and retrieves any read results from the EMRSLT FIFO.

3.2.2.1.1 Maintenance Slot Parameters

The duration of a maintenance slot is the same as the duration of a cell slot (one cell processing time). A cell processing time is defined as the period of ZCLKIN multiplied by 64 (the number of clocks used to process each cell). By setting the number of cell slots that occur between maintenance slots, the period, or rate, of maintenance is established. This number of cell slots is referred to as the maintenance period length (MPL). See Section 7.1.5.3, “Maintenance Control Register (MACTLR).”

The link rate determines the maximum number of cell slots the MC92520 needs to process the cells it receives from the PHY layer. An STM-4 link has a link rate up to 1,412,831 cells per second. The number of available MC92520 cell processing slots per second is the inverse of the cell processing time, or the ZCLKIN frequency divided by 64. For example, if the ZCLKIN frequency is 100 MHz, this results in 1,562,500 cell slots per second.

The difference between the total number of cell slots and the number used for processing arriving cells provides free cell slots that are usually used as maintenance slots. However, if the cell insertion rate must be higher than the number of holes in the link rate cell flow, additional free cell slots can be used for processing inserted cells.

The ZCLKIN frequency must be high enough (relative to the link rate) to provide sufficient free slots for both of these purposes. The division of the free slots between the two purposes is determined by the value of the maintenance period length (MPL) field. In our example of ZCLKIN running at 100 MHz, there are 1,562,500 - 1,412,831 = 149,669 free slots per second. If the MPL is programmed as 19 (every 20th slot is a maintenance slot), there are 1,562,500 / 20 = 78,125 maintenance slots provided per second, leaving 149,669 - 78,125 = 71,544 empty slots per second to be used for cell insertion. Alternatively, if the MPL is programmed as 63, there are 1,562,500/64 = 24,414 maintenance slots provided per second, leaving 149,669 - 24,414 = 125,255 empty slots per second for cell insertion.

Appendix B, “Maintenance Slot Calculations,” provides a table of results of the maintenance slot calculations for specific frequencies, as well as an explanation of the equations used for the calculations.

3.2.2.1.2 External Memory Request (EMREQ) FIFO

The external memory request (EMREQ) FIFO queues external memory accesses of the microprocessor until they can be executed during the next periodic maintenance slot. To enable efficient access request confirmation and external memory coherence, the EMREQ FIFO supports the concept of an access request sequence and the concept of an atomic access group.

Access Request Sequence

Accesses can be grouped into sequences, which are primarily used to minimize the frequency of access confirmations exchanged between the MC92520 and the microprocessor. The grouping of requests as sequences lets the programmer determine when the MC92520 informs the microprocessor that queued accesses have been performed.

Any access request implicitly starts an access request sequence. The sequence is completed by explicitly marking the last access request with an end-of-sequence indication.

The access sequence complete bit, IR[ASC], is set when the last access of a sequence has been processed by the MC92520 and any associated read data has been stored in the EMRSLT FIFO. If the access sequence complete enable bit (IMR[ASCE]) is set, a microprocessor interrupt is generated while ASC stays set. It is assumed that the microprocessor attempts to clear ASC by writing a one to acknowledge the reported status. The MC92520 in turn clears ASC and removes the interrupt indication, unless one or more additional completed access sequences are pending. In this case, the microprocessor is required to repeat the described process until all completed access sequences are acknowledged.

In general, any queued sequence of external memory accesses can occur over more than one maintenance slot. This includes the two accesses of a single destructive read. For some types of external memory maintenance tasks, accesses need to be more controlled and limited to a single maintenance slot. The mechanism to achieve this is described below.

Atomic Access Group

The MC92520 supports a group of access requests being labeled with an atomic attribute, indicating that this group must be executed within a single maintenance slot. The ability to perform atomic accesses is essential if accesses distributed over multiple maintenance slots corrupt external memory. All active cell counter increments accumulated during the cell processing period between two maintenance slots will be lost, for example, if reading and clearing of the counter using a destructive read access happens in different maintenance slots.

An atomic access group is started by a start of atomic sequence indication (IR[SOA]) bit, Figure 3-1. It can consist of one or more access requests, and must result in at least two external memory cycles (such as two reads, two writes, or one destructive read). The end of an atomic access group is indicated by an end of atomic sequence (IR[EOA]) bit, Figure 3-1, or an end of sequence indication (IR[EOS]) bit is set (see Section 3.2.2.1.3, “External Memory Read Access Requests”).

An atomic access group must not take more than 32 ZCLK cycles to complete. If an atomic access group is encountered after the first 32 cycles of a maintenance slot, external memory access is suspended till the next maintenance slot starts. Because each read or write access takes one ZCLK cycle, and each destructive read access (read and 0 write back) takes two clock cycles, the following limitations exist: an atomic sequence can consist of up to 32 reads or writes, 16 destructive reads, or any combination of reads/writes/destructive reads that takes 32 clock cycles or less.

3.2.2.1.3 External Memory Read Access Requests

To request a read or destructive read in operate mode, a memory write is performed to external memory space with MADD[25:24] = b11. The address written within this space

(that is, address bits MADD[23:2]) indicates the address to be read or the starting address of consecutive memory locations to be read. The data written in the low-order word (bits 15:0) controls how the read is to be performed. If a non-destructive read or a 32-bit destructive read is requested, this write should be a 32-bit write. If a 16-bit destructive read is performed, this write should be a 16-bit write to the appropriate word (high-order or low-order). This write to memory space with MADD[25:24] = b11 causes the read request to be loaded into the EMREQ FIFO.

A single-read request can specify a single memory address to be read or specify a starting address and a number of additional reads to be performed on the external memory. The memory address increments with each read. Note that this range of reads can occur in one or more maintenance slots, unless this request is part of an atomic access.

Unless it is part of an atomic access, a single destructive read can be split over two maintenance slots, where the read occurs at the end of one maintenance slot and the write to 0 occurs at the start of the next maintenance slot.

All reads, including destructive reads, return a 32-bit read result. In other words, while MADD[1] is significant to identify the memory to be cleared on destructive 16-bit reads, MADD[1] is assumed to be 0 in respect to the returned read result.

Figure 3-1 shows the format of the data field on read requests, destructive read requests, and control-only writes. Control-only writes will be explained later in this chapter. Table 3-2 defines the bits within the control word.

15	14	13	12	11	10				6	5	4	3	2	1	0
SOA	EOA	EOS	TT1	TT0	Reserved					R5	R4	R3	R2	R1	R0

Figure 3-1. External Memory Read Request Control Word

Table 3-2. External Memory Read Request Control Word Bit Descriptions

Bits	Name	Description
15	SOA	1 = Start of atomic access.
14	EOA	1 = End of atomic access. Note that one request can be both the start and end of an atomic access if the request takes more than one clock cycle to perform (such as destructive read, or a range of reads).
13	EOS	1 = End of sequence. When this request is complete, set the IR[ASC] bit, and issue an interrupt if IMR[ASCE] equals 1. Note that EOS also indicates End of Atomic (regardless of the value in the EOA field).
12–11	TT	00: Control-only. Save the SOA, EOA and EOS fields from this request, and use them as the control fields for the next write request. 01: Normal (non-destructive) read 10: 16-bit destructive read. First perform a 32-bit read on this location (A1 is assumed to be 0), then write 0 to the word specified by A23–A1. 11: 32-bit destructive read. First perform a 32-bit read on this location, then write 0 to this location.

Table 3-2. External Memory Read Request Control Word Bit Descriptions (Continued)

Bits	Name	Description
10–6		Reserved, should be cleared
5–0	R	Indicates the number of additional reads to be performed from consecutive memory locations following this memory address. This is allowed for either normal (non-destructive) reads or 32-bit destructive reads (not 16-bit destructive reads). The number of reads (or 32-bit destructive reads) to be performed is 1 + R[5:0], for any value of R[5:0]. These bits are ignored on a Control-only operation.

When a read request is serviced, the data read from external memory is written to the EMRSLT FIFO. This FIFO's output is mapped into the general register space.

3.2.2.1.4 External Memory Write Access Requests

A write access request is performed by writing to external memory space with MADD[25:24] = b10. This write causes the write request to be loaded into the EMREQ FIFO. The memory write is performed as if the write is to occur immediately even though it is going into the EMREQ FIFO. The microprocessor interface signals MADD[23:2], MWSH, MWSL and MDATA[31:0] indicate the address to be written, whether it is a 16- or 32-bit write, and the data to be written.

A write access request uses internal temporary registers as the source of its start of atomic, end of atomic, and end of sequence indicators (IR[SOA,EOA,EOS]). By default, these indicators are cleared. If a write access request is to be marked with any of these three indicators, a control-only operation must *precede* the write request. A control-only operation is a write to any address in the external memory space with MADD[25:24] = b11 (it does not matter which address within this space) in which the TT field is cleared. A write to external memory space with MADD[25:24] and TT cleared is not interpreted as an external memory read access request. It only loads the internal temporary registers with the data from the SOA, EOA, and EOS fields. These internal registers become the source of the start-of-atomic, end-of-atomic, and end-of-sequence indicators for the next external memory write access request. After the next write access request, these internal temporary registers are cleared.

Note that external memory space with MADD[25:24] = b10 is used directly for access only in setup mode. In operate mode, writing this memory space is always interpreted as an external memory write request to be loaded into the EMREQ FIFO.

3.2.2.1.5 External Memory Access Errors and Interrupts

The EMREQ FIFO must not be allowed to overflow. If the access request error enable bit (IMR[ARQEE]) and the access request error bit (IR[ARQE]) are set, then if more than 64 pending requests are written to this FIFO, an interrupt is generated.

If the access read error enable bit (IMR[ARDEE]) and the access read error bit (IR[ARDE]) are set, an interrupt is generated when reading the EMRSLT FIFO while it is empty.

The EMRSLT FIFO cannot overflow—it can contain 64 read results. Once it is full, neither read requests nor write requests are processed until some of the results are read and the EMRSLT FIFO is no longer full. Therefore, allowing the EMRSLT FIFO to become full can cause the EMREQ FIFO to overflow. If the access result FIFO full enable bit (IMR[ARSFE]) has been set, then once the EMRSLT FIFO is full, the access result FIFO full bit (IR[ARSF]) is set and an interrupt is generated.

If the atomic access not complete error enable bit (IMR[ANCEE]) is set, then if an atomic access does not complete within one maintenance slot (which can happen only if it takes more than 32 clock cycles), the atomic access not complete error bit (IR[ANCE]) is set and an interrupt is generated.

If the access sequence complete enable bit (IMR[ASCE]) is set, if a requested write or read that has been marked end of sequence is performed, the access sequence complete bit (IR[ASC]) is set and an interrupt is generated.

3.2.2.2 Indirect Memory Access

In addition to the maintenance slot access method of accessing external memory, the MC92520 can access memory through a single indirect access during each cell processing period while in operate mode. The indirect access is not performed during maintenance; rather, it is performed using two registers: the indirect external memory access address register (IAAR**)** and the indirect external memory access data register (IADR**)**.

3.2.2.2.1 Write Access

To write to the external memory, the processor performs these functions:

- Polls the indirect external memory access busy bit (IAAR[IAB]) to verify that it can write IAAR and IADR.
- Writes the data into the IADR register, and the address, size and direction = 0 into the indirect external memory access address field (IAA) and the indirect external memory access size bit (IAW) and the indirect external memory access DIR bit (IAD) of the IAAR register.
- Writing to the IAAR register triggers the MC92520 to wait for a dedicated clock, and write the data to external memory using the given address and data.
- Once the MC92520 finishes writing, it clears the indirect external memory access busy bit (IAAR[IAB]).

3.2.2.2.2 Read Access

To read from the external memory, the processor performs the following functions:

- Polls the indirect external memory access busy bit (IAAR[IAB]) to verify that it can write IAAR.
- Writes the address, size, and direction = 1 into the indirect external memory access address (IAA) field, the indirect external memory access size (IAW) bit and the indirect external memory access direction DIR bit (IAD) of the IAAR register.

- Writing to IAAR triggers the MC92520 to wait for a dedicated clock, and to read the data from external memory using the given address and write the data into IADR.
- Once the data is written into IADR, the MC92520 clears the indirect external memory access busy bit (IAAR[IAB]).
- Reads IADR. The address space covered by this interface includes all the non-destructive external memory access.

NOTE:

An indirect write access to an external memory space that can be written by the MC92520 is not recommended. For example, do not perform an indirect write access to a flag table record of an active connection; use the maintenance cell slot for this purpose.

Table 3-3 summarizes the indirect access fields:

Table 3-3. Indirect Access Fields

IAAR [IAD]	IAAR [IAW]	LSB of IAAR [IAA]	MPCONR [DO]	Function
0	0	x	x	Write IADR[31:00] to external memory word bits [31:00]
0	1	0	0	Write IADR[31:16] to external memory word [31:16]
0	1	0	1	Write IADR[15:00] to external memory word [15:00]
0	1	1	0	Write IADR[15:00] to external memory word [15:00]
0	1	1	1	Write IADR[31:16] to external memory word [31:16]
1	x	x	x	Read external memory word bits [31:00] to IADR[31:00]

3.2.3 Cell Insertion And Extraction

Two arrays of registers transfer ATM cells between the MC92520 and the microprocessor: the cell insertion registers for transfers to the MC92520 and the cell extraction registers for transfers from the MC92520. The structure of the inserted and extracted cells is provided in the tables in Section 7.3.1, “Inserted Cell Structure” and Section 7.3.2, “Extracted Cell Structure.” The cell payload, contained in registers 4 through 15 of each array, is considered a collection of bytes. Therefore, the byte order within the 32-bit registers is different if low-endian ordering is being used on the microprocessor bus. The overhead information, on the other hand, consists of 32-bit fields and is not affected by the byte order of the bus. Setting the data order (DO) bit in the microprocessor configuration register (MPCONR) enables byte-swapping of the payload. (See Table 3-3.)

3.2.3.1 Cell Insertion

The processor inserts cells into the ingress or egress cell flows indirectly by using the cell insertion registers. This set of sixteen registers holds one cell together with the overhead information necessary to insert it properly into a cell flow. The registers can be written in

any order. If a series of similar cells is being inserted, causing the register values to be the same, then it is not necessary to write to the unchanged registers.

The cell insertion registers are mapped into two distinct address spaces. See Section 7.1.1, “Cell Insertion Registers (CIR0–CIR15)”. The address spaces differ in the number of registers that can be accessed and in the identity of the trigger register. Writing the trigger register causes the cell to be inserted, so this must always be the last register written. Once the trigger register has been written, the CIRs cannot be written again with the next cell until MC92520 has set the cell insertion queue empty IR[CIQE] bit. The trigger register in the cell insertion address space is CIR15. In the alternative cell insertion address space the trigger register is ACIR1. The alternative address space is useful for inserting cells whose header and payload are generated by the MC92520 and who do not have to be inserted by the user. For example, an AIS cell can be inserted by writing to only ACIR0 and ACIR1.

Additionally, the MC92520 provides support for transferring cells using a DMA device without need for a processor interrupt. Each of the DMA request output signals can be programmed to function as $\overline{\text{MCIREQ}}$, which is asserted in operate mode whenever the cell insertion register array is available to be written. Writing to CIR14 in the cell insertion address space or to ACIR0 in the alternative cell insertion address space causes $\overline{\text{MCIREQ}}$ to be deasserted until the array is available again. Writing to CIR15 or ACIR1 also deasserts $\overline{\text{MCIREQ}}$. See Section 4.5.2.2, “Cell Insertion with DMA Support” for details. Both of the cell insertion register address spaces contain a number of don’t-care bits in their addresses (see Figure 7-2). This effectively maps the registers repeatedly in this address space and allows the transfer of a large buffer of data using a single buffer descriptor in the DMA device.

Table 3-4 and Table 3-5 display the conventions used by Motorola and Intel to order bytes in cell insertion registers.

Table 3-4. Inserted Cell Structure (Motorola Byte-Ordering Convention)

CIR0	Cell Descriptor				ACIR0
CIR1	Connection Descriptor				ACIR1
CIR2	Unused				
CIR3	ATM cell header (VPI,VCI,PTI,CLP)				
CIR4	ATM octet 6	ATM octet 7	ATM octet 8	ATM octet 9	
CIR5	ATM octet 10	ATM octet 11	ATM octet 12	ATM octet 13	
CIR6	ATM octet 14	ATM octet 15	ATM octet 16	ATM octet 17	
CIR7	ATM octet 18	ATM octet 19	ATM octet 20	ATM octet 21	
CIR8	ATM octet 22	ATM octet 23	ATM octet 24	ATM octet 25	
CIR9	ATM octet 26	ATM octet 27	ATM octet 28	ATM octet 29	
CIR10	ATM octet 30	ATM octet 31	ATM octet 32	ATM octet 33	
CIR11	ATM octet 34	ATM octet 35	ATM octet 36	ATM octet 37	
CIR12	ATM octet 38	ATM octet 39	ATM octet 40	ATM octet 41	
CIR13	ATM octet 42	ATM octet 43	ATM octet 44	ATM octet 45	
CIR14	ATM octet 46	ATM octet 47	ATM octet 48	ATM octet 49	
CIR15	ATM octet 50	ATM octet 51	ATM octet 52	ATM octet 53	

Table 3-5. Inserted Cell Structure (Intel Byte-Ordering Convention)

CIR0	Cell Descriptor				ACIR0
CIR1	Connection Descriptor				ACIR1
CIR2	Unused				
CIR3	ATM cell header (VPI,VCI,PTI,CLP)				
CIR4	ATM octet 9	ATM octet 8	ATM octet 7	ATM octet 6	
CIR5	ATM octet 13	ATM octet 12	ATM octet 11	ATM octet 10	
CIR6	ATM octet 17	ATM octet 16	ATM octet 15	ATM octet 14	
CIR7	ATM octet 21	ATM octet 20	ATM octet 19	ATM octet 18	
CIR8	ATM octet 25	ATM octet 24	ATM octet 23	ATM octet 22	
CIR9	ATM octet 29	ATM octet 28	ATM octet 27	ATM octet 26	
CIR10	ATM octet 33	ATM octet 32	ATM octet 31	ATM octet 30	
CIR11	ATM octet 37	ATM octet 36	ATM octet 35	ATM octet 34	
CIR12	ATM octet 41	ATM octet 40	ATM octet 39	ATM octet 38	
CIR13	ATM octet 45	ATM octet 44	ATM octet 43	ATM octet 42	
CIR14	ATM octet 49	ATM octet 48	ATM octet 47	ATM octet 46	
CIR15	ATM octet 53	ATM octet 52	ATM octet 51	ATM octet 50	

3.2.3.2 Cell Extraction

Cell extraction uses a combination of a cell extraction queue, which is a 16-cell FIFO, and a set of cell extraction registers. The following paragraphs include a detailed description of these components used in cell extraction.

3.2.3.2.1 Cell Extraction Queue

The cell extraction queue holds cells that are copied from the ingress or egress cell flows that are awaiting their transfer to the microprocessor. The queue is necessary because both the ingress and egress processing units can copy cells during the same cell processing slot for a number of consecutive cell slots, while the microprocessor cannot read the cells at that rate.

The cell extraction queue is a 16-cell FIFO as shown in Figure 3-2. There are two programmable thresholds defined in the microprocessor control register (MPCTLR):

- The interrupt threshold allows the microprocessor to wait until a number of cells have accumulated and then to read the cells in a burst. The cell extraction queue interrupt threshold bit (IR[CEQI]) is asserted when the cell extraction queue is filled to the interrupt threshold.
- The low priority threshold defines the full limit to the cell extraction queue. The cell extraction queue low priority threshold bit (IR[CEQL]) is asserted when the limit is reached.

When the cell extraction queue exceeds the limit of the low priority threshold, the MC92520 places only high-priority cells on the queue. Low-priority cells that are copied from the ingress or egress cell flows are discarded. If the cell extraction queue is full, the cell extraction queue full bit (IR[CEQF]) is asserted, and no cells are added to the queue.

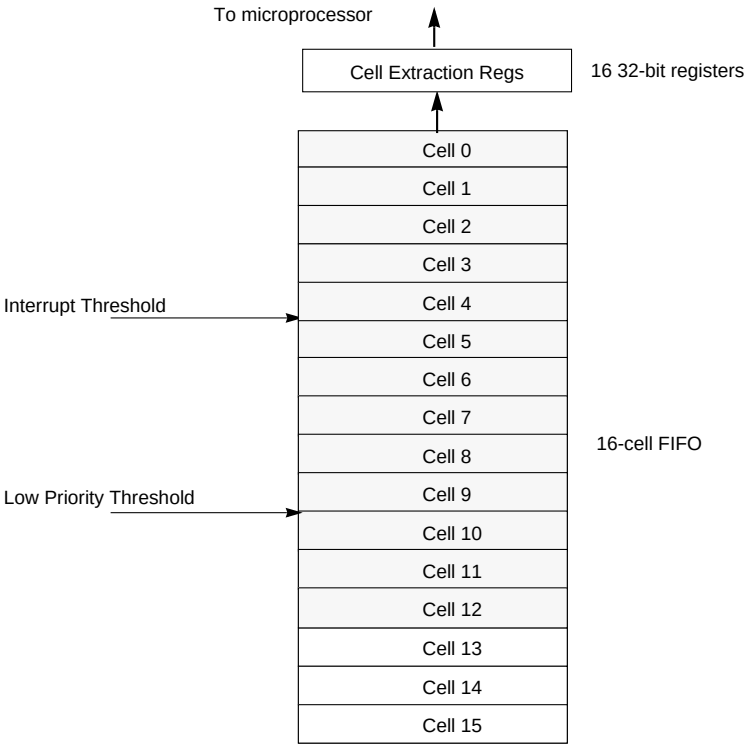


Figure 3-2. Cell Extraction Queue

The microprocessor interface filters the cells to be placed in the cell extraction queue, as illustrated in Figure 3-3. When a cell is to be copied, the cell indication is checked. If the extraction reason field (RSN) indicates Copy_All or Copy_OAM, the cell is filtered according to the indication cell name field (ICN) using the cell extraction queue filtering register 1 (CEQFR1) and, if the queue is above the low-priority threshold, the cell extraction queue priority register 1 (CEQPR1). Otherwise, the cell is filtered using cell extraction queue priority register 0 (CEQPR0) and, if above the low-priority threshold, cell extraction queue priority register 0 (CEQPR0). Additionally, if either of the indication message flag (IMSF) or GFC reason (GFCR) bits is set, the corresponding bits of CEQFR0 and CEQPR0 are used.

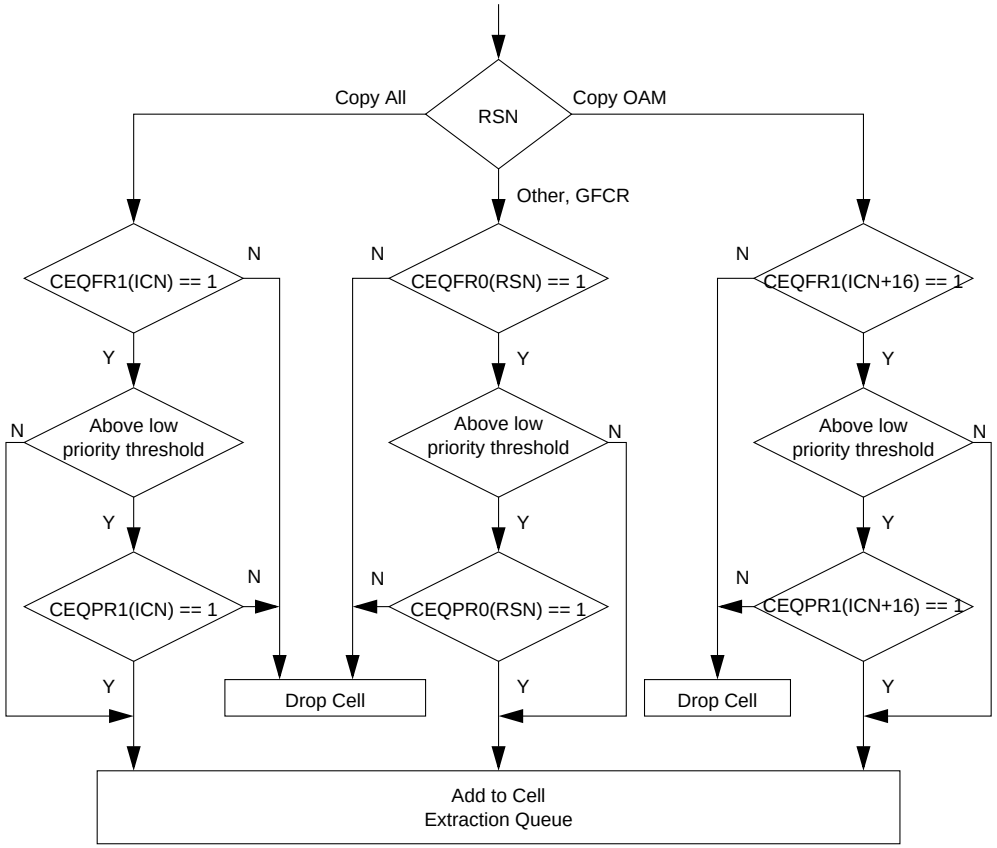


Figure 3-3. Cell Extraction Queue Filtering

3.2.3.2.2 Cell Extraction Registers

Cells that are copied from the ingress or egress cell flows are transferred to the microprocessor by using the cell extraction registers. This set of 16 registers holds one cell together with the overhead information that describes its source and cell type. With the exception of CER15, the registers can be read in any order. Registers whose contents are not of interest in the specific cell need not be read.

The cell extraction registers are mapped into a single address space. See Section 7.1.2, “Cell Extraction Registers (CER0–CER15).” Reading the trigger register, CER15, frees the array for the MC92520 to write the next cell, so this must always be the last register to be read. After the trigger register has been read, the CERs should not be read again for the next cell until the cell extraction queue ready interrupt enable bit (IR[CEQRE]) has been set by the MC92520.

Additionally, the MC92520 provides support for transferring cells using a DMA device without need for a processor interrupt. Each DMA request line can be programmed to function as $\overline{\text{MCOREQ}}$, which is asserted whenever the cell extraction register array contains a cell to be read. Reading from CER14 causes $\overline{\text{MCOREQ}}$ to be deasserted until the next cell is ready. Reading from CER15 also de-asserts $\overline{\text{MCOREQ}}$. See Section 4.5.2.1, “Cell Extraction with DMA Support” for details.

The MC92520 is designed to support back to back cell extraction; if the next cell is ready when CER14 of the current cell is read, $\overline{\text{MCOREQ}}$ is not deasserted, thus allowing continuous DMA operation. The cell extraction register address space contains a number of don’t care bits in its addresses (see Figure 7-3). This effectively maps the registers repeatedly in this address space and allows the transfer of a large buffer of data using a single buffer descriptor in the DMA device.

Table 3-6. Extracted Cell Structure (Motorola Byte-Ordering Convention)

CER0	Cell Indication			
CER1	Connection Indication			
CER2	Time Stamp			
CER3	ATM cell header (VPI,VCI,PTI,CLP)			
CER4	ATM octet 6	ATM octet 7	ATM octet 8	ATM octet 9
CER5	ATM octet 10	ATM octet 11	ATM octet 12	ATM octet 13
CER6	ATM octet 14	ATM octet 15	ATM octet 16	ATM octet 17
CER7	ATM octet 18	ATM octet 19	ATM octet 20	ATM octet 21
CER8	ATM octet 22	ATM octet 23	ATM octet 24	ATM octet 25
CER9	ATM octet 26	ATM octet 27	ATM octet 28	ATM octet 29
CER10	ATM octet 30	ATM octet 31	ATM octet 32	ATM octet 33
CER11	ATM octet 34	ATM octet 35	ATM octet 36	ATM octet 37
CER12	ATM octet 38	ATM octet 39	ATM octet 40	ATM octet 41
CER13	ATM octet 42	ATM octet 43	ATM octet 44	ATM octet 45
CER14	ATM octet 46	ATM octet 47	ATM octet 48	ATM octet 49
CER15	ATM octet 50	ATM octet 51	ATM octet 52	ATM octet 53

Table 3-6 and Table 3-7 display the conventions used by Motorola and Intel to order bytes in cell extraction registers.

Table 3-7. Extracted Cell Structure (Intel Byte-Ordering Convention)

CER0	Cell Indication			
CER1	Connection Indication			
CER2	Timestamp			
CER3	ATM cell header (VPI,VCI,PTI,CLP)			
CER4	ATM octet 9	ATM octet 8	ATM octet 7	ATM octet 6
CER5	ATM octet 13	ATM octet 12	ATM octet 11	ATM octet 10
CER6	ATM octet 17	ATM octet 16	ATM octet 15	ATM octet 14
CER7	ATM octet 21	ATM octet 20	ATM octet 19	ATM octet 18
CER8	ATM octet 25	ATM octet 24	ATM octet 23	ATM octet 22
CER9	ATM octet 29	ATM octet 28	ATM octet 27	ATM octet 26
CER10	ATM octet 33	ATM octet 32	ATM octet 31	ATM octet 30
CER11	ATM octet 37	ATM octet 36	ATM octet 35	ATM octet 34
CER12	ATM octet 41	ATM octet 40	ATM octet 39	ATM octet 38
CER13	ATM octet 45	ATM octet 44	ATM octet 43	ATM octet 42
CER14	ATM octet 49	ATM octet 48	ATM octet 47	ATM octet 46
CER15	ATM octet 53	ATM octet 52	ATM octet 51	ATM octet 50

3.3 PHY-Side Interface Operation

The physical layer interface of ATM devices is well documented and standardized. The MC92520 represents no exception and provides a fully compliant implementation of the UTOPIA level 1 and level 2 standards ([11], [12]). The following sections describe operational aspects and how other MC92520 features relate to specific PHY-side interface configurations. Table 3-5 displays the dependencies between PHY-side and switch-side configuration options. The MC92520-specific signal names and a description of the bus protocols are presented in Section 4.2, “PHY Interfaces.”

3.3.1 PHY-Side Interface Considerations

In interfacing with the PHY side of an MC92520 a number of factors should be considered, such as the number of connecting devices, the use of FIFOs for each PHY port, cell FIFO depth, and how to prevent system failures that spring from failing PHY devices

3.3.1.1 Multiple PHY Devices

One important aspect of an MC92520 application is its associated component and connectivity cost. Besides the MC92520 itself, there is a controlling microprocessor, other system peripherals and processor memory, as well as the MC92520 context memory, one or more PHY devices, and more memory or an optional CAM for address translation. This is a significant amount of infrastructure to support a 600 Mbps line card or access mux. In

practice, applications will share this infrastructure between as many PHY ports as possible to reduce per-port cost.

More is not generally better though, especially if it is certain that not all features are used by all applications. For example, per-PHY FIFOs combined with support for many PHYs increases the device size, which negatively impacts the device cost for all applications, whether they need to support many PHYs or not. In the case of the MC92520, direct support of 16 PHY ports was considered a practical compromise. This number of ports allows the user to connect either 4 OC-3, 6 TAXI, 12 OC-1, 14 DS-3, or 16 other devices with bandwidths less than 40 Mbps to a single MC92520. If more than 16 PHY ports need to be supported, multiple MC92520s can be used. For more information see Section 3.4.1.2, “Supporting a Multi-PHY Interface on the Switch-Side.”

3.3.1.2 Per-PHY FIFOs

Support for multiple PHY devices is often associated with per-PHY-port FIFOs. Per-PHY-port FIFOs are useful for a number of reasons:

1. they may reduce the requirement for a switch-side cell scheduler to be very precise,
2. they help prevent blocking if a single PHY-port fails, and
3. they support the use of a multi-PHY interface on the switch-side.

Per-PHY-port FIFOs have two drawbacks though: 1) they tend to increase the total cell FIFO depth and therefore impact device-induced cell delay variation (CDV), and 2) they introduce a requirement for a cell transfer scheduler such that the highest bandwidth PHY is selected for cell transfer when more than one PHY has cells waiting in its associated FIFO.

The MC92520 supports both single FIFO and multiple, per-PHY FIFO configurations on the egress PHY-side interface. If the per-PHY FIFO configuration is selected, the transfer scheduler is driven via the configuration of per-PHY cell transfer priorities.

3.3.1.3 Total Cell FIFO Depth

ATM-layer devices in switch or access mux applications work within a framework of three clock domains in the cell transfer path:

1. The physical cell transfer clock domain,
2. The ATM-layer cell processing clock domain, and
3. The clock domain of the switch fabric or access mux.

Cell transfer between these three clock domains is typically achieved through cell buffer FIFOs. Under optimal conditions, transition from one domain to the next causes no more than 1 cell-time delay, and 1 cell-time CDV. In practice though, the amount of device-induced CDV depends on how much of the total cell FIFO depth is *not* needed to perform the transfers between the clock domains.

If the application is not sensitive to CDV, tight control of the total FIFO depth may not be an issue. But if it is, keeping the total FIFO depth low is typically more difficult for applications that need multiple, per-PHY FIFOs. In any case, the MC92520 supports configurable FIFO depths for both single and multiple, per-PHY FIFO configurations.

3.3.1.4 Head-Of-Line Blocking

Any application involving more than one PHY device must pay specific attention to device failure detection and recovery. The typical tree structure of such applications is particularly susceptible to single-point failures causing overall system failures. Most of these failures are introduced by what is traditionally call head-of-line (HOL) blocking.

HOL blocking occurs in systems with one input queue and more than one output queue. A failure on an output queue may cause it to fill. When the first content (in our case a cell that is at the ‘head of the line’) that is destined to the full output queue is encountered on the input queue, the processing of the input queue will block, causing it to fill up. While this happens, processing on the other output queues will continue until they are all drained. Finally, when the input queue is completely full and all working output queues are drained, all system processing stops.

The MC92520 provides HOL blocking prevention and reporting functions that can be used to circumvent the scenario outlined above and report failures as soon as they are detectable.

3.3.2 PHY-Side Interface Configuration Options

The MC92520 can be configured to interface with one or many PHY devices, one or many FIFOs, and to use head-of-line blocking detection or prevention.

3.3.2.1 Single- and Multi-PHY Support

The MC92520 can be programmed to support a UTOPIA level 1 and level 2 compliant single-PHY, or a UTOPIA level 2 compliant multi-PHY interface. The MC92520 supports up to 16 PHY devices if their provisioned aggregate bandwidth does not exceed the MC92520’s bandwidth. Any single PHY does not operate at more than 50% of the total provisioned bandwidth when used in conjuction with a switch operating with an optimal scheduler. Additional PHY devices can be supported via multiple MC92520s. For more information see Section 3.3.2.1.2, “Multi-PHY Configuration.”

3.3.2.1.1 Single-PHY Configuration

Single-PHY configuration is the MC92520’s default and is indicated by a cleared UTOPIA multi-PHY bit in the egress and ingress PHY configuration registers (EPHCR[EUM] and IPHCR[IUM]). In single-PHY configurations, the UTOPIA level 2 PHY polling and select address lines of the MC92520 (TXADDR and RXADDR) are not used, and the handshake signals ($\overline{\text{TXFULL}}$/TXCLAV and $\overline{\text{RXEMPTY}}$/RXCLAV) are sampled in accordance with the selected handshake method. The MC92520 supports both octet- or cell-level handshake, and the selection is made by setting the operation mode bit in the egress and ingress PHY

configuration registers (EPHCR[EPOM] and IPHCR[IPOM]) accordingly. Octet mode is supported in single-PHY configuration only.

3.3.2.1.2 Multi-PHY Configuration

Multi-PHY configuration on the PHY side is selected by setting the UTOPIA multi-PHY bit in the egress and ingress PHY configuration registers (EPHCR[EUM] and IPHCR[IUM]). In multi-PHY configurations, the MC92520 TXADDR and RXADDR signals are used to poll cell available status and select individual PHY devices for cell transfer. If the MC92520's PHY interface is operated in multi-FIFO mode, the cell transfer is scheduled via a per-PHY transfer priority configured in the egress link registers (ELNK*n*[EPRI]). The cell transfer handshake method must be configured for cell-level by setting the operation mode bit in the egress and ingress PHY configuration registers (EPHCR[EPOM] and IPHCR[IPOM]) for all multi-PHY configurations.

3.3.2.1.3 Multi-PHY Operation

Besides the support for a PHY-side multi-PHY interface, the MC92520 provides the following features to appropriately process cells by an application that supports multiple PHY devices:

- Cell counters: In addition to the connection cell counters, link cell counters are provided for both the ingress and egress cell flows. Separate counts are maintained according to USER/OAM cell classification and the value of the CLP bit. Also, cells that arrive in the ingress cell flow that do not belong to an active connection are counted separately in an inactive cell counter.
- Address compression: The MC92520 has a set of 16 ingress link registers so that the address compression parameters, including the address compression method, can be defined for each link independently. The ingress processing unit uses the link number received from the PHY device as an index to the link registers.
- Address translation: Each connection supported by the MC92520 belongs to a specific link, so the link number can be considered a parameter of the connection. The link number is stored in the connection address word of the context parameters table. The egress processing unit reads the link number from the external memory together with the new address for the cell. This number identifies the PHY device to which the cell must be transferred.
- Cell extraction queue: The link number is provided together with each cell copied to the cell extraction queue. (Although the link number can usually be determined because it is a parameter of the connection and the connection ID is provided, this is not the case with inactive cells.) Cells copied from the ingress cell flow use the link number provided by the PHY device, and cells copied from the egress cell flow use the link number read from the connection address word.

- Switch-side multi-PHY interface: The MC92520 supports an egress switch-side multi-PHY interface. This interface conveys PHY-side cell available status to a switch-fabric or access mux on a per-link basis. This capability is provided through per-link cell buffer FIFOs that are enabled and configured in 16 egress link registers.
- Multicast translation: When performing multicast in an environment in which a single MC92520 is supporting multiple PHY devices, a single cell arriving at the switch may be presented to the egress MC92520 multiple times, once for each physical link. There are two possibilities for performing multicast translation in this environment. One is for the switch interface block to provide a unique multicast identifier for each link to which the cell is transmitted. In this case the MC92520 performs multicast translation in the same manner as for a single link. Another possibility is for the switch to leave the same multicast identifier in all copies of the cell. In this case the multicast translation uses a unique region of the multicast translation table for each link. In order to do so, the switch must provide the link number in the MTTS field of the cell overhead information. See Section 5.2.2 “Multicast Identifier Translation” for details.
- HOL blocking prevention: The MC92520 supports individually configurable HOL blocking detection and prevention on up to 16 physical links. Configuration is performed through the 16 egress link registers. Cell transfer delay monitoring, automatic FIFO flushing upon HOL blocking detection, aggregate and per-link HOL blocking status, and an optional microprocessor interrupt are supported.
- Egress PHY cell transfer priority: The MC92520 supports a configurable per-PHY transfer priority when per-PHY FIFOs are enabled. Configuration is performed through the 16 egress link registers. The egress PHY transfer priority is used to optimally arrange cell transfer schedules when more than one per-PHY FIFO contains cells ready for transfer (without egress cell transfer priorities, disparate transfer rates in a set of multi-PHYs would cause FIFO underruns in the PHYs with the higher transfer rate).

3.3.2.2 Single- and Multi-FIFO Support

The MC92520 can be programmed to use single or multiple per-PHY FIFOs on the PHY-side. The multi-FIFO configuration should not be confused with the multi-PHY configuration described above. While multi-PHY configuration deals with the support for multiple PHY devices, multi-FIFO configuration determines whether and how many per-PHY cell buffers are used for temporary storage of processed ATM cells. Both single- and multi-FIFO configurations can be used to support multi-PHY applications.

3.3.2.2.1 Single-FIFO Configuration

Single-FIFO configuration is the MC92520’s default and is indicated by a cleared egress PHY multiple FIFOs (EPHCR[EPMF]) bit. In single-FIFO configurations, the FIFO depth can be set to either 2 or 4 cells deep by programming the egress PHY interface FIFO control (EPHCR[EPFC]) bit accordingly.

A 2 cell FIFO depth is recommended for applications with PHY devices that have sufficient internal cell buffering capability (3 or more cell buffers). In all other cases, the FIFO depth should be configured for 4 cells.

3.3.2.2.2 Multi-FIFO Configuration

Setting the egress PHY multiple FIFOs (EPHCR[EPMF]) bit selects multi-FIFO operation. In multi-FIFO configurations, the depth of each FIFO can be individually programmed to be either 1, 2, or 3 cells deep by setting the FIFO depth field in the assigned egress link register (ELNK*n*[FDPTH]). Also, in a multi-FIFO configuration, the multi-PHY cell transfer priority can be programmed to 1 of 16 levels (0 being highest and 15 being lowest priority) in the egress link register (ELNK*n*[EPRI]).

Selecting an appropriate FIFO depth depends on its purpose and is described in detail in Section 3.3.2.3, “HOL Blocking Detection and Prevention” and Section 3.4.2.2, “Switch-Side Multi-PHY Interface Support.”

3.3.2.2.3 Guidelines

The MC92520 single-FIFO configuration and operation is fully backward compatible with the MC9250*x* line of products, but cannot be used with HOL blocking detection and prevention or with a switch-side multi-PHY interface. A multi-FIFO configuration can utilize these MC92520 features.

Generally, it is not recommended to select a single-FIFO configuration for multi-PHY applications because the failure of a single PHY may cause overall system failure without HOL blocking prevention (HOL blocking detection and prevention on a single-FIFO with cells destined to different PHYs will corrupt cell flows to all PHYs.) Nevertheless, if a single-FIFO configuration is selected for multi-PHY applications, the switch-side cell scheduling may need to make sure that cells destined to the PHY devices will not temporarily block the shared FIFO. On the up-side, single-FIFO configurations offer the tightest cell buffer utilization and therefore result in a low cell delay variation introduced by the MC92520.

3.3.2.3 HOL Blocking Detection and Prevention

HOL blocking detection and prevention is crucial for applications supporting multiple PHY devices. Without HOL blocking prevention, a single PHY device failure may cause the complete application to fail.

The MC92520 must be configured to use multiple, per-PHY FIFOs and an associated cell transfer time-out threshold. In addition, the application software can program each individual FIFO to be automatically flushed if an HOL blocking problem is detected. An optional microprocessor interrupt may be generated, and a status register indicates which PHY ports had an HOL blocking problem.

3.3.2.3.1 Configuration

HOL blocking prevention and reporting is enabled by setting the egress PHY HOL blocking prevention (EPHCR[EPHP]) bit and the egress PHY multiple FIFOs (EPHCR[EPMF] bit. For each multi-PHY port, the associated egress link register (ELNK*n*) must be programmed as follows:

1. The PHY port must be enabled by setting the link enable (ELNK*n*[LE]) bit.
2. The automatic flush enable (ELNK*n*[FE]) bit should be set according to the specific application needs. In most cases it would be set. If automatic flushing is turned off (the FE bit is clear), HOL blocking detection is useful as an MC92520-to-PHY cell transfer delay monitoring tool. If automatic flushing is enabled, all cells in the FIFO detecting HOL blocking will be discarded. Note that the FIFO will not be disabled and storing of subsequent cells continues without interruption.
3. A maximum acceptable cell transfer time-out must be configured by setting the time-out (ELNK*n*[TMO]) field. The TMO field is defined in multiples of ATMC cell processing periods (1 ATMC cell processing period is 64 ZCLKs). The cell transfer time-out can be programmed from 1 to 4096 periods (0 implies a value of 4096). For example, in a typical clocking configuration, each port of a quad OC-3 multi-PHY device would accept a new cell every 280 ZCLKs. Thus a value of 5 (resulting from 5 * 64 = 320 ZCLKs) would be the tightest possible time-out value for each of the 4 ports. Any smaller value would result in false time-outs.
4. The associated FIFO depth must be configured by setting the per-link FIFO depth (ELNK*n*[FDPTH]) field. The FIFO depth field should be set to a value as small as possible to minimize cell delay variation, but large enough to hold cells while a HOL blocking time-out cannot be detected. For example, let's assume a switch-side device would send cells destined to the same quad OC-3 multi-PHY port no less than 4 cell times apart and that the TMO field would be set at 5. For this case a second cell may arrive during the 320 ZCLK time-out period of an earlier cell. Therefore, the FIFO depth should be set to 2 to support full bandwidth cell processing by the MC92520.

3.3.2.3.2 Operation

The occurrence of HOL blocking is indicated through the HOL blocking detection (HOLD) status bit in the MC92520 interrupt register (IR). The setting of this bit indicates that HOL blocking in respect to one or more of the configured cell transfer time-out values has been detected. An optional microprocessor interrupt can be generated by the MC92520, if the associated HOL blocking detection enable (HOLDE) bit in the interrupt mask register (IMR) is set.

The HOLD status bit described above represents the result of or-ing up to 16 per-PHY-port status bits. The HOL blocking status of each PHY port can be accessed through the HOL blocking status register (HOLDSR). Bit 0 of the register is associated with PHY port 0, bit 1 with PHY port 1, and so on. All HOLDSR bits are sticky bits, that is, once a HOL

blocking status is detected, the associated bit stays set until it is explicitly reset (For more information on sticky status bits see Section 7.1.4.1, “Interrupt Register (IR).”

The aggregate HOLD status is formed by and-ing the per-PHY-port HOLD status bits with their associated link enable bits and or-ing all results. This mechanism allows a pending HOL blocking status (or interrupt) to be cleared by either clearing the LE bit in the ELNK*n* register, or by resetting the HOLD*n* status bit in HOLDSR for the offending PHY port.

3.3.2.3.3 Guidelines

If the egress switch-side interface is configured for multi-PHY operation, the per-PHY FIFOs must provide sufficient cell buffers to enable the MC92520 internal three stage cell processing pipeline to run at the appropriate per-PHY bandwidth. For more information see Section 3.4.2.2, “Switch-Side Multi-PHY Interface Support.”

3.4 Switch-Side Interface Operation

While the interface to the physical layer is well documented and standardized, there is a fair amount of variability in the interfaces and philosophies applied to link switch fabrics or access muxes to an ATM-layer device like the MC92520. The ATMC designers chose the UTOPIA standard interface to reap the benefits of re-using a well-known and proven design. But this selection just defines a framework. There are many ways of designing switch fabrics or access muxes, and, consequently, there are different ways of how these devices operate and use the UTOPIA interface of the MC92520. The following sections describe these principles and how the MC92520 fits into designs that apply them. Table 3-9 displays the dependencies between PHY-side and switch-side configuration options.

3.4.1 Switch-Side Interface Considerations

In interfacing with the switch side of an MC92520 a number of factors should be considered, such as a strategy for scheduling cell transfers from the switch-side device to the MC92520 and failure detection and prevention.

3.4.1.1 Cell Scheduling

Any device attached to the egress switch-side interface of the MC92520 employs a cell scheduling strategy to select cells for transfer to the MC92520. The cell available signal from the ATMC (STXCLAV) will always be used to control the transfer. While some scheduling methods merely use STXCLAV as a facility to detect system failures, others use it to actually drive cell selection and transfer. The former method is usually described as an open-loop cell scheduling mechanism and the latter as a closed-loop mechanism.

Open-loop scheduling does not rely on the STXCLAV signal to dispatch available cells to the interface block that transfers the cell to the MC92520. This dispatch function is independent of the STXCLAV signal and can be as simple as a periodic decision to transfer, or as complex as an independently running per-multi-PHY-port rate shaping function,

which selects cells from a number of queues based on port cell rates and per-queue cell availability. In some cases, the attached device may not even have a cell output buffer, thus relying completely on the MC92520's ability to accept a cell at predetermined intervals. In all cases, the open-loop scheduler knows whether a specific cell can be transferred. The STXCLAV signal is checked before the transfer, but if it indicates that the transfer cannot occur (within some period), it is interpreted as an indication that the MC92520 or any of its attached PHY devices has failed. Open-loop scheduling typically requires a minimum of functionality in the switch fabric or queueing device as long as the MC92520 is used in a single-PHY application.

Closed-loop cell scheduling, on the other hand, uses the STXCLAV signal to dispatch available cells to the interface block that transfers the cell to the MC92520 (In a switch-side multi-PHY configuration, STXCLAV is reportable for all PHY devices attached to the MC92520). If polled frequently enough, the changes of the STXCLAV signal state reflect the PHY's transfer bandwidth. Therefore, devices employing closed-loop cell scheduling may not need the capability to shape cell rates for individual multi-PHY port flows. On the other hand, per-multi-PHY-port cell queueing is probably necessary to minimize delay, and other functionality may be needed to keep cell delay variations low in the presence of additional cell queueing.

3.4.1.2 Supporting a Multi-PHY Interface on the Switch-Side

An egress, switch-side, multi-PHY interface supports a number of new ATMC applications. The two most important ones are the following:

- Applications that require a glueless, switch-side, multi-PHY interface to one or more ATMCs, each handling one or more multi-PHY ports, and
- Multi-PHY applications that use closed-loop cell scheduling in the attached switch fabric or switch-side queueing device.

The availability of a multi-PHY interface allows the first type of application to configure cost-effective line cards or traffic concentrators with up to 128 PHY ports using two or more ATMCs on a single UTOPIA bus. Both single and multiplexed status polling via one or more CLAV signals can be supported. For more information, see [12], page 30. Note that supporting 32 PHY ports per CLAV signal requires the use of the STXAVALID signal (an extension of the UTOPIA level 2 standard). In these applications it is essential that the PHY ports attached to a specific ATMC can be flexibly mapped into the total PHY port addressing space.

In the second type of application, ATM switch designers use the egress, switch-side, multi-PHY interface to convey PHY-side queue depth status to a switch-side device, which in turn uses this information to trigger the transfer of cells that are destined to the multi-PHY port. Designers cannot do so with a single-PHY interface because it only conveys whether there is any space in the queue, not whether previously sent cells destined to a multi-PHY port have been processed and transferred through the PHY port.

This feature is attractive because it frees the attached device from the requirement to shape each aggregated cell stream destined to a multi-PHY port. Otherwise, shaping on a per-PHY port basis is necessary, because shaping or policing on a VCC or VPC basis does not assure that the combined (peak) cell rate destined to a PHY port is always less than or equal to the PHY port bandwidth. VCC and VPC shaping/policing only assures that the average cell rate is always less than or equal to the rate the line is provisioned with. In practice provisioning below the PHY bandwidth is sometimes used to address the problem, but the disadvantage is under-utilizing the multi-PHY port bandwidth. If a multi-PHY interface is available on the switch-side, the FIFO empty status is conveyed via the CLAV signal on a per multi-PHY port basis. Therefore the switch-side device can now use the status to de-queue cells at the full multi-PHY port bandwidth.

In addition, a switch-side multi-PHY interface provides cell flow information that could be utilized to design a more dynamic, flow-sensitive VCC/VPC cell scheduling mechanism. For example, an ATMC maintenance slot or the insertion of OAM cells by the ATMC will cause a delay of the CLAV signal for the switch-side multi-PHY port. If the switch fabric is using CLAV to select a cell from a number of VCC/VPC candidates, it can now pick the connection as late as possible and select a connection with the tightest cell delay variation (CDV) requirements, meaning that a switch fabric designer has another input to better meet VCC/VPC-specific QOS guarantees.

But an egress, switch-side, multi-PHY interface also has drawbacks. While it provides additional information to the switch-side, it may also add cell buffering elasticity due to the per-PHY separation of cell FIFOs. Even per-PHY-port FIFO configuration cannot side-step this issue because the per-PHY-port FIFO allocation can only happen in steps of full cell buffers. For example, a FIFO depth of 4 would be sufficient to maintain full bandwidth cell transfers on a single PHY port. But if the bandwidth is distributed over 16 PHY ports, 16 single cell buffer FIFOs are needed. Because the UTOPIA polling protocol does not convey how long ago the CLAV status was reached, and how many cell buffers in its per-PHY-port FIFO are unused, the switch-side device typically needs some relative PHY-rate based information to service the individual PHY ports with an appropriate priority and frequency. If the application does not deal with CDV-sensitive traffic or if all PHY ports utilize the same bandwidth, a simple round-robin mechanism is sufficient.

3.4.1.3 Failure Detection and Recovery

While HOL blocking detection and prevention is a PHY-side feature, its use is tightly connected to the architecture of the switch-side device transferring cells. HOL blocking prevention is relevant for both open-loop and closed-loop cell scheduling applications.

3.4.2 Switch-Side Interface Configuration Options

The following describes MC92520 configuration options relating to the different uses of the switch-side interface. Sometimes PHY-side configurations are necessary to achieve the intended behavior and also are described below.

3.4.2.1 Switch-Side Single-PHY Interface Support

The switch-side, single-PHY interface can be used to support applications with single or multiple PHY devices connected to the MC92520. This interface is applicable for both open-loop and closed-loop cell scheduling in applications supporting a single PHY device. If multiple PHY devices need to be supported, this interface can only be used for open-loop cell scheduling systems because no per-PHY status is conveyed to the switch-side.

3.4.2.1.1 Configuration

Switch-side, single-PHY interface configuration is the MC92520's default and is indicated by a cleared egress switch UTOPIA multi-PHY (ESWCR[ESUM]) configuration bit. In this configuration, the egress, switch-side FIFO depth can be set to either 4 or 6 cell buffers by programming the egress switch FIFO control (ESWCR[ESFC]) bit accordingly. The ingress, switch-side FIFO depth is not configurable and is set to 4 cell buffers.

3.4.2.1.2 Guidelines

A 4 cell FIFO depth on the egress side is only recommended for applications with tight CDV requirements in the presence of switch-side devices that can delay cell transfers, which can occurwhenever the MC92520 suspends normal cell processing to perform external memory maintenance or cell insertion. In all other cases, the FIFO depth should be configured for 6 cell buffers.

Supporting applications with multiple PHY devices through a switch-side single-PHY interface requires that the switch-side device carefully controls the spacing of cells destined to the same PHY port. Ideally, the spacing between cells destined to a common PHY port should never fall below the value associated with the provisioned cell transfer rate of the port. Applications violating this requirement expose themselves to somewhat unpredictable transfer delays when a cell is temporarily blocking all egress cell processing due to a FIFO full condition. This may be less of an issue if the provisioned cell rate of a PHY device is lower than the PHY's cell transfer bandwidth. Also, the problem can be addressed by increasing the usable FIFO depth within the PHY device itself.

The MC92520's switch-side, single-PHY configuration and operation is fully backward compatible with the MC92500 line of products.

3.4.2.2 Switch-Side Multi-PHY Interface Support

The egress, switch-side, multi-PHY interface can be used to support applications with single or multiple PHY devices connected to the MC92520. This interface is applicable for both open-loop and closed-loop cell scheduling in applications. Typically, the interface requires a more complex switch-side device and is usually selected for applications supporting multiple PHY devices or closed-loop cell scheduling.

3.4.2.2.1 Configuration

The egress switch-side multi-PHY interface configuration is enabled by setting both the egress switch UTOPIA multi-PHY (ESWCR[ESUM]) bit and the egress PHY multiple FIFOs (EPHCR[EPMF]) bit.

NOTE:

If only the ESUM bit is set, switch-side multi-PHY mode does not operate correctly.

For each used multi-PHY port, the associated egress link register (ELNK*n*) must be programmed as follows:

1. The PHY port must be enabled by setting the link enable (ELNK*n*[LE]) bit.
2. The FIFO depth must be configured by setting the FIFO depth (ELNK*n*[FDPTH] field.
3. The PHY port cell transfer priority should be set in the (ELNK*n*[EPRI]) when disparate PHY rates are configured.

In addition to the FDPTH configuration requirements described in Section 3.3.2.3, “HOL Blocking Detection and Prevention,” the FIFOs must be set sufficiently large to support a given per-PHY link bandwidth in the presence of an MC92520 internal three stage cell processing pipeline. For a list of recommended FDPTH field values see Table 3-8 below.

Table 3-8. Link Bandwidth-Dependent FDPTH Settings

FDPTH	Target Link Bandwidth
1	1/4 or less of full bandwidth
3	1/4 to 1/2 of full bandwidth

The configuration of the position and range of PHY ports to which the ATMC responds during polling is controlled through a configuration register described in Section 7.1.6.7, “Egress Switch Interface Configuration Register 1 (ESWCR1).” The content of the egress switch multi-PHY address (ESWCR1[ESMA]) field determines what switch-side PHY address is associated with PHY port 0 on the PHY-side. The content of the egress switch multi-PHY mask (ESWCR1[ESMM]) field determines what bits of the switch-side PHY address are used to identify PHY ports attached on the PHY-side. The ESMA and ESMM fields combined with the LE bit of ELNK*n* registers are used to enable multi-PHY poll and select transactions as illustrated by the following pseudo code:

```
if (((STXADDR & ~ESCR2.ESMM) == ESCR2.ESMA) &&
    (ELNK[STXADDR & ESCR2.ESMM].LE)
        report CLAV
    else
        tri-state CLAV
```

In words, if the PHY address and-ed with the negated address mask matches the base address, and if the PHY address and-ed with the address mask results in an enabled PHY port, the ATMC will respond by driving CLAV to an appropriate value. In all other cases, the ATMC will tri-state CLAV.

The programming of the priority of each PHY link is accomplished in the ELNK register using the EPRI bits. The highest priority is 0x0 and the lowest is 0xf. This register defaults to all links having a priority of 0x0 (highest: all PHYs having equal priority).

As an example, in a system in which fifteen PHYs each use 5% of the OC-12 rate and one PHY uses 25% of the total bandwidth of the OC-12 bandwidth, egress priority may be configured as follows, assuming the PHY at 25% of the OC-12 bandwidth is link 0.

```
ELNK0[EPRI] = 0x0

ELNK1[EPRI] through ELNK15[EPRI] = 0xF
```

Another application example: when the PHY rates are a linear function, use of linear priorities will force the links to be serviced in a sequence relating to the priority.

In this example, the programming of the ELNK*x* registers are:

```
ELNK0[EPRI] = 0x0 (Highest Priority)
ELNK1[EPRI] = 0x1 (next highest)
ELNK2[EPRI] = 0x2
ELNK3[EPRI] = 0x3
ELNK4[EPRI] = 0x4
ELNK5[EPRI] = 0x5
ELNK6[EPRI] = 0x6
ELNK7[EPRI] = 0x7
ELNK8[EPRI] = 0x8
ELNK9[EPRI] = 0x9
ELNK10[EPRI] = 0xA
ELNK11[EPRI] = 0xB
ELNK12[EPRI] = 0xC
ELNK13[EPRI] = 0xD
ELNK14[EPRI] = 0xE
ELNK15[EPRI] = 0xF (Lowest Priority)
```

3.4.2.2.2 Operation

Taking a simplistic view, switch-side multi-PHY operation merely presents the PHY-side STXCLAV status on the switch-side interface. In practice, such an implementation would severely limit the maximum bandwidth per PHY port (in the ATMC's case to about 1/8 of its full bandwidth). In order to minimize restrictions, the ATMC provides configurable, per-PHY cell FIFOs on the PHY-side and a shared 16-cell FIFO on the switch-side.

The STXCLAV status reported for each polled multi-PHY port is derived by the ATMC from several pieces of information:

1. Is there space in the PHY-side FIFO?
2. How many cells are currently queued for the target PHY on the switch-side?
3. How many cell transfers have been committed via STXCLAVs for other PHYs?
4. Is there additional room in the switch-side FIFO?

The combination of these checks supports operation up to the maximum MC92520 cell processing bandwidth with PHYs of different (disparate) line rates and minimal per-PHY rate limitations.

To minimize PHY rate use limitations, both the FIFO depth and the cell transfer priority must be set to appropriate values matching each multi-PHY port's target bandwidth. Failure to do so will cause a reduction in the maximum achievable cell rate on a given PHY port.

3.4.2.2.3 Egress FIFO Depth

The depth of the egress PHY FIFO is controlled using two registers. Which of these registers will be used depends on the mode of operation: whether there is a single FIFO or a FIFO per PHY.

When the egress FIFO per PHY mode is enabled (EPHCR[EPMF] = 1), the switch will transfer cells to the egress FIFOs on a per-link basis. The depth of each FIFO is separately controlled per PHY in the appropriate register ELNK*n*. (See section Section 7.1.5.15, "Egress Link Registers (ELNK0–ELNK15)" for programming detail.) A FIFO depth of 1, 2, or 3 cells may be selected. Note that the FIFO depth actually includes a "hidden" cell for an additional cell which may occur due to a cell insertion. In FIFO per PHY operation mode, the egress switch continuously monitors the status of the egress FIFO depth for each PHY.

Operation of the egress PHY block in single FIFO mode (EPHCR[EPMF] = 0), uses a single FIFO. The depth of this FIFO is controlled by EPHCR[EPFC] bit, which selects a FIFO depth of either 4 or 2 total cells. (See section Section 7.1.6.4, "Egress PHY Configuration Register (EPHCR)" for programming detail.) The FIFO depth in this case is the total cell depth including cells which are inserted by the ATMC core. Operation of the switch and the egress PHY block in single FIFO mode (EPHCR[EPMF] = 0, ESWCR0[ESUM] = 0) will not cause a FIFO overflow because the switch receives information on the status of the single FIFO in use on the egress PHY block.

Operation of the switch in single-PHY mode with the egress PHY side configured for FIFO per PHY mode is not allowed.

3.4.2.2.4 Guidelines

The fact that there is a 16-cell FIFO on the switch-side does *not* impact CDV or represent a shared extension of the PHY-side FIFOs. STXCLAV is strictly a result of the PHY-side FIFO including cells processed, queued, and committed via STXCLAV. The switch-side FIFO only allows the MC92520 to commit on all ports while the transfer is performed at an independent clock rate. For example, the switch-side device may transfer cells at twice the MC92520's internal cell processing speed. Because the MC92520 plays the role of a UTOPIA slave device, multi-PHY cell transfers cannot be suspended once STXCLAV was reported. Therefore, the MC92520 must have sufficient cell buffer space on the switch-side to receive all committed cells under any clocking scenario.

Table 3-9 shows the relationships between PHY-side and switch-side configuration options:

Table 3-9. Interface Configuration Dependencies

		Single FIFO	Multi FIFO	HOL Blocking Prevention	Open Loop Scheduling	Closed Loop Scheduling
PHY-Side	Single-PHY Interface	Yes	No	No	Yes	Yes
	Multi-PHY Interface	Yes[2]	Yes[3]	Yes[3]	Yes	Yes[1,3]
Egress Switch-Side	Single-PHY Interface	Yes	No	No	Yes	Yes[4]
	Multi-PHY Interface	No	Yes	Yes[1]	Yes[1]	Yes[1]

1 Requires multi-FIFO configuration

2 Requires open-loop scheduling, limited to switch-side single-PHY applications and applications without HOL blocking prevention

3 Requires switch-side multi-PHY configuration

4 Limited to PHY-side single-PHY applications

Chapter 4
External Interfaces

The description of the MC92520 external interfaces is structured in the following groups:

- Clocks and reset
- PHY interfaces
- Switch interfaces
- External memory interface
- Microprocessor interface

The following sections provide detailed descriptions for each of these interface groups. Figure 4-1 is a legend of the conventions used in the timing diagrams.

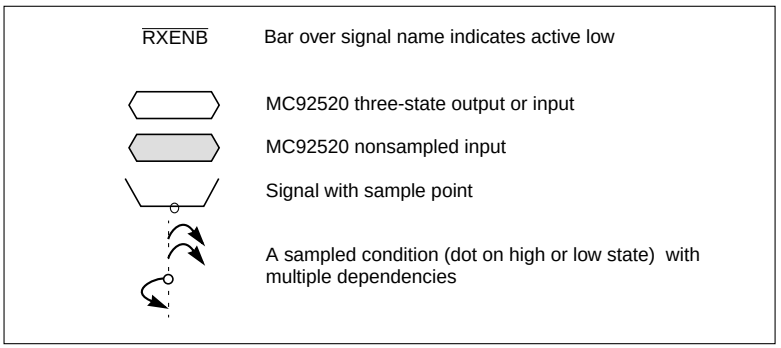


Figure 4-1. Timing Diagram Legend

4.1 Clocks and Reset

The following section describes the clock and reset operations of the MC92520.

4.1.1 Clocks

The MC92520 requires five input clock signals (ACLK, ZCLKIN, SRXCLK, STXCLK, MCLK), and provides three output clock signals (ZCLKO, ZCLKO_2, EACCLK). The following describes the individual signals and how they relate to each other. The clock configuration as well as how the clocks are connected to the MC92520 internal PLL is shown in Figure 4-2.

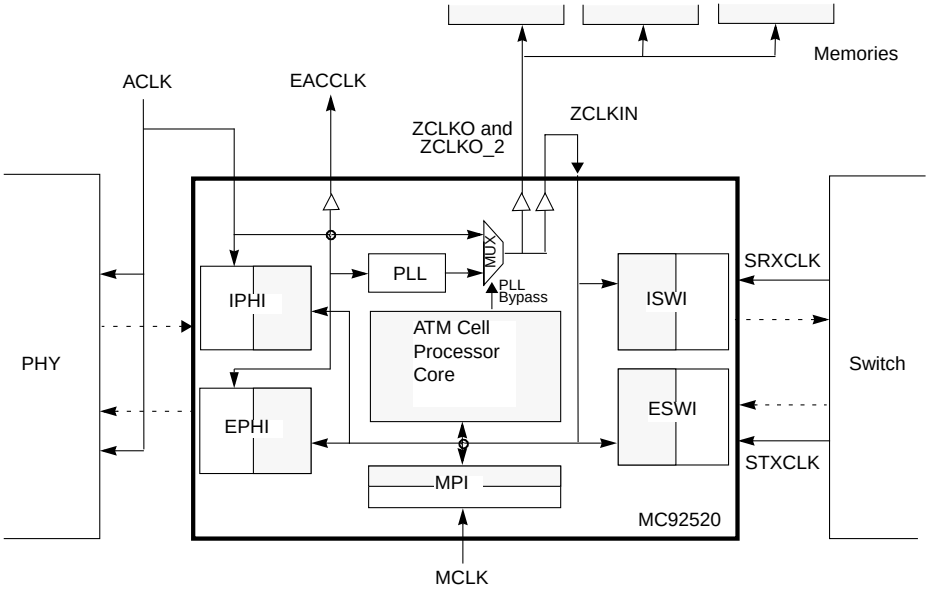


Figure 4-2. MC92520 Clock Configuration

4.1.1.1 Datapath Clocks (ACLK, SRXCLK, and STXCLK)

The ATM layer reference clock (ACLK) is shared by both the PHY devices and the MC92520, that is, the UTOPIA standard PHY RxClk and TxClk signals connect with the MC92520 ACLK to a common clock source.

The switch-side receive and transmit clock signals (SRXCLK and STXCLK) are usually connected to a common clock source. Two separate input pins are provided to support applications where different switch-side transmit and receive cell sizes require different transfer clock rates to meet bandwidth requirements.

4.1.1.2 External Memory and Cell Processing Clocks (ZCLKO, ZCLKO_2, and ZCLKIN)

To achieve the necessary cell processing bandwidth and support power saving configurations, the MC92520 has an internal phase-lock loop (PLL) that can be bypassed or used in two different operational modes. A list of the possible configuration options and how they relate to the usable cell processing bandwidth at the UTOPIA interfaces and in the MC92520 cell processor is provided in Table 4-1 below.

Table 4-1. PLL Configuration Options

PLL Mode	PLL Enable	Clock Range	ACLK [MHz]	ZCLKs [MHz]	Usable Bandwidth [Mbps]	
					Cell Processor	UTOPIA Interface
Standard MC92510	No	N/A	DC–50	DC–50	0–300	0–300
Standard MC92520	Yes	High	25–50	50–100	300–600	306–612
Low Power MC92520	Yes	Low	12.5–25	25–50	150–300	153–306

Note that when a 16-bit UTOPIA interface is used, the bandwidth at the interface is slightly higher than the associated MC92520 cell processing bandwidth owing to the transfer of one additional overhead byte in the UDF word. A description of the PLL configuration process is provided in Section 3.1.1, “Configuring the Phase-Locked Loop (PLL).”

The clock resulting from the PLL configuration is provided by the MC92520 as an output on the ZCLKO and ZCLKO_2 pins, and is used to clock external memory (see Section 4.4, “External Memory Interface”). One of these two outputs (they are identical in function and driver type) is used to clock all of the external memory devices (ZBT RAMs). The other is fed back to pin ZCLKIN. The ZCLKO_2 output can be disabled by setting PLLRR[Z2D] = 1. This should be done to save power if ZCLKO_2 is not used. However both ZCLKO_2 and ZCLKO should be used in most, if not all, applications.

The ZCLKIN pin is used both as the feedback path for the internal PLL and to clock the ATMC cell processor core. Because ZCLKIN is used as the feedback path for the internal PLL, it is important that the clock signal to this input be a clean, glitch-free signal. Care must be taken to drive it from an otherwise-unloaded ZCLKO or ZCLKO_2 pin. The printed circuit board trace should be properly terminated. An R-C termination to ground at the ZCLKIN pin is recommended.

The MC92520 uses 64 ZCLKIN clock cycles to perform the processing of one ATM cell in each direction. Thus the peak cell processing bandwidth is 100,000,000/64 = 1,562,000 cps or 662.5 Mbps. In practice, approximately 600 Mbps are available for cell processing because 5 to 10% of the bandwidth needs to be reserved for external memory maintenance.

4.1.1.3 External Address Compression Clock (EACCLK)

EACCLK is a buffered output of the ACLK signal, intended as the clock to the external address compression CAM match port. It is retimed to provide the required setup/hold timing relationship between the CAM clock and the CAM start match input (SM), which is driven by the MC92520 output $\overline{\text{EACSM}}$.

4.1.1.4 Microprocessor Clock (MCLK)

MCLK is the microprocessor bus clock input and is used in the microprocessor bus interface block of the MC92520. MCLK is asynchronous to ACLK and ZCLKIN. MCLK must not be less than half the rate of ZCLKIN and not greater than twice the rate of ZCLKIN.

4.1.2 Reset

The MC92520 is reset in either of two ways:

- Hardware reset by asserting the ATMC power-up reset ($\overline{\text{ARST}}$) pin, or
- Software reset by writing to the software reset register (SRR).

In case of a hardware reset, the $\overline{\text{ARST}}$ signal must be asserted for at least 10 ACLK or 10 MCLK cycles, whichever cycle is longer. After negation of the $\overline{\text{ARST}}$ signal, the reset process takes 200 ACLK or 200 MCLK cycles to complete, depending on whichever is longer. In contrast to the software reset described below, a hardware reset effects the MC92520 PLL; and after the reset the PLL is disabled.

A microprocessor can initiate an MC92520 reset by writing the software reset register (SRR) at any time, except while a previous reset is still in progress. In case of a software reset, the MC92520 begins the reset process only after the results of any ongoing ATM cell processing have been written back to external memory. This wait for the completion of memory updating assures a coherent external memory state, and the application software is not forced to re-initialize external memory. In addition, a software reset does not affect the MC92520 PLL configuration and operation.

During any reset, the MC92520 registers and attached external memory must not be accessed: this avoids unpredictable initialization results. In addition, the $\overline{\text{MREQ0}}$ and $\overline{\text{MREQ1}}$ signals are configured as $\overline{\text{MCIREQ}}$ and $\overline{\text{MCOREQ}}$ DMA requests, respectively. The following describes the status of the DMA request signals during and after reset:

- $\overline{\text{MCIREQ}}$—negated (pulled high) during and after reset (it is asserted upon entering operate mode).
- $\overline{\text{MCOREQ}}$—negated (pulled high) during and after reset until a cell is available for extraction (after entering operate mode).

Furthermore, the receive PHY interface is disabled by keeping the $\overline{\text{RXENB}}$ output signal negated, and no cells are provided to the ingress switch interface. At the conclusion of any reset, all registers (except PLL registers) are loaded with their default content, and the MC92520 is in setup mode. A hardware reset also affects PLL operation: PLL registers are loaded with their default values and and the PLL is put in bypass mode. A software reset has no effect on the PLL.

4.2 PHY Interfaces

The MC92520 PHY receive and transmit interfaces are UTOPIA level 1 and UTOPIA level 2 compliant. Both PHY interfaces act as ATM layer (UTOPIA master) devices and can be independently programmed to do the following:

- Transfer data with a wide (16-bit) data path or a standard (8-bit) data path
- Operate as a single-PHY or a multi-PHY interface
- Work with octet-, word-, or cell-level handshakes (limited to a single-PHY interface)

In addition, the MC92520 supports the use of UTOPIA level 1 compliant PHY devices as UTOPIA level 2 multi-PHY ports in combination with external hardware.

All configuration options are programmed through the ingress and egress PHY configuration registers described in Section 7.1.6.3 “Ingress PHY Configuration Register (IPHCR)” and Section 7.1.6.4 “Egress PHY Configuration Register (EPHCR),” respectively.

All interface signals are synchronous to the MC92520 ACLK signal. Output signals from the MC92520 are updated following the rising edge of ACLK, and input signals to the MC92520 are sampled at the rising edge of ACLK.

4.2.1 Single-PHY Receive Interface (Ingress)

The PHY-side receive interface of the MC92520 is configured for single-PHY operation if the ingress UTOPIA multi-PHY (IPHCR[IUM]) bit is cleared. The single-PHY receive interface supports octet-, word-, and cell-level handshakes, and the ingress PHY operation mode (IPOM) bit determines the selected handshake method. Similarly, the value of the ingress PHY wide data path (IPWD) bit determines whether a word (16-bit) or an octet (8-bit) is transferred. If the MC92520 is configured to receive octets, the state of RXDATA[15:8] is ignored and the associated pins should be tied to ground.

The signals involved in the single-PHY receive interface are shown in Figure 4-3.

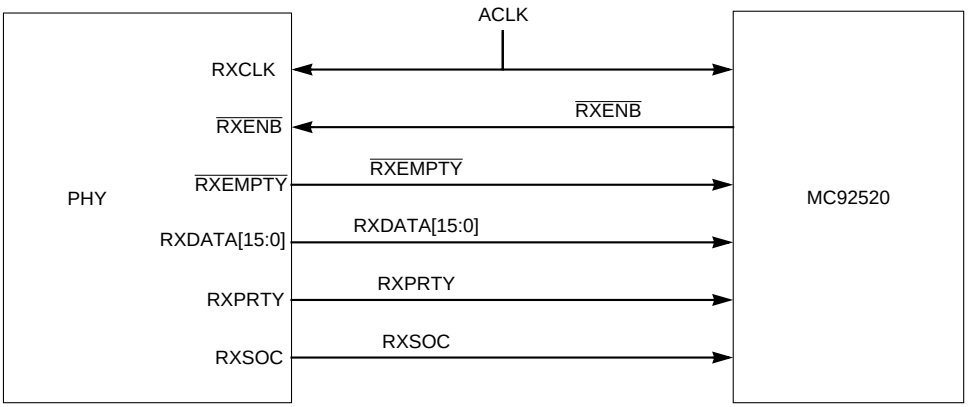


Figure 4-3. MC92520 Single-PHY Receive Interface

The MC92520 requests data transfers in the receive direction by asserting $\overline{\text{RXENB}}$. Because the MC92520 processes entire cells, it requests data on a cell basis and negates $\overline{\text{RXENB}}$ only at the end of a cell transfer if it cannot receive another cell. When the PHY detects $\overline{\text{RXENB}}$ negated, the values that it drives on $\overline{\text{RXEMPTY}}$, RXDATA, RXPRTY, and RXSOC during the following clock cycle are don't cares. Figure 4-4 shows the negation of $\overline{\text{RXENB}}$ at the end of a cell transferred with an 8-bit data bus when the MC92520 is unable to receive another cell. $\overline{\text{RXENB}}$ is negated after the next to last octet of the cell has been received and is not asserted again until the MC92520 is able to receive another entire cell.

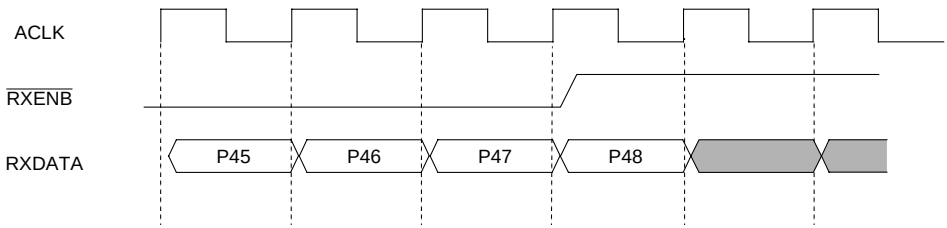


Figure 4-4. MC92520 Receive Timing—End of Cell

The PHY asserts RXSOC together with the first octet or word of each cell in order to synchronize the MC92520 to the beginning of a new cell. RXPRTY contains the odd parity over RXDATA. Dependent on the settings of the IPHCR[IPWD, IPPR], parity is checked for a word-wide data path (RXDATA[15:0]) or an octet-wide data path (RXDATA[7:0]). RXPRTY is not shown in the timing diagrams because its timing is identical to that of RXDATA. For a discussion of parity checking at the ingress receive interface, see Section 5.1.1 "Assembling Cells."

4.2.1.1 Octet- or Word-Level Handshake

When an octet- or word-level handshake is used, $\overline{\text{RXEMPTY}}$ negated indicates that valid data is being presented during the current clock cycle. $\overline{\text{RXEMPTY}}$ must be valid if $\overline{\text{RXENB}}$ was asserted on the previous clock cycle. The MC92520 samples $\overline{\text{RXEMPTY}}$, RXDATA, RXPRTY, and RXSOC at the rising edge of ACLK if $\overline{\text{RXENB}}$ was asserted at the previous rising edge. If $\overline{\text{RXEMPTY}}$ is negated, the values sampled from RXDATA, RXPRTY, and RXSOC are valid. Otherwise, these values are discarded. Figure 4-5 shows the start of a cell when using the octet- or word-level handshake with an 8-bit wide data bus.

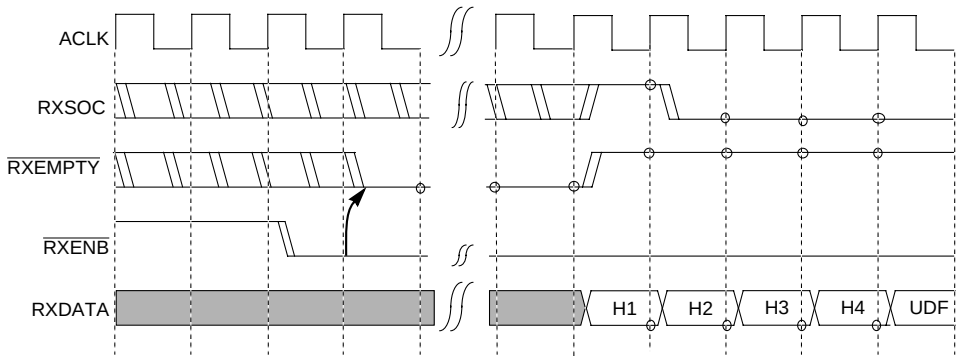


Figure 4-5. Start of Cell—MC92520 Empty (Octet-Level Handshake)

Figure 4-6 shows the end of a cell when the MC92520 is able to receive another cell, but the PHY has no more data to transfer. $\overline{\text{RXEMPTY}}$ is asserted after the last byte of valid data is transferred.

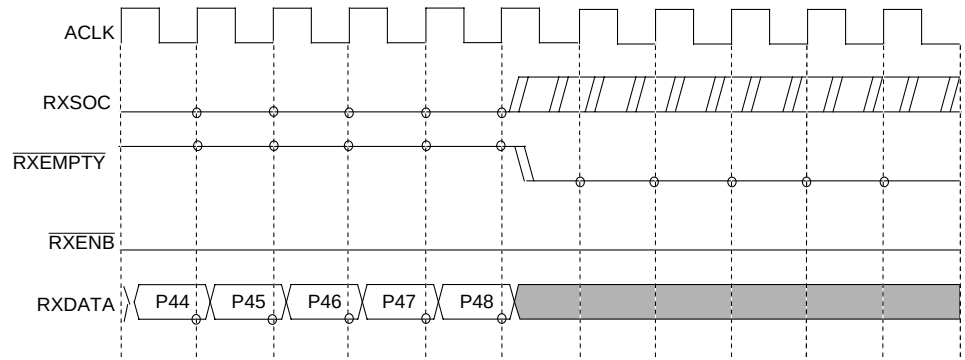


Figure 4-6. End of Cell—PHY Empty (Octet-Level Handshake)

Figure 4-7 shows the reception of back-to-back cells. RXSOC is asserted together with the first octet of the new cell.

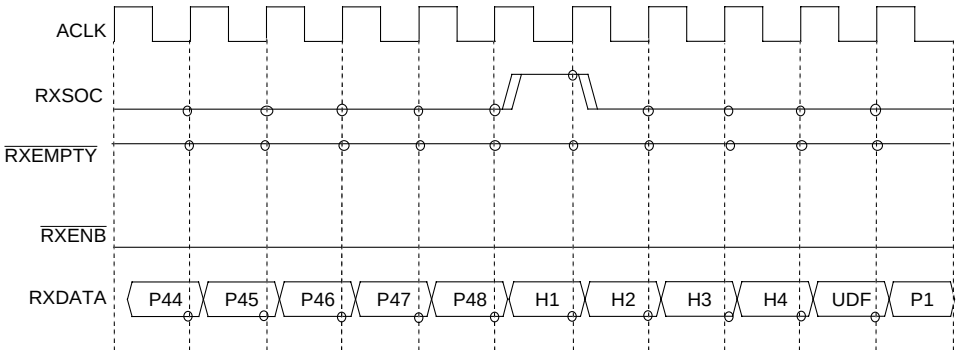


Figure 4-7. Back-to-Back Cell Input (Octet-Level Handshake)

Figure 4-8 shows an example where the PHY runs out of valid data in the middle of the cell and uses RXEMPTY to regulate the transfer of data. When the MC92520 detects RXEMPTY asserted, the value on RXDATA is discarded and is not included in the received cell.

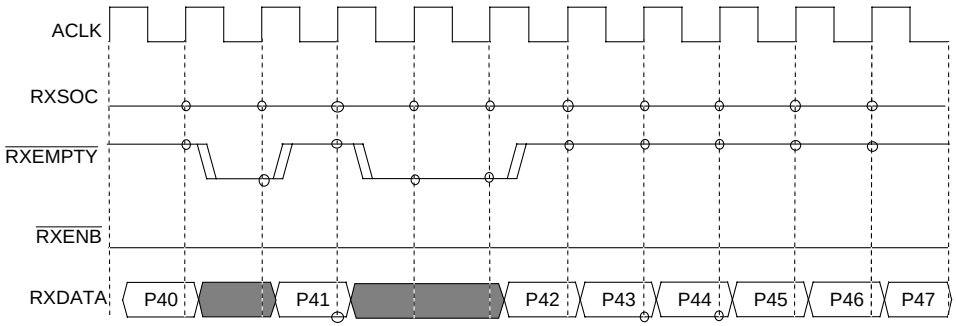


Figure 4-8. Response to RXEMPTY Assertion (Octet-Level Handshake)

4.2.1.2 Cell-Level Handshake

When a cell-level handshake is used, RXEMPTY is connected to a receive cell available (RxClav, UTOPIA level 2 name) pin on the PHY device. In this case, the negation of RXEMPTY means that the PHY has an entire cell available to be transferred in consecutive clock cycles. Therefore, once RXEMPTY is sampled negated by the MC92520, it is ignored until the entire cell has been transferred, and the PHY cannot interrupt the transfer of data in the middle of a cell. The start of a cell is shown in Figure 4-9 with a 16-bit wide data bus. RXEMPTY must be valid from the clock cycle following the assertion of RXENB until it is sampled negated.

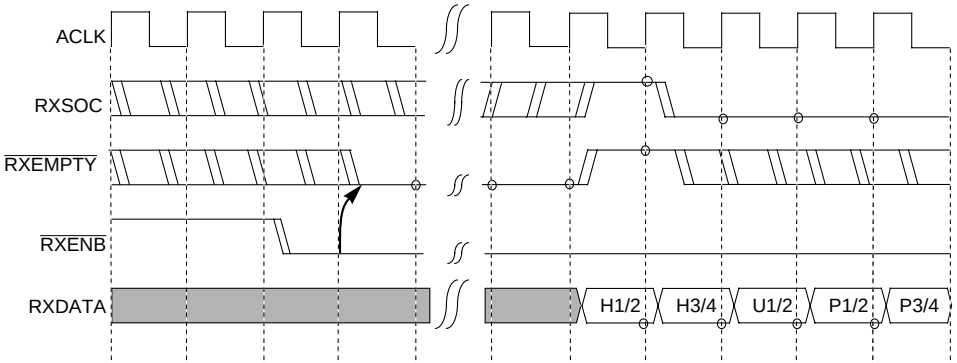


Figure 4-9. Start of Cell—MC92520 Empty (Cell-Level Handshake)

Figure 4-10 shows the end of a cell when the MC92520 can receive another cell, but the PHY does not have an entire cell available. $\overline{\text{RXEMPTY}}$ is sampled starting after the last 16-bit word of the cell has been transferred.

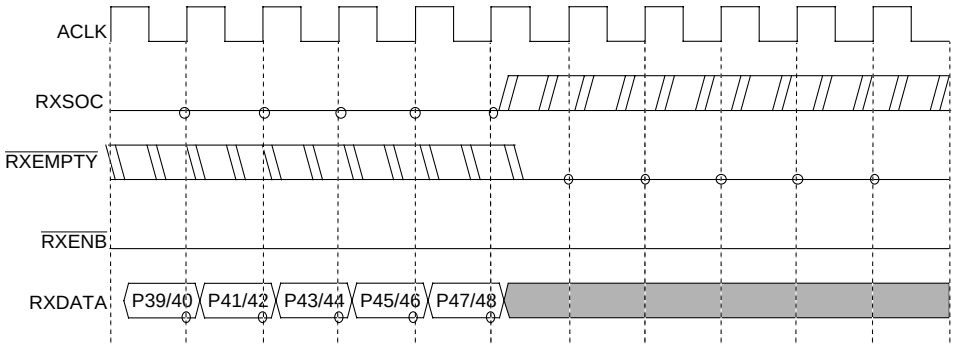


Figure 4-10. End of Cell—PHY Empty (Cell-Level Handshake)

Figure 4-11 shows the reception of back-to-back cells. $\overline{\text{RXEMPTY}}$ is sampled starting after the last word of the cell is transferred. Because $\overline{\text{RXEMPTY}}$ is negated, the next cell transfer begins immediately. RXSOC is asserted together with the first word of the new cell.

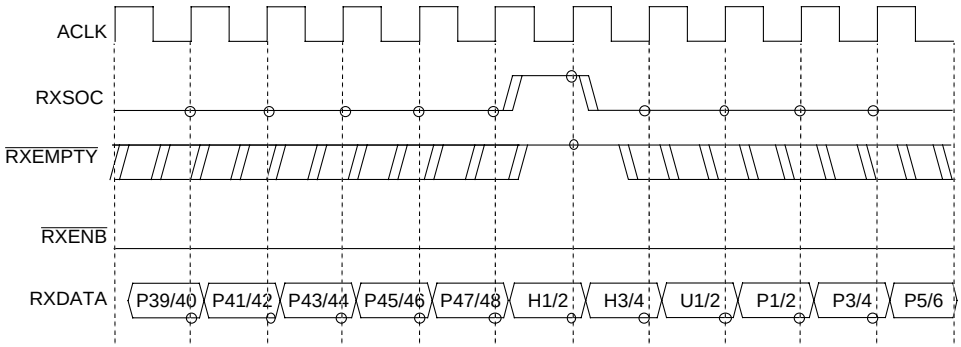


Figure 4-11. Back to Back Cell Input (Cell-Level Interface)

4.2.2 Single-PHY Transmit Interface (Egress)

The PHY-side transmit interface of the MC92520 is configured for single-PHY operation if the egress UTOPIA multi-PHY (EPHCR[EUM]) bit is cleared. The single-PHY transmit interface supports both octet-, word-, and cell-level handshakes, and the value of the egress PHY operation mode (EPOM) bit determines the selected handshake method. Similarly, the value of the egress PHY wide data path (EPWD) bit determines whether a word (16-bit) or an octet (8-bit) is transferred. If the MC92520 is configured to transfer octets, the state of TXDATA[15:8] is not valid. The signals involved in the transmit interface are shown in Figure 4-12.

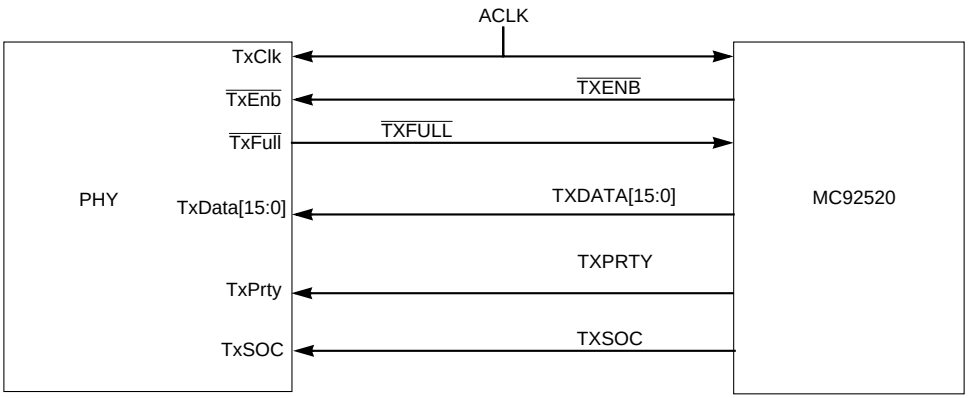


Figure 4-12. MC92520 Single-PHY Transmit Interface

The TXENB output signal is used by the MC92520 to indicate that it is driving valid data on TXSOC, TXDATA, and TXPRTY in the current clock cycle. When TXENB is asserted, the PHY should sample the data values. When TXENB is negated, the data values are

invalid. Figure 4-13 shows the MC92520 with an 8-bit data bus using $\overline{\text{TXENB}}$ at the end of a cell to indicate that it has no more valid data to transmit. One clock later, it is ready to transmit the next cell and asserts $\overline{\text{TXENB}}$.

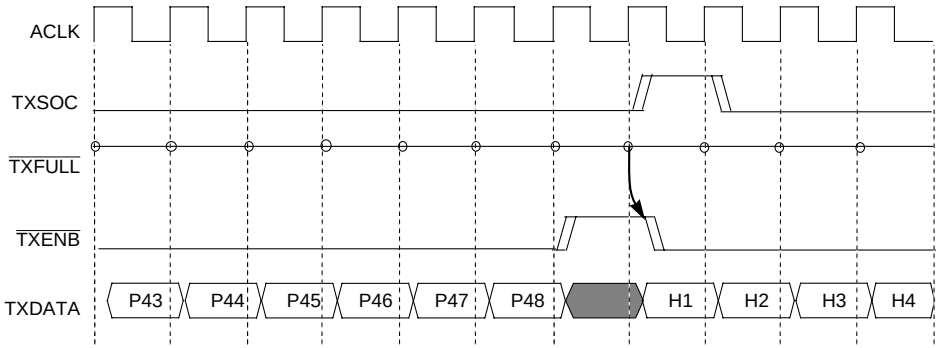


Figure 4-13. MC92520 Transmit Timing—MC92520 is Empty

The MC92520 asserts TXSOC during the first octet of a cell to synchronize the PHY. TXPRTY contains the odd parity over TXDATA. Depending on the settings of the (EPHCR[EPWD]) bit, parity is checked for a word-wide data path (TXDATA[15:0]) or an octet-wide data path (TXDATA[7:0]). The TXPRTY signal is not shown in the timing diagrams because its timing is identical to that of TXDATA.

4.2.2.1 Octet/Word-Level Handshake

When octet- or word-level handshaking is used, the PHY uses the $\overline{\text{TXFULL}}$ signal to regulate the data flow on an octet or word basis. When $\overline{\text{TXFULL}}$ is negated and the MC92520 has a cell to transmit, $\overline{\text{TXENB}}$ is asserted, and the data is transferred. If the PHY cannot receive more data, it asserts $\overline{\text{TXFULL}}$. The MC92520 checks $\overline{\text{TXFULL}}$ at every rising edge of ACLK. If $\overline{\text{TXFULL}}$ is asserted, $\overline{\text{TXENB}}$ is negated until the negation of $\overline{\text{TXFULL}}$. Figure 4-14 shows an example where the MC92520 has a cell waiting to be transmitted with an 8-bit wide data bus. As soon as $\overline{\text{TXFULL}}$ is negated, the MC92520 asserts $\overline{\text{TXENB}}$ and begins to transmit the cell.

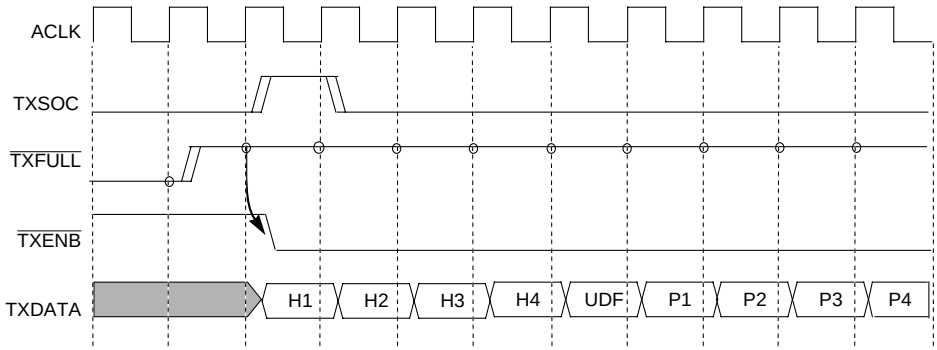


Figure 4-14. Start of Cell (Octet-Level Handshake)

Figure 4-15 shows the transmission of back-to-back cells with an 8-bit wide data bus. TXSOC is asserted together with the first octet of the new cell.

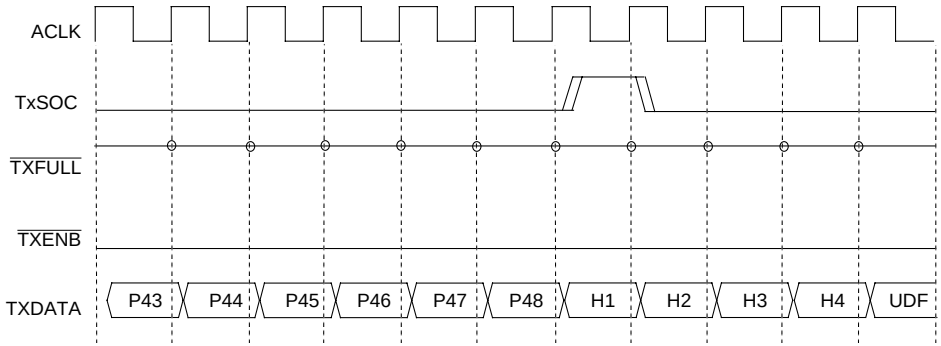


Figure 4-15. Back-to-Back Cell Output (Octet-Level Handshake)

Figure 4-16 shows another 8-bit data bus example where, in the middle of a cell, the PHY is unable to accept more data and uses $\overline{\text{TXFULL}}$ to regulate the transfer of data. In response, the MC92520 negates $\overline{\text{TXENB}}$ and does not update the data.

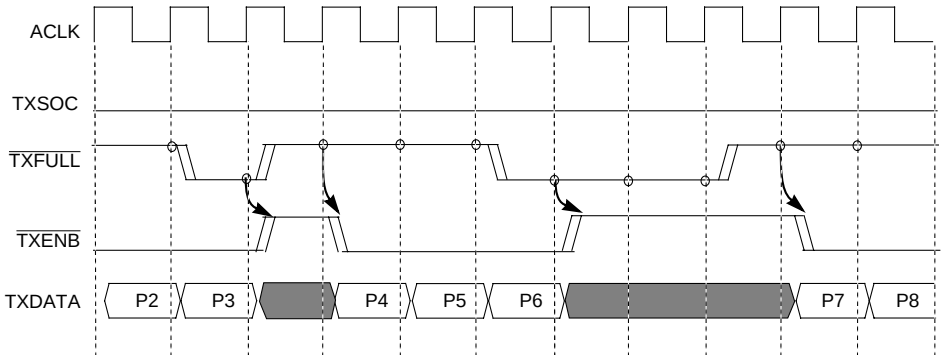


Figure 4-16. Response to TXFULL Assertion (Octet-Level Handshake)

4.2.2.2 Cell-Level Handshake

If cell-level handshaking is used, TXFULL is connected to a transmit cell available (TxClav, UTOPIA level 2 signal name) pin on the PHY device. In this case, the negation of TXFULL means that the PHY layer can accept an entire cell in consecutive clock cycles. Therefore, when the MC92520 begins to transmit a cell, TXFULL is ignored until the entire cell is transferred, and the PHY cannot interrupt the transfer of data in the middle of a cell. The start of a cell using a 16-bit wide data bus is shown in Figure 4-17. TXFULL must remain valid until TXENB is asserted.

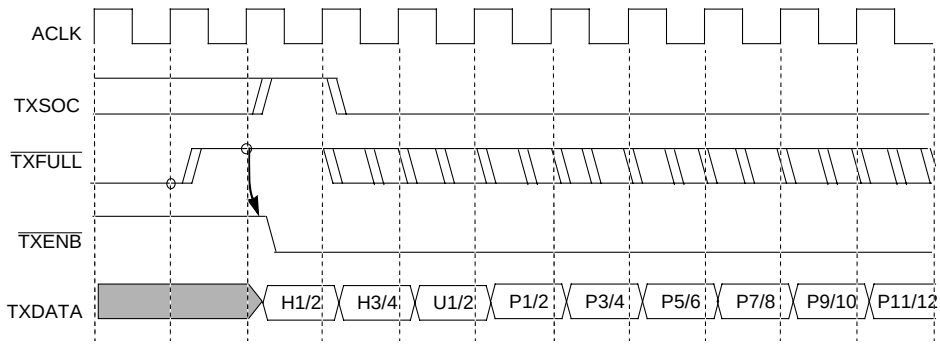


Figure 4-17. Start of Cell (Cell-Level Handshake)

Figure 4-18 shows the transmission of back-to-back cells with a 16-bit wide data bus. TXFULL is sampled starting with the last word of the cell. Because TXFULL is negated, the next cell transfer begins immediately. TXSOC is asserted together with the first word of the new cell.

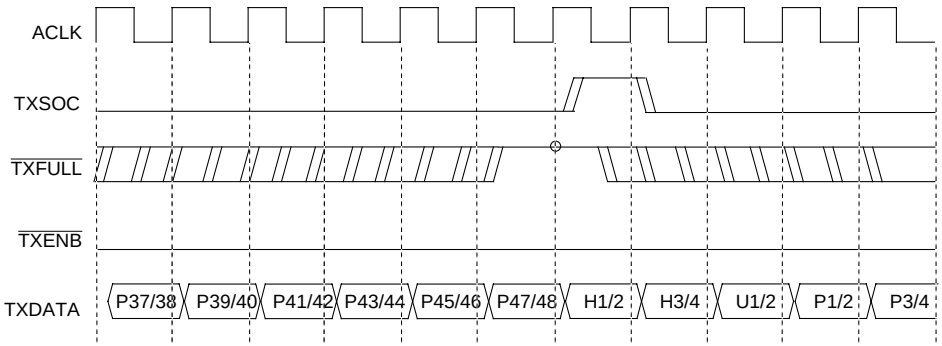


Figure 4-18. Back-to-Back Cell Output (Cell-Level Handshake)

4.2.3 Multi-PHY Receive Interface (Ingress)

The PHY-side receive interface of the MC92520 is configured for multi-PHY operation if the ingress UTOPIA multi-PHY bit (IPHCR[IUM]) is set. The multi-PHY receive interface supports the use of one RxClav signal. Note that only cell-level handshake is supported in multi-PHY mode, and the the ingress PHY operation mode (IPOM) bit must be programmed accordingly. The value of the ingress PHY wide data path (IPWD) bit determines whether a word (16-bit) or an octet (8-bit) is transferred. If the MC92520 is configured to receive octets, the state of RXDATA[15:8] is ignored, and the associated pins should be tied to ground. Similarly, parity checking (if enabled) is performed either on a word or octet basis.

Polling of individual multi-PHY addresses is enabled using the link enable (LE) bit in the ingress link registers ILINK0–ILINK15. The MC92520 has up to a 2-ACLK-cycle latency to enable or disable polling of a PHY. Of course, the moment that a newly enabled PHY is first polled depends upon its relative position in the PHY address range and on the current position of the polling loop.

If the MC92520 is configured to receive words (the ingress PHY wide data path (IPWD) bit is set), the number of available polling cycles per cell time is 11, meaning that it is less than the MC92520-supported maximum of 16 PHY addresses. Using a PHY address higher than 11 may reduce the MC92520 cell processing bandwidth unless at least one PHY responds with a cell available signal during the first 12 polling cycles following a PHY select phase. In most applications, this is not an issue. This condition is satisfied (and no bandwidth reduction is caused), for example, if all PHYs support the same cell rate.

The signals involved in the receive interface are shown in Figure 4-19.

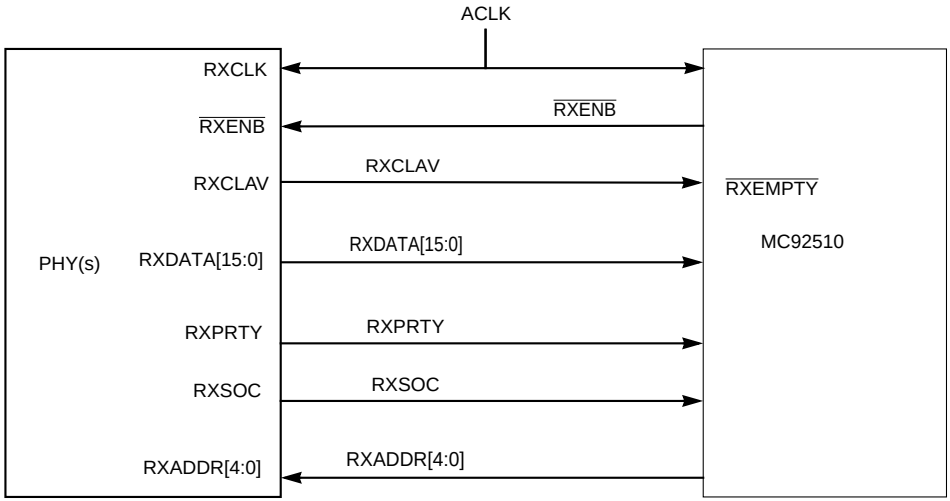


Figure 4-19. MC92520 Multi-PHY Receive Interface

To poll a PHY, the MC92520 assigns the polled PHY address on the RXADDR[4:0] lines on the first clock. The polled PHY drives RXCLAV on the second clock. (Note that the MC92520 pin $\overline{\text{RXEMPTY}}$ is used for the RXCLAV input in multi-PHY applications.) On that clock, the MC92520 assigns 0x1F to RXADDR[4:0] and samples RXCLAV. When there is no cell traffic, the MC92520 polls all the PHYs whose corresponding ingress link register link enable bit (LE) is set in a cyclic descending order. The first PHY that indicates that it is ready is then selected by the MC92520. The MC92520 assigns its address on the RXADDR[4:0] bus with $\overline{\text{RXENB}}$ negated on the first clock, and on the second clock assigns 0x1F on the RXADDR[4:0] lines and asserts $\overline{\text{RXENB}}$. One clock later, the MC92520 outputs the first octet or word of the cell on the RXDATA lines and continues outputting the cell content on subsequent clocks contiguously. See Figure 4-20.

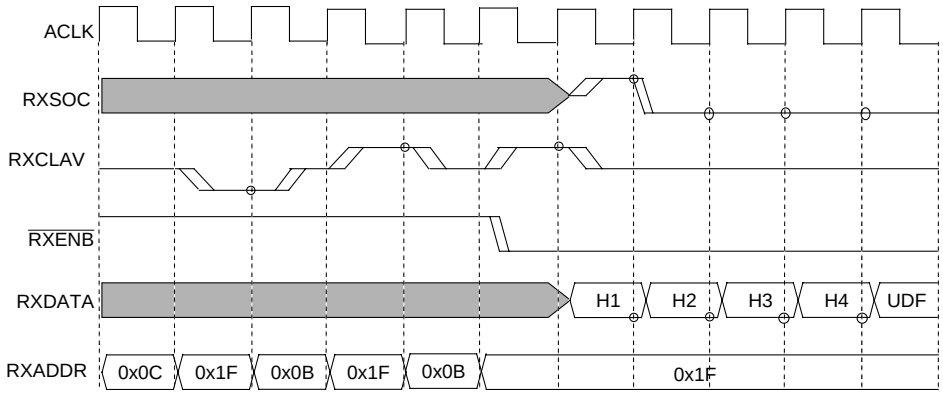


Figure 4-20. Poll PHYs, Select PHY, and Start Reading a Cell

When a cell transfer is in progress, the MC92520 polls the PHYs from the PHY that comes next to the current selected PHY in a descending cyclic order. Once the MC92520 detects a PHY that is ready, it stops the polling process and waits for the current cell transfer to end. It then deselects the current PHY, selects the chosen PHY, and starts transferring the cell data. See Figure 4-21.

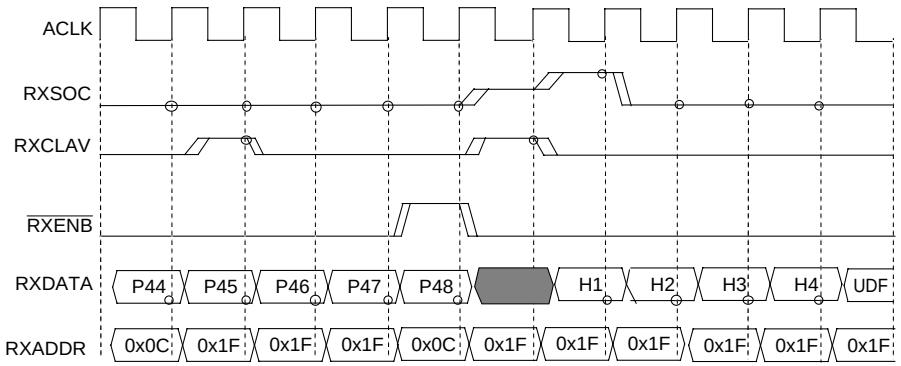


Figure 4-21. Polls the PHYs While Transferring a Cell

The MC92520 does not poll the current selected PHY in the middle of its cell transfer because RXCLAV of its PHY may not indicate the availability of the next cell. (This is certain if external hardware provides polling status for one or more UTOPIA level 1 compliant PHYs.) If no PHY is ready, the MC92520 deselects the current PHY, polls the current PHY, and, if it is ready to transfer another cell, reselects it. Otherwise, cyclic polling of PHYs continues till a PHY ready to transfer is found.See Figure 4-22.

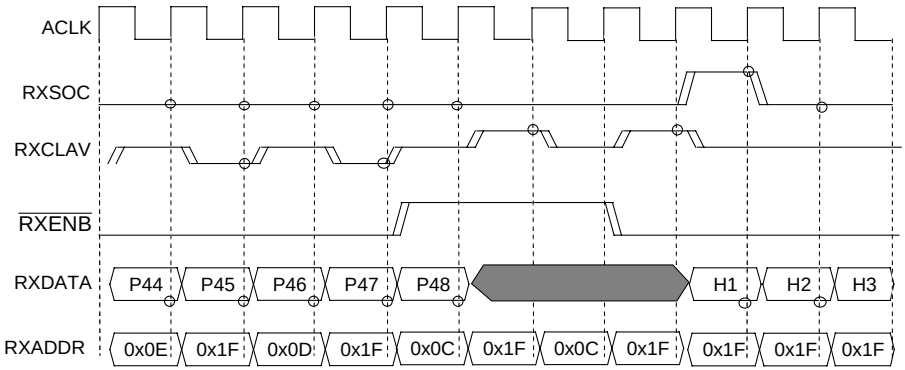


Figure 4-22. The MC92520 Reads Cells from the Same PHY

4.2.4 Multi-PHY Transmit Interface (Egress)

The PHY-side transmit interface of the MC92520 is configured for multi-PHY operation if the egress UTOPIA multi-PHY bit (EPHCR[EUM]) is set. If this bit is set, the egress PHY operation mode bit (EPHCR[EPOM]) must also be set, which will configure the ATMC for cell-level handshake.

The multi-PHY transmit interface supports the use of one TXCLAV signal. The MC92520 pin $\overline{\text{TXFULL}}$ is used for the TXCLAV function in multi-PHY applications.

The value of the egress PHY wide data path (EPWD) bit determines whether a word (16-bit) or an octet (8-bit) is transferred. If the MC92520 is configured to transmit octets, the state of RXDATA[15:8] is not valid. Similarly, parity generation is performed either on a word or octet basis.

The signals involved in the transmit interface are shown in Figure 4-23.

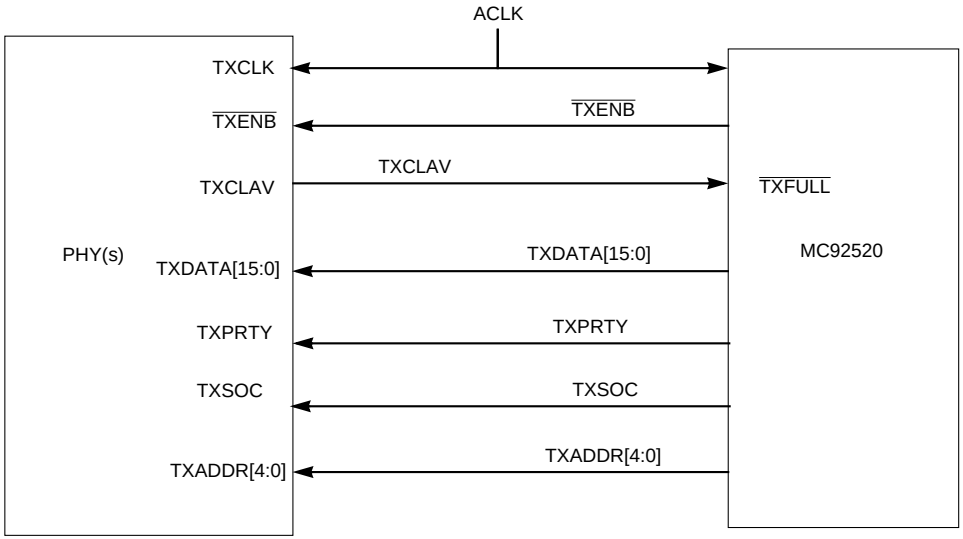


Figure 4-23. MC92520 Multi-PHY Transmit Interface

The PHY-side transmit interface can be configured to use either a single FIFO or multiple, per-PHY FIFOs. Depending on the FIFO configuration, the MC92520 uses different polling strategies:

- In single-FIFO configuration, the MC92520 starts to poll a PHY as soon as a cell for the PHY becomes available for transmission. After the PHY responds with an asserted TXCLAV signal, the PHY is selected by the MC92520 and cell transfer starts. If a new cell becomes available and targets a different PHY, polling is resumed while the current cell is transfered. Once the new target PHY responds with an asserted TXCLAV signal, polling is suspended until the transfer of the new cell starts. On the other hand, if a new cell targets the same PHY, polling is resumed only after cell transfer of the current cell is completed. A typical FIFO polling phase for an 8-bit wide data bus is illustrated in Figure 4-24 below.
- In multi-FIFO configurations, polling is performed regardless of whether a cell is available for transfer. The MC92520 cyclically polls each enabled PHY until it has responded with an asserted TXCLAV signal. Polling of a PHY is suspended after the PHY responds with an asserted TXCLAV signal. All cells ready to be transferred to a PHY are temporarily queued in the per-PHY FIFOs. If more than one FIFO contains a cell ready for transfer, the egress transfer priority is used to select the cell to be transferred. Otherwise, cells are transferred in the order they become available at the PHY-side transmit interface. After a cell is transferred to a PHY, polling of the PHY is resumed.

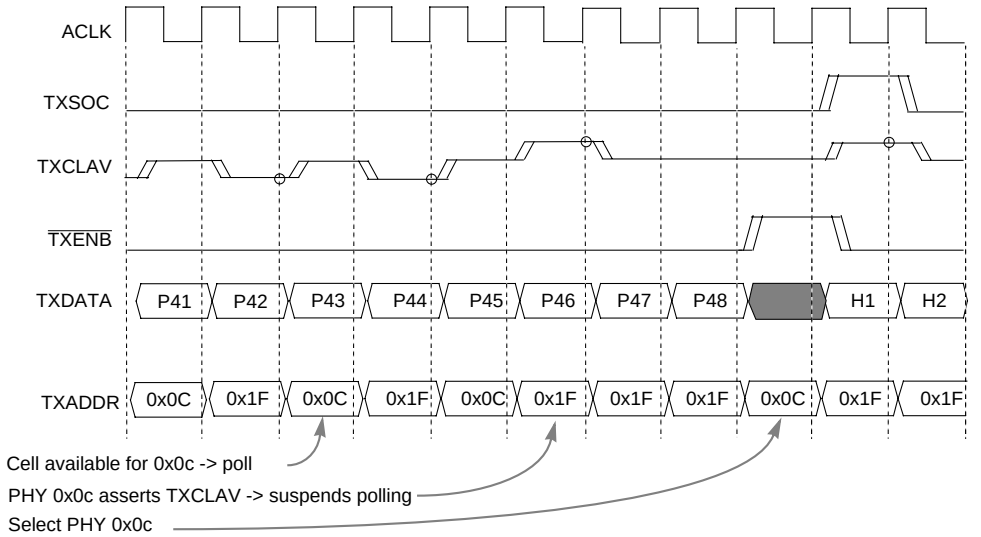


Figure 4-24. MC92520 Transmit Timing—Single-FIFO Polling

4.3 Switch Interface

The switch receive and transmit interfaces are similar to a UTOPIA standard interface using cell-level handshake. Both MC92520 interfaces act as PHY layer (UTOPIA slave) devices and can be independently programmed to transfer data with a wide (16-bit) or a standard (8-bit) data path. In addition, the transmit interface can be configured to support single-PHY or multi-PHY operation.

NOTE:

The MC92520 can achieve its full data rate only with a 16-bit data path.

All configuration options are programmed through the ingress and egress switch interface configuration registers described in Section 7.1.6.5, “Ingress Switch Interface Configuration Register (ISWCR),” Section 7.1.6.6, “Egress Switch Interface Configuration Register (ESWCR),” and Section 7.1.6.7, “Egress Switch Interface Configuration Register 1 (ESWCR1).”

All receive interface signals are synchronous to SRXCLK, and all transmit interface signals are synchronous to STXCLK. Output signals from the MC92520 are updated following the rising edge of the interface clock, and input signals to the MC92520 are sampled at the rising edge of the interface clock. The flow of data is controlled by enable signals in both directions. The general rule is that an enable signal in the direction of the data flow refers to data during the current cycle, and an enable signal in the direction opposite to the data flow refers to data at the end of the next cycle.

4.3.1 Receive Interface (Ingress)

The switch-side receive interface of the MC92520 can be configured for both a word (16-bit), or an octet (8-bit) data path using the ingress switch wide data path (ISWD) bit in the ingress switch interface configuration register (ISWCR). If the MC92520 is configured to transfer octets, the state of SRXDATA[15:8] is not valid.

The signals involved in the MC92520 switch receive interface are shown in Figure 4-25.

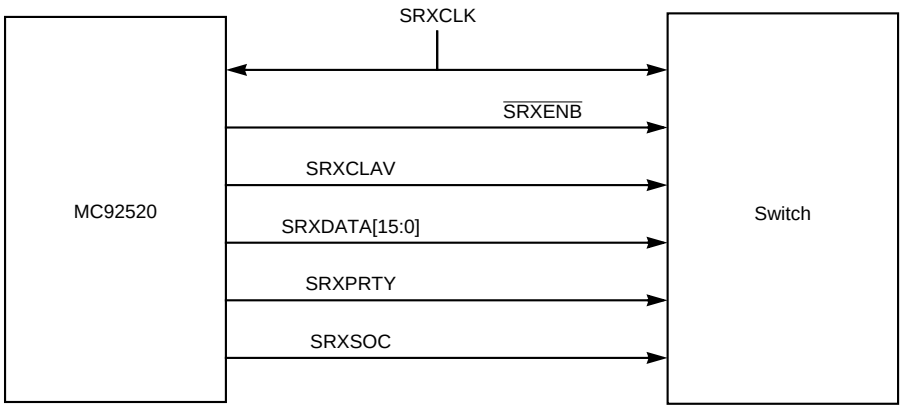


Figure 4-25. MC92520 Switch Receive Interface

The switch requests data transfers in the receive direction by asserting SRXENB. The MC92520 responds by supplying valid data (if available) during the next clock cycle. If SRXENB is detected as negated at the rising edge of SRXCLK, then SRXDATA, SRXPRTY, and SRXSOC are not updated as shown in Figure 4-26. In this way, the switch can throttle the flow of data.

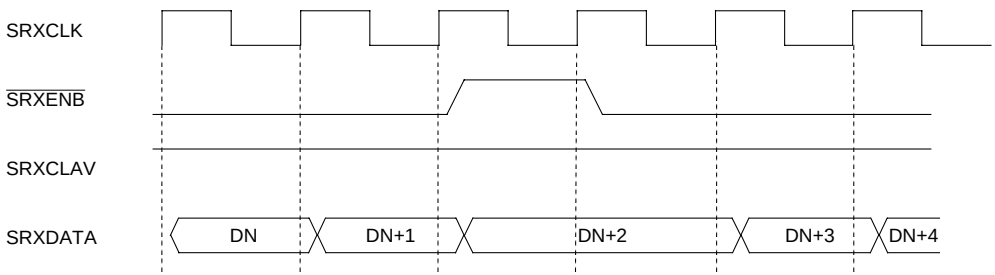


Figure 4-26. Receive Timing without Tri-state—Effect of SRXENB Negation

If the ingress switch SRXDATA driver control bit (ISSDC) of the ingress switch interface configuration register (ISWCR) is reset, then SRXDATA, SRXPRTY, and SRXSOC are not driven when SRXENB is negated as shown in Figure 4-27.

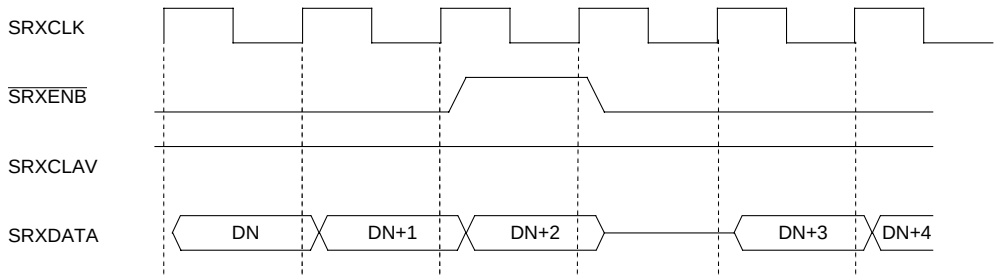


Figure 4-27. Receive Timing with Tri-state—Effect of SRXENB Negation

Figure 4-28 shows the beginning of a cell transfer. Upon detecting SRXCLAV asserted, the switch should assert SRXENB within a cell transfer slot (the number of SRXCLK cycles required to transfer one cell to the switch). Otherwise, the MC92520 may be unable to accept data from the PHY, and cells may be lost. The MC92520 asserts SRXSOC together with the first octet of each cell in order to synchronize the switch to the beginning of a new cell. If 57 octets are transferred, for example, SRXSOC is asserted on the transfer of the first overhead byte and not on the first header byte.

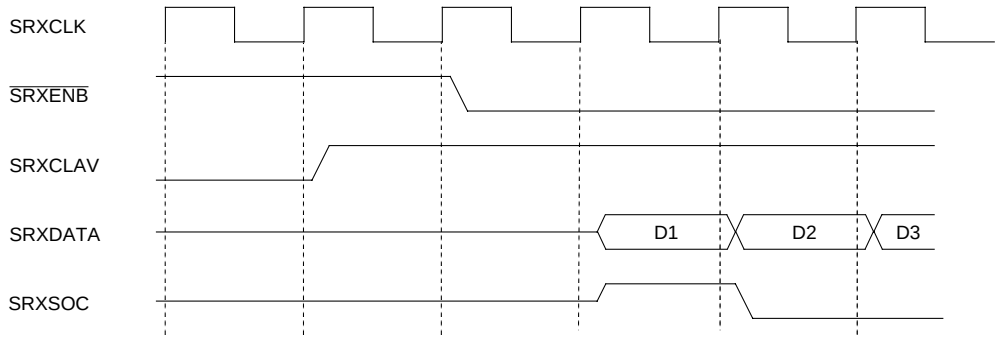


Figure 4-28. Receive Timing—Start of Cell

The MC92520 negates SRXCLAV at the end of a cell transfer if it does not have another cell to provide as shown in Figure 4-29.

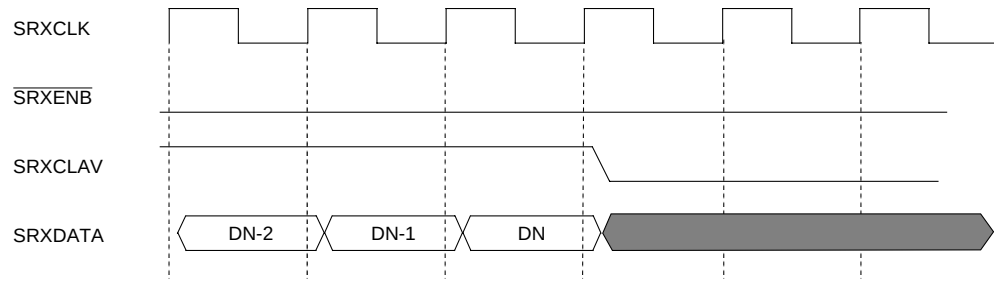


Figure 4-29. Receive Timing—End of Cell

If there is another cell available, SRXCLAV is not negated and the cells can be transferred back-to-back as shown in Figure 4-30.

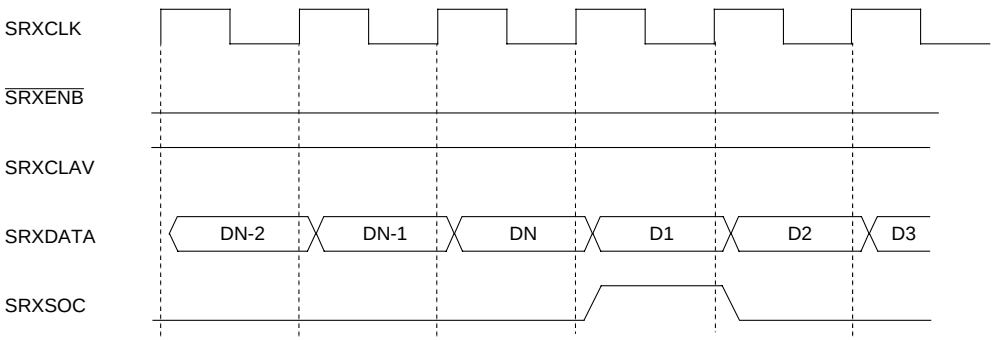


Figure 4-30. Receive Timing—Back-to-Back Cell Transfers

SRXPRTY contains the parity over SRXDATA. It is not shown in the timing diagrams because its timing is identical to that of SRXDATA. The type of parity (even or odd) is determined by the ingress switch parity mode bit (ISPM) defined in Section 7.1.6.5 “Ingress Switch Interface Configuration Register (ISWCR).”

4.3.2 Transmit Interface (Egress)

The switch-side transmit interface of the MC92520 can be configured for both a word (16-bit), or an octet (8-bit) data path using the egress switch wide data path bit (ESWD) in the egress switch interface configuration register (ESWCR). If the MC92520 is configured to transfer octets, the state of STXDATA[15:8] is ignored and the associated pins should be tied to ground.

The MC92520 can be configured to support a single-PHY or a glue-less, multi-PHY interface on its switch-side transmit interface. Section 4.3.2.1, “Single-PHY Interface” and Section 4.3.2.2, “Multi-PHY Interface” describe the signals and interface operation for the two types of interfaces.

4.3.2.1 Single-PHY Interface

The single-PHY interface is selected by clearing the egress UTOPIA multi-PHY bit (ESWCR[ESUM]). The signals involved in the MC92520 single-PHY switch transmit interface are shown in Figure 4-31.

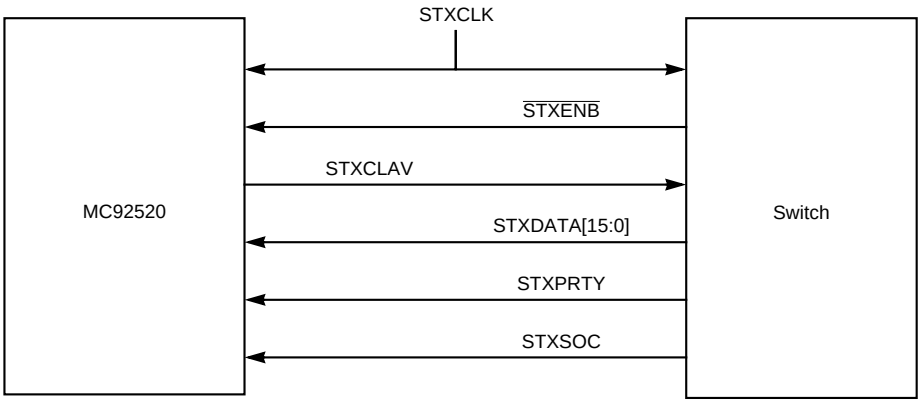


Figure 4-31. MC92520 Single-PHY Switch Transmit Interface

Figure 4-32 shows the relationships among the transmit interface signals. When the switch has data to transfer, and STXCLAV is asserted, the switch asserts STXENB and drives STXDATA with an octet or word of valid data. Once STXCLAV is asserted, the MC92520 must accept an entire cell, and the switch may continue to transfer one byte or word per clock cycle until the end of the cell. The switch asserts STXSOC together with the first octet or word of each cell to synchronize the MC92520 to the start of the cell.

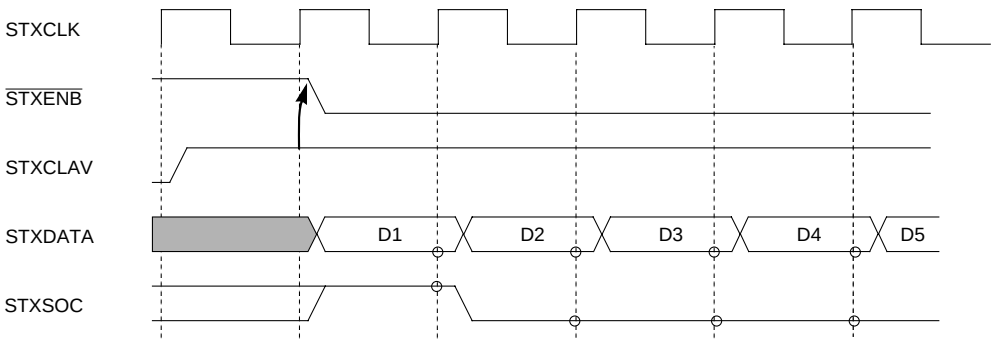


Figure 4-32. Single-PHY Transmit Timing—Start of Cell

When the MC92520 is full, it negates STXCLAV at least 4 clock cycles before the end of the cell as shown in Figure 4-33. When the switch detects the negation of STXCLAV, it

transfers the remainder of the cell and then negates $\overline{\text{STXENB}}$. The switch may assert $\overline{\text{STXENB}}$ only after detecting the assertion of STXCLAV. Asserting $\overline{\text{STXENB}}$ otherwise is reported as a protocol error by the MC92520. The MC92520 samples STXDATA, STXPRTY, and STXSOC on each rising edge of STXCLK during which $\overline{\text{STXENB}}$ is asserted.

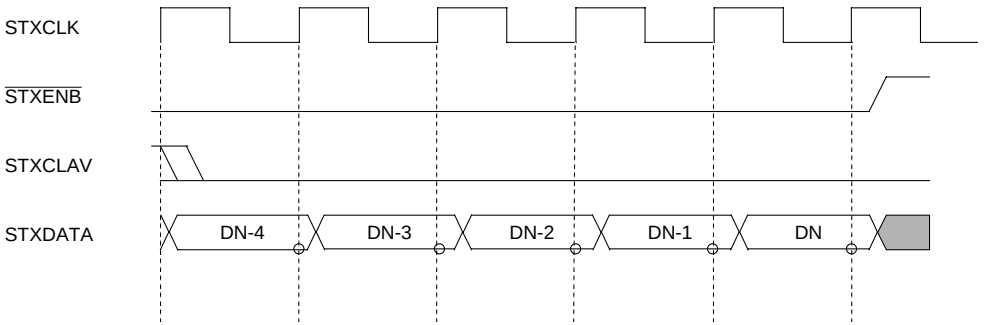


Figure 4-33. Single-PHY Transmit Timing (End of Cell)

STXPRTY contains the parity for STXDATA. It is not shown in the timing diagrams because its timing is identical to that of STXDATA. The type of parity (even or odd) is determined by the egress switch parity control (ESPC) bit . Parity checking is controlled by the egress switch parity enable (ESPR) and egress payload parity enable (EPLP) bits, all defined in Section 7.1.6.6 “Egress Switch Interface Configuration Register (ESWCR).”

4.3.2.2 Multi-PHY Interface

The multi-PHY interface is selected by setting the egress UTOPIA multi-PHY bit (ESWCR[ESUM]). In multi-PHY operation, the MC92520 responds to one or more PHY addresses and shares the UTOPIA bus with other devices (possibly MC92520s) with different PHY addresses. The addresses the MC92520 responds to is determined from the configuration values of a PHY port base address and a PHY port address mask in the egress switch configuration register 1 (ESWCR1). For more information, see Section 3.3.2.1.2, “Multi-PHY Configuration,” and Section 7.1.6.7, “Egress Switch Interface Configuration Register 1 (ESWCR1).”

All signals involved in the MC92520 multi-PHY switch transmit interface are shown in Figure 4-34 below. The MC92520 supports an optional PHY address valid (STXAVALID) signal in addition to the signals defined by the standard UTOPIA level 2 interface. If used, this signal is asserted together with a valid PHY address and negated at least 1 clock cycle between address bus transitions. The use of the STXAVALID signal is controlled via the egress switch multi-PHY extension bit (EPCR1[ESAV]). If ESAV is cleared, the MC92520 views PHY address 0x1F as a null address that never matches an MC92520 attached PHY port. If the ESAV bit is set, the MC92520 does not respond to any PHY address while STXAVALID is negated. But if STXAVALID is asserted, the MC92520 responds to any enabled PHY address, that is, PHY address 0x1F does not have special meaning.

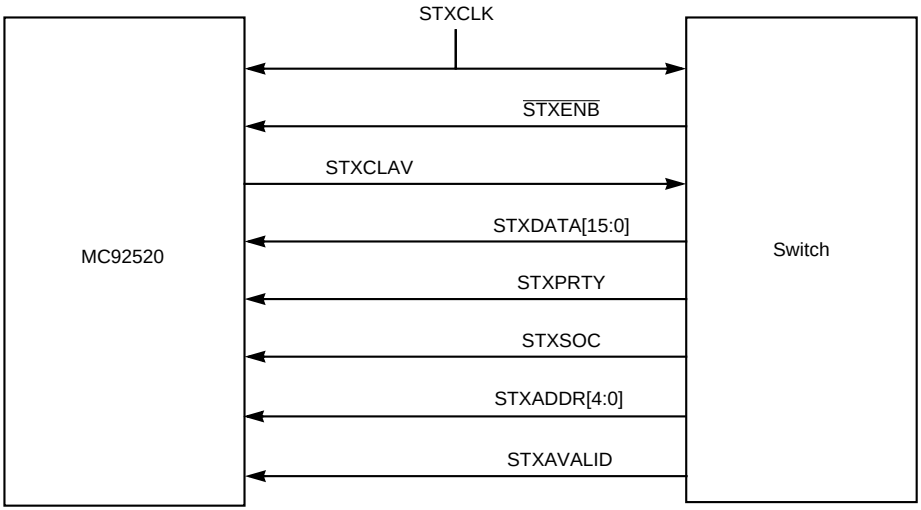


Figure 4-34. MC92520 Multi-PHY Switch Transmit Interface

The MC92520 responds with STXCLAV to any enabled PHY address that has been identified as attached to the MC92520 via the PHY port base address and PHY port address mask configuration. In the clock cycle following a matched address, STXCLAV is asserted if a cell can be accepted and negated if there is no room in the internal per-PHY FIFO. There is no requirement for a specific address polling order, and once STXCLAV is asserted by the MC92520, subsequent polling will result in an asserted STXCLAV until a cell transfer selecting the associated PHY is completed. The status of the associated FIFO is only reevaluated after a cell is transferred to provide an updated STXCLAV in response to a polling request. The MC92520 generates a protocol error if a PHY address is selected for cell transfer in the presence of a negated STXCLAV signal (that is, the MC92520 indicates that there is no space in the per-PHY FIFO).

The following figure shows an example where PHYs are polled until the end of a cell transmission cycle. The STXCLAV signal shows that PHYs 0x09, 0x08, and 0x06 can accept cells and that PHY 0x08 is selected. This selection occurs in the cycle in which $\overline{\text{STXENB}}$ is asserted together with STXSOC.

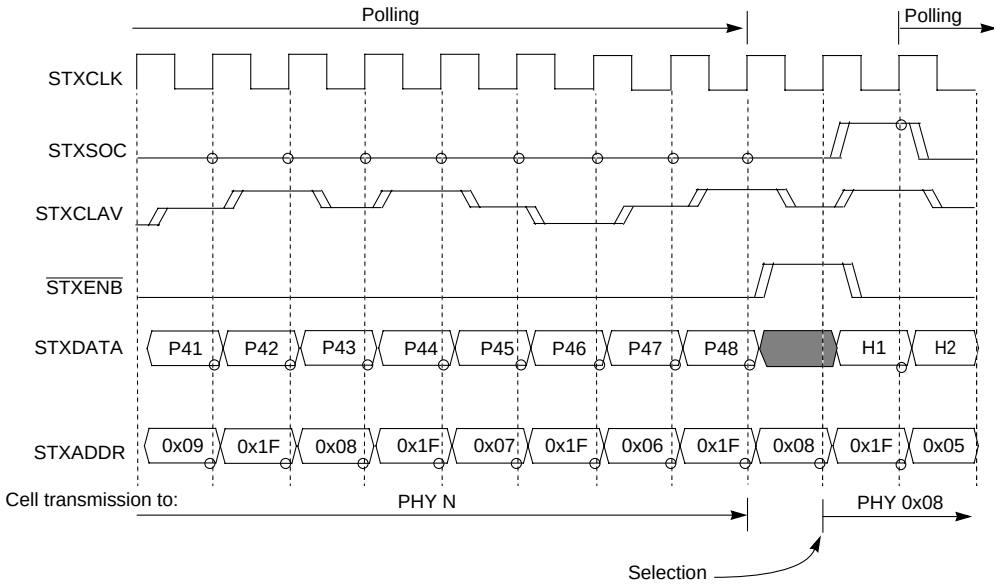


Figure 4-35. Polling Phase and Selection Phase

4.4 External Memory Interface

The MC92520 is the sole master of the external memory (EM) interface. During setup mode the microprocessor can directly write and read EM. In operate mode, EM is accessed indirectly through the external memory request FIFO and an indirect external access method (described in Section 3.2, “Operate Mode”). Owing to the bandwidth needed from the EM and the requirement of one access per ZCLK cycle, the EM interface is designed to work with pipelined ZBT RAMs and not with any other type of RAM.

4.4.1 EM Chip Enables

The pipelined ZBT RAMs required for use with the EM interface have three chip enables per device. The MC92520 outputs the most-significant bits of the EM address in both active high and active low form. To configure the RAM devices in up to eight banks, the three RAM chip enable inputs can be connected to a combination of active high and active low most-significant address bits such that only one RAM device is enabled for any given address. Note that the EM write enable ($\overline{\text{EMWR}}$), EM high word write select ($\overline{\text{EMWSH}}$) and EM low word write select ($\overline{\text{EMWSL}}$) disable all RAMs on clock cycles that have no memory access. (These signals decode to a write operation with neither high nor low word enabled on memory no-operation (NOP) cycles.) Therefore, only the upper address lines are needed for individual RAM selection. Depending on the size of ZBT RAMs used, the designer may choose to use any three consecutive address bits from EMADD[23:19] to decode one of eight RAM banks.

The following example shows the case where sixteen 8-Mbit ZBT RAMs, which are 512K × 16 bits each, are used to create the maximum 16-Mbyte EM space:

- Eight RAMs are connected to EMDATA[31:16] (high-word RAMs), the other eight are connected to EMDATA[15:0] (low-word RAMs).
- The high-word RAMs have both of their synchronous byte write inputs ($\overline{\text{SBb}}$, $\overline{\text{SBa}}$) connected to $\overline{\text{EMWSH}}$.
- The low-word RAMs have both of their synchronous byte write inputs connected to $\overline{\text{EMWSL}}$.
- All RAMs have the following:
  - Synchronous write input connected to pin $\overline{\text{EMWR}}$.
  - Address inputs driven by EMADD20–EMADD2.
  - Asynchronous output enable ($\overline{\text{G}}$) driven by $\overline{\text{EMOE}}$.
  - Advance pin (ADV) tied low, as the ZBT burst capability is not used.
  - Clock enable ($\overline{\text{CKE}}$) input tied low, so that the clock is always enabled.
  - Linear burst order ($\overline{\text{LBO}}$) input tied high or low, as this pin is not used.
  - Sleep mode pin (ZZ) tied low, so that the RAM is never in sleep mode.
  - Clock input (CK) driven by the MC92520 EM clock output.
- One high-word RAM and one low-word RAM is selected for each of the eight possible states (000–111) of the three EMADD outputs that are directly above the EMADD outputs used to drive the ZBT RAM address inputs. In this example, EMADD[20:2] are used to drive the ZBT RAM address inputs, so EMADD[23:21] and/or $\overline{\text{EMADD}}$[23:21] are used as the ZBT RAM select inputs.

4.4.2 EM Interface

The MC92520 EM interface is designed to work only with pipelined ZBT RAMs. On a memory no-operation (NOP) cycle, the $\overline{\text{EMWR}}$ signal indicates a write, and the $\overline{\text{EMWSH}}$ and $\overline{\text{EMWSL}}$ signals indicate neither word selected. On read or write cycles, EMADD contains the memory address, while $\overline{\text{EMWR}}$, $\overline{\text{EMWSH}}$, and $\overline{\text{EMWSL}}$ indicate whether a read or write is to be performed and in the case of a write, which words to write. If it is a write cycle, EMDATA outputs the data 2 clock cycles after the EMADD output. If it is a read cycle, the EMDATA pins are sampled 2 clock cycles after the EMADD output. Note that $\overline{\text{EMOE}}$ disables the ZBT RAM outputs only at reset, so that there is no contention on the EMDATA bus. For more detail, see specifications for pipelined ZBT RAMs, such as the Motorola MCM63Z918.

4.4.3 External Address Compression Device Interface

The external address compression (EAC) device (typically a content addressable memory—CAM) cannot be accessed by the microprocessor through the MC92520. If

microprocessor access to the EAC is required, the system should be designed so that the microprocessor accesses the CAM through its control port.

As shown in Figure 4-36, the MC92520 accesses the EAC device through an interface specifically designed to work with the match port of CAMs MCM69C432 or MCM69C232, requiring no external glue logic. A value to be matched is output on the EACDATA bus when $\overline{\text{EACSM}}$ is asserted. The match result is sampled from the EACDATA bus when $\overline{\text{EACOE}}$ is asserted. $\overline{\text{EACSM}}$ is intended to connect to the start match ($\overline{\text{SM}}$) pin of the CAM. $\overline{\text{EACOE}}$ is intended to connect to the output enable ($\overline{\text{G}}$) pin of the CAM. The EACDATA[31:0] pins should be pulled such that they have the value of 0x7FFF_FFFF if no CAM responds with a match. (This must be done using external pull-up/pull-down resistors.) Note that the MC92520 pre-charges the EACDATA bus with the value 0x7FFF_FFFF before asserting $\overline{\text{EACOE}}$, so that pull-up or pull-down resistors do not have to cause the EACDATA bus state to change on a non-match result.

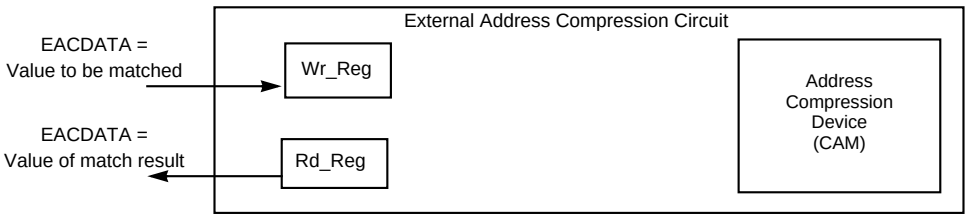


Figure 4-36. Example Implementation of External Address Compression

Due to the stringent timing requirements of the CAM clock input with respect to the CAM $\overline{\text{SM}}$ input, the EACCLK output is provided to clock the CAMs.

4.5 Microprocessor Interface

The microprocessor interface (MPIF) block provides for configuration of the MC92520 in addition to the transfer of cells between the microprocessor and the MC92520. A synchronous 32-bit slave interface is provided for a glueless connection to the MPC860 (PowerQUICC) or MPC8260 (PowerQUICC II) processors. Other processors may be used, but may require external glue logic for proper operation.

The following section describes the read and write operations on the processor port.

4.5.1 Processor Read and Write Operations

All MPIF inputs are sampled only on the rising edge of the MCLK input. To start a bus cycle, $\overline{\text{MSEL}}$ is asserted. On the rising MCLK edge during which $\overline{\text{MSEL}}$ is first sampled as asserted, the state of MADD and $\overline{\text{MWR}}$ are sampled and stored for the duration of the bus cycle. If a read cycle is indicated (by $\overline{\text{MWR}}$ being in a 1 state), the MPIF will assert $\overline{\text{MDTACK0}}$ (and $\overline{\text{MDTACK1}}$, if enabled) when read data is ready on the MDATA pins. If a

write cycle is indicated (by $\overline{MWR}$ being in a 0 state), the MDATA bus must contain valid write data by the rising edge MCLK during which $\overline{MWSH}$ and/or $\overline{MWSL}$ is first asserted. On a write cycle, MDATA must hold the write data until $\overline{MDTACK0}$ asserts (and $\overline{MDTACK1}$, if enabled). The bus cycle will end on the first rising edge MCLK during which $\overline{MENDCYC}$ is asserted and occurs during or after the assertion of the $\overline{MDTACK0}$ output. If $\overline{MSEL}$ remains asserted for the rising edge MCLK after the end of the bus cycle, a new bus cycle will begin.

$\overline{MDTACK0}$ and $\overline{MDTACK1}$ are tri-state output signals. They are driven by the MC92520 from the rising edge of MCLK, which first samples $\overline{MSEL}$ asserted (starting a bus cycle), and remains driven for one half MCLK cycle after the MCLK rising edge that ends the cycle. This extra half clock cycle of drive is used to pre-charge the $\overline{MDTACK}$ outputs back to a non-asserted state. Note that $\overline{MDTACK1}$ is driven only if it is enabled by MPCONR[MDC]. If enabled, $\overline{MDTACK1}$ is identical to $\overline{MDTACK0}$.

NOTE:

This interface is designed to work gluelessly with the MPC860 (PowerQUICC) and MPC8260 (PowerQUICC II) processors.

In the MPC860 application, $\overline{MENDCYC}$ is tied low, as the bus cycle may end as soon as $\overline{MDTACK0}$ asserts. The MPC860 general purpose chip-select machine (GPCM) may be used to generate a chip select output that can be tied directly to the $\overline{MSEL}$ input. The $\overline{MDTACK0}$ output may be used to drive the MPC860 transfer acknowledge ($\overline{TA}$) input. The other MPIF MADD inputs are connected to the address bus of the MPC860. The $\overline{MWR}$ input is connected to the MPC860 R/$\overline{W}$ output. The $\overline{MWSH}$ input is connected to either the $\overline{WE0}$ or $\overline{WE1}$ output of the MPC860. The $\overline{MWSL}$ input is connected to either the $\overline{WE2}$ or $\overline{WE3}$ output of the MPC860.

The only difference between the MPC860 application and the MPC8260 application is that the $\overline{MENDCYC}$ input is driven by $\overline{PSDVAL}$ in the MPC8260 application. This is used to notify the MPIF that the cycle is complete.

There are two $\overline{MDTACK}$ signals instead of one in order to enable glueless interface to systems in which there are two $\overline{MDTACK}$ signals and their combination conveys the bus width of the slave. $\overline{MDTACK1}$ is driven only when the $\overline{MDTACK0}$ signal is driven and when the MDTACK drive control bit (MDC) is set in the MPCONR register.

NOTE:

See Section 9.3 “Electrical and Physical Characteristics” for detailed timing information.

4.5.1.1 Processor Read Operations

Two basic types of microprocessor read operations are described in the following paragraphs: general register reads and cell extraction register reads.

4.5.1.1.1 General Register Read

When reading an MC92520 general register, or the external memory in setup mode, the read operation must be extended long enough to allow for synchronization to the cell processing clock (ZCLKIN). This increase in read time is accomplished by adding wait states. The MC92520 asserts $\overline{\text{MDTACK}}$ when the data is valid. Figure 4-37 describes a register read operation. The MC92520 samples MADD, $\overline{\text{MWR}}$, and $\overline{\text{MSEL}}$ on the MCLK rising edge. When MDATA is valid, the MC92520 asserts $\overline{\text{MDTACK}}$. The MC92520 continues to drive MDATA with valid read data until $\overline{\text{MENDCYC}}$ is asserted. $\overline{\text{MDTACK}}$ will assert for only 1 clock cycle. Once $\overline{\text{MSEL}}$ is negated, the MDATA bus will be tri-stated.

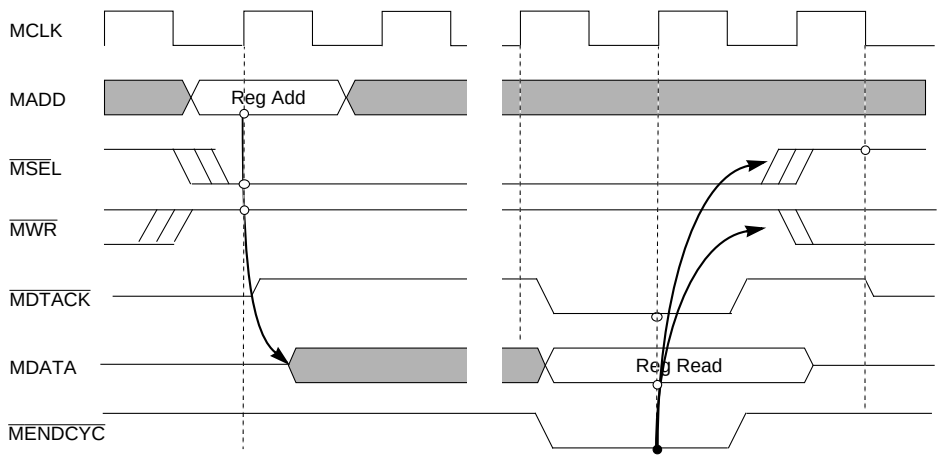


Figure 4-37. MC92520 Register Read Timing

4.5.1.1.2 Cell Extraction Register Read

To improve performance, the MC92520 cell extraction registers receive special treatment. Figure 4-38 describes a cell extraction register read operation. The MC92520 drives the register value on the MDATA bus, and asserts the $\overline{\text{MDTACK}}$ output, after detecting $\overline{\text{MSEL}}$ asserted, $\overline{\text{MWR}}$ negated, and MADD containing a CER address on the MCLK rising edge. The value on the MDATA pins is stable after propagation delays from this MCLK rising edge. The timing of MDATA is sufficient that it may be sampled at the next MCLK rising edge, so no wait states are necessary. The MC92520 continues to drive MDATA with valid read data until the cycle is ended using $\overline{\text{MENDCYC}}$. Once $\overline{\text{MSEL}}$ is negated, the MDATA bus is tri-stated.

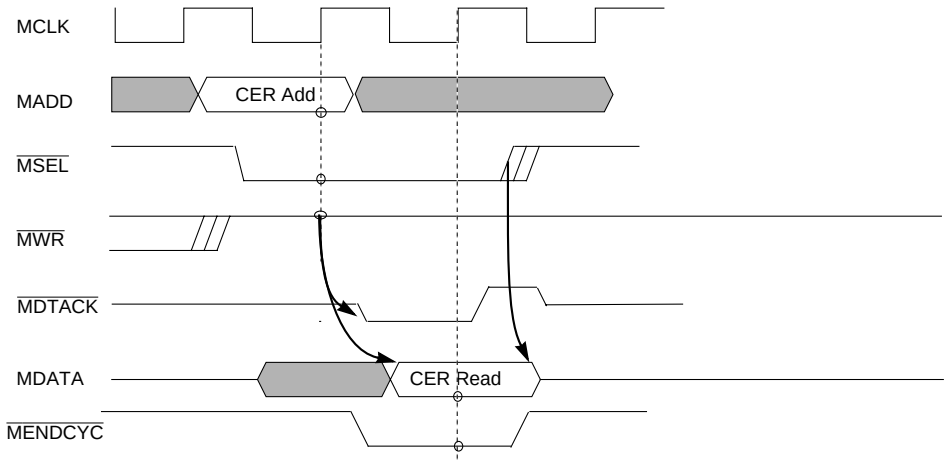


Figure 4-38. Cell Extraction Register Read Timing

4.5.1.2 Processor Write Operations

As with the processor read operation, two basic types of microprocessor write operations are described in the following paragraphs: general register writes and cell insertion register writes.

4.5.1.2.1 General Register Write

When writing to an MC92520 general register or to the external memory in setup mode, the write operation must be extended long enough to allow for synchronization to the cell processing clock (ZCLKIN). This is accomplished by adding wait states. The MC92520 asserts $\overline{\text{MDTACK}}$ when the register has been written. At this point, the write operation may be completed.

Figure 4-39 describes a register write operation. The MC92520 starts the write operation when $\overline{\text{MSEL}}$ is asserted by sampling MADD and $\overline{\text{MWR}}$. $\overline{\text{MWSH}}$ or $\overline{\text{MWSL}}$ should be asserted when MDATA is valid. When the register has been written, the MC92520 drives $\overline{\text{MDTACK}}$ low (asserted). The bus cycle ends when $\overline{\text{MENDCYC}}$ is asserted.

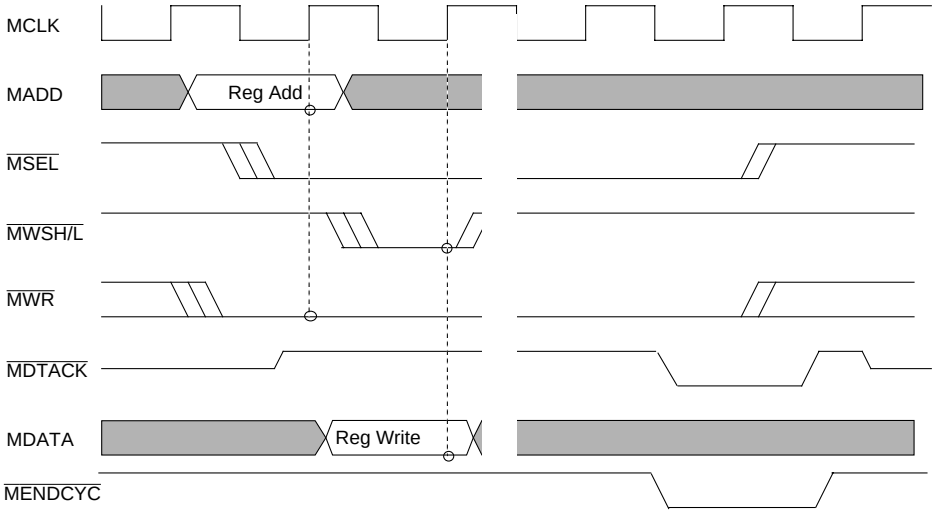


Figure 4-39. MC92520 Register Write Timing

4.5.1.2.2 Cell Insertion Register Write

To improve performance, the MC92520 cell insertion registers receive special treatment. Figure 4-40 describes a cell insertion register write operation. The MC92520 starts the write operation by sampling MADD, $\overline{\text{MWR}}$, and $\overline{\text{MSEL}}$ on the MCLK rising edge. The MC92520 samples MDATA on the first MCLK rising edge during which $\overline{\text{MWSH}}$ or $\overline{\text{MWSL}}$ are asserted. If the assume write strobe (AWS) bit is set in the microprocessor configuration register (MPCONR), MDTACK is asserted on the MCLK rising edge after the cycle begins. If AWS is not set, MDTACK does not assert until the MCLK rising edge after which $\overline{\text{MWSH}}$ and $\overline{\text{MWSL}}$ are asserted.

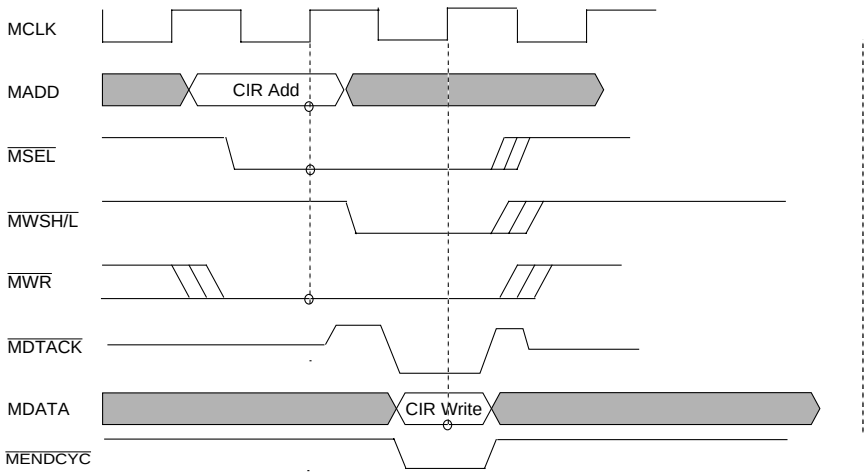


Figure 4-40. Cell Insertion Register Write Timing with AWS Set

4.5.1.3 Use of MENDCYC in Processor Operations

In all of the processor operations described in the previous sections, signal MENDCYC is used to end the bus cycle. On the MCLK rising edge during which the MC92520 has asserted MDTACK, the MENDCYC input will be sampled. If MENDCYC is asserted (low), then the cycle will end, and the MSEL input to the MC92520 must negate if no new cycle is to begin. If MENDCYC is negated (it is high), then the current processor operation will continue until MENDCYC is later asserted during a rising edge of MCLK, and in the case of a read cycle, any read data will be continuously driven until the processor operation ends. This function is intended for use with the MPC8260 (PowerQUICC II) processor, which needs to synchronize MDTACK before it can complete a processor operation. Note that MENDCYC may be tied low if the processor operations can always end once MDTACK is asserted, such as with MPC860 (PowerQUICC) applications.

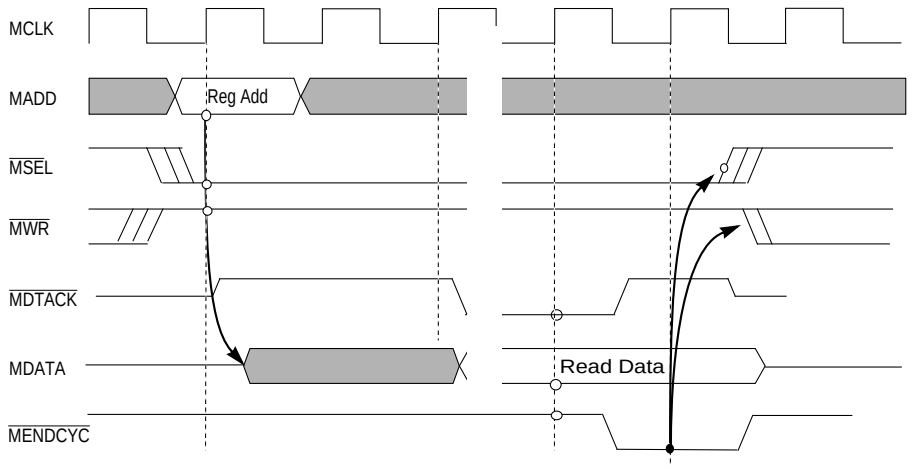


Figure 4-41. MC92520 MENDCYC Extending a Processor Operation

4.5.2 DMA Device Support

The MC92520 provides support for transferring cells using a DMA device without the need for a processor interrupt. It maintains two output signals which can serve as request lines for an external DMA device: $\overline{\text{MREQ0}}$ and $\overline{\text{MREQ1}}$. Each of these signals can be programmed for either of the following dma requests: $\overline{\text{MCIREQ}}$ and $\overline{\text{MCOREQ}}$. See Section 7.1.6.2 “Microprocessor Configuration Register (MPCONR)” for details. In the following sections, the $\overline{\text{MCIREQ}}$ and $\overline{\text{MCOREQ}}$ are referred to as signals instead of as internal DMA requests for the simplicity of the explanation only.

4.5.2.1 Cell Extraction with DMA Support

The output signal $\overline{\text{MCOREQ}}$ is provided by the MC92520 to be used as a request signal to a DMA device. $\overline{\text{MCOREQ}}$ is asserted at a rising edge of MCLK whenever the cell extraction register array contains a cell to be read. $\overline{\text{MCOREQ}}$ is negated following the end of the bus cycle in which CER14 has been read, as shown in Figure 4-42.

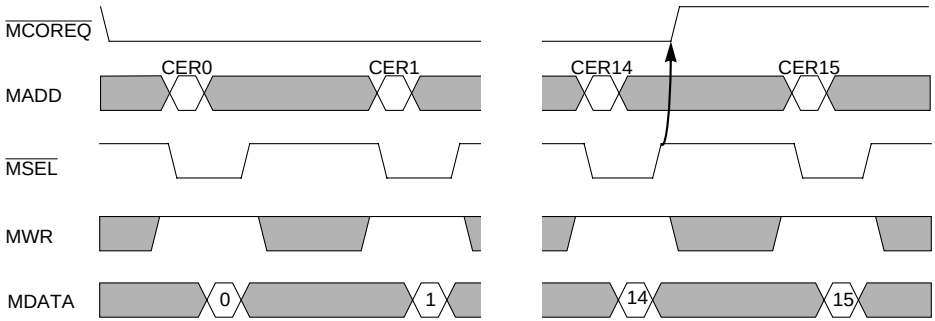


Figure 4-42. DMA Device Support on Cell Extraction Register

If CER15 is read before CER14, $\overline{\text{MCOREQ}}$ is negated after CER15 is read.

NOTE:

The MC92520 supports back-to-back cell extraction; if another cell is waiting to be read, $\overline{\text{MCOREQ}}$ is not negated in order to indicate that the next cell is also ready.

Figure 4-42 shows the negation of $\overline{\text{MCOREQ}}$ at the falling edge of MCLK following the negation of $\overline{\text{MSEL}}$.

4.5.2.2 Cell Insertion with DMA Support

The output signal $\overline{\text{MCIREQ}}$ is asserted at a rising edge of MCLK whenever the cell insertion register array is available for writing. The negation of $\overline{\text{MCIREQ}}$ depends on the method used to write to the cell insertion registers as explained below. If the cell is written using the cell insertion address space, $\overline{\text{MCIREQ}}$ is negated following the bus cycle in which CIR14 is written, as shown in Figure 4-43. If CIR15 is written before CIR14, $\overline{\text{MCIREQ}}$ is negated after CIR15 is written.

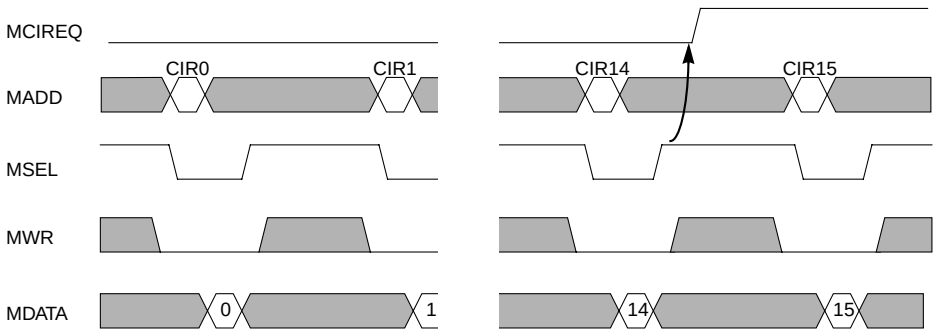


Figure 4-43. DMA Device Support on Cell Insertion Register

If the cell is written using the alternative cell insertion address space, $\overline{\text{MCIREQ}}$ is negated following the bus cycle in which ACIR0 has been written. If ACIR1 is written before ACIR0, $\overline{\text{MCIREQ}}$ is negated after ACIR1 has been written. $\overline{\text{MCIREQ}}$ is always negated at the rising edge of MCLK following the end of the bus cycle. This timing is identical to that of $\overline{\text{MCOREQ}}$ and is illustrated in Figure 4-42.

Chapter 5
Data Path Operation

The MC92520 provides two internal data paths:

- Ingress data path (data is received from the network)
- Egress data path (data is transmitted to the network)

The two data paths are described in detail in the following paragraphs.

5.1 Ingress Data Path Operation

The ingress data path can include the following steps:

1. Assembling cells
2. Compressing addresses
3. Performing context table lookups
4. Incrementing connection and link cell counters
5. Performing UPC/NPC processing
6. Inserting cells in the ingress flow
7. OAM processing
8. Adding switch overhead information
9. Performing address translation
10. Transferring cells to the switch

The cell flow through these steps is shown in Figure 5-1. Each step is described in the subsections below. During processing, the decision can be made to remove a cell from the cell flow based on the connection parameters or the OAM processing, among other reasons. Such a cell may be copied to the cell extraction queue.

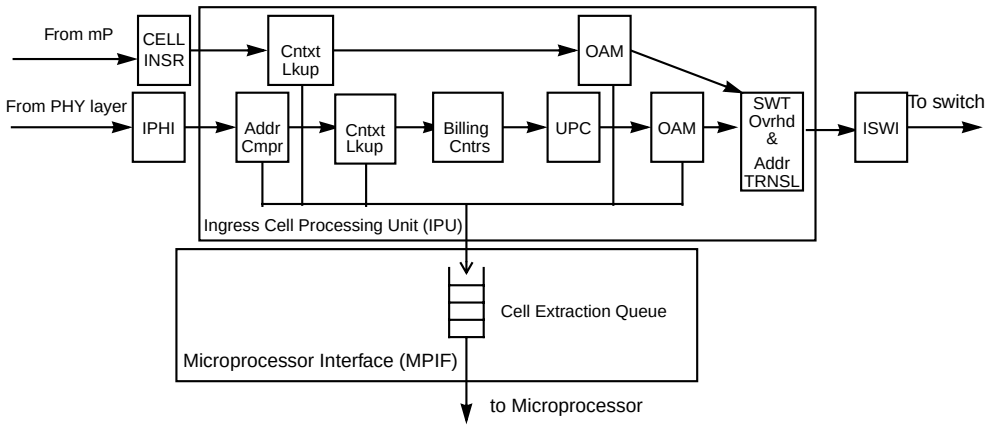


Figure 5-1. Ingress Data Path

5.1.1 Assembling Cells

The ingress physical-layer interface (IPHI) block receives cell data from the physical layer through the UTOPIA standard interface. The received data is assembled into cells because MC92520 processing is on a cell basis. The cells are held in a FIFO, and the cell processing block reads them as each cell slot passes. The MC92520 tracks how much data has been transferred while cells are assembled in its internal FIFO. If the input signal RXSOC is not asserted at the expected first data transfer of a new cell, a protocol error is reported by asserting the ingress PHY protocol handshake error (IR[IPHE]) bit.

The input data pins are parity protected as described in Section 4.2.3 “Multi-PHY Receive Interface (Ingress).” Parity checking by the MC92520 is optional and is enabled by the ingress PHY parity enable (IPPR) bit of the ingress PHY configuration register (IPHCR). When parity checking is enabled, the MC92520 expects RXPRTY to contain odd parity over RXDATA. Dependent on the setting of the ingress PHY wide data path (IPWD) bit, parity is checked for RXDATA bits [15:0] or [7:0]. If a parity error is detected on any of the four octets of the ATM header, the cell is discarded at the PHY interface. A parity error detected on the header error control (HEC) octet or UDF word does not cause the cell to be discarded.

If a parity error is detected in the cell payload, the treatment depends on the ingress payload parity enable (IPLP) bit. If IPLP is reset from IPHCR, the payload parity error is ignored, and the cell is processed normally. If IPLP is set, the cell is removed from the cell flow and copied to the cell extraction queue. When any parity error (header or payload) is detected, the error is reported by asserting the ingress PHY parity error (IR[IPPE]) bit. Parity errors on the payload of the cell are treated differently because some applications can correct

single- or multiple-bit errors using higher-level protocols. For these applications, discarding the cell because of a parity error in the payload would be harmful. A parity error in the header, however, can lead to misrouting of the cell, so the error is not ignored.

The HEC octet or UDF word received from the physical layer is not checked by the MC92520 and discarded. Because the MC92520 processes cells at a higher rate than they are received from the physical layer, the IPHI block cannot assemble a cell during every cell processing slot. When no complete cell is available, the IPHI block informs the ingress cell processing block (IPU), and the MC92520 inserts a hole in the cell flow. The IPHI block checks the cell header and recognizes the header of *unassigned* and *invalid* (PHY-layer idle) cells as defined in Table 5-1 and Table 5-2. Unassigned cells are treated as holes in the MC92520 cell flow. Invalid pattern cells may optionally (see Section 7.1.6.3 “Ingress PHY Configuration Register (IPHCR)”) be treated as holes or be copied to the processor for content analysis or counting.

Table 5-1. Pre-assigned Header Values at the UNI

Use	GFC	VPI	VCI	PTI	CLP
Unassigned cell	XXXX	0000_0000	0000_0000_0000_0000	XXX	0
Invalid (idle) cell	XXXX	0000_0000	0000_0000_0000_0000	XXX	1

X = don’t care bit

Table 5-2. Pre-assigned Header Values at the NNI

Use	VPI	VCI	PTI	CLP
Unassigned cell	0000_0000_0000	0000_0000_0000_0000	XXX	0
Invalid (idle) cell	0000_0000_0000	0000_0000_0000_0000	XXX	1

X = don’t care bit

5.1.2 Compressing Addresses

Ingress address compression maps the address fields in the received cell header to a pointer to the context table entry that relates to the cell’s virtual connection. The MC92520 supports two types of switching service: virtual path (VP) and virtual channel (VC). For VP switching, the address is the cell header’s virtual path identifier (VPI) field. For VC switching, the address consists of two cell header fields: the VPI field and the virtual channel identifier (VCI) field.

When the MC92520 supports multiple PHY devices, ATM addresses are mapped to context table entries for each PHY layer link. For this purpose, the link on which a cell arrived is treated as an additional address field. The VP switching address consists of the link and the VPI fields; the VC switching address consists of the link, VPI, and VCI fields.

If the link is at a UNI, as indicated by the bits of the UNI register (UNIR), the four most significant bits of the VPI mask (VPM) field of the ingress link register should be set to 0,

because the actual VPI consists of only 8 bits. Optionally, the MC92520 also checks that the received GFC bits are all 0 (see Section 7.1.6.11 “Ingress Processing Configuration Register (IPCR)”); if not, the cell is copied to the cell extraction queue.

NOTE:

Cells for which no valid connection is found during address compression (cells for inactive connections), are removed from the cell flow and copied to the cell extraction queue.

5.1.2.1 Address Compression Options

The MC92520 supports two methods for performing address compression:

1. Table lookup using restricted address spaces
2. External address compression

The choice of method is user programmable per link by using the address compression method (ACM) field in the ingress link registers (ILNK0–ILNK15). Table 5-3 summarizes the available options.

Table 5-3. Address Compression Options

ACM	VP Table Lookup	VC Table Lookup	External Address Compression
00	X	X	
01	X		
10			X
11	X		X

The ACM fields and their functions are as follows:

ACM = 00 — This is a two-stage lookup method. First, the VP table is checked. If VP switching is performed, the VP table contains the ingress connection identifier (ICI). If no VP switching is performed, the VP table contains a pointer to the VC table that contains the ICI.

ACM = 01 — This value is intended for applications in which only VP switching is performed. In this case, a one-stage lookup consults only the VP table.

ACM = 10 — This causes address compression to be done externally. No table lookups are performed.

ACM = 11 — This causes address compression to be done externally and the VP table to be consulted. If the VP table contains a valid ICI, this value is used instead of the external result. More details are provided in the following sections.

5.1.2.2 VPI/VCI Table Lookup

When some of the VPI or VCI bits are not allocated, the address range can be reduced enough to make a table lookup scheme practical. The method described here provides a great deal of flexibility because the number and location of the allocated bits of the VPI may be specified per physical link, and the number and location of the allocated bits of the VCI may be specified per VPI. Following the description of the table lookup method are a number of optional variations on the scheme that reduce the external memory requirements. An overview of the table lookup scheme is provided in Figure 5-2. A detailed description follows.

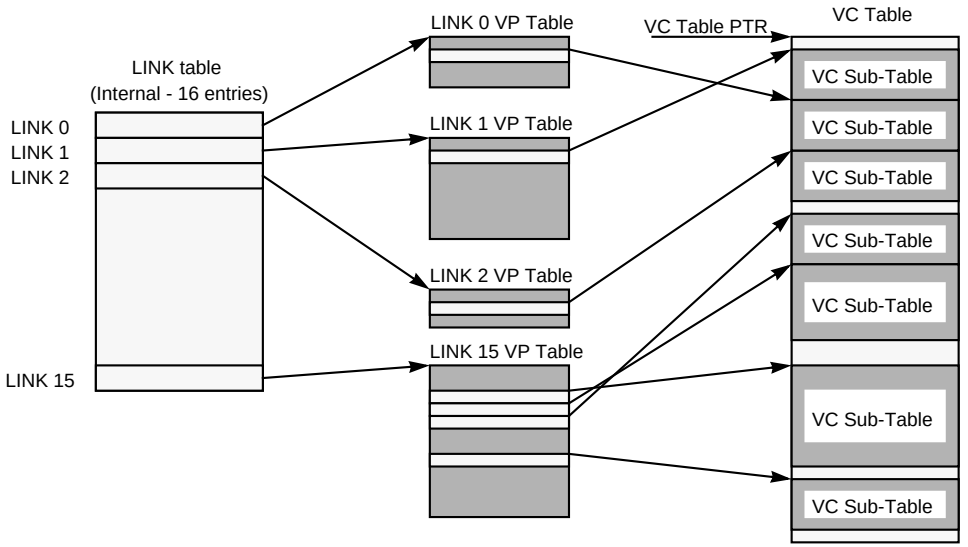


Figure 5-2. Address Compression Tables

5.1.2.2.1 Link Table

The internal link table, which is implemented by 16 registers (see Section 7.1.5.13 “Ingress Link Registers (ILNK0–ILNK15),” determines whether the associated physical links are enabled (and polled in multi-PHY mode), and how to perform the VP table lookup for cells arriving from each of the physical links. The logical structure of the link table for the table lookup scheme is shown in Figure 5-3.

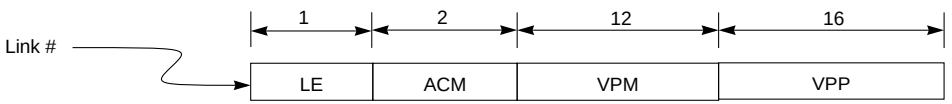


Figure 5-3. Link Table Logical Structure

The first step of address compression is reading the ingress link register that holds the cell's link number address (the number of the link from which the current cell arrived). Secondly, the link enable (LE) bit is checked. If LE is clear, the cell came from a link that was disabled after the cell was already queued in the PHY receive FIFO. Such a cell will be removed from the cell flow and copied to the cell extraction queue. If LE is set, address compression continues. Next, the external memory address of the VP table entry is determined as illustrated in Figure 5-4.

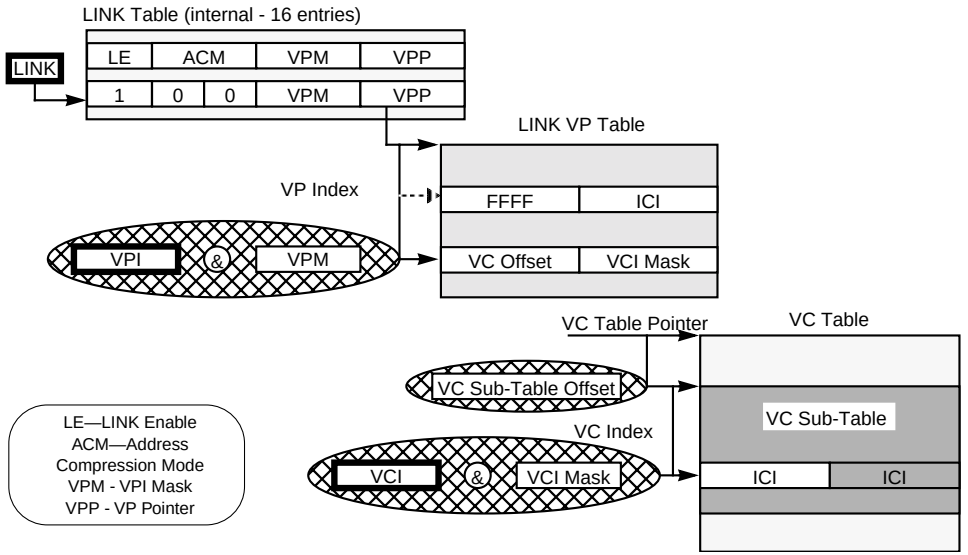


Figure 5-4. Full Table Lookup Scheme

Determining the EM address of a VP table record includes the following processes:

- The VP pointer (VPP) field of the ingress link register is shifted eight bits to the left as shown in Figure 5-5.

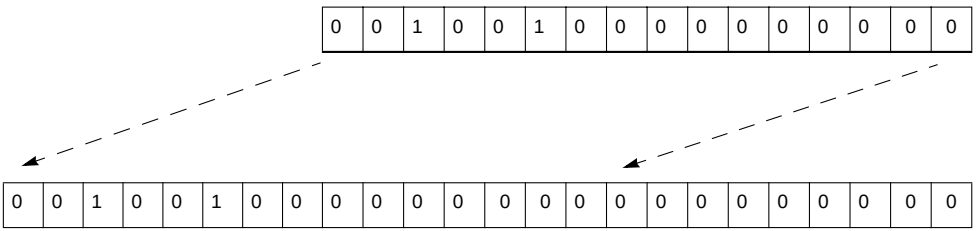


Figure 5-5. Deriving Address of Link VP Table

- The VPI mask (VPM) field of the ingress link register is used to choose the allocated bits of the VPI which are then aligned to the right to form the VP Index. For instance, if the VPM is 0x805, bits 0, 2, and 11 of the VPI will be used to form the 3-bit index to the record of the VP connection within the VP sub-table as shown in Figure 5-6.

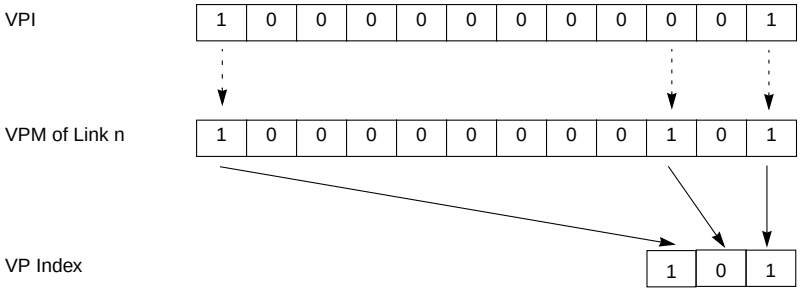


Figure 5-6. Deriving the VP Index

Optionally, the MC92520 also checks that all unallocated bits of the VPI are 0 (see Section 7.1.6.11 “Ingress Processing Configuration Register (IPCR)”); if not, the cell is removed from the cell flow and copied to the cell extraction queue. The size of each link’s VP table should correspond to the number of bits that are set in the link’s VPM.

- The MC92520 computes the actual external memory address by adding the VP index to the link’s VPP. But before the VP index is added to the VPP, it is shifted left two bits to allow for the size of each record, which is one long word. Examples of these calculations can be found in Table 5-4. Note that the second row of Table 5-4 uses the values shown in Figure 5-5 and Figure 5-6.

Table 5-4. VP Table Address Calculations (VC Lookup Enabled)

VPP[1]	Table Size	VPM	VPI	VP Index[2]	EM Address of VP Table Entry
0x2400	32 records	0x037	0x019[3]	0x9	0x2400 << 8 = 0x240000 0x9 << 2 = 0x000024 0x240024
0x2400	8 records	0x805	0x801	0x5	0x2400 << 8 = 0x240000 0x5 << 2 = 0x000014 0x240014
0x2403	256 records	0x0FF	0x086	0x86	0x2403 << 8 = 0x240300 0x86 << 2 = 0x000218 0x240518

1 The VPP, in units of 256 bytes, is shifted eight bits to the left to produce the actual EM address of the VP table.

2 Because the record size is one long word, the VP index is shifted two bits to the left.

3 If unallocated bits are checked, the address compression would fail for this cell. It would be removed from the cell flow and copied to the cell extraction queue.

5.1.2.2.2 VP Table

If VP switching is performed on the VP connection, the VP table entry appears as shown in Figure 5-7. The reserved value of all 1’s in the VC sub-table offset field is used to indicate that VP switching is performed. The ICI points to the context entry for the VP connection unless it contains the reserved value of all 1’s, in which case the connection is not active.

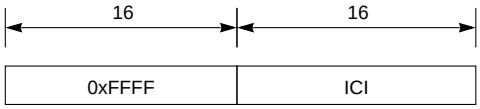


Figure 5-7. VP Table Record Logical Structure for VP Switching

If VC switching is performed, the VP table entry defines how to perform the VC table lookup, and it appears as shown in Figure 5-8.

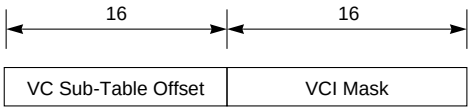


Figure 5-8. VP Table Record Logical Structure as Pointer to VC Table

Determining the external memory address of a VC table record is illustrated in Figure 5-4 and includes these processes:

- The VC table pointer as defined in the VC table pointer register (VCTP) is the base address of the VC table. Each VPI has its own VC sub-table which is offset from the base address by the VC sub-table offset field of the VP table entry which is in units of long words.
- The VCI mask field of the VP table entry is used to choose the allocated bits of the VCI which are then aligned to the right to form the VC index. For instance, if the VCI mask is 0x1805, bits 12, 11, 2, and 0 of the VCI form the four-bit index to the record of the VC connection within the VC sub-table.

  The MC92520 also optionally checks that all un-allocated bits of the VCI are 0 (see Section 7.1.6.11 “Ingress Processing Configuration Register (IPCR)”); if not, the cell is removed from the cell flow and copied to the cell extraction queue. The size of each VPI’s VC sub-table should correspond to the number of bits that are set in the VPI’s VCI mask as shown in Table 5-5.
- The actual external memory address is computed by adding the VC table pointer, the VPI’s VC sub-table offset, and the VC index. Examples of these calculations can be found in Table 5-5.

Table 5-5. VC Table Address Calculations

VC Table Pointer[1]	VC Sub-Table Offset[2]	Table Size	VCI Mask	VCI	VC Index[3]	EM Address of VC Table Entry
0x8400	0x100	32 records	0x0037	0x0019[4]	0x9	0x8400 << 8 = 0x840000 0x100 << 2 = 0x000400 0x9 << 1 = 0x000012 0x840412
0x8403	0x201	8 records	0x0805	0x0801	0x5	0x8403 << 8 = 0x840300 0x201 << 2 = 0x000804 0x5 << 1 = 0x00000A 0x840B0E

1 The VC table pointer is in units of 256 bytes. It is shifted eight bits to the left to produce the actual EM address.

2 The VC sub-table offset is in units of long words. It is shifted two bits to the left.

3 Because the record size is two bytes, the VC index is shifted one bit to the left.

4 If unallocated bits are checked, the address compression would fail for this cell. It would be removed from the cell flow and copied to the cell extraction queue.

5.1.2.2.3 VC Table

The VC table entry appears as shown in Figure 5-9. The ingress connection identifier (ICI) points to a valid VCC entry in the context table unless it contains the reserved value of all 1's, in which case the connection is not active. Cells belonging to inactive connections are removed from the cell flow and copied to the cell extraction queue.

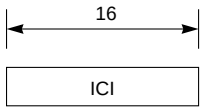


Figure 5-9. VC Table Record Logical Structure

5.1.2.3 VPI-only Table Lookup

If the ACM field in the ingress link register is 01, the VC table lookup is skipped, and no VC sub-tables exist for this link. When the ACM field in the ingress link register is 01, and the ICI field in the VP table is all 1's, no context table entry exists for the cell. The cell is removed from the cell flow and copied to the cell extraction queue as an inactive cell. Additionally, the VP sub-table for this link is condensed by 50%, because all the entries are of the form of Figure 5-7 and two entries are placed in each 32-bit word as shown in Figure 5-10. This option should be used only if all the connections on this link require VP switching.

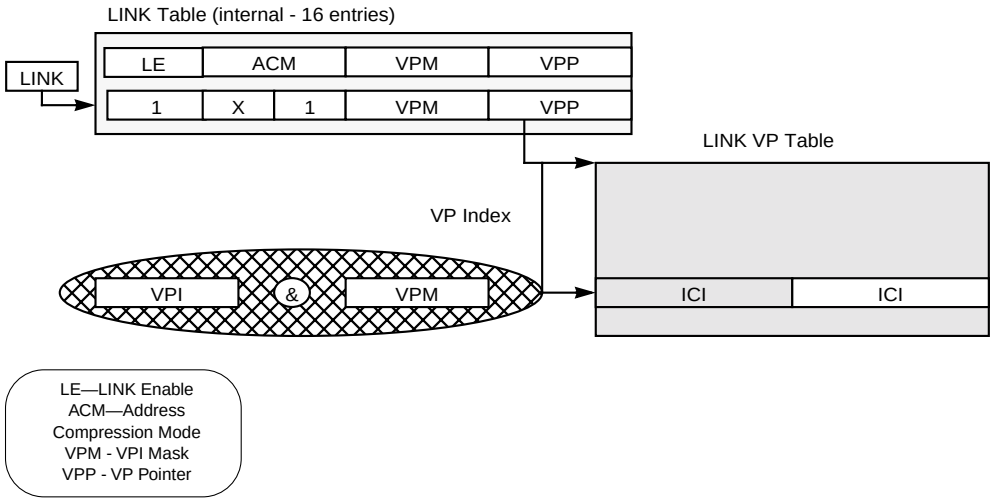


Figure 5-10. Address Compression with VPI-only Table Lookup

Table 5-6 shows an example of the calculation of the VP table address when there is no VC table lookup. In this case, the VP index is shifted one bit to the right before adding it to the VPP, because each long word contains two records.

Table 5-6. VP Table Address Calculations (VPI-only)

VPP[1]	Table Size	VPM	VPI	VP Index[2]	EM Address of VP Table Entry
0x2400	32 records	0x037	0x019	0x9	0x2400 << 8 = 0x240000 0x9 << 1 = 0x000012 0x240012

1 The VPP is in units of 256 bytes It is shifted 8 bits to the left to produce the actual EM address of the VP table.

2 Because the record size is two bytes, the VP Index is shifted one bit to the left.

5.1.2.4 External Address Compression

The external address compression method allows the user total flexibility in performing ingress address compression. External compression requires that the MSB of ILINK*n*[ACM] be set. Setting ACM to 10 causes compression to be done externally with no table lookups, and setting ACM to 11 (setting the LSB also) indicates that VP lookup is to be performed as a first stage before providing the address for external compression. In either case, the ingress link register (addressed by the link number from which the current cell arrived) is read as the first step of the address compression, and the link enable (LE) bit is checked. If LE is clear, the cell came from a link that was disabled after the cell was already queued in the PHY receive FIFO. Such a cell is removed from the cell flow and copied to the cell extraction queue. If LE is set, address compression continues.

Next, the MC92520 performs an EAC write operation requesting address compression for

a given ATM address qualified by the physical link the cell was received on. A specified number of ACLK periods, N_{EAC} (≥ 40), after the EAC write access, the MC92520 performs a read operation from the EAC device and receives a response as illustrated in Figure 5-11.

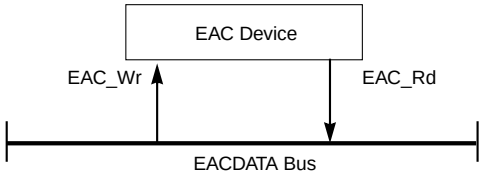


Figure 5-11. External Address Compression Events

5.1.2.4.1 EAC Write Structure

The address of a cell received from the PHY layer is presented by the MC92520 on the EACDATA bus using the structure shown in Figure 5-12:

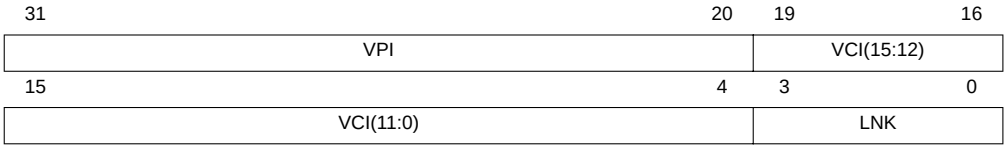


Figure 5-12. EAC Write (Request) Structure

Table 5-7. EAC Write (Request) Field Descriptions

Bits	Name	Description
31–20	VPI	Virtual path identifier. This field is taken from the ATM cell header.
19–4	VCI	Virtual channel identifier. This field is taken from the ATM cell header.
3–0	LNK	Physical link number. This field is valid only when the MC92520 is supporting multiple PHY devices. It contains the number of the physical link from which the cell was received. When only one PHY device is supported, this field is set to 0000.

5.1.2.4.2 EAC Read Structure

The external address compression device's response presented on the EACDATA bus must comply with the structure shown in Figure 5-13.

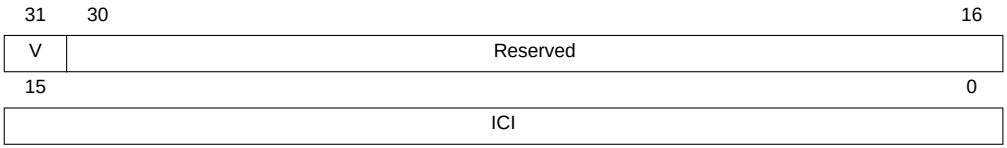


Figure 5-13. EAC Read (Response) Structure

Table 5-8. EAC Read (Response) Field Descriptons

Bits	Name	Description
31	V	Valid. If the valid (V) bit is set, the ingress connection identifier (ICI) points to a valid entry in the context parameters table unless it contains the reserved value of all 1's. If the valid bit is reset, or if the ICI is all 1's, there is no context parameters table entry for the cell, and it is removed from the cell flow and copied to the cell extraction queue as an inactive cell.
30–16	—	Reserved
15–0	ICI	Ingress connection identifier. This field contains an index to the context parameter table records of the connection to which the cell belongs. The reserved value of all ones indicates that the address compression has failed even if the valid bit is set. The reserved value of all ones should not be used as the connection identifier of any connection.

Details of the external address compression device interface are provided in Section 4.4.3 "External Address Compression Device Interface."

5.1.2.4.3 External Address Compression with VPI Lookup

When using the external address compression method with VPI lookup, VPI compression is performed identically to the table lookup method with VPI-only table lookup (see Figure 5-10). In addition, the link-VPI-VCI combination is written to the external address compression device. If the VP lookup was successful (ICI not all 1's), the ICI read from the VP table is used.

5.1.2.4.4 VPI Lookup Disable

If the ACM field in the ingress link register is 10, the VPI table lookup is skipped entirely, and no VP table exists for this link.

5.1.3 Performing Context Table Lookups

Once the ingress connection identifier (ICI) of the cell is known, the context parameters can be read from the context parameters table. The structure of the context parameters table is presented in Section 7.2.3 "Context Parameters Table" and how the context parameter table is used by the MC92520 is presented in Chapter 6, "Protocol Support."

5.1.4 Incrementing Connection and Link Cell Counters

If the processed cell was received from the physical layer (not inserted internally), one of the connection cell counters from the ingress billing counters table is incremented, unless the table does not exist—see Section 7.1.5.20 "Ingress Processing Control Register (IPLR)." One of the link cell counters from the ingress link counters table is also incremented if the table exists—see Section 7.1.6.15 "General Configuration Register (GCR)." The appropriate counter is chosen based on the CLP bit and whether the cell is an OAM cell.

5.1.5 Performing UPC/NPC Processing

The MC92520 performs the UPC/NPC function for the ingress flow if the UPC flow (UPCF) bit in the ATMC CFB configuration register (ACR) is reset. See Section 6.1 “Traditional UPC/NPC Support.”

5.1.6 Inserting Cells in the Ingress Flow

The cell insertion rate is paced by a single leaky bucket to ensure that the switch-side is not flooded with inserted cells beyond its capacity. Beyond insertion rate pacing, the MC92520 can be configured to give either received cells a higher priority than inserted cells, or inserted cells a higher priority than received cells. For more information see the ingress insertion priority (IIP) bit description in Section 7.1.6.11, “Ingress Processing Configuration Register (IPCR).”

The parameters of the leaky bucket are determined by the ingress insertion leaky bucket register (IILB). The bucket contents’ value, contained in the ingress insertion bucket fill register (IIBF), is incremented by the ingress average insertion period (IAIP) when a cell is inserted in the ingress cell flow and is decremented by one in each cell slot. The MC92520 inserts a cell in an available hole only if the bucket contents’ value in the IIBF is smaller than the ingress insertion bucket limit (IIBL).

The general algorithm internal to the MC92520 is thus:

```
IIBF = 0 at reset
For each cell slot
        IIBF = max(IIBF - 1, 0)
        If (IIBF < IIBL && cell insertion queue not empty)
                Insert cell
                IIBF = IIBF + IAIP
        EndIf
EndFor
```

NOTE:

The IIBL field is zero after reset and must be written with a non-zero value to enable cell insertion.

The IAIP consists of a 12-bit integer part and a 4-bit fractional part that provide for values as large as 4095 cell processing times (equivalent to inserted cells being 0.024% of the cell flow) with a precision of 1/16 of a cell processing time. At the typical value of 100 cell processing times (1% of the cell flow), the precision is 0.0006 percentage point. Note that programming the IAIP with a value of zero provides unlimited cell insertion. The 16-bit IIBL provides for a back-to-back burst of 16 cells when using the maximum value of IAIP and proportionately larger bursts when using smaller values of IAIP. The bursts occur while the IIBF value is being incremented to the threshold set by the IIBL.

For example, if the inserted cells are 1.3% of the cell flow the IAIP is 1/.013 = 76.92 that rounds to 76 15/16, or 0x04C.F. If the switch can handle a burst of up to 10 cells, the IIBL

is defined to be one IAIP value less than the required bucket size. Therefore, in our example the IIBL should be 10 – 1 = 9 times the IAIP value: 76.94 × 9 = 692, or 0x02B4.

For the typical insertion rate of 1% of the cell flow, the IAIP is 1/.01 = 100, or 0x064.0. If no back-to-back cell burst is desired, the IIBL may be defined as a value of less than 100, so that the value of IIBF does not satisfy the condition (IIBF < IIBL), discussed above, immediately after the first inserted cell (because IIBF has been incremented by IAIP).

The types of cells that can be inserted in the ingress cell flow are:

- Operations and maintenance (OAM) cells generated internally by the MC92520 including:
  - Alarm indication signal (AIS) cells
  - Remote defect indicator (RDI) cells
  - Continuity check cells
  - PM forward monitoring cells
- OAM cells generated by the microprocessor
- Other cells generated by the microprocessor

The various types of cells that can be inserted in the ingress cell flow are classified by their insertion priority and held in separate queues. The insertion priorities are (from highest to lowest):

1. PM forward monitoring cells generated internally by the MC92520
2. Cells from the microprocessor
3. AIS, RDI, and CC cells generated internally by the MC92520

The data structure of inserted cells from the microprocessor is provided in Section 7.3.1 “Inserted Cell Structure.” Note that inserted cells have their connection identifier explicitly available, so they do not undergo address compression. Inserted cells are not presented to the UPC/NPC mechanism, nor are they counted in the connection counters.

5.1.7 OAM Processing

If the ingress copy all (ICA) cells bit is set, the cell is added to the cell extraction queue to be transferred to the microprocessor. If the ingress remove all (IRA) cells bit is set, the cell is removed from the cell flow after undergoing UPC/NPC and OAM processing. The option exists to copy or remove those cells whose VCI is identified as “reserved.” See Section 7.1.6.29 “Ingress VCI Copy Register (IVCR)” and Section 7.1.6.31 “Ingress VCI Remove Register (IVRR)” for details. This option is enabled on a connection basis by the ingress VCR/VRR registers enable (IVRE) bit of the ingress parameters word. There is also an option to copy those cells whose PTI value is 110 or 111 to the cell extraction queue. This option is controlled by the ingress PTI 6 copy (IP6C) and ingress PTI 7 copy (IF7C) bits of the ingress parameters word.

The CRC-10 field of received OAM cells is checked. If an error is detected in this field, or if the cell is an illegal OAM cell (see Section 6.5.3.1 “Illegal OAM Cells”), no OAM processing is performed on the cell. The cell is removed from the cell flow and copied to the cell extraction queue. All OAM cells are further classified by the OAM cell type and OAM function type fields (see Section 6.5, “Operations and Maintenance (OAM) Support and Table 6-6). The OAM cells that receive special processing are:

- Cells received from the physical layer:
  - AIS (see Section 6.5.4.1.1 “Alarm Indication Signal Cells”
  - RDI (see Section 6.5.4.1.2 “Remote Defect Indicator Cells”)
  - Loop-back (see Section 6.5.4.2.1 “Loopback Cell Format”**)**
  - Continuity check (see Section 6.5.4.1.3 “Continuity Check Cells”**)**
  - Forward monitoring (see Section 6.5.6 “Performance Monitoring”
  - Backward reporting (see Section 6.5.6 “Performance Monitoring”
  - If this is a segment or connection termination point (see Section 6.5.7 “Activation/Deactivation OAM Cells”) of the OAM flow, the segment or end-to-end OAM cell is removed from the cell flow.
- Inserted cells:
  - Forward monitoring from the processor (see Section 6.5.6 “Performance Monitoring”**)**
  - Forward monitoring internally generated (see Section 6.5.6 “Performance Monitoring”**)**

When a user data cell is processed, the traffic bits (see Section 6.5.4.1.3 “Continuity Check Cells”) are set. User data cells that belong to an active PM block test are processed as described in Section 6.5.6 “Performance Monitoring.”

5.1.8 Adding Switch Overhead Information

The source of the switch overhead information provided by the MC92520 is the context parameters table entry for the connection. This information is appended to the cell as shown in Figure 5-14, Figure 5-15, and Figure 5-16. The overhead information is transferred before the cell is transmitted, in most-significant byte or word order (left-to-right in the tables). The number of long words of switch parameters (0, 1, or 2) is indicated by the ingress switch parameter control (SPC) field defined in Section 7.1.6.14 “ATMC CFB Configuration Register (ACR).” The size and composition of the data structure extracted from the switch parameters and transferred to the switch is controlled by three fields of the ingress switch interface configuration register (ISWCR): The ingress switch number of bytes (ISNB) field, the ingress switch HEC field (ISHF), and the ingress wide data path (ISWD) bit.

Figure 5-14 shows the data structure transferred to the switch as a function of ISNB when no HEC octet (8-bit mode) or UDF word (16-bit mode) is inserted (ISHF = 00). Note that for 16-bit mode operation (ISWD = 1), ISNB must be an even value.

ISNB	Transmitted Octets (shaded)																
	Switch Parameters 2				Switch Parameters 1				Switch Parameters 0				ATM Header				Payload
0000																	
1111																	
1110																	
1101																	
1100																	
1011																	
1010																	
1001																	
1000																	
0111																	
0110																	
0101																	
0100																	

Figure 5-14. Data Structure with No HEC/UDF (ISHF=00)

Figure 5-15 shows the data structure transferred to the switch as a function of ISNB when the HEC octet (8-bit mode) or UDF word (16-bit mode) is inserted and presented as zero (ISHF = 10). In this case, the only purpose of the HEC octet/UDF word is to provide for compatibility with switches that expect the HEC octet/UDF word in their cell structure. Note that for 16-bit mode operation (ISWD = 1), ISNB must be an even value.

NOTE:

The MSB (8-bit mode) or MSW (16-bit mode) of the Switch Parameters 2 column is unused even if 64 bytes are transferred. to the switch.

ISNB	Transmitted Octets (shaded)																	
	Switch Parameters 2				Switch Parameters 1				Switch Parameters 0				ATM Header					Payload
0000																	0	
1111																	0	
1110																	0	
1101																	0	
1100																	0	
1011																	0	
1010																	0	
1001																	0	
1000																	0	
0111																	0	
0110																	0	
0101																	0	

Figure 5-15. Data Structure with HEC/UDF = 0 (ISHF = 10)

Figure 5-16 shows the data structure transferred to the switch as a function of ISNB when the switch interface is configured for an 8-bit wide data path (ISWD = 0) and the HEC octet is taken from the switch parameters (ISHF = 11). Depending on the number of used switch parameters defined in the ingress switch parameter control (SPC) field of the ATMC configuration register (ACR), the HEC octet is fetched from the MSB of the highest numbered switch parameter and inserted after the ATM header. For example, octet X_2 is fetched as HEC from switch parameter 2 if SPC is configured to b11. Note that the HEC octet may also be transmitted as cell overhead if the ISNB field is configured accordingly.

ISNB	Transmitted Octets (shaded)																	
	Switch Parameters 2				Switch Parameters 1				Switch Parameters 0				ATM Header					Payload
0000	X_2				X_1				X_0								X_N	
1111	X_2				X_1				X_0								X_N	
1110	X_2				X_1				X_0								X_N	
1101	X_2				X_1				X_0								X_N	
1100	X_2				X_1				X_0								X_N	
1011	X_2				X_1				X_0								X_N	
1010	X_2				X_1				X_0								X_N	
1001	X_2				X_1				X_0								X_N	
1000	X_2				X_1				X_0								X_N	
0111	X_2				X_1				X_0								X_N	
0110	X_2				X_1				X_0								X_N	
0101	X_2				X_1				X_0								X_N	

Figure 5-16. Data Structure with HEC Octet from Switch Parameter (ISHF = 11)

Figure 5-17 shows the data structure transferred to the switch as a function of ISNB when the switch interface is configured for a 16-bit wide data path (ISWD = 1) and the UDF word is taken from the switch parameters (ISHF = 11). Depending on the number of used switch parameters defined in the ingress switch parameter control (SPC) field of the ATMC

configuration register (ACR), the UDF word is fetched from the MSW of the highest numbered switch parameter and inserted after the ATM header. For example, word X_2 is fetched as UDF from switch parameter 2 if SPC is configured to b11. Note that the UDF word may also be transmitted as cell overhead if the ISNB field is configured accordingly. For a 16-bit wide data path, ISNB must be even.

ISNB	Transmitted Octets (shaded)									
	Switch Parameters 2		Switch Parameters 1		Switch Parameters 0		ATM Header			Payload
0000	X_2		X_1		X_0				X_N	
1110	X_2		X_1		X_0				X_N	
1100	X_2		X_1		X_0				X_N	
1010	X_2		X_1		X_0				X_N	
1000	X_2		X_1		X_0				X_N	
0110	X_2		X_1		X_0				X_N	

Figure 5-17. Data Structure with UDF Word from Switch Parameters (ISHF = 11)

5.1.9 Performing Address Translation

The MC92520 optionally performs address translation on the ingress cell flow. The new address fields are taken from the ingress translation address word of the context parameter table in the external memory. The ingress address translation VPI enable (IAPE) and ingress address translation VCI enable (IACE) fields of the ingress processing configuration register (IPCR) determine exactly which fields of the ATM cell header are overwritten. See Section 7.1.6.11 “Ingress Processing Configuration Register (IPCR)” for more details.

5.1.10 Transferring Cells to the Switch

The ingress switch interface (ISWI) block receives cells from the cell processing block, queues them, and transfers the data structure to the switch. The switch interface signals are identical to the UTOPIA level 1 receive interface with the MC92520 playing the role of the PHY layer and the switch playing the role of the ATM layer. The switch interface signals are clocked by an independent clock signal, SRXCLK. To synchronize the PHY and ATM layers, the switch is required to accept cells from the MC92520 when they are presented on the interface with a delay of up to one cell slot. Note that the cells may be presented at a higher rate than they are received from the PHY layer due to cell insertion. The switch must be capable of receiving the cells at a sustained rate of one cell per cell slot. Otherwise, the cells may back up in the MC92520, processing is halted, and cells are not accepted from the PHY layer. Although the maximum sustained rate is one cell per cell slot, the rate can be limited by the insertion pacing mechanism described in Section 5.1.6, “Inserting Cells in the Ingress Flow.”

5.2 Egress Data Path Operation

The egress data path includes the following steps:

1. Transferring cells from the switch
2. Multicast identifier translation (if necessary)
3. Inserting cells into the egress flow
4. Performing context table lookups
5. Performing UPC processing
6. Performing OAM processing
7. Translating addresses
8. Incrementing connection and link cell counters
9. Transmitting cells to the physical layer

The cell flow through these steps is shown in Figure 5-18. Each step is described in the subsections below. During the processing, the decision can be made to remove a cell from the cell flow for any of several reasons. Such a cell may be copied to the cell extraction queue.

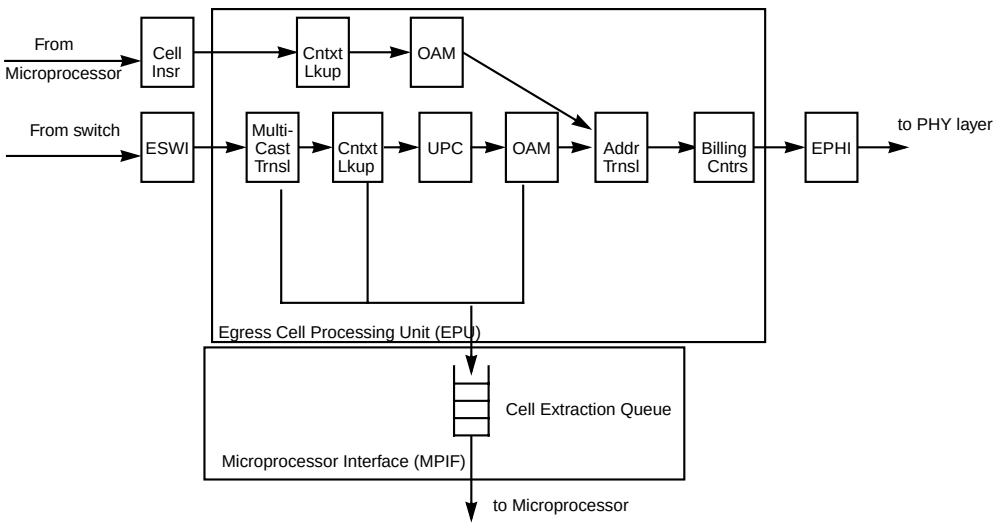


Figure 5-18. Egress Data Path

5.2.1 Transferring Cells from the Switch

The egress switch interface (ESWI) block contains a cell FIFO. Data is received from the switch at the rate of one byte per clock cycle in 8-bit mode (ESWD = 0) or one word per

clock cycle in 16-bit mode (ESWD = 1). The data structure received from the switch includes overhead routing information in addition to the ATM cell. While such a cell is transferred to the MC92520, it is transformed into an internal data structure and presented to the egress cell processing block. The MC92520 supports a UTOPIA single-PHY or multi-PHY switch interface with the MC92520 playing the role of the PHY layer and the switch playing the role of the ATM layer. The switch interface signals are clocked by an independent clock signal, STXCLK. The input signal STXSOC is used to delineate the beginning of a cell.

The input data pins are parity protected. In 8-bit mode operation (ESWD = 0), parity checking is performed on a byte basis. In 16-bit mode operation (ESWD = 1), parity checking is performed on a word basis. If a parity error is detected on the input pins, the error is reported by asserting the egress switch parity error (IR[ESPE]) bit. If the parity error occurs on a byte or word containing any of the overhead fields used by the MC92520 or on a byte or word of the cell header, the cell is discarded. If the parity error occurs on a payload byte or word, the cell is optionally discarded. If a protocol error is detected on the input pins, the current cell is discarded and the error is reported by asserting the egress switch protocol handshake error (IR[ESHE]) bit.

The ESWI block contains a cell FIFO to assemble the data received from the switch and synchronize the cells to the cell processing time of the egress cell processing block. If a single-PHY interface is selected, the FIFO size is programmed to be either four or six cells using the egress switch FIFO control (ESFC) bit (see Section 7.1.6.6 “Egress Switch Interface Configuration Register (ESWCR)” for more information). If a multi-PHY interface is used, the FIFO size can be programmed to any depth of 1 to 16 via the egress FIFO depth (ESFD) field in the egress switch configuration register 1 (ESWCR1). The FIFO is read by the cell processing block at a rate limited by the PHY layer and by cell insertion. When the ESWI FIFO is full, the MC92520 will not accept cells from the switch by deasserting STXCLAV. If a multi-PHY interface is selected, the MC92520 also considers the depth of the per-PHY FIFOs on the PHY-side interface, how many cells are currently processed, and how many cell transfers have been committed on other PHY ports via polling.

The number of bytes in the cell data structure received from the switch is programmable as described in Section 7.1.6.8 “Egress Switch Overhead Information Register 0 (ESOIR0).” The egress switch number of bytes (ESNB) field defines the number of bytes received from the switch. This field must be an even value for 16-bit mode operation (ESWD = 1). Note that the HEC octet or UDF word that is provided when the egress switch HEC field (ESHF) bit is set is considered overhead information and is not passed through the MC92520 as part of the cell.

The bytes are provided by the switch in the following order:

1. Overhead bytes (number determined by ESNB)
2. ATM cell header (4 bytes; PTI, CLP valid; VCI valid if VP switching)

3. HEC octet (8-bit mode) or UDF word (16-bit mode) (provided only if ESHF is set)—This octet/word may be used for overhead information because no HEC/UDF value is stored in the internal data structure.
4. ATM cell payload (48 bytes)

The fields contained in the overhead bytes are:

- Egress connection identifier (ECI) or multicast identifier (MI) (see Section 5.2.2 “Multicast Identifier Translation”)
- Multicast bit (M) (see Section 5.2.2 “Multicast Identifier Translation”)
- Explicit forward congestion indication (EFCI) (see Section 5.2.7 “Translating Addresses”)
- Multicast translation table section (MTTS) (see Section 5.2.2 “Multicast Identifier Translation”)

The location of these fields in the overhead, header, and HEC bytes is programmed using fields described in Section 7.1.6.6 “Egress Switch Interface Configuration Register (ESWCR)” and Section 7.1.6.9 “Egress Switch Overhead Information Register 1 (ESOIR1).” This mechanism is illustrated in the following figures.

Figure 5-19 shows the extraction of the ECI field from the switch cell data structure. Each of the two bytes of the ECI may be taken from any of the non-payload bytes of the structure using the identifier most-significant byte (IMSB) and identifier least-significant byte (ISLB) fields of the egress switch interface configuration register (ESWCR).

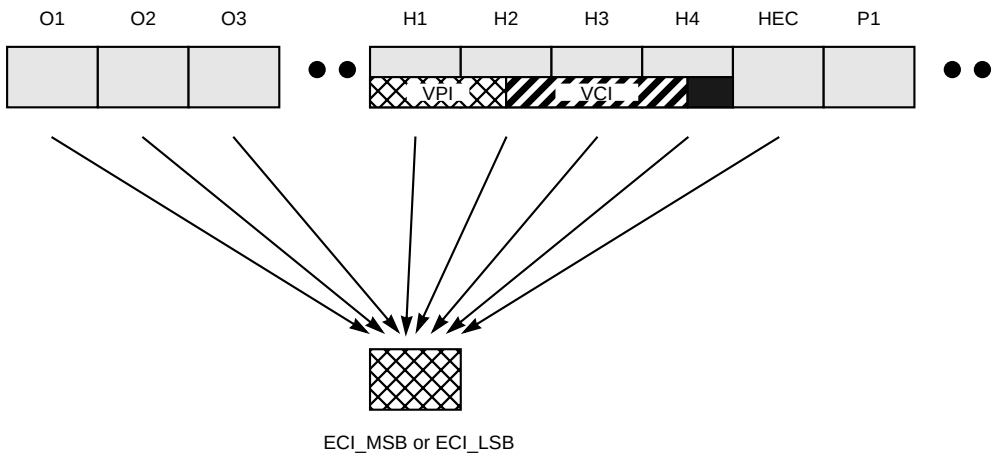


Figure 5-19. ECI Extraction from Switch Cell Data Structure

Figure 5-20 shows the extraction of the EFCI or M bits from the switch cell data structure. Each of these bits may be taken from any bit of any non-payload byte of the structure. The

location is specified in two stages. The byte is specified using the EFCI byte location (EFBY) or M byte location (MBY) fields, and the bit location within the byte is specified using the EFCI bit location (EFBI) or M bit location (MBI) fields of the egress switch overhead information register (ESOIR0).

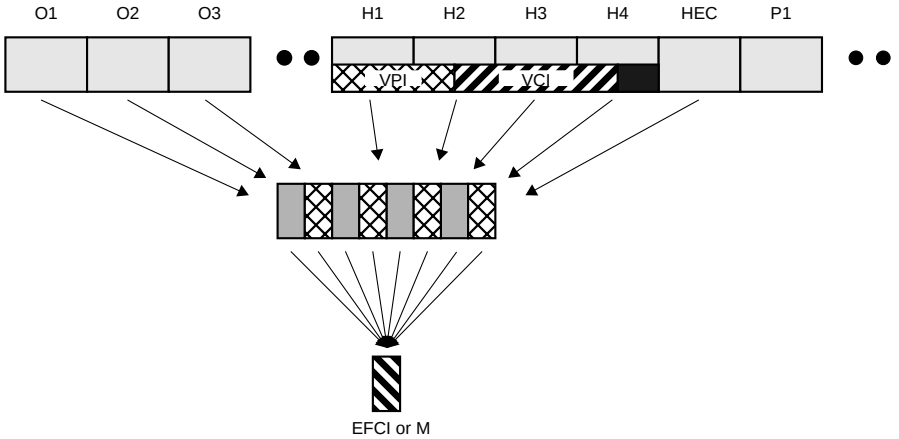


Figure 5-20. EFCI and M Extraction from Switch Cell Data Structure

Figure 5-21 shows the extraction of the MTTS field from the switch cell data structure. This field may be taken from any of the five possible bit alignments within any non-payload byte of the structure. The location is specified in two stages. The byte is specified using the MTTS byte location (MTBY) field, and the bit alignment within the byte is specified using the MTTS bit location (MTBI) field of the egress switch overhead information register (ESOIR).

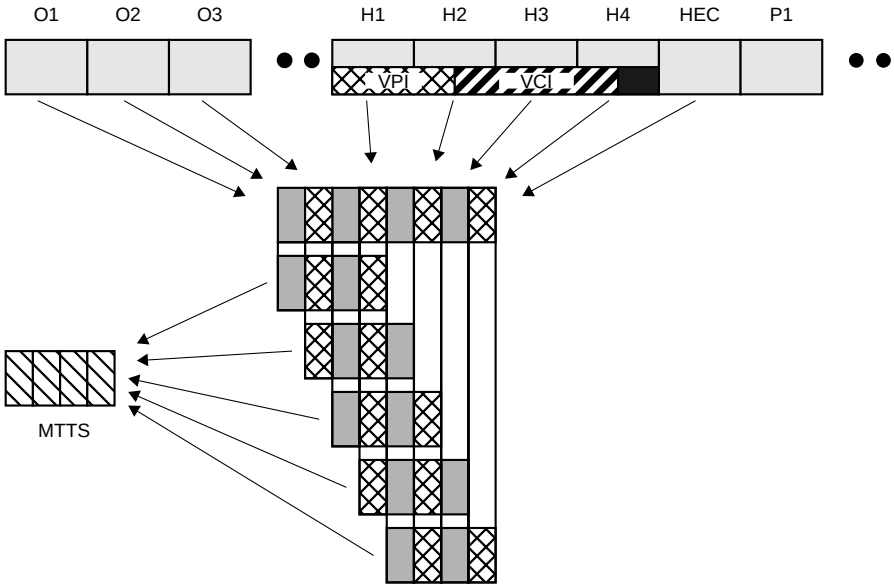


Figure 5-21. MTTS Extraction from Switch Cell Data Structure

Figure 5-22 shows an example configuration and the resulting overhead extraction. The ECI can be extracted from the header by setting the identifier in header address fields (IHAF) bit. In this case, the header VPI field size can be programmed to either 12 bits or 8 bits using the VPI size in ECI on header (VPS) mode bit of the egress switch interface configuration register (ESWCR). Once the valid fields have been retrieved, the remaining overhead bytes received from the switch are discarded because they are of no use to the MC92520 and are not transferred to the PHY layer.

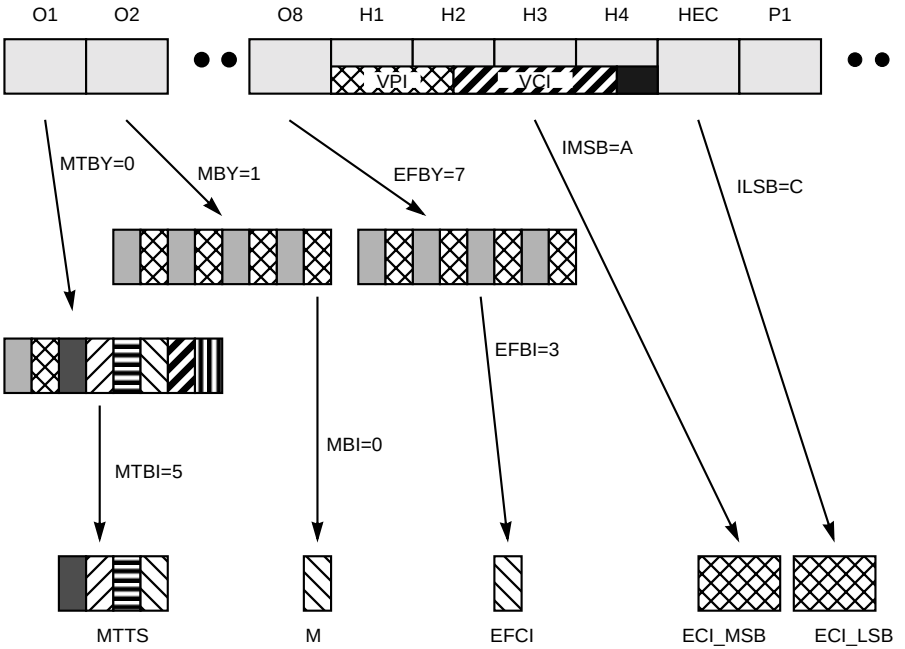


Figure 5-22. Overhead Extraction Example

The egress switch interface block provides overhead routing information to the cell processing block, as follows:

- The size of the ECI/MI field used by the egress cell processing block can be programmed by writing to the egress cell processing block ECI size (ECES) field of the egress overhead manipulation register EGOMR.
- The size of the MTTS field used by the egress cell processing block can be programmed by writing to the egress cell processing block MTTS size (EGOMR[ECTS]) field.
  The M bit used by the egress cell processing block can be either the M bit that was extracted from the cell's overhead, or the logical not of the M bit that was extracted from the cell's overhead, or 1 or 0 by programming the egress cell processing block M bit source (EGOMR[ECMS]) field.

NOTE:

The M, MTTS, and MI fields are used for multicast identifier translation, described in the next section. If multicast identifier translation is not performed, the ECI field contains the egress connection identifier of the connection to which the cell belongs. An ECI value of all 1's is invalid and causes the cell to be removed from the cell flow and copied to the cell extraction queue.

5.2.2 Multicast Identifier Translation

Multicasting involves copying a cell that arrived at a switch and transmitting it on multiple physical links. In the general case, the ECI of a connection to which a cell belongs is different on each link. If the switch can provide the correct ECI to each ATMC device, the multicast operation is transparent to the MC92520. However, if the switch cannot provide separate ECIs for each link, a common multicast identifier may be provided to all of the ATMC devices. Each MC92520 translates the multicast identifier into the ECI for its physical link.

Multicast identifier translation is enabled globally by the multicast translation table control (MLTC) field of the egress multicast configuration register (EMCR). By setting the multicast (M) bit in the overhead information provided with each cell, the switch informs the MC92520 of the necessity of performing multicast identifier translation on the cell. If this bit is set, the overhead information contains a multicast translation table section (MTTS) and a multicast identifier (MI). The MTTS field is effectively concatenated to the left of the less significant portion of the MI to obtain the index to the multicast translation table, as shown in Figure 5-23.

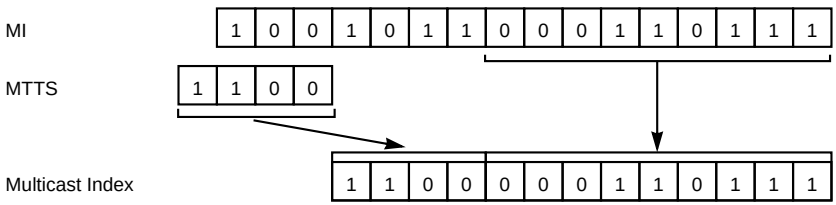


Figure 5-23. Multicast Derivation Example

The ECI is found by reading from the multicast translation table using this index as shown in Figure 5-24. This real ECI is used for all further processing.

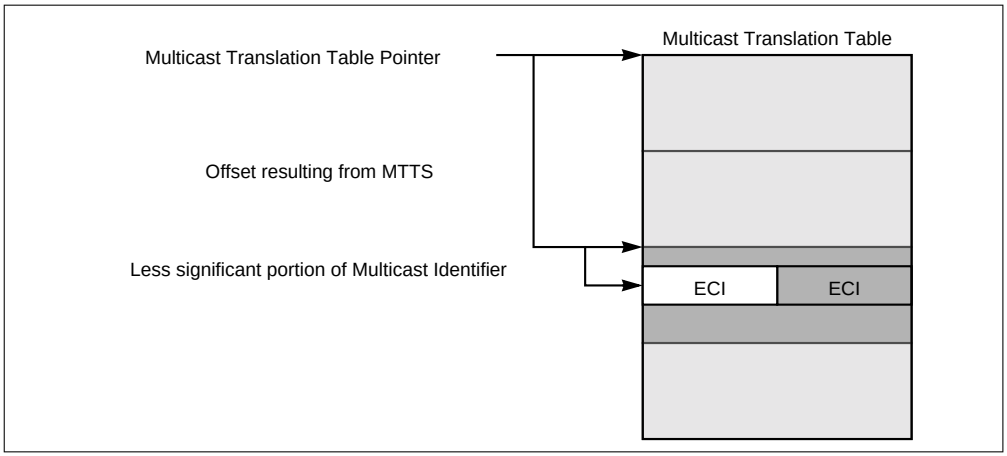


Figure 5-24. Multicast Translation

The number of bits to be used from the MI is programmable (see Section 7.1.6.13 “Egress Multicast Configuration Register (EMCR)”) as shown in Table 5-9. For example, if 10 bits of the MI are used when the MI is 0x1234 and the MTTS is 0xF, the resulting index is 0x3E34.

Table 5-9. Number of MI Bits Used for Sectioned Multicast Translation Table

EMIC	Multicast Identifier Bits Allocated for Multicast Translation
000	9
001	10
010	11
011	12
100	13
101	14
110	15
111	16

A possible application of the MTTS field is an MC92520 supporting multiple physical links. If the link number is provided in the MTTS field, a separate lookup is performed for each physical link. If MLTC is reset, the M bit is ignored, and the ECI is taken from the switch overhead information. If the ECI is all 1s, the cell is removed from the cell flow and copied to the cell extraction queue.

5.2.3 Inserting Cells into the Egress Flow

The cell insertion rate is paced by a single leaky bucket to ensure that the cell queueing capability of a switch-side device is not exceeded. Beyond insertion rate pacing, the MC92520 can be configured to give either received cells a higher priority than inserted cells, or inserted cells a higher priority than received cells. For more information see the egress insertion priority (EIP) bit description in Section 7.1.6.12, “Egress Processing Configuration Register (EPCR).”

The parameters of the leaky bucket are determined by the egress insertion leaky bucket register (EILB). The bucket contents’ value, contained in the egress insertion bucket fill register (EIBF), is incremented by the egress average insertion period (EAIP) when a cell is inserted in the egress cell flow and is decremented by one in each cell slot. The MC92520 inserts a cell only if the bucket contents’ value in the EIBF is smaller than the egress insertion bucket limit (EIBL).

The general algorithm internal to the MC92520 is thus:

```
EIBF = 0 at reset
For each cell slot
        EIBF = max(EIBF - 1, 0)
        If (EIBF < EIBL && cell insertion queue not empty)
                Insert cell
                EIBF = EIBF + EAIP
        EndIf
EndFor
```

Note that if the insertion cell’s destination link is disabled at the time of insertion, the cell may be discarded.

NOTE:

The EIBL field is zero after reset and must be written with a non-zero value to enable cell insertion.

The EAIP consists of a 12-bit integer part and a 4-bit fractional part that provide for values as large as 4095 cell processing times (equivalent to inserted cells being 0.024% of the cell flow) with a precision of 1/16 of a cell processing time. At the typical value of 100 cell processing times (1% of the cell flow), the precision is 0.0006 percentage point. Note that programming the EAIP with a value of zero provides unlimited cell insertion. The 16-bit EIBL provides for a back-to-back burst of 16 cells when using the maximum value of EAIP and proportionately larger bursts when using smaller values of EAIP. The bursts occur while the EIBF value is being incremented to the threshold set by the EIBL.

NOTE:

Insertion cell bursts should be limited to 1 (a single cell insertion) when the MC92520 is in an operating mode using per-PHY FIFOs. Otherwise one or more back-to-back inserted cells may be discarded.

For example, if the inserted cells are 1.3% of the cell flow the EAIP is 1/.013 = 76.92 that rounds to 76 15/16, or 0x04C.F. If the switch can handle a burst of up to 10 cells, the EIBL is defined to be one EAIP value less than the required bucket size. Therefore, in our example the EIBL should be 10 – 1 = 9 times the EAIP value: 76.94 × 9 = 692, or 0x02B4.

For the typical insertion rate of 1% of the cell flow, the EAIP is 1/.01 = 100, or 0x064.0. If no back-to-back cell burst is desired, the EIBL may be defined as a value of less than 100, so that the value of EIBF does not satisfy the condition (EIBF < EIBL), discussed above, immediately after the first inserted cell (because EIBF has been incremented by EAIP).The types of cells that can be inserted in the egress cell flow are:

- OAM cells generated internally by the MC92520 including:
  - AIS cells
  - RDI cells
  - Continuity check cells
  - PM forward monitoring cells
- OAM cells generated by the microprocessor
- Other cells generated by the microprocessor

The various types of cells that can be inserted in the egress cell flow are classified by their insertion priority and held in separate queues. The insertion priorities are (from highest to lowest):

1. PM forward monitoring cells generated internally by the MC92520
2. Cells from the microprocessor
3. AIS, RDI, and CC cells generated internally by the MC92520

The data structure of inserted cells from the microprocessor is provided in Section 7.3.1 “Inserted Cell Structure.”

5.2.4 Performing Context Table Lookups

Egress context table lookup is necessary to access connection-specific processing options and parameters defined in the context parameter table on a per-cell basis. Because the egress connection identifier (ECI) of the cell is known, either directly fetched from the cell or indirectly determined through the multicast translation table, all necessary information can be read from the context parameters table (whose structure is presented in Section 7.2.3 “Context Parameters Table”).

For all cells that do not originate from the insertion queue, the MC92520 checks that the physical link associated with the cell is enabled. If a switch-side multi-PHY interface is used, the switch-side port id is taken as the PHY-side link id. Otherwise, the link id associated with the cell is fetched from the egress address translation word, or, if the address translation word is not present, a link id determined from the egress link number selection (ESWCR[ELNS]) bit is used. If the target link id of a cell is not enabled in the

egress link enable register (ELER) or in the associated egress link register (ELNK*n*), the cell is removed from the cell flow and copied to the cell extraction queue.

5.2.5 Performing UPC/NPC Processing

The MC92520 performs the UPC/NPC function for the egress flow if the UPC flow (UPCF) bit is set in the ATMC CFB configuration register. See Section 6.1 “Traditional UPC/NPC Support.”

5.2.6 Performing OAM Processing

If the egress copy all (ECA) cells bit is set, the cell is added to the cell extraction queue to be transferred to the microprocessor. If the egress remove all (ERA) cells bit is set, the cell is removed from the cell flow after undergoing OAM processing. The option exists to copy or remove those cells whose VCI is identified as “reserved.” See Section 7.1.6.30 “Egress VCI Copy Register (EVCR)” and Section 7.1.6.32 “Egress VCI Remove Register (EVRR)” for details. This option is enabled on a connection basis by the egress VCR/VRR registers enable (EVRE) bit of the egress parameters word. There is also an option to copy those cells whose PTI value is 110 or 111 to the cell extraction queue. This option is controlled by the egress PTI 6 copy (EP6C) and egress PTI 7 copy (EP7C) bits of the egress parameters word.

The CRC-10 field of received OAM cells is checked. If an error is detected in this field, or if the cell is an illegal OAM cell (see Section 6.5.3.1 “Illegal OAM Cells”), no OAM processing is performed on the cell. The cell is removed from the cell flow and copied to the cell extraction queue. All OAM cells are further classified by the OAM cell type and OAM function type fields (see Table 6-6 and Figure 6-22). The OAM cells that receive special processing are:

- Cells received from the physical layer such as:
  - AIS (see Section 6.5.4.1.1 “Alarm Indication Signal Cells”)
  - RDI (see Section 6.5.4.1.2 “Remote Defect Indicator Cells”)
  - Loop-back (see Section 6.5.4.2.1 “Loopback Cell Format”)
  - Continuity check (see Section 6.5.4.1.3 “Continuity Check Cells”)
  - Forward monitoring (see Section 6.5.6 “Performance Monitoring”)
  - Backward reporting (see Section 6.5.6 “Performance Monitoring”)
  - If this is a segment/connection termination point (see **Section 6.5.3 “Generic OAM Support**”) of the OAM flow, the segment/end-to-end OAM cell is removed from the cell flow.
- Inserted cells such as:
  - Forward monitoring from the processor (see Section 6.5.6 “Performance Monitoring”**)**
  - Forward monitoring internally generated (see Section 6.5.6 “Performance Monitoring”**)**

When a user data cell is processed, the traffic bits (see Section 6.5.4.1.3 “Continuity Check Cells”) are set. User data cells that belong to an active PM block test are processed as described in Section 6.5.6 “Performance Monitoring.”

5.2.7 Translating Addresses

The address fields of the cell header are optionally replaced by the outgoing address of the outgoing link as read from the egress translation address word of the context parameter table. The address translation is controlled by the egress address translation disable (EATD) bit of the egress switch interface configuration register (ESWCR). If the outgoing address is provided by the switch interface block, no address translation is performed. If the outgoing address is not provided with the cell, the address is translated using the value read from the context parameters table. If the cell belongs to a VPC, only the VPI field is replaced. If the cell belongs to a VCC, both the VPI and VCI fields are replaced. If the cell belongs to a UNI link, the replacement of the GFC field is controlled by the replace GFC field (RGFC) bit of the egress processing configuration register (EPCR). The MC92520 sets the middle bit of the PTI on cells whose received PTI is 000 or 001 when the EFCI bit received from the egress switch interface block is set.

5.2.8 Incrementing Connection and Cell Counters

For each cell transmitted to the PHY layer, one of the counters from the egress billing counters table for this connection is incremented, unless the table does not exist - see Section 7.1.6.12 “Egress Processing Configuration Register (EPCR).” One of the link cell counters from the egress link counters table is also incremented if the table exists - see Section 7.1.6.15 “General Configuration Register (GCR).” The appropriate counter is chosen based on the CLP bit and whether the cell is an OAM cell. Inserted cells and internally generated cells are included in the usage counts, but cells that are removed from the cell flow are not included.

5.2.9 Transmitting Cells to the Physical Layer

The egress physical-layer interface (EPHI) block receives cells from the cell processing block, queues them in a FIFO, and transmits the cell data to the physical layer through a standard UTOPIA interface.

The MC92520 can interface to one or more physical layer devices through its single-PHY or multi-PHY interface operation. For multi-PHY operation, the MC92520 can be further configured to use a single FIFO or multiple, per-PHY FIFOs. If a single FIFO is selected, the size of the FIFO is programmable to either 2 or 4 cells using the egress PHY interface FIFO control (EPFC) bit. If multiple FIFOs are enabled, the size of each FIFO is programmable to a depth of 1 to 4 cells via the egress link registers (ELNK*n*). Furthermore, in multi-FIFO operation, the egress PHY port UTOPIA cell transfer priority can be configured for each port to a value of 0 through 15 (0 being highest) via the egress link

registers (ELNK*n*). For more information, see Section 3.4.2.2, “Switch-Side Multi-PHY Interface Support,” Section 7.1.6.4 “Egress PHY Configuration Register (EPHCR),” and Section 7.1.5.15, “Egress Link Registers (ELNK0–ELNK15),” respectively.

If a single PHY interface is selected, the MC92520 can be programmed to generate cells to provide a continuous cell flow while the EPHI FIFO is empty (see Section 7.1.6.4 “Egress PHY Configuration Register (EPHCR).” The type of cell used to fill is either “unassigned” (an ATM layer cell) or “idle” (a physical layer cell) according to the egress generate idle cells (EGIC) bit defined in Section 7.1.6.4 “Egress PHY Configuration Register (EPHCR).” See Figure 7-120 and Figure 7-130 for the header values used for unassigned and invalid (idle) cells. If a multi-PHY interface is selected, the generation of unassigned or idle cells is not supported and must not be enabled.

Because the MC92520 processes cells at a higher rate than they are transmitted to the physical layer, the EPHI block cannot transfer a cell during every cell processing slot. Over time, cells may accumulate in the EPHI FIFO until it is full. When this happens, the MC92520 does not process a cell during the next cell processing slot: rather it allows the FIFO to drain to the physical layer. TXPRTY is always driven with odd parity over TXDATA, regardless of whether or not parity checking is enabled on the ingress PHY interface. Dependent on the setting of the egress PHY wide data path (EPWD) bit, parity is generated for TXDATA bits [15:0] or [7:0]. The HEC octet or UDF word of the transmitted cell is always transmitted as zero, regardless of the value passed to the MC92520 by the switch interface block.

Chapter 6
Protocol Support

The MC92520 ATM cell processor incorporates support for a number of industry protocols to provide maximum functional flexibility. The protocol support is described in detail in the following paragraphs and includes:

- Traditional usage parameter control (UPC) support, including:
  - Cell-based UPC
  - Partial packet discard (PPD)
  - Early packet discard (EPD)
  - Limited early packet discard (limited EPD)
  - Selective discard support
- CLP bit management and transparency support
- Guaranteed frame rate (GFR) support, including:
  - Conformance checking
  - Eligibility checking
  - Fair-share administration
- Available bit rate (ABR) support, including:
  - FRM and BRM relative rate marking
  - EFCI bit marking on non-RM cells
  - RM cell priority, extraction, or removal
- Operations and maintenance (OAM) support, including:
  - General OAM including activation/deactivation
  - Fault management (AIS, RDI, CC, Loopback)
  - Performance management (FM, BR)

6.1 Traditional UPC/NPC Support

One of the major advantages of ATM is the ability to distribute the available bandwidth among many connections dynamically. However, this same feature makes congestion in an ATM network difficult to predict. In order to facilitate network management, limits are imposed on connection traffic parameters. Typically, the maximum average bandwidth and

burstiness are defined. Even when the usage parameters are defined, a single user not conforming to the agreed-upon parameters can cause congestion that reduces the quality of service for other users. Therefore, usage parameters should be enforced at the entrance to the network so that only violating users suffer any reduced service quality. This enforcement is called usage parameter control (UPC) when it is performed at a user-network interface (UNI) and network parameter control (NPC) when it is performed at a network-network interface (NNI). The differentiation between UPC and NPC is irrelevant for the following MC92520 descriptions. Rather, generic references to UPC apply to both UNI and NNI applications.

This section presents traditional cell-based and simple packet-based UPC functions that assume neither full duplex cell flows nor a combination of ingress- and egress-side UPC processing. UPC support for service categories that assume such requirements is described in Section 6.3, “Guaranteed Frame Rate (GFR) Support” and Section 6.4, “Available Bit-Rate Support.”

Because the traditional UPC functions depend only on a simplex cell flow, the MC92520 can support the option to perform UPC on either the ingress or the egress cell flow. For more information, see the description of the UPC flow bit (ARR[UPCF]) in Section 7.2.4.8, “ATMC CFB Revision Register (ARR).”

NOTE:

UPC support for ABR is not possible and UPC support for GFR is restricted if the MC92520 is configured to perform UPC on the egress cell flow.

6.1.1 Cell-Based UPC

The default UPC mode for the ATM cell processor is cell-based UPC. The MC92520’s UPC algorithm, based on the concept of leaky buckets, detects cells that violate the traffic agreement and either tags violating cells (by changing the cell loss priority (CLP) field from 0 to 1) or discards them (by removing them from the cell flow). A flexible arrangement of 0 to 4 leaky buckets, leaky bucket parameters, and UPC enforcement algorithms are used to define and maintain the cell traffic contract. At connection setup time, a set of bucket characteristics is loaded into the bucket table section of context memory to define the expected cell arrival pattern for a particular connection. During cell processing, the UPC function uses these characteristics to enforce the agreed-upon user traffic requirements.

For constant bit rate and variable bit rate connections, constant bucket characteristics are normally defined when the connection is set up. Other types of connections may require dynamic UPC/NPC enforcement in which the processor updates bucket characteristics while the connection is active. Updating bucket characteristics must be done cautiously in order to maintain consistency among various enforcement parameters. Also, sharing a bucket (by placing the identical bucket pointer in the common parameters word of each connection) allows the UPC mechanism to enforce the sum of several connections. This

method is likely to be used at a boundary point where many VCCs are combined into a VPC.

All user data cells successfully associated with a connection are subject to UPC policing according to the connection parameters. UPC processing can include the following functions:

- Maintaining counts of discarded or tagged cells per connection in the policing counters table.
- Applying the UPC algorithm for statistical purposes without tagging or discarding violating cells. An ingress-only “don’t touch” option is provided for this.

6.1.2 Packet-Based UPC

The ATM Forum Traffic Management Specification states that if a network elementneeds to discard cells, then it is typically more effective to discard them at the packet level rather than the cell level. Adhering to this recommendation, the MC92520’s UPC function can perform packet-based discard on AAL5 packets (excluding inserted OAM cells). AAL5 defines a packet as a stream of one or more user cells belonging to the same VC connection on which the payload type identifier [0] bit equals 1 on the last cell and PTI[0] bit equals 0 on all the other cells.

The MC92520 offers three UPC modes of simple, packet-based policing functions on a per-connection basis:

- Partial packet discard (PPD)
- Early packet discard (EPD)
- Limited early packet discard (limited EPD)

These UPC modes are selected using the UPC operation mode (UOM) bit in the context parameters extension table.

6.1.2.1 Partial Packet Discard (PPD)

With this UPC mode, once a cell is discarded or tagged, all succeeding cells (except the last cell) belonging to the same packet are discarded or tagged. If the PPD admit last cell bit (PALC) in the general configuration register (GCR) is set, the last cell of a packet is admitted if any previous cell of the packet has been admitted.

If GCR[PALC] is cleared, the last cell of a packet is processed by the UPC algorithm like any other cell of the packet; that is, the last cell may be discarded, tagged, or passed unchanged.

The PPD UPC functions include the following:

- Discarding or not discarding: the UPC can be in either discarding or not discarding state
- Updating of tagging and policing counters when it is in the not discarding state

- Transitioning from the not discarding to the discarding state on the first discarded cell
- Incrementing the policing discard counter (and not updating the UPC bucket) while the UPC is in the discarding state
- Discarding the last cell received from a packet if all previous cells in that packet were discarded, so long as the MC92520 is in the discarding mode
- Admitting the last cell of a packet if the packet was truncated (not all the cells were discarded). This delineates the corrupted packet from the next packet. If this last cell violates cell-based UPC, however, the configuration of GCR[PALC] determines whether the last cell is discarded or admitted.

Figure 6-1 shows PPD algorithm usage assuming GCR[PALC] is cleared. The first packet is truncated. The last cell of the first packet is transmitted and thus avoids the concatenation of the corrupted packet to packet 2. Packet 3 is truncated as well. Its last cell is not transmitted because it cannot be admitted by the cell-basedUPC while GCR[PALC] is cleared. Packet 4 is not transmitted at all.

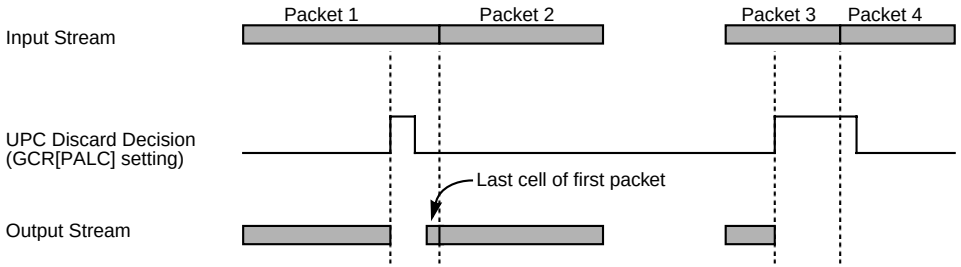


Figure 6-1. PPD Algorithm Example

6.1.2.2 Early Packet Discard (EPD)

In EPD mode, the decision to discard a packet occurs only at the beginning of a packet. This means that a packet is either fully discarded or fully passed. EPD functions include the following:

- Incrementing the policing discard counter (and not updating buckets) when the EPD is discarding cells
- When passing a packet:
  - All tagging buckets continue to work in a cell-based fashion.
  - All discarding buckets perform their calculations as if the limit parameter is infinite and therefore increment the bucket content and do not discard any cells. As a result, their bucket content can be greater than their bucket limit.
  - The MC92520 may increment its police tagging counter.

Figure 6-2 is an example of the EPD algorithm. The first packet cells violate UPC, but owing to EPD, this packet is fully passed. The second packet is fully discarded. The third packet cells violate UPC, but this packet is passed, not discarded. Because the fourth packet arrives after a relatively long time, the UPC buckets are drained and the UPC is no longer violated. Therefore, the fourth packet is passed.

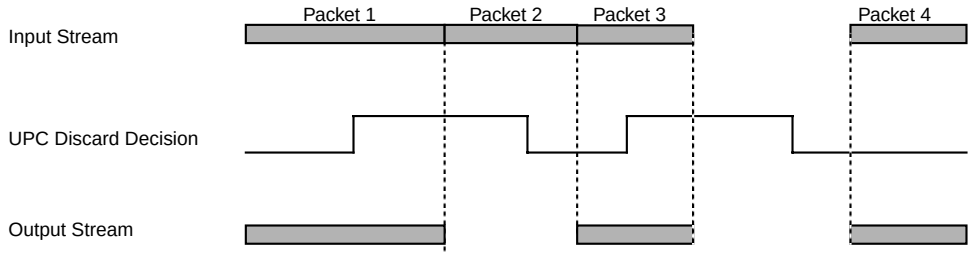


Figure 6-2. EPD Algorithm Example

6.1.2.3 Limited Early Packet Discard (Limited EPD)

One disadvantage of the EPD algorithm is that once a packet is passed, the decision of retaining or discarding the packet cannot be changed until the last cell of that packet has arrived. In the case of large packets, this waiting for the last cell can cause switch congestion. Using the limited EPD algorithm, a connection can stop passing cells once it reaches a predefined limit. In the MC92520, that limit is reached once the first bucket starts discarding cells. The first bucket should have the same parameters as one of the other buckets except for the bigger limit. Figure 6-3 shows how the three buckets function together.

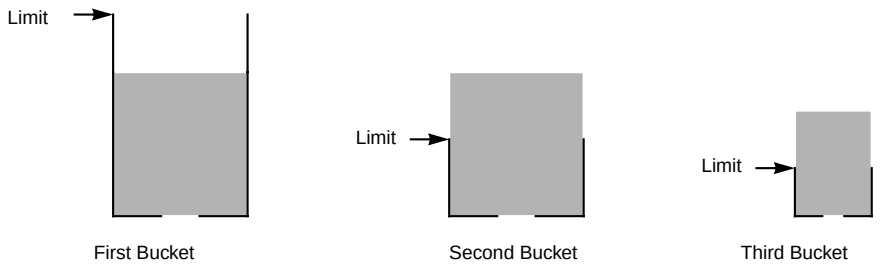


Figure 6-3. Three Bucket Example

The first bucket limits the EPD algorithm.The first and second buckets share the same parameters except for the limit; therefore, their bucket content is always the same. Although the second bucket content is higher than its limit, cells are admitted by the EPD algorithm. When the first bucket reaches its limit, then cells are discarded.

Figure 6-4 illustrates the difference between EPD and limited EPD function. It shows that

EPD cannot start discarding cells until the first cell of the next packet arrives, whereas limited early packet discard can begin discarding as soon as congestion is detected.

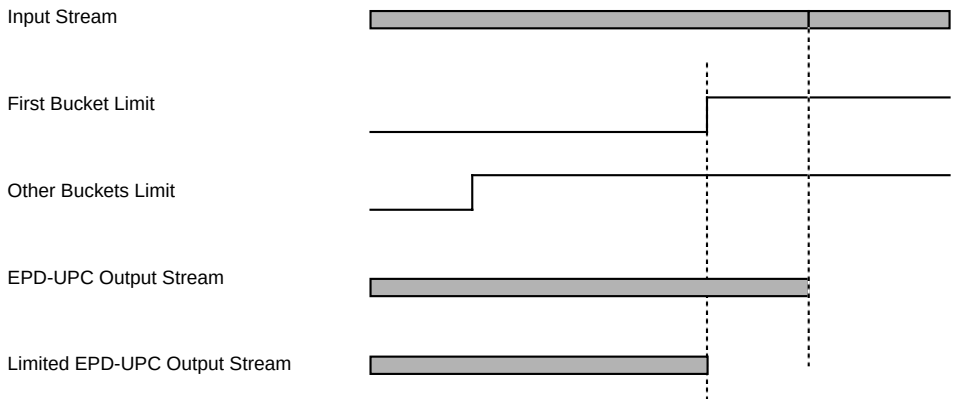


Figure 6-4. Comparison of EPD vs. Limited EPD

Similar to PPD described earlier, the PPD admit last cell (PALC) bit of the general configuration register (GCR) can be configured to circumvent a discard decision of the cell-based UPC on the last cell of the packet.

6.1.3 Selective Discard Support

The ATM Forum Traffic Management Specification defines procedures according to which cells can be discarded by network elements. A switching element may discard cells belonging to selected connections or a specific cell flow (for example, cells whose CLP = 1) in case of congestion. This function is called selective discard and it is implemented by the MC92520. Selective discard is globally activated and deactivated by setting or clearing the global ingress congestion notification (ICNG) bit in the ingress processing control register (IPLR). Selective discard is enabled on a per-connection basis through the ingress selective discard mode (ISDM) field in the common parameter extension word. The content of the ISDM field also determines whether selective discard is performed on CLP=1 or on CLP=0+1 cell flow.

6.2 CLP Bit Management and Transparency Support

The MC92520 provides several options to convey an ATM cell's cell loss priority (CLP) bit and optional UPC tagging information to a switch fabric. These options are intended to support these three objectives:

- CLP transparency in conjunction with simple switch fabric designs,
- Preservation of CLP for GFR.1 while using local tagging for UPC, and
- Transference of local cell tagging information to UPC performing switch fabrics.

These objectives are achieved through the MC92520's ability to preserve and replace the CLP bit in the cell header before the cell is passed to the switch fabric. Similarly, on the egress side of the switch fabric, the MC92520 can be configured to restore a CLP bit before the cell is passed to the PHY device.

NOTE:

This type of CLP bit management is only relevant for applications that police on the ingress-side cell flow, that is, the UPCF bit in the ATMC CFB configuration register (ACR) is cleared.

The basic mechanism is outlined in Figure 6-5 below:

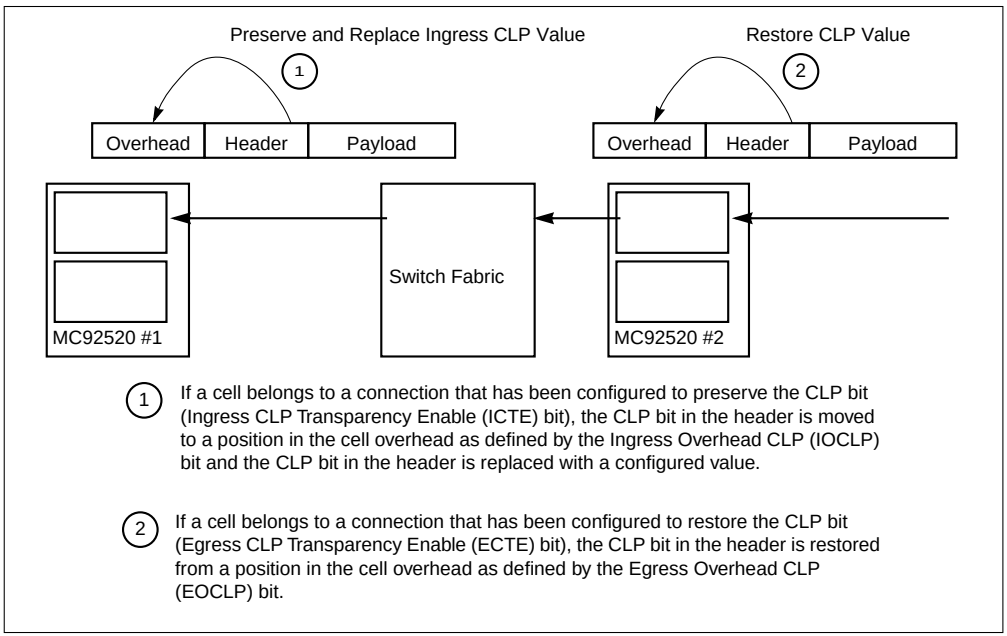


Figure 6-5. CLP Preservation and Restoration

CLP bit management must be globally enabled by setting the ingress global CLP transparency enable (IGCTE) bit in the ingress processing configuration register (IPCR) and the egress global CLP transparency enable (EGCTE) bit in the egress processing configuration register (EPCR), respectively. The byte and bit positions used to access the CLP bit in the cell overhead are defined in the CLP transparency overlay register (CTOR) on the ingress side, and in the egress switch overhead information register 1 (ESOIR1) on the egress side.

Per-VC CLP bit preservation and restoration is enabled in the context parameters extension table with the ingress CLP transparency enable (ICTE) and egress transparency enable

(ECTE) bits. If the CLP bit management function is disabled (if marking is not being allowed), the post-enforcement version of the CLP bit is passed to the switch fabric in the ATM cell header.

If the CLP bit management is enabled, several options exist to determine whether the pre-enforcement or the post-enforcement version of the CLP bit is preserved in the cell overhead, and whether the post-enforcement version or a fixed CLP bit value (0 or 1) is used to replace the CLP bit in the cell header that is passed to the switch fabric. The available options are shown in Figure 6-6 below:

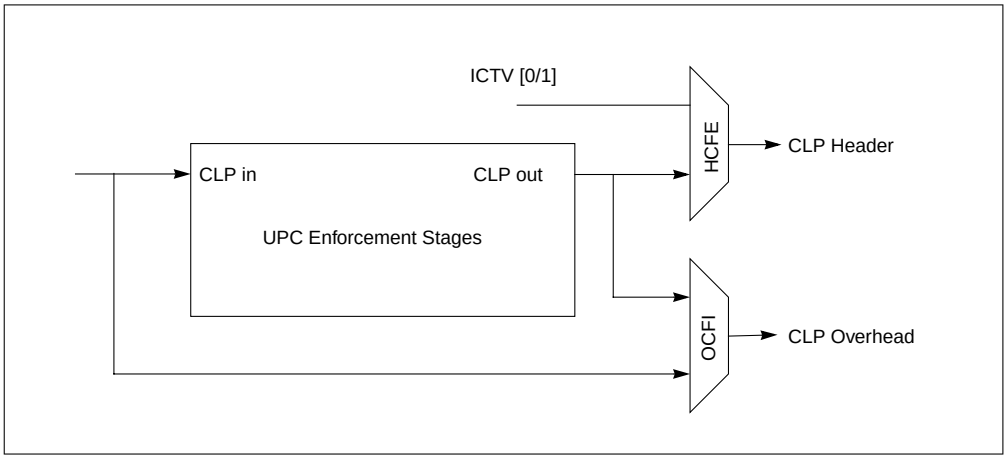


Figure 6-6. CLP Selection for Preservation and Replacement

All CLP bit management options are configured in the context parameters extension table (meaning that they may be defined on a per-VC basis). The overhead CLP from input (OCFI) bit determines whether the pre-enforcement or post-enforcement version of the CLP bit is preserved in the cell overhead information. The header CLP from enforcement (HCFE) bit determines whether the post-enforcement version of the CLP bit or a fixed value (defined by the ingress CLP transparency value (ICTV) bit) is stored in the cell header before the cell is passed to the switch fabric.

6.2.1 CLP Bit Management for Simple Switch Fabrics

The traffic management specification defines two network operation models with relation to CLP = 1 flow: CLP-transparent and CLP-significant. A connection which is CLP-transparent doesn't have different cell loss ratio (CLR) for CLP = 0 or CLP = 1 traffic, and therefore, doesn't prefer discarding CLP = 1 over CLP = 0 cells on congestion. Simple switch fabrics do distinguish globally between CLP = 0 and CLP = 1 traffic and the latter is generally selected for discarding in the presence of congestion.

The following MC92520 CLP management options can be used to address this problem, or to protect frame-based VCs from cell-based discard, without requiring changes to the switch fabric:

1. Enable global ingress and egress CLP bit management by setting the IGCTE and EGCTE bits in the IPCR and EPCR, respectively.
2. Define suitable byte and bit positions for a preserved CLP bit in the cell overhead by setting the OCBI and OCBL fields in the CTOR register for the ingress side, and the EOBY and EOBI fields in the ESOIR1 register for the egress side.
3. Perform the following configurations for all VCs needing CLP transparency around the switch fabric:
   - — Enable CLP bit management by setting the ICTE and ECTE bits in the individual context parameter extension table entries.
   - — Select the source of the CLP bit in the overhead with OCFI in the individual context parameter extension table entries. Generally the intent is to restore this overhead CLP bit upon egress.
   - — Select ICTV as the source for the CLP in the header and select a 0 value by clearing HCFE and ICTV in the individual context parameter extension table entries.
4. Perform the following configuration for all VCs that do not need CLP transparency around the switch fabric:
   - — Disable CLP bit management by clearing the ICTE and ECTE bits in the individual context parameter extension table entries (implies post-enforcement CLP is in header).

The combination of these configuration steps results in the following behavior:

Cells of VCs with a CLP-transparent flow are forwarded to the switch fabric with a CLP value of 0 in the cell header and a copy of the original CLP value in the cell overhead. After the cell is routed by the switch fabric (and processed in accordance with a CLP = 0 cell flow), the cell is passed to the egress MC92520 and the original CLP value is restored.

Cells of VCs without CLP transparency around the switch fabric are forwarded to the switch fabric with a CLP bit value derived from the UPC enforcement stages. No CLP is stored in the cell overhead. After the cell is routed by the switch fabric (and processed in accordance with CLP = 0 and CLP = 1 cell flows), the cell is passed to the egress MC92520 and the CLP value of the header remains unchanged.

6.2.2 CLP Bit Management for Frame-Based Connections

MC92520 support of frame-based connections is discussed in detail in Section 6.1.2, “Packet-Based UPC” and Section 6.3, “Guaranteed Frame Rate (GFR) Support.”

Connections using packet-based UPC and simple switch fabrics generally benefit from CLP transparency around the switch fabric as described in Section 6.2.1, “CLP Bit Management for Simple Switch Fabrics.”

For the purpose of illustrating the use of the MC92520 CLP bit management functions on GFR connections, it is sufficient to point out that there are two types of GFR traffic contracts:

- GFR.1: The CLP bit is transparently conveyed by the network and tagging is not allowed.
- GFR.2: The CLP bit is not transparently conveyed by the network and tagging is allowed.

Both types describe CLP-significant cell flows in which the ATM network uses the value of the CLP bit to provide the GFR service guarantee. In GFR.2, the ATM network may add tagging (by setting all CLP bits of selected frames), but in GFR.1, any user-provided CLP bit marking is transparently conveyed to the end system.

The GFR service guarantee provides low cell loss ratio to a minimum cell rate (MCR) flow of cells in complete, conforming, CLP = 0 frames. Users of GFR may also expect that a network will fairly allocate available resources to handle service requests that exceed the GFR service guarantee. This implies that switching systems need to differentiate, at least internally, between frames that must be transferred to satisfy the service guarantee and frames that should be transferred if resources are available, but may be discarded.

Frames exceeding the service guarantee can be tagged as they pass through the UPC enforcement stages and discarded by following enforcement stages or the switch fabric to allocate a fair share of available resources. Some of the frames may be tagged and not discarded. Since GFR.1 requires that no frame tagging leave the switching node, tagging can only be used on these connections as a method to differentiate GFR frames within a switching node. On GFR.2 connections, UPC tagging can be used to prevent excess CLP = 0 traffic from interfering with the service guarantees of other GFR connections at a downstream network node that is incapable of identifying excess CLP = 0 traffic for each GFR connection.

If a switch fabric cannot be set up to prevent GFR frame corruption (e.g., the fabric discards some CLP = 1 cells on congestion), any CLP = 1 information in the header must be suppressed before a cell is passed to the switch fabric as described in Section 6.2.1, “CLP Bit Management for Simple Switch Fabrics.” But if a switch fabric is frame-aware and provides valuable UPC functions to manage the utilization of available resources, it is essential to pass CLP information to the switch fabric.

The MC92520 CLP bit management options can be used to address all the requirements described above as follows:

1. Enable global ingress and egress CLP bit management by setting the IGCTE and EGCTE bits in the IPCR and EPCR, respectively.

2. Define suitable byte and bit positions for a preserved CLP bit in the cell overhead by setting the OCBI and OCBL fields in the CTOR register for the ingress side, and the EOBY and EOBI fields in the ESOIR1 register for the egress side.

Setup variation A for switch fabrics that are frame-aware (CLP=1 cell is conveyed to fabric):

1. Perform the following configurations for all GFR.1 VCs:
   - Enable CLP bit management by setting the ICTE and ECTE bits in the individual context parameter extension table entries.
   - Select pre-enforcement as the source of the CLP bit in the overhead and post-enforcement as the source of the CLP bit in the cell header by setting OCFI and HCFE in the individual context parameter extension table entries.
2. Perform the following configuration for all GFR.2 VCs:
   - Disable CLP bit management by clearing the ICTE and ECTE bits in the individual context parameter extension table entries (implies post-enforcement CLP is in header).

Setup variation B for switch fabrics that are *not* frame-aware (CLP=1 cell is hidden from fabric):

1. Perform the following configurations for all GFR.1 VCs:
   - Enable CLP bit management by setting the ICTE and ECTE bits in the individual context parameter extension table entries.
   - Select pre-enforcement as the source of the CLP bit in the overhead and ICTV as the source of the CLP bit in the cell header by setting OCFI and clearing the HCFE in the individual context parameter extension table entries.
   - Select a 0 value for the CLP in the header by clearing ICTV in the individual context parameter extension table entries.
2. Perform the following configuration for all GFR.2 VCs:
   - Perform all steps described in step 1 immediately above, except if tagging is desired select post-enforcement as the source of the CLP bit in the overhead by clearing OCFI in the individual context parameter extension table entries.

6.3 Guaranteed Frame Rate (GFR) Support

The MC92520 provides a complete guaranteed frame rate (GFR) solution in accordance with the ATM Forum Traffic Management Specification Version 4.1. Beyond determining traffic conformance and eligibility, the MC92520 provides a number of additional features to manage the fair allocation of resources available for VCs carrying GFR traffic.

In the following feature list, the policing actions PPD and EPD are described in Section 6.1.2.1, “Partial Packet Discard (PPD)” and Section 6.1.2.2, “Early Packet Discard (EPD).” The actions PPT and EPT stand for partial packet tagging (not recommended) and

early packet tagging. Note, the terms *packet* and *frame* are equivalent in this document (and in most ATM literature) and refer to the sequence of cells building an AAL5 protocol data unit (PDU).

The MC92520 GFR implementation has the following features:

- Optional per-frame CLP consistency policing (PPD)
- Optional maximum frame size (MFS) policing (PPD)
- Optional peak cell rate (PCR) policing (PPD, EPD, PPT, or EPT)
- Optional minimum cell rate (MCR) policing (PPD, EPD, PPT, or EPT)
- Optional fair cell rate A (FCRA) policing threshold (EPT or EPD). This is typically used to discard tagged frames.
- Optional fair cell rate B (FCRB) policing threshold (EPD). This is typically used to discard user-marked frames.
- Optional random frame drop (or tag) below FCRA
- Optional random frame drop below FCRB
- Support for up to 64 VC groups with the ability to adjust VC specific FCRs via a VC group-specific multiplication factor
- Support for network tagging of whole frames as well as CLP transparent operation with the option of intra-switch tagging

These features also enhance the set of behaviors available for managing other frame-based connections typically used with the unspecified bit rate (UBR) service category. Additional local behaviors are expected to play a major role in mapping the IETF specification of IP differentiated services (Diffserv) and IEEE 802.1D user priority levels to ATM. Although the IETF specification of Diffserv and its mapping to ATM have not been completed at the time of this writing, it is very likely that connections carrying Diffserv traffic will be established with a UBR or GFR service category and a parameter indicating the behavior class selector (BCS). Each switch is configured by network administration to associate a given BCS value with certain traffic management behaviors. Thus state information is not required for each TCP flow within the network core. The resulting service is a combination of traffic conditioning behaviors at the network edge and per-hop behavior in the network core.

6.3.1 GFR Policing and Fair Share Administration

The specific configuration of each UPC enforcement stage in supporting GFR cell and frame conformance, eligibility for a GFR service guarantee, administration of per-VC fair cell rates and congestion avoidance is described in the following sections:

- Section 6.3.1.1, “Global GFR Configurations”
- Section 6.3.1.2, “Cell Loss Priority (CLP) Consistency”
- Section 6.3.1.3, “Maximum Frame Size (MFS)”

- Section 6.3.1.4, “Peak Cell Rate (PCR)”
- Section 6.3.1.5, “Minimum Cell Rate (MCR)”
- Section 6.3.1.6, “Fair Cell Rate (FCR)”
- Section 6.3.1.7, “FCR Administration”
- Section 6.3.1.8, “Congestion Avoidance Through Random Frame Drop (RFD)”

Figure 6-7 lists all GFR enforcement stages with their typical scope configuration and all possible actions. Note that the MC92520 GFR implementation supports the differentiation of marked cells (CLP M = CLP 1 provided by service user) and tagged cells (CLP T = CLP 1 determined by the network, meaning determined by the MC92520) in fair cell rate (FCR) administration.

Cell discard or tagging decisions are latched and applied to the remainder of the frame. The partial packet discard function is extended to include the option of forcing the last cell of a frame to be unconditionally accepted if any other cell of the frame was accepted. As always, final cell disposition is fed back to all enforcement stages so that they can update their portion of the per-VC bucket information accordingly.

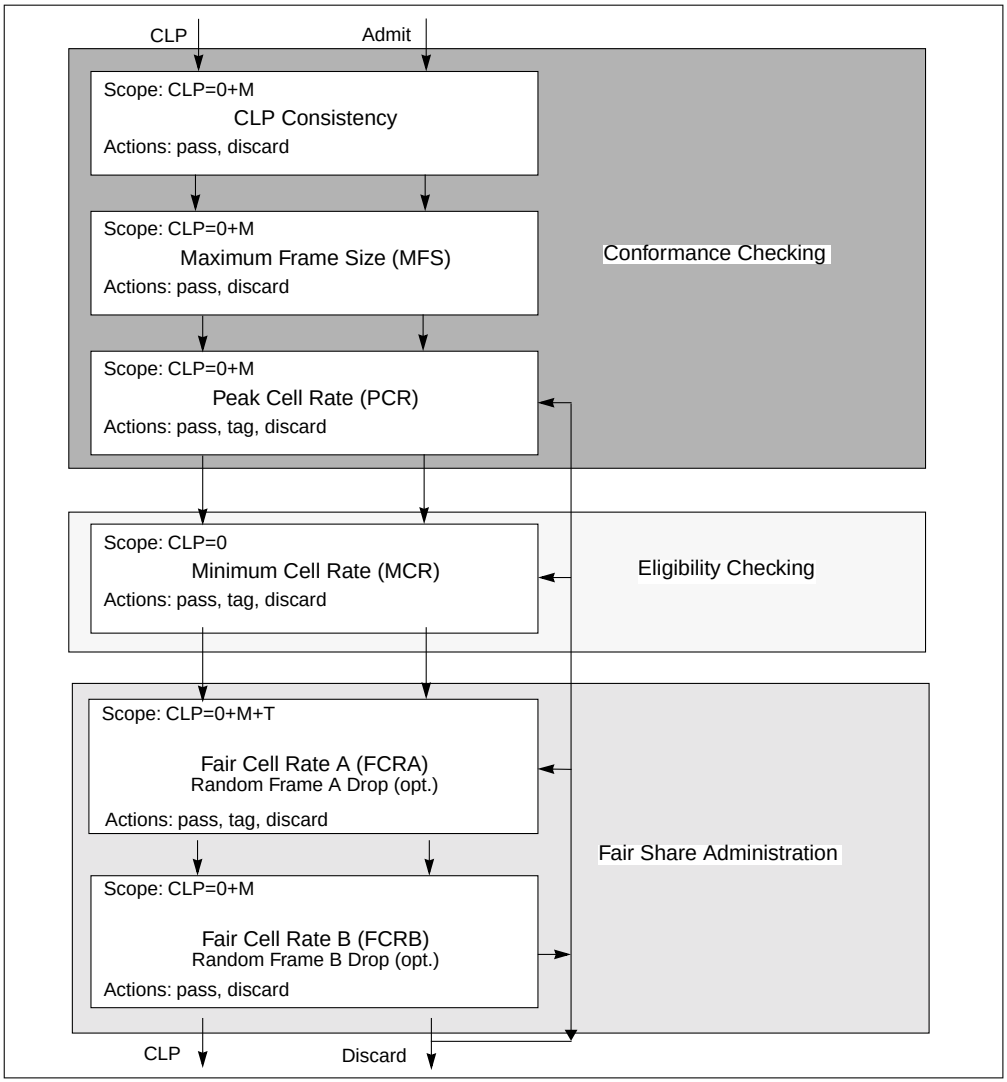


Figure 6-7. GFR UPC Enforcement Stages

6.3.1.1 Global GFR Configurations

GFR operation is enabled on a per-VC basis by setting the extended UPC operating mode (EUOM) field in the context parameter extension table (CPET) to a value of b001. Selecting GFR mode enables options that are controlled by the GFR configuration register (GFRCR).

The treatment of the last cell in a frame undergoing partial packet discard (PPD) is controlled for GFR-mode connections by the PPD accept last cell (PALC) bit of the

GFRCR. Since frames are delineated by the last cell of the frame, admitting the last cell during PPD avoids corrupting the following frame. PALC forces the last cell of a frame to be accepted if any other cell of the frame was accepted, even if the last cell violates UPC or CLP consistency check.

The number of buckets used for enforcement is set in the number of bucket (NBK) field of the common parameter word in the context parameter table (CPT). If GFR fair share administration is required, the NBK field must be set to b00 and four leaky buckets will be used. If the NBK field is set to b11, only three leaky buckets are used and no GFR fair share administration support is possible.

A single leaky bucket is only sufficient to enforce CLP consistency and MFS. Adding a second leaky bucket allows the enforcement of either PCR or MCR. Both PCR and MCR will be enforced only when a third leaky bucket is available.

The CLP transparency configuration of GFR VCs must be handled in a GFR-type and switch fabric specific manner. The following table lists the main configuration options (configuration bits are referred to by name and given in brackets). For additional CLP handling options refer to Section 6.3, “Guaranteed Frame Rate (GFR) Support.”

Table 6-1. GFR CLP Transparency Configurations

GFR Type	Switch Fabric Is Frame-Aware	Switch Fabric Is Not Frame-aware
GFR.1	Pre-enforcement CLP -> Cell Overhead [ICTE = 1, OCFI = 1] Post-enforcement CLP -> Cell Header [HCFE = 1]	Pre-enforcement CLP -> Cell Overhead [ICTE = 1, OCFI = 1] CLP 0 -> Cell Header [HCFE = 0, ICTV = 0]
GFR.2	No CLP -> Cell Overhead [ICTE = 0] Post-enforcement CLP -> Cell Header [default for ICTE = 0]	Post-enforcement CLP -> Cell Overhead [ICTE = 1, OCFI = 0] CLP 0 -> Cell Header [HCFE = 0, ICTV = 0]

6.3.1.2 Cell Loss Priority (CLP) Consistency

CLP consistency checking within a frame can be performed in the first GFR enforcement stage, which does not consume any space in the bucket table. CLP inconsistency may be ignored or enforced with PPD applied to the remainder of the frame. If the PPD admit last cell (PALC) option is enabled, CLP consistency cannot be established in all cases. The last cell of a frame may have an inconsistent CLP bit when compared to the initial part of the frame.

If enabled, the scope of the CLP consistency enforcement is *all* cells (that is, CLP=0+1 cell flows) and cannot be changed.

The enforcement action is determined through the CLP inconsistency behavior selection (CIBS) bit in the context parameter extension table (CPET) of each VC.

In practice, CLP inconsistency may indicate any of the following:

- non-conforming user marking,
- cell misinsertion,
- a missing last cell of a frame, or
- equipment along the VC's path that performs frame-unaware cell tagging.

6.3.1.3 Maximum Frame Size (MFS)

Enforcement of the VC's maximum frame size (MFS) can be performed in the second GFR UPC enforcement stage. Exceeding MFS may be ignored or enforced with PPD applied to the remainder of the frame.

If enabled, the scope of the MFS enforcement is *all* cells (that is, CLP = 0 +1 cell flows) and cannot be changed.

The enforcement action is determined through the setting of the maximum frame size (MFS) field in the first bucket information field of the buckets record for the connection. If the MFS field is zero, no maximum frame size is enforced. If the MFS field is non-zero, the MFS field defines the maximum frame size in terms of cells. If a frame exceeds MFS, PPD is applied to the remainder of the frame. The resulting frame, including the last cell of the frame, is MFS cells long.

When using MC92520 GFR features to support traditional ATM service categories such as UBR or VBR, MFS enforcement requires determining the maximum frame size through configuration or through some other means than signalling at VC setup time. At least a limit corresponding to the maximum AAL5 message size might be used to provide some protection of shared resources.

6.3.1.4 Peak Cell Rate (PCR)

Enforcement of a peak cell rate (PCR) can be performed in the third GFR UPC enforcement stage. Exceeding PCR may be ignored or enforced with the actions PPD or PPT applied to the remainder of the frame, or with the actions EPD or EPT "applied" to the following frame.

PCR is normally enforced using the second leaky bucket. With the exception of frame-based actions, the configuration options and rules to configure the leaky bucket information for PCR enforcement are the same as described in Section 6.1.1, "Cell-Based UPC" and Appendix A.

For the purpose of enforcing PCR for GFR VCs, the scope (SCP field) of the enforcement would be typically set to b11, that is, the CLP = 0 +1 cell flow.

The PCR enforcement action is determined through the TAG bit in the second leaky bucket and the bucket 2 early packet discard (B2EPD) bit in the GFRCR. Configuration options are shown in Table 6-2.

Table 6-2. PCR and MCR Enforcement Actions for GFR

TAG	BnEPD	Action
0	0	Partial packet discard (PPD)
0	1	Early packet discard (EPD)
1	0	Partial packet tagging (PPT)
1	1	Early packet tagging (EPT)

The MC92520 GFR implementation provides different PCR enforcement actions to support different resource management and bandwidth utilization philosophies:

At the conservative end of the spectrum, PPD can be selected to assure that the least amount of local resources are allocated for non-compliant GFR connections. The PPD setting implies that some frames of non-compliant VCs will be partially destroyed and probably discarded at the VC end-point. These partial frames will consume network bandwidth.

If the requirement to use a minimum amount of local resources can be relaxed, better network bandwidth utilization can be achieved when the discard decision is delayed until a new frame starts (when the enforcement action is set to EPD).

The purpose of the enforcement tagging action variants (PPT and EPT) is to identify non-compliant traffic, but leave the decision to discard to the fair cell rate enforcement stage or the switch fabric. PPT may only be used to support local GFR UPC functions.

NOTE:

PPT destroys a frame's GFR conformance if the partial tagging is transmitted by the switching node.

6.3.1.5 Minimum Cell Rate (MCR)

Enforcement of a minimum cell rate (MCR) can be performed in the fourth GFR UPC enforcement stage. Exceeding MCR may be ignored or enforced with the action PPD or PPT applied to the remainder of the frame, or with the actions EPD or EPT applied to the following frame.

MCR is normally enforced using the third leaky bucket. As with PCR enforcement, the configuration options and rules to configure the leaky bucket information for MCR enforcement are the same as described in Section 6.1.1, "Cell-Based UPC" and Appendix A.

For the purpose of enforcing MCR for GFR VCs, the scope (SCP field) of the enforcement would be typically set to b01, that is, the CLP = 0 cell flow.

The MCR enforcement action is determined through the TAG bit in the third leaky bucket and the bucket 3 early packet discard (B3EPD) bit in the GFRCR. The configuration options are listed in Table 6-2 above.

The MC92520 GFR implementation provides different MCR enforcement actions to support different resource management and bandwidth utilization philosophies:

The GFR user expectation is that a switch will use available resources to provide service in excess of the MCR service guarantee. The MCR enforcement stage can merely detect and tag the frames that would exceed the service guarantee. Therefore, the typical enforcement actions for MCR will be configured either for PPT or EPT. The effect of PPT on subsequent fair share allocation is probably somewhat quicker, but PPT is only applicable if it will not result in corruption of frame conformance (that is, for GFR.1 where the original CLP is restored).

The PPD action assures that the least amount of local resources are allocated for non-eligible frames. This setting implies that all non-eligible frames will be destroyed and discarded at the VC end-point. If the requirement to use a minimum amount of local resources is relaxed, better network bandwidth utilization can be achieved when the discard decision is delayed until a new frame starts, that is, when the enforcement action is set to EPD. Note that although the PPD and EPD enforcement actions result in compliance with the GFR specification, the expectations of the GFR user to get service in excess of the service guarantee will *not* be satisfied.

6.3.1.6 Fair Cell Rate (FCR)

The fair cell rate (FCR) represents a VC-specific rate that considers the availability and fair share of available resources for GFR VCs. The MC92520 GFR implementation supports the use and enforcement of one (FCRA) or two FCRs (FCRA and FCRB): FCRA represents the fifth, and FCRB represents the sixth and last enforcement stages of GFR UPC.

The behavior of the FCR enforcements differs from that of a standard cell rate enforcement. The scope of a standard rate enforcement determines both the cell flow that is used to fill the leaky bucket and the cell flow that is policed. Both FCR enforcements share a common leaky bucket content (BKC); thus, they also share a common leaky bucket fill scope (FS). But the two FCR enforcements have separate leaky bucket limits (BKLs) and associated action scopes that define what cell flow is policed. This separation of the fill scope action scope is essential for GFR because it allows the MC92520 to enforce the GFR service guarantee for the combination of CLP = 0 + 1 cell flow below MCR while policing the CLP = 1 cell flow above MCR (via FCR).

In addition, the MC92520 provides an option to discard all frames exceeding FCRA, independent of the FCRA action scope and the frame's CLP bit value. This option must be used whenever the MC92520 needs to assure that the combined CLP = 0 + 1 flow never exceeds FCRA. Note that the FCR leaky bucket fill scope is typically set to CLP = 0 + 1, but the FCR enforcement action polices the CLP = 1 flow only. This implies that a conformant CLP = 0 frame will always be admitted, even if the FCR bucket limit was reached before the first cell of the frame was received. In practice there are two choices: 1) FCR can be set to a lower value reflecting the maximum burst size (MBS) of the CLP = 0

flow, or 2), a hard FCR bucket limit is enforced. The former will cause frequent underutilization of bandwidth, whereas the latter will cause an occasional violation of the service guarantee by discarding CLP = 0 frames.

FCRs are enforced using the fourth leaky bucket, additional fields in the first leaky bucket, and the GFR configuration register (GFRCR).

The fill scope for the leaky bucket content shared by both FCR enforcements is defined in the bucket 4 fill scope (B4FS) field of the GFRCR. Note that the CLP = 1 flow includes both user-marked and UPC-tagged cells.

Except for policing scope and actions, the options and rules for the FCRA configuration in the fourth leaky bucket are the same as described in Section 6.1.1, “Cell-Based UPC” and Appendix A.

The configuration of the FCRB bucket limit (BKL and LMS) together with the FCRB action scope (SCP) is located in the first leaky bucket at their standard locations in the second 32-bit word. As described earlier, FCRB shares the bucket content (BKC and TS) located in the fourth leaky bucket with FCRA. The FCRB policing decision to tag or discard frames is implied as discard (TAG = 0) and cannot be changed.

All FCR policing actions are applied to full frames. Dependent on the (implicit or explicit) setting of the TAG bit and SCP field, the associated FCR enforcement action can be selected as defined in Table 6-3.

Table 6-3. FCR Enforcement Actions

TAG	SCP	Action
0	00	No action
0	01	EPD on CLP=0 frames
0	10	EPD on user-marked CLP=1 frames
0	11	EPD on both user-marked and UPC-tagged CLP=1 frames
1	00	No action
1	01	EPT on CLP=0 frames
1	10	EPT on CLP=0 frames and EPD on user-marked CLP=1 frames
1	11	EPT on CLP=0 frames and EPD on both user-marked and UPC-tagged CLP=1 frames

The optional hard limiting of the combined CLP = 0 + 1 flow to the FCRA bucket limit is enabled by setting the bucket 4 all-packet discard (B4APD) bit in the GFRCR.

6.3.1.7 FCR Administration

Dynamic per-VC updates of the FCRs are only feasible for a small number of GFR VCs. In many applications it is impossible to dynamically update the FCRs for all connections in response to changing switch-internal resource allocations. For these cases, the MC92520 provides the ability to assign individual GFR VCs to VC groups (up to 64 groups) and update the FCRs for all VCs within a group with a single register access. Note that the

MC92520 does not imply specific meaning with VC groups. In most cases, VC groups will relate to some resource shared by all VCs in a group (that is, to a shared switch fabric input or output queue), but the association choice is established and maintained by the MC92520 application.

Updating the FCRs of many connections with a single access is achieved by adjusting the cell arrival period (CAP) field with a programmable multiplier every time a cell is processed by the FCR enforcement. The multiplier is identified through the CAP multiplier index (CAPMI) field in the first leaky bucket of each GFR VC. The CAPMI is used to index into the cell arrival period multiplier (CAPM) control register array. Each individual CAPM register holds an 8-bit value which in turn is used to form the CAP multiplier by dividing CAPM with 256. A CAPM value of 0 is interpreted as a CAP multiplier of 1. Thus the effective CAP ($CAP_{effective}$) is:

$$CAP >= CAP_{effective} >= CAP * 1 / 256$$

And the associated $FCR_{effective}$ is:

$$FCR <= FCR_{effective} <= FCR * 256$$

The setting of the CAPMI field is ignored if GFR policing is configured to use less than four leaky buckets. The FCR CAP multiplier capability can be turned off by clearing the CAPM00 register and the CAPMI field in the first leaky bucket for all such VCs. If the CAP multiplier capability is used, the FCR CAP of a given GFR VC is typically set according to MCR or a similar (low) cell rate for which service can be guaranteed by the switching node. The GFR VC is linked to a VC group through CAPMI and the associated CAPM register is set such that $CAP_{effective}$ approximates the current fair cell rates based on the availability of resources for the VC group. During operation, a microprocessor monitors resource allocation or utilization and adjusts the relevant CAPMs as needed.

The FCR administration capabilities described above support a significant range of effective FCR values. Depending on the type of traffic flowing through the VC, the usable range and the magnitude of changes may be limited though. For example, if a VC carries TCP traffic, large reductions in the effective FCR (and the associated discarding of frames) will lead to the triggering of unnecessarily drastic back-off algorithms and create the potential for resource use oscillations. To avoid this behavior, large CAPM increases and the lower values of CAPM (that is, a transition from one to two results in a 50% rate decrease) should be used cautiously.

6.3.1.8 Congestion Avoidance Through Random Frame Drop (RFD)

One of the major GFR applications is carrying TCP/IP traffic. Support for TCP/IP requires that a switch node deal effectively with the behavior of TCP/IP during congestion. The MC92520 provides the capability to define two random frame drop (RFD) functions that can be used to selectively discard frames in order to prevent congestion. The validity of this approach has been documented in a significant number of publications and is typically referred to as random early drop (RED) or similar terms.

In the following sections, the characteristics and configuration of the MC92520's RFD capability are described in detail after a short introduction to the TCP/IP related congestion issue.

6.3.1.8.1 GFR and TCP/IP Congestion

TCP provides a reliable transport service and uses protocol data units (PDUs) to transfer data and manage data transmissions as well as PDU flow rates. When TCP/IP traffic is carried over ATM, each PDU is encapsulated through a series of ATM protocol layers, ultimately resulting in an AAL5 frame.

TCP is a greedy transport protocol, that is, it always attempts to increase the flow rate at a rate of 1 PDU per round trip time (RTT) until all data is sent or a PDU is lost. There is no explicit mechanism by which a destination node indicates congestion to a source node. TCP detects congestion only when at least 1 of its PDUs is lost (not acknowledged by the peer TCP layer). Through a variety of mechanisms, the average flow rate is effectively halved in response to a lost PDU. But if more than 1 PDU is lost, the flow rate is drastically lowered.

In respect to GFR carrying TCP/IP traffic, a switching node implementation is faced with the following issues:

- The resource utilization of individual TCP/IP connections is likely to follow a saw-tooth pattern, driven by a rate oscillating between 50% and 100% of the assigned FCR. The average resource utilization can be expected to be around 75%.
- Dynamic lowering of the FCR (to re-allocate available resources) will cause the loss of more than 1 PDU, that is, an increase in the oscillation amplitude, leading to further underutilization of resources.

RFD addresses the problem by discarding some frames in an aggregate TCP/IP stream before congestion is reached. This will trigger a back-off behavior for a few randomly selected TCP/IP connections and thus avoid oscillations that would be caused if many TCP/IP connections backed-off at the same time.

6.3.1.8.2 RFD Configuration Overview

The configuration of RFD centers around the two FCR leaky bucket limit parameters (BKLA and BKLB) described above in Section 6.3.1.6, "Fair Cell Rate (FCR)," and additional parameters. There are two linear probability functions: one associated with FCRA and one associated with FCRB. Both probability functions use one parameter: the current FCR-specific leaky bucket content (BKC). The probability function is shown in Figure 6-8 below and is defined by the following parameters:

- The FCR-specific leaky bucket limit (BKLA or BKLB, respectively).
- The "Probability Step Resolution" field (PSRA or PSRB), which defines the resolution of the BKC that will be used to select the frame drop probability as the BKC nears the limit BKLA or BKLB.

- A minimum and maximum probability (MINPB and MAXPB), representing lower and upper limits on both probability functions.

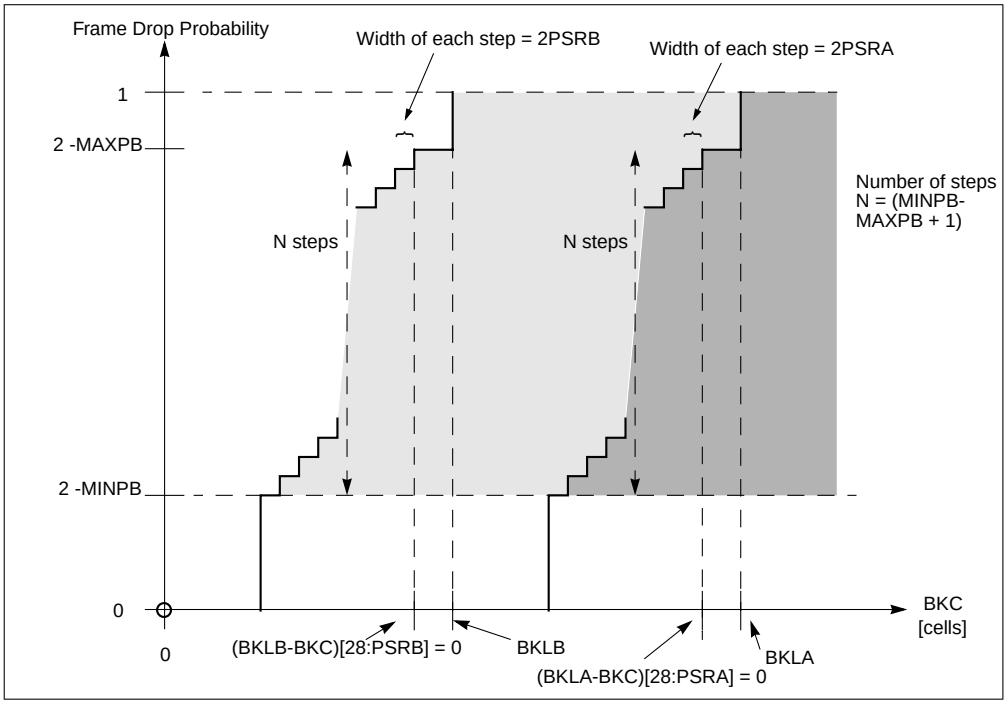


Figure 6-8. Random Frame Drop (RFD) Probability Functions

RFD is enabled by setting a non-zero value for the minimum probability (MINPB) field located in the second 32-bit word of the first leaky bucket. If GFR policing is setup to use less than four leaky buckets, the setting of MINPB is ignored and no RFD function is available.

6.3.1.8.3 Configuration of RFD Probability Function Limits

The minimum probability (MINPB) field, located in the second 32-bit word of the first leaky bucket, is used to determine the lowest non-zero probability of action (discard or tag) for GFR frames received while the leaky bucket content (BKC) is below the leaky bucket limit (BKLA or BKLB, respectively). The minimum probability cut-off point $Prob_{min}$ is expressed in powers of 2,

$$Prob_{min} = 2^{-MINPB}$$

and ranges from 2^{-1} (0.5) to 2^{-15} (3.0518e-5).

The maximum probability (MAXPB) field, located in the first 32-bit word of the first leaky bucket, is used to determine the highest probability of action (discard or tag) for GFR

frames received while the leaky bucket content (BKC) is below the leaky bucket limit (BKLA or BKLB, respectively). If the BKC is equal to or exceeds the bucket limit, action is taken on all frames. The maximum probability cut-off point $Prob_{max}$ is expressed in powers of 2,

$$Prob_{max} = 2^{-MAXPB}$$

and ranges from 2^{-0} (1.0) to 2^{-15} (3.0518e-5). Note, MAXPB must express a probability greater than or equal to the one expressed by MINPB; therefore, the value programmed to MAXPB must be less than or equal to the one programmed to MINPB.

The difference between the programmed values of MINPB and MAXPB defines the number and value of the possible probabilities of action, shown as steps in Figure 6-8. For programmed values MINPB and MAXPB, the possible probabilities of action (steps shown in the figure) are:

$$\{2^{-MINPB}, 2^{-(MINPB-1)}, \ldots, 2^{-(MAXPB+1)}, 2^{-MAXPB}\}$$

6.3.1.8.4 Configuration of the RFD Probability Function

The parameters PSRA and PSRB for the RFD probability functions for FCRA and FCRB are defined in the first 32-bit word of the first leaky bucket. PSRA and PSRB are coded in a 5-bit field and define the resolution of the BKC value used to step between each possible probability as defined in the previous paragraph. For example, programming a value of b01011 (decimal 11) for PSRA would indicate that the probability of action would step to the next possible probability for each 2^{11} difference in BKC.

6.3.1.8.5 Understanding RFD from the Implementation Standpoint

To better understand the usage of MAXPB, MINPB, PSRA, and PSRB, it may help to understand how these parameters are used within the MC92520.

When the first cell of a frame arrives, the frame drop rate is determined from the BKC, BKLx, PSR*x*, and MAXPB. The parameter “BSRx” will be used herein to indicate the bucket space remaining before it reaches the bucket limit, that is, BKLx-BKC.

```
Probability value = BSRx[28:PSRx] + MAXPB
```

If this probability value is greater than MINPB, corresponding to a drop rate less than 2^{-MINPB}, then no action is taken. If the bucket contents are greater than the bucket limit, then action is taken on all frames within scope.

Here is an example:

MAXPB is 3 (that is, 2^{-3} frame drop rate, or 1 in 8 chance of drop)

MINPB is 7 (2^{-7} frame drop rate, or 1 in 128 chance of drop)

Bucket contents are less than or equal to bucket limit.

PSRx is 16, so bits 28:16 of BSRx are selected for drop rate calculations.

If BSRx[28:16] are 0, indicating that the bucket contents have nearly reached the bucket limit, then the probability value is 3 (0 + MAXPB), and the probability of action is 2^{-3}.

If BSRx[28:16] equal 1, the probability of action is $2^{-(1 + MAXPB)}$ or 2^{-4}.

If BSRx[28:16] equal 2, the probability of action is 2^{-5}.

If BSRx[28:16] equal 3, the probability of action is 2^{-6}.

If BSRx[28:16] equal 4, the probability of action is 2^{-7}.

If BSRx[28:16] are 5 or more, then the actual probability value is 8 or more, which is greater than MINPB. Therefore, no drop will occur.

6.3.2 GFR Configuration Examples

Examples of guaranteed frame rate configurations are explained in the following sections.

6.3.2.1 GFR Conformance and Eligibility Filter

One example is a GFR configuration that does not use the MC92520's FCR and RFD capabilities. ATM traffic is policed to be conformant in a strict sense, and frame rates exceeding MCR are tagged for further processing in the switch fabric.

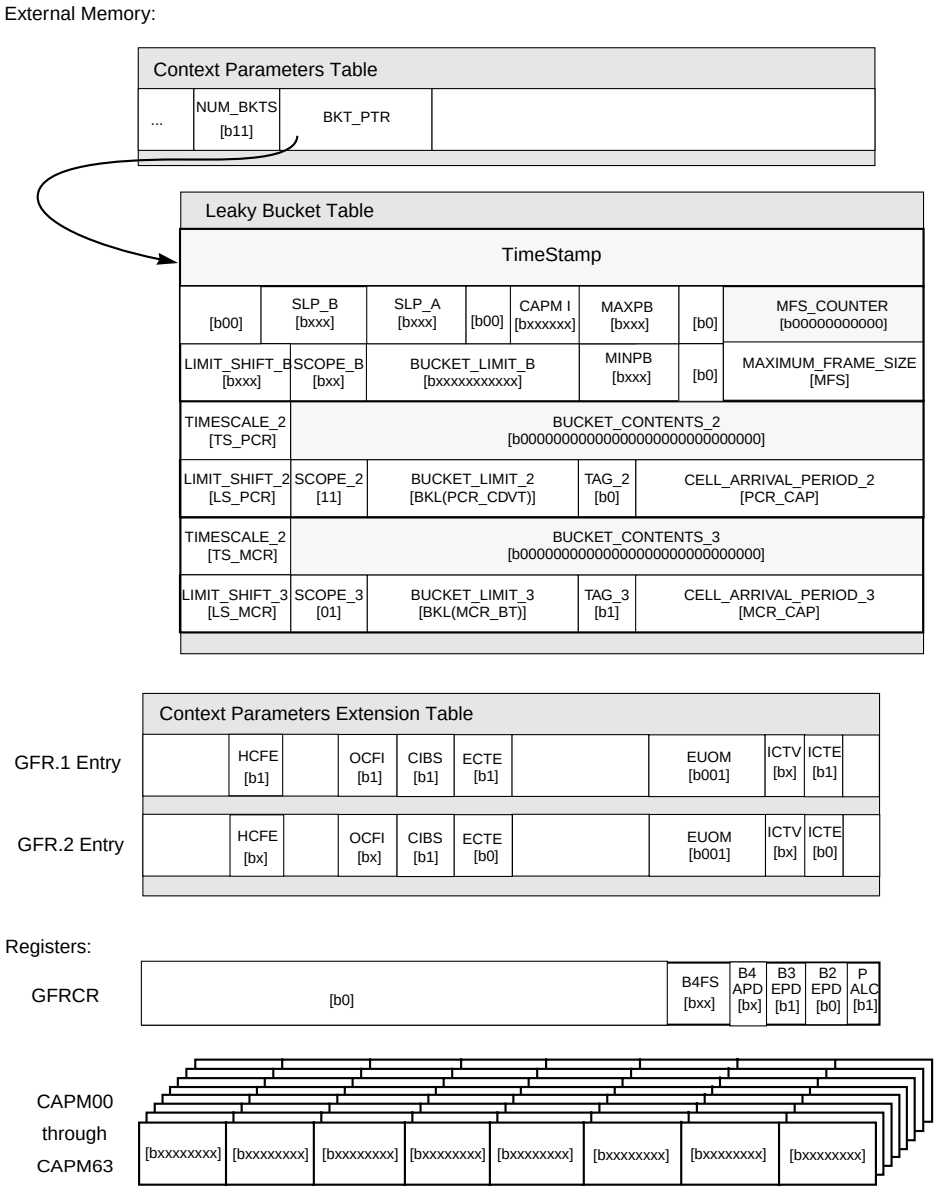


Figure 6-9. GFR Conformance and Eligibility Filter

In effect, the MC92520 acts as a GFR conformance and eligibility filter, that is, it is assumed that the fabric is GFR- or frame-aware.The following steps result in a configuration as shown in Figure 6-9 above:

1. Enable global ingress and egress CLP bit management by setting IPCR[IGCTE] and EPCR[EGCTE], respectively, and define suitable byte and bit positions for a preserved CLP bit in the cell overhead by setting the OCBI and OCBL fields in the CTOR register for the ingress side, and the EOBY and EOBI fields in the ESOIR1 register for the egress side. (Not shown in Figure 6-9.)
2. Define the global GFR UPC to perform the PCR conformance policing actions on partial frames (set B2EPD bit to b0), perform MCR eligibility policing action on whole frames (set B3EPD bit to b1), and configure the MC92520 to always admit the last cell of a GFR VC frame (set PALC bit to b1) in the GFRCR register.
3. Perform the following configuration selections in the context parameter extension table:
   - For each GFR VC, enable GFR-type UPC (set EUOM field to b001) and CLP consistency enforcement (set CIBS bits to b1).
   - For each GFR.1 VC, enable CLP bit management for both ingress and egress (set ICTE and ECTE bits to b1), select pre-enforcement as the source of the CLP bit in the overhead (set OCFI bit to b1), and post-enforcement as the source of the CLP bit in the cell header (set HCFE bit to b1).
   - For each GFR.2 VC, disable CLP bit management for both ingress and egress (set ICTE and ECTE bits to b0). (This implies that the post-enforcement is the source of the CLP bit in the header.)
4. Set the number of leaky buckets to b11 and a suitable leaky bucket pointer in the NBK and BKT_PTR fields of the common parameter word of the context parameter table.
5. Set the first word of the leaky bucket table entry to the value read from the CLTM register to prime the TIME-STAMP field.
6. Clear the second word of the leaky bucket table entry to initialize the MFS_COUNTER. (All other fields of the second word are not applicable for this configuration example.)
7. Set the maximum frame size in the MFS field of the third word of the leaky bucket table entry. (All other fields of the third word are not applicable for this configuration example.)
8. Configure PCR related policing parameters as described in Section 6.1.1 and Appendix A in the fourth and fifth word of the leaky bucket table entry. Set the SCOPE_2 field to b11 to police the combined CLP=0+1 flow, and the TAG_2 field to b0 to select PPD (due to the cleared B2EPB bit).

9. Configure MCR related policing parameters as described in Section 6.1.1 and Appendix A in the sixth and seventh word of the leaky bucket table entry. Set the SCOPE_3 field to b01 to police the CLP=0 flow, and the TAG_3 field to b1 to select EPT (due to the set B3EPB bit).

6.3.2.2 GFR Support for a Switch Fabric That is Not Frame-Aware

Another example shows a GFR configuration that uses the MC92520's FCR and RFD capabilities. ATM GFR traffic is policed to be conformant in a strict sense and frame rates exceeding MCR are temporarily tagged for further processing with the FCR enforcer stages. Dependent on the VC's GFR type, egress-side processing is configured to either restore the original CLP value (GFR.1), or pass an updated CLP value (GFR.2) reflecting that a frame may, although passed, exceed the service guarantee. In addition, the example configuration shows how the FCR administration parameters are configured. In effect, the MC92520 provides both GFR conformance and eligibility filtering, as well as support to manage resources for service requests beyond the GFR service guarantee without relying on support from a GFR- or frame-aware fabric.

External Memory:

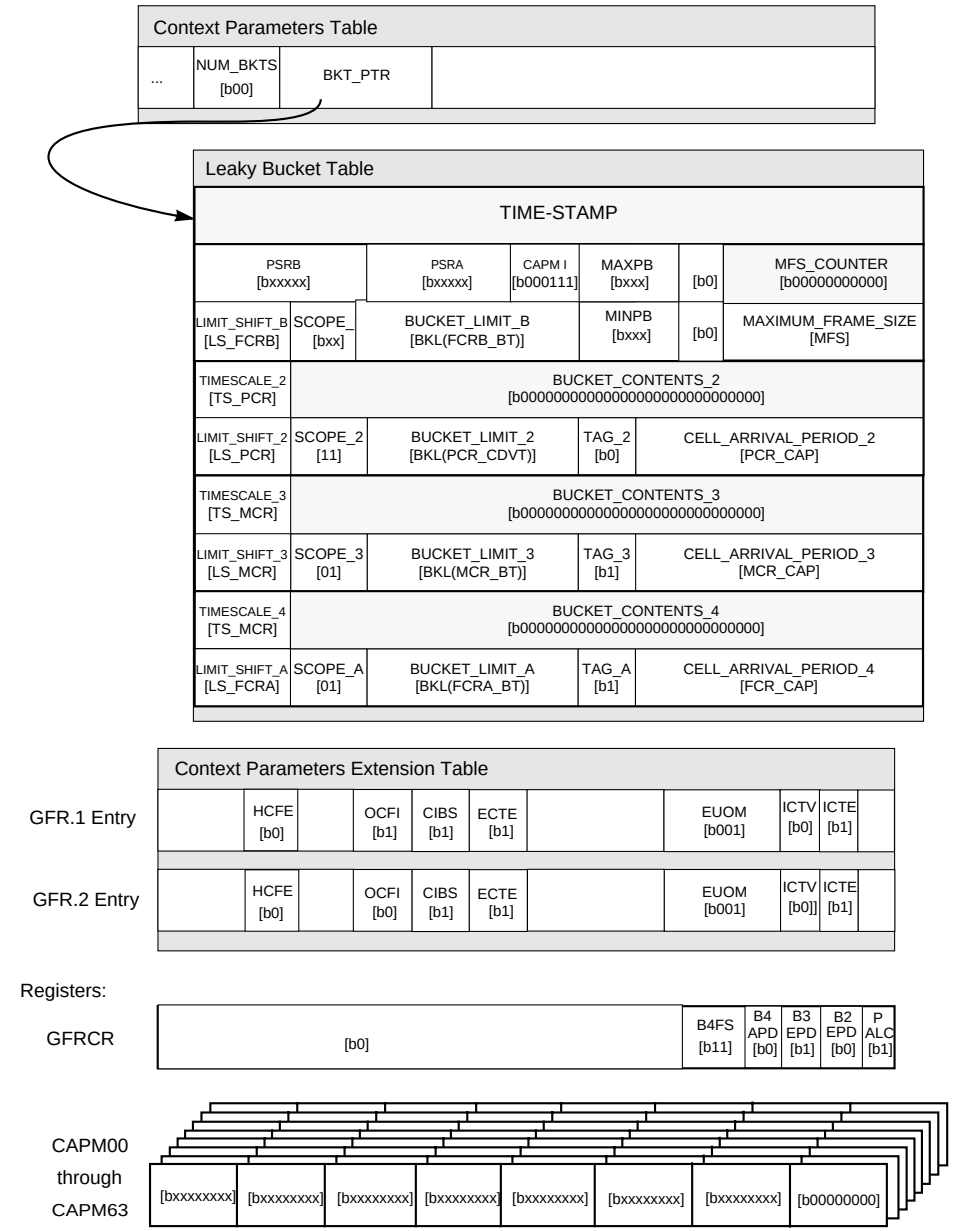


Figure 6-10. GFR Conformance/Eligibility Filter and Fair Cell Rate Administration

The following steps result in a configuration as shown in Figure 6-10 above:

1. Enable global ingress and egress CLP bit management by setting IPCR[IGCTE] and EPCR[EGCTE], respectively, and define suitable byte and bit positions for a preserved CLP bit in the cell overhead by setting the OCBI and OCBL fields in the CTOR register for the ingress side, and the EOBY and EOBI fields in the ESOIR1 register for the egress side. (Not shown in Figure 6-10.)
2. Define the global GFR UPC function to perform PCR conformance policing actions on partial frames (set B2EPD bit to b0), perform MCR eligibility policing action on whole frames (set B3EPD bit to b1), set the FCRx bucket fill scope to the combined CLP0+1 flow (set B4FS field to b11), turn the FCRA hard limiting for the combined CLP0+1 flow off (set B4APD bit b0), and configure the MC92520 to always admit the last cell of a GFR VC frame (PALC bit to b1) in the GFRCR register.
3. Perform the following configuration selections in the context parameter extension table:
   - For each GFR VC, enable GFR-type UPC (set EUOM field to b001), enable CLP consistency enforcement (set CIBS bit to b1), enable CLP bit management for both ingress and egress (set ICTE and ECTE bits to b1), and enforce 0 as the CLP bit value in the cell header passed to the fabric (set HCFE and ICTV bits to b0).
   - For each GFR.1 VC, select pre-enforcement as the source of the CLP bit in the overhead (set OCFI bit to b1).
   - For each GFR.2 VC, select post-enforcement as the source of the CLP bit in the overhead (set OCFI bit to b0).
4. Set the number of leaky buckets to 4 (set NBK field to b00) and a suitable leaky bucket pointer in the BKT_PTR field of the common parameter word of the context parameter table.
5. Set the first word of the leaky bucket table entry to the value read from the CLTM register to prime the TIME-STAMP field.
6. Clear the second word of the leaky bucket table entry to initialize the MFS_COUNTER. (See FCR configuration below for all other fields that must be set in the first leaky bucket word.)
7. Set the maximum frame size in the MFS field of the third word of the leaky bucket table entry. (See FCR configuration below for all other fields that must be set in the first leaky bucket word.)
8. Configure PCR related policing parameters as described in Section 6.1.1, “Cell-Based UPC” and Appendix A in the fourth and fifth word of the leaky bucket table entry. Set the SCOPE_2 field to b11 to police the combined CLP=0+1 flow, and the TAG_2 field to b0 to select PPD (due to the cleared B2EPB bit).

9. Configure MCR related policing parameters as described in Section 6.1.1, “Cell-Based UPC” and Appendix A in the sixth and seventh word of the leaky bucket table entry. Set the SCOPE_3 field to b01 to police the CLP=0 flow, and the TAG_3 field to b1 to select EPT (due to the set B3EPB bit).
10. Configure FCRx related policing parameters LIMIT_SHIFT_x, TIME_SCALE_x, and BKL_x, as described in Section 6.1.1, “Cell-Based UPC” and Appendix A. Configure the other FCRx related policing parameters PSRx, SCP_x, MINPB, MAXPB, and CAPMI, as described inSection 6.3.1.6, “Fair Cell Rate (FCR)” and subsequent chapters. All parameters mentioned above are located in the first, second, seventh, and eighth word of the leaky bucket table entry.
    - For each GFR VC, select an FCRx cell arrival period, a time scale shift value, zero as initial leaky bucket content, and store all parameters in the seventh and eight word of the leaky bucket table.
    - For each GFR VC, determine whether 1- or 2-stage FCR policing is needed. If 1-stage policing is sufficient, select either the FCRA or the FCRB fields to store parameters and set the scope field of the FCRx not selected to ‘No action’. For example, set the BKL_B, LIMIT_SHIFT_B, and SCP_B fields to appropriate values and SCP_A to b00. If 2-stage policing is needed, store all FCRA and FCRB parameters in the second and eight word of the leaky bucket table. Note, FCRA supports an extended action scope using the TAG_A bit to extend the number of supported action scopes.
    - For each GFR VC, determine whether FCR administration needs to be applied. If so, select an appropriate system resource that can be used to drive the CAP multiplier changes, assign an unused CAPMnn register, and store the selected CAPMnn index in the CAPMI field of the first word in the leaky bucket entry. Otherwise, store a well-known, pre-defined CAPMnn index in the CAPMI field that identifies a CAPMnn register initialized with 0 (i.e., $FCRx_{effective}$ = FCRx).
    - For each GFR VC, determine whether RFD need to be applied. If not, store a value of b0000 in the MINPB field of the second word of the leaky bucket table. If RFD should be applied, select an appropriate maximum and minimum frame drop probability and store them in the MAXPB and MINPB fields of the first and second word of the leaky bucket table, respectively. Note that frame drop probability cut-off points are applied to both FCRA and/or FCRB policing as necessary. Furthermore, dependent on the choice of 1-stage or 2-stage FCR policing, select 1 or 2 PSRx values and add them in the configuration of the first word in the leaky bucket table.
11. If the CAPMnn register selected in the previous step has not been already initialized, calculate an initial multiplier and configure the selected CAP multiplier register. (Use the selected CAPMI value from the previous step as an index into the CAPMnn register array.) An initial value of 0 results in $FCRx_{effective}$ = FCRx. For more information on CAPMnn use, see Section 6.3.1.7, “FCR Administration.”

6.4 Available Bit-Rate Support

The MC92520 provides a full available bit rate (ABR) solution for ‘switch behavior’ relative-rate (RR) marking and egress flow connection indicator (EFCI) marking in accordance with TM-4.0. It also provides hooks for the switch fabric in order to give higher priority to resource management (RM) cells. Following is a list of available bit rate features:

- Performs RR marking on forward resource management (FRM) and backward resource management (BRM) cells on selected connections. This feature is enabled either by control registers or by fields that it gets from the overhead of cells that are received from the switch fabric.
- Performs EFCI marking on non-RM cells whose PTI[2] = 0 on selected connections. This feature is enabled either by control registers or by fields that it gets from the overhead of cells that are received from the switch fabric.
- Resets EFCI on non-RM cells whose PTI[2] = 0, on selected connections.
- Checks CRC on received RM cells and generates CRC for transmitted RM cells in both the ingress and the egress flow.
- Provides different priority to RM cells.
- Can copy RM cells to the microprocessor or remove them from the flow.

6.4.1 Resource Management (RM) Cell Definition

A cell is a resource management (RM) cell if and only if at least one of the following conditions are met:

- The cell belongs to a VC connection and its PTI = 6.
- The cell belongs to a VP connection, its VCI = 6 and its PTI = 6.
- The cell belongs to a VP connection, its VCI = 6 and the VP RM cell PTI (VPRP) bit is set.

Figure 6-11 shows the RM cell fields and their position within the RM cell.

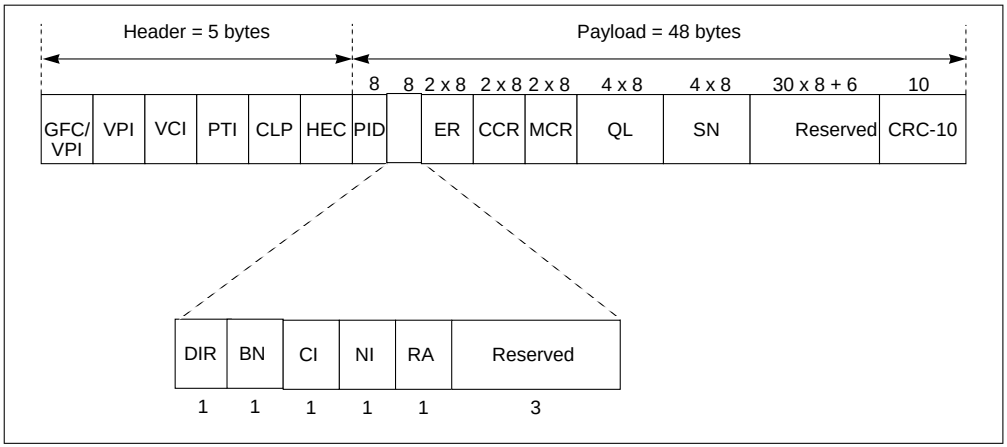


Figure 6-11. RM Cell Fields

The following list provides short definitions of the RM cell fields (detailed descriptions can be found in the ATM Forum Traffic Management Specification):

- PID = Protocol ID (ABR service = 1)
- DIR = Direction (0 = forward RM cell; 1 = backward RM cell)
- BN = Backward explicit congestion notification (0 = generated by source; 1 = not generated by the source)
- CI = Congestion indication
- NI = No increase
- RA = Request acknowledge, which is not used in ABR by the ATM Forum.
- ER = Explicit rate
- CCR = Current cell rate
- MCR = Minimum cell rate
- QL = Queue length
- SN = Sequence number
- CRC-10 = CRC as used for OAM cells

6.4.2 Cell Marking (CI, NI, PTI)

The MC92520 accomplishes cell marking, setting the CI, NI, or PTI bits, for resource management and data cells. Both FRM and BRM cells can be marked. The marking function is implemented at both the ingress and egress flow of the switch fabric, and the bit that is set depends on the flow status.

Figure 6-12 illustrates two MC92520 devices connected to a switch fabric. For the sake of simplicity, consider an ABR flow that goes from left to right. This means that data cells are flowing from left to right, forward resource management (FRM) cells are flowing from left to right, and backward resource management (BRM) cells are flowing from right to left.

The switch marks FRM and data cells flowing downstream and BRM cells flowing upstream. This switch function can be implemented in the ingress flow of the first MC92520 and in the egress flow of the second MC92520. The first MC92520 marks cells because of the ingress flow status (that is, ingress flow congestion) while the second MC92520 marks cells because of the egress flow status.

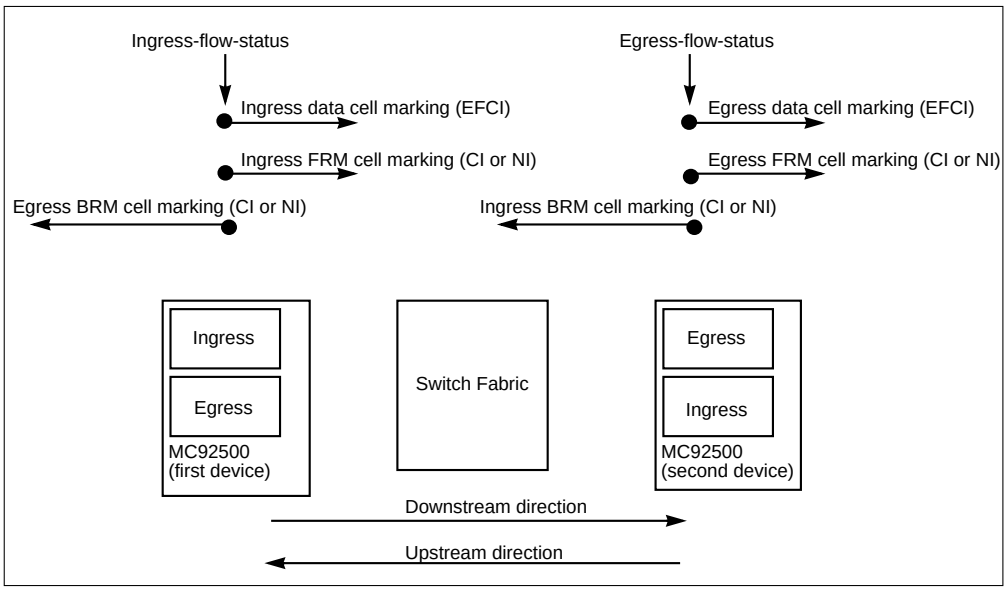


Figure 6-12. MC92520 to Switch Connections

The MC92520 can make these responses to the ingress flow status:

- Perform EFCI marking on ingress cells (that is, set the PTI[1] bit in cells on which PTI[2] = 0).
- Set CI or NI in ingress FRM cells.
- Set CI or NI in egress BRM cells.

The MC92520 can make these responses to the egress flow status:

- Perform EFCI marking on egress cells (that is, set the PTI[1] bit in cells on which PTI[2] = 0).
- Set CI or NI in egress FRM cells.
- Set CI or NI in ingress BRM cells.

Figure 6-13 presents an overview of the MC92520 marking scheme.

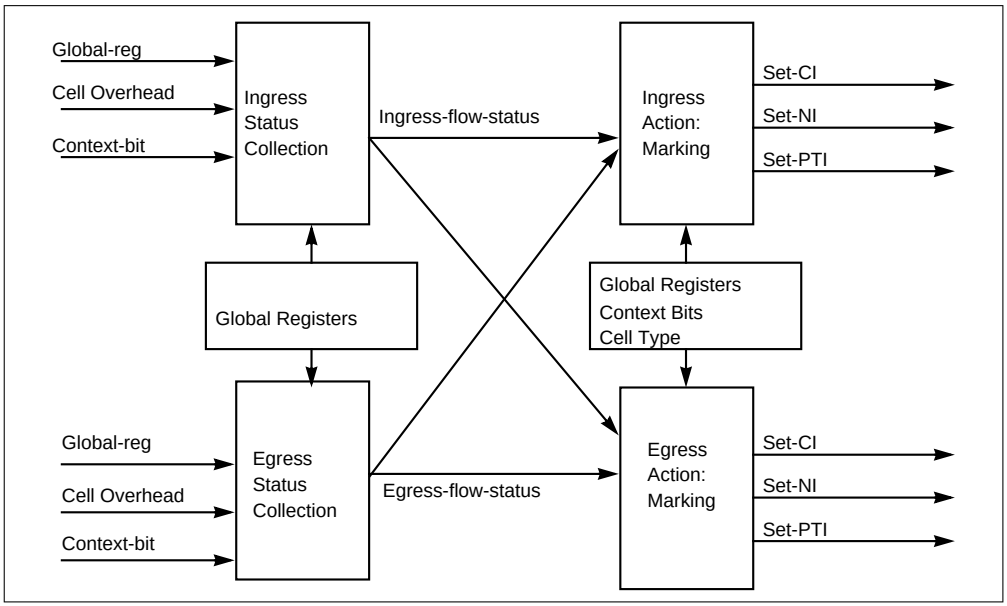


Figure 6-13. MC92520 Marking Scheme

This scheme shows that there are various ways to inform the MC92520 to mark a cell due to the ingress flow or egress flow status. It also shows that the status of the ingress and egress flow (as determined by the global registers, a context bit, and the cell type; see Section 6.4.2.1, “Sources for Ingress Flow Status” and Section 6.4.2.2, “Sources for Egress Flow Status,” below) impact the decision of setting CI, NI, and PTI.

6.4.2.1 Sources for Ingress Flow Status

The ingress flow status is gathered from the following three sources:

- Global register—The switch fabric can notify the MC92520 to mark cells because of ingress flow status by setting the global ingress ABR mark enable (IAME) bit in the ingress processing control register (IPLR). See Section 7.2.5.20, “Ingress Processing Control Register (IPLR).” In this case, the MC92520 can be programmed to mark cells in the ingress or the egress.

- Cell overhead—The switch fabric can notify the MC92520 to mark cells because of the ingress flow status of connection *n* by setting the overhead ingress flow status (IFS) bit in the overhead of egress cells belonging to that connection. The bit location in the overhead is programmed using the IFS byte location (EIBY) bit and the IFS bit location (EIBI) bit in the egress switch overhead information register 1 (ESOIR1). See Section 7.2.6.9, “Egress Switch Overhead Information Register 1 (ESOIR1).” This bit is enabled by the global IFS enable (EIAS) bit in the egress switch interface configuration register (ESWCR). See Section 7.2.6.6, “Egress Switch Interface Configuration Register (ESWCR).” In this case, the MC92520 can be programmed to mark egress BRM cells.
- Context bit—The switch fabric can notify the MC92520 to mark cells because of the ingress flow status of connection *n* by setting the overhead ingress flow status (IFS) bit in the overhead of egress cells belonging to that connection. The bit location in the overhead is programmed using the IFS byte location (EIBY) bit and the IFS bit location (EIBI) bit in the egress switch overhead information register 1 (ESOIR1). See Section 7.2.6.9, “Egress Switch Overhead Information Register 1 (ESOIR1).” This bit is enabled by the global IFS enable (EIAS) bit in the egress switch interface configuration register (ESWCR). See Section 7.2.6.6, “Egress Switch Interface Configuration Register (ESWCR).” When the MC92520 receives the cell, it copies the bit into the connection ingress flow status (CIFS) bit in the common parameters extension word of connection *n*. In this case, the MC92520 can be programmed to mark ingress FRM cells or perform EFCI-marking.

Figure 6-14 shows the ingress-flow-status logic.

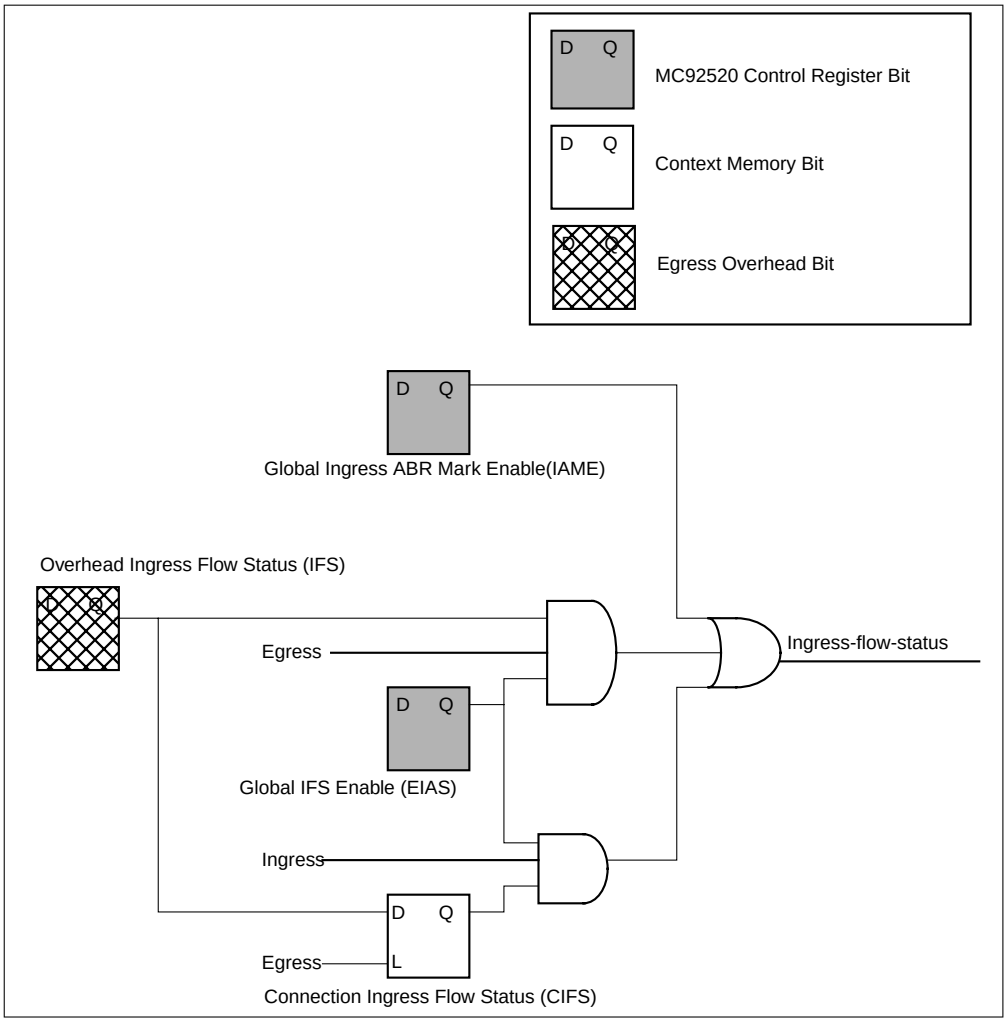


Figure 6-14. Ingress-Flow-Status Logic

6.4.2.2 Sources for Egress Flow Status

The egress flow status is gathered from the following three sources:

- Global register—The switch fabric can notify the MC92520 to mark cells because of the egress flow status by setting the global egress ABR mark enable (EAME) bit in the egress processing control register (EPLR). The MC92520 can be programmed to mark cells in the ingress or the egress.
- Cell overhead—The switch fabric can notify the MC92520 to mark cells because of the egress flow status of connection *n* by setting the overhead egress flow status (EFS) bit in the overhead of egress cells belonging to that connection. The location of this bit in the overhead is programmed using the EFS byte location (EEBY) field and the EFS bit location (EEBI) field in the egress switch overhead information register 1 (ESOIR1). See Section 7.2.6.9, “Egress Switch Overhead Information Register 1 (ESOIR1).” This bit is enabled by the global EFS enable (EEAS) bit in the egress switch interface configuration register (ESWCR). See Section 7.2.6.6, “Egress Switch Interface Configuration Register (ESWCR).” In this case, the MC92520 can be programmed to mark egress FRM cells or perform EFCI-marking.
- Context memory—The switch fabric can notify the MC92520 to mark cells because of the egress flow status of connection *n* by setting the overhead egress flow status (EFS) bit in the overhead of egress cells belonging to that connection. When the MC92520 receives that cell, it copies the bit into the connection egress flow status (CEFS) bit in the common parameters extension word of connection *n*. The MC92520 can be programmed to mark ingress BRM cells.

Figure 6-15 shows the egress-flow-status logic.

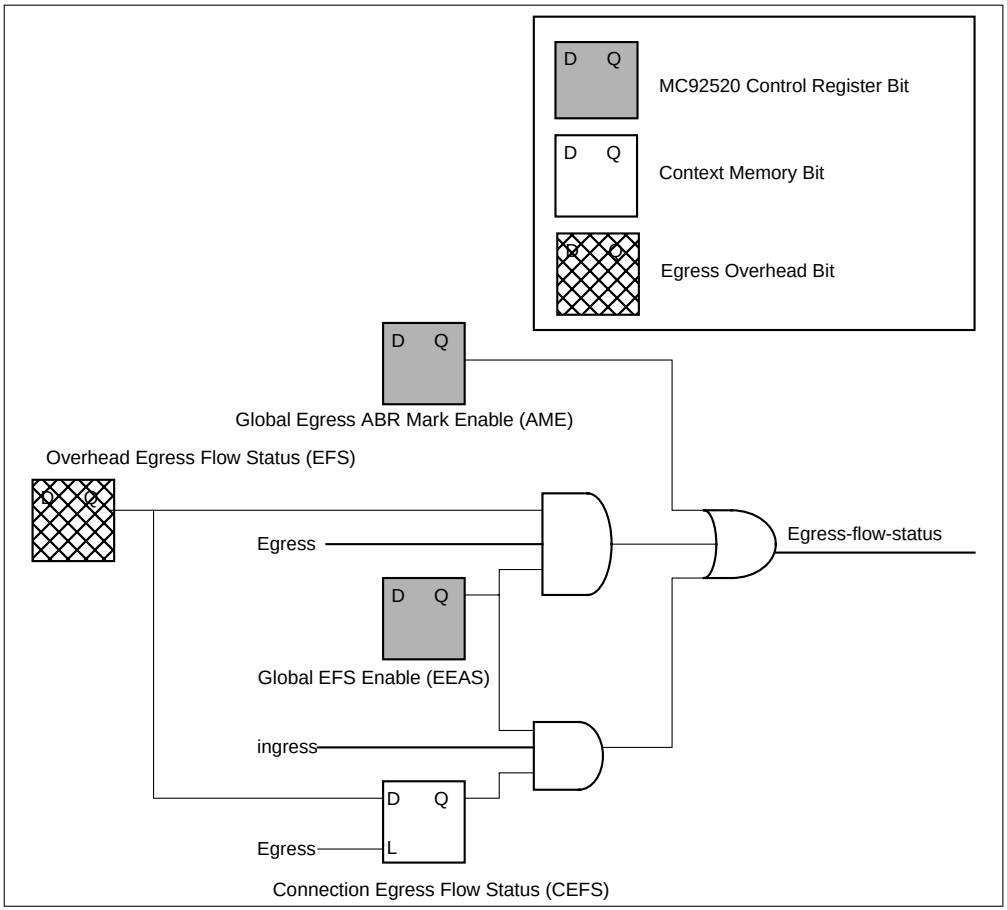


Figure 6-15. Egress-Flow-Status Logic

6.4.2.3 MC92520 Actions in the Ingress Direction

Figure 6-16 illustrates the actions taken by the MC92520 in the ingress direction. The MC92520 can mark cells as a result of either ingress-flow-status or egress-flow-status.

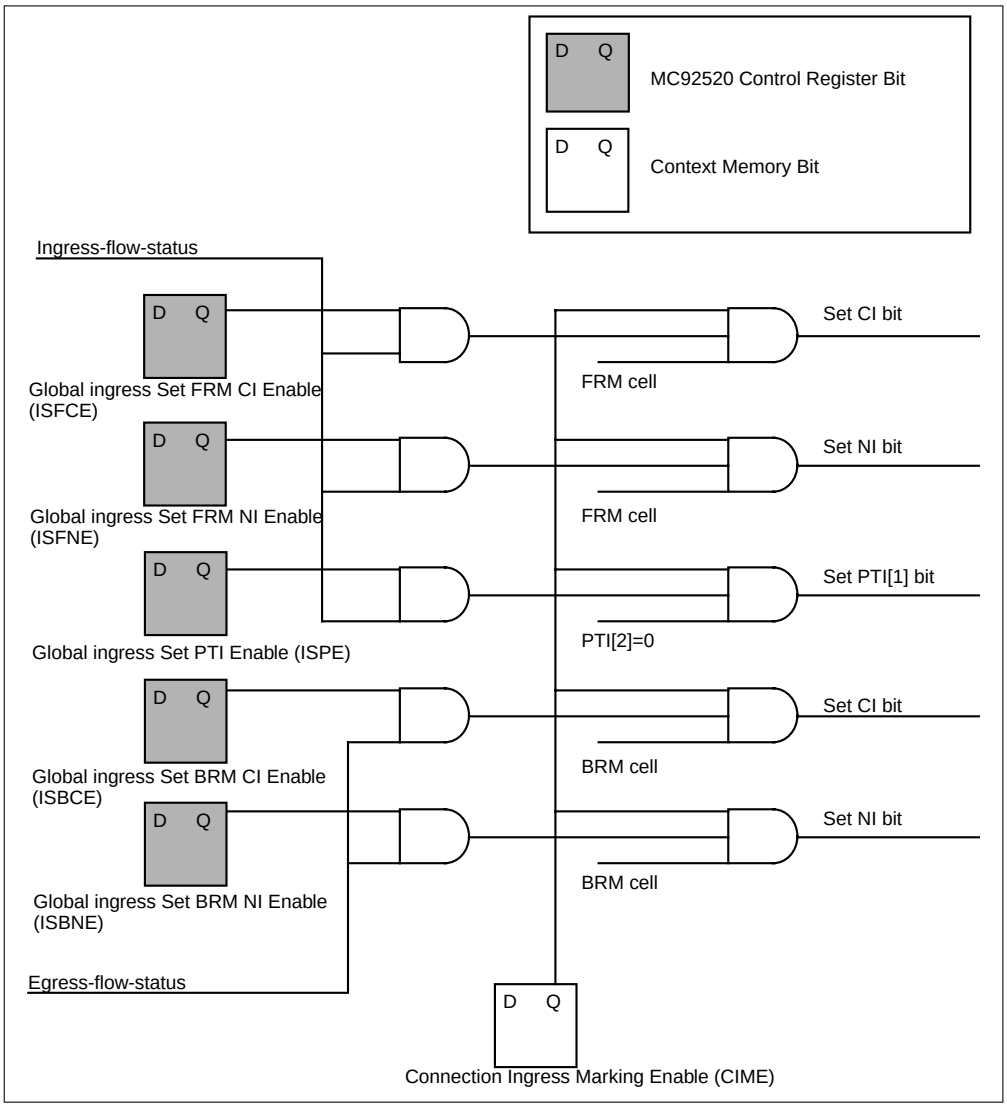


Figure 6-16. Ingress Direction Actions

In the case where the ingress-flow-status is asserted, the MC92520 can perform one or more of the following:

- Set the CI bit in an ingress FRM cell when the global ingress set FRM CI enable (ISFCE) bit in the ingress processing configuration register (IPCR) is set.
- Set the NI bit in an ingress FRM cell when the global ingress set FRM NI enable (ISFNE) bit in the ingress processing configuration register (IPCR) is set.
- Set the PTI[1] bit in an ingress cell whose PTI[2] = 0 when the global ingress set PTI enable (ISPE) bit in the ingress processing configuration register (IPCR) is set.

In the case where the egress-flow-status is asserted, the MC92520 can perform one or more the following:

- Set the CI bit in an ingress BRM cell when the global ingress set BRM CI enable (ISBCE) bit in the ingress processing configuration register (IPCR) is set.
- Set the NI bit in an ingress BRM cell when the global ingress set BRM NI enable (ISBNE) bit in the ingress processing configuration register (IPCR) is set.

All cell marking on the ingress is enabled on a per-connection basis by the connection ingress marking enable (CIME) bit in the common parameters extension word**.**

6.4.2.4 MC92520 Actions in the Egress Direction

Figure 6-17 illustrates the actions taken by the MC92520 in the egress direction. The MC92520 can mark cells as a result of either ingress-flow-status or egress-flow-status. In the case where egress-flow-status is asserted, the MC92520 can perform one or more of the following:

- Set CI bit in an egress FRM cell when the global egress set FRM CI enable (ESFCE) bit in the egress processing configuration register (EPCR) is set.
- Set NI bit in an egress FRM cell when the global egress set FRM NI enable (ESFNE) bit in the egress processing configuration register (EPCR) is set.
- Set PTI[1] bit in an egress cell whose PTI[2] = 0 when the global egress set PTI enable (ESPE) bit in the egress processing configuration register (EPCR) is set.

In the case where ingress-flow-status is asserted the MC92520 can perform one or more the following:

- Set CI bit in an egress BRM cell when the global egress set BRM CI enable (ESBCE) bit in the egress processing configuration register (EPCR) is set.
- Set NI bit in an egress BRM cell when the global egress set BRM NI enable (ESBNE) bit in the egress processing configuration register (EPCR) is set.

All cell marking on the egress is enabled on a per-connection basis by the connection egress marking enable (CEME) bit in the common parameters extension word.

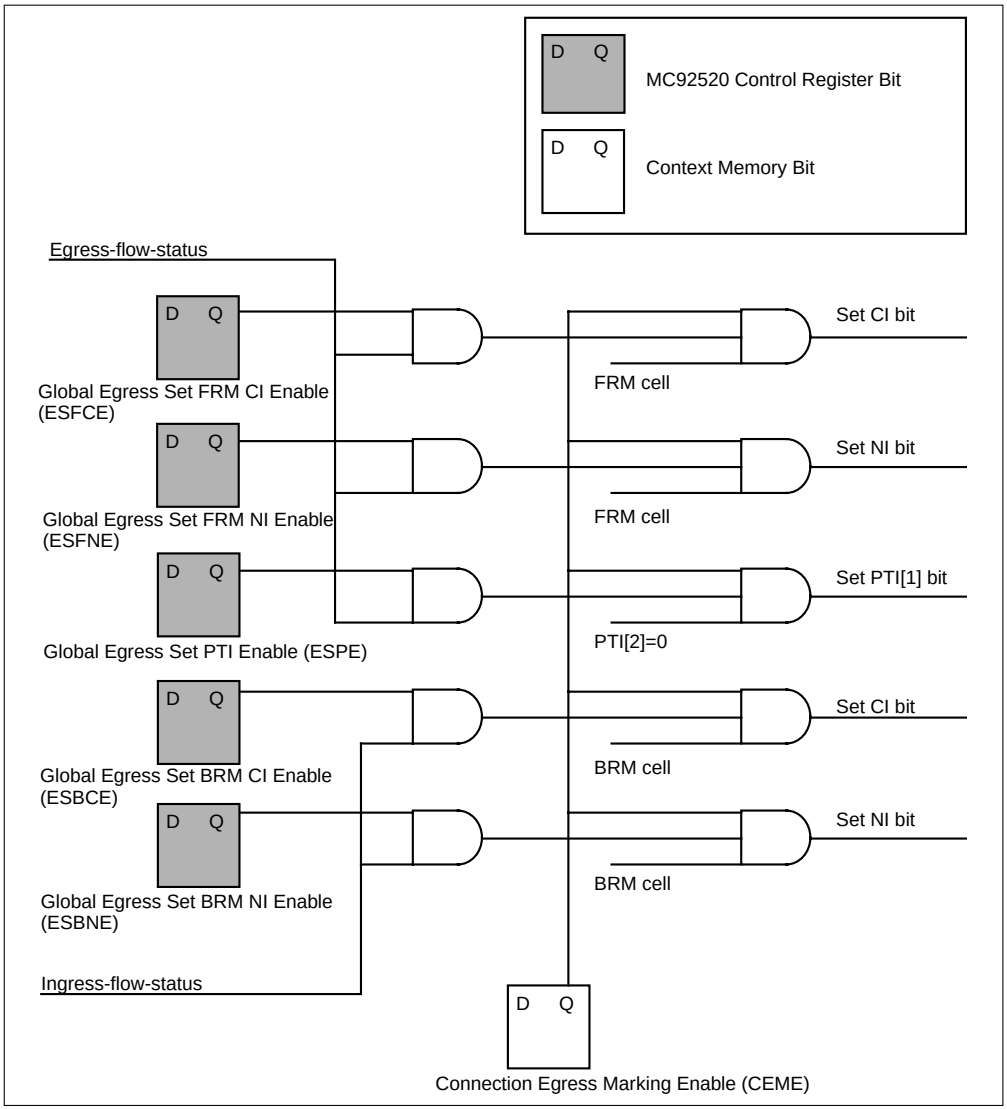


Figure 6-17. Egress Direction Actions

6.4.2.5 Cell Marking Examples

The following figures illustrate MC92520 cell marking examples: Figure 6-18 shows the marking of the CI bit of ingress FRM cells.

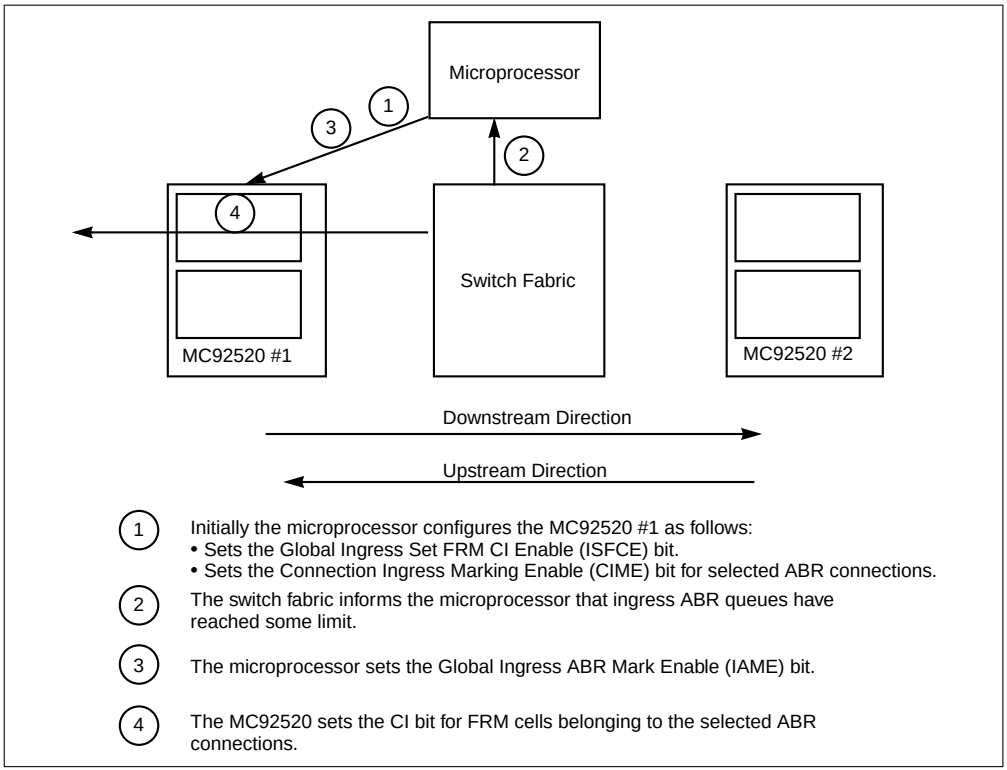


Figure 6-18. Marking CI Bits of Ingress FRM Cells

Figure 6-19 shows the marking of the NI field.

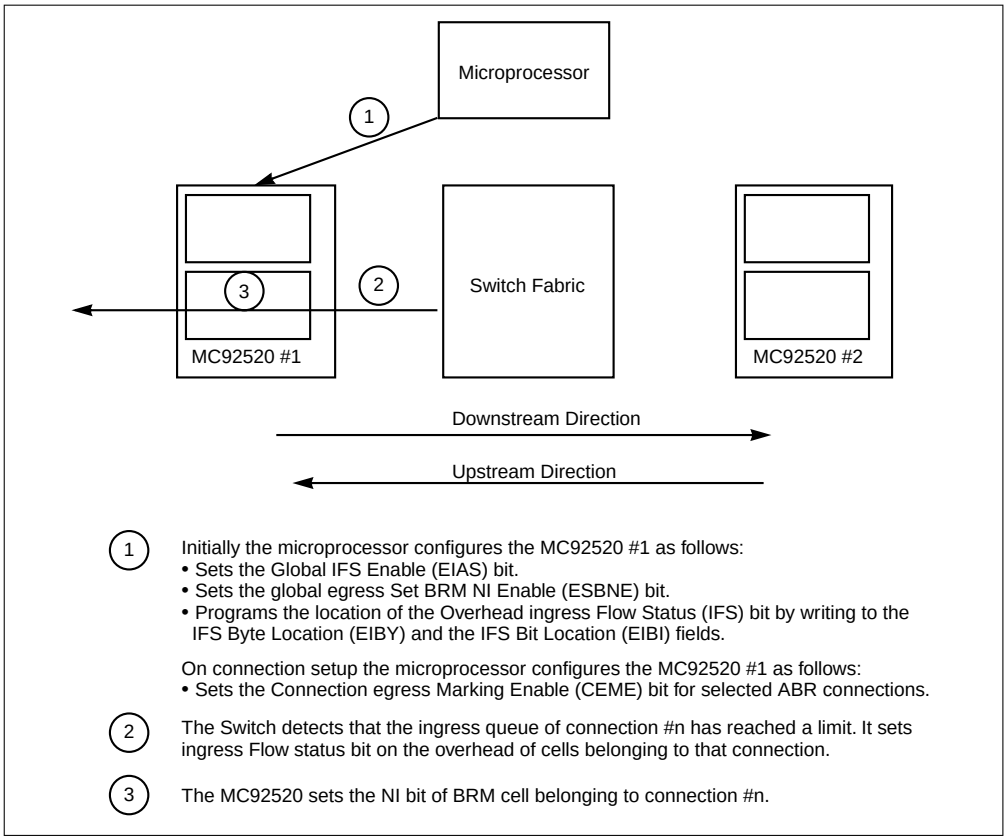
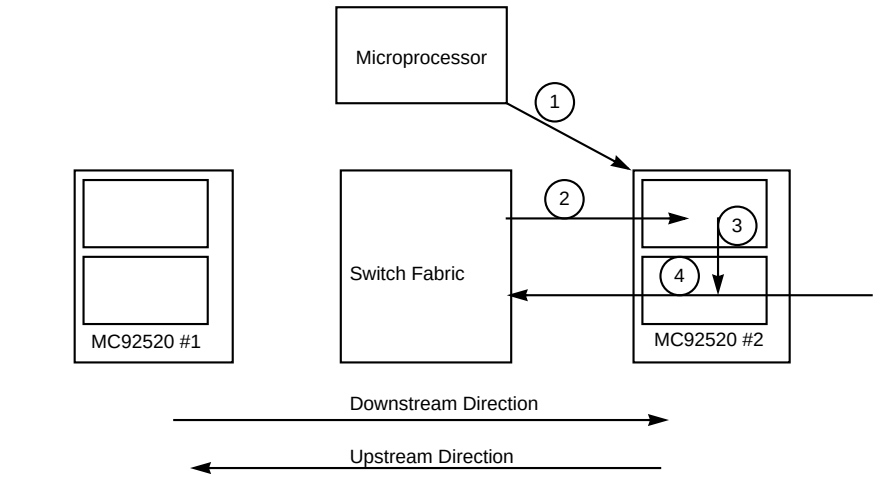


Figure 6-19. Marking a BRM Cell NI Field

Figure 6-20 shows the marking of the CI bit for all ingress BRM cells.



1 Initially the microprocessor configures the MC92520 #2 as follows:
- Sets the global EFS Enable (EEAS) bit.
- Sets the global ingress Set BRM CI Enable (ISBCE) bit.
- Programs the location of the Overhead egress Flow Status (EFS) bit by writing to the EFCI Byte Location (EFBY) and the EFCI Bit Location (EFBI) fields.

On connection setup the microprocessor configures the MC92520 #2 as follows:
- Sets the Connection egress Marking Enable (CEME) bit for selected ABR connections.

2 The Switch detects that the egress queue of connection #n has reached a limit. It sets egress Flow status bit on the overhead of cells belonging to that connection.

3 The MC92520 copies the overhead egress flow status bit into the Connection egress Flow Status (CEFS) bit and effectively sets it.

4 The MC92520 sets the CI bit of BRM cells belonging to connection #n.

Figure 6-20. Marking the CI Bit for All Ingress BRM Cells for a Connection

6.4.3 Ingress Switch Parameters Hooks

The MC92520 defines an 8-bit field that can be overlayed on bits of the ingress switch parameters belonging to RM cells. In applications where the overlayed switch parameter field is a priority field used by the switch fabric, RM cells can gain higher priority in passing the switch. This enables shortening the feedback loop for ABR. The MC92520 performs this field overlay if one of the following occurs:

- An ingress BRM cell is received and both the ingress BRM overlay enable (IBOE) and the ingress RM overlay enable (IROE) bits are set.
- An ingress FRM cell is received and both the ingress FRM overlay enable (IFOE) bit and the ingress RM overlay enable (IROE) bit are set.

Once the MC92520 is enabled, it uses the RM overlay location (ROL) field to locate one byte out of 12 bytes in the ingress switch parameters. This byte is overlaid by the RM overlay field (ROF) only on bits that are enabled by the RM overlay mask (ROM) field. See Section 7.2.6.34, “RM Overlay Register (RMOR)” for a description of the RM overlay register fields.

For example, if the ROF = 11001101, the ROM = 01111000, and the ROL = 9, then the IBOE bit is set and the current cell is a backward RM cell, as shown in Figure 6-21.

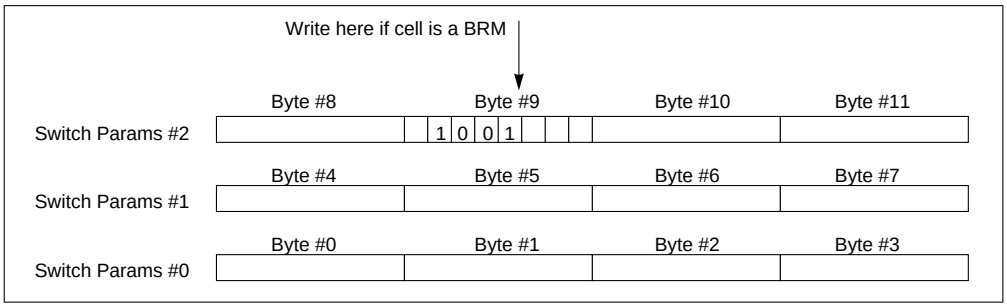


Figure 6-21. Ingress Switch Parameter Example

6.4.4 Egress Reset EFCI

The MC92520 resets PTI[1] on a cell that meets the following conditions:

- PTI[2] = 0.
- The cell is a non-RM cell.
- The egress reset EFCI (EREF) bit in the context parameters extension table for the connection to which the cell belongs is set.

EFCI is a feature that can be used in ABR destination behavior.

6.5 Operations and Maintenance (OAM) Support

The MC92520 supports OAM protocol functions for virtual path (F4) and virtual channel (F5) cell flows. Both segment and end-to-end OAM cells are recognized and can be individually processed on a virtual connection (VCC) or virtual path (VPC) basis. In practice, most OAM protocol functions require cooperative processing and communication between the MC92520 and a control processor. The following subsections describe what functions the MC92520 performs and how these functions relate to the OAM processing task of a control processor:

- Section 6.5.1, “General Concepts and Definitions”
- Section 6.5.2, “Internal Scan”
- Section 6.5.3, “Generic OAM Support”
- Section 6.5.4, “Fault Management”
- Section 6.5.5, “Performance Management”
- Section 6.5.6, “Performance Monitoring”
- Section 6.5.7, “Activation/Deactivation OAM Cells”

NOTE:

The operation of the OAM functions at VP/VC boundaries is discussed in Appendix D.

6.5.1 General Concepts and Definitions

This subsection provides a short summary of OAM definitions and how they relate to MC92520 OAM processing concepts. For a full OAM specification, see ITU-T I-Series recommendation I.610 and related publications.

In the following MC92520 references to cell movement, “copying a cell” indicates that the cell remains in the cell flow and a copy is transferred to the microprocessor extraction queue, while “removing a cell” means that it is removed from the cell flow and optionally transferred to the microprocessor extraction queue.

6.5.1.1 OAM Cell Header Definition

Table 6-4 and Table 6-5 illustrate the standard OAM cell header definitions for the UNI and NNI, respectively.

Table 6-4. Preassigned Header Values at the NNI

Use	VPI	VCI	PTI	CLP
Segment OAM F4 flow cell	AAAA AAAA AAAA	0000 0000 0000 0011	0B0	A
End-to-end OAM F4 flow cell	AAAA AAAA AAAA	0000 0000 0000 0100	0B0	A
Segment OAM F5 flow cell	AAAA AAAA AAAA	AAAA AAAA AAAA AAAA	100	A
End-to-end OAM F5 flow cell	AAAA AAAA AAAA	AAAA AAAA AAAA AAAA	101	A

A = available for use by the appropriate ATM layer function

B = transmitter shall set bit to 0, receiver shall ignore bit value

Table 6-5. Preassigned Header Values at the UNI

Use	GFC	VPI	VCI	PTI	CLP
Segment OAM F4 flow cell	XXXX	AAAA AAAA	0000 0000 0000 0011	0B0	A
End-to-end OAM F4 flow cell	XXXX	AAAA AAAA	0000 0000 0000 0100	0B0	A
Segment OAM F5 flow cell	XXXX	AAAA AAAA	AAAA AAAA AAAA AAAA	100	A
End-to-end OAM F5 flow cell	XXXX	AAAA AAAA	AAAA AAAA AAAA AAAA	101	A

A = available for use by the appropriate ATM layer function
B = transmitter shall set bit to 0, receiver shall ignore bit value
X = don't care bit

6.5.1.2 Virtual Path (F4) Flow Mechanism

The F4 flow is recognized by means of preassigned virtual channel identifiers (VCI = 3 and VCI = 4) within the virtual path. There are two kinds of F4 flows that can exist simultaneously:

- Segment (identified by preassigned VCI 3)—This flow is used for communicating operations information restricted to one VPC link or multiple interconnected VPC links. The concatenation of VPC links is called a VPC segment. Cells inserted into this flow may only be removed by the segment end points. Segment end points must remove these cells to prevent confusion in adjacent segments.
- End-to-end (identified by preassigned VCI 4)—This flow is used for end-to-end VPC operations communications. Cells inserted into this flow can be removed only by the end points of the virtual path.

Cells can be inserted into the flow at any connecting point. However, cells may be terminated only at the flow-specific F4 end points. The MC92520 recognizes F4 OAM cells on any connection where the egress virtual path connection (EVPC) or ingress virtual path connection (IVPC) bit is set, and the header is as shown in Table 6-4 and Table 6-5. Cells with F4 OAM headers received on a connection with EVPC or IVPC = 0 are treated as described in Section 6.5.3.1, "Illegal OAM Cells."

6.5.1.3 Virtual Channel (F5) Flow Mechanism

The F5 flow is recognized by means of preassigned payload type identifiers (PTI = 4 and PTI = 5). There are two kinds of F5 flows that can exist simultaneously:

- Segment (identified by PTI 4)—This flow is used for communicating operations information restricted to one VCC link or multiple interconnected VCC links. The concatenation of VCC links is called a VCC segment. Segment end points must remove these cells to prevent confusion in adjacent segments.
- End-to-end (identified by PTI 5)—This flow is used for end-to-end VCC operations communications. Cells inserted into this flow can only be removed by the end points of the virtual channel.

Cells can be inserted into the flow at any connecting point; however, cells may be terminated only at the F5 end points. The MC92520 recognizes F5 OAM cells on any connection where the EVPC or IVPC bit is reset and the header is as shown in Table 6-4 and Table 6-5.

6.5.1.4 OAM Cell and Function Types

End-to-end and segment flows may use all OAM functions unless otherwise noted. The following descriptions do not differentiate between end-to-end and segment flows. Table 6-6 defines the values of OAM cell type and function type recognized by the MC92520.

Table 6-6. OAM Types Explicitly Identified by the MC92520

OAM Cell Type	Code	Function Type	Code
Fault management	0001	AIS	0000
		RDI	0001
		Continuity check	0100
		Loopback	1000
Performance management	0010	Forward monitoring	0000
		Backward reporting	0001

The location of these fields in the OAM cell is shown in Figure 6-22.

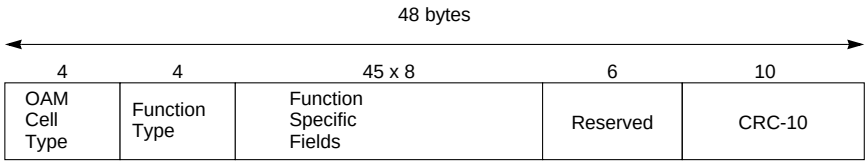


Figure 6-22. OAM Cell Payload Structure

Any unused function-specific fields in received OAM cells are ignored by the MC92520. When internally generating OAM cells, the MC92520 fills the unused field with the standard default value of 0x6A and fills the 6-bit reserved field with zeroes. When inserting OAM cells with the payload provided by the microprocessor, the MC92520 does not modify the unused and reserved fields.

6.5.2 Internal Scan

The MC92520 uses the internal scan process to generate alarm indication signal (AIS), remote defect indicator (RDI), and continuity check (CC) cells and insert them into the cell flows. When the scan process is activated by the microprocessor writing to the start SCAN register (SSR), the MC92520 scans the context parameters table records (see Section 7.2.3.5, “Egress Parameters” and Section 7.2.3.6, “Ingress Parameters”) starting from the record indicated by the context memory highest value (CMHV) field of the internal scan control register (ISCR) down to the first record (CI = 0) to see if any of the

send AIS, send RDI, or send CC bits are set, in which case the appropriate cell is generated and queued for insertion in the ingress or egress insertion queue.

When the scan is completed and the generated cells areinserted into the cell flow, the scan process done (IR[SPD]) bit is set. The types of cells included in the scan are defined by the internal scan control register (ISCR) (see **Section 7.1.5.12, "Internal Scan Control Register (ISCR)**"). The scan should be activated normally about once per second, in accordance with OAM standards.

6.5.3 Generic OAM Support

At the VC switch (VCX) at which a VPC is terminated, the VCCs are accessible. Context table records are maintained for each of the VCCs (VPC bit is 0) as well as the reserved VCI values used for the OAM F4 flows (VPC bit is 1) (see Appendix D for details). In this way, all OAM features that the MC92520 supports on all connections (fault management, activation/deactivation) are supported on each VCC individually as well as on the VPC collectively. Table 6-7 describes the general OAM bits that define the scope of OAM processing at this node.

Table 6-7. General OAM Bits

Logical Name	Bit Name	Context Parameters Table Word	Stat/ Dyn	Explanation
VPC	EVPC IVPC	Egress and ingress parameters	S	Virtual path connection. See Table 7-64 and Table 7-65.
Segment OAM termination	ESOT ISOT	Egress and ingress parameters	S	This node is the terminating point of the VCC/VPC segment. All arriving segment OAM cells should be removed from the cell flow after the OAM processing. Also, segment loopback OAM cells with the loopback location ID = "end point" are looped back here. See Table 7-64 and Table 7-65.
Segment OAM origin	ESOO ISOO	Egress and ingress parameters	S	This node is the originating point of the VCC/VPC segment. See Table 7-64 and Table 7-65.
End-to-end OAM termination	EEOT IEOT	Egress and ingress parameters	S	This node is the terminating point of the VCC/VPC OAM flow. All arriving OAM cells should be removed from the cell flow after the OAM processing. Also, end-to-end loopback OAM cells with the loopback location ID = "end point" are looped back here. See Table 7-64 and Table 7-65.
Copy other OAM cells	ECOT ICOT	Egress and ingress parameters	S	Copy OAM cells whose OAM type is not recognized by the MC92520. See Table 7-64 and Table 7-65.
Copy all OAM cells	ECAO ICAO	Egress and ingress parameters	S	Copy all received OAM cells. This bit may be set for monitoring all OAM traffic on a connection. See Table 7-64 and Table 7-65.

Figure 6-23 shows endpoint visibility.

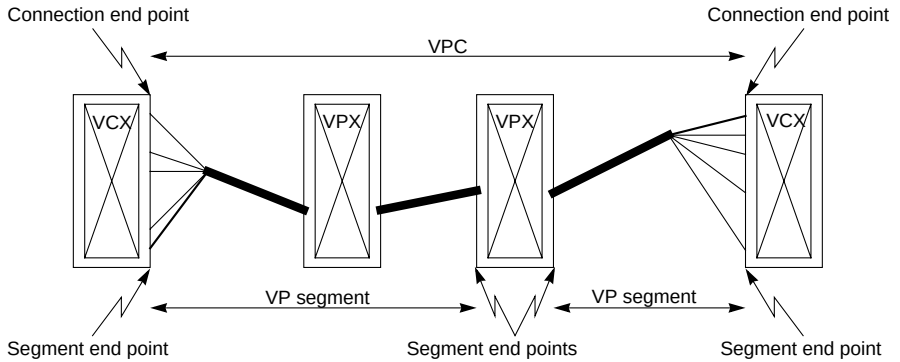


Figure 6-23. Visibility of VCCs at the End Points of VPCs

Figure 6-24 illustrates the use of the OAM origin and termination bits at the end points of the OAM flows.

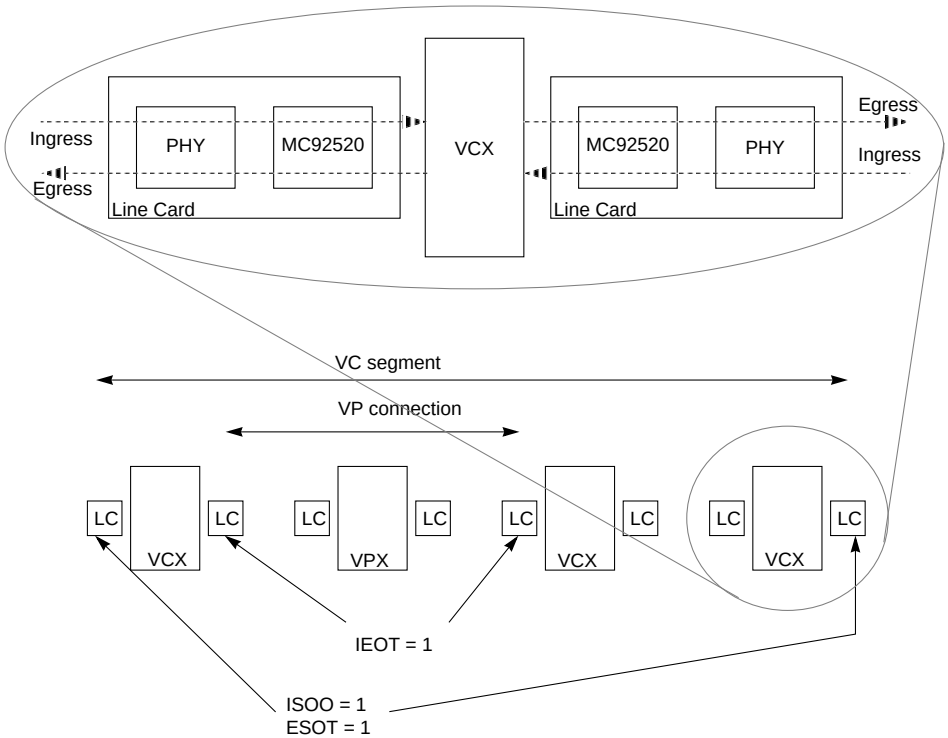


Figure 6-24. Example of VP Connection / VC Segment

6.5.3.1 Illegal OAM Cells

OAM cells that are illegal or outside of their legal scope are copied to the cell extraction queue. This includes a segment OAM cell received at a segment OAM origin point (ESOO/ISOO = 1) that is not a segment OAM termination (ESOT/ISOT = 0).

6.5.3.2 Other OAM Cells

OAM cells with an OAM cell type or OAM function type not explicitly handled by the MC92520 (any combination not appearing in Table 6-6) are copied to the cell extraction queue if egress copy received other OAM cells/ingress copy received other OAM cells (ECOT/ICOT) is set in the context parameters table.

NOTE:

Activation/deactivation OAM cells are considered “other OAM cells”.

6.5.4 Fault Management

The MC92520 handles the OAM feature of fault management by means of the following:

- Alarm surveillance through generation of the following cells: alarm indication signal (AIS) cells, remote defect indicator (RDI) cells, and continuity check (CC) cells. See Section 6.5.4.1, “Alarm Surveillance.”
- Failure localization and testing (through use of VPC/VCC loopback cells). See Section 6.5.4.2, “Failure Localization and Testing.”

6.5.4.1 Alarm Surveillance

As traffic is surveyed, faults are indicated by generating fault management cells. The function specific fields of the AIS/RDI fault management cell as shown in Figure 6-25 are:

- Failure type—No values are currently standardized. The default value is 0x6A. This field is ignored by the MC92520 when processing received AIS/RDI cells.
- Failure location—No values are currently standardized. The default value is 0x6A in each octet. This field is ignored by the MC92520 when processing received AIS/RDI cells.

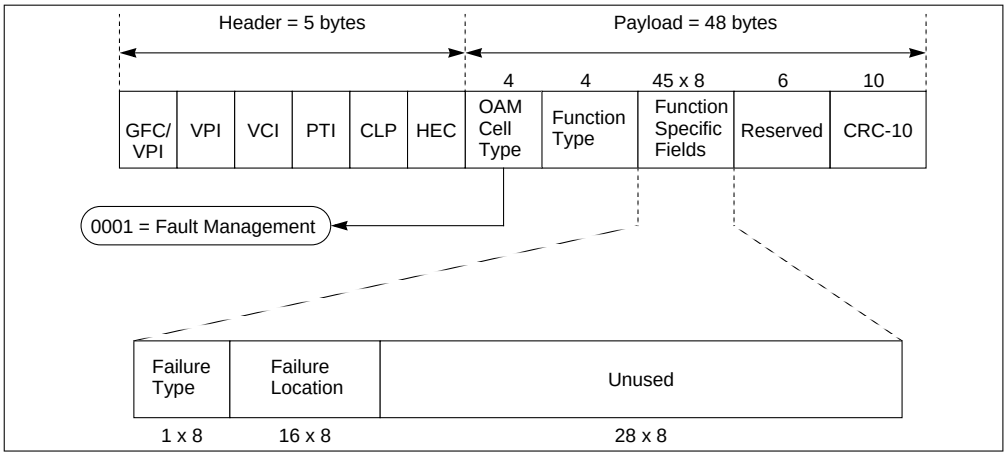


Figure 6-25. AIS/RDI Fault Management Cell

6.5.4.1.1 Alarm Indication Signal Cells

The MC92520 provides full support for generating alarm indication signal (AIS) cells and for reporting that an AIS cell has been received. To initiate the generation of AIS cells in the egress cell flow, the microprocessor sets the egress send AIS (ESAI) cell bit for a specific connection. This bit causes the MC92520 to generate an AIS cell for this connection on each pass of the internal scan if the egress enable AIS (ISCR[EEAI]) bit is set. To initiate the generation of AIS cells in the ingress cell flow, the microprocessor sets the ingress send AIS (ISAI) cell bit for a specific connection. This bit causes the MC92520 to generate an AIS cell for this connection on each pass of the internal scan if the ingress enable AIS (ISCR[IEAI]) bit is set. See Section 6.5.2, “Internal Scan” for more details. In order to meet the Bellcore requirement to insert the first AIS cell in less than 500 ms, the first AIS cell may be inserted directly by the microprocessor using the cell insertion registers, while the succeeding cells are generated by the internal scan.

NOTE:

If the ERET/IRET bit (see Table 6-10) is set by the MC92520 while AIS cells are being generated, a valid cell has been received, and the ESAI/ISAI bit can most likely be reset.

AIS cells that are generated by the internal scan are always end-to-end. The failure type and failure location fields are coded with their default value of 0x6A in each octet since no values are currently standardized. The remaining 28 octets of the function-specific fields are also coded with 0x6A since they are unused in AIS cells. AIS cells that have been generated as a result of the ESAI bit are inserted in the egress cell flow, and those that have been generated as a result of the ISAI bit are inserted in the ingress cell flow. When an end-to-end AIS cell is received on a connection, the appropriate (ingress or egress) receive AIS bit of that connection is set.

Note that segment AIS cells are treated as OAM cells, but not as AIS cells. Therefore, they do not cause the receive AIS bits to be set, and they may be copied by using the ECAO/ICAO bit, but not by using the ECAS/ICAS bit. All connections should be scanned by the microprocessor for the receive AIS bits at regular intervals. The default scan rate should be once per second to allow the AIS state to be declared within one second of the AIS cell having been received.

The egress copy received AIS cells (ECAS) bit and the ingress copy received AIS cells (ICAS) bit may be set at any point along the connection to indicate that all AIS cells received on the connection should be copied to the microprocessor. If the failure type and failure location fields are set to their default values, the AIS cells do not contain useful information so the ECAS and ICAS bits would not normally be set. AIS cells are removed from the cell flow at the connection end point (see Table 6-7).

Table 6-8 describes the external memory bits that are used for AIS processing.

Table 6-8. AIS Bits

Logical Name	Bit Name	Table	Stat/ Dyn	Used by	Explanation
Send AIS	ESAI ISAI	Context parameters	S	Internal scan	Insert an AIS cell in the egress/ingress direction
Receive AIS on egress	ERAS	Flags	D	Egress	AIS cell has been received in the egress cell flow.
Receive AIS on ingress	IRAS	Flags	D	Ingress	AIS cell has been received in the ingress cell flow.
Copy received AIS cells	ECAS ICAS	Context parameters	S	Egress and ingress	Copy AIS cells to the cell extraction queue
Enable AIS	EEAI IEAI	Context parameters	S	Egress and ingress	Enables AIS cells to be generated on internal scans

6.5.4.1.2 Remote Defect Indicator Cells

The MC92520 provides full support for generating remote defect indicator (RDI) cells and for reporting that an RDI cell has been received. To initiate the generation of RDI cells in the egress cell flow, the microprocessor sets the egress send RDI (ESRD) cell bit for a specific connection. This bit causes an RDI cell to be generated on this connection on each pass of the internal scan if the egress enable RDI ([EERD]) bit is set in the internal scan control register (ISCR). To initiate the generation of RDI cells in the ingress cell flow, the microprocessor sets the ingress send RDI (ISRD) cell bit for a specific connection. This bit causes an RDI cell to be generated on this connection on each pass of the internal scan if the ingress enable RDI (ISCR[IERD]) bit is set. See Section 6.5.2, “Internal Scan” for more details.

The first RDI cell may be inserted directly to reduce the initial delay, while the succeeding cells are generated by the internal scan. Note that if the receive end-to-end traffic bits (see Table 6-10) are set while RDI cells are being generated, a valid cell has been received, and

the ESRD/ISRD bit can most likely be reset. RDI cells that are generated by the internal scan are always end-to-end. The failure type and failure location fields are coded with their default value of 0x6A in each octet since no values are currently standardized. The remaining 28 octets of the function-specific fields are also coded with 0x6A since they are unused in RDI cells. RDI cells that have been generated as a result of the ESRD bit are inserted in the egress cell flow, and those that have been generated as a result of the ISRD bit are inserted in the ingress cell flow.

When an end-to-end RDI cell is received on a connection, the appropriate (ingress or egress) receive RDI bit of that connection is set. Note that segment RDI cells are treated as OAM cells, but not as RDI cells. Therefore, they do not cause the receive RDI bits to be set, and they may be copied by using the ECAO/ICAO bit, but not by using the ECRD/ICRD bit.

All connections should be scanned by the microprocessor for the receive RDI bit at regular intervals. The default scan rate should be once per second. That allows the RDI state to be declared within one second of the RDI cell being received. The egress copy received RDI cells (ECRD) bit or the ingress copy received RDI cells (ICRD) bit may be set at any point along the connection to indicate that all RDI cells received on the connection should be copied to the cell extraction queue. If the failure type and failure location fields are set to their default values, the RDI cells do not contain useful information so the ECRD and ICRD bits would not normally be set. RDI cells are removed from the cell flow at the connection end point (see Table 6-7). Table 6-9 describes the external memory bits that are used for RDI processing.

Table 6-9. RDI Bits

Logical Name	Bit Name	Table	Stat/ Dyn	Used by	Explanation
Send RDI	ESRD ISRD	Context parameters	S	Internal scan	Insert an RDI cell in the egress/ingress direction
Receive RDI on egress	ERRD	Flags	D	Egress	RDI cell has been received in the egress cell flow.
Receive RDI on ingress	IRRD	Flags	D	Ingress	RDI cell has been received in the ingress cell flow.
Copy received RDI cells	ECRD ICRD	Context parameters	S	Egress and ingress	Copy RDI cells to the cell extraction queue
Enable RDI	EERD IERD	Context parameters	S	Ingress and egress	Enables RDI cells to be generated on internal scans

Figure 6-26 shows the F4 flows for the case where the VP connection end points are logically at the edges of the VCX switches at which a set of VCCs are collected to be a VPC.

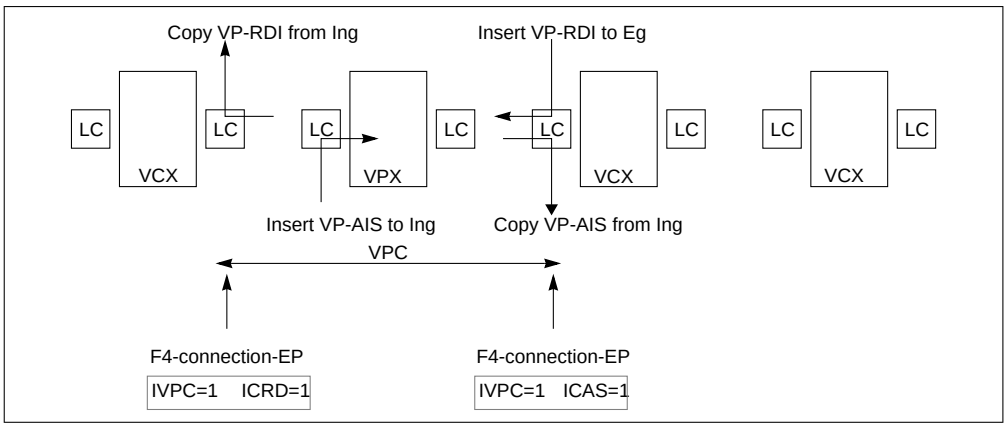


Figure 6-26. F4 AIS/RDI Flows for a VPC Internal to the Network

Figure 6-27 shows the F4 flows for the case where the VP connection is routed to the user end point through a VPX. In this case the connection end point must be on the line card closest to the user.

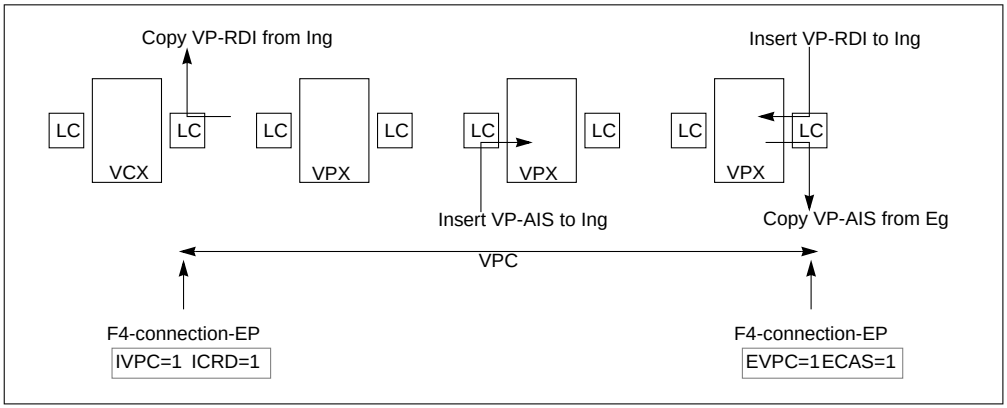


Figure 6-27. F4 AIS/RDI Flows for a VPC that Crosses the UNI

Figure 6-28 shows the F5 flows for a virtual channel connection where one end point is on the inside of the switch and the other end point is on the outside of the switch.

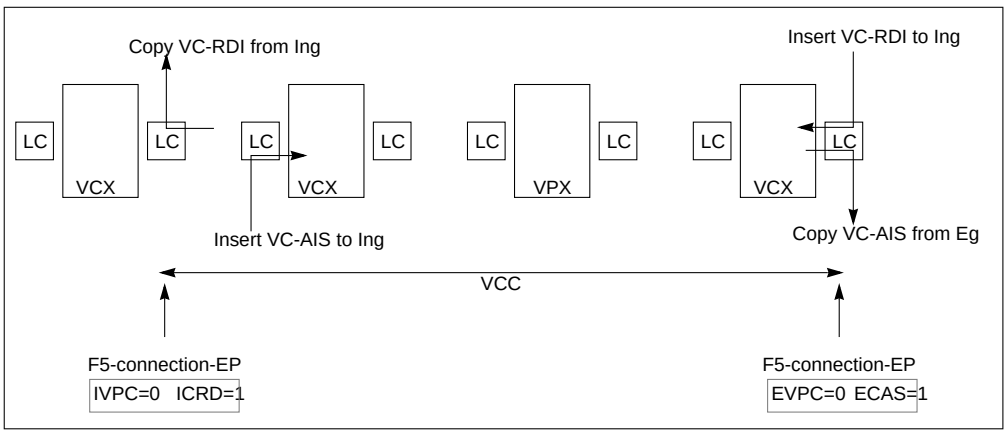


Figure 6-28. F5 AIS/RDI Flows

These examples illustrate where to set the copy AIS and copy RDI bits for the given flows. As previously stated, setting these bits to copy the cells is normally not necessary, and the flag table may be used to determine that an AIS or RDI cell has been received.

6.5.4.1.3 Continuity Check Cells

The MC92520 provides full support for generating continuity check (CC) cells. The MC92520 also provides support for reporting about the connection traffic in order to enable the microprocessor to determine when connections have lost continuity. Both end-to-end and segment continuity checks can be run simultaneously on the same connection. To initiate the generation of CC cells, the microprocessor sets any of the send CC bits (ESCS, ISCS, ESCE, ISCE) for a specific connection. Each bit causes a CC cell to be generated on this connection during a pass of the internal scan if the corresponding bit of the internal scan control register (ISCR) is set. See Section 6.5.2, “Internal Scan” for more details.

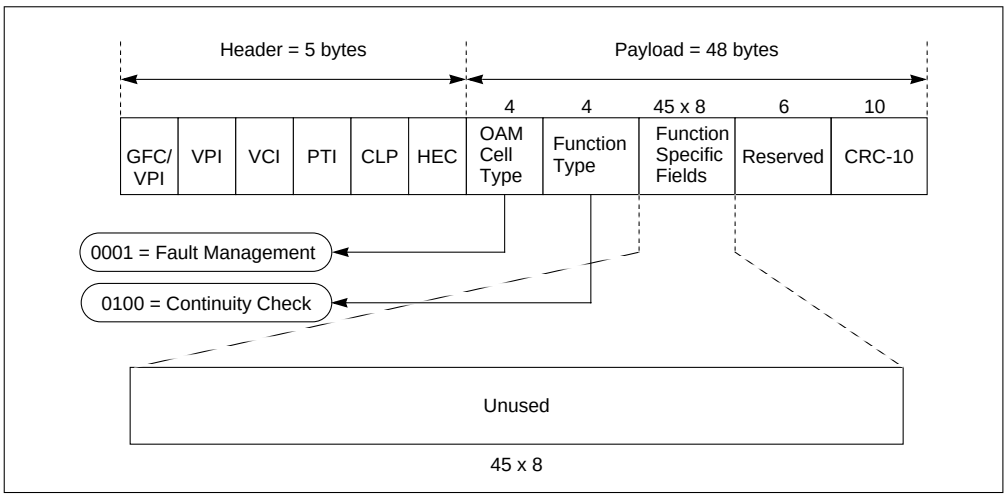


Figure 6-29. Continuity Check Fault Management Cell

The receive traffic bits are used to record if a valid cell (user information cell or CC cell) has been processed on the connection. On both the ingress and egress sides there are two bits to identify cases in which there is continuity on the segment but continuity from the end point has been lost. When a user cell or end-to-end CC cell is received, both receive traffic bits are set. When a segment CC cell is received, only the receive segment traffic bit is set.

NOTE:

At the termination of a VPC, the user traffic is recorded by setting the receive traffic bits of the individual VCCs since the VCCs are visible. In order to check for continuity on the VPC, the microprocessor may logically OR the receive traffic bits of all the VCCs belonging to the VPC.

CC cells are removed from the egress or ingress flow at the connection-segment end point (see Table 6-7). Table 6-10 describes the external memory bits that are used for continuity check processing.

Table 6-10. Continuity Check Bits

Logical Name	Bit Name	Table	Stat/ Dyn	Used by	Explanation
Send CC segment	ESCS ISCS	Context parameters	S	Internal Scan	Insert a segment CC cell in the egress/ingress direction
Send CC end-to-end	ESCE ISCE	Context parameters	S	Internal Scan	Insert an end-to-end CC cell in the egress/ingress direction
Receive segment traffic on egress	ERST	Flags	D	Egress	User data cell or CC cell received in the egress cell flow
Receive end-to-end traffic on egress	ERET	Flags	D	Egress	User data cell or end-to-end CC cell received in the egress cell flow
Receive segment traffic on ingress	IRST	Flags	D	Ingress	User data cell or CC cell received in the ingress cell flow
Receive end-to-end traffic on ingress	IRET	Flags	D	Ingress	User data cell or end-to-end CC cell received in the ingress cell flow

6.5.4.2 Failure Localization and Testing

Another fault management utility provided by the MC92520 is failure localization and testing by using VPC and VCC loopback cells to identify cells that must be removed from the cell flow. The loopback cell format and available processing options are discussed in the following sections.

6.5.4.2.1 Loopback Cell Format

The format of OAM loopback cells is given in Figure 6-30 below. The function-specific fields of a loopback cell are defined as follows:

- Loopback indication—This field provides a boolean indication as to whether or not the cell has already been looped back. The seven most significant bits are always coded as 0. The least significant bit is 1 before the cell is looped back and 0 after the cell has been looped back.
- Correlation tag—Multiple loopback cells may have be inserted in the stream. The correlation tag provides a means of correlating transmitted loopback cells with received cells.
- Loopback location ID—This field identifies the point along the connection where the loopback is to occur. The default value is all ones, indicating the end point of the connection or segment.
- Source ID—This field identifies the originator of the loopback cell. Default value is all ones.

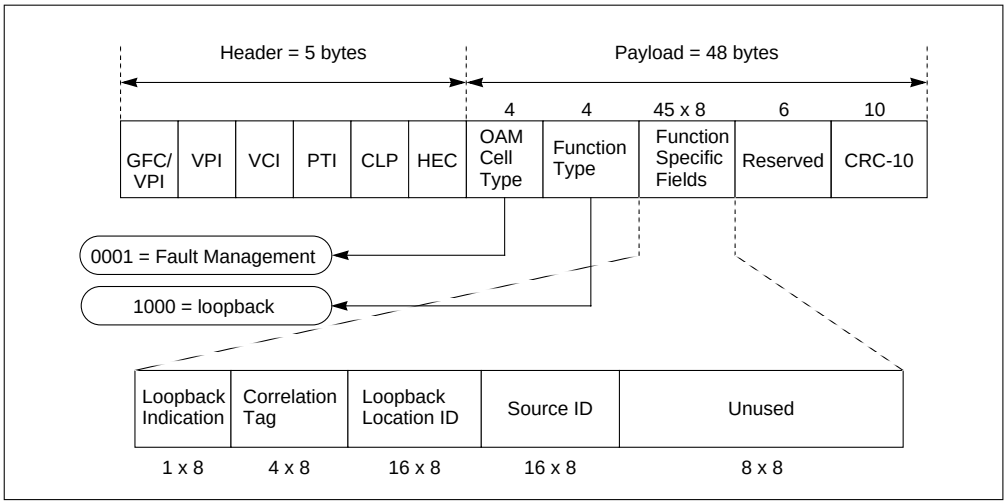


Figure 6-30. OAM Loopback Cell

6.5.4.2.2 Loopback Cell Processing

The MC92520 provides support for looping back and copying loopback OAM cells. Loopback OAM cells are prepared by the microprocessor and inserted through the MC92520 to the specified cell flow. The MC92520 checks the loopback location ID of received loopback OAM cells for a match with the MC92520's programmable node ID (see Section 7.1.6.25–Section 7.1.6.28). It also checks for the end point loopback location ID (all 1s) on both the ingress and egress sides and declares a match if the corresponding OAM termination bit is set for the connection. If the loopback location ID matches and the LSB of the loopback indication is 1, the cell is removed from the cell flow. The cell is copied to the cell extraction queue so the microprocessor can simply decrement (clear) the loopback indication and insert the cell in the opposite direction.

NOTE:

An intra-switch loopback can be performed by inserting a loopback cell in the ingress with a loopback location ID that provides a match at the egress side of the switch.

The MC92520 checks the source ID of received loopback OAM cells in the ingress cell flow for a match with the MC92520's programmable node ID (see Section 7.1.6.25–Section 7.1.6.28). If the source ID matches and the LSB of the loopback indication is 0, the cell is removed from the cell flow and copied to the cell extraction queue. A cell whose loopback indication equals 0 and whose source ID contains the default value of all 1s is copied to the cell extraction queue. (Such a cell is copied at every point along the OAM flow since the MC92520 cannot determine the location of the source.) Loopback cells with a loopback indication of 0 are removed at the connection/segment end point.

Table 6-11 summarizes the treatment of OAM loopback cells.

Table 6-11. Processing of OAM Loopback Cells

Loopback Indication	Source ID	Loopback Location ID	Treatment	
			Intermediate Pt	End Pt
1	X	Node ID	Copy and remove	Copy and remove
1	X	All 1s	—	Copy and remove
0	Node ID	X	Copy and remove	Copy and remove
0	All 1s	X	Copy	Copy and remove
0	X	X	—	Remove

Figure 6-31 shows an example where the loopback location ID is not an end point, so the actions taken do not depend on the flow type (F4/F5).

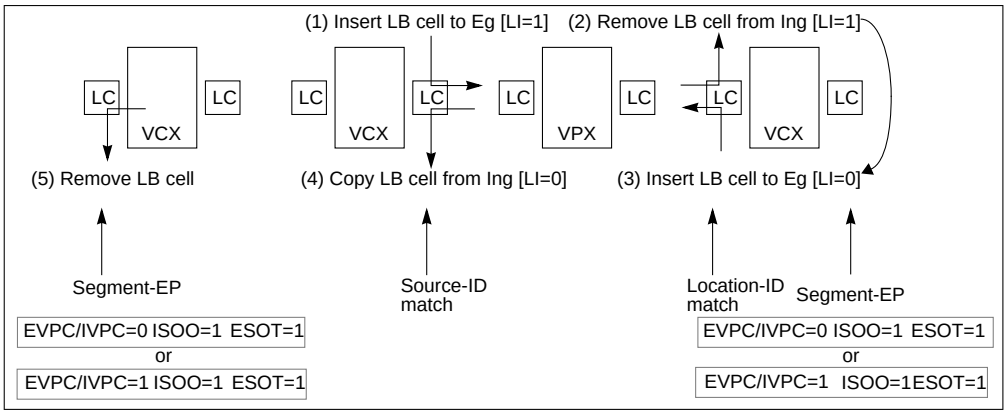


Figure 6-31. Loopback Not at End Point

Figure 6-32 shows the flow of an F5 OAM loopback cell with both the loopback_location_id and the source ID set to all 1s.

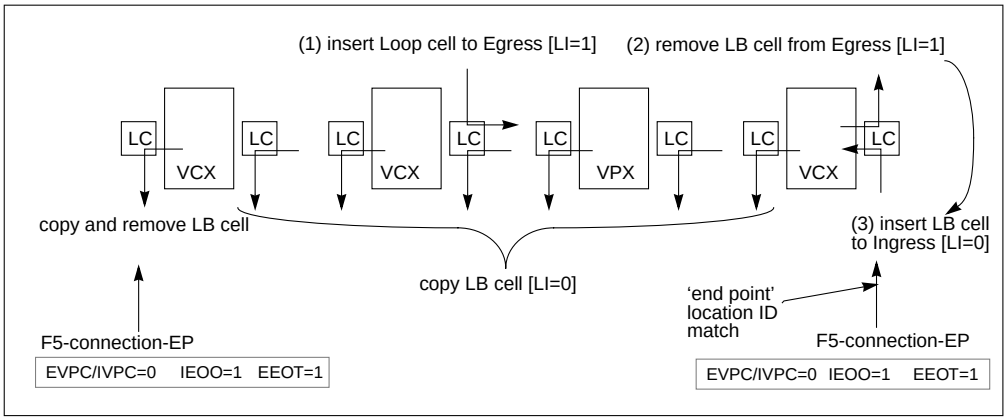


Figure 6-32. Loopback at End Point of VCC

6.5.5 Performance Management

Performance monitoring of a connection is accomplished by monitoring blocks of cells sent between end points of connections or segments. Blocks of cells are delimited by forward monitoring cells. Each forward monitoring cell (FMC) contains statistics about the immediately preceding block of cells. When an end point receives a forward monitoring cell, the statistics that the end point generated locally across the same block are added to produce a backward reporting cell (BRC) that is returned to the opposite end point.

The performance monitoring cell format and available processing options are discussed in the following sections.

6.5.5.1 Performance Management Cell Format

The following function specific fields as displayed in Figure 6-33 are used for both forward monitoring and backward reporting cells:

- Monitoring cell sequence number (MCSN)—The sequence number of the performance monitoring cell modulo 256.
- Total user cell 0+1 number (TUC0+1)—Indicates the total number of user cells (modulo 65,536) transmitted before the forward monitoring cell was inserted.
- Total user cell 0 number (TUC0)—Indicates the total number of CLP=0 user cells (modulo 65,536) transmitted before the forward monitoring cell was inserted.
- Time-stamp (TSTP)—May be used to represent the time when the cell was inserted.

The following field is used for forward monitoring cells:

- Block error detection code (BEDC0+1)—Even parity over the payload of the block of user cells transmitted since the last forward monitoring cell.

The following fields are used for backward reporting cells:

- Total received cell count 0 (TRCC0)—Indicates the total number of CLP = 0 user cells (modulo 65,536) received before the forward monitoring cell was received.
- Block error result (BLER)—Carries the number of errored parity bits detected by the BEDC of the received forward monitoring cell.
- Total received cell count 0+1 (TRCC0+1)—Indicates the total number of user cells (modulo 65,536) received before the forward monitoring cell was received.

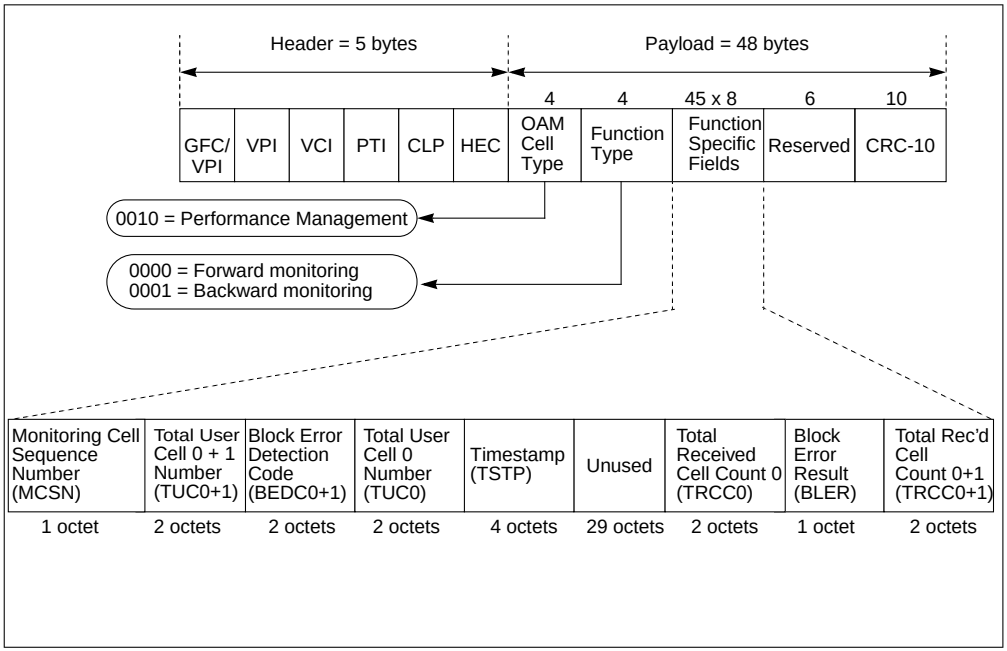


Figure 6-33. Performance Management Cell

6.5.6 Performance Monitoring

The MC92520 supports simultaneous bidirectional PM block tests on all connections. When running a bidirectional test, forward monitoring cells are generated for one direction and checked for the other direction. At the end point of a VPC, where the VCCs are accessible, performance monitoring (PM) cell generation is supported on either the individual VCCs or on the VPC as a whole, but not both simultaneously (see Appendix D). At a connection, end point PM cell generation is supported on the connection or on a segment, but not both simultaneously. In other words, an individual user information cell may belong to at most one active PM test. This limitation is imposed due to the potential complexity of multiple PM calculations during the processing of a single cell.

To run a block test, initialize the static bits of the OAM table at both the originating and terminating points. This defines the type of test to be run. The MCSN, TUC, BEDC, and

TUC0 fields of the OAM table are also initialized, normally to 0. To start the block test, the microprocessor inserts the first forward monitoring cell using format V of the cell descriptor (see Section 7.3.1.1, “Cell Descriptor”). When this cell is processed at the originating point, the test_running bit of the OAM table entry is set, indicating that the block test has begun. As long as the test_running bit is set, all cells involved in the block test (user cells and those with reserved values of VCI or PTI that are not excluded by the performance monitoring exclusion register (PMER)) are counted in the TUC0+1 field and included in the BEDC calculation. CLP = 0 cells are also counted in the TUC0 field. Stop the block test at any time by resetting the test_running bit.

Each time the value of the TUC0+1 field reaches a multiple of the block size, the MC92520 stores the fact that an FMC must be inserted in the same direction. At the next insertion opportunity, an FMC is generated using the current MCSN, TUC, BEDC, and TUC0 fields of the OAM table entry and inserted into the cell flow. In order to meet the requirement of inserting a forward monitoring cell within one-half of the block size on end-to-end block tests, the MC92520 must have enough opportunities to insert the FMCs. The insertion opportunities are limited by the insertion leaky bucket parameters (see Section 5.1.6, “Inserting Cells in the Ingress Flow” and Section 5.2.3, “Inserting Cells into the Egress Flow”). Additionally, the insertion opportunities in the ingress cell flow are limited by the number of empty slots (see Section B.4, “Maintenance Slot Parameters).

If an end-to-end FMC is not inserted within one-half of the block size and another user cell is processed, the FMC queue end-to-end overflow (IR[FQEO]) bit is set. This is a warning only, and the FMC is still inserted when the opportunity arises. In the event that any FMC is not inserted for an entire block and another user cell arrives, the FMC queue overflow (IR[FQO]) bit is set. This indicates that there is an FMC missing from the block test because only one FMC is generated to cover two blocks. The CLP bit of internally generated FMCs is taken from the FCLP bit of the OAM table. This provides the programmability required by Bellcore [see Reference 10, Appendix G].

When the cell is actually inserted, the function-specific fields of generated FMCs are coded as follows:

- Monitoring cell sequence number (MCSN)—The MCSN field of the OAM table is inserted in this field and then incremented.
- Total user cell 0+1 number (TUC0 +1)—The TUC field of the OAM table is inserted in this field.
- Block error detection code (BEDC)—The BEDC field of the OAM table is inserted in this field and then cleared.
- Total user cell 0 number (TUC0)—The TUC0 field of the OAM table is inserted in this field.
- Time-stamp (TSTP)—This field is coded either with the default value of all 1’s or with the current MC92520 cell time, depending on the value of the FMC time-stamp enable (FTM) bit of the ATMC CFB configuration register (ACR).

- The remaining 34 octets of the function-specific fields are coded with 0x6A since they are unused in forward monitoring cells.

As each FMC arrives at the terminating end point, the block error result is calculated in accordance with ATM standards [see Reference 2, Appendix G]. It is placed in the payload of an OAM fields template (as described in Section 7.3.2.1.5, “OAM Fields Template Indication”). The TUC and TUC0 fields of the template are taken from the payload of the FMC. The TRCC and TRCC0 fields of the template are taken from the TUC and TUC0 fields of the OAM table entry. Then the OAM fields template is sent to the cell extraction queue. The FMC is removed from the cell flow at the connection-segment OAM termination point.

The microprocessor may use the payload of the OAM fields template structure in preparing a cell for insertion (see Section 7.3.1, “Inserted Cell Structure” for the inserted cell structure options). If one of the “payload generation from OAM fields template” options for the inserted BRC is chosen, the MC92520 generates the payload of a backward reporting cell and inserts the BRC in the backward flow.

When the cell is actually inserted, the function-specific fields of generated BRCs are coded as follows:

- Monitoring cell sequence number (MCSN)—The BMCSN field of the OAM table is inserted in this field and then incremented.
- Total user cell 0+1 number (TUC0 + 1)—The TUC field of the OAM fields template is inserted in this field.
- Block error detection code (BEDC)—This field is coded with 0x6A.
- Total user cell 0 number (TUC0)—The TUC0 field of the OAM fields template is inserted in this field.
- Time-stamp (TSTP)—This field is coded either with the default value of all 1’s or with the current MC92520 cell time, depending on the value of the FMC time-stamp enable (FTM) bit of the ATMC CFB configuration register (ACR).
- Total received cell count 0 (TRCC0)—The TRCC0 field of the OAM fields template is inserted in this field.
- Block error result (BLER)—The BLER field of the OAM fields template is inserted in this field.
- Total received cell count 0 + 1 (TRCC0 + 1)—The TRCC field of the OAM fields template is inserted in this field.
- The remaining 29 octets of the function-specific fields are coded with 0x6A since they are unused in backward reporting cells.

The results of the block test are collected at data storage points. The storage point may be the originating point, the terminating point, or any point in between. In support of data storage points, the MC92520 provides options to copy FMCs and BRCs on a connection basis. The context parameter table bits that control these options are listed in Table 6-12.

BRCs are removed from the cell flow at the originating end point.

Table 6-12. Performance Monitoring Bits

Logical Name	Bit Name	Table	Stat/ Dyn	Used By	Explanation
Copy segment FMCs	ECSF ICSF	Context parameters	S	Egress/ Ingress	Copy received segment FMC to cell extraction queue. See Table 7-64 and Table 7-65.
Copy end-to-end FMCs	ECEF ICEF	Context parameters	S	Egress/ Ingress	Copy received end-to-end FMC to cell extraction queue. See Table 7-64 and Table 7-65.
Copy segment BRCs	ECSB ICSB	Context parameters	S	Egress/ Ingress	Copy received segment BRC to cell extraction queue. See Table 7-64 and Table 7-65.
Copy end-to-end BRCs	ECEB ICEB	Context parameters	S	Egress/ Ingress	Copy received end-to-end BRC to cell extraction queue. See Table 7-64 and Table 7-65.

An alternative method for collecting the results of the PM test is to define an OAM entry at an intermediate point. This point is treated like the terminating point described above, except that the FMC is not removed from the cell flow. When an FMC is received, the calculation of the BLER is performed at this point also, and an OAM fields template containing the BRC fields is sent to the cell extraction queue. Because performance monitoring tests are run on a small fraction of the connections and each test requires considerable storage, a separate OAM table is defined in external memory for storing the fields needed to support active PM tests. The context parameters table records contain pointers to the records of the OAM table. The OAM_PTR is valid only if the ingress OAM pointer is valid (IOPV) bit or the egress OAM pointer is valid (EOPV) bit is set (refer to Section 7.2.3.4, “Common Parameters”). Table 6-13 describes the fields of the OAM table.

Table 6-13. OAM Table Fields

Logical Name	Field Name	Number of bits	Stat/ Dyn	Meaning
FMC_Gen	FMCG	1	S	This is the originating point. Generate FMCs when necessary.
Connection_Identifier	ECI/ICI	16	S	Connection identifier of connection on which PM is being performed. See Appendix D for an explanation of the use of this field when VCCs are bundled.
F4_level	F4	1	S	Defines the level of the block test
BT_SEG/E2E	SEG	1	S	Defines the scope of the block test
FMC CLP bit	FCLP	1	S	Is used as the CLP bit of generated FMC cells
Block_Size	BLK	2	S	Encodes the block size (128, 256. 512, 1024)
Test_Running	TR	1	D	Set when an FMC is processed.
MCSN	MCSN	8	D	Monitoring cell sequence number
TUC	TUC	16	D	Total user cell count

Table 6-13. OAM Table Fields (Continued)

Logical Name	Field Name	Number of bits	Stat/ Dyn	Meaning
BEDC	BEDC	16	D	Block error detection code (BIP-16)
TUC0	TUC	16	D	Total user cell (CLP=0) count
BRC MCSN	BMCSN	8	D	Monitoring cell sequence number for backward reporting cells
Last_MCSN	LMCSN	8	D	Used by the MC92520 to store the MCSN from the previous FMC
TUC_Difference	TUCD	16	D	Used by the MC92520 to store the difference between the received TUC and the local TUC (TRCC)

Figure 6-34 shows the flow of a block test on a VPC segment where the segment end points are outside, thereby including the switch in the test.

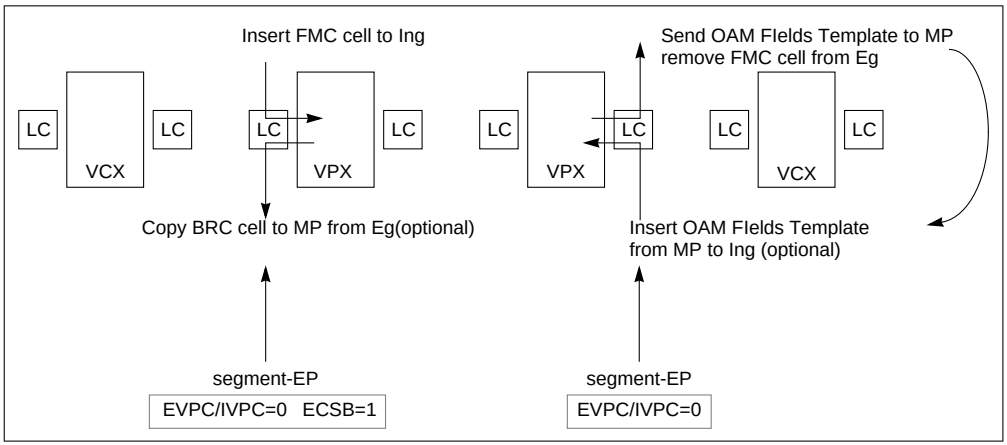


Figure 6-34. Performance Management Block Test on a VPC Segment

Figure 6-35 shows the flow of a block test on a VCC using an intermediate data collection point. The end points of the F5 flow have been defined outside the switches.

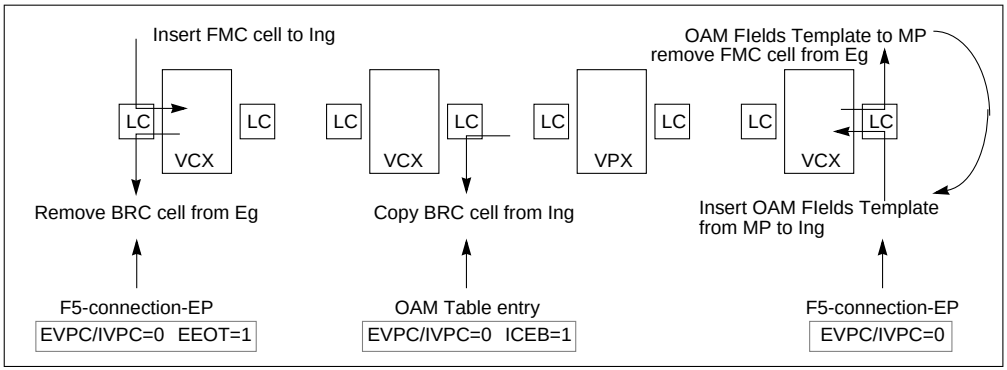


Figure 6-35. Performance Management Block Test on a VCC

Figure 6-36 shows the F4 flows for the case where the VP connection end points are logically at the edges of the VCX switches at which a set of VCCs are collected to be a VPC.

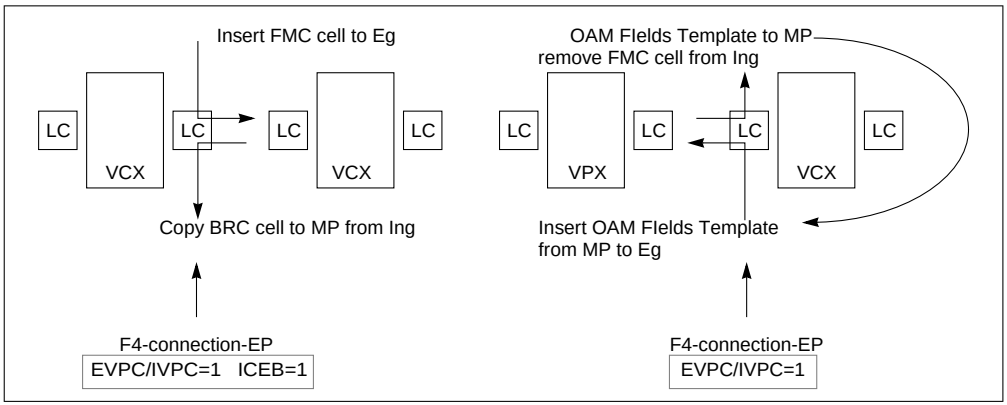


Figure 6-36. F4 PM Block Test on a VPC Internal to the Network

Figure 6-37 shows the F4 flows for the case where the VP connection is routed to the user end point through a VPX. In this case the connection end point must be on the LC board closest to the user.

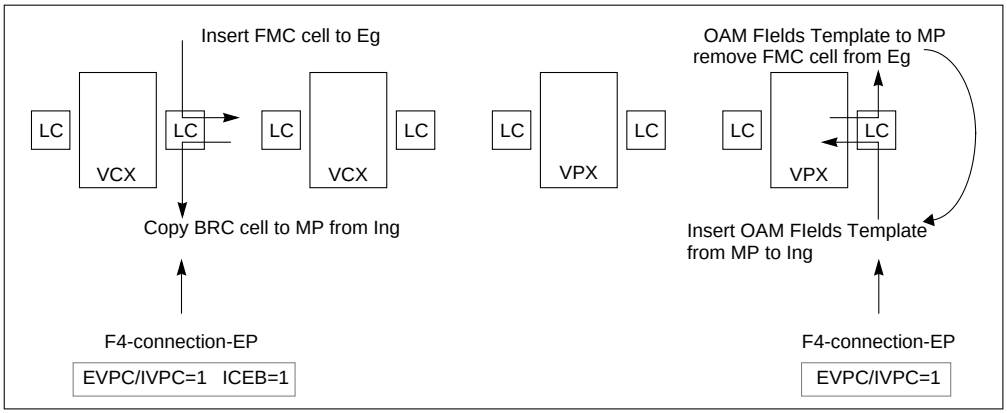


Figure 6-37. F4 PM Block Test on a VPC that Crosses the UNI

6.5.7 Activation/Deactivation OAM Cells

The MC92520 does not provide direct activation or deactivation of OAM features. Instead, activation/deactivation OAM cells are generated by the microprocessor. The MC92520 simply inserts these cells into one of the cell flows (ingress or egress), as directed. This procedure allows the microprocessor to maintain tight control of the OAM. Received activation/deactivation cells may be copied to the microprocessor at any point by setting the appropriate copy other OAM bit. Setting the ECOT/ICOT bit in order to copy activation/deactivation cells at points other than the end point could be useful for intermediate points that are collecting performance monitoring (PM) data and need to know when the PM test has been deactivated. All received activation/deactivation OAM cells are removed from the cell flow at the segment/connection end point. Some examples of common activation/deactivation flows are shown in the following figures.

Figure 6-38 shows the F4 flows for the case where the VP connection end points are logically at the edges of the VCX switches at which a set of VCCs are collected to be a VPC.

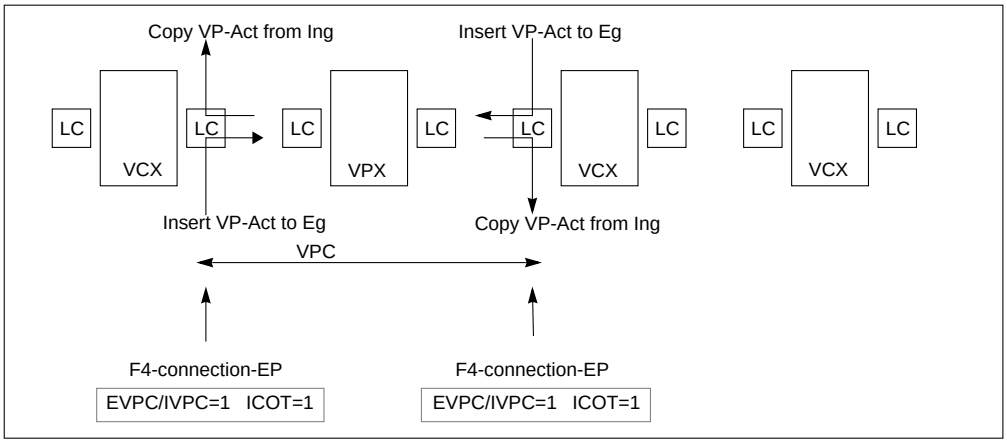


Figure 6-38. F4 OAM Flow for a VPC Internal to the Network

Figure 6-39 shows the F4 flows for the case where the VP connection is routed to the user end point through a VPX. In this case the connection end point must be on the LC board closest to the user.

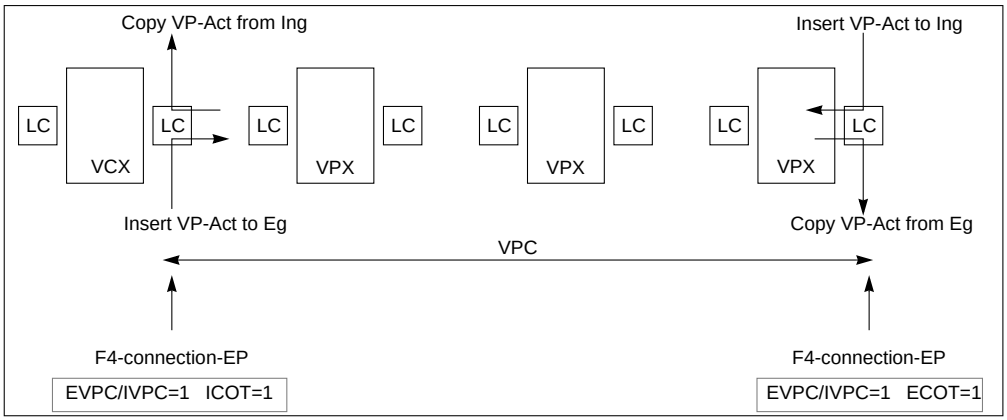


Figure 6-39. F4 OAM Flow for a VPC that Crosses the UNI

Figure 6-40 shows the F5 flows for a VCC that has one end point within the network and the other termination point outside the network. Therefore, the end point on the right is outside.

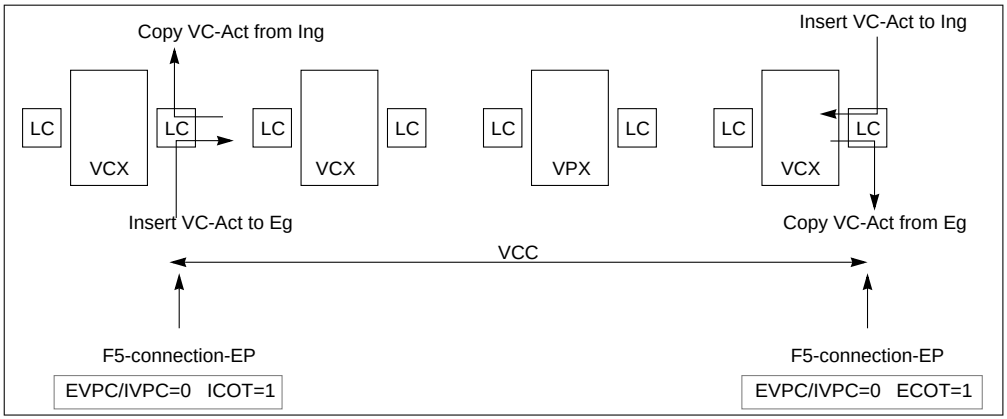


Figure 6-40. F5 OAM Flow

Chapter 7
Programming Model

The MC92520 programming model includes registers, external memory mapping, and a description of the data structures used in programming the chip. The following sections provide a detailed description of the programming model.

7.1 Registers Description

The MC92520 registers are divided into several groups. Some of the register groups may be accessed only when the MC92520 is in one of the two operating modes, setup mode and operate mode. The register groups are:

- Status reporting registers—These registers report on the MC92520 status and generally can be read and written by the processor in either of the MC92520 modes of operation (setup mode or operate mode).
- Control registers—These registers control the MC92520 operation and may be read and written by the processor in either mode of operation.
- Configuration registers—These registers are used to define the MC92520 configuration and can be read by the processor in either mode of operation (setup mode or operate mode). These registers can be written by the processor only in the setup mode.
- Cell insertion registers—These registers are used for cell insertion into the MC92520 cell flow, and are written by the processor when the MC92520 is in operate mode. In order to improve performance, the MC92520 cell insertion registers receive special treatment and can be accessed without wait states. See Section 4.5.1.2.2, “Cell Insertion Register Write.”
- Cell extraction registers—These registers are used for copying cells from the MC92520 cell flows, and can be read by the processor when the MC92520 is in operate mode. To improve performance, these registers receive special treatment and can be accessed without wait states. See Section 4.5.1, “Processor Read and Write Operations” for more information.
- Pseudo registers—The processor can write to these registers in either mode to perform certain operations on the MC92520.

- External memory—The processor uses this memory space to access external memory. The MC92520 drives the external memory interface as described in Section 4.4, “External Memory Interface.” If the destructive memory space is used, the MC92520 automatically provides a write-back of zeros to each external memory location that is read. The external memory can be accessed when the MC92520 is in setup mode or access can be requested for the next maintenance slot.

The status reporting registers, control registers, and configuration registers are set to their default values after power-up reset. A software reset sets the same registers, but does not affect the PLL-related registers and register bits. Figure 7-1 shows the MC92520 memory space addressable by the microprocessor using the MADD bus.

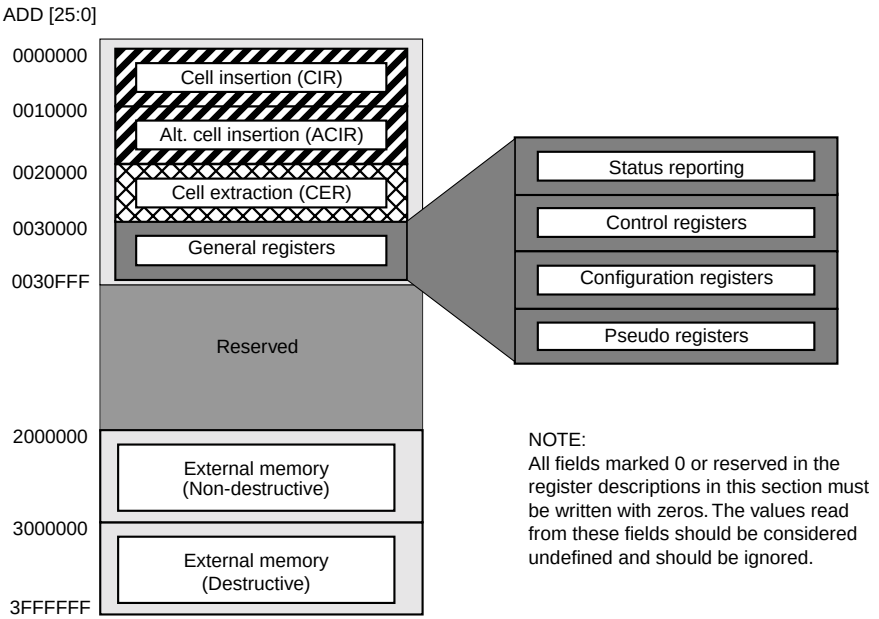


Figure 7-1. Memory Map

7.1.1 Cell Insertion Registers (CIR0–CIR15)

These 16 write-only 32-bit registers are used for cell insertion. For the cell insertion format, refer to Section 7.3.1. For the cell insertion modes of operation, refer to Section 3.2.3.1. The cell insertion registers are mapped into two distinct address spaces, a full address space (Section 7.1.1.1) and an alternate address space (Section 7.1.1.2).

7.1.1.1 Cell Insertion Address Space

In the cell insertion address space, all 16 of the registers are addressable. The trigger register that instructs the MC92520 to transfer the cell from the cell insertion registers to

the ingress or egress insertion queues is CIR15. The address space is intended for inserting full cells. The physical addresses of the cell insertion registers are shown in Figure 7-2.

CIR0	00_0000_0000_xxxx_xxxx_xx00_0000
CIR1	00_0000_0000_xxxx_xxxx_xx00_0100
CIR2	00_0000_0000_xxxx_xxxx_xx00_1000
CIR3	00_0000_0000_xxxx_xxxx_xx00_1100
CIR4	00_0000_0000_xxxx_xxxx_xx01_0000
CIR5	00_0000_0000_xxxx_xxxx_xx01_0100
CIR6	00_0000_0000_xxxx_xxxx_xx01_1000
CIR7	00_0000_0000_xxxx_xxxx_xx01_1100
CIR8	00_0000_0000_xxxx_xxxx_xx10_0000
CIR9	00_0000_0000_xxxx_xxxx_xx10_0100
CIR10	00_0000_0000_xxxx_xxxx_xx10_1000
CIR11	00_0000_0000_xxxx_xxxx_xx10_1100
CIR12	00_0000_0000_xxxx_xxxx_xx11_0000
CIR13	00_0000_0000_xxxx_xxxx_xx11_0100
CIR14	00_0000_0000_xxxx_xxxx_xx11_1000
CIR15	00_0000_0000_xxxx_xxxx_xx11_1100

ACIR0	00_0000_0001_xxxx_xxxx_xxxx_x000
ACIR1	00_0000_0001_xxxx_xxxx_xxxx_x100

x = don't care

Figure 7-2. Cell Insertion Register Addresses

7.1.1.2 CIR Alternate Address Space

In the cell insertion register alternate address space, only ACIR0 and ACIR1 are addressable. The trigger register that instructs the MC92520 to transfer the cell from the cell insertion registers to the ingress or egress insertion queues is ACIR1. This address space is used for inserting cells whose header and payload are generated by the MC92520. See Figure 7-2 for the physical addresses of the CIR alternate address space.

7.1.2 Cell Extraction Registers (CER0–CER15)

These 16 read-only 32-bit registers are used for cell extraction. For the cell out format, refer to Section 7.3.2; for the cell extraction modes of operation, refer to Section 3.2.3.2.

7.1.2.1 Cell Extraction Address Space

In the cell extraction address space, all 16 registers are addressable. The trigger register is CER15. It informs the MC92520 that the cell in the cell extraction registers was read and that a new cell can be loaded from the cell extraction queue into the cell extraction registers. This address space is intended for extracting full cells. The physical addresses of the cell extraction registers are shown in Figure 7-3

CER0	00_0000_0010_xxxx_xxxx_xx00_0000
CER1	00_0000_0010_xxxx_xxxx_xx00_0100
CER2	00_0000_0010_xxxx_xxxx_xx00_1000
CER3	00_0000_0010_xxxx_xxxx_xx00_1100
CER4	00_0000_0010_xxxx_xxxx_xx01_0000
CER5	00_0000_0010_xxxx_xxxx_xx01_0100
CER6	00_0000_0010_xxxx_xxxx_xx01_1000
CER7	00_0000_0010_xxxx_xxxx_xx01_1100
CER8	00_0000_0010_xxxx_xxxx_xx10_0000
CER9	00_0000_0010_xxxx_xxxx_xx10_0100
CER10	00_0000_0010_xxxx_xxxx_xx10_1000
CER11	00_0000_0010_xxxx_xxxx_xx10_1100
CER12	00_0000_0010_xxxx_xxxx_xx11_0000
CER13	00_0000_0010_xxxx_xxxx_xx11_0100
CER14	00_0000_0010_xxxx_xxxx_xx11_1000
CER15	00_0000_0010_xxxx_xxxx_xx11_1100

x = don't care

Figure 7-3. Cell Extraction Register Addresses

7.1.3 General Register List

The registers that comprise each register group are displayed in Table 7-1, "General Register List."

Table 7-1. General Register List

Register Group	Register Name	Mnemonic	ADD (25:0)	Ref. Page
Status reporting	Interrupt register	IR	00301E0	7-7
	Interrupt mask register	IMR	00301E4	7-10
	HOL blocking detection status register	HOLDSR	00301FC	7-12
	PLL status register	PLLSR	0030318	7-12
	External memory maintenance slot read access results	EMRSLT	00304xx	7-12
	MC92520 revision register	RR	0030BFC	7-13
	Last cell processing time register	LCPTR	0030EE0	7-14
	ATMC CFB revision register	ARR	0030FE0	7-14

Table 7-1. General Register List (Continued)

Register Group	Register Name	Mnemonic	ADD (25:0)	Ref. Page
Control registers	PLL control register	PLLCR	0030310	7-14
	Microprocessor control register	MPCTLR	00301E8	7-15
	Maintenance control register	MACTLR	00301A0	7-15
	Cell extraction queue filtering register 0	CEQFR0	00301EC	7-16
	Cell extraction queue filtering register 1	CEQFR1	00301F0	7-17
	Cell extraction queue priority register 0	CEQPR0	00301F4	7-17
	Cell extraction queue priority register 1	CEQPR1	00301F8	7-18
	Ingress insertion leaky bucket register	IILB	0030224	7-18
	Ingress insertion bucket fill register	IIBF	0030220	7-19
	Egress insertion leaky bucket register	EILB	0030204	7-19
	Egress insertion bucket fill register	EIBF	0030200	7-20
	Internal scan control register	ISCR	0030040	7-20
	Ingress link register 0–15	ILNK0–ILNK15	0030000–003003C	7-21
	Egress link register 0-15	ELNK0–ELNK15	0030100–003013C	7-23
	Egress link enable register	ELER	0030208	7-22
	Ingress billing counter table pointer	IBCTP	0030180	7-24
	Egress billing counter table pointer	EBCTP	0030184	7-24
	Policing counter table pointer	PCTP	0030188	7-24
	Cell time register	CLTM	0030240	7-25
	Ingress processing control register	IPLR	0030824	7-25
	Egress processing control register	EPLR	0030828	7-25
	Indirect external memory access address register	IAAR	0030810	7-26
	Indirect external memory access data register	IADR	0030814	7-27
	Cell arrival period multiplier	CAPM00–CAPM63	00400xx	7-27

Table 7-1. General Register List (Continued)

Register Group	Register Name	Mnemonic	ADD (25:0)	Ref. Page
Configuration registers	PLL range register	PLLRR	0030314	7-28
	Microprocessor configuration register	MPCONR	0030E80	7-28
	Ingress PHY configuration register	IPHCR	0030CA0	7-29
	Egress PHY configuration register	EPHCR	0030C80	7-30
	Ingress switch interface configuration register	ISWCR	0030800	7-32
	Egress switch interface configuration register 0	ESWCR0	0030804	7-33
	Egress switch interface configuration register 1	ESWCR1	0030834	7-37
	Egress switch overhead information register 0	ESOIR0	0030808	7-38
	Egress switch overhead information register 1	ESOIR1	0030818	7-39
	UNI register	UNIR	0030EC0	7-40
	Ingress processing configuration register	IPCR	0030E20	7-40
	Egress processing configuration register	EPCR	0030E00	7-42
	Egress multicast configuration register	EMCR	0030D00	7-44
	ATMC CFB configuration register	ACR	0030EA0	7-45
	MC92520 general configuration register	GCR	003080C	7-46
	Context parameters table pointer	CPTP	0030D60	7-47
	OAM table pointer	OTP	0030D64	7-48
	Dump vector table pointer	DVTP	0030D68	7-48
	VC table pointer	VCTP	0030D74	7-48
	Multicast translation table pointer	MTTP	0030D6C	7-48
	Flags table pointer	FTP	0030D70	7-48
	Egress link counters table pointer	ELCTP	0030D80	7-49
	Ingress link counters table pointer	ILCTP	0030D84	7-49
	Context parameters extension table pointer	CPE TP	0030D88	7-49
	Node ID registers 0–3	ND0–ND3	0030E40–0030E4C	7-49
	Ingress VCI copy register	IVCR	0030E24	7-50
	Egress VCI copy register	EVCR	0030E04	7-51
	Ingress VCI remove register	IVRR	0030E28	7-52
	Egress VCI remove register	EVRR	0030E08	7-52
	Performance monitoring exclusion register	PMER	0030E60	7-53
	RM overlay register	RMOR	003081C	7-53
	CLP transparency overlay register	CTOR	003082C	7-54
	Egress overhead manipulation register	EGOMR	0030820	7-54
	GFR configuration register	GFRCR	0030830	7-56

Table 7-1. General Register List (Continued)

Register Group	Register Name	Mnemonic	ADD (25:0)	Ref. Page
Pseudo registers	Software reset register (pseudo)	SRR	0030300	7-57
	Enter operate mode register (pseudo)	EOMR	0030304	7-58
	Start scan register (pseudo)	SSR	0030308	7-58

7.1.4 Status Reporting Registers

The status reporting registers of the MC92520 may be read and written by the processor in either setup mode or operate mode.

7.1.4.1 Interrupt Register (IR)

The MC92520 interrupt register (IR) includes all the MC92520 general status information that can cause an interrupt if the appropriate interrupt mask bit is set in the interrupt mask register (IMR); see Section 7.1.4.2 for more information. The IR bits can be divided into three classes:

- Real status bit—The MC92520 sets or clears a real status bit to indicate the current status. The user cannot change the value of this bit.
- Threshold bit—The MC92520 sets a threshold bit when a threshold is crossed. The user can reset the threshold bit by writing to the IR with the bit location set after the value drops below the threshold. While the value remains at or above the threshold, the threshold bit cannot be reset and any reset attempt is ignored.
- Sticky bit—The MC92520 sets a sticky bit when an event occurs. The user can reset the sticky bit by writing to the IR with the bit location set. However, if the event has occurred again after the IR was last read, the sticky bit is not reset. This prevents missed interrupts that could lead to deadlock situations. Unless it is defined as a specific class, IR bits are sticky bits by default.

7.1.4.1.1 Interrupt Register Fields

Figure 7-4 shows the IR field locations within the register.

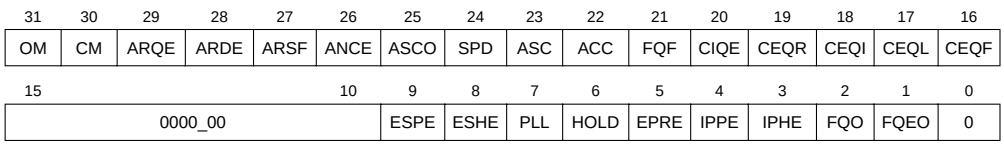


Figure 7-4. Interrupt Register (IR) Fields

Table 7-2. IR Field Descriptions

Bits	Name	Description
31	OM	Operate mode. The OM bit reports on the MC92520 mode of operation. It is a real status bit. 0 setup mode 1 operate mode
30	CM	Cycle mode. The CM bit reports on the MC92520 cycle mode (maintenance or normal). It is a real status bit. The processor may use the CM bit to verify that maintenance slots occur at all, or that they occur when expected. The CM bit is set after reset because no cells are processed and all cycles are maintenance cycles during setup mode. 0 Normal cycle. The MC92520 uses the external memory for cell processing. 1 Maintenance cycle. The MC92520 use the external memory to perform queued access requests to execute microprocessor initiated external memory maintenance.
29	ARQE	Access request error. This bit is set to indicate that an error occurred while requesting maintenance slot external memory accesses. This occurs if the FIFO that holds the requests overflows due to requests being received from the MPI bus faster than they can be serviced during maintenance slots.
28	ARDE	Access read error. This bit is set to indicate that an error occurred while reading the results of maintenance slot external memory accesses. This occurs if the FIFO that holds the results is read while empty.
27	ARSF	Access result FIFO full. This bit is set to indicate that the FIFO that holds results of maintenance slot external memory read cycles (the EMRSLT FIFO) is full. This indicates that no more maintenance slot external memory accesses can occur until some of the results are read. Note that the EMRSLT FIFO cannot overflow.
26	ANCE	Access not complete error. This bit is set to indicate that an error occurred while servicing an atomic access group. Such an error occurs if a sequence of requests is marked for completion within one maintenance slot (that is, an atomic access), but does not complete within the slot. Non-completion occurs if atomic accesses that require more than 32 external memory cycles are requested, or if an access result FIFO full (ARSF) condition occurs during the processing of an atomic access group.
25	ASCO	Access sequence complete overflow. The ASCO bit is set if the MC92520 internal access sequence complete counter overflows. This counter has a maximum value of 127 and is incremented for every completed access request that is marked with an end-of-sequence indication. The counter is decremented by the microprocessor through clearing the access sequence complete (ASC) status bit described below.
24	SPD	Scan process done. The SPD bit is set when the MC92520 has completed the internal scan process operation.
23	ASC	Access sequence complete. The ASC bit is a real status bit and is set when the MC92520 completed one or more external memory access requests that were marked with an end-of-sequence indication; that is, the bit reflects a non-zero count of completed access request sequences. It is assumed that the microprocessor acknowledges the status report for each completed access request sequence by clearing (writing a 1 to) the ASC bit and re-reading the status to detect (and acknowledge) any additional completed sequences.
22	ACC	Access error. This bit is set when any MPI bus cycle attempts to read or write into a block of reserved space, to write into a read-only space (the cell extraction space), or to read from a write-only space (the cell insertion space).
21	FQF	FMC queue full. This bit reports that the internal FMC queue is full. The reason for this is that the FMC generation rate is higher then the allocated insertion bandwidth. Insertion rate is controlled by the insertion leaky bucket. The FQF bit is valid only if PM on all connections (PMAC) in the ATMC CFB configuration register (ACR) is set.

Table 7-2. IR Field Descriptions (Continued)

Bits	Name	Description
20	CIQE	Cell insertion queue empty. The cell insertion queue is empty and a new cell may be inserted into the ingress or egress cell flow. The MCIREQ output signal is also asserted and may be used instead of the interrupt.
19	CEQR	Cell extraction queue ready. This bit informs the processor that a cell is ready to be read from the cell extraction registers (CER0–CER15). The MCOREQ output signal is also asserted and may be used instead of the interrupt.
18	CEQI	Cell extraction queue interrupt threshold. This bit is set when the cell extraction queue has reached the programmable interrupt threshold defined by the extraction queue interrupt threshold (EQIT) field of the microprocessor control register (MPCTLR). Because this is a threshold bit, the MC92520 ignores any attempt to clear this bit while the number of cells in the cell extraction queue equals or exceeds the interrupt threshold value.
17	CEQL	Cell extraction queue low-priority threshold. This bit is set when the cell extraction queue has reached the programmable low-priority threshold defined by the extraction queue low-priority threshold (EQLPT) field of the microprocessor control register (MPCTLR). low-priority cells that are directed to this queue are dropped until the queue is read and contains fewer cells than the low-priority threshold value. This is a threshold bit; thus, an attempt to reset this bit while the number of cells in the cell extraction queue equals or exceeds the low-priority threshold value is ignored.
16	CEQF	Cell extraction queue full. This bit is set when the cell extraction queue is full. Additional cells that are directed to this queue are dropped until this queue is read. This is a threshold bit; thus, an attempt to reset this bit while the cell extraction queue is full is ignored.
15–10	—	Reserved, should be cleared.
9	EPSE	Egress switch parity error. This bit is set when the egress switch interface block detects a parity error.
8	ESHE	Egress switch protocol handshake error. This bit is set when the egress switch interface block detects a protocol handshake error. The MC92520 produces the ESHE interrupt in the following cases: • The STXSOC signal is asserted too early if the number of enabled bytes between one assertion of the STXSOC signal and the next is smaller than the data structure size as defined by the ESNB field • The STXSOC signal is late if the number of enabled bytes between one assertion of the STXSOC signal and the next is larger than the data structure size as defined by the ESNB field. • The $\overline{\text{STXENB}}$ signal is asserted while STXCLAV is deasserted.
7	PLL	PLL lost lock. This bit is set when the PLL lost lock after lock was acquired; that is, this bit is never set if the MC92520 is used in PLL bypass mode or before the PLL has been enabled after an MC92520 reset. The current PLL lock status can be read any time from the PLL status register (PLLSR).
6	HOLD	HOL blocking detected. The HOLD bit is set when the egress PHY interface detects a HOL blocking condition on one or more of the enabled PHY ports. The HOLD status bit is not a sticky bit, but a combination of individual PHY port HOLD status and link enable bits. For more information, see Section 7.1.4.3, “HOL Blocking Detection Status Register (HOLDSR)” and Section 3.3.2.1.3, “Multi-PHY Operation” for information on HOL blocking prevention.

Table 7-2. IR Field Descriptions (Continued)

Bits	Name	Description
5	EPRE	Egress PHY cell routing error. The EPRE bit is set when the egress PHY detects that an arriving cell is destined to a full FIFO when operating in the per-PHY FIFO configuration. In this case, the offending cell is discarded. In practice, this indicates an ATMC misconfiguration, caused by either of the following: • The global cell insertion rate is higher than a multi-PHY's drain rate (global cell insertion is not gated by individual FIFO full status), or • HOL blocking occurred or the HOL blocking prevention was not set up to assure that automatic flushing occurs before two cell insertions targeting the same multi-PHY port.
4	IPPE	Ingress PHY parity error. The IPPE bit is set when the ingress PHY interface detects a parity error. The cell that contains the parity error may be discarded at the PHY interface (see Section 5.1.1, “Assembling Cells”), and the ingress PHY interface continues its operation without any effect on the next cell.
3	IPHE	Ingress PHY protocol handshake error. The IPHE bit is set when the ingress PHY interface detects a protocol handshake error. The cell during whose reception the protocol handshake error occurred is discarded at the PHY interface (see Section 5.1.1), and the ingress PHY interface continues its operation without any effect on the next cell.
2	FQO	FMC queue overflow. The FQO bit is set when an attempt to insert a forward monitor cell (FMC) into the cell flow within one block size fails. The new FMC is inserted into the FMC queue. The previous FMC is not generated.
1	FQEO	FMC queue end-to-end overflow. The FQEO bit is set when an attempt to insert an end-to-end forward monitor cell (FMC) into the cell flow within one half of the block size fails. The MC92520 continues its attempt to insert the FMC. If it is not inserted within one block size, the FQO bit is also set.
0	—	Reserved, should be cleared.

7.1.4.2 Interrupt Mask Register (IMR)

The IMR contains an interrupt enable bit for each bit in the IR. The MC92520 generates an interrupt when both the IR bit and its corresponding enable bit are set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OME	CME	ARQEE	ARDEE	ARSFE	ANCEE	ASCOE	SPDE	ASCE	ACCE	FQFE	CIQEE	CEQRE	CEQIE	CEQLE	CEQFE
15					10	9	8	7	6	5	4	3	2	1	0
0000_00						ESPEE	ESHEE	PLLE	HOLDE	EPREE	IPPEE	IPHEE	FQOE	FQEOE	0

Figure 7-5. Interrupt Mask Register (IMR)

Table 7-3. IMR Field Descriptions

Bits	Name	Description
31	OME	Operate mode interrupt enable. When OME and OM are both set, an interrupt is generated.
30	CME	Cycle mode interrupt enable. When CME and CM are both set, an interrupt is generated.
29	ARQEE	Access request error interrupt enable. When ARQEE and ARQE are both set, an interrupt is generated.

Table 7-3. IMR Field Descriptions (Continued)

Bits	Name	Description
28	ARDEE	Access read error interrupt enable. When ARDEE and ARDE are both set, an interrupt is generated.
27	ARSFE	Access result FIFO full interrupt enable. When ARSFE and ARSF are both set, an interrupt is generated.
26	ANCEE	Access not complete error interrupt enable. When ANCEE and ANCE are both set, an interrupt is generated.
25	ASCOE	Access sequence complete overflow interrupt enable. When ASCOE and ASCO are both set, an interrupt is generated.
24	SPDE	Scan process done interrupt enable. When SPDE and SPD are both set, an interrupt is generated.
23	ASCE	Access sequence complete interrupt enable. When ASCE and ASC are both set, an interrupt is generated.
22	ACCE	Access error interrupt enable. When ACCE and ACC are both set, an interrupt is generated.
21	FQFE	FMC queue full interrupt enable. When FQF and FQFE are set, an interrupt is generated.
20	CIQEE	Cell insertion queue empty interrupt enable. When CIQEE and CIQE are both set, an interrupt is generated.
19	CEQRE	Cell extraction queue ready interrupt enable. When CEQRE and CEQR are both set, an interrupt is generated.
18	CEQIE	Cell extraction queue interrupt threshold interrupt enable. When CEQIE and CEQI are both set, an interrupt is generated.
17	CEQLE	Cell extraction queue low-priority threshold interrupt enable. When CEQLE and CEQL are both set, an interrupt is generated.
16	CEQFE	Cell extraction queue full interrupt enable. When CEQFE and CEQF are both set, an interrupt is generated.
15–10	—	Reserved, should be cleared.
9	ESPEE	Egress switch parity error interrupt enable. When ESPE and ESPEE are both set, an interrupt is generated.
8	ESHEE	Egress switch protocol handshake error interrupt enable. When ESHE and ESHEE are both set, an interrupt is generated.
7	PLLE	PLL lost lock interrupt enable. When PPL and PLLE are both set, an interrupt is generated.
6	HOLDE	HOL blocking detection interrupt enable. When HOLDE and HOLD are both set, an interrupt is generated.
5	EPREE	Egress PHY cell routing error interrupt enable. When EPREE and EPCRE are both set, an interrupt is generated.
4	IPPEE	Ingress PHY parity error interrupt enable. When IPPEE and IPPE are both set, an interrupt is generated.
3	IPHEE	Ingress PHY protocol handshake error interrupt enable. When IPHEE and IPHE are both set, an interrupt is generated.
2	FQOE	FMC queue overflow interrupt enable. When FQOE and FQO are both set, an interrupt is generated.

Table 7-3. IMR Field Descriptions (Continued)

Bits	Name	Description
1	FQEOE	FMC queue end-to-end overflow interrupt enable (FQEOE)—When FQEOE and FQEO are both set, an interrupt is generated.
0	—	Reserved, should be cleared.

7.1.4.3 HOL Blocking Detection Status Register (HOLDSR)

This status register reports the current HOL blocking detection status for all MC92520 egress multi-PHY ports. All HOLDSR bits are sticky, that is, once HOL blocking status is indicated, it must be explicitly reset by writing a 1 to the associated bit location. The individual bit values of HOLDSR are and-ed with the associated link enable (LE) bits and the result is or-ed to form the HOLD interrupt status bit described in Section 7.1.4.1, “Interrupt Register (IR).” This implies that a pending HOLD status can be cleared by either clearing the per-PHY status bit in HOLDSR or clearing the per-PHY link enable bit in the ELER or ELNK*n* register.

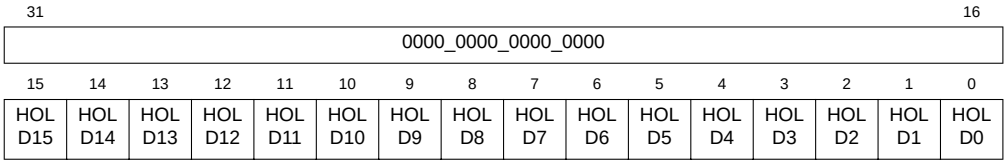


Figure 7-6. HOL Blocking Status Register (HOLDSR)

7.1.4.4 PLL Status Register (PLLSR)

This read-only register reports the current MC92520 phase-locked loop (PLL) lock status.

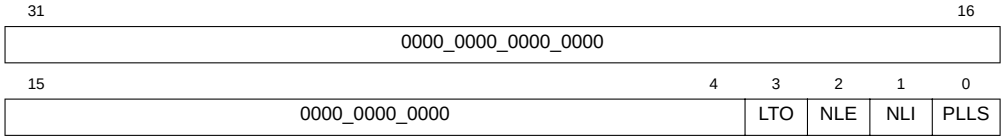


Figure 7-7. PLL Status Register (PLLSR)

Note that due to inter-clock domain synchronization, it is possible that momentarily none of the four indicated status bits may be asserted, or that two of the bits may be asserted simultaneously.

Table 7-4. PLLSR Field Descriptions

Bits	Name	Description
31–4	—	Reserved, should be cleared.
3	LTO	PLL lock time-out. This bit is set if the PLL timed-out while attempting to lock. A possible cause of time-out is improper connection of the ZCLKO output pin to the ZCLKI input pin.

Table 7-4. PLLSR Field Descriptions (Continued)

Bits	Name	Description
2	NLE	PLL not locked, attempting external lock. This bit is set while the PLL is attempting to achieve lock using the external feedback path.
1	NLI	PLL not locked, attempting internal lock. This bit is set while the PLL is attempting to achieve lock using the internal feedback path.
0	PLLS	Phase-locked loop status. The bit remains cleared if the MC92520 is used in PLL bypass mode. If the PLL is enabled, the PLLS bit is cleared while PLL lock status has not been acquired or re-acquired. The PLLS bit is set while the MC92520 PLL maintains lock.

7.1.4.5 External Memory Maintenance Slot Read Access Result (EMRSLT)

This read-only register contains the value available from the external memory maintenance slot read access result FIFO. This is the next available result from reading the external memory during a maintenance slot in operate mode.

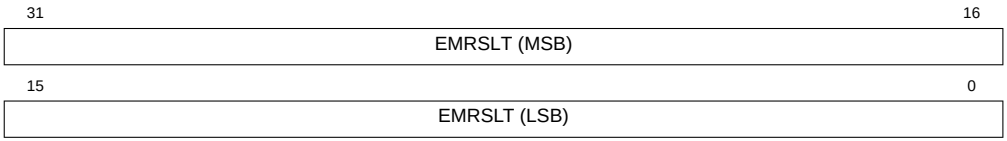


Figure 7-8. EM Maintenance Slot Read Access Result (EMRSLT) Fields

7.1.4.6 MC92520 Revision Register (RR)

This read-only register contains the MC92520 identification information.

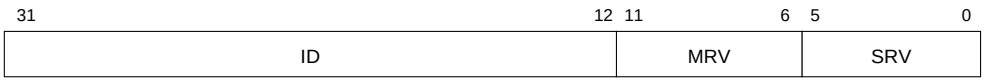


Figure 7-9. Revision Register (RR)

Table 7-5. RR Field Descriptions

Bits	Name	Description
31–12	ID	CSP identification number. Identifies the device as either an MC92520 or an MC92510
11–6	MRV	Major revision. This field indicates the processor's major revision number.
5–0	SRV	Sub-revision. This field indicates the processor's sub-revision number.

The following values of ID, MRV, and SRV are currently defined:

Table 7-6. Values of MC92520/MC92510 Revision Fields

ID	MRV	SRV	Processor / Revision
1000_0000_0000_0000_0010	00_0000	00_0000	MC92520, Revision A
1000_0000_0000_0000_0011	00_0000	00_0000	MC92510, Revision A

7.1.4.7 Last Cell Processing Time Register (LCPTR)

This read-only register contains the cell time of the most recent non-maintenance cycle. This value may be used to find the most recent entries in the dump vector table, which is updated in a cyclical fashion.

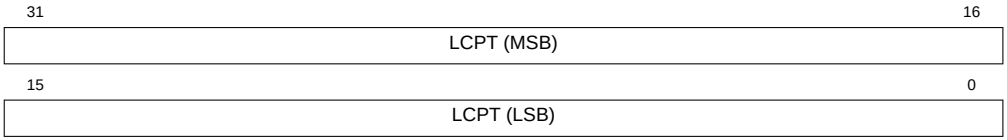


Figure 7-10. Last Cell Processing Time Register (LCPTR) Fields

7.1.4.8 ATMC CFB Revision Register (ARR)

This read-only register contains the ATMC CFB revision number.

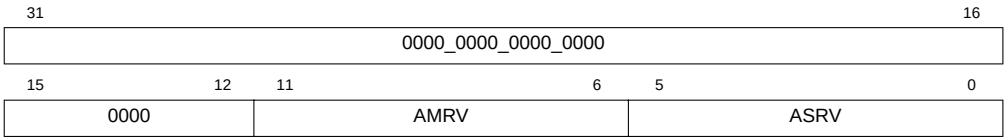


Figure 7-11. ATMC CFB Revision Register (ARR)

Table 7-7. ARR Field Descriptions

Bits	Name	Description
31–12	—	Reserved, should be cleared
11–6	AMRV	ATMC CFB major revision. This field indicates the ATMC CFB major revision number.
5–0	ASRV	ATMC CFB sub-revision. This field indicates the ATMC CFB sub-revision number.

The following values of AMRV and ASRV are currently defined:

Table 7-8. Values of ATMC CFB Revision Fields

AMRV	ASRV	ATMC CFB Revision
000000	000000	Revision A

7.1.5 Control Registers

The control registers controls the MC92520 operation, and may be read and written by the processor in either setup mode or operate mode.

7.1.5.1 PLL Control Register (PLLCR)

This register specifies PLL-related control parameters. Although the PLLCR can be modified any time, it is recommended to perform the optional PLL configuration and startup process only after a hardware or software reset, that is, in setup mode. See Section 3.1.1, “Configuring the Phase-Locked Loop (PLL)” for more details.

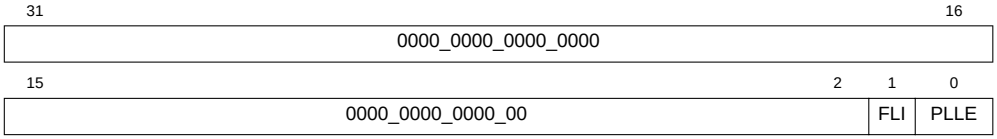


Figure 7-12. PLL Control Register (PLLCR)

Table 7-9. PLLCR Field Descriptions

Bits	Name	Description
31–2	—	Reserved, should be cleared.
1	FLI	PLL force lock internal. Setting this bit to one forces the PLL to attempt lock only via the internal feedback path. This setting may cause skew problems between ACLK and ZCLKI, and should only be used as a debugging tool.
0	PLLE	PLL enable. Setting a clear PLLE bit starts the MC92520 PLL lock acquisition process. Clearing a set PLLE bit disables the MC92520 PLL.

7.1.5.2 Microprocessor Control Register (MPCTLR)

This register specifies microprocessor-related control parameters.

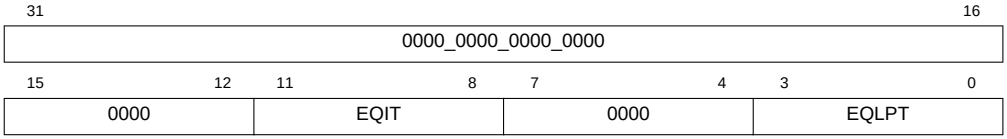


Figure 7-13. Microprocessor Control Register (MPCTLR) Fields

Table 7-10. MPCTLR Field Descriptions

Bits	Name	Description
31–12	—	Reserved, should be cleared.
11–8	EQIT	Extraction queue interrupt threshold. This field defines the value of the extraction queue interrupt threshold. When the extraction queue reaches the interrupt threshold, and the cell extraction queue interrupt threshold interrupt enable (CEQIE) bit in the IR is set, then the cell extraction queue interrupt (CEQI) bit is set. If EQIT is 0, the interrupt threshold is undefined and CEQI is never set.
7–4	—	Reserved, should be cleared.
3–0	EQLPT	Extraction queue low-priority threshold. This field defines the value of the extraction queue low-priority threshold. When the extraction queue reaches the low-priority threshold, the cell extraction queue low-priority threshold interrupt enable (CEQLE) bit in the IR is set. Low-priority cells that are directed to this queue are dropped until this queue is read and contains fewer cells than the low-priority threshold value. If EQLPT is 0, the low-priority threshold is undefined, CEQL is never set, and low-priority cells are not filtered out.

7.1.5.3 Maintenance Control Register (MACTLR)

This register defines parameters related to maintenance accesses.

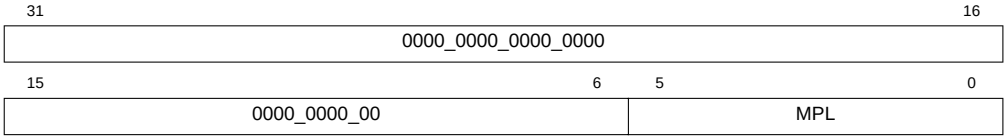


Figure 7-14. Maintenance Control Register (MACTLR)

Table 7-11. MACTRL Field Descriptions

Bits	Name	Description
31–6	—	Reserved, should be cleared.
5–0	MPL	Maintenance period length. This field specifies the number of cell processing slots between maintenance slots. Note that higher values of MPL result in fewer maintenance slots. The fraction of slots that are maintenance slots in normal operation ranges from 1/2 (MPL = 1) to 1/64 (MPL = 63). If MPL is 0, all the slots are maintenance slots. This is the situation after reset, and it may be used for setup of the external memory.

NOTE:

The MC92520 does not process cells while MPL = 0. MPL should be programmed with a non-zero value before switching to operate mode. If MPL is programmed to zero and is then placed in operate mode (for continuous maintenance accesses), cells may be lost.

7.1.5.4 Cell Extraction Queue Filtering Register 0 (CEQFR0)

The cell extraction queue filtering register 0 controls the reason-based filtering of cells that are copied to the cell extraction queue.

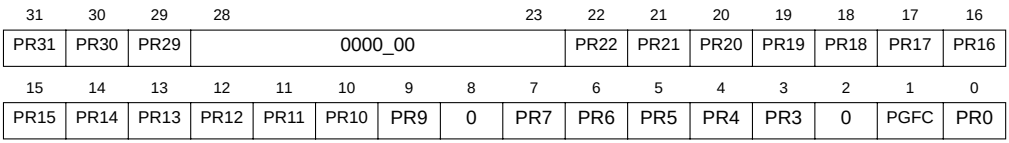


Figure 7-15. Cell Extraction Queue Filtering Register 0 (CEQFR0) Fields

Table 7-12. CEQFR0 Field Descriptions

Bits	Name	Description
31–29, 22–9, 7–3, 0	PRn	Pass reason n cells. If the PRn bit is set, a cell that is copied to the cell extraction queue with the extraction reason (RSN) field equal to n is actually placed in the cell extraction queue, rather than being dropped.)
28–23, 8, 2	—	Reserved, should be cleared.
1	PGFC	Pass GFC cells. If the PGFC bit is set, a cell that is copied to the cell extraction queue with the GFC reason (GFCR) bit set is actually placed in the cell extraction queue, rather than being dropped.

7.1.5.5 Cell Extraction Queue Filtering Register 1 (CEQFR1)

The cell extraction queue filtering register 1 controls the cell name-based filtering of cells that are copied to the cell extraction queue.

31	26	25	24	23	22	21	20	19	18	17	16
0000_00		PN9	PN8	PN7	PN6	PN5	PN4	PN3	PN2	PN1	0
15	10	9	8	7	6	5	4	3	2	1	0
0000_00		PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	0

Figure 7-16. Cell Extraction Queue Filtering Register 1 (CEQFR1)

Table 7-13. CEQFR1 Field Descriptions

Bits	Name	Description
31–26	—	Reserved, should be cleared.
25–17	PNn	Pass name n cells. If the PNn bit is set, a cell that is copied to the cell extraction queue with the extraction reason (RSN) field indicating that it was copied because an OAM copy bit is set and with the indication cell name (ICN) field equal to n is actually placed in the cell extraction queue, rather than being dropped.
16–10	—	Reserved, should be cleared.
9–1	PCn	Pass copy all name n cells. If the PCn bit is set, a cell that is copied to the cell extraction queue with the extraction reason (RSN) field indicating that it was copied because the copy all bit is set and with the indication cell name (ICN) field equal to n is actually placed in the cell extraction queue, rather than being dropped.
0	—	Reserved, should be cleared.

7.1.5.6 Cell Extraction Queue Priority Register 0 (CEQPR0)

The cell extraction queue priority register 0 controls the reason-based priority of cells that are copied to the cell extraction queue.

31	30	29	28	23	22	21	20	19	18	17	16
HR31	HR30	HR29	0000_00		HR22	HR21	HR20	HR19	HR18	HR17	HR16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HR15	HR14	HR13	HR12	HR11	HR10	HR9	0	HR7	HR6	HR5	HR4	HR3	0	HGFC	HR0

Figure 7-17. Cell Extraction Queue Priority Register 0 (CEQPR0) Fields

Table 7-14. CEQPR0 Field Descriptions

Bits	Name	Description
31–29, 22–9, 7–3, 0	HRn	High-priority reason n cells. If the HRn bit is set, a cell that is copied to the cell extraction queue with the extraction reason (RSN) field equal to n is actually placed in the cell extraction queue, rather than being dropped, when the cell extraction queue is above the low-priority threshold.
28–23, 8	—	Reserved, should be cleared.

Table 7-14. CEQPR0 Field Descriptions (Continued)

Bits	Name	Description
2	—	Reserved, should be cleared.
1	HGFC	High-priority GFC cells. If the HGFC bit is set, a cell that is copied to the cell extraction queue with the GFC reason (GFCR) bit set is actually placed in the cell extraction queue, rather than being dropped, when the cell extraction queue is above the low-priority threshold.

7.1.5.7 Cell Extraction Queue Priority Register 1 (CEQPR1)

The cell extraction queue priority register 1 controls the cell name-based priority of cells that are copied to the cell extraction queue.

31	26	25	24	23	22	21	20	19	18	17	16
0000_00		HN9	HN8	HN7	HN6	HN5	HN4	HN3	HN2	HN1	0
15	10	9	8	7	6	5	4	3	2	1	0
0000_00		HC9	HC8	HC7	HC6	HC5	HC4	HC3	HC2	HC1	0

Figure 7-18. Cell Extraction Queue 000 1 (CEQPR1)

Table 7-15. CEQPR1 Field Descriptions

Bits	Name	Description
31–26, 16–10	—	Reserved, should be cleared.
25–17	HNn	High-priority name n cells. If the HNn bit is set, a cell that is copied to the cell extraction queue with the extraction reason (RSN) field indicating that it was copied because an OAM copy bit is set and with the indication cell name (ICN) field equal to n is actually placed in the cell extraction queue, rather than being dropped, when the cell extraction queue is above the low-priority threshold.
9–1	HCn	High-priority copy all name n cells. If the HCn bit is set, a cell that is copied to the cell extraction queue with the extraction reason (RSN) field indicating that it was copied because the copy-all bit is set and with the indication cell name (ICN) field equal to n is actually placed in the cell extraction queue, rather than being dropped, when the cell extraction queue is above the low-priority threshold.
0	—	Reserved, should be cleared.

7.1.5.8 Ingress Insertion Leaky Bucket Register (IILB)

The ingress cell insertion rate is controlled by a leaky bucket algorithm whose parameters are defined in this register. For more information on the ingress insertion leaky bucket refer to Section 5.1.6.

31	16
IAIP	
15	0
IIBL	

Figure 7-19. Ingress Insertion Leaky Bucket Register (IILB)

Table 7-16. IILB Field Descriptions

Bits	Name	Description
31–16	IAIP	Ingress average insertion period. This field determines how often, on average, cells may be inserted to the ingress cell flow. It consists of a 12-bit integer part and a 4-bit fractional part. In other words, the IAIP is defined in units of 1/16 of a cell time.
15–0	IIBL	Ingress insertion bucket limit. The value of this field is compared with the current content of the ingress insertion bucket fill register (IIBF) to determine whether cell insertion is possible. The IIBL field is defined in terms of cell times. In other words, the IIBL field can represent a limit of up to 16 times larger than defined by the IAIP field.

7.1.5.9 Ingress Insertion Bucket Fill Register (IIBF)

This register contains the current state of the ingress insertion leaky bucket.

Figure 7-20. Ingress Insertion Bucket Fill Register (IIBF)

Table 7-17. IIBF Field Description

Bits	Name	Description
31–0	IIBF	Ingress insertion bucket fill. This field contains the current ingress insertion leaky bucket fill value. Like the IAIP, the IIBF is in units of 1/16 of a cell time, and the four least significant bits represent the fractional part. Normally, the IIBF does not need to be user-defined, but it can be used to reset the leaky bucket.

7.1.5.10 Egress Insertion Leaky Bucket Register (EILB)

The egress cell insertion rate is controlled by a leaky bucket algorithm whose parameters are defined in this register. For more information on the egress insertion leaky bucket refer to Section 5.2.3, “Inserting Cells into the Egress Flow.”

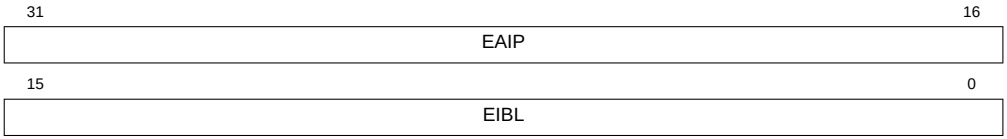


Figure 7-21. Egress Insertion Leaky Bucket Register (EILB)

Table 7-18. EILB Field Descriptions

Bits	Name	Description
31–16	EAIP	Egress average insertion period. This field determines how often, on average, cells may be inserted to the egress cell flow. It consists of a 12-bit integer part and a 4-bit fractional part. In other words, the EAIP is defined in units of 1/16 of a cell time.
15–0	EIBL	Egress insertion bucket limit. The value of this field is compared with the current content of the egress insertion bucket fill register (EIBF) to determine whether cell insertion is possible. The EIBL field is defined in terms of cell times. In other words, the EIBL field can represent a limit of up to 16 times larger than defined by the EAIP field.

7.1.5.11 Egress Insertion Bucket Fill Register (EIBF)

This register contains the current state of the egress insertion leaky bucket.

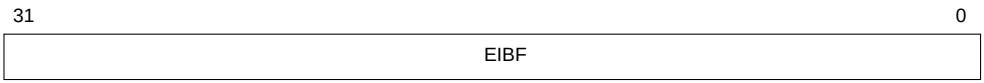


Figure 7-22. Egress Insertion Bucket Fill Register (EIBF)

Table 7-19. EIBF Field Description

Bits	Name	Description
31–0	EIBF	Egress insertion bucket fill. This field contains the current fill value of the egress insertion leaky bucket. Like the EAIP, the EIBF is in units of 1/16 of a cell time, and the four least significant bits represent the fractional part. Normally, the EIBF does not need to be user-defined, but can be used to reset the leaky bucket.

7.1.5.12 Internal Scan Control Register (ISCR)

The internal scan control register contains the highest-valued connection identifier (CI) whose context parameters table entry should be scanned. The internal scan process starts its scan from this value and scans downwards to CI = 0. The enable bits in the ISCR are used to define which cells should be inserted by the current internal scan. This register may be written at any time.

If the ISCR is written while the internal scan is active, the updated values of the enable bits take effect immediately. Clearing the ISCR disables cell insertion from the internal scan process.

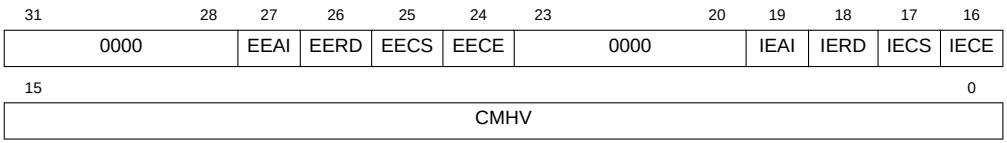


Figure 7-23. Internal Scan Register (ISCR)

Table 7-20. ISCR Field Descriptions

Bits	Name	Description
31–28	—	Reserved, should be cleared.
27	EEAI	Egress enable AIS. This bit determines whether the internal scan inserts OAM AIS cells to the egress cell flow. See Section 6.5.4.1.1. 0 Disabled 1 Enabled
26	EERD	Egress enable RDI. This bit determines whether the internal scan inserts OAM RDI cells to the egress cell flow. See Section 6.5.4.1.2 for more details. 0 Disabled 1 Enabled
25	EECS	Egress enable continuity check segment. This bit determines whether the internal scan inserts OAM segment CC cells to the egress cell flow. 0 Disabled 1 Enabled
24	EECE	Egress enable continuity check end-to-end. This bit determines whether the internal scan inserts OAM end-to-end CC cells to the egress cell flow. 0 Disabled 1 Enabled
23–20	—	Reserved, should be cleared.
19	IEAI	Ingress enable AIS. This bit determines whether the internal scan inserts OAM AIS cells to the ingress cell flow. See Section 6.5.4.1.1. 0 Disabled 1 Enabled
18	IERD	Ingress enable RDI. This bit determines whether the internal scan inserts OAM RDI cells to the ingress cell flow. See Section 6.5.4.1.2 for more details. 0 Disabled 1 Enabled
17	IECS	Ingress enable continuity check segment. This bit determines whether the internal scan inserts OAM segment CC cells to the ingress cell flow. 0 Disabled 1 Enabled
16	IECE	Ingress enable continuity check end-to-end. This bit determines whether the internal scan inserts OAM end-to-end CC cells to the ingress cell flow. 0 Disabled 1 Enabled
15–0	CMHV	Context memory highest value. The highest-valued connection identifier whose context table entry is to be scanned.

7.1.5.13 Ingress Link Registers (ILNK0–ILNK15)

The link registers contain information used by the address compression about the treatment of the physical links that the MC92520 is supporting. If only one physical link is being supported, ILNK0 is used. See Section 7.1.6.3, “Ingress PHY Configuration Register (IPHCR).”

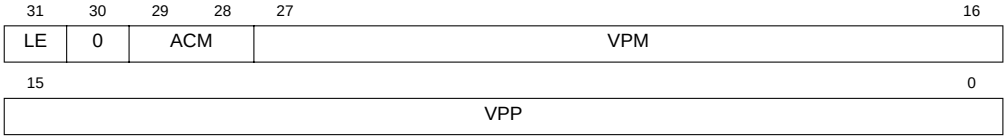


Figure 7-24. Ingress Link Register (ILNK*n*)

Table 7-21. ILNK*n* Field Descriptions

Bits	Name	Description
31	LE	Link enable. When set, this bit indicates that the physical link is enabled and polled. This bit should only be set for supported physical links. 0 Disabled 1 Enabled
30	—	Reserved, should be cleared.
29–28	ACM	Address compression method. This field determines the method used for address compression on the cells arriving from the PHY on this link. See Section 5.2.2 for more information. 00 VP and VC table lookup 01 VP table lookup only 10 External address compression without VP table lookup 11 External address compression with VP table lookup
27–16	VPM	VPI mask. This field indicates which bits of the VPI are allocated on this link. Each bit that is set in this field indicates that the corresponding bit of the VPI should be used in the VP table lookup. See Section 5.1.2 for more information. Note: At a UNI the four most significant bits of the VPM field should be zero because the VPI contains only eight bits.
15–0	VPP	VP pointer. This field contains the 16 MSBs of the external memory address of the VP table belonging to the link. Eight zeros are concatenated to the right of this field to construct the actual external memory address.

7.1.5.14 Egress Link Enable Register (ELER)

The egress link enable register is used to enable or disable transmission to each physical link individually. Each bit that is set enables transmission to the corresponding link.

NOTE:

All egress link enable bits are readable and writable through the egress link enable register (ELER[LE*n*]) and the egress link registers (ELNK*n*[LE]) described below.

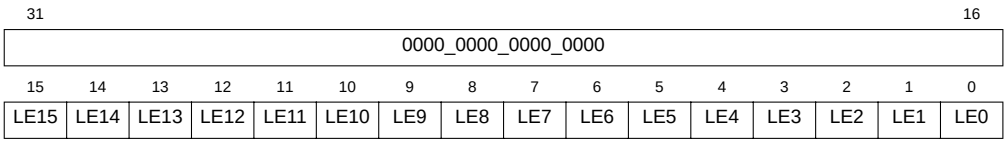


Figure 7-25. Egress Link Enable Register (ELER)

Table 7-22. ELER Field Descriptions

Bits	Name	Description
31–16	—	Reserved, should be cleared.
15–0	LE0–LE15	Link enable. Each LE bit refers to one of the physical links supported by the MC92520. The LE bit, when set, enables transmission to the link.

7.1.5.15 Egress Link Registers (ELNK0–ELNK15)

The egress link registers contain information that control the use of the physical links attached to the egress side of the MC92520. If only one physical link is needed, ELNK0 is used.

NOTE:

All egress link enable bits are readable and writable through the the egress link registers (ELNK*n*[LE]) and egress link enable register (ELER[LE*n*]) described above.

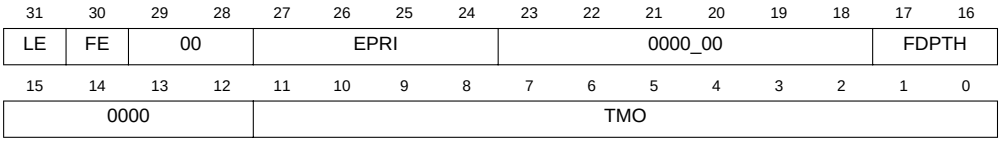


Figure 7-26. Egress Link Register (ELNK*n*)

Table 7-23. ELNK*n* Field Descriptions

Bits	Name	Description
31	LE	Link enable. When set, this bit indicates that the physical link is enabled and polled. This bit should only be set for physical links that are physically present. If a cell destined to a disabled link is encountered during address translation, the cell is extracted. If a link is disabled while not all cells have been transferred to the target PHY (that is, address translation was already completed), processing and transfer of such cells continues without MC92520 error indications. 0 Disabled 1 Enabled
30	FE	Automatic flush enable. This bit indicates whether the FIFO associated with this link should be automatically flushed, if a HOL blocking condition is detected. The functionality is only available if HOL blocking prevention is enabled via the EPHP bit in the egress PHY configuration register (EPHCR). 0 Automatic FIFO flushing is disabled 1 Automatic FIFO flushing is enabled
29–28	—	Reserved, should be cleared.
27–24	EPRI	Egress PHY priority. This field indicates the relative cell transfer priority of the associated PHY in relation to other PHYs when operating in FIFO per-PHY mode. The priority values may range from 0 to 15, 0 being the lowest priority. The same priority can be selected for more than one PHY. The selection of an egress PHY priority value is only available if per-PHY FIFOs are enabled via the EPHCR(EPMF) bit.
23–18	—	Reserved, should be cleared.

Table 7-23. ELNK*n* Field Descriptions (Continued)

Bits	Name	Description
17–16	FDPTH	Per-PHY FIFO depth. This field indicates the per-PHY FIFO depth. This functionality is only available if the per-PHY FIFOs are enabled via the EPHCR(EPMF) bit. 01 FIFO depth is 1 cell 10 FIFO depth is 2 cells 11 FIFO depth is 3 cells. This is the default value at reset.
15–12	—	Reserved, should be cleared.
11–0	TMO	Cell transfer delay time-out. This field indicates the per-PHY cell transfer delay time-out that is used to detect a HOL blocking condition. This functionality is only available if HOL blocking prevention is enabled via the EPHP bit in the egress PHY configuration register (EPHCR). The time-out threshold is specified in ATMC cell times, namely, 64 ZCLKs.

7.1.5.16 Ingress Billing Counters Table Pointer Register (IBCTP)

This register contains the pointer to the first word of the ingress billing counters table. The pointer is in units of 256 bytes. This pointer register may be written in operate mode to allow the processor to relocate the ingress billing counters table. In this way the counters of all the connections can be frozen simultaneously (at 15-minute intervals) and read at leisure.

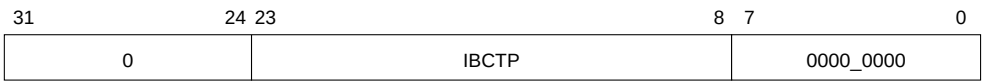


Figure 7-27. Ingress Billing Counters Table Pointer (IBCTP) Fields

7.1.5.17 Egress Billing Counters Table Pointer Register (EBCTP)

This register contains the pointer to the first word of the egress billing counters table. The pointer is in units of 256 bytes. This pointer register may be written in operate mode to allow the processor to relocate the egress billing counters table. In this way the counters of all the connections can be frozen simultaneously (at 15-minute intervals) and read at leisure.

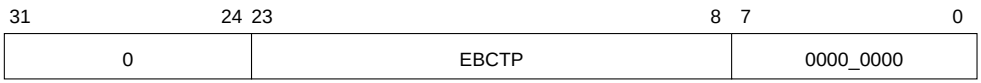


Figure 7-28. Egress Billing Counters Table Pointer (EBCTP) Fields

7.1.5.18 Policing Counters Table Pointer Register (PCTP)

This register contains the pointer to the first word of the policing counters table. The pointer is in units of 256 bytes. This pointer register may be written in operate mode to allow the processor to relocate the policing counters table. In this way the counters of all the connections can be frozen simultaneously (at 15 minute intervals) and read at leisure.

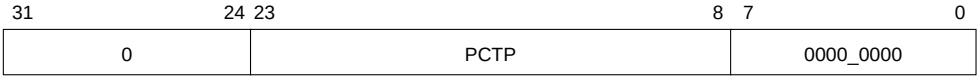


Figure 7-29. Policing Counters Table Pointer (PCTP) Fields

7.1.5.19 Cell Time Register (CLTMR)

This register contains a count of the cell processing times. It is a 32-bit counter that is initialized to zero on reset and counts cell processing times when the MC92520 is in operate mode.

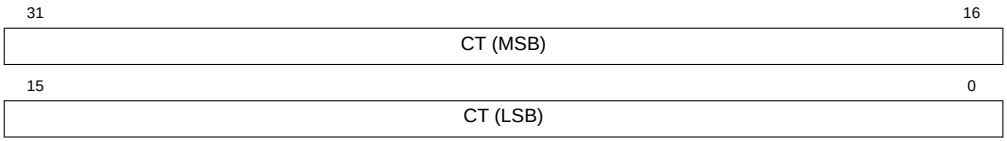


Figure 7-30. CLTMR Fields

7.1.5.20 Ingress Processing Control Register (IPLR)

This register defines the MC92520 ingress processing parameters that can be changed in operate mode.

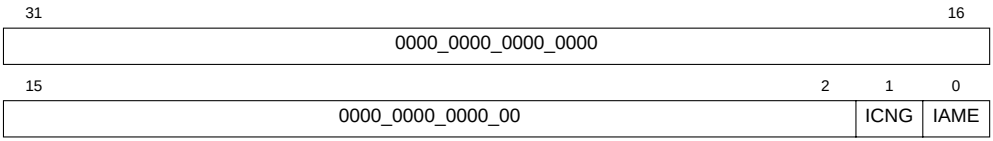


Figure 7-31. Ingress Processing Control Register (IPLR)

Table 7-24. IPLR Field Descriptions

Bits	Name	Description
31–2	—	Reserved, should be cleared.
1	ICNG	Global ingress congestion notification. This bit notifies the MC92520 whether there is congestion in the ingress flow. 0 No ingress congestion. 1 Ingress congestion. The MC92520 performs selective discard according to per-connection ingress selective discard operation mode (ISDM) field. See Section 6.6 and Section 7.2.16 for more information.
0	IAME	Global ingress ABR mark enable. This bit, when set, indicates that current egress flow status implies that the MC92520 should perform RM cell and/or EFCI marking if enabled. See Section 6.3.3.1 for more information.

7.1.5.21 Egress Processing Control Register (EPLR)

This register defines the MC92520 ingress processing parameters that can be changed in operate mode.

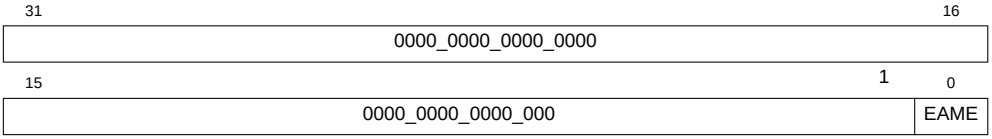


Figure 7-32. Egress Processing Control Register (EPLR)

Table 7-25. EPLR Field Descriptions

Bits	Name	Description
31–1	—	Reserved, should be cleared.
0	EAME	Global egress ABR mark enable. This bit, when set, indicates that current egress flow status implies that the MC92520 should perform RM cell and/or EFCI marking if enabled. See Section 6.3.3.2 for more information.

7.1.5.22 Indirect External Memory Access Address Register (IAAR)

This register contains the address, width and busy bit for accessing the MC92520 external memory or the external memory device. Refer to Section 3.2.2.2 for details.

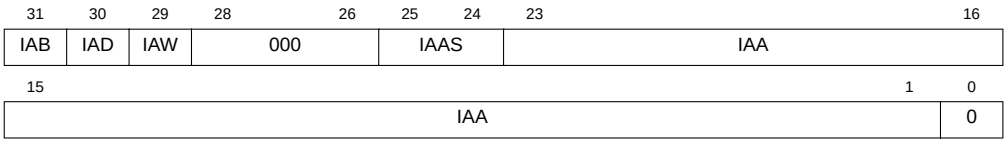


Figure 7-33. Indirect External Memory Access Address Register (IAAR)

Table 7-26. IAAR Field Descriptions

Bits	Name	Description
31	IAB	Indirect external memory access busy. This bit indicates that indirect external memory access mechanism is busy. 0 Indirect access mechanism is free and therefore indirect external memory access data register can be accessed 1 Indirect access mechanism is busy and therefore indirect access data register should not be accessed
30	IAD	Indirect external memory access direction. This bit indicates indirect access direction. 0 Indirect write access 1 Indirect read access
29	IAW	Indirect external memory access size. This bit indicates the size of the access. 0 32 bits 1 16 bits
28–26	—	Reserved, should be cleared.

Table 7-26. IAAR Field Descriptions (Continued)

Bits	Name	Description
25–24	IAAS	Indirect external memory access address space. This field indicates the accessed address space. 00 Reserved 01 Reserved 10 Non-destructive external memory 11 Reserved
23–1	IAA	Indirect external memory access address. This field indicates bits 23 through 1 of the address within the address space specified in the indirect external memory access address space (IAAS) field.

7.1.5.23 Indirect External Memory Access Data Register (IADR)

This register contains the data that should be written to the external memory in case of an indirect write access or the data that was last read from external memory in case of an indirect read access. Refer to Section 3.2.2.2 for details.

7.1.5.24 Cell Arrival Period Multiplier Registers 0 - 63 (CAPM*nn*)

All 64 CAPM*nn* registers are used in combination with the GFR fair cell rate (FCR) administration. Each individual CAPM*nn* register holds an 8-bit value, which is used to dynamically recalculate a target cell arrival period (CAP) for each VC in one of up to 64 groups of VCs. A CAPM index (CAPMI) field in the first leaky bucket parameter word of GFR VCs indicates which CAPM belongs to which VC. For more information see Section 6.3.1.7, “FCR Administration.”

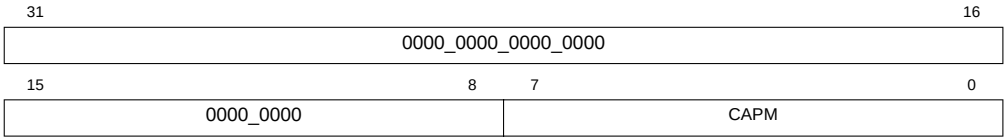


Figure 7-34. Cell Arrival Period Multiplier (CAPM*nn*) Fields

The 8-bit CAPM field is used to form a GFR VC’s effective CAP by multiplying the FCR CAP field from the leaky bucket parameter word with CAPM and dividing the result by 256. A CAPM value of 0 is interpreted as 256; that is, the resulting CAP multiplier is 1. Thus the range of $CAP_{effective}$ is:

$$CAP * 1 / 256 <= CAP_{effective} <= CAP * 256 / 256$$

7.1.6 Configuration Registers

The configuration registers are used to define the MC92520 configuration, and may be read by the processor in either setup mode or operate mode. These registers may be written by the processor only in setup mode.

7.1.6.1 PLL Range Register (PLLRR)

This register defines the MC92520 PLL input clock frequency range provided through the ACLK pin. If the MC92520 is used in PLL bypass mode, the content of this register is ignored.

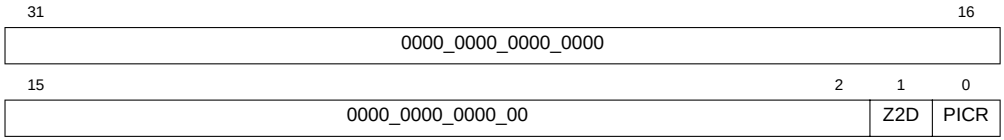


Figure 7-35. PLL Range Register (PLLRR)

Table 7-27. PLLRR Field Descriptions

Bits	Name	Description
31–2	—	Reserved, should be cleared.
1	Z2D	ZCLKO_2 disable. If set, the ZCLKO_2 output is disabled (tri-stated). This can be used as a power-saving feature in systems where only the ZCLKO output is used.
0	PICR	PLL input clock range. The bit selects a high- or low-input clock frequency range. 0 ACLK is in 25 – 50 MHz range (standard MC92520 mode). 1 ACLK is in 12.5 – 25 MHz range (low-power MC92520 mode).

7.1.6.2 Microprocessor Configuration Register (MPCONR)

This register defines the processor interface configuration parameters.

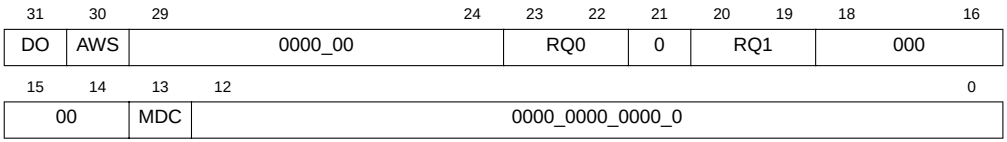


Figure 7-36. Microprocessor Configuration Register (MPCONR)

Table 7-28. MPCONR Field Descriptions

Bits	Name	Description
31	DO	Data order. This bit defines the order of the bytes on the data bus inside the 32-bit data structures of the cell insertion payload registers (CIR4–CIR15) and the cell extraction payload registers (CER4–CER15). It also defines the order of the records in the external memory tables that contain 16-bit records. 0 Most significant byte first (big endian - Motorola or IBM style) 1 Least significant byte first (little endian - Intel or Digital style)
30	AWS	Assume write strobe. If this bit is 0 (default), $\overline{\text{MDTACK}}$ does not occur on a write cycle until the MCLK rising edge after the clock edge in which MWSH* and/or MWSL* are first asserted. If this bit is 1, it is assumed that MWSH* and/or MWSL* occur on the first clock after the cycle begins, and $\overline{\text{MDTACK}}$ may occur as early as the first clock after the cycle begins.
29–24	—	Reserved, should be cleared.

Table 7-28. MPCONR Field Descriptions (Continued)

Bits	Name	Description
23–22	RQ0	$\overline{\text{MREQ0}}$ signal functionality. This field defines the functionality of the MP request 0 ($\overline{\text{MREQ0}}$) signal. See Section 4.5.2 for details. 00 Cell in request 01 Cell in request 10 Cell out request 11 Reserved
21	—	Reserved, should be cleared.
20–19	RQ1	$\overline{\text{MREQ1}}$ signal functionality. This field defines the functionality of the MP request 1 ($\overline{\text{MREQ1}}$) signal. See Section 4.5.2 for details. 00 Cell out request 01 Cell in request 10 Cell out request 11 Reserved
18–14	—	Reserved, should be cleared.
13	MDC	$\overline{\text{MDTACKn}}$ drive control. This bit determines which MDTACK signals are driven. 0 $\overline{\text{MDTACK0}}$ is driven and $\overline{\text{MDTACK1}}$ is not driven. 1 Both $\overline{\text{MDTACK0}}$ and $\overline{\text{MDTACK1}}$ are driven.
12–0	—	Reserved, should be cleared.

7.1.6.3 Ingress PHY Configuration Register (IPHCR)

This register controls the operation of the ingress PHY interface block. See Section 4.2 for more details.

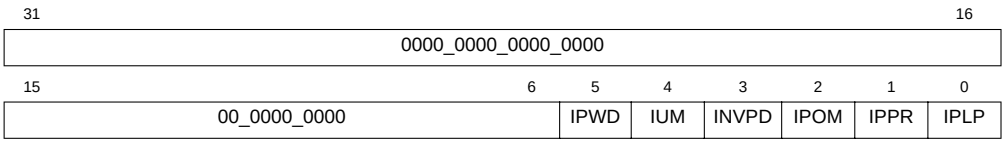


Figure 7-37. Ingress PHY Configuration Register (IPHCR)

Table 7-29. IPHCR Field Descriptions

Bits	Name	Description
31–6	—	Reserved, should be cleared.
5	IPWD	Ingress PHY wide data path. This bit controls the width of the data path. 0 8-bit data path using RX_DATA[7:0], RX_DATA[15:8] should be tied to ground. 1 16-bit data path using RX_DATA[15:0].
4	IUM	Ingress UTOPIA multi-PHY. This bit determines whether the interface operates in UTOPIA single-PHY or multi-PHY mode. See Section 4.2 for more information. 0 UTOPIA single-PHY operation. 1 UTOPIA multi-PHY operation.
3	INVPD	Invalid pattern discard. The INVPD bit determines the treatment of cells received with the “invalid” pattern (VPI/VCI0, CLP1) in the header. 0 Cells with the “invalid” pattern are removed from the cell flow and copied to the cell extraction queue. 1 Cells with the “invalid” pattern are discarded at the PHY interface in the same manner as unassigned cells and do not occupy a cell processing slot.

Table 7-29. IPHCR Field Descriptions (Continued)

Bits	Name	Description
2	IPOM	Ingress PHY operation mode. The IPOM bit determines whether the ingress PHY interface operates with an octet- or word-level handshake or a cell-level handshake. 0 The PHY interface uses octet- or word-level handshake. 1 The PHY interface uses cell-level handshake.
1	IPPR	Ingress PHY parity enable. This bit defines whether there is parity checking on the ingress PHY interface. If a parity error is detected on the cell header by the ingress PHY interface, the cell that contains the parity error is discarded, and the ingress PHY continues its operation without any effect on the next cell. 0 Parity checking is disabled. 1 Parity checking is enabled.
0	IPLP	Ingress payload parity enable. This bit determines whether a parity error that is detected on the payload of a cell arriving from the PHY should cause the cell to be removed from the cell flow. Parity checking is enabled by ingress PHY parity enable (IPPR). 0 A parity error detected on the payload of a cell does not cause the cell to be removed. 1 A parity error detected on the payload of a cell causes the cell to be removed from the cell flow and copied to the cell extraction queue.

Table 7-30. Parity Checking at Ingress PHY Interface

IPPR	IPLP	Action When Parity Error is Detected on a Header Byte	Action When Parity Error is Detected on a Payload Byte
0	0	Ignore	Ignore
0	1	Ignore	Ignore
1	0	Discard cell at the PHY interface	Ignore
1	1	Discard cell at the PHY interface	Remove cell and copy to the cell extraction queue

7.1.6.4 Egress PHY Configuration Register (EPHCR)

This register controls the operation of the egress PHY interface block.

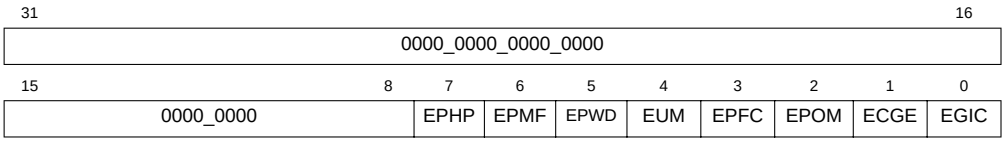


Figure 7-38. Egress PHY Configuration Register (EPHCR)

Table 7-31. EPHCR Field Descriptions

Bits	Name	Description
31–8	—	Reserved, should be cleared.
7	EPHP	Egress PHY HOL blocking prevention (EPHP)—This bit determines whether HOL blocking prevention and reporting is enabled or disabled. If enabled, HOL blocking reporting requires the configuration of a cell transfer delay threshold and HOL blocking prevention requires the additional configuration of an automatic FIFO flush request bit in the egress link register (ELNKn). The EPHP bit may only be set in combination with a set egress PHY multiple FIFOs (EPMF) bit (see below). 0 HOL blocking prevention is disabled. 1 HOL blocking prevention is enabled.
6	EPMF	Egress PHY multiple FIFOs. This bit determines whether the egress PHY interface uses a single FIFO or multiple, per-PHY FIFOs. If a single FIFO is used, the EPFC bit (see below) defines the FIFO depth for the single FIFO. If per-PHY FIFOs are used, EPFC is ignored and the FDPTH field of the egress link registers (ELNKn) define the per-PHY FIFO depths. 0 Egress interface is using a single FIFO. 1 Egress interface is using a FIFO per PHY.
5	EPWD	Egress PHY wide data path (EPWD)—This bit determines the width of the data path. To prevent the generation of corrupted idle or unassigned cells, changes to EPWD are ignored after the ECGE bit has been set. See Section 4.2.2 for more information. 0 8-bit data path using TX_DATA[7:0], TX_DATA[15:8] is 0 and not valid. 1 16-bit data path using TX_DATA[15:0].
4	EUM	Egress UTOPIA multi-PHY. This bit determines whether the interface operates in UTOPIA single-PHY or multi-PHY mode. See Section 4.2.2 for more information. 0 UTOPIA single-PHY operation. 1 UTOPIA multi-PHY operation.
3	EPFC	Egress PHY interface FIFO control. This bit determines the size of the egress PHY interface FIFO if the MC92520 is configured for single-FIFO operation (EPMF bit is cleared). 0 The egress PHY interface has a 4-cell FIFO. 1 The egress PHY interface has a 2-cell FIFO. If the PHY device has at least a 4-cell FIFO, EPFC may be 1 to reduce the delay of the egress cell flow. If the PHY device has only a small FIFO, EPFC should be 0.
2	EPOM	Egress PHY operation mode. The EPOM bit determines whether the egress PHY interface operates in an octet-based mode or a cell-based mode. See Section 4.2.2. 0 The PHY interface is octet-based. 1 The PHY interface is cell-based.
1	ECGE	Egress cell generation enable. This bit determines whether the egress PHY interface block generates unassigned or idle cells when there are no cells for transmission. The cell type generated by the egress PHY is defined by egress generate idle cells (EGIC). 0 Do not generate unassigned or idle cells 1 Generate unassigned or idle cells If the MC92520 is configured for multi-PHY operation, ECGE should not be set.
0	EGIC	Egress generate idle cells (EGIC)—This bit determines the type of cells generated by the egress PHY interface. The cell generation is enabled by egress cell generation enable (ECGE). 0 Unassigned (ATM layer) cells are generated 1 Idle (PHY layer) cells are generated

Table 7-32. Cell Generation at Egress PHY Interface

ECGE	EGIC	Cell Generation
0	0	None
0	1	None

Table 7-32. Cell Generation at Egress PHY Interface

ECGE	EGIC	Cell Generation
1	0	Unassigned cells
1	1	Idle cells

7.1.6.5 Ingress Switch Interface Configuration Register (ISWCR)

This register controls the operation of the ingress switch interface block.

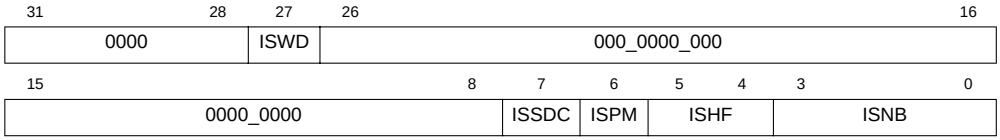


Figure 7-39. Ingress Switch Interface Configuration Register (ISWCR)

Table 7-33. ISWCR Field Descriptions

Bits	Name	Description
31–28	—	Reserved, should be cleared.
27	ISWD	Ingress switch wide data. This bit determines whether the ingress switch interface block uses an 8-bit or a 16-bit wide data path. See Section 4.3.1. 0 8-bit wide data using SRXDATA[7:0], SRXDATA[15:8] is 0 and not valid. 1 16-bit wide data using SRXDATA[15:0].
26–8	—	Reserved, should be cleared.
7	ISSDC	Ingress switch SRXDATA driver control. This bit determines whether the outputs of the ingress switch interface block are tri-stated when $\overline{\text{SRXENB}}$ is deasserted 0 SRXDATA, SRXPRTY, and SRXSOC are driven only if $\overline{\text{SRXENB}}$ is asserted on the previous clock. 1 SRXDATA, SRXPRTY, and SRXSOC are always driven.
6	ISPM	Ingress switch parity mode. This bit selects the ISWI parity mode, odd or even. See Section 4.3.1. 0 Odd parity is generated over the data transferred to the switch 1 Even parity is generated over the data transferred to the switch

Table 7-33. ISWCR Field Descriptions (Continued)

Bits	Name	Description
5–4	ISHF	Ingress switch HEC field. The ISHF field determines if the HEC octet (8-bit mode) or UDF word (16-bit mode) is inserted before the payload of cells transferred to the switch and what data it should contain. 00 No data is inserted 01 Reserved 10 Depending on the data path width, a zero HEC octet or UDF word is inserted. 11 In 8-bit mode, the HEC octet is inserted and contains the MSB of the most-significant long word of the switch parameters. In 16-bit mode, the UDF word is inserted and contains the MSW of the most-significant long word of the switch parameters.
3–0	ISNB	Ingress switch number of bytes. The ISNB field determines the size of the data structure transferred to the switch. Note that for 16-bit operation (ISWD1), this field must be a value that results in an even number of bytes. 0000 64 bytes per cell are transferred to the switch 0001 Reserved 0010 Reserved 0011 Reserved 0100 52 bytes per cell are transferred to the switch 0101 53 bytes per cell are transferred to the switch 0110 54 bytes per cell are transferred to the switch 0111 55 bytes per cell are transferred to the switch 1000 56 bytes per cell are transferred to the switch 1001 57 bytes per cell are transferred to the switch 1010 58 bytes per cell are transferred to the switch 1011 59 bytes per cell are transferred to the switch 1100 60 bytes per cell are transferred to the switch 1101 61 bytes per cell are transferred to the switch 1110 62 bytes per cell are transferred to the switch 1111 63 bytes per cell are transferred to the switch

7.1.6.6 Egress Switch Interface Configuration Register (ESWCR)

This register determines the operation of the egress switch interface block.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EIAS	EEAS	VPS	0	ESWD	IHAF	ESUM	ESFC	EFE	MTSE	EATD	ELNS	ESPC	ESPR	EPLP	ESHF

15	11	10	9	5	4	0
ESNB		EVPZ	IMSB		ILSB	

Figure 7-40. Egress Switch Interface Configuration Register (ESWCR)

Table 7-34. ESWCR Field Descriptions

Bits	Name	Description
31	EIAS	Global IFS enable. This bit enables the MC92520 to use overhead ingress flow status (IFS) bit in the egress switch overhead. See Section 6.4.2.1 for details. 0 The overhead ingress flow status (IFS) bit is not defined in the egress overhead fields so it cannot trigger ABR cell marking. 1 The overhead ingress flow status (IFS) bit is defined in the egress overhead fields and is used by the MC92520 for marking cells as part of relative rate ABR.
30	EEAS	Global EFS enable. This bit enables the MC92520 to use overhead egress flow status (EFS) bit in the egress switch overhead. See Section 6.4.2.2 for details. 0 The overhead egress flow status (EFS) bit is not defined in the egress overhead fields, so it cannot trigger ABR cell marking. 1 The overhead egress flow status (EFS) bit is defined in the egress overhead fields and is used by the MC92520 for marking cells as part of relative rate ABR.
29	VPS	VPI size in ECI on header mode. This bit determines the size of the VPI field for ECI on header mode' (IHAF=1). See Section 5.2.1 for details. 0 VPI size is 12 bits 1 VPI size is 8 bits
28	—	Reserved, should be cleared.
27	ESWD	Egress switch wide data. This bit determines whether the egress switch interface block uses an 8- or a 16-bit wide data path. See Section 4.3.2 for more information. 0 8-bit wide data using STXDATA[7:0], STXDATA[15:8] should be tied to ground. 1 16-bit wide data using STXDATA[15:0].
26	IHAF	Identifier in header address fields. The IHAF bit, when set, indicates that the ECI is located in the header address fields of the cell structure as follows: If the EVPZ bit is set, the VPI field of the header is used as the ECI. If the EVPZ bit is clear, the VPI field is only used as ECI if it is non-zero. If the EVPZ bit is clear and the VPI is zero, the VCI field is used as the ECI. When IHAF is set, the identifier most-significant byte (IMSB) and identifier least-significant byte (ILSB) fields are not used.
25	ESUM	Egress switch UTOPIA multi-PHY mode. The ESUM bit determines whether the UTOPIA interface acts like a single PHY or a multi-PHY device. In multi-PHY mode, the PHY port base address and PHY port mask fields of the egress switch configuration register 1 (ESWCR1) must be also configured. Also, the depth of the effective per-PHY FIFO is independent of the switch-side FIFO configuration in multi-PHY mode, namely, the setting of ESFC is ignored if ESUM is set to 1. For more information on the configuration and operation, refer to Section 3.4.2.2.1. 0 Single-PHY interface 1 Multi-PHY interface
24	ESFC	Egress switch FIFO control. The ESFC bit determines the size of the switch-side FIFO in the egress switch interface block for single-PHY interface configurations. The value of ESFC is ignored if the ESUM bit (see above) is set. Refer to Section 3.4.2.2.1. 0 6-cell FIFO 1 4-cell FIFO Normally, use a 6-cell FIFO. For applications in which it is necessary to reduce maximum cell delay through the MC92520, a 4-cell FIFO might be appropriate with certain switch interface architectures.
23	EFE	EFCI enable. This bit enables the MC92520 to set the middle bit of the PTI in the header of an incoming cell if the EFCI bit received from the switch is set. 0 The EFCI bit is not defined in the overhead fields so EFCI0 is provided to the cell processing block. 1 The EFCI bit from the overhead fields causes the PTI1 bit of the cell header to be set.

Table 7-34. ESWCR Field Descriptions (Continued)

Bits	Name	Description
22	MTSE	Multicast translation table section enable. This bit determines whether the MTTS field is defined in the overhead fields received from the switch. 0 The MTTS field is not defined in the overhead fields, so MTTS0000 is provided to the cell processing block. 1 The MTTS field provided to the cell processing block is taken from the location in the overhead fields defined by the MTTS byte location (MTBY) and MTTS bit (MTBI) location fields.
21	EATD	Egress address translation disable. This bit determines whether the MC92520 performs address translation on the cells arriving from the switch. See Table 7-35. 0 The MC92520 performs address translation in the egress direction. 1 The MC92520 does not perform address translation in the egress direction, and the cell header is transferred transparently to the PHY.
20	ELNS	Egress link number selection. This bit selects the source for the link number when egress address translation disable (EATD) is set. This field is ignored if ESUM1 (multi-PHY mode). See Table 7-35. 0 The link number is zero. 1 The link number is taken from the MTTS field whose location is determined by the MTTS byte (MTBY) and MTTS bit (MTBI) location fields.
19	ESPC	Egress switch parity control. This bit selects the egress switch parity mode, odd or even. See Section 4.3.2.1. 0 Odd parity is expected 1 Even parity is expected
18	ESPR	Egress switch parity enable. This bit defines whether there is parity checking on the egress switch interface. SeeTable 7-36. 0 Parity checking is disabled. 1 Parity checking is enabled.
17	EPLP	Egress payload parity enable. This bit determines whether a parity error that is detected on the payload of a cell arriving from the switch should cause the cell to be removed from the cell flow. Parity checking is enabled by the egress switch parity enable (ESPR) bit. See Section 4.3.2.1 and Table 7-36. 0 A parity error detected on the payload of a cell does not cause the cell to be dropped. 1 A parity error detected on the payload of a cell causes the cell to be dropped.
16	ESHF	Egress switch HEC field. The ESHF bit determines if the cells that are transferred from the switch contain the HEC octet (8-bit mode) or UDF word (16-bit mode). 0 The switch does not transfer the HEC octet or UDF word. 1 The switch transfers the HEC octet/UDF word. This octet or UDF word is discarded by the ATMC.

Table 7-34. ESWCR Field Descriptions (Continued)

Bits	Name	Description
15–11	ESNB	Egress switch number of bytes. The ESNB field determines the number of bytes in the data structure received from the switch. Note that for 16-bit operation (ESWD1), this field must be a value that results in an even number of bytes. 00000 65 bytes per cell are received from the switch 00001 66 bytes per cell are received from the switch 00010 67 bytes per cell are received from the switch 00011 68 bytes per cell are received from the switch 00100 69 bytes per cell are received from the switch 00101 70 bytes per cell are received from the switch 00110 71 bytes per cell are received from the switch 00111 72 bytes per cell are received from the switch 01000 73 bytes per cell are received from the switch 01001 74 bytes per cell are received from the switch 01010 75 bytes per cell are received from the switch 01011 76 bytes per cell are received from the switch 01100 77 bytes per cell are received from the switch 01101 78 bytes per cell are received from the switch 01110 79 bytes per cell are received from the switch 01111 80 bytes per cell are received from the switch 10000 Reserved 10001 Reserved 10010 Reserved 10011 52 bytes per cell are received from the switch 10100 53 bytes per cell are received from the switch 10101 54 bytes per cell are received from the switch 10110 55 bytes per cell are received from the switch 10111 56 bytes per cell are received from the switch 11000 57 bytes per cell are received from the switch 11001 58 bytes per cell are received from the switch 11010 59 bytes per cell are received from the switch 11011 60 bytes per cell are received from the switch 11100 61 bytes per cell are received from the switch 11101 62 bytes per cell are received from the switch 11110 63 bytes per cell are received from the switch 11111 64 bytes per cell are received from the switch
10	EVPZ	Egress VPI zero valid. This bit determines whether a zero value extracted from the VPI field for ECI on header mode' (IHAF=1) should be used as a valid value or used as an indication to extract the ECI from the VCI field. See Section 5.2.1 for details. 0 Extract ECI from VCI field if extracted VPI value is 0. 1 Use extracted VPI value of 0 as ECI.
9–5	IMSB	Identifier most-significant byte. The IMSB field contains the byte number of the most-significant byte of the MI/ECI field within the switch data structure. The byte on which STXSOC is asserted is byte number 0.
4–0	ILSB	Identifier least-significant byte. The ILSB field contains the byte number of the least-significant byte of the MI/ECI field within the switch data structure. The byte on which STXSOC is asserted is byte number 0.

Table 7-35. Destination Link Number

EATD	ELNS	Destination Link Number
0	0	Taken from the egress translation address word in the context parameters table
0	1	Taken from the egress translation address word in the context parameters table
1	0	0
1	1	Taken from MTTS field

Table 7-36. Parity Checking at Egress Switch Interface

ESPR	EPLP	Action When Parity Error is Detected on a Header/Overhead/HEC Byte	Action When Parity Error is Detected on a Payload Byte
0	0	Ignore	Ignore
0	1	Ignore	Ignore
1	0	Discard cell at the switch interface	Ignore
1	1	Discard cell at the switch interface	Remove cell and copy to the cell extraction queue

7.1.6.7 Egress Switch Interface Configuration Register 1 (ESWCR1)

This register determines the multi-PHY operation of the egress switch interface block. For more information on the configuration and use of ESWCR1, see Section 4.3.2, “Transmit Interface (Egress).”

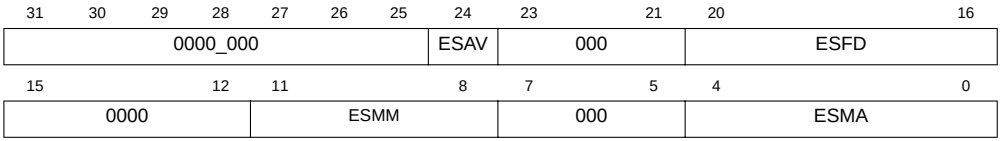


Figure 7-41. Egress Switch Interface Configuration Register 1(ESWCR1)

Table 7-37. ESWCR1 Field Descriptions

Bits	Name	Description
31–25	—	Reserved, should be cleared.
24	ESAV	Egress switch multi-PHY address valid pin enable. This bit determines whether the STXAVALID pin is used to qualify valid multi-PHY addresses. 0 The STXAVALID pin is ignored (standard UTOPIA Level 2 mode). 1 Any STXADDR[4:0] is ignored while STXAVALID is not asserted and considered a valid address when STXAVALID is asserted.
20–16	ESFD	Egress switch multi-PHY FIFO depth. This field defines the depth of the switch-side FIFO if a multi-PHY interface is selected. Valid configuration values range from 1 to 16. The default value of 0 indicates 16.
15–10	—	Reserved, should be cleared.
11–8	ESMM	Egress switch multi-PHY port address mask. This field defines a PHY port address mask. This mask and the PHY port base address defined below are used to detect whether the PHY address provided by a switch-side device is selecting an MC92520 PHY port.

Table 7-37. ESWCR1 Field Descriptions

Bits	Name	Description
7–5	—	Reserved, should be cleared.
4–0	ESMA	Egress switch multi-PHY port base address. This field defines a PHY port base address. A PHY address provided by a switch-side device and matching ESMA is associated with the MC92520 PHY port addressed by link id 0.

7.1.6.8 Egress Switch Overhead Information Register 0 (ESOIR0)

This register determines the location of the overhead information in the data structure received from the switch.

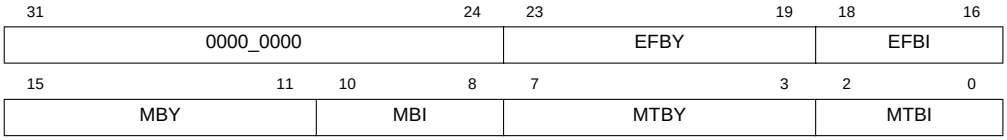


Figure 7-42. Egress Switch Overhead Information Register 0 (ESOIR0)

Table 7-38. ESOIR0 Field Descriptions

Bits	Name	Description
31–24	—	Reserved, should be cleared.
23–19	EFBY	EFCI byte. The EFBY field contains the byte number of the switch data structure in which the EFCI can be found: overhead, header, and HEC bytes. The byte on which STXSOC is asserted is byte number 0.
18–16	EFBI	EFCI bit. The EFBI field contains the number of the EFCI bit within the byte specified by the EFCI byte (EFBY) location field. The most-significant bit is number 7, and the least-significant bit is number 0.
15–11	MBY	M byte. The MBY field contains the byte number of the switch data structure in which the M bit can be found. The byte on which STXSOC is asserted is byte number 0.
10–8	MBI	M bit. The MBI field contains the number of the M bit within the byte specified by the M byte (MBY) location field. The most-significant bit is number 7 and the least-significant bit is number 0.

Table 7-38. ESOIR0 Field Descriptions (Continued)

Bits	Name	Description
7–3	MTBY	MTTS byte. The MTBY field contains the byte number of the switch data structure in which the MTTS field can be found. The byte on which STXSOC is asserted is byte number 0.
2–0	MTBI	MTTS bit. This field indicates the location of the MTTS field within the byte specified by MTTS byte (MTBY) location field. 0 MTTS equals the value that resides in bits 7:5 of the byte pointed to by MTTS byte (MTBY) location field. 1 MTTS equals the value that resides in bits 7:6 of the byte pointed to by MTTS byte (MTBY) location field. 2 MTTS equals the value that resides in bit 7 of the byte pointed to by MTTS byte (MTBY) location field. 3 MTTS equals the value that resides in bits 3:0 of the byte pointed to by MTTS byte (MTBY) location field. 4 MTTS equals the value that resides in bits 4:1 of the byte pointed to by MTTS byte (MTBY) location field. 5 MTTS equals the value that resides in bits 5:2 of the byte pointed to by MTTS byte (MTBY) location field. 6 MTTS equals the value that resides in bits 6:3 of the byte pointed to by MTTS byte (MTBY) location field. 7 MTTS equals the value that resides in bits 7:4 of the byte pointed to by MTTS byte (MTBY) location field.

7.1.6.9 Egress Switch Overhead Information Register 1 (ESOIR1)

This register determines the location of the overhead information in the data structure received from the switch. The register has the following structure:

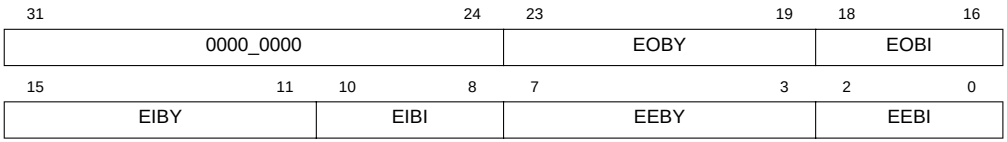


Figure 7-43. Egress Switch Overhead Information Register 1 (ESOIR1) Fields

Table 7-39. ESOIR1 Field Descriptions

Bits	Name	Description
31–24	—	Reserved, should be cleared.
23–19	EOBY	EOCLP byte. This field contains the byte number of the switch data structure in which the egress overhead CLP (EOCLP) bit can be found (overhead, header, and HEC bytes). The byte on which STXSOC is asserted is byte number 0. See Section 6.2.1 and Section 6.2.2 for details.
18–16	EOBI	EOCLP bit location. This field contains the number of the egress overhead CLP (EOCLP) bit within the byte specified by the EOCLP byte (EOBY) location field. The most-significant bit is 7, and the least-significant bit is 0. See Section 6.2.1 and Section 6.2.2 for details.
15–11	EIBY	IFS byte location. This field contains the byte number of the switch data structure in which the overhead ingress flow status (IFS) bit can be found (overhead, header, and HEC bytes). The byte on which STXSOC is asserted is byte number 0. See Section 6.4.2.1 for details.
10–8	EIBI	IFS bit location. This field contains the number of the overhead ingress flow status (IFS) bit within the byte specified by the IFS byte (EIBY) location field. The most-significant bit is 7, and the least-significant bit is 0. See Section 6.4.2.1 for details.

Table 7-39. ESOIR1 Field Descriptions

Bits	Name	Description
7–3	EEBY	EFS byte location. This field contains the byte number of the switch data structure in which the overhead egress flow status (EFS) bit can be found (overhead, header, and HEC bytes). The byte on which STXSOC is asserted is byte number 0.
2–0	EEBI	EFS bit location (EEBI)—This field contains the number of the overhead egress flow status (EFS) bit within the byte specified by the EFS byte (EEBY) location field. The most-significant bit is 7, and the least-significant bit is 0.

7.1.6.10 UNI Register (UNIR)

The UNI register determines whether each of the links is treated as a UNI or as an NNI.

31															16
0000_0000_0000_0000															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNI15	UNI14	UNI13	UNI12	UNI11	UNI10	UNI9	UNI8	UNI7	UNI6	UNI5	UNI4	UNI3	UNI2	UNI1	UNI0

Figure 7-44. UNI Register (UNIR)

Table 7-40. UNIR Field Descriptions

Bits	Name	Description
31–16	—	Reserved, should be cleared.
15–0	UNI15–UNI0	User-network interface. The UNI bits define the type of interface that the physical link crosses. It affects the interpretation of the four most significant bits of the ATM cell header. 0 The physical link does not cross a user-network interface (UNI). It may cross a network-network interface (NNI), or it may be an intranetwork link. 1 The physical link crosses a UNI.

7.1.6.11 Ingress Processing Configuration Register (IPCR)

This register defines the MC92520 ingress processing parameters.

31							24	23	22	21	20	19	18	17	16
0000_0000								IGCTE	ICCR	IRCR	ISFCE	ISFNE	ISPE	ISBCE	ISBNE
15		13	12	11	10	9	8	7	6		4	3	2	1	0
IPCC			0	IGZC	IUHC	IIP	IROE	0	IBCC			0	IAPE		IACE

Figure 7-45. Ingress Processing Configuration Register (IPCR)

Table 7-41. IPCR Field Descriptions

Bits	Name	Description
31–24	—	Reserved, should be cleared.
23	IGCTE	Global ingress CLP transparency enable. This bit enables CLP transparency function on the ingress. See Section 6.2 for details.

Table 7-41. IPCR Field Descriptions (Continued)

Bits	Name	Description
22	ICCR	Ingress check CRC on RM cells. This bit determines whether the CRC of RM cells that are received in the ingress is checked. 0 The CRC of RM cells that are received in the ingress is not checked. 1 The CRC of RM cells that are received in the ingress is checked and if it is not o.k, then the cell is removed and can be copied to the microprocessor.
21	IRCR	Ingress recalculate CRC on RM cells. This bit determines whether the CRC of ingress RM cells is recalculated. 0 The CRC of ingress RM cells is not recalculated. 1 The CRC of ingress RM cells is recalculated.
20	ISFCE	Global ingress set FRM CI enable. This bit enables the CI bit to be set in forward RM cells received in ingress. See Section 6.4.2.3 for details. 0 CI bit is disabled in forward RM cells received in ingress. 1 Setting CI bit in forward RM cells received in ingress is enabled.
19	ISFNE	Global ingress set FRM NI enable. This bit enables setting NI bit in forward RM cells received in ingress. See Section 6.4.2.3 for details. 0 Setting NI bit in forward RM cells received in ingress is disabled. 1 Setting NI bit in forward RM cells received in ingress is enabled.
18	ISPE	Global ingress set PTI enable. This bit enables setting PTI[1] bit in cells with PTI[2]=0 that are received in ingress. See Section 6.4.2.3 for details. 0 Setting PTI[1] bit in cells with PTI[2]=0 that are received in ingress is disabled. 1 Setting PTI[1] bit in cells with PTI[2]=0 that are received in ingress is enabled.
17	ISBCE	Global ingress set BRM CI enable. This bit enables setting CI bit in backward RM cells received in ingress. See Section 6.4.2.3 for details. 0 Setting CI bit in backward RM cells received in ingress is disabled. 1 Setting CI bit in backward RM cells received in ingress is enabled.
16	ISBNE	Global ingress set BRM NI enable. This bit enables setting NI bit in backward RM cells received in ingress. See Section 6.4.2.3 for details. 0 Setting NI bit in backward RM cells received in ingress is disabled. 1 Setting NI bit in backward RM cells received in ingress is enabled.
15–13	IPCC	Ingress policing counters control. This field determines which counters appear in the policing counters table if ingress UPC is enabled (The UPC flow (UPCF) bit in the ATMC CFB configuration register (ACR) is reset). It also determines the size of each record in the table. See Section 7.2.6 for more details. 000 The policing table does not exist. 001 The policing table contains three counters and one reserved long word: DSCD0, DSCD1, TAG, reserved. 010 The policing table contains three counters: DSCD0, DSCD1, TAG. 011 The policing table contains two counters: DSCD, TAG. 100 The policing table contains one counter: TAG. 101 The policing table contains one counter: DSCD. 110 Reserved 111 Reserved
12	—	Reserved, should be cleared.
11	IGZC	Ingress GFC zero check. This bit determines whether the MC92520 checks that the GFC header field is zero on cells arriving over a UNI. 0 The GFC field is not checked. 1 The GFC field is checked. If it is non-zero, the cell is copied to the cell extraction queue.

Table 7-41. IPCR Field Descriptions (Continued)

Bits	Name	Description
10	IUHC	Ingress unallocated header bits check. This bit determines whether the MC92520 checks that unallocated bits of the cell header are zero (as defined by the VPI mask and VCI mask, see Section 5.1.2). 0 The unallocated bits of the header are not checked. 1 The unallocated bits of the header are checked. If one or more bits are set, the cell is copied to the cell extraction queue and removed from the cell flow.
9	IIP	Ingress insertion priority. This bit determines the priority between inserted/generated cells and ingress received cells. Note that insertion is always limited by the leaky bucket mechanism 0 Ingress received cells priority is higher than inserted/generated cells. 1 Inserted/generated cells priority is higher than ingress received cells.
8	IROE	Ingress RM overlay enable. This bit enables updating switch parameters words in the case of RM cell. See Section 6.4.3 for details.
7	—	Reserved, should be cleared.
6–4	IBCC	Ingress billing counters control. The IBCC field determines which counters appear in the ingress billing counters table. It also determines the size of each record in the table. See Section 7.2.4 for more details. 000 The ingress billing table does not exist. 001 The ingress billing table contains four counters: IUCLP0, IUCLP1, IOCLP0, IOCLP1. 010 The ingress billing table contains three counters: IUCLP0, IUCLP1, IOAM. 011 The ingress billing table contains three counters and one reserved long word: IUCLP0, IUCLP1, IOAM, reserved. 100 The ingress billing table contains two counters: IU, IOAM. 101 The ingress billing table contains two counters: ICLP0, ICLP1. 110 The ingress billing table contains one counter: ICNTR. 111 Reserved
3	—	Reserved, should be cleared.
2—1	IAPE	Ingress address translation VPI enable. The IAPE field determines whether and how address translation is performed on the VPI field in the ingress cell flow. 00 No translation is performed on the VPI field. 01 The entire VPI field is replaced. 10 The VPI field is replaced based on the UNI bit of the link (see Section 7.1.6.10). If the link is a UNI, only bits 7:0 of the VPI are replaced. Otherwise, the entire VPI field is replaced. 11 Reserved
0	IACE	Ingress address translation VCI enable. The IACE bit determines whether address translation is performed on the VCI field in the ingress cell flow. 0 No translation is performed on the VCI field. 1 The entire VCI field is replaced if the ingress virtual path connection (IVPC) bit of the connection's ingress parameters word is reset, indicating VC switching.

7.1.6.12 Egress Processing Configuration Register (EPCR)

This register defines the MC92520 egress processing parameters.

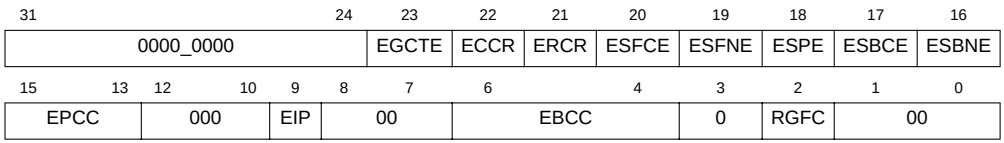


Figure 7-46. Egress Processing Configuration Register (EPCR)

Table 7-42. EPCR Field Descriptions

Bits	Name	Description
31–24	—	Reserved, should be cleared.
23	EGCTE	Global egress CLP transparency enable. This bit enables CLP transparency function on the egress. See Section 6.2 for details.
22	ECCR	Egress check CRC on RM cells. This bit determines whether the CRC of RM cells that are received in the egress is checked. 0 The CRC of RM cells that are received in the egress is not checked. 1 The CRC of RM cells that are received in the egress is checked and if it is, then the cell is removed and can be copied to the microprocessor.
21	ERCR	Egress recalculate CRC on RM cells. This bit determines whether the CRC of egress RM cells is recalculated. 0 The CRC of egress RM cells is not recalculated. 1 The CRC of egress RM cells is recalculated.
20	ESFCE	Global egress set FRM CI enable. This bit enables setting CI bit in forward RM cells received in egress. See Section 6.4.2.4 for details. 0 Setting CI bit in forward RM cells received in egress is disabled. 1 Setting CI bit in forward RM cells received in egress is enabled.
19	ESFNE	Global egress set FRM NI enable. This bit enables setting NI bit in forward RM cells received in egress. See Section 6.4.2.4 for details. 0 Setting NI bit in forward RM cells received in egress is disabled. 1 Setting NI bit in forward RM cells received in egress is enabled.
18	ESPE	Global egress set PTI enable. This bit enables setting PTI[1] bit in cells with PTI[2]=0 that are received in egress. See Section 6.4.2.4 for details. 0 Setting PTI[1] bit in cells with PTI[2]=0 that are received in egress is disabled. 1 Setting PTI[1] bit in cells with PTI[2]=0 that are received in egress is enabled.
17	ESBCE	Global egress set BRM CI enable. This bit enables setting CI bit in backward RM cells received in egress. See Section 6.4.2.4 for details. 0 Setting CI bit in backward RM cells received in egress is disabled. 1 Setting CI bit in backward RM cells received in egress is enabled.
16	ESBNE	Global egress set BRM NI enable. This bit enables setting NI bit in backward RM cells received in egress. See Section 6.4.2.4 for details. 0 Setting NI bit in backward RM cells received in egress is disabled. 1 Setting NI bit in backward RM cells received in egress is enabled.
15–13	EPCC	Egress policing counters control. This field determines which counters appear in the policing counters table if egress UPC is enabled (The UPC flow (UPCF) bit in the ATMC CFB configuration register (ACR) is set). It also determines the size of each record in the table. See Section 7.2.6 for more details. 000 The policing table does not exist. 001 The policing table contains three counters and one reserved long word: DSCD0, DSCD1, TAG, Reserved. 010 The policing table contains three counters: DSCD0, DSCD1, TAG. 011 The policing table contains two counters: DSCD, TAG. 100 The policing table contains one counter: TAG. 101 The policing table contains one counter: DSCD. 110 Reserved 111 Reserved
12–10	—	Reserved, should be cleared.
9	EIP	Egress insertion priority. This bit determines the priority between inserted or generated cells and egress received cells. Insertion is always limited by the leaky bucket mechanism. 0 Inserted or generated cells' priority is higher than egress received cells. 1 Egress received cells' priority is higher than inserted or generated cells.

Table 7-42. EPCR Field Descriptions (Continued)

Bits	Name	Description
8–7	—	Reserved, should be cleared.
6–4	EBCC	Egress billing counters control. The EBCC field determines which counters appear in the egress billing counters table. It also determines the size of each record in the table. See Section 7.2.5 for more details. 000 The egress billing table does not exist. 001 The egress billing table contains four counters: EUCLP0, EUCLP1, EOCLP0, EOCLP1. 010 The egress billing table contains three counters: EUCLP0, EUCLP1, EOAM. 011 The egress billing table contains three counters and one reserved long word: EUCLP0, EUCLP1, EOAM, reserved. 100 The egress billing table contains two counters: EU, EOAM. 101 The egress billing table contains two counters: ECLP0, ECLP1. 110 The egress billing table contains one counter: ECNTR. 111 Reserved
3	—	Reserved, should be cleared.
2	RGFC	Replace GFC field. The RGFC bit determines whether, during address translation, the GFC field of the ATM header is replaced by the value provided in the egress translation address word in the external memory. This bit is only relevant for connections belonging to a UNI. 0 The GFC field is not replaced. 1 The GFC field is replaced.
1–0	—	Reserved, should be cleared.

7.1.6.13 Egress Multicast Configuration Register (EMCR)

This register contains the MC92520 egress multicast translation parameters.

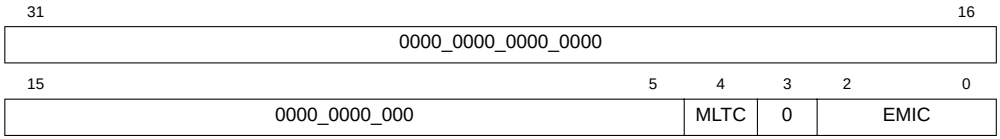


Figure 7-47. Egress Multicast Configuration Register (EMCR)

Table 7-43. EMCR Field Descriptions

Bits	Name	Description
31–5	—	Reserved, should be cleared.
4	MLTC	Multicast translation table control. The MLTC bit determines the existence of the multicast translation table in external memory. 0 The multicast translation table does not exist. 1 the multicast translation table exists.

Table 7-43. EMCR Field Descriptions

Bits	Name	Description
3	—	Reserved, should be cleared.
2–0	EMIC	Egress multicast identifier control. This field determines the number of bits of the multicast identifier that are allocated for multicast translation (see Section 5.2.2). 000 9 bits of the multicast identifier are allocated 001 10 bits of the multicast identifier are allocated 010 11 bits of the multicast identifier are allocated 011 12 bits of the multicast identifier are allocated 100 13 bits of the multicast identifier are allocated 101 14 bits of the multicast identifier are allocated 110 15 bits of the multicast identifier are allocated 111 16 bits of the multicast identifier are allocated

7.1.6.14 ATMC CFB Configuration Register (ACR)

This register defines ATMC CFB general processing parameters.

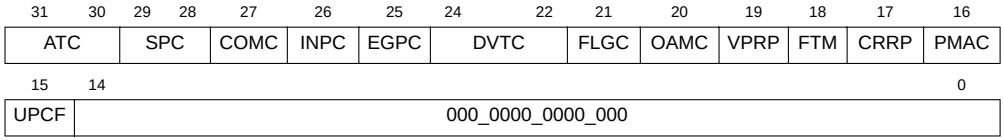


Figure 7-48. ATMC CFB Configuration Register (ACR)

Table 7-44. ACR Field Descriptions

Bits	Name	Description
31–30	ATC	Address translation control. The ATC bit determines whether the address translation long words exist in the context parameters table record. 00 No address translation words exist. 01 One common address translation word exists. 10 Both address translation words exist. 11 Reserved
29–28	SPC	Ingress switch parameter control. This field determines the number of long words of switch parameters per record in the context parameters table in external memory. 00 No switch parameters 01 One long word of switch parameters per record in the context parameters table 10 Two long words of switch parameters per record in the context parameters table 11 Three long words of switch parameters per record in the context parameters table
27	COMC	Common parameters control. The COMC bit determines whether the common parameters long word exists in the context parameters table record. 0 The common parameters long word does not exist. 1 The common parameters long word exists.
26	INPC	Ingress parameters control. The INPC bit determines whether the ingress parameters long word exists in the context parameters table record. 0 The ingress parameters long word does not exist. 1 The ingress parameters long word exists.
25	EGPC	Egress parameters control. The EGPC bit determines whether the egress parameters long word exists in the context parameters table record. 0 The egress parameters long word does not exist. 1 The egress parameters long word exists.

Table 7-44. ACR Field Descriptions (Continued)

Bits	Name	Description
24–22	DVTC	Dump vector table control. The DVTC field determines the size of the dump vector table in external memory. Each record consists of two long words. See Section 7.2.13 for details. 000 The dump vector table does not exist. 001 The dump vector table contains 128 records. 010 The dump vector table contains 512 records. 011 The dump vector table contains 2K records. 100 The dump vector table contains 8K records. 101 The dump vector table contains 32K records. 110 The dump vector table contains 128K records. 111 The dump vector table contains 512K records.
21	FLGC	Flags table control. The FLGC bit determines the existence of the flags table in external memory. 0 The flags table does not exist. 1 The flags table exists.
20	OAMC	OAM table control. The OAMC bit determines the existence of the OAM table in external memory. 0 The OAM table does not exist. 1 The OAM table exists.
19	VPRP	VP RM cell PTI. This bit determines whether a cell is a VP RM cell only if its PTI6. 0 A cell is a VP RM cell if and only if it belongs to a VP connection, its VCI6 and its PTI6. 1 A cell is a VP RM cell only if it belongs to a VP connection and its VCI6.
18	FTM	FMC timestamp enable. The FTM bit determines the encoding of the time-stamp field of OAM forward monitoring cells generated by the MC92520. 0 The time-stamp field is encoded with the default value of all ones. 1 The time-stamp field is encoded with the current value of the cell time register (CLTM).
17	CRRP	VC RM cell removal point. This bit determines whether a VC cell whose PTI6 or 7 is removed at the OAM termination point or whether its removal is subjected to the per-connection enable bits for PTI = 6 or PTI = 7. 0 A VC cell whose PTI6 or 7 is removed at the OAM termination point as defined by the egress end-to-end OAM termination (EEOT) bit in the egress and by the ingress end-to-end OAM termination (IEOT) bit in the ingress. 1 A VC cell is removed at the egress if the egress PTI=6 remove (EP6R) bit is set and its PTI = 6 or if the egress PTI = 7 remove (EP7R) bit is set and its PTI = 7. A VC cell is removed at the ingress if the ingress PTI 6 remove (EP6R) bit is set and its PTI = 6 or if the ingress PTI = 7 remove (IP7R) bit is set and its PTI = 7.
16	PMAC	PM on all connections. This bit determines whether OAM performance monitoring test can be done on all connections or on 64 connections. 0 Performance monitoring can be done only on 64 selected connections. 1 Performance monitoring can be done on all connections.
15	UPCF	UPC flow. This bit determines whether the UPC is active in the ingress flow or in the egress flow. 0 The UPC is active in the ingress flow. 1 The UPC is active in the egress flow.
14–0	—	Reserved, should be cleared.

7.1.6.15 General Configuration Register (GCR)

This register determines the configuration of those sections of the MC92520 not contained in the ATMC CFB.

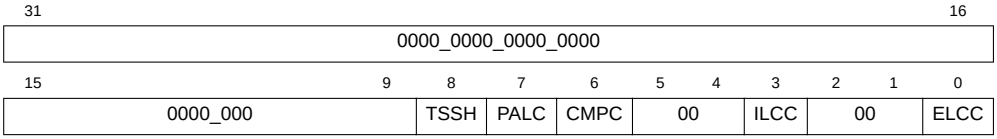


Figure 7-49. General Configuration Register (GCR)

Table 7-45. GCR Field Descriptions

Bits	Name	Description
31–9	—	Reserved, should be cleared.
8	TSSH	Time scale shift. This bit is used to determine whether time scaling in the leaky buckets is shifted to account for differing internal clock speeds in different clock modes. For more details, see Appendix A, “UPC/NPC Design.” The default value of this bit is 0. 0 No shift of time scaling. 1 All time scaling is shifted to account for 4x clock operation (that is, a 100MHz internal clock in MC92520 mode vs. a 25MHz internal clock in MC92510 mode).
7	PALC	PPD admit last cell. This bit determines whether the last cell of a partially discarded packet is admitted or discarded independent of an enforcer decision for connections NOT using guaranteed frame rate (GFR) UPC. 0 The last cell of a partial packet is policed just like any other cell of the packet. 1 The last cell of a partial packet is always admitted.
6	CMPC	Context parameters extension table control. This bit determines the existence of the context parameters extension table in external memory. See Section 7.2.16 for details. 0 The context parameters extension table does not exist. 1 The context parameters extension table exists.
5–4	—	Reserved, should be cleared.
3	ILCC	Ingress link counters table control. The ILCC bit determines the existence of the ingress link counters table in external memory. 0 The ingress link counters table does not exist. 1 The ingress link counters table exists.
2–1	—	Reserved, should be cleared.
0	ELCC	Egress link counters table control. The ILCC bit determines the existence of the egress link counters table in external memory. 0 The egress link counters table does not exist. 1 The egress link counters table exists.

7.1.6.16 Context Parameters Table Pointer Register (CPTP)

This register contains the pointer to the first word of the context parameters table. The pointer is in units of 256 bytes.

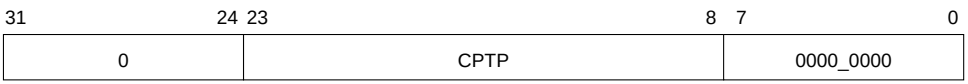


Figure 7-50. Context Parameters Table Pointer (CPTP) Fields

7.1.6.17 OAM Table Pointer Register (OTP)

This register contains the pointer to the first word of the OAM table. The pointer is in units of 256 bytes.

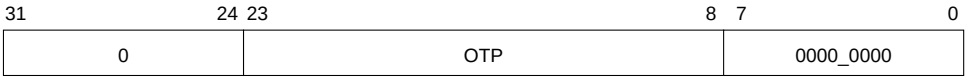


Figure 7-51. OAM Table Pointer (OTP) Register Fields

7.1.6.18 Dump Vector Table Pointer Register (DVTP)

This register contains the pointer to the first word of the dump vector table. The pointer is in units of 256 bytes.

Figure 7-52. Dump Vector Table Pointer (DVTP) Register Fields

7.1.6.19 VC Table Pointer Register (VCTP)

This register contains the pointer to the first word of the VC table. The pointer is in units of 256 bytes.

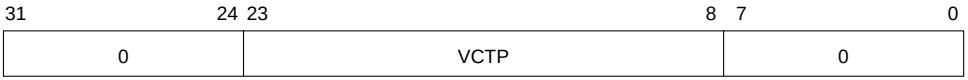


Figure 7-53. VC Table Pointer (VCTP) Register Fields

7.1.6.20 Multicast Translation Table Pointer Register (MTTP)

This register contains the pointer to the first word of the multicast translation table. The pointer is in units of 256 bytes.

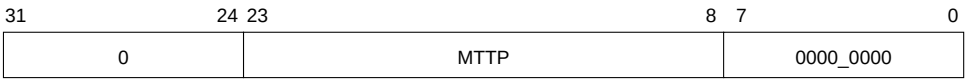


Figure 7-54. Multicast Translation Table Pointer (MTTP) Register Fields

7.1.6.21 Flags Table Pointer Register (FTP)

This register contains the pointer to the first word of the flags table. The pointer is in units of 256 bytes.

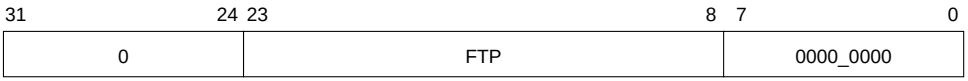


Figure 7-55. Flags Table Pointer (FTP) Register Fields

7.1.6.22 Egress Link Counters Table Pointer Register (ELCTP)

This register contains the pointer to the first word of the egress link counters table. Each pointer is in units of 256 bytes.

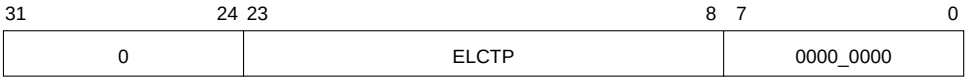


Figure 7-56. Egress Link Counters Table Pointer (ELCTP) Register Fields

7.1.6.23 Ingress Link Counters Table Pointer Register (ILCTP)

This register contains the pointer to the first word of the ingress link counters table. Each pointer is in units of 256 bytes.

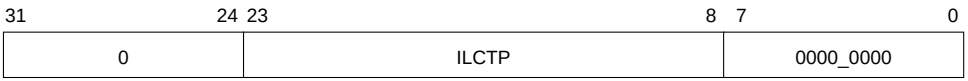


Figure 7-57. Ingress Link Counters Table Pointer (ILCTP) Register Fields

7.1.6.24 Context Parameters Extension Table Pointer Register (CPETP)

This register contains the pointer to the first word of the context parameters extension table. The pointer is in units of 256 bytes.

Figure 7-58. Context Parameters Extension Table Pointer (CPETP) Register Fields

7.1.6.25 Node ID Register 0 (ND0)

This register contains the most significant bits of the node ID.

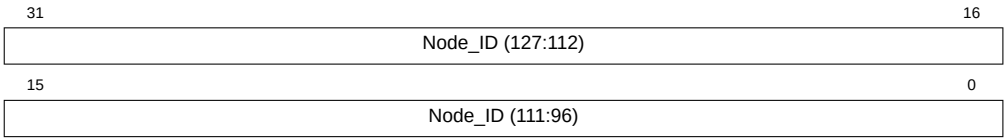


Figure 7-59. Node ID Register 0 (ND0) Fields

7.1.6.26 Node ID Register 1 (ND1)

This register contains the upper-middle portion of the node ID.

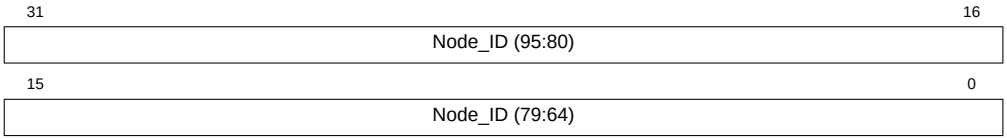


Figure 7-60. Node ID Register 1 (ND1) Fields

7.1.6.27 Node ID Register 2 (ND2)

This register contains the lower-middle portion of the node ID.

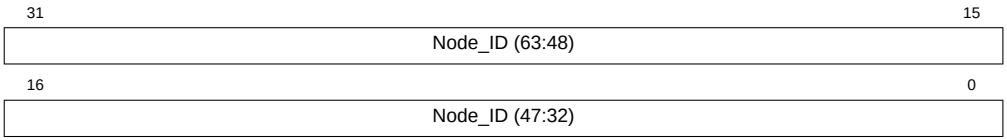


Figure 7-61. Node ID Register 2 (ND2) Fields

7.1.6.28 Node ID Register 3 (ND3)

This register contains the least significant bits of the node ID.

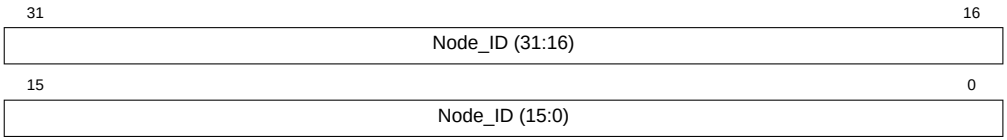


Figure 7-62. Node ID Register 3 (ND3) Fields

7.1.6.29 Ingress VCI Copy Register (IVCR)

The ingress VCI copy register is used to select specific reserved values of VCI for copying to the processor from the ingress cell flow. The values contained in this register are used by all VPCs that have the ingress VCR/VRR registers enable (IVRE) bit of the ingress parameters word in external memory set.

NOTE:

The definition of this register may change in future revisions of the MC92520 to reflect the evolving ATM standards.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVC 31	IVC 30	IVC 29	IVC 28	IVC 27	IVC 26	IVC 25	IVC 24	IVC 23	IVC 22	IVC 21	IVC 20	IVC 19	IVC 18	IVC 17	IVC 16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVC 15	IVC 14	IVC 13	IVC 12	IVC 11	IVC 10	IVC 9	IVC 8	IVC 7	IVC 6	IVC 5	00		IVC 2	IVC 1	IVC 0

Figure 7-63. Ingress VCI Copy Register (IVCR)

Table 7-46. IVCR Field Descriptions

Bits	Name	Description
31–5, 2–0	IVC31–IVC5, IVC2–IVC0	Ingress VCI copy. Each IVC bit refers to a VCI value. Each IVC bit, when set, indicates that cells received in the ingress cell flow with that VCI value should be copied to the cell extraction queue.
4–3	—	Reserved, should be cleared.

7.1.6.30 Egress VCI Copy Register (EVCR)

The egress VCI copy register is used to select specific reserved values of VCI for copying to the processor from the egress cell flow. The values contained in this register are used by all VPCs that have the egress VCR/VRR registers enable (EVRE) bit of the egress parameters word in external memory set.

The definition of this register may change in future revisions of the MC92520 to reflect the evolving ATM standards.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EVC 31	EVC 30	EVC 29	EVC 28	EVC 27	EVC 26	EVC 25	EVC 24	EVC 23	EVC 22	EVC 21	EVC 20	EVC 19	EVC 18	EVC 17	EVC 16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVC 15	EVC 14	EVC 13	EVC 12	EVC 11	EVC 10	EVC 9	EVC 8	EVC 7	EVC 6	EVC 5	00		EVC 2	EVC 1	EVC 0

Figure 7-64. Egress VCI Copy Register (EVCR)

Table 7-47. EVCR Field Descriptions

Bits	Name	Description
31–5, 2–0	EVC31–EVC5, EVC2–EVC0	Egress VCI copy. Each EVC bit refers to a VCI value. Each EVC bit, when set, indicates that cells received in the egress cell flow with that VCI value should be copied to the cell extraction queue.
4–3	—	Reserved, should be cleared.

7.1.6.31 Ingress VCI Remove Register (IVRR)

The ingress VCI remove register is used to select specific reserved values of VCI for removal from the cell flow. The values contained in this register are used by all VPCs that have the ingress VCR/VRR registers enable (IVRE) bit of the ingress parameters word in external memory set.

NOTE:

The definition of this register may change in future revisions of the MC92520 to reflect the evolving ATM standards.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVR 31	IVR 30	IVR 29	IVR 28	IVR 27	IVR 26	IVR 25	IVR 24	IVR 23	IVR 22	IVR 21	IVR 20	IVR 19	IVR 18	IVR 17	IVR 16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVR 15	IVR 14	IVR 13	IVR 12	IVR 11	IVR 10	IVR 9	IVR 8	IVR 7	IVR 6	IVR 5	00		IVR 2	IVR 1	IVR 0

Figure 7-65. Ingress VCI Remove Register (IVRR)

Table 7-48. IVRR Field Descriptions

Bits	Name	Description
31–5, 2–0	IVR31–IVR5, IVR2–IVR0	Ingress VCI remove. Each IVR bit refers to a VCI value. The IVR bit, when set, indicates that cells received in the ingress cell flow with that VCI value should be removed from the cell flow.
4–3	—	Reserved, should be cleared.

7.1.6.32 Egress VCI Remove Register (EVRR)

The egress VCI remove register is used to select specific reserved values of VCI for removal from the cell flow. The values contained in this register are used by all VPCs that have the egress VCR/VRR registers enable (EVRE) bit of the egress parameters word in external memory set.

The definition of this register may change in future revisions of the MC92520 to reflect the evolving ATM standards.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EVR 31	EVR 30	EVR 29	EVR 28	EVR 27	EVR 26	EVR 25	EVR 24	EVR 23	EVR 22	EVR 21	EVR 20	EVR 19	EVR 18	EVR 17	EVR 16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVR 15	EVR 14	EVR 13	EVR 12	EVR 11	EVR 10	EVR 9	EVR 8	EVR 7	EVR 6	EVR 5	00		EVR 2	EVR 1	EVR 0

Figure 7-66. Egress VCI Remove Register (EVRR)

Table 7-49. EVRR Field Descriptions

Bits	Name	Description
31–5, 2–0	EVR31–EVR5, EVR2–EVR0	Egress VCI remove. Each EVR bit refers to a VCI value. The EVR bit, when set, indicates that cells received in the egress cell flow with that VCI value should be removed from the cell flow.
4–3	—	Reserved, should be cleared.

7.1.6.33 Performance Monitoring Exclusion Register (PMER)

The performance monitoring exclusion register determines which of the reserved values of VCI and PTI should be excluded from OAM performance monitoring blocks. This register is provided because this issue has not yet been standardized, and any aberration from the standard may destroy the block test results.

NOTE:

The definition of this register may change in future revisions of the MC92520 to reflect the evolving ATM standards.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PVE 31	PVE 30	PVE 29	PVE 28	PVE 27	PVE 26	PVE 25	PVE 24	PVE 23	PVE 22	PVE 21	PVE 20	PVE 19	PVE 18	PVE 17	PVE 16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PVE 15	PVE 14	PVE 13	PVE 12	PVE 11	PVE 10	PVE 9	PVE 8	PVE 7	PVE 6	PVE 5	PTE 7	PTE 6	PVE 2	PVE 1	PVE 0

Figure 7-67. Performance Monitoring Exclusive Register (PMER)

Table 7-50. PMER Field Descriptions

Bits	Name	Description
31–5, 2–0	PVE31–PVE5, PVE2–PVE0	Performance monitoring VCI exclude. Each PVE bit refers to a VCI value. The PVE bit, when set, indicates that cells belonging to a virtual path connection and having that VCI value should be excluded from any performance monitoring block test at the F4 level.
4–3	PTE7–PTE6	Performance monitoring PTI exclude. Each PTE bit refers to a PTI value. The PTE bit, when set, indicates that cells belonging to a virtual channel connection and having that PTI value should be excluded from any performance monitoring block test at the F5 level.

7.1.6.34 RM Overlay Register (RMOR)

This register contains all the parameters that are related to RM cell overlay. Refer to Section 6.4.3 for details. The register has the following structure

31	30	29 28	27 24	23 16
IBOE	IFOE	00	ROL	ROM

15 8	7 0
0000_0000	ROF

Figure 7-68. RM Overlay Register (RMOR)

Table 7-51. RMOR Field Descriptions

Bits	Name	Description
31	IBOE	Ingress BRM overlay enable. This bit determines whether the MC92520 overlays the RM overlay field (ROF) on the switch parameters for ingress backward RM cells. 0 Switch parameters are not overlayed when a backward RM cell is received in the ingress. 1 Switch parameters are overlayed when a backward RM cell is received in the ingress
30	IFOE	Ingress FRM overlay enable. This bit determines whether the MC92520 overlays the RM overlay field (ROF) on the switch parameters for ingress forward RM cells. 0 Switch parameters are not overlayed when a forward RM cell is received in the ingress. 1 Switch parameters are overlayed when a forward RM cell is received in the ingress
29–28	—	Reserved, should be cleared.
27–24	ROL	RM overlay location. This field contains the number of the switch parameters byte that should be overlayed.
23–16	ROM	RM overlay mask. This field contains the byte mask that serves for overlaying the RM overlay field (ROF) over ingress switch parameters byte.
15–8	—	Reserved, should be cleared.
7–0	ROF	RM overlay field. This field contains the byte that is overlayed on ingress switch parameters byte. Each bit in this field is overlayed on the corresponding bit in the ingress switch parameters only if it is enabled by the corresponding bit in the RM overlay mask (ROM) field.

7.1.6.35 CLP Transparency Overlay Register (CTOR)

This register contains the location of the ingress overhead CLP (IOCLP) bit in the ingress switch parameters. See Section 6.2 for details. The register has the following structure:

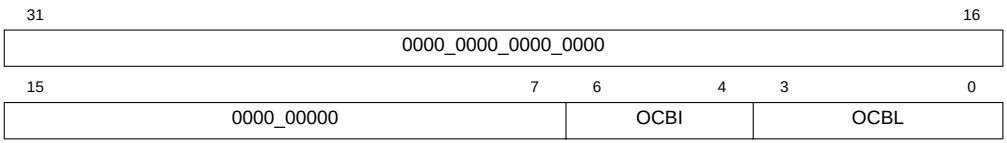


Figure 7-69. CLP Transparency Overlay Register (CTOR)

Table 7-52. CTOR Field Descriptions

Bits	Name	Description
31–7	—	Reserved, should be cleared.
6–4	OCBL	IOCLP byte location. This field defines the byte offset of a location in the switch data structure (overhead, header, and HEC bytes) that is used to store the ingress overhead CLP (IOCLP) bit. Byte numbering starts with 0 and identifies the first byte transferred (the left-most byte of the shaded areas in figures Figure 5-14 through Figure 5-17.)
3–0	OCBI	IOCLP bit location. This field contains the number of the ingress overhead CLP (IOCLP) bit within the byte specified by the IOCLP byte location (OCBL) field. The most-significant bit is number 7, and the least-significant bit is number 0.

7.1.6.36 Egress Overhead Manipulation Register (EGOMR)

This register contains fields for manipulating egress overhead fields. See Section 5.2.1 for details.

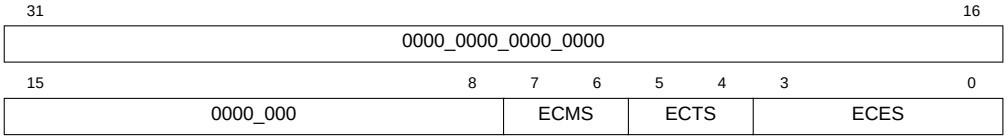


Figure 7-70. Egress Overhead Manipulation Register (EGOMR)

Table 7-53. EGOMR Field Descriptions

Bits	Name	Description
31–8	—	Reserved, should be cleared.
7–6	ECMS	Egress cell processing block M bit source. This field contains the source for the M bit that is used by the egress cell processing block. 00 The M bit used by the egress cell processing block is taken from the M bit that is extracted from the switch cell data structure. 01 The M bit used by the egress cell processing block is taken from the logical NOT of the M bit that is extracted from the switch cell data structure. 10 The M bit used by the egress cell processing block is 0. 11 The M bit used by the egress cell processing block is 1.
5–4	ECTS	Egress cell processing block MTTS Size. This field contains the size of the MTTS field that is used by the egress cell processing block. 00 The MTTS field that is used by the egress cell processing block is the MTTS field that is extracted from the switch cell data structure. 01 The MTTS field that is used by the egress cell processing block is the least significant bit of the MTTS field that is extracted from the switch cell data structure. 10 The MTTS field that is used by the egress cell processing block is the 2 least significant bits of the MTTS field that is extracted from the switch cell data structure. 11 The MTTS field that is used by the egress cell processing block is the 3 least significant bits of the MTTS field that is extracted from the switch cell data structure.

Table 7-53. EGOMR Field Descriptions (Continued)

Bits	Name	Description
3–0	ECES	Egress cell processing block ECI size. This field contains the size of the ECI field that is used by the egress cell processing block if the IHAF bit in the egress switch interface configuration register is reset. 0000 The ECI field that is used by the egress cell processing block is the ECI field that is extracted from the switch cell data structure. 0001 Reserved. 0010 Reserved. 0011 Reserved. 0100 Reserved. 0101 Reserved. 0110 The ECI field that is used by the egress cell processing block is the 6 least significant bits of the ECI field that is extracted from the switch cell data structure. 0111 The ECI field that is used by the egress cell processing block is the 7 least significant bits of the ECI field that is extracted from the switch cell data structure. 1000 The ECI field that is used by the egress cell processing block is the 8 least significant bits of the ECI field that is extracted from the switch cell data structure. 1001 The ECI field that is used by the egress cell processing block is the 9 least significant bits of the ECI field that is extracted from the switch cell data structure. 1010 The ECI field that is used by the egress cell processing block is the 10 least significant bits of the ECI field that is extracted from the switch cell data structure. 1011 The ECI field that is used by the egress cell processing block is the 11 least significant bits of the ECI field that is extracted from the switch cell data structure. 1100 The ECI field that is used by the egress cell processing block is the 12 least significant bits of the ECI field that is extracted from the switch cell data structure. 1101 The ECI field that is used by the egress cell processing block is the 13 least significant bits of the ECI field that is extracted from the switch cell data structure. 1110 The ECI field that is used by the egress cell processing block is the 14 least significant bits of the ECI field that is extracted from the switch cell data structure. 1111 The ECI field that is used by the egress cell processing block is the 15 least significant bits of the ECI field that is extracted from the switch cell data structure.

7.1.6.37 GFR Configuration Register (GFRCR)

This register defines the global behavior of all connections utilizing the frame-based UPC (GFR) enforcer configuration. See Section 6.3.1.1, “Global GFR Configurations” for details.

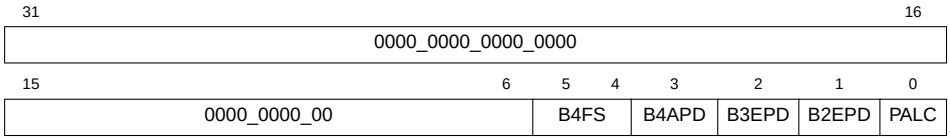


Figure 7-71. GFR Configuration Register (GFRCR)

Table 7-54. GFRCR Field Descriptions

Bits	Name	Description
31–6	—	Reserved, should be cleared.
5–4	B4FS	Bucket 4 fill scope. This field defines the fill scope of the leaky bucket shared by both FCRA and FCRB enforcer stages. The B4FS field definition is the same as for cell-based fill scopes. For GFR applications, it will be typically set to the CLP = 0 +1 flow. Note, the CLP=1 flow includes both user-marked and UPC-tagged cells. See Section 6.3.1.6, “Fair Cell Rate (FCR)” for details. 00 The FCR enforcer stages are bypassed. 01 The FCR enforcer leaky bucket is filled with CLP = 0 cells. 10 The FCR enforcer leaky bucket is filled with CLP = 1 cells. 11 The FCR enforcer leaky bucket is filled with CLP = 0 +1 cells.
3	B4APD	Bucket 4 all packet discard. This bit determines whether all frames exceeding the FCRA bucket limit are discarded, independent of the FCRA action scope or the frame’s CLP bit value (in most configurations CLP=0 frames are not discarded by FCR enforcers). See Section 6.3.1.6, “Fair Cell Rate (FCR)” for details. 0 The action scope defined by the FCRA TAG and SCP fields in the fourth leaky bucket are applied. This implies that the FCRA bucket limit is not a hard limit, because CLP=0 frames may be admitted based on some action scopes. 1 The FCRA bucket limit will be enforced. All frames exceeding the FCRA bucket limit will be discarded with EPD.
2	B3EPD	Bucket 3 early packet discard. This bit determines whether the GFR enforcer associated with the third leaky bucket (typically used to enforce MCR eligibility) will act immediately (partial packet policing), or delayed (policing the next packet early and whole, if necessary). See Section 6.3.1.5, “Minimum Cell Rate (MCR)” for details. 0 Policing starts immediately, resulting in partial packet tagging (PPT) or partial packet discard (PPD). 1 Policing is delayed and confirmed at the first (early) cell of a new packet, resulting in early packet tagging (EPT) or early packet discard (EPD).
1	B2EPD	Bucket 2 early packet discard. This bit determines whether the GFR enforcer associated with the second leaky bucket (typically used to enforce PCR eligibility) will act immediately (partial packet policing), or delayed (policing the next packet early and whole, if necessary). See Section 6.3.1.4, “Peak Cell Rate (PCR)” for details. 0 Policing starts immediately, resulting in partial packet tagging (PPT) or partial packet discard (PPD). 1 Policing is delayed and confirmed at the first (early) cell of a new packet, resulting in early packet tagging (EPT) or early packet discard (EPD).
0	PALC	PPD admit last cell. This bit determines whether the last cell of a partially discarded packet is admitted or discarded independent of an enforcer decision. See Section 6.3.1.2, “Cell Loss Priority (CLP) Consistency” for details. 0 The last cell of a partial packet is policed just like any other cell of the packet. 1 The last cell of a partial packet is always admitted.

7.1.7 Pseudo-Registers

These registers are use to perform certain operations on the MC92520, and may be written by the processor in either of the part’s modes of operation (setup mode or operate mode).

7.1.7.1 Software Reset Register (SRR)

A write access to the software reset pseudo-register (SSR) resets all MC92520 registers and internal FIFOs, except the PLL registers and PLL logic. As a result of the software reset, all non-PLL registers are loaded with their default values and the MC92520 is transferred to (or remains in) setup mode. See Section 3.2 for more information.

7.1.7.2 Start Scan Register (SSR)

A write access to this pseudo-register location starts the internal scan operation. If the internal scan process is still operating (didn't finish scanning all the connections), this access is ignored.

7.1.7.3 Enter Operate Mode Register (EOMR)

A write access to this pseudo-register location transfers the MC92520 to operate mode within 1–3 cell times. The processor may use the operate mode (OM) bit in the interrupt register (IR) to find out when the MC92520 is in operate mode. (See Section 3.2 for more information.)

7.1.8 External Memory Accesses

This memory space (ADD(25) = 1) provides the processor with access to the MC92520 external memory in both setup and operate modes. In setup mode, the value of bits 23:2 of the MADD input pins are copied to the EMADD output pins, and the external memory interface control pins are driven as necessary. The data is transferred between MDATA and EMDATA. If bit 24 of MADD is set, the MC92520 automatically writes back zero to each word of external memory that is read. For details on how the memory space is accessed in operate mode, see Section 3.2.2.1, "Maintenance Slot Access."

Table 7-55. Address Space for Accesses That Use the External Memory Interface

ADD[25:0]	Access Type	
01_AAAA_AAAA_AAAA_AAAA_AAAA_AA00	Reserved/Undefined	
10_AAAA_AAAA_AAAA_AAAA_AAAA_AA00	External memory R/W. See Section 3.1.3, "Initializing External Memory."	**Setup Mode**
11_AAAA_AAAA_AAAA_AAAA_AAAA_AA00	External memory destructive reads.	
10_AAAA_AAAA_AAAA_AAAA_AAAA_AA00	External memory Write requests, stored until next maintenance slot. See Section 3.2.1.1, "Periodic External Memory Maintenance."	**Operate Mode**
11_AAAA_AAAA_AAAA_AAAA_AAAA_AA00	External memory Read requests, stored until next maintenance slot. See Section 3.2.1.1, "Periodic External Memory Maintenance."	

7.2 External Memory Description

The MC92520 uses external memory to store the database of information relating to the processing of cells on a per-connection basis. The MC92520 accesses the external memory using 16- or 32-bit accesses.

7.2.1 Memory Partitioning

The external memory is partitioned into several tables:

- Context parameters table—This table contains a record for each active connection that contains connection-specific information for processing and routing the cells belonging to the connection. See Section 7.2.3 for details.
- Context parameters extension table—This table includes a record for each active connection that contains additional connection-specific information for processing and routing the cells belonging to the connection. See Section 7.2.16 for details.
- Ingress billing counters table—This table includes a record for each active connection that contains the cell counters used by the connection during the normal ingress cell flow. The table is dynamic and updated by the MC92520. The microprocessor is responsible for collecting the contents of the counters on a regular basis. See Section 7.2.4 for details.
- Egress billing counters table—This table includes a record for each active connection that contains the counters used by the connection during the normal egress cell flow. The table is dynamic and updated by the MC92520. The microprocessor is responsible for collecting the contents of the counters on a regular basis. See Section 7.2.5 for details.
- Policing counters table—This table includes a record for each active connection that contains the counters used to record the results of the UPC/NPC policing. This table is dynamic and updated by the MC92520. The microprocessor is responsible for collecting the contents of the counters on a regular basis. See Section 7.2.6 for details.
- Flags table—This table includes a record for each active connection that contains OAM flags used by all the connections during the normal cell flow. This table is dynamic and updated by the MC92520. The microprocessor is responsible for checking the flags on a regular basis. See Section 7.2.7 for details.
- VP tables—The VP table registers contain either an ingress connection identifier (ICI) defined by the microprocessor as an active connection or a reference to the VC table (see Section 5.2.2.2.3). The size and location of the VP table are determined by the ingress link registers (ILNK*n*). If multiple links are supported, each link register defines a separate VP table. Multiple VP tables are not required to be contiguous. See Section 7.2.8 for details.
- VC table—This table contains a list of all the ingress connection identifiers (ICIs) defined by the microprocessor as active virtual channel connections. This table exists only if the table lookup method of address compression is used with VC table lookup enabled. See Section 7.2.9 for details.
- Multicast translation table—This table contains the egress connection identifiers (ECIs) associated with multicast identifiers. See Section 7.2.10 for details.

- Virtual bucket table—Each record in this table contains the information for the UPC/NPC enforcement. This is not a physical table, but a virtual one. Since the parameters table contains a full address for the location of the bucket record of each connection, there is no need to put all the bucket records in consecutive physical locations. Although the user can distribute the records in any manner, the term bucket table in this document refers to the set of all the bucket records. See Section 7.2.11 for details.
- OAM table—This table contains the additional information required to run OAM performance monitoring. See Section 7.2.12 for details.
- Dump vector table—This table contains the dump vectors describing the recent history of the cell processing. This table is generally used for debugging purposes only. See Section 7.2.13 for details.
- Ingress link counters table—This table includes a record for each link that contains the cell counters used by the link during the normal ingress cell flow. This table is dynamic and updated by the MC92520. The microprocessor is responsible for collecting the contents of the counters on a regular basis. See Section 7.2.14 for details.
- Egress link counters table—This table includes a record for each link that contains the cell counters used by the link during the normal egress cell flow. This table is dynamic and updated by the MC92520. The microprocessor is responsible for collecting the contents of the counters on a regular basis. See Section 7.2.15.

Each of the per-connection tables is defined by a pointer only. The sizes of the tables are determined by the number of connections being used. The pointers are programmable with a granularity of 64 long words. These tables include the following:

- Context parameters table pointer register (CPTP)
- Ingress billing counters table pointer register (IBCTP)
- Egress billing counters table pointer register (EBCTP)
- Policing counters table pointer register (PCTP)
- Flags table pointer register (FTP)

The other tables are also defined by pointers that are programmable with a granularity of 64 long words.These tables are:

- VC table pointer register (VCTP)
- Multicast translation table pointer register (MTTP)
- OAM table pointer register (OTP)
- Dump vector table pointer register (DVTP)
- Egress link counters table pointer register (ELCTP)
- Ingress link counters table pointer register (ILCTP)

NOTE:

All fields marked “0” or “reserved” in the descriptions in this section must be written with zeros. The values read from these fields should be considered undefined and should be ignored.

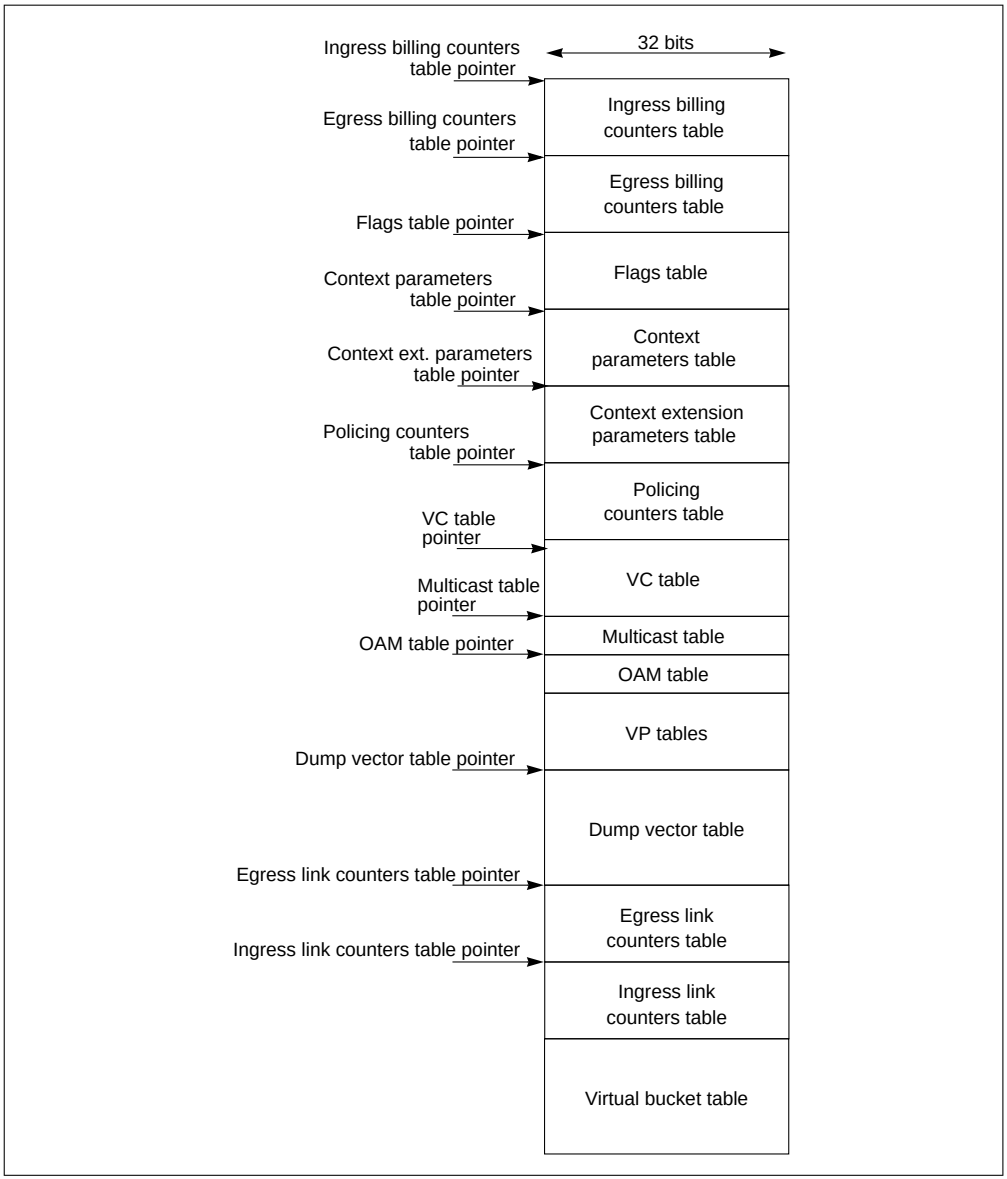


Figure 7-72. External Memory Partitioning

7.2.2 Memory Allocation

The amount of external memory required and its allocation among the various tables can be derived from the following tables. Table 7-56 deals with the tables whose size depends on the number of active connections, while Table 7-57, Table 7-58, and Table 7-59 deal with the remaining tables. More details on these calculations, as well as example configurations, can be found in Appendix C, “VC Bundling.”

Table 7-56. Number of Long Words per Connection in Each Table

Table Name	Option	Long Words
Ingress billing counters		1–4
Policing counters		1–4
Egress billing counters		1–4
Flags		1
Context parameters		1–8
Context extension parameters		1
VP table		see Table 7-57
VC table		see Table 7-58
Multicast translation table		see Table 7-59
Buckets table	One bucket	3
	Two buckets	5
	Three buckets	7
	Four buckets	9
OAM table		8

Table 7-57. VP Table Size (Per Link)

Allocated VPI bits	VP Table Records	VP Table Long Words (without VC Table Lookup)	VP Table Long Words (with VC Table Lookup)
1	2	1	2
2	4	2	4
3	8	4	8
4	16	8	16
5	32	16	32
6	64	32	64
7	128	64	128
8	256	128	256
9	512	256	512
10	1024	512	1024
11	2048	1024	2048

Table 7-57. VP Table Size (Per Link) (Continued)

Allocated VPI bits	VP Table Records	VP Table Long Words (without VC Table Lookup)	VP Table Long Words (with VC Table Lookup)
12	4096	2048	4096

See Section 5.1.2.2 for more information.

Table 7-58. VC Sub-Table Size (Per VPI)

Allocated VCI bits	VC Table Records	VC Table Long Words
1 (see note)	2	1
2 (see note)	4	2
3 (see note)	8	4
4 (see note)	16	8
5 (see note)	32	16
6	64	32
7	128	64
8	256	128
9	512	256
10	1024	512
11	2048	1024
12	4096	2048
13	8192	4096
14	16384	8192
15	32768	16384
16	65536	32768

NOTE: ATM standards indirectly require at least 5–6 allocated VCI bits because the smallest VCI values are reserved.

Table 7-59. Multicast Translation Table Size (Per Link)

Allocated Multicast ID bits	Multicast Table Records	Multicast Table Long Words
9	512	256
10	1K	512
11	2K	1K
12	4K	2K
13	8K	4K
14	16K	8K
15	32K	16K
16	64K	32K

Calculations for addressing records of tables in external memory are shown in Table 7-60.

Table 7-60. Addressing External Memory Records

Table	Width	Address Calculation	Parameters
Ingress billing counters	Long word	{IBCTP[23:8], 0000_0000} + {ICI [15:0] * N, 00}	IBCTP: ingress billing counters table pointer from Register ICI: ingress connection identifier from address compression N: number of long words per record (1–4)
Policing counters	Long word	{PCTP[23:8], 0000_0000} + {ICI [15:0] * N, 00}	PCTP: Policing counters table pointer from Register ICI: ingress connection identifier from address compression N: number of long words per record (1–4)
Egress billing counters	Long word	{EBCTP[23:8], 0000_0000} + {ECI [15:0] * N, 00}	ECTP: egress counters table pointer from Register ECI: egress connection identifier from cell overhead bytes or multicast translation N: number of long words per record (1–4)
Flags	Word	{FTP[23:8], 0000_0000} + {ICI [15:0], 00}	FTP: flags table pointer from Register ICI: ingress connection identifier from address compression
Context parameters	Long word	{CPTP[23:8], 0000_0000} + {CI [15:0] * N, 00}	CPTP: context parameters table pointer from Register CI: ECI or ICI as above N: number of long words per record (1–8)
Context parameters extension	Long word	{CEPTP[23:8], 0000_0000} + {CI [15:0] * 1, 00}	CEPTP: Context extension parameters table pointer from Register CI: ECI or ICI as above
VP table (Full Lookup)	Long word	{VPP[23:8], 0000_0000} + {Index[11:0], 00}	VPP: VP pointer from ILNKn register Index: VP Index = VPI from cell masked by VP_MASK from ILNKn
VP table (VC lookup disable)	Word	{VPP[15:0], 0000_0000} + {Index[11:0], 0}	VPP: VP pointer from ILNKn register Index: VP Index = VPI from cell masked by VP_MASK from ILNKn
VC table	Word	{VCTP[23:8], 0000_0000} + {Offset[15:0], 00} + {Index[15:0], 0}	VCTP: VC table pointer from Register Offset: VC Offset from VP table Index: VC Index = VCI from cell masked by VC_MASK from VP table
Multicast translation	Word	{MTTP[23:8], 0000_0000} + {MTTS[3:0], MI [(EMIC+8):0], 0}	MTTP: multicast translation table pointer from register MTTS: multicast translation table section from cell overhead bytes MI: multicast identifier from cell overhead bytes EMIC: egress multicast identifier control from egress multicast configuration register (EMCR)
Buckets	Long word	{BKT_Ptr[21:0], 00}	BKT_Ptr: bucket pointer from context parameters table
OAM table	Long word	{OTP[23:8], 0000_0000} + {OAM_Ptr, 0_0000}	OTP: OAM table pointer from register OAM_Ptr: OAM pointer from context parameters table

Table 7-60. Addressing External Memory Records (Continued)

Table	Width	Address Calculation	Parameters
Dump vector	Long word	{DVTP[23:8], 0000_0000} + {CT [(2*(DVTC+2)):0], 000}	DVTP: dump vector table pointer from register CT: continuous cell time counter DVTC: dump vector table control from ATMC CFB configuration register (ACR)
Ingress link counters	Long word	{ILCTP[23:8], 0000_0000} + {LINK * 5, 00}	ILCTP: ingress link counters table pointer from register LINK: the link from which the cell is received
Egress link counters	Long word	{ELCTP[23:8], 0000_0000} + {LINK * 4, 00}	ELCTP: egress link counters table pointer from register LINK: the link to which the cell is transmitted

7.2.3 Context Parameters Table

The context parameters table consists of specific records for each connection. There are various options regarding the format (size and contents) of each record. These options are programmable on a global basis using the ATMC CFB configuration register (ACR), such that all of the records in this table have the same format. The long words that may be included in a record of the context parameter table are shown in Figure 7-73 in the order in which they appear.

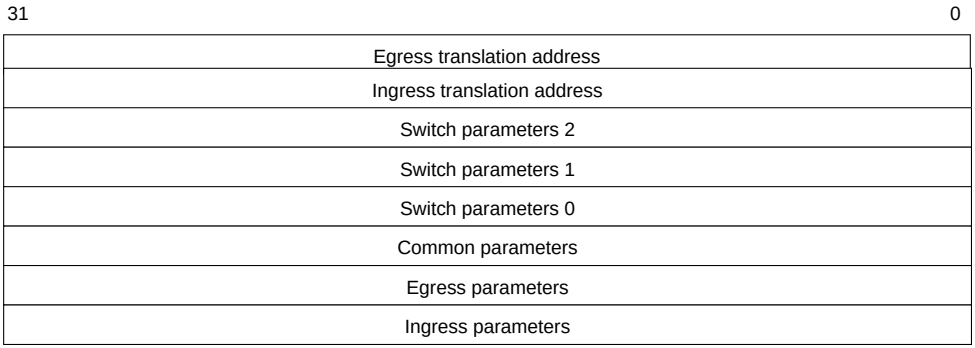


Figure 7-73. Context Parameters Table Record (Full Configuration)

If some of the long words do not exist, the record is shortened, but the order of the long words that remain does not change. An example is shown in Figure 7-74. This chapter describes in detail the long words that make up each record.

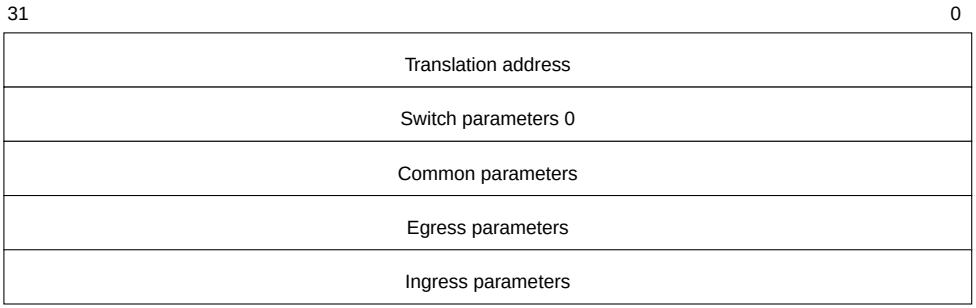


Figure 7-74. Context Parameters Table Record (ATC=01, SPC=01)

7.2.3.1 Egress Translation Address

This long word contains the connection address and should be initialized during the connection setup process. It is used for address translation in the egress cell flow. The long word structure is shown in Figure 7-75.

Figure 7-75. Egress Translation Address Long Word

Table 7-61. Egress Translation Address Long Word Field Descriptions

Bits	Name	Description
31–20	VPI	Virtual path identifier. This field is part of the new address used by the egress address translation mechanism. At a UNI the four most significant bits of this field are the GFC and they are user-defined.
19–4	VCI	Virtual channel identifier. This field is part of the new address used by the egress address translation mechanism.
3–0	LNK	Physical link number. This field is valid only when multiple PHY devices are supported (see Section 7.1.6.3 for details). It determines the physical link to which this connection belongs. If a single PHY device is used, the LNK field should be programmed as 0.

7.2.3.2 Ingress Translation Address

This long word is generally used if ingress address translation is enabled. It contains the new address of the connection and is used for address translation in the ingress cell flow. This long word should be initialized during the connection setup process. The long word structure is shown in Figure 7-76.

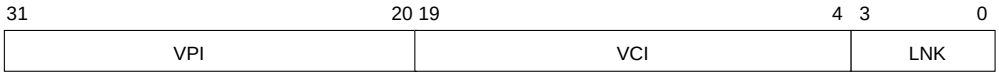


Figure 7-76. Ingress Address Translation Long Word

Table 7-62. Ingress Address Translation Long Word Field Descriptions

Bits	Name	Description
31–20	VPI	Virtual path identifier. This field is part of the new address used by the ingress address translation mechanism.
19–4	VCI	Virtual channel identifier. This field is part of the new address used by the ingress address translation mechanism.
3–0	LNK	Physical link number. This field is valid only when multiple PHY devices are supported (see Section 7.1.6.3 for more information). It determines the physical link to which this connection belongs. If a single PHY device is used, the LNK field should be programmed as 0.

7.2.3.3 Switch Parameters

These optional long words contain the switch parameter records for the connection. Any of the three long words that exist are transferred to the switch interface along with each processed cell. For more details see Section 5.1.8. These long words should be initialized during the connection setup process.

7.2.3.4 Common Parameters

This long word is shared by the ingress and egress processing. It contains static parameters for the connection and should be initialized during the connection setup process. The size and the location of some of the fields is changed according to the PM on all connections (PMAC) bit in the ATMC CFB configuration register (ACR). See Section 7.1.4.8 for details. Figure 7-77 shows the long word fields when the PMAC bit is cleared.

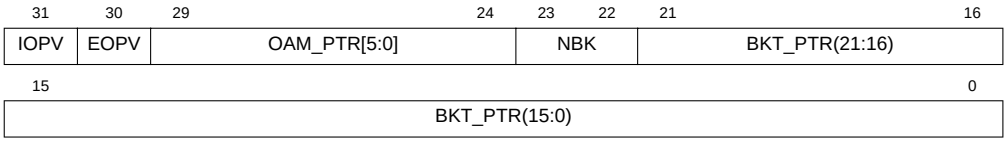


Figure 7-77. Common Parameters Long Word Fields (PMAC = 0)

Figure 7-78 shows the long word field definition when the PMAC bit is set.

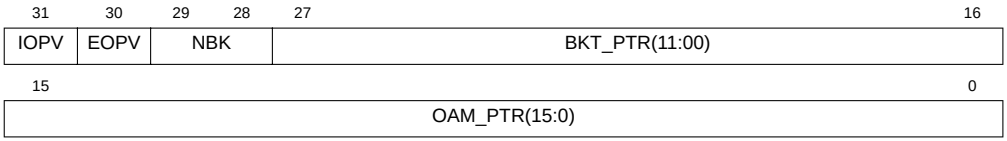


Figure 7-78. Common Parameters Long Word Fields (PMAC = 1)

Table 7-63. PMAC Field Definitions

Bits		Name	Definition
PMAC = 0	PMAC = 1		
31	31	IOPV	Ingress OAM pointer is valid. When IOPV is set, the OAM pointer is valid. Note that even if IOPV is reset, the OAM pointer is still valid if EOPV is set.
30	30	EOPV	Egress OAM pointer is valid. When EOPV is set, the OAM pointer is valid. Note that even if EOPV is reset, the OAM pointer is still valid if IOPV is set.
29–24	15–0	OAM_PTR	OAM pointer. This field is interpreted as an index into the OAM table and used only to support OAM performance management functions. This index is only used if either or both of the ingress OAM pointer valid (IOPV) and the egress OAM pointer valid (EOPV) bits are set (see Section 6.5.6 for details).
23–22	29–28	NBK	Number of buckets. This field defines the number of UPC/NPC leaky buckets that are active on this connection. 00 Four leaky buckets 01 One leaky bucket 10 Two leaky buckets 11 Three leaky buckets Zero leaky buckets (no UPC/NPC) is indicated by all ones in the BKT_PTR field.
21–0	27–16	BKT_PTR	Bucket pointer. This field defines the most significant 22 bits of a 24-bit address to a bucket record in external memory. The bucket record contains information for ingress or egress UPC/NPC processing. A reserved value of all ones is used to indicate that there is no buckets record for this connection. (See Appendix A, “UPC/NPC Design” for more details). When the performance management on all connections (PMAC) bit is set, bits [21:12] of the bucket pointer are located in the common parameters extension word. See Section 7.2.16 for details.

7.2.3.5 Egress Parameters

This long word is used by the egress cell processing. It contains static parameters for the connection and should be initialized during the connection setup process. The long word structure is shown in Figure 7-79.

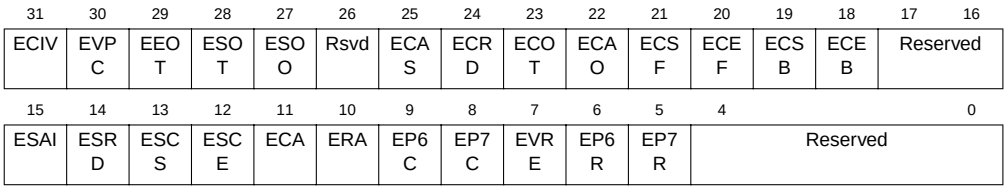


Figure 7-79. Egress Parameters Long Word Fields

Table 7-64. Egress Parameters Long Word Field Descriptions

Bits	Name	Description
31	ECIV	Egress connection identifier valid. This bit controls whether the ESAI, ESRD, ESCS, and ESCE bits are considered for internal scan processing. 0 Not valid 1 Valid
30	EVPC	Egress virtual path connection. This bit defines whether the connection is a virtual channel or virtual path. See Table 6-7. 0 The connection is a virtual channel connection. 1 The connection is a virtual path connection.
29	EEOT	Egress end-to-end OAM termination. When this bit is set, the egress flow is treated as the terminating point of the OAM end-to-end cell flow for the connection. Additionally if the VC RM cell removal point (CRRP) bit is reset then cells with PTI6 or 7 are removed at this point. See Table 6-7.
28	ESOT	Egress segment OAM termination. When ESOT is set, the egress flow is treated as the terminating point of the OAM segment cell flow for the connection.
27	ESOO	Egress segment OAM origin. When ESOO is set, the egress flow is treated as the originating point of the OAM segment cell flow for the connection. Incoming segment OAM cells are removed at this point to prevent contamination of the OAM flows within the segment. See Table 6-7.
26	—	Reserved
25	ECAS	Egress copy received AIS cells. When ECAS is set, OAM end-to-end AIS cells belonging to this connection that are received in the egress cell flow are copied to the cell extraction queue. See Section 6.5.4.1.1 for more details.
24	ECRD	Egress copy received RDI cells. When ECRD is set, OAM end-to-end RDI cells belonging to this connection that are received in the egress cell flow are copied to the cell extraction queue. See Section 6.5.4.1.2 for more details.
23	ECOT	Egress copy received other OAM cells. When ECOT is set, miscellaneous or unidentified OAM cells belonging to this connection that are received in the egress cell flow are copied to the cell extraction queue. See Table 6-7 and Section 6.5.7 for more details.
22	ECAO	Egress copy all received OAM cells. When ECAO is set, all OAM cells belonging to this connection that are received in the egress cell flow are copied to the cell extraction queue. See Table 6-7 and Section 6.5.4.1.1.
21	ECSF	Egress copy segment FMCs. When ECSF is set, OAM segment forward monitoring cells belonging to this connection that are received in the egress cell flow are copied to the cell extraction queue. See Section 6.5.6 and Table 6-12 for more details.
20	ECEF	Egress copy end-to-end FMCs. When ECEF is set, OAM end-to-end forward monitoring cells received in the egress cell flow belonging to the connection are copied to the cell extraction queue. See Section 6.5.6 and Table 6-12 for details.
19	ECSB	Egress copy segment BRCs. When ECSB is set, OAM segment backward reporting cells belonging to this connection that are received in the egress cell flow are copied to the cell extraction queue. See Section 6.5.6 and Table 6-12 for details.
18	ECEB	Egress copy end-to-end BRCs. When ECEB is set, OAM end-to-end backward reporting cells belonging to this connection that are received in the egress cell flow are copied to the cell extraction queue. See Section 6.5.6 and Table 6-12 for details.
17–16	—	Reserved.
15	ESAI	Egress send AIS cell. When ESAI is set, an AIS cell is inserted in the egress cell flow during the internal scan.

Table 7-64. Egress Parameters Long Word Field Descriptions (Continued)

Bits	Name	Description
14	ESRD	Egress send RDI cell. When ESRD is set, an RDI cell is inserted in the egress cell flow during the internal scan.
13	ESCS	Egress send continuity check segment OAM cell. When ESCS is set, a segment continuity check cell is inserted in the egress cell flow during the internal scan.
12	ESCE	Egress send continuity check end-to-end OAM cell. When ESCS is set, an end-to-end continuity check cell is inserted in the egress cell flow during the internal scan.
11	ECA	Egress copy all cells. When ECA is set, all cells belonging to this connection that are received in the egress cell flow are copied to the cell extraction queue.
10	ERA	Egress remove all cells. When ERA is set, all cells belonging to this connection that are received from the switch interface are removed from the egress cell flow.
9	EP6C	Egress PTI 6 copy. When EP6C is set, all cells belonging to this connection with PTI6 that are received in the egress cell flow are copied to the cell extraction queue.
8	EP7C	Egress PTI 7 copy. When EP7C is set, all cells belonging to this connection with PTI7 that are received in the egress cell flow are copied to the cell extraction queue.
7	EVRE	Egress VCR/VRR registers enable. The EVRE bit, when set, enables the copying or removing of cells with reserved VCI values according to the programming of the egress VCI copy register (EVCR) and the egress VCI remove register (EVRR).
6	EP6R	Egress PTI 6 remove. When this bit is set and the VC RM cell removal point (CRRP) bit is set, and then an egress cell whose PTI6 is removed provided that the connection is a VC connection.
5	EP7R	Egress PTI 7 remove. When this bit is set and the VC RM cell removal point (CRRP) bit is set, and then an egress cell whose PTI7 is removed provided that the connection is a VC connection.
4–0	—	Reserved

7.2.3.6 Ingress Parameters

This long word is used by the ingress cell processing. It contains static parameters for the connection and should be initialized during the connection setup process. The long word structure is shown in Figure 7-80.

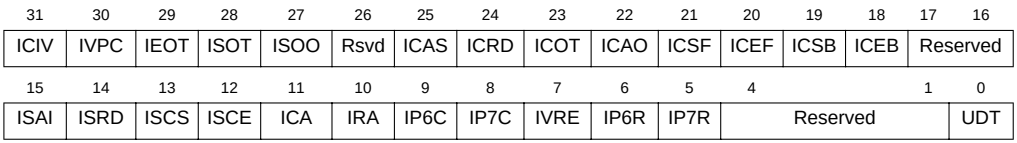


Figure 7-80. Ingress Parameters Long Word Fields

Table 7-65. Ingress Parameters Long Word Field Descriptions

Bits	Name	Description
31	ICIV	Ingress connection identifier valid. This bit controls whether the ISAI, ISRD, ISCS, and ISCE bits are considered for internal scan processing. 0 Not valid 1 Valid
30	IVPC	Ingress virtual path connection. This bit defines whether the connection is a virtual channel or virtual path. See Table 6-7. 0 The connection is a virtual channel connection. 1 The connection is a virtual path connection.
29	IEOT	Ingress end-to-end OAM termination. When IEOT is set, the ingress flow is treated as the terminating point of the OAM end-to-end cell flow for the connection. Also, if the VC RM cell removal point (CRRP) bit is cleared, then cells that belong to a VC connection and whose PTI 6 or 7 are removed at this point. See Table 6-7.
28	ISOT	Ingress segment OAM termination. When ISOT is set, the ingress flow is treated as the terminating point of the connection OAM segment cell flow.
27	ISOO	Ingress segment OAM origin. When ISOO is set, the ingress flow is treated as the originating point of the OAM segment cell flow for the connection. Incoming segment OAM cells are removed at this point to prevent contamination of the OAM flows within the segment. See Table 6-7.
26	—	Reserved
25	ICAS	Ingress copy received AIS cells. When ICAS is set, OAM end-to-end AIS cells belonging to this connection that are received in the ingress cell flow are copied to the cell extraction queue. See Section 6.5.4.1.1 for more details.
24	ICRD	Ingress copy received RDI cells. When ICRD is set, OAM end-to-end RDI cells belonging to this connection that are received in the ingress cell flow are copied to the cell extraction queue. See Section 6.5.4.1.2 for more details.
23	ICOT	Ingress copy received other OAM cells. When ICOT is set, miscellaneous or unidentified OAM cells belonging to this connection that are received in the ingress cell flow are copied to the cell extraction queue. See Table 6-7 and Section 6.5.3.2 for more details.
22	ICAO	Ingress copy all received OAM cells. When ICAO is set, all OAM cells belonging to this connection that are received in the ingress cell flow are copied to the cell extraction queue. See Table 6-7.
21	ICSF	Ingress copy segment FMCs. When ICSF is set, OAM segment forward monitoring cells belonging to this connection that are received in the ingress cell flow are copied to the cell extraction queue. See Section 6.5.6 and Table 6-12 for details.
20	ICEF	Ingress copy end-to-end FMCs. When ICEF is set, OAM end-to-end forward monitoring cells received in the ingress cell flow that belong to this connection are copied to the cell extraction queue. See Section 6.5.6 and Table 6-12 for details.
19	ICSB	Ingress copy segment BRCs. When ICSB is set, OAM segment backward reporting cells belonging to this connection that are received in the ingress cell flow are copied to the cell extraction queue. See Section 6.5.6 and Table 6-12 for details.
18	ICEB	Ingress copy end-to-end BRCs. When ICEB is set, OAM end-to-end backward reporting cells received in the ingress cell flow that belong to this connection are copied to the cell extraction queue. See Section 6.5.6 and Table 6-12 for details.
17–16	—	Reserved
15	ISAI	Ingress send AIS cell. When ISAI is set, an AIS cell is inserted in the ingress cell flow during the internal scan.

Table 7-65. Ingress Parameters Long Word Field Descriptions (Continued)

Bits	Name	Description
14	ISRD	Ingress send RDI cell. When ISRD is set, an RDI cell is inserted in the ingress cell flow during the internal scan.
13	ISCS	Ingress send continuity check segment OAM cell. When ISCS is set, a segment continuity check cell is inserted in the ingress cell flow during the internal scan.
12	ISCE	Ingress send continuity check end-to-end OAM cell. When ISCS is set, an end-to-end continuity check cell is inserted in the ingress cell flow during the internal scan.
11	ICA	Ingress copy all cells. When ICA is set, ALL cells received in the ingress cell flow that belong to this connection are copied to the cell extraction queue.
10	IRA	Ingress remove all cells. When IRA is set, ALL cells received from the PHY that belong to this connection are removed from the ingress cell flow.
9	IP6C	Ingress PTI 6 copy. When IP6C is set, all cells received in the ingress cell flow with PTI6 that belong to the connection are copied to the cell extraction queue.
8	IP7C	Ingress PTI 7 copy. When IP7C is set, all cells received in the ingress cell flow with PTI7 belonging to this connection are copied to the cell extraction queue.
7	IVRE	Ingress VCR/VRR registers enable. The IVRE bit, when set, enables the copying and/or removing of cells with reserved VCI values according to the programming of the ingress VCI copy register (IVCR) and the ingress VCI remove register (IVRR).
6	IP6R	Ingress PTI 6 remove. When this bit is set and the RRP bit is set, then an ingress cell with PTI 6 is removed if the connection is a VC connection.
5	IP7R	Ingress PTI 7 remove. When this bit is set and the RRP bit is set, then an ingress cell with PTI 7 is removed if the connection is a VC connection.
4–1	—	Reserved
0	UDT	UPC/NPC don't touch ()—The UDT bit indicates that the UPC/NPC block should perform the normal operation, but the cell should not be tagged or discarded regardless of the UPC results. When UDT is set, the UPC/NPC counters are incremented, and they indicate the number of cells that would have been tagged or discarded if the cells were processed normally. 0 Process the incoming cell normally. 1 Do not touch the incoming cell.

7.2.4 Ingress Billing Counters Table

This table contains the ingress billing counter records for all the active connections. Details such as when the counters are updated and how and when the microprocessor aggregates them can be found in Section 9.2.3, "Ingress Switch Interface Signals" and Section 3.2.1, "External Memory Maintenance." Each counter wraps to zero after reaching its maximum value. A number of options exist regarding the number of counters and their definition. Note that in each case one and only one counter is incremented for each received cell belonging to this connection. There is also an option to eliminate the ingress billing counters table entirely. The ingress billing counters control (IBCC) field of the ingress processing configuration register (IPCR) is used for programming the various options. If all four counters exist, each record has the structure shown in Figure 7-81.

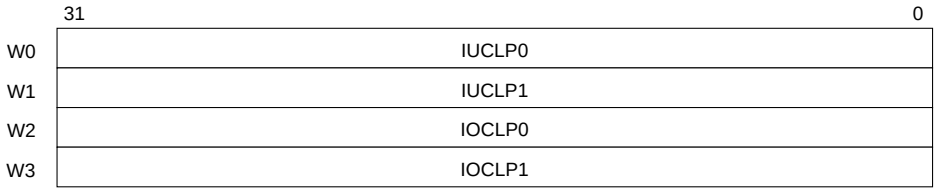


Figure 7-81. Ingress Counters (Full Configuration—IBCC = 001)

- Ingress user CLP0 counter (IUCLP0)—ingress uses the 32-bit counter to count the incoming user (non-OAM) cells from the PHY with cell loss priority (CLP) = 0.
- Ingress user CLP1 counter (IUCLP1)—ingress uses this 32-bit counter to count the incoming user (non-OAM) cells from the PHY with CLP = 1.
- Ingress OAM CLP0 counter (IOCLP0)—ingress uses this 32-bit counter to count the incoming OAM cells from the PHY with CLP = 0.
- Ingress OAM CLP1 counter (IOCLP1)—ingress uses this 32-bit counter to count the incoming OAM cells from the PHY whose CLP = 1.
- If the two OAM counters are combined, each record can be maintained in one of the two forms shown in Figure 7-82 and Figure 7-83:

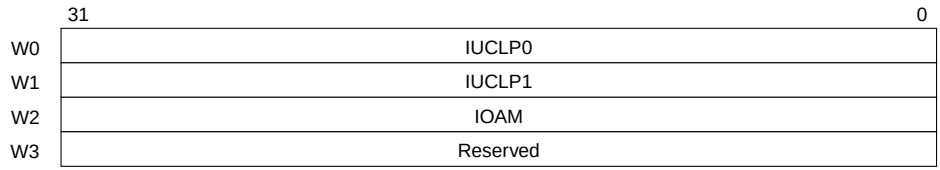


Figure 7-82. Ingress Counters (Single OAM Configuration—IBCC = 011)

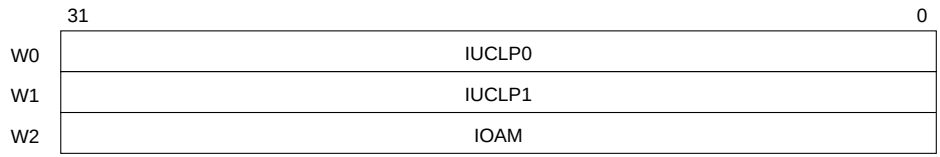


Figure 7-83. Ingress Counters (Single OAM Configuration—IBCC = 010)

- Ingress user CLP0 counter (IUCLP0)—This 32-bit counter is used by the ingress to count the incoming user (non-OAM) cells from the PHY with CLP = 0.
- Ingress user CLP1 counter (IUCLP1)—This 32-bit counter is used by the ingress to count the incoming user (non-OAM) cells from the PHY with CLP = 1.
- Ingress OAM counter (IOAM)—This 32-bit counter is used by the ingress to count the incoming OAM cells from the PHY.

If the two user counters are also combined, each record has the structure shown in Figure 7-84.

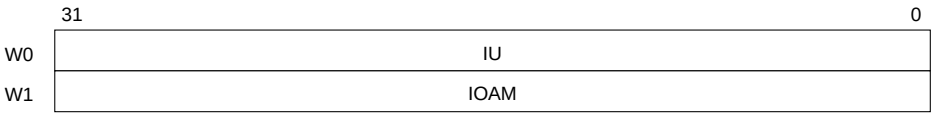


Figure 7-84. Ingress Counters (No CLP Distinction Configuration—IBCC = 100)

- Ingress user counter (IU)—This 32-bit counter is used by the ingress to count the incoming user (non-OAM) cells from the PHY.
- Ingress OAM counter (IOAM)—This 32-bit counter is used by the ingress to count the incoming OAM cells from the PHY.

If the OAM counter is eliminated, each record has the structure shown in Figure 7-85.

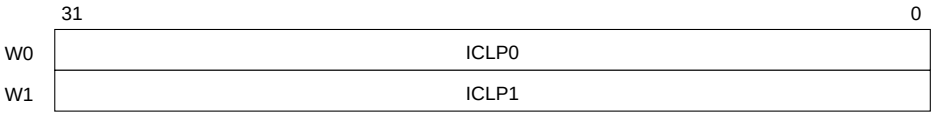


Figure 7-85. Ingress Counters (No OAM Distinction Configuration—IBCC = 101)

- Ingress CLP0 counter (ICLP0)—This 32-bit counter is used by the ingress to count the incoming cells from the PHY whose cell loss priority (CLP) is zero.
- Ingress CLP1 counter (ICLP1)—This 32-bit counter is used by the ingress to count the incoming cells from the PHY whose cell loss priority (CLP) is one.

Finally, all of the counters can be combined into one, in which case each record has the following structure:

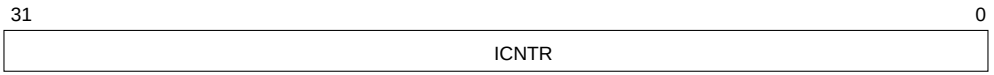


Figure 7-86. Ingress Counters (Single Counter Configuration—IBCC = 110)

- Ingress counter (ICNTR)—This 32-bit counter is used by the ingress to count the incoming cells from the PHY.

7.2.5 Egress Billing Counters Table

This table contains the egress billing counter records for all the active connections. Details such as when the counters are updated and how and when the microprocessor aggregates them can be found in Section 5.2.4, “Performing Context Table Lookups” and Section 3.2.1, “External Memory Maintenance.” Each counter wraps to zero after reaching its

maximum value. A number of options exist regarding the number of counters and their definition. In each case one and only one counter is incremented for each received cell belonging to this connection. There is also an option to eliminate the egress billing counters table entirely. The egress billing counters control (EBCC) field of the egress processing configuration register (EPCR) is used for programming the various options. If all four counters exist, each record has the structure shown in Figure 7-87.

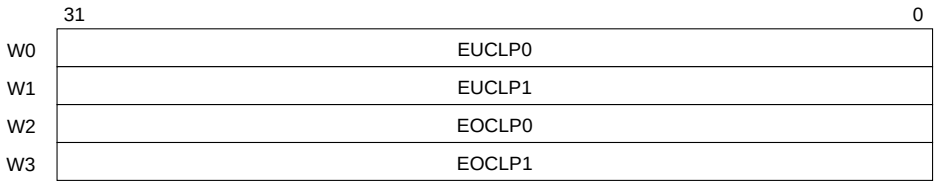


Figure 7-87. Egress Counters (Full Configuration—EBCC = 001)

- Egress user CLP0 counter (EUCLP0)—This 32-bit counter is used by the egress to count the outgoing user (non-OAM) cells whose cell loss priority (CLP) is zero.
- Egress user CLP1 counter (EUCLP1)—This 32-bit counter is used by the egress to count the outgoing user (non-OAM) cells whose cell loss priority (CLP) is one.
- Egress OAM CLP0 counter (EOCLP0)—This 32-bit counter is used by the egress to count the outgoing OAM cells whose cell loss priority (CLP) is zero.
- Egress OAM CLP1 counter (EOCLP1)—This 32-bit counter is used by the egress to count the outgoing OAM cells whose cell loss priority (CLP) is one.

If the two OAM counters are combined, each record can be maintained in one of the following two forms shown in Figure 7-88 and Figure 7-89:

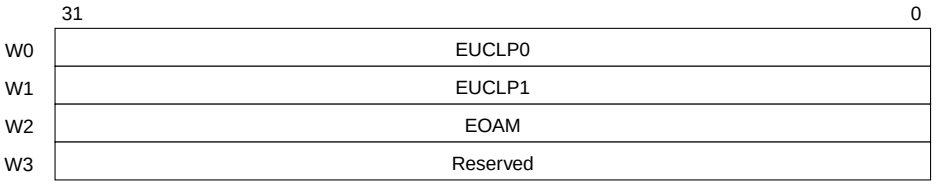


Figure 7-88. Egress Counters (Single OAM Configuration—EBCC = 011)

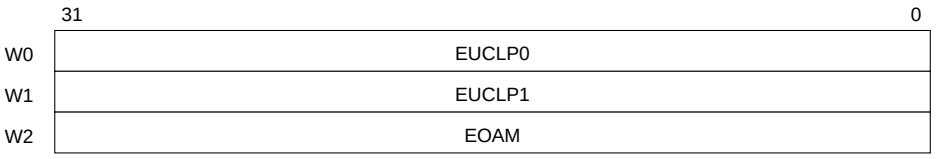


Figure 7-89. Egress Counters (Single OAM Configuration—EBCC = 010)

- Egress user CLP0 counter (EUCLP0)—This 32-bit counter is used by the egress to count the outgoing user (non-OAM) cells whose cell loss priority (CLP) is zero.
- Egress user CLP1 counter (EUCLP1)—This 32-bit counter is used by the egress to count the outgoing user (non-OAM) cells whose cell loss priority (CLP) is one.
- Egress OAM counter (EOAM)—This 32-bit counter is used by the egress to count the outgoing OAM cells.

For two combined user counters, Figure 7-90 shows the structure for each record.

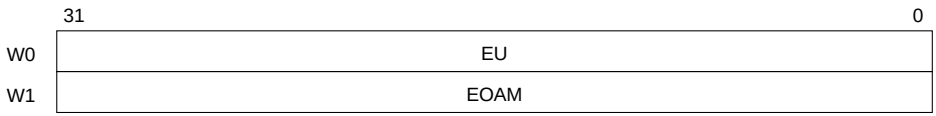


Figure 7-90. Egress Counters (No CLP Distinction Configuration—EBCC = 100)

- Egress user counter (EU)—This 32-bit counter is used by the egress to count the outgoing user (non-OAM) cells.
- Egress OAM counter (EOAM)—This 32-bit counter is used by the egress to count the outgoing OAM cells.

If the OAM counter is eliminated, each record has the structure shown in Figure 7-91.

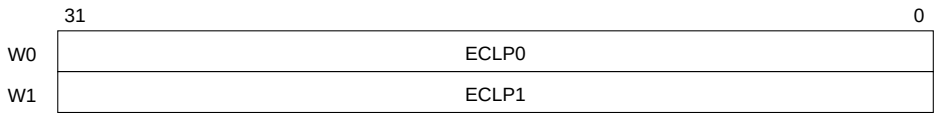


Figure 7-91. Egress Counters (No OAM Distinction Configuration—EBCC = 101)

- Egress CLP0 counter (ECLP0)—This 32-bit counter is used by the egress to count the outgoing cells whose cell loss priority (CLP) is zero.
- Egress CLP1 counter (ECLP1)—This 32-bit counter is used by the egress to count the outgoing cells whose cell loss priority (CLP) is one.

Finally, all of the counters can be combined into one, in which case each record has the structure shown in Figure 7-92.

Figure 7-92. Egress Counters (Single Counter Configuration—EBCC = 110)

- Egress counter (ECNTR)—This 32-bit counter is used by the egress to count the outgoing cells.

7.2.6 Policing Counters Table

This table contains the policing counter records for all the active connections. Details such as when the counters are updated and how and when the microprocessor aggregates them can be found in Section 5.2.5, “Performing UPC/NPC Processing” and Section 3.2.1, “External Memory Maintenance.” Each counter wraps to zero after reaching its maximum value. A number of options exist regarding the number of counters and their definition. Note that in each case at most one policing counter is incremented for each received cell belonging to this connection. There is also an option to eliminate the policing counters table entirely. The ingress policing counters control (IPCC) field of the ingress processing configuration register (IPCR) is used in the case of ingress UPC for programming the various options. The egress policing counters control (EPCC) field of the egress processing configuration register (EPCR) is used in the case of egress UPC for programming the various options.

If all three counters exist, each record has one of the following two forms shown in Figure 7-93 and Figure 7-94:

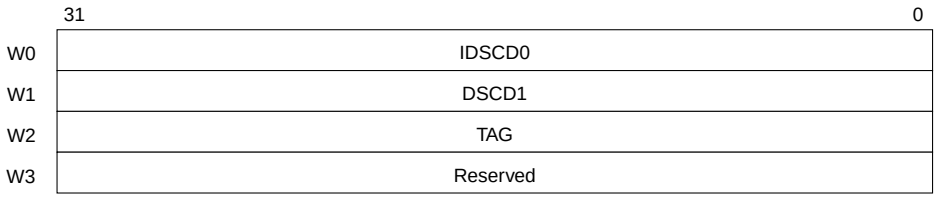


Figure 7-93. Policing Counters (Full Configuration—PCC = 001)

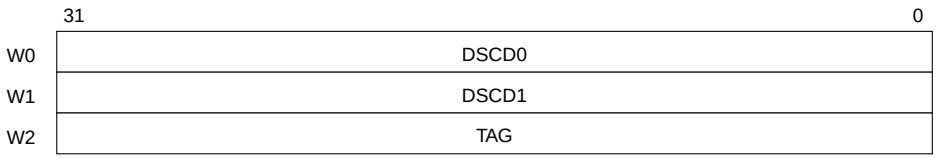


Figure 7-94. Policing Counters (Full Configuration—PCC = 010)

- Ingress discard CLP0 counter (DSCD0)—This 32-bit counter is used by the UPC/NPC to count cells that were discarded (removed from the cell flow) by the enforcer whose cell loss priority (CLP) was zero.
- Ingress discard CLP1 counter (DSCD1)—This 32-bit counter is used by the UPC/NPC to count cells that were discarded (removed from the cell flow) by the enforcer whose cell loss priority (CLP) was one.
- Ingress TAG counter (TAG)—This 32-bit counter is used by the UPC/NPC to count cells that were tagged (CLP was changed from zero to one) by the enforcer.

Figure 7-95 shows the individual record structure if two discard counters are combined.

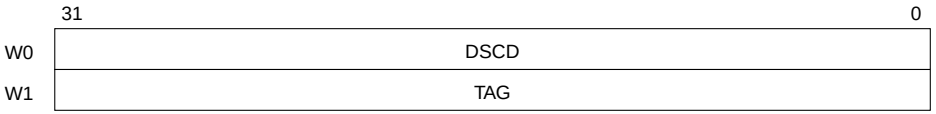


Figure 7-95. Policing Counters (No CLP Distinction Configuration—PCC = 011)

- Ingress discard counter (DSCD)—This 32-bit counter is used by the UPC/NPC to count cells that were discarded (removed from the cell flow) by the enforcer.
- Ingress TAG counter (TAG)—This 32-bit counter is used by the UPC/NPC to count cells that were tagged (CLP was changed from zero to one) by the enforcer.

Options using only the TAG or DSCD counters (as defined above) use individual records with one long word as shown in Figure 7-96 and Figure 7-97.

Figure 7-96. Policing Counters (Only Tag Counter Configuration—PCC = 100)

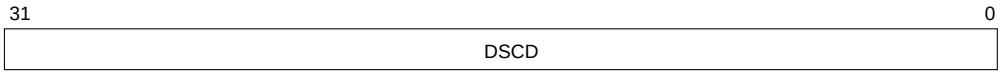


Figure 7-97. Policing Counters (Only Discard Counter Configuration—PCC=101)

7.2.7 Flags Table

This table contains the dynamic flags records for all the active connections. The ATMC is responsible for updating them during the cell flow, and the microprocessor is responsible for reading them on a regular basis. These flags are used mostly for operation and maintenance tasks (fault management tests such as alarm surveillance and continuity check). See Section 6.5.4.1 for more details.

The records of the flags table consist of one long word as follows:

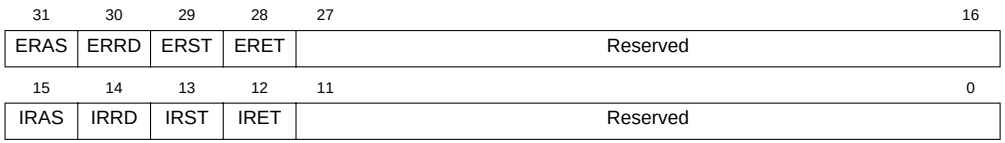


Figure 7-98. Flags Table Fields

Table 7-66. Flags Table Field Descriptions

Bits	Name	Description
31	ERAS	Egress received an AIS cell. This bit is set when an OAM end-to-end AIS cell is received in the egress. This bit remains set until the microprocessor reads the entry and writes back zeros. 0 No AIS cell has been received in the egress cell flow because this bit was cleared by the processor. 1 One or more AIS cells have been received in the egress cell flow because this bit was cleared by the processor.
30	ERRD	Egress received an RDI cell. This bit is set when an OAM end-to-end RDI cell is received in the egress. This bit remains set until the microprocessor reads the entry and writes back zeros. 0 No RDI cell has been received in the egress cell flow because this bit was cleared by the processor. 1 One or more RDI cells have been received in the egress cell flow because this bit was cleared by the processor.
29	ERST	Egress received a segment traffic cell. This bit is set when a user cell or any type of continuity check cell is received in the egress. This bit remains set until the microprocessor reads the entry and writes back zeros. 0 No segment traffic cell has been received in the egress cell flow because this bit was cleared by the processor. 1 One or more segment traffic cells have been received in the egress cell flow because this bit was cleared by the processor.
28	ERET	Egress received an end-to-end traffic cell. This bit is set when a user cell or end-to-end continuity check cell is received in the egress. This bit remains set until the microprocessor reads the entry and writes back zeros. 0 No end-to-end traffic cell has been received in the egress cell flow because this bit was cleared by the processor. 1 One or more end-to-end traffic cells have been received in the egress cell flow because this bit was cleared by the processor.
27–16	—	Reserved
15	IRAS	Ingress received an AIS cell. This bit is set when an OAM end-to-end AIS cell is received in the ingress. This bit remains set until the microprocessor reads the entry and writes back zeros. 0 No AIS cell has been received in the ingress cell flow because this bit was cleared by the processor. 1 One or more AIS cells have been received in the ingress cell flow because this bit was cleared by the processor.
14	IRRD	Ingress received an RDI cell. This bit is set when an OAM end-to-end RDI cell is received in the ingress. This bit remains set until the microprocessor reads the entry and writes back zeros. 0 No RDI cell has been received in the ingress cell flow because this bit was cleared by the processor. 1 One or more RDI cells have been received in the ingress cell flow because this bit was cleared by the processor.
13	IRST	Ingress received a segment traffic cell. This bit is set when a user cell or any type of continuity check cell is received in the ingress. This bit remains set until the microprocessor reads the entry and writes back zeros. 0 No segment traffic cell has been received in the ingress cell flow because this bit was cleared by the processor. 1 One or more segment traffic cells have been received in the ingress cell flow because this bit was cleared by the processor.

Table 7-66. Flags Table Field Descriptions (Continued)

Bits	Name	Description
12	IRET	Ingress received an end-to-end traffic cell. This bit is set when a user cell or end-to-end continuity check cell is received in the ingress. This bit remains set until the microprocessor reads the entry and writes back zeros. 0 No end-to-end traffic cell has been received in the ingress cell flow because this bit was cleared by the processor. 1 One or more end-to-end traffic cells have been received in the ingress cell flow because this bit was cleared by the processor.
11–0	—	Reserved

7.2.8 VP Table

As described earlier, the VP table is used to find the ingress connection identifier (ICI) for VPCs or a pointer to the VC table for VCCs during the address compression process. The table contains up to 4K records for the maximum of 4K VPCs that can be defined on a single link. There are several formats for the VP table record. If VC table lookup is disabled, each record is 16 bits wide and two records are stored in each long word. If VC table lookup is enabled, each record occupies one long word. See Section 5.1.2.2 for more details.

7.2.8.1 VP Table Record without VC Table Lookup

When VC table lookup is not performed (ACM = 01 or 11), each long word entry of the VP table contains the records of two connections in order to save space. Since each record is 16 bits wide, the order of the records within a long word of external memory is reversed if low-endian ordering is being used on the system bus (see Figure 7-99). The MC92520 interprets the long words according to the data order (DO) bit in the microprocessor configuration register (MPCONR).

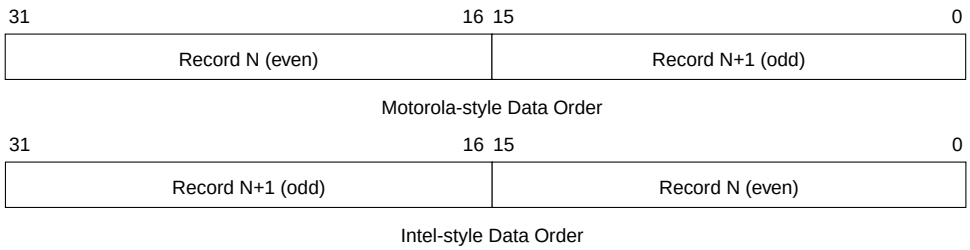


Figure 7-99. Arrangement of 16-Bit Records in External Memory

Each 16-bit record has the structure shown in Figure 7-100.

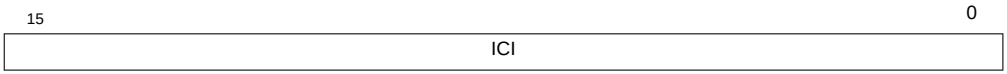


Figure 7-100. VP Table Structure

- Ingress connection identifier (ICI)—This pointer is the index to the connection records in the context parameter table and the counter tables. A value of all 1's indicates that the ICI is not valid.

7.2.8.2 VP Table Record with VC Table Lookup

When VC table lookup is enabled, each long word entry contains the record of one connection. Each record has one of the formats shown in the following figures.

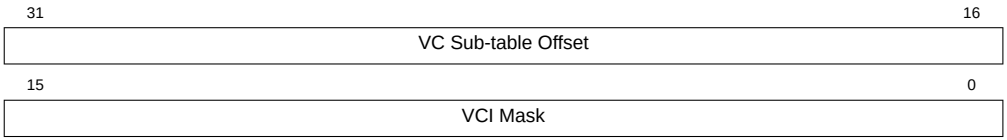


Figure 7-101. VP Table Record Fields with VC Switching

This format shown in Figure 7-101 is used when VC switching is being performed on the connection. It is identified by the VC sub-table offset field containing any value other than all ones.

- VC sub-table offset—This field points to the beginning of this VP's VC sub-table within the VC table. It is in units of long words and is added to the VC table pointer (see Section 7.1.6.19) to obtain the address of the VC sub-table.
- VCI mask—This field indicates which bits of the cell's VCI are used to index the VC connection in the VC sub-table. Each bit of the VCI mask that is set indicates that the corresponding bit in the VCI should be included in the index. The bits that are included are shifted to the right, such that the number of bits in the index is equal to the number of 1's in the VCI mask.

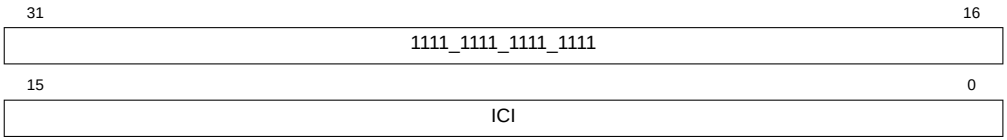


Figure 7-102. VP Table Record Fields with VP Switching

The format shown in Figure 7-102 is used when VP switching is being performed on the connection. It is identified by the 16 most significant bits being set.

- Ingress connection identifier (ICI)—This pointer is the index to the connection records in the context parameter table and the counter tables. A value of all ones indicates that the ICI is not valid, and no entry exists in the context parameters table for this VPI value.

7.2.9 VC Table

As described earlier, the VC table is used to find the ingress connection identifier (ICI) of VCCs during the address compression process (see Section 5.1.2). Each long word entry includes the records of two connections in order to save space. Since each record is 16 bits wide, the order of the records within a long word of external memory is reversed if low-endian ordering is being used on the system bus (see Figure 7-99). The ATMC interprets the long words according to the data order (DO) bit in the microprocessor configuration register (MPCONR). The microprocessor is responsible for initializing the records during the connection set-up process. Each record uses the structure shown in Figure 7-103.

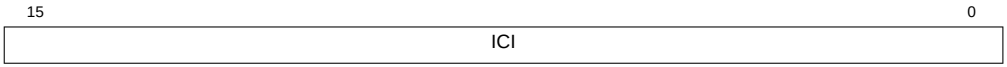


Figure 7-103. VC Record Structure

- Ingress connection identifier (ICI)—This pointer is the index to the connection records in the context parameter table and the counter tables. A value of all 1’s indicates that the ICI is not valid, and no entry exists in the context parameters table for this VPI/VCI value.

7.2.10 Multicast Translation Table

The multicast translation table is used to look up the egress connection identifier (ECI) of cells that have been multicast in the switch. The multicast identifier (MI), described in Section 5.2.2**,** is used as the index to this table. Each long word contains two connection records to save space. Since each record is 16 bits wide, the order of the records within a long word of external memory is reversed if low-endian ordering is being used on the system bus (see Figure 7-99). The ATMC interprets the long words according to the data order (DO) bit in the microprocessor configuration register (MPCONR). Each record has the following structure:

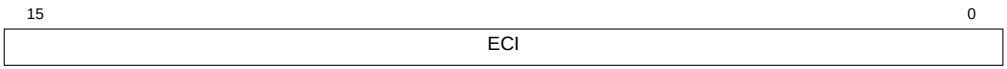


Figure 7-104. Multicast Translation Table Record Structure

- Egress connection identifier (ECI)—This pointer is the index to the connection records in the context parameter table and the counter tables. A value of all ones indicates that the ECI is not valid, and no entry exists in the context parameters table for this MI value.

7.2.11 Buckets Record

As described earlier, the buckets information is organized in records. Each record contains the buckets information of one connection. Since the bucket pointer (BKT_PTR) is a full

address, the buckets records may be distributed anywhere in external memory.

7.2.11.1 Bucket Entries for One Connection

	31 … 0
W0	Time stamp
W1	First bucket information
W2	First bucket information
W3	Second bucket information
W4	Second bucket information
W5	Third bucket information
W6	Third bucket information
W7	Fourth bucket information
W8	Fourth bucket information

Figure 7-105. Bucket Entries

Bucket entry fields are the following:

- Time stamp—This field is used to store the last time at which a cell was admitted in order to be able to calculate the leak from the bucket. This field is mandatory.
- First bucket information—These fields contain the dynamic and static information of a bucket. These fields are mandatory.
- Second, third and fourth bucket information— The NBK field in the context parameters table determines the number of buckets (and fields) being used.

7.2.11.2 Bucket Information

The structure of the first leaky bucket information depends on whether the connection is setup for GFR policing (CPET[EUOM] = b001), or any of the other policing options. The structure of all other leaky bucket information words is common, but in the case of GFR policing, bucket 4 information is interpreted differently.

7.2.11.2.1 GFR Policing Not Selected

If GFR policing is not selected, the structure of each of the four bucket information entries is identical. Bucket words W1, W3, W5, and W7 contain the fields TSC and BKC as shown in Figure 7-106, whereas bucket words W2, W4, W6, and W8 contain the fields LMS, SCP, BKL, TAG, and CAP as shown in Figure 7-107. See also Section A.2 in Appendix A, “UPC/NPC Design” for more details.

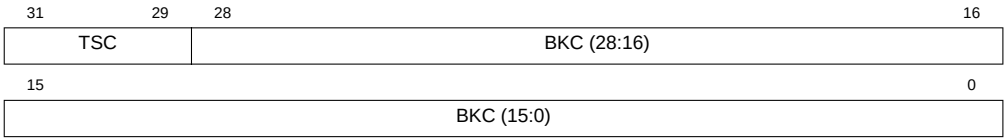


Figure 7-106. Bucket Information Word 1, 3, 5, and 7 Fields - GFR Policing Not Selected

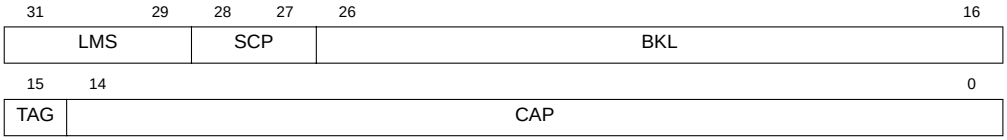


Figure 7-107. Bucket Information Word 2, 4, 6, and 8 Fields - GFR Policing Not Selected

Table 7-67. Word 1, 3, 5, 7 Field Descriptions

Bits	Name	Description
31–29	TSC	Timescale. The eight timescale values are used for encoding the BKC and CAP fields by moving the binary point in a way that keeps the UPC error to a minimum. See Section A.6 for details.
28–0	BKC	Bucket contents. This field stores the contents of a bucket in units of cell slots and is updated dynamically whenever a cell is admitted.

Table 7-68. Word 2, 4, 6, 8 Field Descriptions

Bits	Name	Description
31–29	LMS	Limit shift The limit shift is used for encoding the BKL by moving the binary point. See Section A.6 for details.
28–27	SCP	Scope. This field defines the scope on which the enforcer should work (on CLP0, CLP1, or both). 00 No enforcement by this bucket. 01 Enforcement only on cells with CLP0. 10 Enforcement only on cells with CLP1. 11 Enforcement on both CLPs.
26–16	BKL	Bucket limit. The bucket limit is used by the enforcer to limit the burstiness of the channel. It is defined as one cell arrival period less than the bucket size. See Section A.6 for details.
15	TAG	This bit defines if a violating cell is tagged (set CLP to one) or discarded (removed from the cell flow). 0 Violating cell is discarded. 1 Violating cell is tagged.
14–0	CAP	Cell arrival period. This field represents the bandwidth of the connection by defining the average time (in cell slots) between cells.

7.2.11.2.2 GFR Policing Selected

If GFR policing is selected, the structure of the first bucket is different and each of the three remaining bucket information entries are structured identically to those in which GFR policing is not selected.

As in the discussion of GFR policing not being selected (Section 7.2.11.2.1), bucket words W3, W5, and W7 have the same structure and contain the fields TSC and BKC. Likewise, bucket words W4, W6, and W8 have the same structure and contain the fields LMS, SCP, BKL, TAG, and CAP. GFR policing, however, also uses definitions from the common parameter extension table (CPET) and the GFR configuration register (GFRCR) to supplement the interpretation of information stored in buckets 2, 3, and 4. That is, although the structure of the information is the same as for non-GFR policing, the effect of some field values may be different. For detailed information on GFR support and configuration see Section 6.3, “Guaranteed Frame Rate (GFR) Support.”

For GFR policing, bucket words W1 and W2 have the following structure:

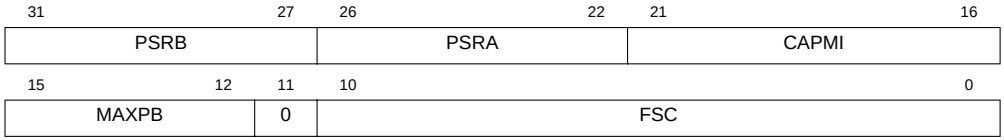


Figure 7-108. Bucket Information Word 1 Fields - GFR Policing Selected

Table 7-69. Bucket Information Word 1 Field Descriptions

Bits	Name	Description
31–27	PSRB	Probability step resolution for FCRB. The value of the PSRB field defines how the random frame drop (RFD) probability function of the FCRB enforcer changes in relation to the shared FCRA and FCRB bucket content. For more information see Section 6.3.1.8.3, “Configuration of RFD Probability Function Limits.”
26–22	PSRA	Probability step resolution for FCRA. The value of the PSRA field is defined as the PSRB field defined above and used for RFD in the FCRA enforcer stage.
21–16	CAPMI	Cell arrival period multiplier index. The value of the CAPMI field is an index into the cell arrival period multiplier (CAPMnn) register array. A value of 0 refers to CAPM00, a value of 1 to CAPM01, and so on. The value stored in the CAPMnn register identified by CAPMI is used to dynamically calculate the effective cell arrival period (CAP) used to update the FCRA and FCRB leaky bucket fields BKL_A and BKL_B. For more information see Section 6.3.1.7, “FCR Administration” and Section 7.1.5.24, “Cell Arrival Period Multiplier Registers 0 - 63 (CAPMnn).”
15–12	MAXPB	RFD maximum probability. The MAXPB field value, associated with a maximum BKL threshold value, defines the highest probability applied to discard or tag frames due to random frame drop (RFD) by one of the FCR enforcer stages. The probability cut-off point, $Prob_{max}$, is expressed in powers of 2, $Prob_{max} = 2^{-MAXPB,}$ and ranges from 2^{-0} (1.0) to 2^{-15} (0.00003). **Note**, MAXPB must express a probability greater than or equal to the one expressed by MINPB defined below.
11	—	Reserved
10–0	FSC	Frame size count. The frame size count is used to keep track of how many cells have been received for the current frame. The count is maintained in terms of cells and used by the MFS enforcer. The FSC field is maintained by the MC92520.

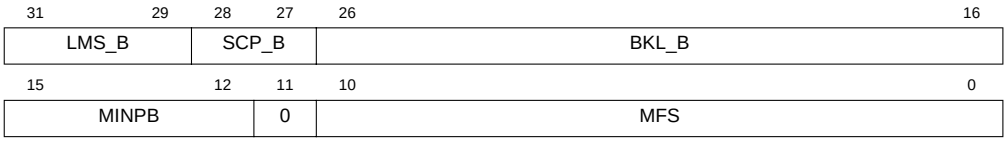


Figure 7-109. Bucket Information Word 2 Fields - GFR Policing Selected

Table 7-70. Bucket Information Word 2 Field Descriptions

Bits	Name	Description
31–29	LMS_B	Limit shift for FCRB. The limit shift is used for encoding the BKL_B by moving the binary point. See Section A.6 for details.
28–27	SCP_B	Action scope for FCRB. This field defines the action scope of the FCRB enforcer. 00 No action. 01 EPD on CLP = 0 frames. 10 EPD on user-marked CLP = 1 frames. 11 EPD on both user-marked and UPC-tagged CLP = 1 frames.
26–16	BKL_B	Bucket limit for FRCB. The bucket limit is used by the FCRB enforcer. It is defined as one cell arrival period less than the bucket size. See Section 6.3.1.6, “Fair Cell Rate (FCR)” and Section A.6 for details.
15–12	MINPB	RFD minimum probability. The MINPB field value, associated with a minimum BKL threshold value, defines the lowest probability applied to discard or tag frames due to random frame drop (RFD) by one of the FCR enforcer stages. The probability cut-off point $Prob_{min}$ is expressed in powers of 2, $Prob_{min} = 2^{-MINPB}$ and ranges from 2^{-1} (0.5) to 2^{-15} (0.00003). Note, MINPB must express a probability less than or equal to the one expressed by MAXPB defined above. Programming a value of 0 in the MINPB field will disable all RFD functions.
11	—	Reserved
10–0	MFS	Maximum frame size. The maximum size of a frame that will be admitted by the MFS enforcer. The MFS field is specified in terms of cells. MFS policing is disabled by a value of 0. If enabled, all non-end cells that would bring the admitted cell count (stored in FSC) to MFS are discarded, that is, the policing action is PPD and the last cell of a frame is always admitted (unless discarded due to other enforcer actions).

Furthermore, GFR policing interprets the bucket words W4, W6, W7, and W8 as follows:

W4 [TAG] and W6 [TAG] — The TAG bit defines whether the enforcer tags or discards. But the selected action is further refined by the setting of the B2EPD and B3EPD bits in the GFR configuration register (GFRCR). If the B*n*EPD bit is clear, the setting of the TAG bit in bucket n defines that either PPD or PPT is selected. If the BnEPD bit is set, the TAG bit defines that EPD or EPT is selected as policing action.

W7 [BKC] — The BKC field value of bucket 4 is shared by both FCRA and FCRB enforcer stages. The fill scope for BKC is not defined by W8[SCP], but the B4FS field of the GFR configuration register (GFRCR).

W8 [SCP] and W8 [TAG] —The SCP field and TAG bit values of bucket 4 are used in combination to define the action scope of FCRA. The following action scopes are supported (TAG + SCP):

0 + 00 = No action.

0 + 01 = EPD on CLP = 0 frames.

0 + 10 = EPD on user-marked CLP = 1 frames.

0 + 11 = EPD on both user-marked and UPC-tagged CLP = 1 frames.

1 + 00 = No action.

1 + 01 = EPT on CLP = 0 frames.

1 + 10 = EPT on CLP = 0 frames and EPD on user-marked CLP = 1 frames.

1 + 11 = EPT on CLP = 0 frames and EPD on both user-marked and UPC-tagged CLP = 1 frames.

The FCRA enforcer action is further modified by the setting of the B4APD bit in the GFR configuration register (GFRCR). If B4APD is set, the FCRA enforcer will discard all frames exceeding the bucket limit, independent of the selected action scope, or the CLP bit value of the frame. If B4APD is clear, CLP = 0 frames are admitted even if the cells of the frame force the bucket content above the specified limit.

W8 [CAP] —The effective CAP of a GFR connection is dynamically calculated by multiplying the content of the W8[CAP] field with a multiplier defined through the CAPM*nn* register array and identified by W1b[CAPMI] described above.

7.2.12 OAM Table

Each OAM table record consists of eight long words and contains the fields necessary for running a bidirectional performance monitoring test on a connection. The OAM pointer in a connection context parameters table record is used as an index to the OAM table record relating to the connection. Each record is logically divided into two parts: an egress area (consisting of the first four long words: E0–E3) and an ingress area (consisting of the last four long words: I0–I3), as illustrated in Figure 7-110. At most, one of the two areas may be used as an originating point (where forward monitoring cells are generated).

	31 24	23 16	15 0
W0 = E0	Control bits	MCSN	TUC
W1 = E1	BEDC		TUC0
W2 = E2	Reserved	BMCSN	ECI
W3 = E3	Reserved	LMCSN	TUCD
W4 = I0	Control bits	MCSN	TUC
W5 = I1	BEDC		TUC0
W6 = I2	Reserved	BMCSN	ICI
W7 = I3	Reserved	LMCSN	TUCD

Figure 7-110. OAM Table Record

- MCSN—The MC92520 uses this field for storing the monitoring cell sequence number of the next forward monitoring cell.
- TUC—The MC92520 uses this field for storing the running total user cell count of the current block.
- BEDC—The MC92520 uses this field for storing the running block error detection code (BIP-16) of the current block of user cells.
- TUC0—The MC92520 uses this field for storing the running total CLP=0 user cell count of the current block.
- BMCSN—The MC92520 uses this field for storing the monitoring cell sequence number of the next backward reporting cell.
- ECI/ICI—This field contains the connection identifier of the connection on which the performance monitoring test is being performed. In the egress area this is the egress connection identifier, and in the ingress area it is the ingress connection identifier. This field is initialized by the user when the test is set up.
- LMCSN—The MC92520 uses this field for storing the monitoring cell sequence number of the previous FMC. This field does not require initialization by the user.
- TUCD—The MC92520 uses this field for storing the difference between the total user cell counts when the previous FMC arrived. This field does not require initialization by the user.
- Control bits—Control bits for the egress and ingress performance monitoring test are shown in Figure 7-111.

31	30	29 28	27	26	25	24
FMCG	SEG	BLK	TR	FCLP	F4	Rsvd

Figure 7-111. Control Bits

Table 7-71 describes the performance monitoring control bit fields:

Table 7-71. Control Bit Field Descriptions

Bits	Name	Description
31	FMCG	Forward monitoring cell generation. The FMCG bit indicates that this is the originating point of the performance monitoring test, and FMCs should be generated here. 0 FMCs should not be generated. 1 FMCs should be generated.
30	SEG	Segment. The SEG bit determines whether performance monitoring is performed on the entire connection or only on a connection segment. 0 The performance monitoring test is being performed on an entire connection. Generated monitoring cells should be end-to-end OAM cells. 1 The performance monitoring test is being performed on a connection segment. Generated monitoring cells should be segment OAM cells.
29–28	BLK	Block size. This field defines the block size for the performance monitoring test. It should be initialized by the user and is not changed by the MC92520. 00 128 cells. 01 256 cells. 10 512 cells. 11 1024 cells.
27	TR	Test running. This bit indicates that the test is currently running. It should be reset when the record is initialized. It is set by the MC92520 when an FMC is processed. It may be reset by the processor in case of a failure. See Section 6.5.6 for details.
26	FCLP	FMC CLP bit. This bit is used as the CLP bit in the header of generated FMCs. It should be initialized by the user and is not changed by the MC92520.
25	F4	This bit determines whether performance monitoring is performed at the F4 or F5 level. 0 The performance monitoring test is being performed at the F5 level. 1 The performance monitoring test is being performed at the F4 level.
24	—	Reserved

7.2.13 Dump Vector Table

The dump vector table contains a trace of the activities of the MC92520. The MC92520 writes one record per cell time to the dump vector table. Each record consists of two long words, one for the egress processing (shown in Figure 7-112) and one for the ingress processing (shown in Figure 7-113). The size of the dump vector table is determined by the dump vector table control (DVTC) field of the ATMC CFB configuration register (ACR). When the end of the table is reached, the write pointer wraps around to the beginning of the table and the records are overwritten.

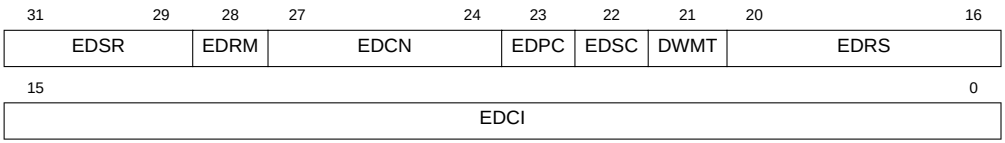


Figure 7-112. Dump Vector Table Egress Long Word Fields

Table 7-72. Dump Vector Table Egress Long Word Field Descriptions

Bits	Name	Description
31–29	EDSR	Egress dump cell source. The EDSR field specifies the source of the egress-processed dumped cell. See Section 7.3.2.1 for the field definition.
28	EDRM	Egress dump removed. The EDRM bit, when set, indicates that the cell was removed from the egress cell flow.
27–24	EDCN	Egress dump cell name. The EDCN field specifies the type of cell that was processed. See Section 7.3.3.1 for the field definition.
23	EDPC	Egress dump primary copy. When set, this bit indicates that the primary cell copied to the cell extraction queue from the egress during the previous cell time was copied to the MPI and read by the microprocessor.
22	EDSC	Egress dump secondary copy. When set, this bit indicates that the MPI did not filter out the secondary cell copied to the cell extraction queue from the egress during the previous cell time and was read by the microprocessor.
21	DWMT	Dump was maintenance. The DWMT bit indicates that the previous cell time was a maintenance slot. Therefore, the previous record is invalid.
20–16	EDRS	Egress dump action reason. The EDRS field specifies the conclusion of the cell processing. If the cell is copied to the microprocessor, this value is the reason that the cell was copied. See Section 7.3.3.2 for the field definition.
15–0	EDCI	Egress dump connection identifier. The EDCI field contains the ECI of the processed cell.

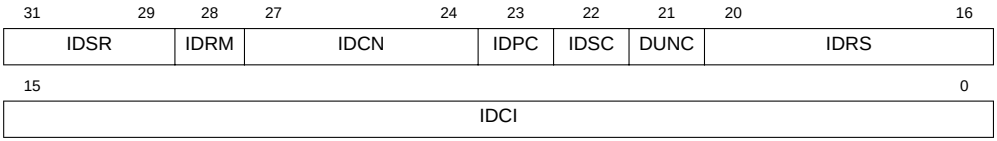


Figure 7-113. Dump Vector Table Ingress Long Word Fields

Table 7-73. Dump Vector Table Ingress Long Word Field Descriptions

Bits	Name	Description
31–29	IDSR	Ingress dump cell source. The IDSR field specifies the source of the cell processed in the ingress cell flow. See Section 7.3.2.1 for the field definition.
28	IDRM	Ingress dump removed. The IDRM bit, when set, indicates that the cell was removed from the ingress cell flow.
27–24	IDCN	Ingress dump cell name. The IDCN field specifies the type of cell that was processed. See Section 7.3.3.1 for the field definition.
23	IDPC	Ingress dump primary copy. The IDPC bit, when set, indicates that the primary cell copied to the cell extraction queue from the ingress cell flow during the previous cell time was copied to the MPI and is actually read by the microprocessor.
22	IDSC	Ingress dump secondary copy. The IDSC bit, when set, indicates that the secondary cell copied to the cell extraction queue from the ingress cell flow during the previous cell time was not filtered out by the MPI and is actually read by the microprocessor.
21	DUNC	Dump UPC non-conforming. The DUNC bit indicates that the UPC/NPC mechanism found the cell to be non-conforming.

Table 7-73. Dump Vector Table Ingress Long Word Field Descriptions

Bits	Name	Description
20–16	IDRS	Ingress dump action reason. The IDRS field specifies the conclusion of the cell processing. If the cell is copied to the microprocessor, this value is the reason that the cell was copied. See Section 7.3.3.2 for the field definition.
15–0	IDCI	Ingress dump connection identifier. The IDCI field contains the ICI of the processed cell.

7.2.14 Ingress Link Counters Table

This table contains the ingress link counter records. Each counter wraps to zero after reaching its maximum value. Figure 7-114 shows the record structure.

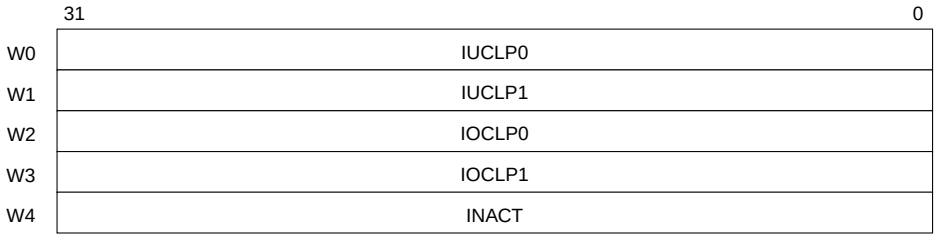


Figure 7-114. Ingress Link Counter Record

- Ingress user CLP0 counter (IUCLP0)—This 32-bit counter is used to count the incoming user (non-OAM) cells from the PHY whose cell loss priority (CLP) is zero.
- Ingress user CLP1 counter (IUCLP1)—This 32-bit counter is used to count the incoming user (non-OAM) cells from the PHY whose cell loss priority (CLP) is one.
- Ingress OAM CLP0 counter (IOCLP0)—This 32-bit counter is used to count the incoming OAM cells from the PHY whose cell loss priority (CLP) is zero.
- Ingress OAM CLP1 counter (IOCLP1)—This 32-bit counter is used to count the incoming OAM cells from the PHY whose cell loss priority (CLP) is one.
- Inactive cell counter (INACT)—This 32-bit counter is used to count the incoming cells from the PHY on which the address compression algorithm failed to produce a valid ICI value.

7.2.15 Egress Link Counters Table

This table contains the egress link counter records. Each counter wraps to zero after reaching its maximum value. Figure 7-115 shows the record structure.

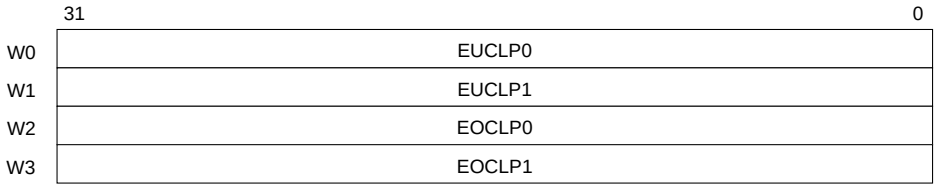


Figure 7-115. Egress Link Counter Record

- Egress user CLP0 counter (EUCLP0)—This 32-bit counter is used to count the outgoing user (non-OAM) cells to the PHY whose cell loss priority (CLP) is zero.
- Egress user CLP1 counter (EUCLP1)—This 32-bit counter is used to count the outgoing user (non-OAM) cells to the PHY whose cell loss priority (CLP) is one.
- Egress OAM CLP0 counter (EOCLP0)—This 32-bit counter is used to count thee outgoing OAM cells to the PHY whose cell loss priority (CLP) is zero.
- Egress OAM CLP1 counter (EOCLP1)—This 32-bit counter is used to count the outgoing OAM cells to the PHY whose cell loss priority (CLP) is one.

7.2.16 Context Parameters Extension Table

Each context parameters extension table record contains one common parameters extension word. There are four modes for this word (shown in Figure 7-116 through Figure 7-119) depending on the values of the PM on all connections (PMAC) bit and the UPC flow (UPCF) bit in the ATMC CFB configuration register (ACR)

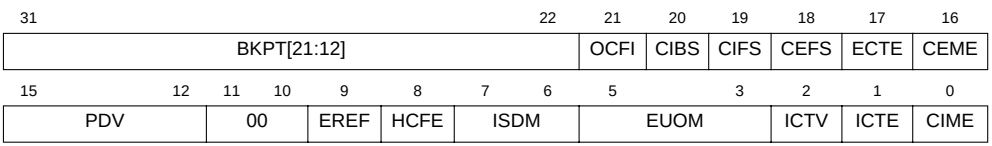


Figure 7-116. Parameter Extension Word Fields (PMAC = 1 and UPCF = 0)

Table 7-74. PMAC = 1 and UPCF = 0 Field Descriptions

Bits	Name	Description
31–22	BKPT[21:12]	Bucket pointer. When the performance management on all connections (PMAC) bit is set, this field contains bits [21:12] of the bucket pointer. When the PMAC is cleared, this field is reserved and should equal 0.
21	OCFI	Overhead CLP from input. If ingress CLP transparency support is enabled via the ICTE bit (described below), the OCFI bit determines whether the CLP bit in the cell overhead is set based on the CLP bit input to or based on the CLP bit output of the UPC enforcer stages. Note, storing a CLP bit in the overhead only works on the ingress flow, when ACR[UPCF] is clear. 0 Set overhead CLP bit value based on UPC enforcer output. 1 Set overhead CLP bit value based on UPC enforcer input.

Table 7-74. PMAC = 1 and UPCF = 0 Field Descriptions (Continued)

Bits	Name	Description
20	CIBS	CLP inconsistency behavior selection. If frame-based UPC (GFR) is enabled (EUOM = b001), the CIBS bit determines whether CLP consistency within a frame is enforced or ignored. If enabled, CLP consistency is enforced via partial packet discard (PPD). See Section 6.3.1.2, “Cell Loss Priority (CLP) Consistency” for details.
19	CIFS	Connection ingress flow status. The MC92520 copies the overhead ingress flow status (IFS) bit to this bit. This bit is used by the ingress processing block for ABR cell marking and is, therefore, intended for MC92520 internal use. See Section 6.4.2.1 for details.
18	CEFS	Connection egress flow status. The MC92520 copies the overhead egress flow status (EFS) bit to this bit. This bit is used by the ingress processing block for ABR cell marking and is, therefore, intended for MC92520 internal use. See Section 6.4.2.2 for details.
17	ECTE	Egress CLP transparency enable. This bit determines whether CLP should be copied from the egress overhead CLP (EOCLP) bit to the cell header. See Section 6.4 for details. 0 CLP should not be copied from the switch overhead to the cell header. 1 CLP should be copied from the switch overhead to the cell header.
16	CEME	Connection egress marking enable. This bit enables marking of cells that are received in the egress. See Section 6.4.2.4 for details. 0 Marking of cells that are received in the egress is disabled 1 Marking of cells that are received in the egress is enabled.
15–12	PDV	Packet discard variables. This field is accessed only by the MC92520.
11–10	—	Reserved
9	EREF	Egress reset EFCI. This bit determines if PTI[1] of an egress cell is reset. 0 PTI[1] of an egress cell is not reset. 1 PTI[1] of an egress cell is to be reset.
8	HCFE	Header CLP from enforcer. If ingress CLP transparency support is enabled via the ICTE bit (described below), the HCFE bit determines whether the CLP bit in the cell header is set based on the result of the UPC enforcer stages or based on the value defined by the ICTV bit (also described below). 0 Set header CLP bit value from ICTV bit configuration. 1 Set header CLP bit value from UPC enforcer output.
7–6	ISDM	Ingress selective discard operation mode. This field determines the selective discard operation mode. See Section 6.1.3 for details. 00 No selective discard. 01 Reserved. 10 Selective discard on CLP1 flow. 11 Selective discard on CLP0 +1 flow.
5–3	EUOM	This field determines the UPC operation mode. 000 Cell-based UPC. 001 Frame-based UPC (GFR). See Section 6.3 for details. 010 Partial packet discard (PPD). See Section 6.1.2.1 for details. 100 Early packet discard (EPD). See Section 6.1.2.2 for details. 110 Limited EPD. See Section 6.1.2.3 for details.
2	ICTV	Ingress CLP transparency value. This bit determines the value that should be written to a cell’s header if the ingress CLP Transparency enable (ICTE) bit is set. See Section 6.2.1 for details.

Table 7-74. PMAC = 1 and UPCF = 0 Field Descriptions (Continued)

Bits	Name	Description
1	ICTE	Ingress CLP transparency enable. This bit determines whether CLP should be copied to the ingress overhead CLP (IOCLP) bit and whether the ingress CLP transparency value (ICTV) bit should be written to the cell header CLP. See Section 6.2.1 for details. 0 The ingress header CLP bit is not touched. 1 CLP should be copied from the cell header to the ingress switch parameters. The ICTV bit should be written to the cell header CLP.
0	CIME	Connection ingress marking enable. This bit enables marking of cells that are received in the ingress. See Section 6.4.2.3 for details. 0 Marking of cells that are received in the ingress is disabled. 1 Marking of cells that are received in the ingress is enabled.

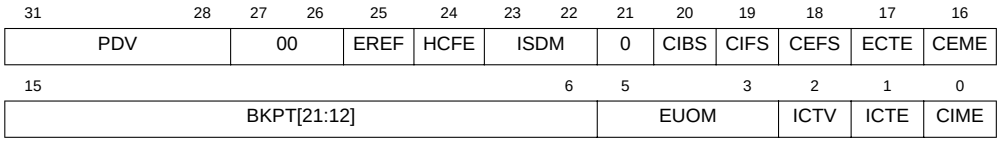


Figure 7-117. Parameter Extension Word Fields (PMAC = 1 and UPCF = 1)

Table 7-75. PMAC = 1 and UPCF = 1 Field Descriptions

Bits	Name	Description
31–28	PDV	Packet discard variables. This field is accessed only by the MC92520.
27–26	—	Reserved
25	EREF	Egress reset EFCI. This bit determines if PTI[1] of an egress cell is reset. 0 PTI[1] of an egress cell is not reset. 1 PTI[1] of an egress cell is to be reset.
24	HCFE	Header CLP from enforcer. If ingress CLP transparency support is enabled via the ICTE bit (described below), the HCFE bit determines whether the CLP bit in the cell header is set based on the result of the UPC enforcer stages or based on the value defined by the ICTV bit (also described below). 0 Set header CLP bit value from ICTV bit configuration. 1 Set header CLP bit value from UPC enforcer output.
23–22	ISDM	Ingress selective discard operation mode. This field determines the selective discard operation mode. See Section 6.1.3 for details. 00 No selective discard. 01 Reserved. 10 Selective discard on CLP1 flow. 11 Selective discard on CLP0 +1 flow.
21	—	Reserved
20	CIBS	CLP inconsistency behavior selection. If frame-based UPC (GFR) is enabled (EUOM = b001), the CIBS bit determines whether CLP consistency within a frame is enforced or ignored. If enabled, CLP consistency is enforced via partial packet discard (PPD). See Section 6.3.1.2, “Cell Loss Priority (CLP) Consistency” for details.
19	CIFS	Connection ingress flow status. The MC92520 copies the overhead ingress flow status (IFS) bit to this bit. This bit is used by the ingress processing block for ABR cell marking and is, therefore, intended for MC92520 internal use. See Section 6.4.2.1 for details.

Table 7-75. PMAC = 1 and UPCF = 1 Field Descriptions (Continued)

Bits	Name	Description
18	CEFS	Connection egress flow status. The MC92520 copies the overhead egress flow status (EFS) bit to this bit. This bit is used by the ingress processing block for ABR cell marking and is, therefore, intended for MC92520 internal use. See Section 6.4.2.2 for details.
17	ECTE	Egress CLP transparency enable. This bit determines whether CLP should be copied from the egress overhead CLP (EOCLP) bit to the cell header. See Section 6.2 for details. 0 CLP should not be copied from the switch overhead to the cell header. 1 CLP should be copied from the switch overhead to the cell header.
16	CEME	Connection egress marking enable. This bit enables marking of cells that are received in the egress. See Section 6.4.2.4 for details. 0 Marking of cells that are received in the egress is disabled 1 Marking of cells that are received in the egress is enabled.
15–6	BKPT[21:12]	Bucket pointer. When the performance management on all connections (PMAC) bit is set, this field contains bits [21:12] of the bucket pointer. When the PMAC is cleared, this field is reserved and should equal 0.
5–3	EUOM	This field determines the UPC operation mode. 000 Cell-based UPC. 001 Frame-based UPC (GFR). See Section 6.3 for details. 010 Partial packet discard (PPD). See Section 6.1.2.1 for details. 100 Early packet discard (EPD). See Section 6.1.2.2 for details. 110 Limited EPD. See Section 6.1.2.3 for details.
2	ICTV	Ingress CLP transparency value. This bit determines the value that should be written to a cell's header if the ingress CLP Transparency enable (ICTE) bit is set. See Section 6.2.1 for details.
1	ICTE	Ingress CLP transparency enable. This bit determines whether CLP should be copied to the ingress overhead CLP (IOCLP) bit and whether the ingress CLP transparency value (ICTV) bit should be written to the cell header CLP. See Section 6.2.1 for details. 0 The ingress header CLP bit is not touched. 1 CLP should be copied from the cell header to the ingress switch parameters. The ICTV bit should be written to the cell header CLP.
0	CIME	Connection ingress marking enable. This bit enables marking of cells that are received in the ingress. See Section 6.4.2.3 for details. 0 Marking of cells that are received in the ingress is disabled. 1 Marking of cells that are received in the ingress is enabled.

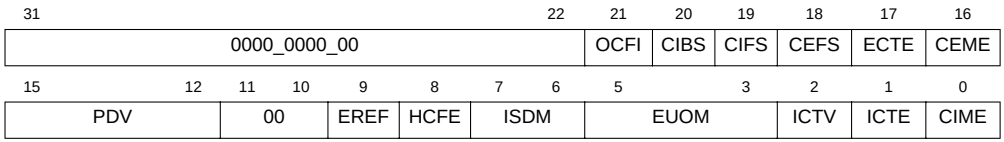


Figure 7-118. Parameter Extension Word Fields (PMAC = 0 and UPCF = 0)

Table 7-76. PMAC = 0 and UPCF = 0 Field Descriptions

Bits	Name	Description
31–22	—	Reserved
21	OCFI	Overhead CLP from input. If ingress CLP transparency support is enabled via the ICTE bit (described below), the OCFI bit determines whether the CLP bit in the cell overhead is set based on the CLP bit input to or based on the CLP bit output of the UPC enforcer stages. Note, storing a CLP bit in the overhead only works on the ingress flow, when ACR[UPCF] is clear. 0 Set overhead CLP bit value based on UPC enforcer output. 1 Set overhead CLP bit value based on UPC enforcer input.
20	CIBS	CLP inconsistency behavior selection. If frame-based UPC (GFR) is enabled (EUOM = b001), the CIBS bit determines whether CLP consistency within a frame is enforced or ignored. If enabled, CLP consistency is enforced via partial packet discard (PPD). See Section 6.3.1.2, “Cell Loss Priority (CLP) Consistency” for details.
19	CIFS	Connection ingress flow status. The MC92520 copies the overhead ingress flow status (IFS) bit to this bit. This bit is used by the ingress processing block for ABR cell marking and is, therefore, intended for MC92520 internal use. See Section 6.4.2.1 for details.
18	CEFS	Connection egress flow status. The MC92520 copies the overhead egress flow status (EFS) bit to this bit. This bit is used by the ingress processing block for ABR cell marking and is, therefore, intended for MC92520 internal use. See Section 6.4.2.2 for details.
17	ECTE	Egress CLP transparency enable. This bit determines whether CLP should be copied from the egress overhead CLP (EOCLP) bit to the cell header. See Section 6.2 for details. 0 CLP should not be copied from the switch overhead to the cell header. 1 CLP should be copied from the switch overhead to the cell header.
16	CEME	Connection egress marking enable. This bit enables marking of cells that are received in the egress. See Section 6.4.2.4 for details. 0 Marking of cells that are received in the egress is disabled 1 Marking of cells that are received in the egress is enabled.
15–12	PDV	Packet discard variables. This field is accessed only by the MC92520.
11–10		Reserved
9	EREF	Egress reset EFCI. This bit determines if PTI[1] of an egress cell is reset. 0 PTI[1] of an egress cell is not reset. 1 PTI[1] of an egress cell is to be reset.
8	HCFE	Header CLP from enforcer. If ingress CLP transparency support is enabled via the ICTE bit (described below), the HCFE bit determines whether the CLP bit in the cell header is set based on the result of the UPC enforcer stages or based on the value defined by the ICTV bit (also described below). 0 Set header CLP bit value from ICTV bit configuration. 1 Set header CLP bit value from UPC enforcer output.
7–6	ISDM	Ingress selective discard operation mode. This field determines the selective discard operation mode. See Section 6.1.3 for details. 00 No selective discard. 01 Reserved. 10 Selective discard on CLP1 flow. 11 Selective discard on CLP0 +1 flow.
5–3	EUOM	This field determines the UPC operation mode. 000 Cell-based UPC. 001 Frame-based UPC (GFR). See Section 6.3 for details. 010 Partial packet discard (PPD). See Section 6.1.2.1 for details. 100 Early packet discard (EPD). See Section 6.1.2.2 for details. 110 Limited EPD. See Section 6.1.2.3 for details.

Table 7-76. PMAC = 0 and UPCF = 0 Field Descriptions (Continued)

Bits	Name	Description
2	ICTV	Ingress CLP transparency value. This bit determines the value that should be written to a cell's header if the ingress CLP Transparency enable (ICTE) bit is set. See Section 6.2.1 for details.
1	ICTE	Ingress CLP transparency enable. This bit determines whether CLP should be copied to the ingress overhead CLP (IOCLP) bit and whether the ingress CLP transparency value (ICTV) bit should be written to the cell header CLP. See Section 6.2.1 for details. 0 The ingress header CLP bit is not touched. 1 CLP should be copied from the cell header to the ingress switch parameters. The ICTV bit should be written to the cell header CLP.
0	CIME	Connection ingress marking enable. This bit enables marking of cells that are received in the ingress. See Section 6.4.2.3 for details. 0 Marking of cells that are received in the ingress is disabled. 1 Marking of cells that are received in the ingress is enabled.

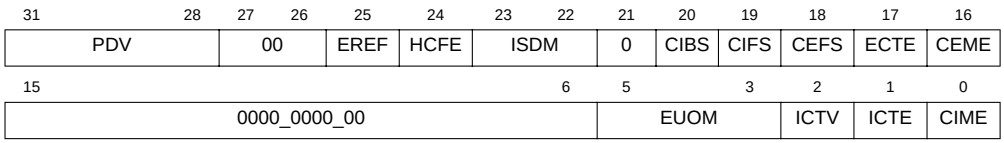


Figure 7-119. Parameter Extension Word Fields (PMAC = 0 and UPCF = 1)

Table 7-77. PMAC = 0 and UPCF = 1 Field Descriptions

Bits	Name	Description
31–28	PDV	Packet discard variables. This field is accessed only by the MC92520.
27–26	—	Reserved
25	EREF	Egress reset EFCI. This bit determines if PTI[1] of an egress cell is reset. 0 PTI[1] of an egress cell is not reset. 1 PTI[1] of an egress cell is to be reset.
24	HCFE	Header CLP from enforcer. If ingress CLP transparency support is enabled via the ICTE bit (described below), the HCFE bit determines whether the CLP bit in the cell header is set based on the result of the UPC enforcer stages or based on the value defined by the ICTV bit (also described below). 0 Set header CLP bit value from ICTV bit configuration. 1 Set header CLP bit value from UPC enforcer output.
23–22	ISDM	Ingress selective discard operation mode. This field determines the selective discard operation mode. See Section 6.1.3 for details. 00 No selective discard. 01 Reserved. 10 Selective discard on CLP1 flow. 11 Selective discard on CLP0 +1 flow.
21	—	Reserved
20	CIBS	CLP inconsistency behavior selection. If frame-based UPC (GFR) is enabled (EUOM = b001), the CIBS bit determines whether CLP consistency within a frame is enforced or ignored. If enabled, CLP consistency is enforced via partial packet discard (PPD). See Section 6.3.1.2, "Cell Loss Priority (CLP) Consistency" for details.

Table 7-77. PMAC = 0 and UPCF = 1 Field Descriptions (Continued)

Bits	Name	Description
19	CIFS	Connection ingress flow status. The MC92520 copies the overhead ingress flow status (IFS) bit to this bit. This bit is used by the ingress processing block for ABR cell marking and is, therefore, intended for MC92520 internal use. See Section 6.4.2.1 for details.
18	CEFS	Connection egress flow status. The MC92520 copies the overhead egress flow status (EFS) bit to this bit. This bit is used by the ingress processing block for ABR cell marking and is, therefore, intended for MC92520 internal use. See Section 6.4.2.2 for details.
17	ECTE	Egress CLP transparency enable. This bit determines whether CLP should be copied from the egress overhead CLP (EOCLP) bit to the cell header. See Section 6.2 for details. 0 CLP should not be copied from the switch overhead to the cell header. 1 CLP should be copied from the switch overhead to the cell header.
16	CEME	Connection egress marking enable. This bit enables marking of cells that are received in the egress. See Section 6.4.2.4 for details. 0 Marking of cells that are received in the egress is disabled 1 Marking of cells that are received in the egress is enabled.
15–6	—	Reserved
5–3	EUOM	This field determines the UPC operation mode. 000 Cell-based UPC. 001 Frame-based UPC (GFR). See Section 6.3 for details. 010 Partial packet discard (PPD). See Section 6.1.2.1 for details. 100 Early packet discard (EPD). See Section 6.1.2.2 for details. 110 Limited EPD. See Section 6.1.2.3 for details.
2	ICTV	Ingress CLP transparency value. This bit determines the value that should be written to a cell's header if the ingress CLP Transparency enable (ICTE) bit is set. See Section 6.2.1 for details.
1	ICTE	Ingress CLP transparency enable. This bit determines whether CLP should be copied to the ingress overhead CLP (IOCLP) bit and whether the ingress CLP transparency value (ICTV) bit should be written to the cell header CLP. See Section 6.2.1 for details. 0 The ingress header CLP bit is not touched. 1 CLP should be copied from the cell header to the ingress switch parameters. The ICTV bit should be written to the cell header CLP.
0	CIME	Connection ingress marking enable. This bit enables marking of cells that are received in the ingress. See Section 6.4.2.3 for details. 0 Marking of cells that are received in the ingress is disabled. 1 Marking of cells that are received in the ingress is enabled.

7.3 Data Structures

All fields marked “0” or “Reserved” in the descriptions in this section must be written with zeros. The values read from these fields should be considered undefined and should be ignored.

7.3.1 Inserted Cell Structure

The inserted cell structure contains 16 long words as shown in Figure 7-120. The contents of the first and second long words are described below. The third long word is unused. The fourth long word contains the ATM cell header (not including the HEC octet), and the last 12 long words contain the cell payload.

CIR0	Cell Descriptor	ACIR0
CIR1	Connection Descriptor	ACIR1
CIR2	Unused	
CIR3	ATM cell header (VPI,VCI,PTI,CLP)	
CIR4	Payload 1–4	
CIR5	Payload 5–8	
CIR6	Payload 9–12	
CIR7	Payload 13–16	
CIR8	Payload 17–20	
CIR9	Payload 21–24	
CIR10	Payload 25–28	
CIR11	Payload 29–32	
CIR12	Payload 33–36	
CIR13	Payload 37–40	
CIR14	Payload 41–44	
CIR15	Payload 45–48	

Figure 7-120. Inserted Cell Structure

7.3.1.1 Cell Descriptor

The cell descriptor informs the MC92520 what should be done with the inserted cell. Five formats are used for the cell descriptor, depending on how the cell header is provided. If the user provides the ATM header and payload, Format I is used, shown in Figure 7-121.

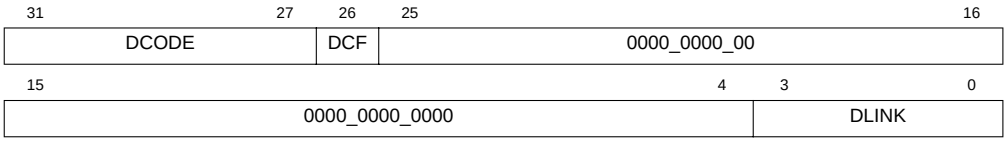


Figure 7-121. Cell Descriptor Format I

Following are the fields used in the cell descriptor formats, collectively and individually:

Table 7-78. Cell Descriptor Field Descriptions

Name	Description
DCODE	Descriptor code. The DCODE field indicates the type of processing the MC92520 should perform on the inserted cell. It also determines the format of the remainder of the cell descriptor word (listed in parentheses). 00000 The user provides the header and payload. The MC92520 generates the CRC-10 field. (Format I) 00001 The user provides the header and payload. The MC92520 does not modify the cell. (Format I) 00010 The user provides the non-address part of the header and payload. The ATMC CFB performs address translation and generates the CRC-10 field. (Format II) 00011 The user provides the non-address part of the header and payload. The MC92520 performs address translation. (Format II) 00100 The MC92520 generates the header and the payload. This value is only used for OAM cells. (Format V) 00110 The user provides the non-address part of the header and an OAM fields template (see Section 7.3.1.4). The MC92520 performs address translation and generates the payload from the OAM fields template. (Format IV) 00111 The user provides the header and an OAM fields template as defined in Section 7.3.1.4. The MC92520 generates the payload from the OAM fields template. (Format III) 01000 The user provides the header and the payload, but does not provide a connection identifier (see Section 7.3.1.2). The MC92520 generates the CRC-10 field and inserts the cell, but does not perform any connection-specific processing. (Format I) 01001 The user provides the header and the payload, but does not provide a connection identifier (see Section 7.3.1.2). The MC92520 inserts the cell, but does not perform any connection-specific processing. (Format I) 01100 The MC92520 inserts a hole (that is, a cell processing slot is used, but no cell is inserted). (Format II) 01101 The user provides the header, the payload, and a connection identifier (see Section 7.3.1.2). The MC92520 generates the CRC-10 field and inserts the cell, but does not perform any connection-specific processing except for appending the switch parameters. (Format I) 01110 The user provides the header, the payload, and a connection identifier (see Section 7.3.1.2). The MC92520 inserts the cell, but does not perform any connection-specific processing except for appending the switch parameters. (Format I) If ingress address translation is enabled, it is performed for all cells that undergo connection-specific processing, even if the user provides the header (DCODE = 00000 or 00001). When a cell is inserted into the ingress cell flow without a connection identifier (DCODE = 01000 or 01001), switch parameters from the context parameters table are not provided and address translation is not performed. To insert a cell that is not processed, but does have switch parameters appended, use DCODE = 01101 or 01110. The non-address part of the ATM header consists of the fields that are not modified during address translation. Generally, this includes the PTI and CLP fields. If VP switching is performed, it generally includes the VCI field.
DCF	Descriptor cell flow. This bit determines into which cell flow the cell is to be inserted (ingress or egress). (All Formats) 0 The cell is to be inserted into the egress cell flow. 1 The cell is to be inserted into the ingress cell flow.
DSEG	Descriptor segment. The DSEG bit determines whether the inserted OAM cell is end-to-end or segment, and the header is generated accordingly. 0 The inserted cell is an end-to-end OAM cell. 1 The inserted cell is a segment OAM cell. (Format II)
DCLP	Descriptor CLP. Used as the CLP bit in the ATM header of the generated cell. (Format V)

Table 7-78. Cell Descriptor Field Descriptions (Continued)

Name	Description
DCN	Descriptor cell name. The DCN field identifies the type of OAM cell to be generated. The coding of this field can be found in Table 7-87. Currently, the only values valid in the cell descriptor are BRC for Formats III and IV and AIS, RDI, CC, and FMC for Format V.
DLINK	Descriptor link. For a cell inserted into the ingress cell flow, the DLINK field indicates to which physical link the cell belongs. The MC92520 ignores this field, but the customer-specific logic or switch interface blocks may use this information. For a cell inserted into the egress cell flow, the DLINK field indicates to which physical link the cell should be transferred. (Formats I & III)

When the user provides the non-address part of the ATM header and payload, and the MC92520 performs address translation, Format II is used, as shown in Figure 7-122. This format is also used for inserting holes in the cell flow.

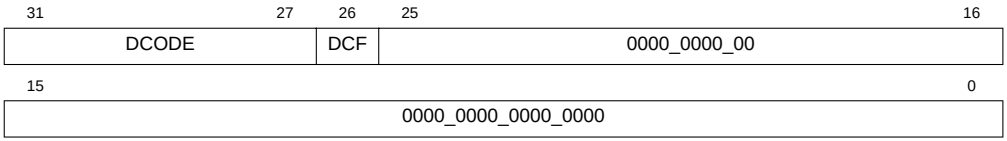


Figure 7-122. Cell Descriptor Format II

When the user provides the ATM header and an OAM fields template, Format III is used (shown in Figure 7-123).

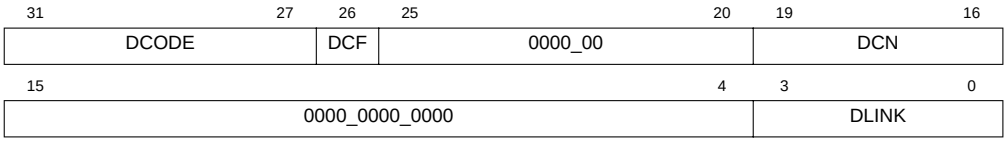


Figure 7-123. Cell Descriptor Format III

When the user provides the non-address part of the ATM header and an OAM fields template, and the MC92520 performs address translation, Format IV is used (see Figure 7-124).

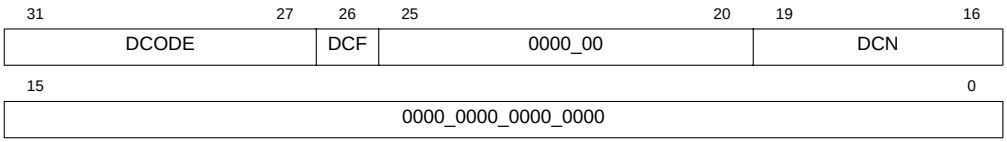


Figure 7-124. Cell Descriptor Format IV

When the MC92520 generates the OAM cell header and payload, Format V is used (see Figure 7-125).

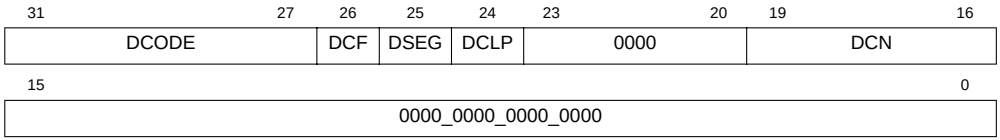


Figure 7-125. Cell Descriptor Format V

7.3.1.2 Connection Descriptor

The second long word of the inserted cell contains the connection descriptor. The fields contained in this word determine the connection into which the cell is inserted.

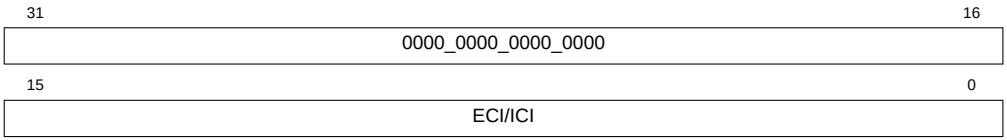


Figure 7-126. Connection Descriptor Structure

Table 7-79. Connection Indication Field Description

Bits	Name	Description
31-16	—	Reserved
15-0	ECI/ ICI	This field identifies the connection to which the inserted cell belongs. Depending on the cell flow bit described in Section 7.3.1.1, this field is interpreted as either the egress connection identifier or the ingress connection identifier. In either case, it is used as the index to the context parameters table during the cell processing.

7.3.1.3 ATM Cell Header

The fourth long word of the inserted cell contains the ATM cell header. The fields are defined by ATM standards.

NOTE:

Depending on the DCODE of the inserted cell, portions of the header may be altered during the cell processing.

At a UNI, the header fields are structured as shown in Figure 7-127.

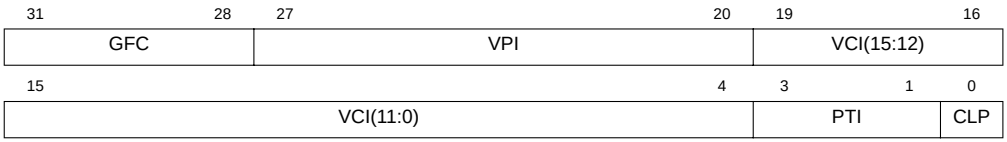


Figure 7-127. Inserted Cell ATM Cell Header Fields (UNI)

At an NNI, the header fields are structured as shown in Figure 7-128.

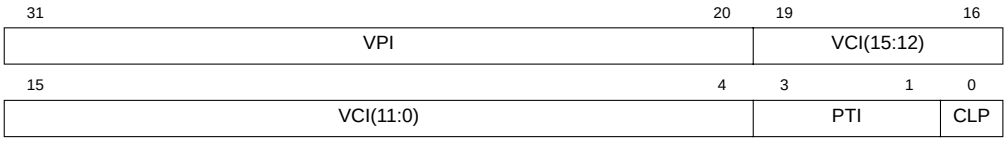


Figure 7-128. Inserted Cell ATM Cell Header Fields (NNI)

The UNI and NNI fields are the following:

- Generic flow control (GFC)
- Virtual path identifier (VPI)
- Virtual channel identifier (VCI)
- Payload type identifier (PTI)
- Cell loss priority (CLP)

7.3.1.4 Inserted OAM Fields Template

When the MC92520 is requested to generate the payload of an OAM cell from a template, several of the long words of the payload portion of the inserted cell structure contain the OAM fields template. The remaining payload long words are undefined. The only type of OAM fields template currently defined is a BRC fields template. The BRC fields template is normally copied directly from the template provided by the MC92520 (see Section 7.3.2.5, “Extracted OAM Fields Template”).

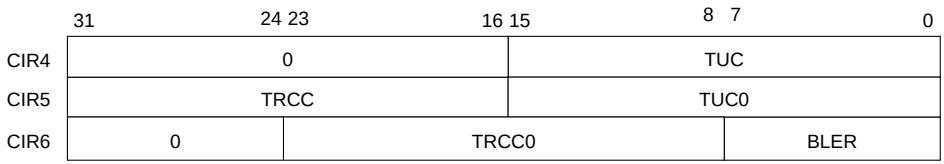


Figure 7-129. Inserted BRC Fields Template

- TUC—This field is used for the third and fourth octets of the generated cell payload. See Section 6.5.5, “Performance Management.”
- TRCC—This field is used for the forty-fifth and forty-sixth octets of the generated cell payload. See Section 6.5.5, “Performance Management.”
- TUC0—This field is used for the seventh and eighth octets of the generated cell payload. See Section 6.5.5, “Performance Management.”
- TRCC0—This field is used for the forty-second and forty-third octets of the generated cell payload. See Section 6.5.5, “Performance Management.”
- Block error result (BLER)—This field is used for the forty-fourth octet of the generated cell payload. See Section 6.5.5, “Performance Management.”

7.3.2 Extracted Cell Structure

The extracted cell structure contains 16 long words as shown in Figure 7-130. The contents of the first through third long words are described below. The fourth long word contains the ATM cell header (not including the HEC octet), and the last 12 long words contain the cell payload.

CER0	Cell Indication
CER1	Connection Indication
CER2	Time Stamp
CER3	ATM cell header (VPI,VCI,PTI,CLP)
CER4	Payload 1–4
CER5	Payload 5–8
CER6	Payload 9–12
CER7	Payload 13–16
CER8	Payload 17–20
CER9	Payload 21–24
CER10	Payload 25–28
CER11	Payload 29–32
CER12	Payload 33–36
CER13	Payload 37–40
CER14	Payload 41–44
CER15	Payload 45–48

Figure 7-130. Extracted Cell Structure

7.3.2.1 Cell Indication

The cell indication informs the processor where the cell came from and why it was extracted. There are five formats for the cell indication. The formats can be distinguished by the values of the most-significant bits.

7.3.2.1.1 User/OAM Cell Indication

This format is used for a user or OAM cell. When this format is used, the entire cell structure contains valid data.

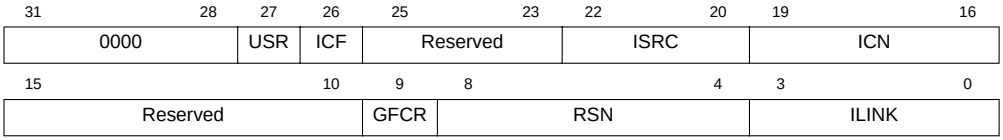


Figure 7-131. User/OAM Cell Indication Fields

Table 7-80. User/OAM Cell Indication Field Descriptions

Bits	Name	Description
31–28	—	Reserved, should be cleared.
27	USR	User cell. This bit distinguishes between OAM cells and user cells. 0 OAM cell 1 User cell
26	ICF	Indication cell flow. This bit indicates from which cell flow the cell has been copied (ingress or egress). 0 The cell was in the egress cell flow. 1 The cell was in the ingress cell flow.
25–23	—	Reserved, should be cleared.
22–20	ISRC	Indication source. The ISRC field indicates the source of the cell. 001 =The cell arrived from the PHY/switch. 010 The cell was inserted through the processor interface. 011 The FMC was internally generated. 100 The cell was generated by the internal scan process. 101 The cell was inserted by the customer-specific logic. All unused values are reserved.
19–16	ICN	Indication cell name. This field identifies the type of cell being extracted. The coding of this field can be found in Table 7-87.
15–10	—	Reserved
9	GFCR	GFC reason. The GFCR bit, when set, indicates that the GFC field in the cell's header is non-zero.
8–4	RSN	Extraction reason. This field contains the reason that the MC92520 copied this cell to the cell extraction queue. The coding of this field can be found in Table 7-88. Note that there may be instances in which more than one reason applies, but only one reason is provided.
3–0	ILINK	Indication link. This field identifies the physical link on which the cell arrived (ingress) or would be transmitted (egress).

7.3.2.1.2 Egress Multicast Translation Failure Cell Indication

The format shown in Figure 7-132 is used for a cell from the egress flow if the multicast translation failed. When this format is used, the entire cell structure contains valid data.

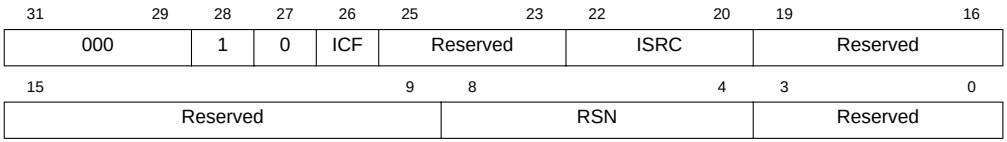


Figure 7-132. Egress Multicast Translation Failure Cell Indication Fields

Table 7-81. Egress Multicast Translation Failure Cell Indication Field Descriptions

Bits	Name	Description
31–29	—	Reserved
28	1	Set
27	—	Reserved
26	ICF	Indication cell flow. This bit is reset to indicate that the cell was copied from the egress cell flow.

Table 7-81. Egress Multicast Translation Failure Cell Indication Field Descriptions

Bits	Name	Description
25–23	—	Reserved
22–20	ISRC	Indication source. The ISRC field is 001, which indicates that the cell arrived from the switch.
19–9	—	Reserved
8–4	RSN	Extraction reason. This field contains the reason that the MC92520 copied this cell to the cell extraction queue. The coding of this field can be found in Table 7-88. The only defined value is address compression failure.When this cell indication is used, the connection indication word appears as described in Section 7.3.2.2.
3–0	—	Reserved

7.3.2.1.3 Error Cell Indication

This format is used if an error was found in the first stages of processing. When this format is used, the entire cell structure contains valid data.

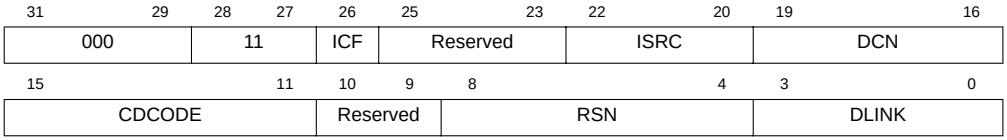


Figure 7-133. Error Cell Indication Fields

Table 7-82. Error Cell Indication Field Descriptions

Bits	Name	Description
31–29	—	Reserved
28–27	11	Set
26	ICF	Indication cell flow. This bit indicates from which cell flow the cell has been copied (ingress or egress). 0 The cell was in the egress cell flow. 1 The cell was in the ingress cell flow.
25–23	—	Reserved
22–20	ISRC	Indication source. The ISRC field indicates the source of the cell. 001 The cell arrived from the PHY/Switch. 010 The cell was inserted through the processor interface. 011 The FMC was internally generated. 100 The cell was generated by the internal scan process. 101 The cell was inserted by the customer-specific logic. All unused values are reserved.
19–16	DCN	Descriptor cell name. This field contains the cell name field from the cell descriptor. Depending on the source of the cell, this field might not be meaningful. The coding of the cell name can be found in Table 7-87.
15–11	CDCODE	Cell descriptor code. This field contains the DCODE field from the cell descriptor. Depending on the source of the cell, the value of this field might not be identifiable.
10–9	—	Reserved

Table 7-82. Error Cell Indication Field Descriptions (Continued)

Bits	Name	Description
8–4	RSN	Extraction reason. This field contains the reason that the MC92520 copied this cell to the cell extraction queue. The coding of this field can be found in Table 7-88. Note that there may be instances in which more than one reason applies, but only one reason is provided.
3–0	DLINK	Descriptor link. This field contains the DLINK field from the cell descriptor. Depending on the source of the cell, this field might not be meaningful.

7.3.2.1.4 Short Report Indication

The format shown in Figure 7-134 is used to report various events that occurred in the course of the cell processing. When this format is used only CER0–CER2 contain valid data.

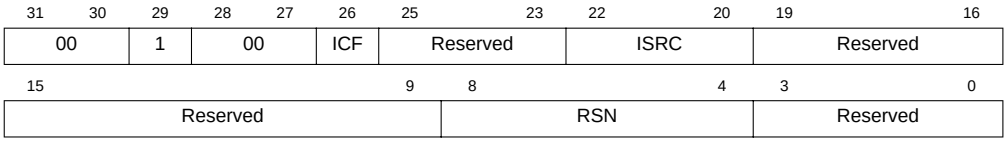


Figure 7-134. Short Report Indication Fields

Table 7-83. Short Report Indication Field Descriptions

Bits	Name	Description
31–30	—	Reserved
29	1	Set
28–27	—	Reserved
26	ICF	Indication cell flow. This bit indicates from which cell flow the report originated (ingress or egress). 0 The report is from the egress cell flow. 1 The report is from the ingress cell flow.
25–23	—	Reserved
22–20	ISRC	Indication source. The ISRC field indicates the source of the cell that triggered the report generation. 001 The cell arrived from the PHY/Switch. 010 The cell was inserted through the processor interface. 101 The cell was inserted by the customer-specific logic. All unused values are reserved.
19–9	—	Reserved
8–4	RSN	Extraction reason. This field contains the reason that the MC92520 generated this report. The coding of this field can be found in Table 7-88. Currently, the only defined reasons for the short report indication are the two values regarding FMCs not being inserted. When using the short report indication, the connection identifier is found in the connection indication word described in Section 7.3.2.2, “Connection Indication.”
3–0	—	Reserved

7.3.2.1.5 OAM Fields Template Indication

This format is used for an OAM fields template.

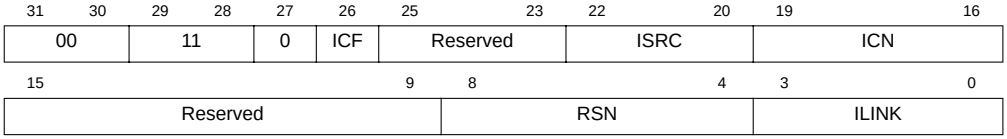


Figure 7-135. OAM Fields Template Fields

Table 7-84. OAM Fields Template Field Descriptions

Bits	Name	Description
31–30	—	Reserved
29–28	1	Set
27	—	Reserved
26	ICF	Indication cell flow. This bit indicates from cell flow in which the template was generated (ingress or egress). 0 The template is from the egress cell flow. 1 The template is from the ingress cell flow.
25–23	—	Reserved
22–20	ISRC	Indication source. The ISRC field indicates the source of the cell that triggered the generation of the template. 001The cell arrived from the PHY/Switch.
19–16	ICN	Indication cell name. This field identifies the cell type to generate from the template. Field coding is in Table 7-87. Currently, only BRC is defined.
15–9	—	Reserved
8–4	RSN	Extraction reason. This field defines the template type. The field coding is in Table 7-88. Currently, only BRC is defined.
3–0	ILINK	Indication link. This field identifies the physical link on which the cell that triggered the generation of the template arrived (ingress) or would be transmitted (egress).

When the OAM fields template indication is used, the payload portion of the extracted cell structure contains the OAM fields template as described in Section 7.3.2.5, “Extracted OAM Fields Template.”

7.3.2.2 Connection Indication

The second long word of the extracted cell contains the connection indication. The fields contained in this word indicate the connection to which the cell or report or template belongs.

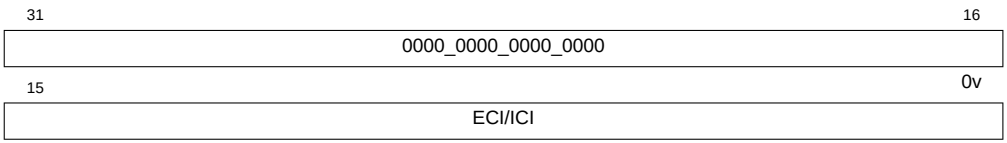


Figure 7-136. Connection Indication Fields

Table 7-85. Connection Indication Field Description

Bits	Name	Description
31-16	—	Reserved
15-0	ECI/ ICI	Egress connection identifier / ingress connection identifier (ECI/ICI)—This field identifies the connection to which the extracted cell or report or template belongs. Depending on the cell flow bit of the cell indication, this field is interpreted as either the egress connection identifier or the ingress connection identifier.

When the egress multicast translation failure indication described in Section 7.3.2.1.2, “Egress Multicast Translation Failure Cell Indication” is used, the connection indication word contains overhead information obtained from the switch that may be helpful in determining the cause of the failure.

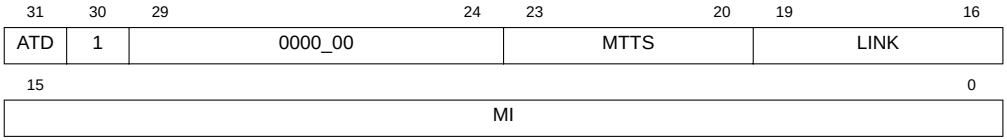


Figure 7-137. Egress Multicast Translation Failure Connection Overhead Information

Table 7-86. Egress Multicast Translation Failure Connection Field Descriptions

Bits	Name	Description
31	ATD	Address translation disable. The ATD bit, when set, indicates that address translation is disabled. This bit is taken from the egress address translation disable (EATD) bit of the egress switch interface configuration register (ESWCR).
30	1	Set
29–24	—	Reserved
23–20	MTTS	Multicast translation table section. The MTTS field is used in multicast identifier translation. See Section 5.2.2, “Multicast Identifier Translation.”
19–16	LINK	The LINK field contains the link number if ATD is set. Otherwise, this field contains 0000.
15–0	MI	Multicast identifier. This field contains the multicast identifier used in multicast identifier translation. See Section 5.2.2, “Multicast Identifier Translation.”

7.3.2.3 Time-Stamp

The third long word of the extracted cell contains a time stamp indicating when the cell was processed. The time stamp field is in units of MC92520 cell processing times and represents the number of cell processing times because the MC92520 was reset (modulo 2^{32}).

7.3.2.4 ATM Cell Header

The fourth long word of the extracted cell contains the incoming ATM cell header. The fields are defined by ATM standards. At a UNI, the header fields are as follows:

- Generic flow control (GFC)
- Virtual path identifier (VPI)
- Virtual channel identifier (VCI)
- Payload type identifier (PTI)
- Cell loss priority (CLP)

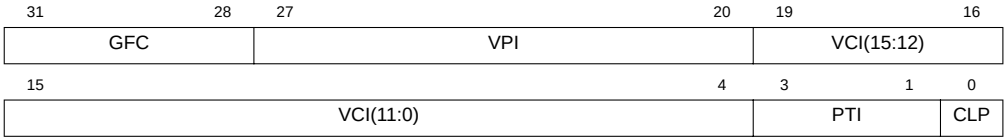


Figure 7-138. Extracted Cell ATM Cell Header (UNI)

Figure 7-139 shows NNI header fields.

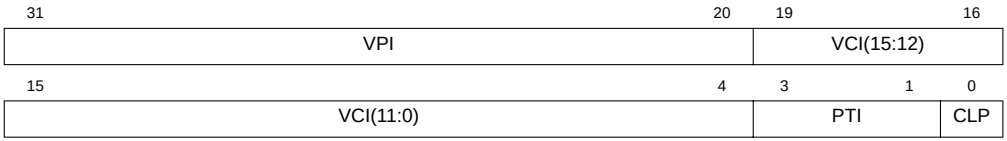


Figure 7-139. Extracted Cell ATM Cell Header (NNI)

7.3.2.5 Extracted OAM Fields Template

When the OAM fields template indication is used, several of the long words of the payload portion of the extracted cell structure contain the OAM fields template. The remaining payload long words are undefined. The only type of OAM fields template currently defined is a BRC fields template. The BRC fields template is normally copied directly to the template inserted to the MC92520 (see Section 7.3.1.4, “Inserted OAM Fields Template”).

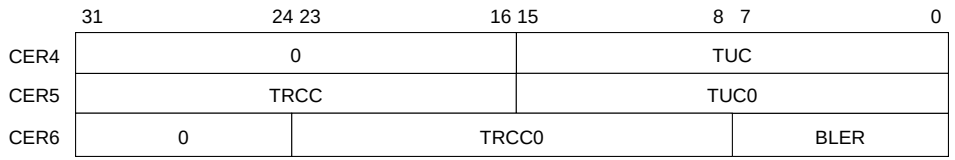


Figure 7-140. Extracted BRC Fields Template

- TUC—This field is copied from the TUC of the received FMC. See Section 6.5.5, “Performance Management.”
- TRCC—This field is copied from the TUC of the OAM table entry. See Section 6.5.5, “Performance Management.”
- TUC0—This field is copied from the TUC0 of the received FMC. See Section 6.5.5, “Performance Management.”

- TRCC0—This field is copied from the TUC0 of the OAM table entry. See Section 6.5.5, “Performance Management.”
- Block error result (BLER)—This field is computed from the BEDC of the received FMC and the OAM table entry. See Section 6.5.5, “Performance Management.”

7.3.3 General Fields

This section contains the coding for fields that are shared by multiple data structures. Not all of the values are valid for all of the structures.

7.3.3.1 Cell Name Field

Table 7-87 describes the coding of the cell name field (CN).

Table 7-87. Cell Name Field Coding

CN	Description
0000	Undefined
0001	OAM AIS cell
0010	OAM RDI cell
0011	OAM loopback cell
0100	OAM continuity check cell
0101	OAM forward monitoring cell
0110	OAM backward reporting cell
1000	Other OAM cell
1001	User cell

7.3.3.2 Reason Field

Table 7-88 describes the coding of the reason field (RSN). The reasons are listed in order of increasing priority, rather than in strict numerical order. If more than one reason applies, the last reason listed is the one that is provided with the cell.

Table 7-88. Reason Field Coding

RSN	Description
00000	Undefined
00001	The cell was copied because the egress copy all (ECA) cells or ingress copy all (ICA) cells bit was set. See Section 7.2.3, “Context Parameters Table.”
00010	The cell is an OAM cell that was copied because one of the OAM copy bits in the context parameters table was set. See Section 7.2.3, “Context Parameters Table.”
00100	The cell is a loopback cell that returned to its source.
00101	The cell is a loopback cell to be looped back at this point.
00110	The cell was copied due to a reserved VCI value as indicated by the VCR register. See Section 7.1.6.29, “Ingress VCI Copy Register (IVCR)” and Section 7.1.6.30, “Egress VCI Copy Register (EVCR).”

Table 7-88. Reason Field Coding (Continued)

RSN	Description
00111	The cell was copied due to a reserved value of PTI. See Section 7.2.3.5, “Egress Parameters” and Section 7.2.3.6, “Ingress Parameters.”
01001	A CRC error was detected.
01010	Illegal OAM cell. See Section 6.5.3.1, “Illegal OAM Cells.”
01100	The PHY/Switch interface detected a parity error.
01101	The cell has an invalid link number. (egress only)
01110	The connection ECIV/ICIV bit is reset.
01111	The address compression failed.
10000	One or more of the unmasked header bits of the cell is set.
01011	Illegal cell
10001	The cell is an invalid (PHY-layer) cell.
10010	The PHY/Switch interface detected a protocol error.
10011	No cell was processed because the output queue is full.
10100	No cell was processed because there was no cell to process.
10101	No cell was processed because it was a maintenance slot.
11101	This is a BRC fields template.
11110	This connection is running a PM block test, and an end-to-end FMC was not inserted on time.
11111	This connection is running a PM block test, and a potential FMC was discarded because it could not be inserted.

Some of the reasons listed here never appear in the cell extraction queue.

Chapter 8
Test Operation

The IEEE JTAG boundary scan architecture and test access port (TAP) controller are discussed in the following subsections.

8.1 JTAG Overview

The MC92520 provides a test access port (TAP) that is compatible with the IEEE 1149.1 standard test access port and boundary scan architecture [6]. This implementation of IEEE 1149.1 allows boundary scan compatibility with other JTAG components in a circuit-board. A boundary scan description language (BSDL) file for this device is provided in Appendix E.

The TAP includes five dedicated signal pins, a 16-state TAP controller, a bypass register, an instruction register, and a device identification (ID) register. The ID register contains the specific JTAG ID code. A boundary scan register links all device I/O signal pins into a shift-register chain around the periphery of the device. This path is provided with serial input and output signal pins and is controlled by appropriate clock and control signals.

The JTAG test logic is independent of the device system logic. The JTAG implementation on the MC92520 provides for the following capabilities:

- Boundary scan operations
- BYPASS mode, which bypasses the MC92520 boundary scan chain
- Sampling system data from I/O signals (for inputs) and core system data (for outputs) and shifting out the result through the boundary scan register
- Test of component interconnect with EXTEST mode. This instruction updates I/O and system logic with data that is shifted in serially. Output pins are driven with the updated values, and input pins have the updated values driven into the core logic.
- CLAMP mode, which drives as in EXTEST mode, but puts boundary scan chain in BYPASS mode
- IDCODE mode allows the ID code to be shifted from the ID register.
- HIGHZ mode tristates all system outputs and bi-directionals for component isolation.

Other JTAG instruction codes are reserved. They are listed as “PRIVATE” in the BSDL code in Appendix E.

8.1.1 Functional Blocks

The MC92520 implementation of the IEEE 1149.1 standard includes a TAP controller, an instruction register, a bypass register, an ID code register, and the I/O boundary scan register. The associated signal pins for the JTAG logic are:

- TCK—Test clock input for the JTAG test logic
- TMS—Test mode select input used to sequence the TAP controller’s state machine, sampled on the rising edge of TCK
- TDI—Test data input sampled on the rising edge of TCK
- TDO—Test data output active during the Shift-DR and Shift-IR controller states
- $\overline{\text{TRST}}$—Asynchronous test reset signal that initializes the TAP controller and other JTAG logic, active low

A block diagram overview is shown in Figure 8-1.

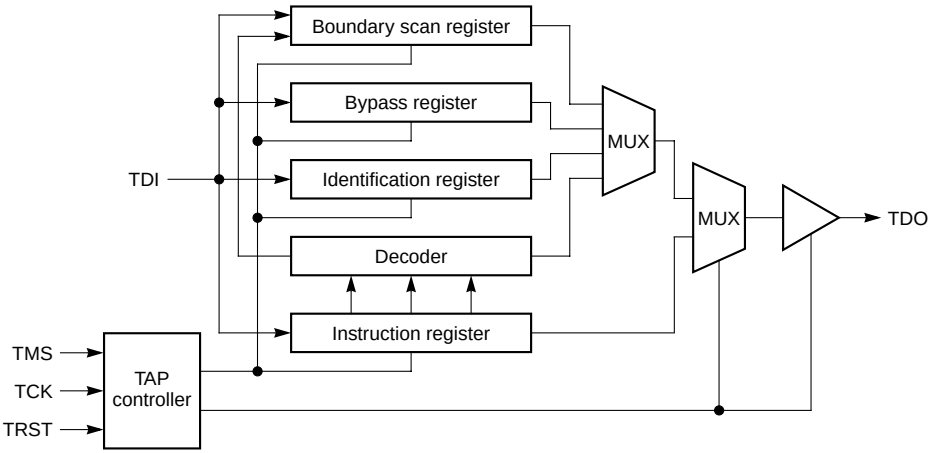


Figure 8-1. JTAG Logic Block Diagram

8.1.1.1 TAP Controller

The TAP controller is a synchronous, 16-state machine that controls and manages the mode of operation for the test circuitry. The controller interprets the sequence of logical values on the TMS signal. The TAP controller states are explained in the IEEE 1149.1 document. The state diagram for the controller is shown in Figure 8-2.

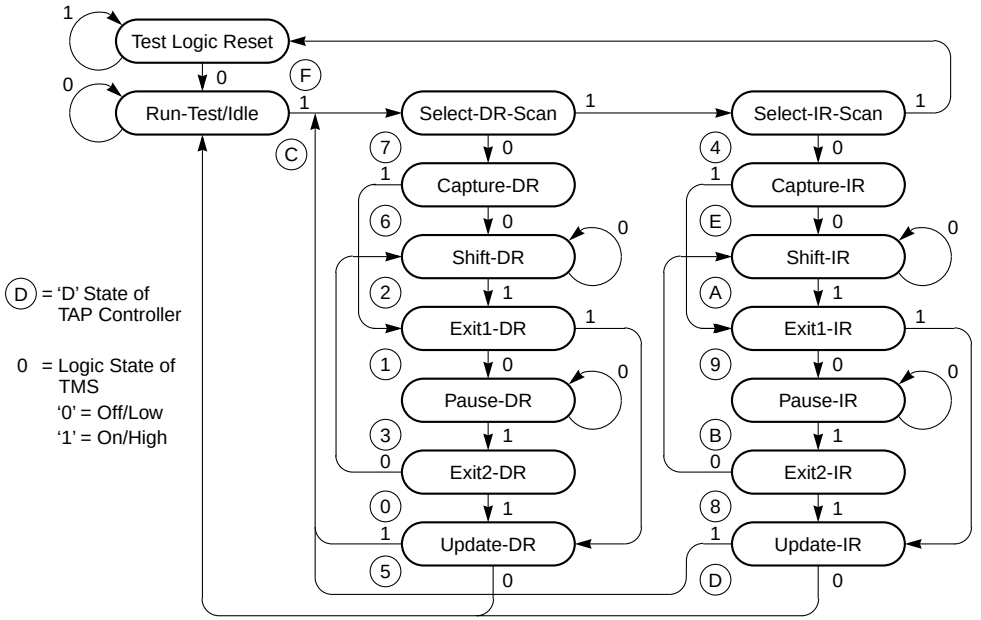


Figure 8-2. TAP Controller State Diagram

8.1.1.2 Instruction Register

The 4-bit instruction register holds various instructions that define which test registers are to be used and the serial test data register path between TDI and TDO signals. The instructions are described in Section 8.1.2 JTAG Instruction Support.

8.1.1.3 Device Identification Register

The device identification register is a 32-bit register that holds the manufacturer's identity code, sequence number (equivalent to a part number), and version code. The bit assignment for the ID code is shown in Table 8-1. The ID code data is shifted out with the least significant bit (0) shifted out first.

Table 8-1. Device Identification Register ID Codes

Bit No.	Code Use	MC92520 Value	MC92510 Value
31-28	Version Number	(Revision A) 0000	(Revision A) 0000
27-22	Motorola ASIC Identification	00_1010	00_1010
21-12	Sequence Number	00_0000_1010	00_0000_1011
11-0	Motorola Identification	0000_0001_1101	0000_0001_1101

8.1.1.4 Bypass Register

The 1-bit bypass register is connected between TDI and TDO when either the BYPASS instruction or the CLAMP instruction is active. This allows for bypassing the boundary scan register while other components on a board are being tested. When the bypass register is selected by the current instruction, the data that is shifted out on the first clock edge is the previously latched data.

8.1.1.5 Boundary Scan Register

The boundary scan register allows testing of circuitry external to the MC92520. It also allows the signals flowing through the system pins to sampled and examined without interfering with the operation of the MC92520. The boundary scan register includes all the functional pins of the MC92520 and additional stages that control the direction and driving state of some of the pins.

8.1.2 JTAG Instruction Support

The following instructions are supported by the MC92520:

- SAMPLE—The SAMPLE instruction captures the input or output data from the I/O pads. The data is captured on the rising edge of TCK when the TAP controller is in the capture-DR state, and can be observed when shifted out through the boundary scan register. This instruction does not update the core logic or the I/O pads. This makes it useful for preloading the boundary scan register with known data when entering the EXTEST instruction.
- BYPASS—The BYPASS instruction selects the bypass register for serial access between TDI and TDO. This instruction does not update the core logic or the I/O pads.
- EXTEST—The EXTEST instruction selects the boundary scan register. When the EXTEST instruction is active, the output pins and the internal core input signals are updated with test data that has been previously shifted into the boundary scan register. Also, the state of all input ports of the chip core are defined by the data shifted into the boundary scan register, and are updated on the falling edge of TCK in the update-DR controller state. Refer to the IEEE 1149.1 document for more details on the function and use of EXTEST.
- IDCODE—The MC92520 includes a device identification register. The IDCODE instruction selects only the device identification register to be connected for serial access between TDI and TDO in the shift-DR controller state. When the IDCODE instruction is selected, the vendor identification code is loaded into the ID register in the capture-DR state.

- CLAMP—The CLAMP instruction selects the bypass register for serial access between TDI and TDO. The CLAMP instruction drives the core data and the chip I/O signals in the same manner as EXTEST, while the bypass register is selected as the serial TDI/TDO path.
- HIGHZ—The HIGHZ instruction selects the bypass register for serial access between TDI and TDO. This instruction also puts all system logic output pins (including bi-directional pins) in their inactive drive state. This instruction is provided to help isolate the component during circuit board testing.

Table 8-2 shows the coding of the instructions supported by the MC92520. Bit 0 is the first bit shifted in from TDI and the first bit shifted out through TDO. The shaded rows of the table indicate the code point that is defined in the BSDL file for each instruction. The other values are provided for completeness, but are not expected to be used.

Table 8-2. Instruction Decoding

Code				Instruction
Bit3	Bit2	Bit1	Bit0	
0	0	0	0	EXTEST
0	0	0	1	IDCODE
0	0	1	0	SAMPLE
0	0	1	1	RESERVED
0	1	0	0	RESERVED
0	1	0	1	RESERVED
0	1	1	0	RESERVED
0	1	1	1	PRIVATE
1	0	0	0	PRIVATE
1	0	0	1	HIGHZ
1	0	1	0	PRIVATE
1	0	1	1	PRIVATE
1	1	0	0	CLAMP
1	1	0	1	PRIVATE
1	1	1	0	PRIVATE
1	1	1	1	BYPASS

8.1.3 Boundary Scan Register Path

The MC92520 boundary scan register path is defined in Appendix E, in Boundary Scan Description Language (BSDL) format.

Chapter 9
Product Specifications

9.1 Introduction

This section includes information about the MC92520 product specifications, including:

- Signal names
- Physical and electrical characteristics
- Ordering information
- Packaging specifications

The following sections provide detailed descriptions of the product specifications.

9.2 Signal Description

This section contains brief descriptions of the input and output signals in their functional groups, as shown in Figure 9-1. Each signal is explained briefly in a paragraph with references to other sections that contain more details about the signal and the related operations.

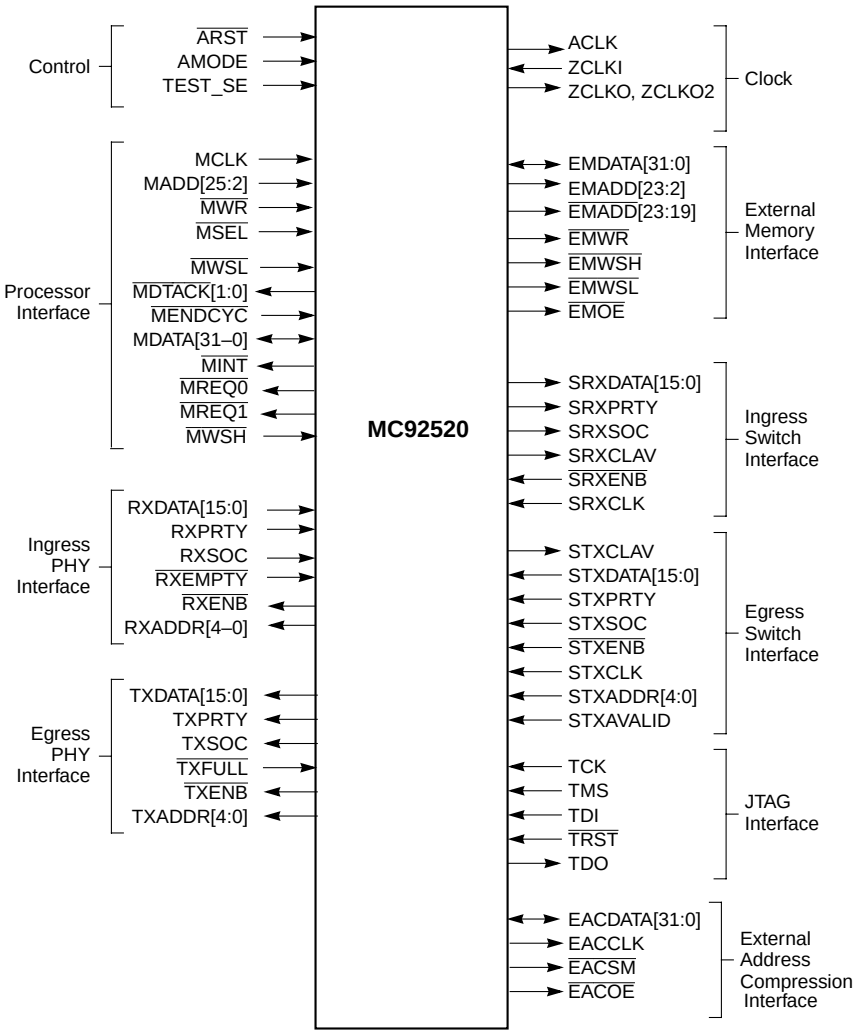


Figure 9-1. Functional Signal Groups

9.2.1 Ingress PHY Signals

The following signals relate to the ingress PHY interface that is used to connect PHY devices supporting the UTOPIA standard. Refer to Section 4.1.1, Section 4.2, and Section 7.1.6.2 for a detailed description of the interface and its configuration register.

All of the input signals are sampled at the rising edge of ACLK, and all of the output signals are updated at the rising edge of ACLK.

- Receive data bus (RXDATA0–RXDATA15)—This input data bus receives data from one or more PHY devices. When $\overline{\text{RXENB}}$ is active, RXDATA is sampled by the MC92520. The used width of the input data bus is determined by the IPWD bit in the IPHCR register. When IPWD is low, RXDATA8-RXDATA15 should be grounded.
- Receive data bus parity (RXPRTY)—This input is the odd parity over the used width of RXDATA; that is, depending on the setting of the IPWD bit, parity is checked for an octet or a 16-bit word. This input is ignored if $\overline{\text{RXENB}}$ was not active or parity check is disabled via the IPPR and IPLP bits.
- Receive start of cell (RXSOC)—This input, when high, indicates that the current RXDATA is the first byte or word of a cell. This input is sampled when RXENB is active.
- Receive PHY empty ($\overline{\text{RXEMPTY}}$)—This input, when low, indicates that currently the PHY chip has no available data. This pin is used for the “RXCLAV” function in multi-PHY mode. In this mode, a high indicates a cell is available to be received from the PHY.
- Receive enable ($\overline{\text{RXENB}}$)—This output, when low, indicates that the MC92520 is ready to receive data.
- Receive multi-PHY address (RXADDR0–RXADDR4)—This output bus is only used if multi-PHY operation is enabled via the IUM bit. The address bus indicates the PHY ID number of PHY ports being polled or selected by the MC92520.

9.2.2 Egress PHY Signals

The following signals relate to the egress PHY interface that is used to connect PHY devices supporting the UTOPIA standard. Refer to Section 4.1.1, Section 4.2, and Section 7.1.6.2 for a detailed description of the interface and its configuration register.

All of the input signals are sampled at the rising edge of ACLK, and all of the output signals are updated at the rising edge of ACLK.

- Transmit data bus (TXDATA0–TXDATA15)—This output data bus transmits data to one or more PHY devices. When $\overline{\text{TXENB}}$ is active, TXDATA contains valid data for the PHY. The used width of the data bus is determined by the EPWD bit in the EPHCR register. When the EPWD bit is clear, TXDATA8-TXDATA15 are 0 and not valid.

- Transmit data bus parity (TXPRTY)—This output signal is the odd parity over the used width of TXDATA, i.e., depending on the setting of the EPWD bit, parity is computed for an octet or a 16-bit word. When $\overline{\text{TXENB}}$ is active, TXPRTY is a valid parity bit for the PHY.
- Transmit enable ($\overline{\text{TXENB}}$)—This output signal, when low, indicates that TXDATA, TXPRTY, and TXSOC are valid data for the PHY.
- Transmit start of cell (TXSOC)—This output signal indicates, when high, that the current data on TXDATA is the first byte or word of a cell. TXSOC is valid when $\overline{\text{TXENB}}$ is asserted.
- Transmit PHY full ($\overline{\text{TXFULL}}$)—This input signal indicates, when low, that the PHY device is full. In multi-PHY mode, this pin is used for the “TXCLAV” function. In this mode, a high indicates that the PHY has space available for cell to be transmitted.
- Transmit multi-PHY address (TXADDR0–TXADDR4)—This output bus is only used if multi-PHY operation is enabled via the EUM bit. The address bus indicates the PHY ID number of PHY ports being polled or selected by the MC92520.

9.2.3 Ingress Switch Interface Signals

The following signals relate to the ingress switch interface. Refer to Section 4.2.1, “Single-PHY Receive Interface (Ingress)” and Section 7.1.6.5, “Ingress Switch Interface Configuration Register (ISWCR)” for more information.

All of the input signals are sampled at the rising edge of SRXCLK, and all of the output signals are updated at the rising edge of SRXCLK.

- Receive clock (SRXCLK)—This input is used to clock the ingress switch interface signals.
- Receive DATA BUS (SRXDATA0-SRXDATA15)—This three-state output data bus transmits data to the switch. When $\overline{\text{SRXENB}}$ is active, SRXDATA contains valid data for the switch. The used width of the data bus is determined by the ISWD bit in the ISWCR register. When the ISWD bit is clear, SRXDATA8-SRXDATA15 are 0 and not valid.
- Receive data bus parity (SRXPRTY)—This three-state output is the parity protection of SRXDATA transmitted to the switch. The type of parity (even or odd) is defined in Section 7.1.6.5, “Ingress Switch Interface Configuration Register (ISWCR).”
- Receive start of cell (SRXSOC)—This three-state output, when high, indicates that the current data on SRXDATA is the first byte or word of a cell structure (including the overhead bytes).

- Receive switch cell available (SRXCLAV)—This output, when asserted (high), indicates that the MC92520 has a cell ready to transfer to the switch. When deasserted, it indicates that currently there is no data available for the switch.
- Receive enable ($\overline{\text{SRXENB}}$)—This input, when low, enables new values on SRXDATA, SRXPRTY, and SRXSOC.

9.2.4 Egress Switch Interface Signals

The following signals relate to the egress switch interface. Refer to Section 4.2.2, “Single-PHY Transmit Interface (Egress)” and Section 7.1.6.6, “Egress Switch Interface Configuration Register (ESWCR)” for more information.

All of the input signals are sampled at the rising edge of STXCLK, and all of the output signals are updated at the rising edge of STXCLK.

- Transmit clock (STXCLK)—This input signal is used to clock the egress switch interface signals.
- Transmit data bus (STXDATA0–STXDATA15)—This input data bus receives data from the switch. When $\overline{\text{STXENB}}$ is asserted, STXDATA is sampled into the MC92520 on the rising edge of STXCLK. The used width of the data bus is determined by the ESWD bit in the ESWCR register. When the ESWD bit is clear, STXDATA8-STXDATA15 should be grounded.
- Transmit data bus parity (STXPRTY)—This input is the parity over STXDATA. The type of parity (even or odd) and the parity check control are defined in Section 7.1.6.5, “Ingress Switch Interface Configuration Register (ISWCR).” This input is ignored if $\overline{\text{STXENB}}$ is deasserted or the parity check is disabled. It is sampled on the rising edge of STXCLK.
- Transmit start of cell (STXSOC)—This input indicates, when high, that the current data is the first byte or word of a cell structure (including overhead). This input is sampled on the rising edge of STXCLK when $\overline{\text{STXENB}}$ is asserted.
- Transmit enable ($\overline{\text{STXENB}}$)—This input, when low, enables STXDATA, STXPRTY, and STXSOC.
- Transmit cell available (STXCLAV)—This output, when asserted (high), indicates that the MC92520 is prepared to receive a complete cell.
- Transmit multi-PHY address (STXADDR0–STXADDR4)—This input is only used if multi-PHY operation is enabled via the ESUM bit. The address bus indicates the PHY ID number of the PHY ports being polled or selected by the interface user.
- Transmit multi-PHY address valid (STXAVALID)—This is an optional input and only used in combination with a set ESAV bit. This signal is asserted together with STXADDR and indicates that the transmit address is valid. When in use, the input can be used to differentiate between a valid PHY id of 0x1F and the null address.

9.2.5 External Memory Signals

The following signals relate to the external memory interface. Refer to Section 4.4, “External Memory Interface” for more information.

- External memory data bus (EMDATA31–EMDATA0)—This tri-statable, bi-directional bus is the data path between the MC92520 and external memory.
- External memory address bus (EMADD23–EMADD2) and ($\overline{\text{EMADD23}}$–$\overline{\text{EMADD19}}$)—This output bus is the general address bus used by the MC92520 to access the external memory. $\overline{\text{EMADD23}}$–$\overline{\text{EMADD19}}$ are the logical inverses of EMADD23—EMADD19.
- External memory write ($\overline{\text{EMWR}}$)—When asserted (low), this output signal indicates that the current cycle to the external memory is a write cycle. This signal is active low and is asserted within the cycle.
- External memory word select high ($\overline{\text{EMWSH}}$)—This active-low output signal is used to select the high word of external memory for write operations. The MC92520 is the sole master of the external memory (EM) interface. The microprocessor can access the EM during setup mode, or during operate mode maintenance slots, by using the external memory address space. Due to the bandwidth needed from the EM and the requirement of one access per ZCLKI cycle, the EM interface is designed to work with pipelined ZBT RAM, and not with any other type of RAM. During a setup mode write access from the microprocessor, the value detected on $\overline{\text{MWSH}}$ is driven on the $\overline{\text{EMWSH}}$ signal.
- External memory word select low ($\overline{\text{EMWSL}}$)—This active-low output signal is used to select the low word of the external memory for write operations. During a setup mode write access from the microprocessor, the value detected on $\overline{\text{MWSL}}$ is driven on the $\overline{\text{EMWSL}}$ signal.
- External memory output enable ($\overline{\text{EMOE}}$)—This output signal is intended to connect to the external memory asynchronous output enable pins. It is used to prevent bus contention while the chip is being reset. It is activated at all times other than during reset. This signal is active low.

9.2.6 External Address Compression Interface Signals

The following signals relate to the external address compression interface. Refer to Section 4.4.3, “External Address Compression Device Interface” for more information.

- External address compression data bus (EACDATA31–EACDATA0)—This tri-statable bidirectional bus is the data path between the MC92520 and external address compression memory.
- External address compression start match ($\overline{\text{EACSM}}$)—This active-low output signal is used to start a match search in the external address compression device. It is intended to connect to the SM pin of a CAM.

- External address compression output enable ($\overline{\text{EACOE}}$)—This active-low output signal is used to enable the output of the external address compression device. It is intended to connect to the G pin of a CAM. It asserts multiple clock cycles before the CAM match cycle is complete, therefore it is not a critically-timed signal.
- External address compress clock (EACCLK)—This is a buffered form of ACLK intended to be used to clock the external address compressions device (CAM).

9.2.7 Control Signals

These signals are used to control the MC92520.

- ATMC power-up reset ($\overline{\text{ARST}}$)—This input signal is used for power-up reset of the entire chip. It must be asserted for at least the time required by the PLL.
- ATMC mode (AMODE)—This input signal enables some of the chip's internal test features. In normal usage this pin should be grounded.
- Test scan enable (TEST_SE)—This input enables the internal test scan chain multiplexers for scan shifting. In normal usage this pin should be grounded.

9.2.8 Microprocessor Signals (MP)

The following signals relate to the microprocessor interface. Refer to Section 4.5, "Microprocessor Interface" for more information.

- MP clock (MCLK)—This input signal is used as the microprocessor clock inside the MC92520. This signal drives the microprocessor logic in the MC92520. The duty cycle should be in the range of 40–60%. This clock must be no less than 1/2 the frequency of ZCLKI and must be no more than twice the frequency of ZCLKI.
- MP data bus (MDATA31–MDATA0)—This three-state bidirectional bus provides the general data path between the MC92520 and the microprocessor.
- MP address bus (MADD25–MADD2)—This input bus contains the address which is used by the microprocessor to define the register being accessed. This bus is used by the MC92520 at the assertion of MSEL and sampled on the rising edge of MCLK.
- MP select ($\overline{\text{MSEL}}$)—This input signal is used to determine that the current access to the MC92520 is valid. This signal is active low and sampled by the MC92520 on the rising edge of MCLK.
- MP write ($\overline{\text{MWR}}$)—This input signal is used to determine whether the MP is reading from the MC92520 or writing to it. This signal is active low and sampled by the MC92520 on the rising edge of MCLK. The MC92520 drives MDATA when MSEL = 0 and MWR = 1.

- MP word write enable high ($\overline{\text{MWSH}}$)—This input signal indicates that the high word is being written, and that valid data is on the MP data bus. During a setup mode external memory write access, the value detected on $\overline{\text{MWSH}}$ is driven to the $\overline{\text{EMWSH}}$ signal. This signal is active low and is sampled on the rising edge of MCLK.
- MP word write enable low ($\overline{\text{MWSL}}$)—This input signal indicates that the low word is being written, and that valid data is on the MP data bus. During a setup mode external memory write access, the value detected on $\overline{\text{MWSL}}$ is driven to the $\overline{\text{EMWSL}}$ signal. This signal is active low and is sampled on the rising edge of MCLK.

NOTE:

Table 9-1 describes the combined $\overline{\text{MWSH}}$ and $\overline{\text{MWSL}}$ functionality:

Table 9-1. Host Interface Fields

$\overline{\text{MWSH}}$	$\overline{\text{MWSL}}$	Function
0	0	Write D(31:0)
0	1	Write D(31:16)
1	0	Write D(15:00)

NOTE:

All cell extraction register, cell insertion register, and general register accesses are long-word (32-bit) accesses, so both $\overline{\text{MWSH}}$ and $\overline{\text{MWSL}}$ should be asserted low for these write accesses when write-enable mode is selected.

- MP data acknowledge0 ($\overline{\text{MDTACK0}}$)—This tri-statable output signal is used to indicate the end of an access from the MC92520. At the end of each access, this signal is actively pulled up and then released. This signal is active low and is output synchronous to the rising edge of MCLK.
- MP data acknowledge1 ($\overline{\text{MDTACK1}}$)—This tri-statable output signal is used to indicate the end of an access from the MC92520. At the end of each access, this signal is actively pulled up and then released. The user may program the MC92520 to disable this signal if it is not used. See Section 7.1.6.2, “Microprocessor Configuration Register (MPCONR)” for details. This signal is active low and is output synchronous to the rising edge of MCLK. If enabled in the MPCONR register, $\overline{\text{MDTACK1}}$ is identical to $\overline{\text{MDTACK0}}$.

- MP end cycle ($\overline{\text{MENDCYC}}$)—This active-low input is used to end the access after $\overline{\text{MDTACK}}$ has been asserted. In MPC860 applications, this input would normally be tied low (active). In MPC8260 applications, the MPC8260 asserts this input to indicate that it has seen the $\overline{\text{MDTACK}}$ signal. The 92520 samples this signal on the rising edge of MCLK.
- MP interrupt ($\overline{\text{MINT}}$)—This output signal is used to notify the microprocessor of the occurrence of interrupting events. This signal is asserted on the rising edge of ACLK (asynchronous with respect to MCLK).
- MP request 0 ($\overline{\text{MREQ0}}$)—This output signal can be programmed to one of two options but its default value is MP cell in request ($\overline{\text{MCIREQ}}$), see Section 9.2.8.1, “DMA Requests.” See Section 7.1.6.2, “Microprocessor Configuration Register (MPCONR)” for details on the $\overline{\text{MREQ0}}$ signal.
- MP request 1 ($\overline{\text{MREQ1}}$)—This output signal can be programmed to one of two options but its default value is MP cell out request ($\overline{\text{MCOREQ}}$), see Section 9.2.8.1, “DMA Requests.” See Section 7.1.6.2, “Microprocessor Configuration Register (MPCONR)” for details on the $\overline{\text{MREQ1}}$ signal.

NOTE:

$\overline{\text{MREQ0}}$ and $\overline{\text{MREQ1}}$ signals are fully backward compatible to the MC92500 Revision A $\overline{\text{MCIREQ}}$ and $\overline{\text{MCOREQ}}$ signals, respectively. The $\overline{\text{MREQ}}[n]$ signals are used by DMA devices and can be programmed to support DMA requests as explained in Section 9.2.8.1 below.

9.2.8.1 DMA Requests

Following are the available options for programming MP request signals:

- MP cell in request ($\overline{\text{MCIREQ}}$)—The $\overline{\text{MREQ}}[n]$ is an output signal that can be used by an external DMA device as a control line indicating when to start a new cell insertion cycle into the MC92520. It is asserted whenever the cell insertion register array is available to be written. Refer to Section 3.2.3.1, “Cell Insertion” for more information. This signal is active low and is output on the falling edge of MCLK.
- MP cell out request ($\overline{\text{MCOREQ}}$)—The $\overline{\text{MREQ}}[n]$ is an output signal that may be used by an external DMA device as a control line indicating when to start a new cell extraction cycle from the MC92520. It is asserted whenever the cell extraction register array is available to be read. Refer to Section 3.2.3.2, “Cell Extraction” for more information. This signal is active low and is output on the falling edge of MCLK.

9.2.9 PLL Signals

The following signals are connected to the analog PLL macro that must be used in the MC92520.

- ATMC master clock (ACLK)—This input signal is used by the PLL to generate the internal master clock of MC92520. The duty cycle should be in the range of 40–60%.
- TPA—This PLL test point output must be left open in normal application.
- ZCLKO—This is the clock generated by the PLL. If the PLL is enabled (not in bypass mode), ZCLKO is twice the frequency of ACLK. This pin should be routed directly back into the ZCLKI input, with no other loads other than an RC termination.
- ZCLKO_2—This is an additional ZCLKO output, used to disitribute ZCLKO to all of the external memory (EM) devices (RAMs).
- ZCLKI—This input must be connected to ZCLKO. It is used for all internal clocking of the ATMC. This external feedback path provides for minimized skew between the EM clock timing and the internal ATMC clock.

9.2.10 Test Signals

The following test signals are provided by the MC92520:

- Test clock (TCK)—This input pin is the JTAG clock. The TDO, TDI, and TMS pins are synchronized by this pin.
- Test mode select (TMS)—This input pin is sampled on the rising edge of TCK. TMS is responsible for the state change in the test access port state machine.
- Test data input (TDI)—This input pin is sampled on the rising edge of TCK. TDI is the data to be shifted toward the TDO output.
- Test data output (TDO)—This three-state output pin changes its logical value on the falling edge of TCK.
- Test reset ($\overline{\text{TRST}}$)—This input pin is the JTAG asynchronous reset. When asserted low, the test access port is forced to the test_logic_reset state. When JTAG is not being used, this signal should be tied to ARST or hard-wired to GND.

9.3 Electrical and Physical Characteristics

This section provides the following sets of physical and electrical specifications for the MC92520:

- Absolute maximum ratings
- Recommended operating conditions
- DC electrical characteristics

- Clocks
- Microprocessor interface timing
- PHY interface timing
- Switch interface timing
- External memory interface timing
- External address compression device (CAM) timing

9.3.1 Recommended Operating Conditions

Table 9-2 shows the operating conditions recommended to ensure functionality of the MC92520.

Table 9-2. Recommended Operating Conditions

Symbol	Parameter	Min	Max	Unit
Core V_{DD}	DC supply voltage for core, V_{DD} = 1.8V (nominal)	1.65	1.95	V
I/O V_{DD}	DC supply voltage for I/O pads, V_{DD} = 3.3V (nominal)	3.135	3.465	V
V_{in}	Input voltage	0	3.6	V
T_A	Operating temperature (ambient)	0	70	°C

All parameters are characterized for DC conditions after thermal equilibrium has been established.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{ss} or V_{DD}).

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit.

These are the recommended and test operating conditions. Proper device operation outside of these conditions is not guaranteed.

9.3.2 Absolute Maximum Ratings

Table 9-3 shows the absolute maximum stress ratings for the MC92520. Recommended operating conditions are shown in Table 9-2.

Table 9-3. Absolute Maximum Ratings

Symbol	Parameter	Value/Value Range[1]	Unit
Core V_{DD}	DC supply voltage for core logic	-0.3 to 1.95	V
I/O V_{DD}	DC supply voltage for I/O pads	-0.3 to 3.6	V
V_{IN}	LVTTL DC input voltage	-0.3 to 3.6	V
T_{STG}	Storage temperature	-55 to +150	°C

[1]Absolute maximum ratings are stress ratings only and functional operation at the maximum limits is not guaranteed. Stress beyond the maximum rating may affect device reliability or cause permanent damage to the device.

9.3.3 DC Electrical Characteristics

DC characteristics are shown in Table 9-4.

Table 9-4. DC Electrical Characteristics

Symbol	Parameter	Condition	Min.	Max.	Unit
Core V_{DD}	Internal core voltage supply	—	3.0	3.6	V
I/O V_{DD}	I/O pad voltage suppy	—	1.65	1.95	V
V_{IH}	Input high voltage	—	1.7		V
V_{IL}	Input low voltage	—		1.1	V
I_{IN}	Input leakage current	$V_{IN} = V_{DD}$ or V_{SS}	-1	1	µA
I_{IN}[1]	Input leakage current, $\overline{\text{MDTACK0/1}}$	$V_{IN} = V_{DD}$ or V_{SS}	16	66	µA
I_{IN}[2]	Input leakage current, ZCLKO_2	$V_{IN} = V_{DD}$ or V_{SS}	16	66	µA
I_{OH}	Output high current, outputs: EMDATAx, $\overline{\text{EMWR}}$, EMADDx, $\overline{\text{EMADDx}}$, $\overline{\text{EMWSH}}$, $\overline{\text{EMWSL}}$, ZCLKO, ZCLKO_2	Core V_{DD} = 1.65V, I/O V_{DD} = 3.0V, V_{OH} = 2.5V	—	13	mA
	Output high current, outputs: RXADDRx, $\overline{\text{RXENB}}$, TXDATAx, TXSOC $\overline{\text{TXENB}}$, TXPRTY, TXADDRx, SRXDATAx, SRXCLAV, SRXSOC, SRXPRTY, STXCLAV, $\overline{\text{MDTACKx}}$, MDATAx, EACDATAx, $\overline{\text{EACSM}}$, EACCLK,	Core V_{DD} = 1.65V, I/O V_{DD} = 3.0V, V_{OH} = 2.5V	—	10	mA
	Output high current, outputs: $\overline{\text{MREQx}}$, $\overline{\text{MINT}}$, $\overline{\text{EMOE}}$, $\overline{\text{EACOE}}$, TDO, and test-only outputs: $\overline{\text{STXENB}}$, STXPRTY, STXSOC	Core V_{DD} = 1.65V, I/O V_{DD} = 3.0V, V_{OH} = 2.5V	—	5	mA
I_{OL}	Output low current, outputs: EMDATAx, $\overline{\text{EMWR}}$, EMADDx, $\overline{\text{EMADDx}}$, $\overline{\text{EMWSH}}$, $\overline{\text{EMWSL}}$, ZCLKO, ZCLKO_2	Core V_{DD} = 1.65V, I/O V_{DD} = 3.0V, V_{OH} = 0.4V	-10	—	mA

Table 9-4. DC Electrical Characteristics (Continued)

Symbol	Parameter	Condition	Min.	Max.	Unit
	Output low current, outputs: RXADDRx, RXENB, TXDATAx, TXSOC TXENB, TXPRTY, TXADDRx, SRXDATAx, SRXCLAV, SRXSOC, SRXPRTY, STXCLAV, MDTACKx, MDATAx, EACDATAx, EACSM, EACCLK,	Core V_{DD} = 1.65V, I/O V_{DD} = 3.0V, V_{OH} = 0.4V	-10	—	
	Output low current, outputs: MREQx, MINT, EMOE, EACOE, TDO, and test-only outputs: STXENB, STXPRTY, STXSOC	Core V_{DD} = 1.65V, I/O V_{DD} = 3.0V, V_{OH} = 0.4V	-5	—	
I_{OZ}	Output leakage current, Tri-state output	Output = high impedance V_{OUT} = V_{DD} or V_{SS}	-5	5	μA
C_I	Input capacitance	—	—	TBD	pF

T_A = 0˚C to 70˚C, I/O V_{DD}=3.3 V ±0.165 V guaranteed

Inputs may be modified to include pullup resistors at any time.

See Section 9.2, “Signal Description for pin input/output type.

Note 1: Pins MDTACK0 and MDTACK1 have an internal pull-up, 50k - 100k ohms.

Note 2: Pin ZCLKO_2 has an internal pull-down, 50k - 100k ohms.

9.3.4 Clocks

Clock timings are shown in Table 9-5, and diagrams for each are shown in Figure 9-2.

Table 9-5. Clock Timing

Num	Characteristics	Min	Max	Unit
C1	ACLK cycle time[1]	20	80	ns
C2	ACLK pulse width low	8	—	ns
C3	ACLK pulse width high	8	—	ns
C4	ACLK rise/fall time	—	2	ns
C5	MCLK cycle time[2]	15	—	ns
C6	MCLK pulse width low	6	—	ns
C7	MCLK pulse width high	6	—	ns
C8	MCLK rise/fall time	—	2	ns
C9	SRXCLK cycle time[3]	15	—	ns
C10	SRXCLK pulse width low	6	—	ns
C11	SRXCLK pulse width high	6	—	ns
C12	SRXCLK rise/fall time	—	2	ns
C13	STXCLK cycle time[3]	15	—	ns

Table 9-5. Clock Timing (Continued)

Num	Characteristics	Min	Max	Unit
C14	STXCLK pulse width low	6	—	ns
C15	STXCLK pulse width high	6	—	ns
C16	STXCLK rise/fall time	—	2	ns
C17	ZCLKI input requirements - must be driven by ZCLKO[4]	—	---	ns

[1]ACLK Max cycle time given is for PLL enabled with PLLRR[PICR] = 1 (slow PLL range). Max cycle time is 40ns for PLLRR[PICR] = 0. There is no max cycle time if the PLL is disabled.

[2]MCLK must be at least 1/2 the frequency of ZCLKI, but no more than twice the frequency of ZCLKI. Note that ZCLKI frequency = ACLK frequency if the PLL is disabled. ZCLKI = 2 x ACLK if the PLL is enabled.

[3] For OC-12 performance, SRXCLK and STXCLK should not need to run at the specified minimum cycle time of 15ns.

[4]ZCLKI must be driven from ZCLKO with no other loads on this connection besides an RC termination. Output ZCLKO_2 should be used to drive all External Memory devices connected to the “EM” interface pins.

[5]There are no inter-clock timing requirements between ACLK, MCLK, SRXCLK and STXCLK. They can be asynchronous with respect to each other.

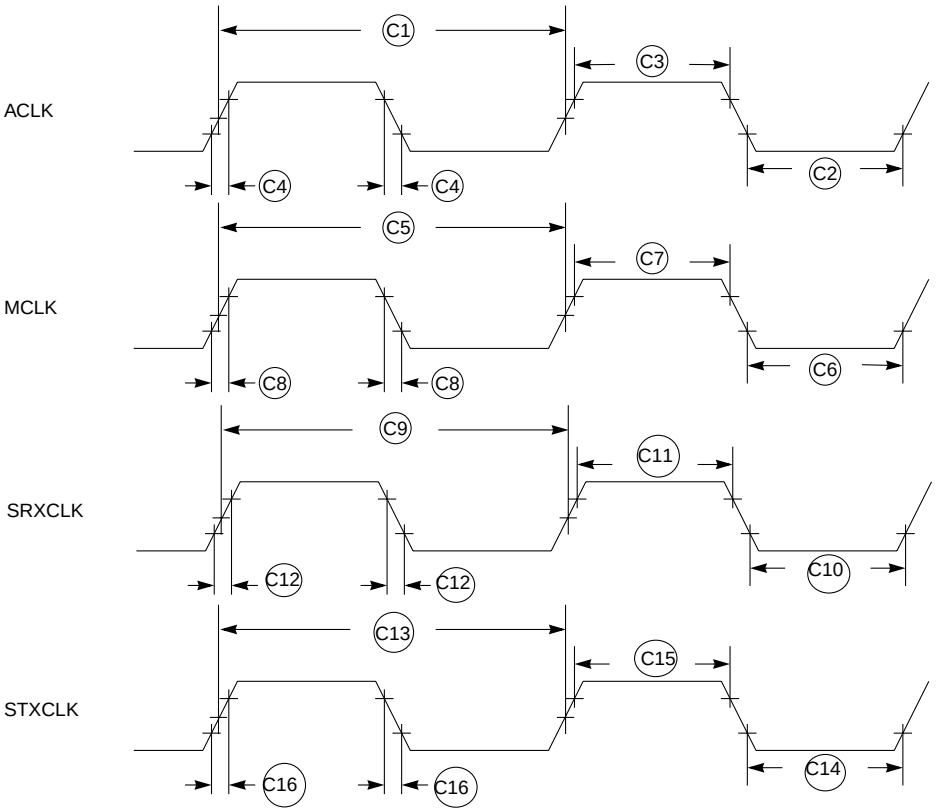


Figure 9-2. Clock Timing Diagrams

9.3.5 Microprocessor Interface Timing

The timing diagrams in this section are intended to convey setup and hold values for input signals and propagation delay values for output signals. For functional timing diagrams, see Section 4.5, "Microprocessor Interface."

Table 9-6. Microprocessor Interface Timings

Num	Characteristics	Min	Max	Unit
1	$\overline{\text{MSEL}}$ setup time before MCLK rising edge	1.5		ns
2	$\overline{\text{MSEL}}$ hold time after MCLK rising edge	0.6		ns
3	MADD/$\overline{\text{MWR}}$ setup time before MCLK rising edge	0.5		ns
4	MADD/$\overline{\text{MWR}}$ hold time after MCLK rising edge	1.5		ns

Table 9-6. Microprocessor Interface Timings (Continued)

Num	Characteristics	Min	Max	Unit
5	$\overline{\text{MWSH}}$, $\overline{\text{MWSL}}$, $\overline{\text{MENDCYC}}$ setup time before MCLK rising edge	1.4		ns
6	$\overline{\text{MWSH}}$, $\overline{\text{MWSL}}$, $\overline{\text{MENDCYC}}$ hold time after MCLK rising edge	0.8		ns
7	MDATA delay after MCLK rising edge (processor read cycle)	2	13	ns
8	MDATA setup time before MCLK rising edge (processor write cycle)	1		ns
9	MDATA hold time after MCLK rising edge (processor write cycle)	1		ns
10	MCLK rising to $\overline{\text{MDTACK1,0}}$ on	2	8	ns
11	MCLK rising to $\overline{\text{MDTACK1,0}}$ rising or falling[1]		7.8	ns
12	MCLK falling to $\overline{\text{MDTACK1,0}}$ off	2	5.5	ns
13	MCLK rising to $\overline{\text{MREQ1,0}}$ rising/falling		9.6	ns

[1]Max delays are for 25pF lumped capacitive load. Add 1.6ns to $\overline{\text{MDTACK}}$ delays for 50pF load. Min delays are for unloaded outputs.

Figure 9-3 shows timing diagrams for the microprocessor interface signals.

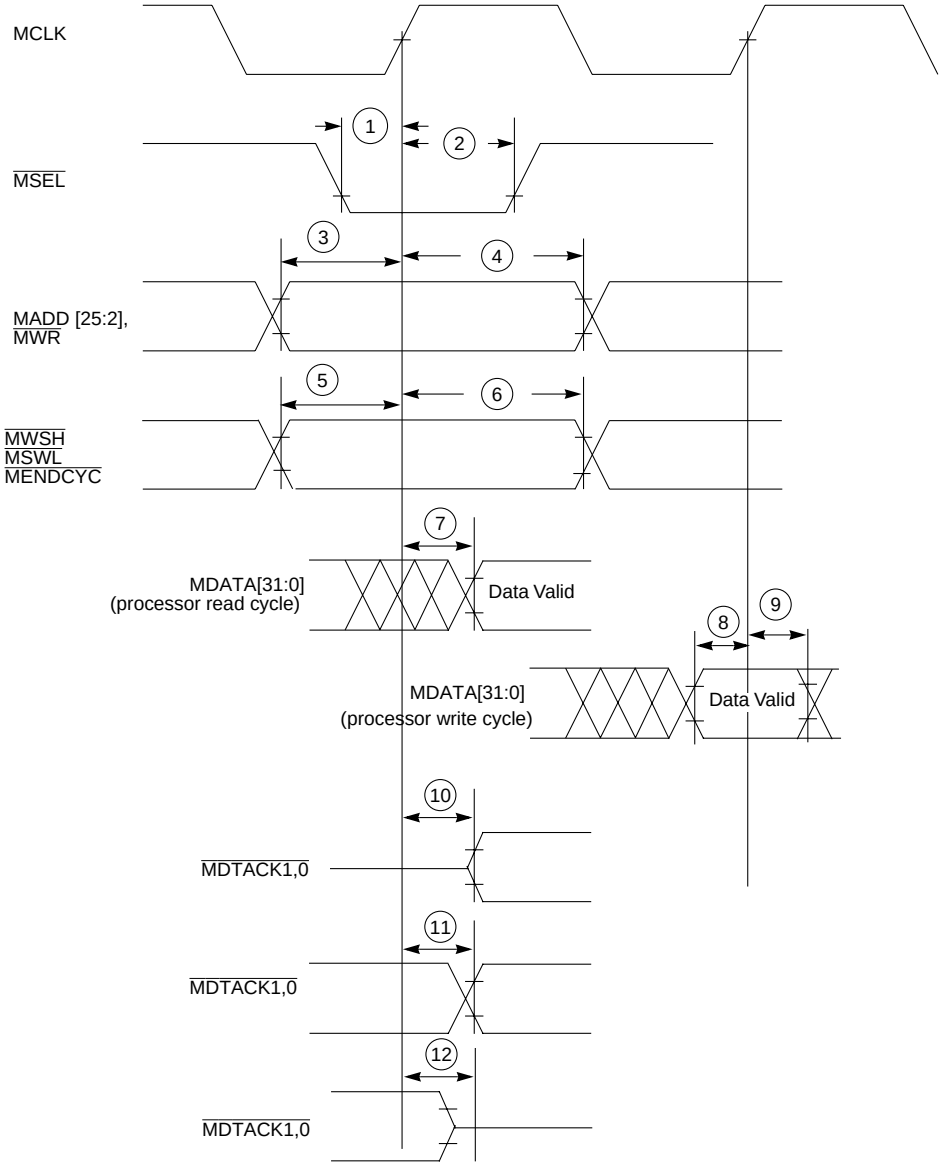


Figure 9-3. Processor Interface Timing

Timing diagrams for DMA request signals are shown in Figure 9-4.

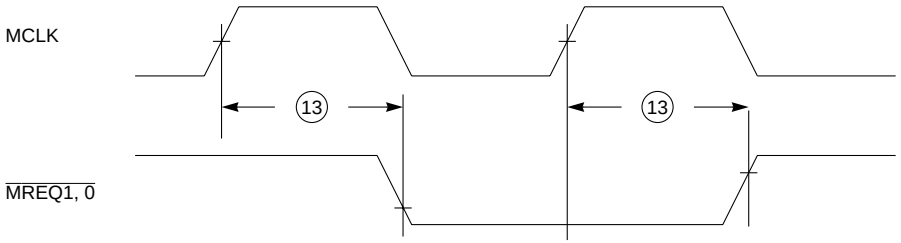


Figure 9-4. DMA Request Signals Timing

9.3.6 PHY Interface Timing

Table 9-7 shows the times for the receive and transmit PHY interface signals.

Table 9-7. PHY Interface Timings

Num	Characteristics	Min	Max	Unit
51	Setup time before ACLK rising edge (all TX and RX inputs)	2.5	—	ns
52	Hold time after ACLK rising edge	1	—	ns
53	Propagation delay from rising edge of ACLK	1.5	10	ns

Max propagation delay for a 50 pF load. Add 3.2ns for 100pF load.
Min propagation delay for an unloaded output.

Timing diagrams for the receive PHY interface signals are shown in Figure 9-5.

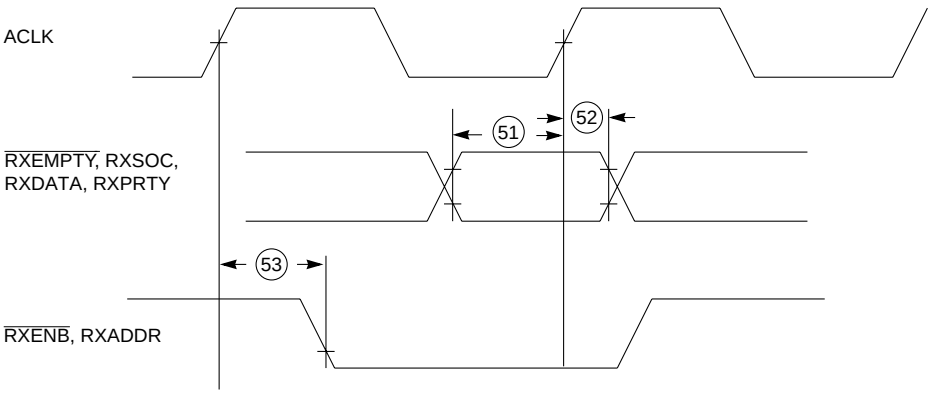


Figure 9-5. Receive PHY Interface Timing

Transmit PHY interface timing diagrams are shown in Figure 9-6.

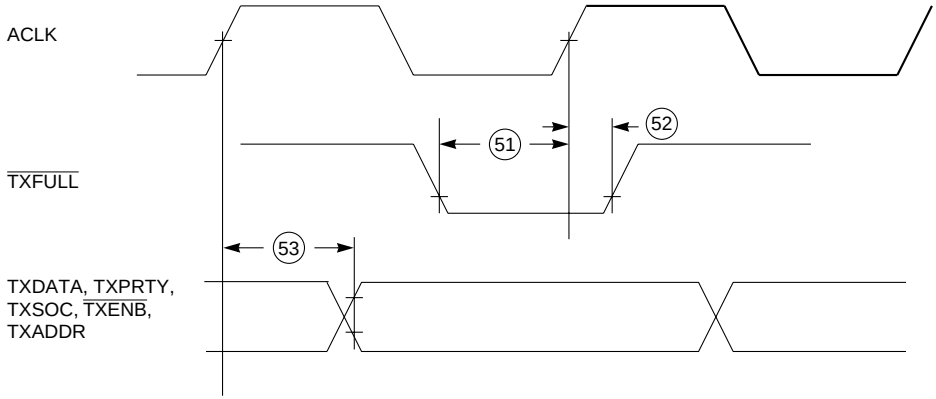


Figure 9-6. Transmit PHY Interface Timing

9.3.7 Ingress Switch Interface Timing

Times for the ingress switch interface signals are shown in Table 9-8.

Table 9-8. Ingress Switch Interface Timing

Num	Characteristics	Min	Max	Unit
61	Setup time before SRXCLK rising edge	2		ns
62	Hold time after SRXCLK rising edge	1		ns
63	Propagation delay from rising edge of SRXCLK	1.5	8.5	ns
64	SRXCLK rising edge to outputs active	1.5	9.0	ns
65	SRXCLK rising edge to outputs inactive	1.5	5.0	ns

Max propagation delay for a 50 pF load. Add 3.2ns for 100pF load.
Min propagation delay for an unloaded output.

Figure 9-7 shows the timing diagrams for the ingress switch interface signals.

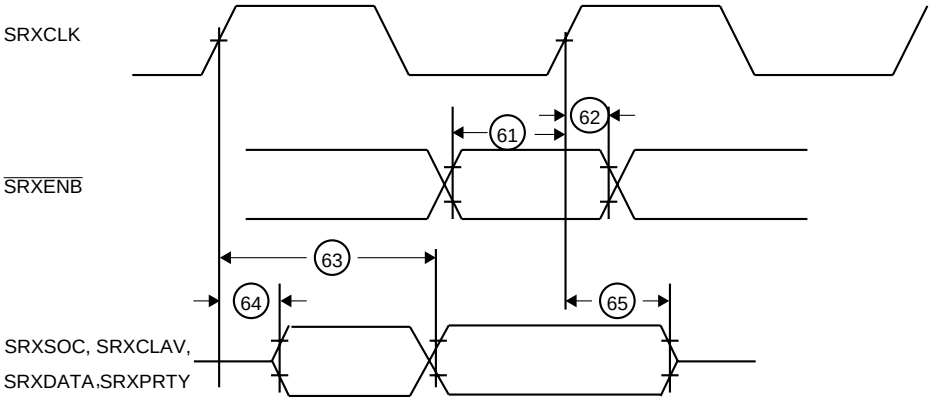


Figure 9-7. Ingress Switch Interface Timing

9.3.8 Egress Switch Interface Timing

Timings for the egress switch interface signals are shown in Table 9-9.

Table 9-9. Egress Switch Interface Timing

Num	Characteristics	Min	Max	Unit
71	Setup time before STXCLK rising edge	2		ns
72	Hold time after STXCLK rising edge	1		ns
73	Propagation delay from rising edge of STXCLK	1.5	9.2	ns
74	STXCLK rising edge to outputs active	1.5	9.7	ns
75	STXCLK rising edge to outputs inactive	1.5	5.0	ns

Max propagation delay for a 50 pF load. Add 3.2ns for 100pF load.
Min propagation delay for an unloaded output.

Figure 9-8 shows the timing diagrams for the egress switch interface.

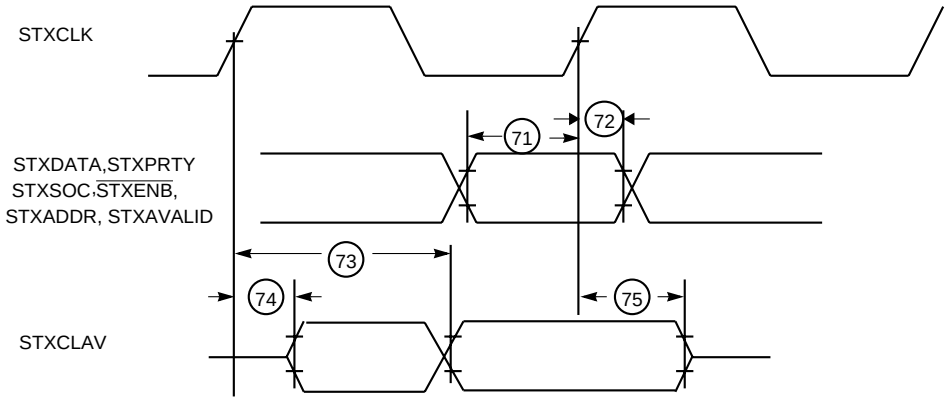


Figure 9-8. Egress Switch Interface Timing

9.3.9 External Memory Interface Timing

Table 9-10 displays the times for external memory interface signals.

Table 9-10. External Memory Interface Timing

Num	Characteristics	Min	Max	Unit
81	ZCLKI rising to EMWSL, EMWSH, EMWR valid	2	7.8	ns
82	ZCLKI rising to EMADD[23:2], EMADD[23:19] valid	2	7.8	ns
83	ZCLKI rising to EMDATA[31:0] drivers on (write cycle)	2	8	ns
84	ZCLKI rising to EMDATA{31:0] drivers off (write cycle)	2.5	5	ns
85	ZCLKI rising to EMDATA[31:0] valid (write cycle)	2	7.8	ns
86	EMDATA[31:0] setup time before ZCLKI rising (read cycle)	0	—	ns
87	EMDATA[31:0] hold time after ZCLKI rising (read cycle)	2	—	ns

Max propagation delays are given for a 25 pF load. However, output drivers are optimized driving the EM RAM inputs. See application notes.

Min propagation delay for an unloaded output.

Note that EMOE only switches at reset. It is used to prevent bus contention during reset. It is not a time-critical signal.

External memory interface timing diagrams are shown in Figure 9-9.

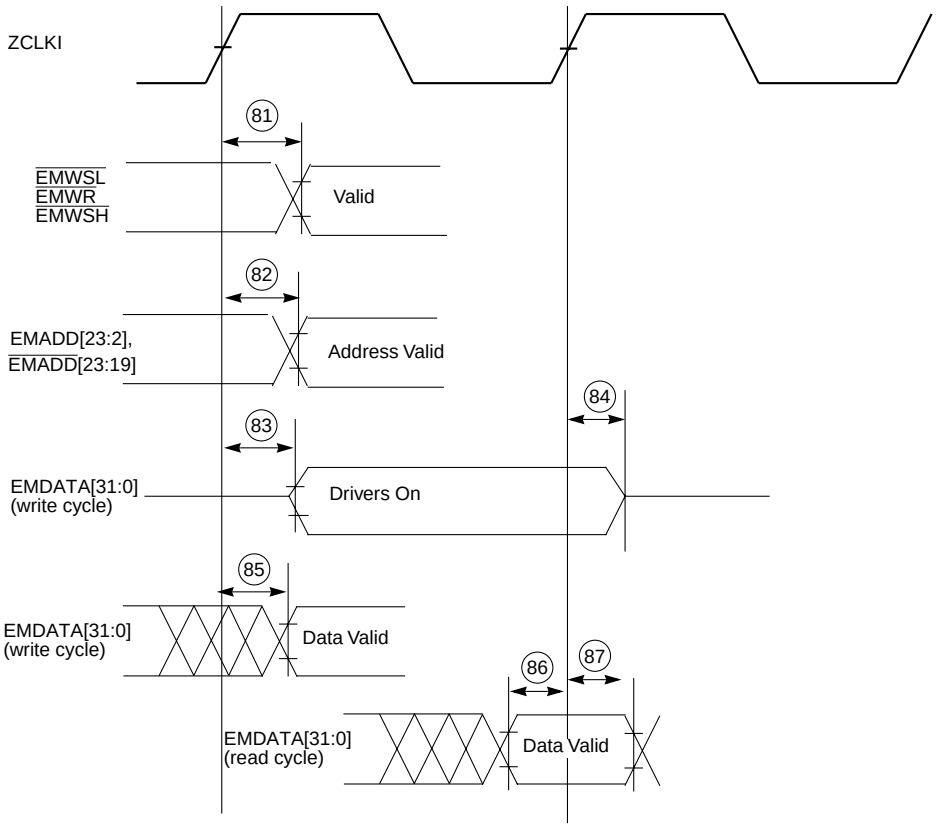


Figure 9-9. External Memory Interface Timing

9.3.10 External Address Compression Interface Timing

External address compression interface signal times are shown in Table 9-11.

Table 9-11. External Address Compression Interface Timing

Num	Characteristics	Min	Max	Unit
91	ACLK rising input to EACCLK rising output[1]	1.7	6.7	
92	ACLK rising to $\overline{\text{EACSM}}$ rise/fall	2	7.8	ns
93	ACLK rising to $\overline{\text{EACOE}}$ rise/fall[2]	---	---	ns
94	ACLK rising to EACDATA[31:0] valid (match data output)	2	8.25	ns
95	EACDATA[31:0] valid setup to EACCLK rise (CAM match result)[3]	---	---	ns
96	EACDATA[31:0] valid hold to EACCLK rise (CAM match result)[3]	---	---	ns

Max propagation delays are given for a 25 pF load. Min delays are given for unloaded outputs.

[1]The use of EACCLK output is optional. As CAM devices may require that “Start Match” (driven by $\overline{\text{EACSM}}$) switches while the CAM’s clock input is high, the use of EACCLK as a delayed clock to the CAM may prevent timing issues to the CAM Start Match input. The delay from ACLK to EACCLK is guaranteed to be at least 0.25ns less than the delay from ACLK to $\overline{\text{EACSM}}$ or EACDATA under the same loading conditions.

[2]EACOE asserts multiple clock cycles before the CAM “Match Complete” asserts. Therefore, this specification is not needed.

[3]System is designed such that CAM match results will occur many EACCLK cycles before it is needed by the MC92520, and will be available for multiple clocks after being sampled. Therefore, these specifications are not needed.

Timing diagrams for the external address compression signals are shown in Figure 9-10.

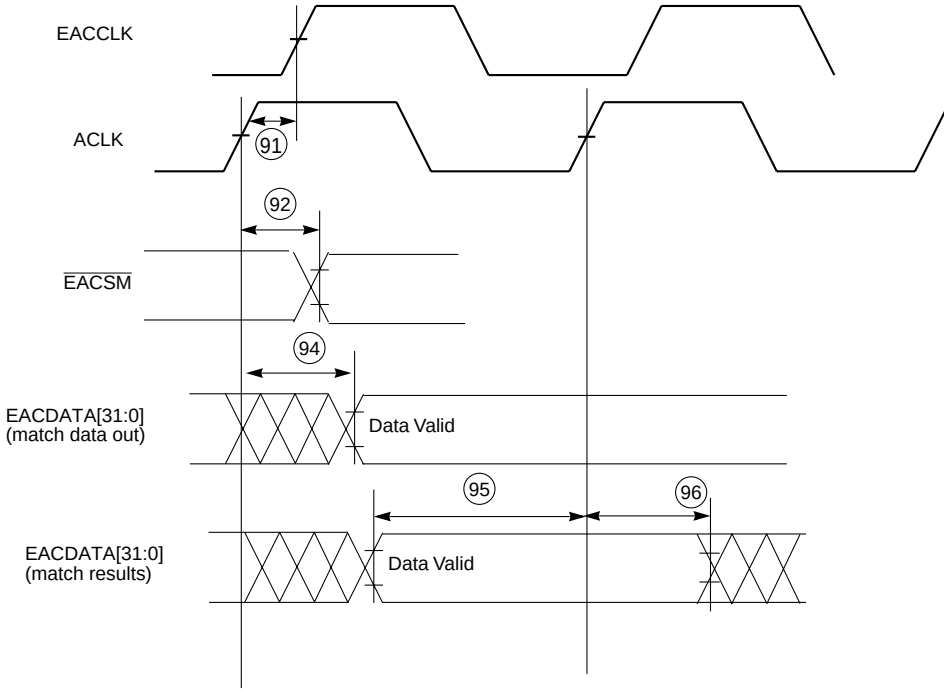


Figure 9-10. External Address Compression Interface TIming

9.4 Ordering Information

Order the MC92520 with the order number PC92520GC (PBGA).

9.5 Mechanical Data

Mechanical data includes:

- Pin assignments
- Package dimensions

9.5.1 Pin Assignments

Figure 9-11 shows the MC92520 pin assignments.

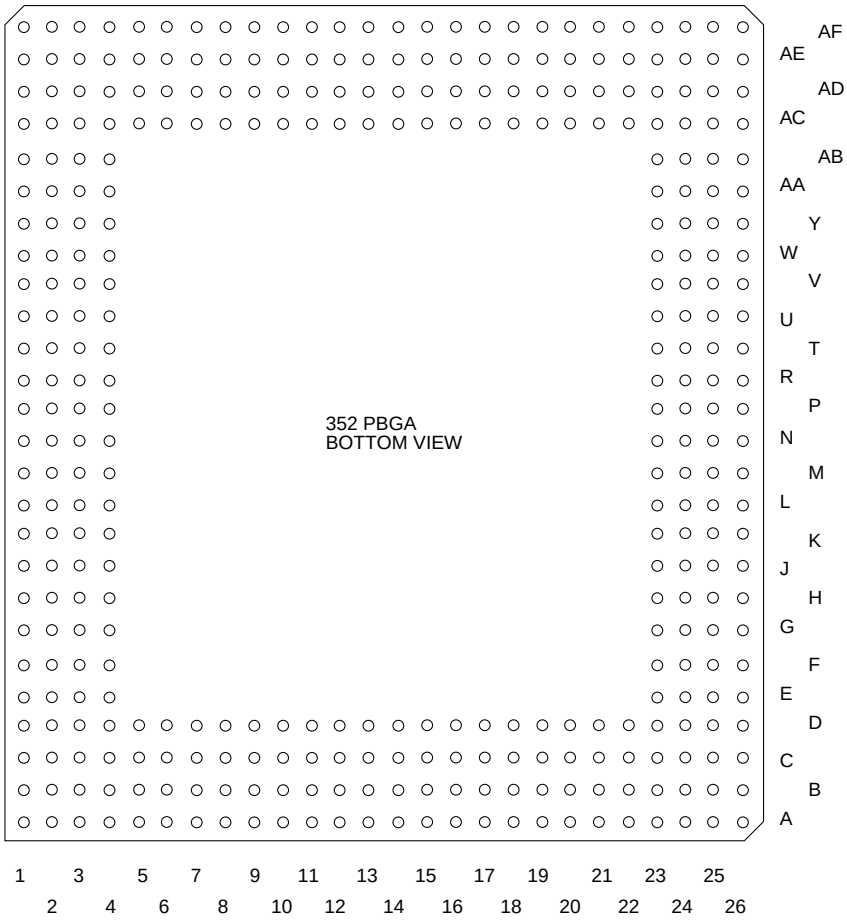


Figure 9-11. 352-Pin PBGA Pin Diagram

Table 9-12 provides signal type information for the power pins.

Table 9-12. Power Pin Assignment

Signal Type	Pin Assignment
Core V_{DD}	G2, M1, T1, Y2, AF10, AF17, Y26, R26, L26, H24, A17, D10
Core V_{SS}	G4, L1, P1, W1, AD9, AD16, Y24, T24, M23, F25, D17, B9
I/O V_{DD}	D6, D11, D16, D21, F4, F23, L4, L23, T4, T23, AA4, AA23, AC6, AC11, AC16, AC21
I/O V_{SS}	A1, A2, A26, B2, B25, B26, C3, C24, D4, D9, D14, D19, D23, H4, J23, N4, P23, V4, W23, AC4, AC8, AC13, AC18, AC23, AD3, AD24, AE1, AE2, AE25, AF1, AF25, AF26
AV_{DD}[1]	A4
AV_{SS}[1]	B4

[1]To eliminate coupling of digital switching noise into the PLL through pins AV_{DD} and AV_{SS}, connect these pins to isolated power and ground.

The complete pin assignment list is provided in Table 9-13.

Table 9-13. MC92520 Functional Pin Assignment

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
B1	EMDATA31	C1	EMDATA30	C2	EMDATA29	D2	EMDATA28	D3	EMDATA27
D1	EMDATA26	E2	EMDATA25	E4	EMDATA24	E3	EMDATA23	E1	EMDATA22
F2	EMDATA21	F3	EMDATA20	F1	EMDATA19	G3	EMDATA18	G1	EMDATA17
H2	EMDATA16	J4	EMDATA15	H1	EMDATA14	H3	EMDATA13	J2	EMDATA12
J1	EMDATA11	K2	EMDATA10	J3	EMDATA9	K1	EMDATA8	K4	EMDATA7
L2	EMDATA6	K3	EMDATA5	M2	EMDATA4	L3	EMDATA3	N2	EMDATA2
M4	EMDATA1	N1	EMDATA0	M3	$\overline{\text{EMOE}}$	P2	EMADD23	P4	$\overline{\text{EMADD23}}$
N3	EMADD22	R2	$\overline{\text{EMADD22}}$	P3	EMADD21	R1	$\overline{\text{EMADD21}}$	T2	EMADD20
R3	$\overline{\text{EMADD20}}$	R4	EMADD19	U2	$\overline{\text{EMADD19}}$	T3	EMADD18	U1	EMADD17
U4	EMADD16	V2	EMADD15	U3	EMADD14	V1	EMADD13	W2	EMADD12
V3	EMADD11	W4	EMADD10	Y1	EMADD9	W3	EMADD8	AA2	EMADD7
Y4	EMADD6	AA1	EMADD5	Y3	EMADD4	AB2	EMADD3	AB1	EMADD2
AA3	$\overline{\text{EMWSH}}$	AC2	$\overline{\text{EMWSL}}$	AB4	$\overline{\text{EMWR}}$	AD2	EACDATA31	AB3	EACDATA30
AC3	EACDATA29	AD1	EACDATA28	AF2	EACDATA27	AE3	EACDATA26	AE4	EACDATA25
AF3	EACDATA24	AD4	EACDATA23	AF4	EACDATA22	AC5	EACDATA21	AE5	EACDATA20
AD5	EACDATA19	AE6	EACDATA18	AF5	EACDATA17	AC7	EACDATA16	AF6	EACDATA15
AD6	EACDATA14	AE7	EACDATA13	AD7	EACDATA12	AF7	EACDATA11	AE8	EACDATA10
AF8	EACDATA9	AC9	EACDATA8	AD8	EACDATA7	AF9	EACDATA6	AE9	EACDATA5
AE10	EACDATA4	AC10	EACDATA3	AE11	EACDATA2	AD10	EACDATA1	AF11	EACDATA0
AE12	EACCLK	AF12	$\overline{\text{EACSM}}$	AD11	$\overline{\text{EACOE}}$	AE13	RXADDR4	AC12	RXADDR3
AF13	RXADDR2	AD12	RXADDR1	AE14	RXADDR0	AC14	$\overline{\text{RXENB}}$	AF14	RXDATA15
AD13	RXDATA14	AE15	RXDATA13	AD14	RXDATA12	AF15	RXDATA11	AE16	RXDATA10

Table 9-13. MC92520 Functional Pin Assignment (Continued)

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
AD15	RXDATA9	AF16	RXDATA8	AC15	RXDATA7	AE17	RXDATA6	AC17	RXDATA5
AE18	RXDATA4	AF18	RXDATA3	AD17	RXDATA2	AF19	RXDATA1	AE19	RXDATA0
AE20	RXPRTY	AD18	RXSOC	AC19	RXEMPTY	AD19	TXDATA15	AF20	TXDATA14
AE21	TXDATA13	AF21	TXDATA12	AC20	TXDATA11	AD20	TXDATA10	AF22	TXDATA9
AE22	TXDATA8	AD21	TXDATA7	AC22	TXDATA6	AE23	TXDATA5	AF23	TXDATA4
AE24	TXDATA3	AD22	TXDATA2	AD23	TXDATA1	AF24	TXDATA0	AE26	TXPRTY
AD25	TXSOC	AB23	TXADDR4	AC25	TXADDR3	AD26	TXADD2	AC24	TXADDR1
AC26	TXADDR0	AB25	TXENB	AB24	TXFULL	AA25	MDATA31	Y23	MDATA30
AA24	MDATA29	AA26	MDATA28	Y25	MDATA27	W25	MDATA26	V23	MDATA25
W26	MDATA24	W24	MDATA23	V25	MDATA22	V26	MDATA21	U25	MDATA20
V24	MDATA19	U26	MDATA18	U23	MDATA17	T25	MDATA16	U24	MDATA15
T26	MDATA14	R25	MDATA13	P25	MDATA12	R23	MDATA11	P26	MDATA10
R24	MDATA9	N25	MDATA8	N23	MDATA7	N26	MDATA6	P24	MDATA5
M25	MDATA4	N24	MDATA3	M26	MDATA2	L25	MDATA1	M24	MDATA0
K25	MDTACK1	L24	MDTACK0	K26	MINT	K23	MREQ1	J25	MREQ0
K24	MCLK	J26	MWR	H25	MSEL	H26	MENDCYC	J24	MWSH
G25	MWSL	H23	MADD25	G26	MADD24	G23	MADD23	F26	MADD22
E25	MADD21	G24	MADD20	F24	MADD19	E26	MADD18	E23	MADD17
D25	MADD16	E24	MADD15	D26	MADD14	C25	MADD13	D24	MADD12
C26	MADD11	A25	MADD10	B24	MADD9	A24	MADD8	C23	MADD7
B23	MADD6	B22	MADD5	A23	MADD4	C22	MADD3	D22	MADD2
B21	SRXCLK	A22	SRXENB	D20	SRXDATA15	A21	SRXDATA14	C21	SRXDATA13
B20	SRXDATA12	C20	SRXDATA11	A20	SRXDATA10	B19	SRXDATA9	A19	SRXDATA8
D18	SRXDATA7	C19	SRXDATA6	B18	SRXDATA5	A18	SRXDATA4	B17	SRXDATA3
C18	SRXDATA2	B16	SRXDATA1	C17	SRXDATA0	A16	SRXPRTY	B15	SRXSOC
A15	SRXCLAV	C16	STXCLAV	B14	STXSOC	D15	STXPRTY	A14	STXENB
C15	STXDATA15	B13	STXDATA14	D13	STXDATA13	A13	STXDATA12	C14	STXDATA11
B12	STXDATA10	C13	STXDATA9	A12	STXDATA8	B11	STXDATA7	C12	STXDATA6
A11	STXDATA5	D12	STXDATA4	B10	STXDATA3	C11	STXDATA2	A10	STXDATA1
C10	STXDATA0	A9	STXCLK	B8	STXADDR4	A8	STXADDR3	B7	STXADDR2
C9	STXADDR1	A7	STXADDR0	D8	STXAVALID	B6	AMODE	C8	TEST_SE
A6	TCK	D7	TRST	B5	TDI	C7	TMS	A5	TDO
C6	TPA	D5	ARST	C5	ACLK	B3	ZCLKO	C4	ZCLKO_2
A3	ZCLKI								

9.5.2 Package Dimensions

For a detailed package drawing, please refer to:

Motorola document number: 98ASS23639W

Motorola Case Outline: 1124-01

Appendix A
UPC/NPC Design

This chapter describes specific design parameters.

A.1 Time-Stamped Leaky Time Bucket

A UPC design is normally conceptualized as a bucket with a hole in the bottom that allows water to drain out at a steady rate. Each admitted cell is considered a fixed quantity of water added to the bucket. Arriving cells are admitted at the rate at which the bucket drains. Any cell that would cause the bucket to overflow is not admitted. The water drainage rate represents the average data flow bandwidth. Cell processing at each update is defined by the following equation:

new_contents = old_contents – [elapsed_time / avg_time_between_cells] + 1

where

- new_contents is the new contents of the bucket in cells
- old_contents is the previous contents of the bucket in cells
- elapsed_time is the amount of time that has elapsed since the previous cell arrived
- avg_time_between_cells is the expected elapsed time between consecutive cells, in units of time per cell.

NOTE:

The term [elapsed_time / avg_time_between_cells] represents the number of cells that drained from the bucket since the previous update.

The same flow control can also be achieved by re-dimensioning the quantities involved. The size of the hole (representing the bandwidth) may be held constant, while the quantity of water added with each cell may be varied. In effect, this changes the dimension of the bucket from cells to time. The amount of water in the bucket (the current bucket contents) now represents the amount of time that should pass without any cells being admitted if we wish to produce the stated average bandwidth. The processing of the bucket can now be obtained by:

new_contents = old_contents – elapsed_time + avg_time_between_cells

where

- new_contents is the new contents of the bucket in units of time
- old_contents is the previous contents of the bucket in units of time
- elapsed_time is the amount of time elapsed since the previous cell arrived
- avg_time_between_cells is the average amount of time that should elapse between consecutive cells.

NOTE:

This equation is obtained from the previous one by multiplying each term by the avg_time_between_cells. The primary advantage of this proposal, as compared to the equivalent design in which the bucket is dimensioned in units of cells, is to eliminate a complicated floating point divider circuit that would be needed to divide the elapsed_time by the avg_time_between_cells.

The following synopsis of the UPC design uses the method described above. The bucket content value indicates how much time must elapse before the bucket becomes empty. A time-stamp indicates the arrival time of the most recently admitted cell. The next cell arrival subtracts the time-stamp from the current time to compute the elapsed time. It subtracts this elapsed time value from the bucket content value indicating that less time is now required to deplete the bucket fully. As long as the bucket contents value is less than a defined threshold, the arriving cell is admitted to the cell flow, and the bucket value is incremented by a programmable increment value (avg_time_between_cells), indicating that more time is required to deplete the bucket.

A.2 Multi-Enforcer UPC/NPC

The MC92520 UPC/NPC design provides up to four enforcers per connection. These enforcers may be configured to check compliance with the peak cell rate and/or the sustainable cell rate for combinations of CLP = 0, CLP = 1, and CLP = 0 + 1 cell streams. The enforcers are applied to the cell stream in a serial manner, such that a “tag” decision by one enforcer results in the cell being treated as belonging to the CLP = 1 cell stream by subsequent enforcers.

Each enforcer produces one of four results:

1. Bypass—The cell is outside the scope of this enforcer
2. Pass—The cell is conforming
3. Tag—The cell is non-conforming and the tag option is in effect
4. Discard—The cell is non-conforming and the discard option is not in effect

The combined result of the enforcers is a single decision regarding the cell:

- Admit as is,
- Admit and tag, or
- Discard.

As each cell is processed, the bucket parameters of all of the enforcers are updated as follows:

- The time-stamp is set to the current time value
- The elapsed time is subtracted from the bucket contents of each enforcer.
- If the cell is admitted, each enforcer that produced a “pass” is updated by adding the avg_time_between_cells to the bucket contents.

A.3 Data Structure

The UPC/NPC enforcement process uses the data structure shown in Figure A-1. The context parameters table contains a pointer (BKT_PTR) to a set of entries in the bucket table which hold the key parameters that customize the UPC/NPC function for a specific connection. See Section 7.2.3.4, “Common Parameters,” and Section 7.2.11, “Buckets Record,” for the physical structure of the external memory tables.

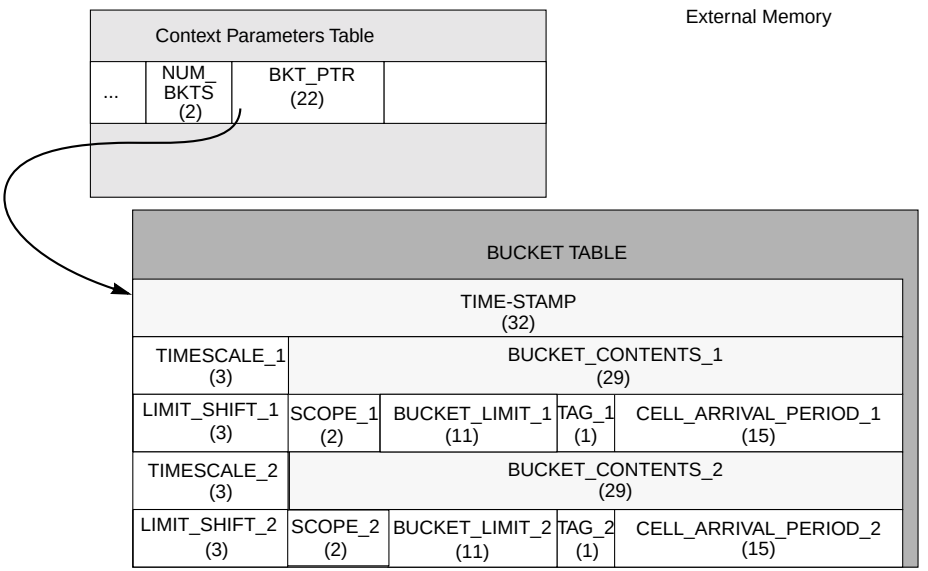


Figure A-1. UPC Data Structures

Note the following details:

- The BKT_PTR field points to the time-stamp word. If the bucket pointer is all 1s, it is the NULL pointer, indicating NO buckets are applied to the connection.
- The NUM_BKTS (NBK) field in the context parameters table indicates how many leaky bucket enforcers (1–4) are active on this connection. The buckets are sequential in the bucket memory address space following the time-stamp word. See Section 7.2.3.4, “Common Parameters.”
- The BUCKET_CONTENTS and TIME-STAMP fields are dynamic fields that are updated whenever a cell is admitted.
- All other entries in the bucket table are normally defined upon connection setup. They are considered to be static parameters in that they are not changed by the UPC/NPC enforcement process.

A.3.1 Detailed Flowchart

A flow chart of the multi-enforcer UPC design is shown in Figure A-2.

NOTE:

If the cell is discarded, no write back occurs.

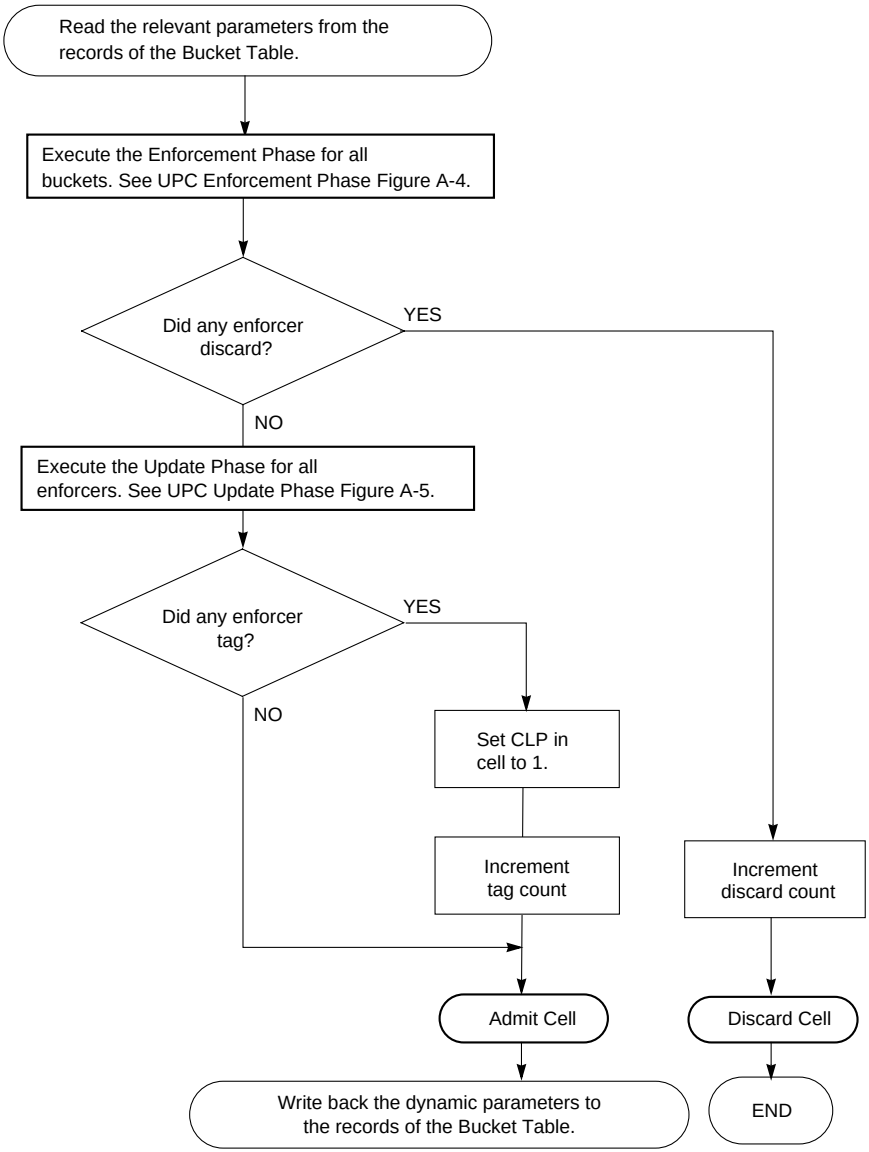


Figure A-2. Detailed UPC Flowchart

A.4 Control/Data Flowcharts

The control/data flowchart shown in Figure A-3 shows the functional operation of the UPC design. A global timer computes the current time. The end result of the process is to update

the time-stamp and the bucket contents values in the bucket table, to make an admit or discard judgement for every submitted cell, and to determine the CLP of the cell to be admitted.

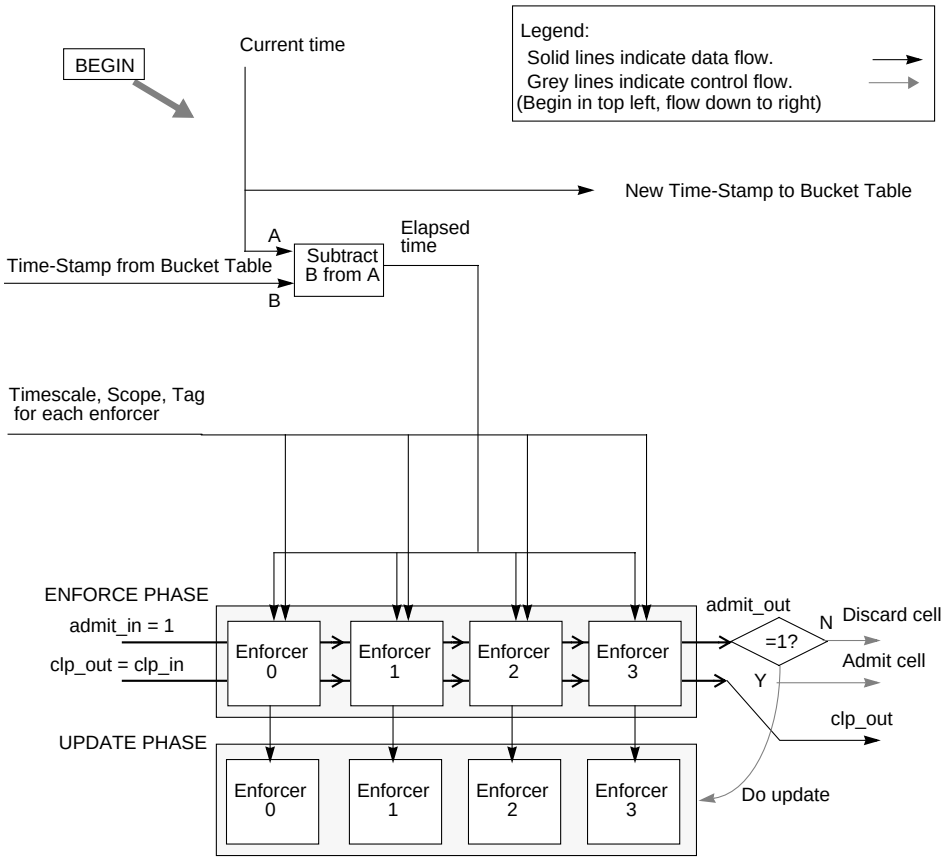


Figure A-3. UPC Control and Data Flowchart

Since the time-stamp stored in the bucket record is 32 bits, the elapsed time is also limited to 32 bits. If a connection is silent for more than 2^{32} cell times (> 45 minutes), the current time wraps, and there is a small probability that a cell is discarded unnecessarily. This unlikely occurrence can be prevented if the microprocessor clears the bucket contents of any connection that is silent for long periods of time. Another possible solution is to ensure that OAM continuity check cells are occasionally transmitted on such a connection. The CC cells trigger updates of the bucket record.

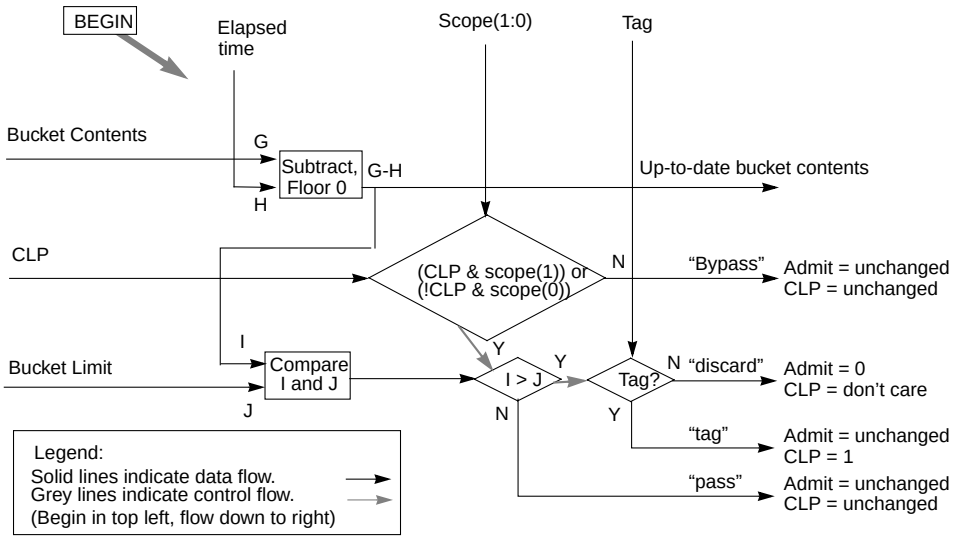


Figure A-4. UPC Enforcement Phase

In the enforcement phase each enforcer block executes the operations shown in Figure A-4. The elapsed time is subtracted from the previous bucket contents to compute the up-to-date bucket contents. The bucket contents value has a floor of zero applied such that it never goes negative. The CLP is checked against the scope field from the bucket record for the enforcer according to Table A-1. If it is not in the scope, the enforcer is bypassed, that is, further processing is aborted, and no changes are made to ADMIT or CLP. If the CLP is in the scope, the up-to-date bucket contents value is compared to the bucket limit to determine if the enforcer should pass the cell. If the bucket contents value is larger than the bucket limit, the result of the enforcement is either "tag" or "discard" depending on the setting of the tag bit from the bucket record.

Table A-1. Enforcer SCOPE Field

SCOPE (1:0)	CLP = 1	CLP = 0
00	Bypass	Bypass
01	Bypass	Operate
10	Operate	Bypass
11	Operate	Operate

If the enforcer passed the cell, no changes are made to ADMIT or CLP. If the enforcement result is "discard," ADMIT is reset so that the cell is discarded. If the enforcement result is

“tag,” CLP is set. Note that the tag option is only relevant if the scope field is 01 (operate only on CLP = 0 cells). If an enforcer produces a “tag” result, subsequent enforcers operates on this cell only if SCOPE(1) is set. Therefore, if the tag option is used in any of the enforcers, the order of the bucket records becomes significant.

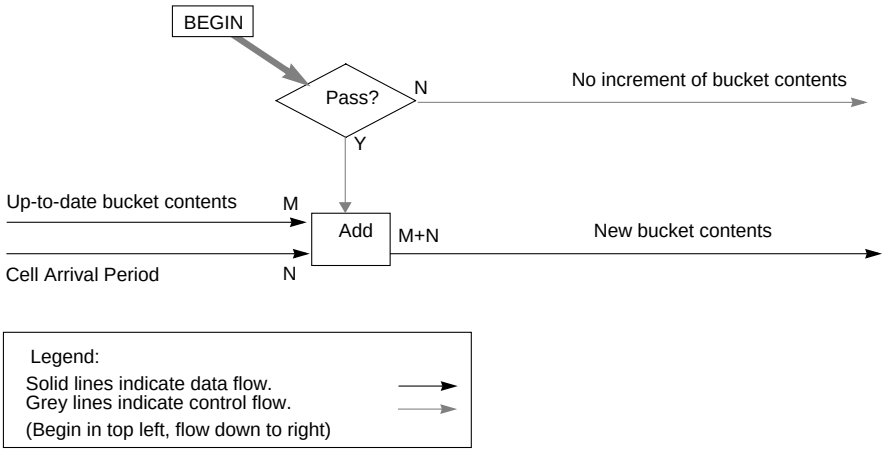


Figure A-5. UPC Update Phase

The update phase takes place only if the processed cell is actually admitted, i.e. no enforcer produced a “discard” result. In the update phase each enforcer that passed the cell increments the bucket contents as shown in Figure A-5.

A.5 Bucket Parameter Encoding

The choice of an encoding format for the cell arrival period (CAP) is driven by the need to keep enforcement errors small. With insufficient bits allocated to this field, the granularity of enforcement becomes coarse, and at high bandwidths, the difference between two settings 64 Kb/s apart becomes indistinguishable. The enforcement errors result from rounding the cell arrival period to a finite number of bits. The cell arrival period should always be rounded down, not up, since errors in the user’s favor are preferable as they do not affect the quality of service of the connection. Using 15 bits with timescales that shift the decimal point 2 bits at a time produces a worst case error of 2^{-13} of the enforced bandwidth. This results in a maximum error smaller than 32 Kb/s for bandwidths up to STS-3c.

The bucket contents (BKC) field needs the same precision as the CAP, and it must contain larger values in order to accommodate a burst. 29 bits are used for the contents value which means that bursts of at least 16,000 cells can be accommodated. If larger bursts are needed, they can be obtained by reducing the precision of the CAP and BKC values. This is done by changing the value of the timescale field and filling the MSBs of the CAP with 0s. The

justification for this trade-off is that the precision of the average bandwidth (as specified by the CAP) becomes less meaningful as we allow greater variations from this value over long periods of time (i.e. larger bursts).

The CAP and BKC fields are dimensioned in units of time. The basic unit is referred to as a cell time. It is defined as the period of the rate at which the MC92520 processes the ATM cells. This number can be derived by multiplying the number of clock cycles used to process a cell (64) by the period of ZCLK, the clock signal provided to the MC92520. For example, using a 25 MHz ZCLK results in a cell time of 2.56 μs. Using a 100MHz ZCLK results in a cell time of 0.64 μs.

Eight timescale values are used for encoding the CAP and BKC fields. The timescale defines the units of these fields as a fraction of a cell time. The timescale value can be shifted by setting GCR[TSSH]. Table A-2 contains the units of the CAP and BKC fields as a function of the timescale with TSSH = 0.

Table A-2. Cell Arrival Period and Bucket Contents Encoding, TSSH = 0

Timescale	Units of Cell Arrival Period (cell times)	Number of Bits the Elapsed Time is Shifted
0	1	0
1	1/4	2
2	1/16	4
3	1/64	6
4	1/256	8
5	1/1024	10
6	1/4096	12
7	1/16384	14

Table A-3 contains the units of the CAP and BKC fields as a function of the timescale with TSSH = 1. In terms of implementation, the timescale would indicate how much to shift the elapsed time (which is in whole cell time units) to the left before subtracting it from the BKC.

Table A-3. Cell Arrival Period and Bucket Contents Encoding, TSSH = 1

Timescale	Units of Cell Arrival Period (cell times)	Number of Bits the Elapsed Time is Shifted
0	1	No TSC shift. Shift CAP left 2 bits
1	1	0
2	1/4	2
3	1/16	4
4	1/64	6
5	1/256	8
6	1/1024	10
7	1/4096	12

To enforce a bandwidth of *N* cells per second, the cell arrival period needed is 1/*N* seconds. This must be normalized to units of MC92520 cell times by dividing by the duration of a cell time. This leads to the following equation:

$$CAP = \frac{1}{N} / (CellTime) = [N \cdot CellTime]^{-1}$$

A.5.1 Cell Arrival Computation Example

Here is a quick example of the computation of the cell arrival period. We wish to enforce a bandwidth of 20,012 cells per second. A cell time is 0.64 μs, as defined above. Therefore, the cell arrival period is 78.0782. The best precision is obtained by expressing this value in smaller units. With TSSH = 1, and checking Table A-3, we find that using timescale 5 where the units are 1/256 of a cell time is the best we can do, since timescale 6 would require multiplying the cell arrival period by 1024 which would overflow the 15 bits allocated to the CAP. Multiplying 78.0782 by 256 to convert it to timescale 5 yields 19,988.01. We round this down to 19,988 and encode this value in the CAP field.

A.5.2 Bucket Limit Encoding

The size of the bucket determines the amount that the cells can get ahead of the bandwidth. The criterion for conformance is that adding a cell arrival period to the bucket would not cause it to overflow. In order to simplify the calculations, the value used by the MC92520 to specify the bucket size is the bucket limit (BKL) which is defined as one cell arrival period less than the required bucket size. By comparing the up-to-date bucket contents to the BKL, the MC92520 can determine if there is room in the bucket for a cell arrival period without performing the addition.

The BKL is defined in units of cell times. Its value may range from a few cell times (constant bit rate with little jitter) up to tens of thousands of cell times (large bursts). However, the precision of this field does not need to be extraordinarily high. Therefore, we can save space by providing a limited number of bits in which to encode the BKL, along with a shift factor. The BKL is chosen to be an 11-bit field. The shift factor indicates how much to shift the BKL to the left before the bucket contents value is compared to it. Table A-4 defines the shift of the BKL field in terms of the limit shift (LMS) field.

Table A-4. Bucket Limit Encoding

Limit Shift (LMS)	Number of Bits the Bucket Limit (BKL) is Shifted
0	0
1	3
2	6
3	9
4	12
5	15
6	18
7	Reserved

A.6 UPC Parameter Calculations

The following examples demonstrate how the static UPC parameters may be calculated based on the traffic parameters of the connection.

A.6.1 Example A

This example involves a 64 Kb/s data connection providing circuit emulation using AAL 1, allowing a 10% jitter. A connection that provides 64 Kbps using AAL 1 needs 170.21 cells per second. Using the formula given above with a cell time of 0.64μs, the cell arrival period should be 9179.84 cell times which can be approximated by encoding the CAP as 0x23DB (decimal 9179) using timescale 1 (9179 × 1) with TSSH = 1. A jitter of 10% means that a cell can arrive up to 9179.84/10 = 917.98 cell times early without being discarded. Therefore, the bucket size used by this connection should be 1.10 × 9179.84 = 10097.8. Since timescale 1 is being used, the bucket size should be rounded up to the nearest 1 cell time (10,098). The bucket limit is calculated by subtracting one CAP value (9179) from the bucket size (10,098), yielding 919. Thus, the value to be placed in the BKL field is 0x397, and the LMS value is 0.

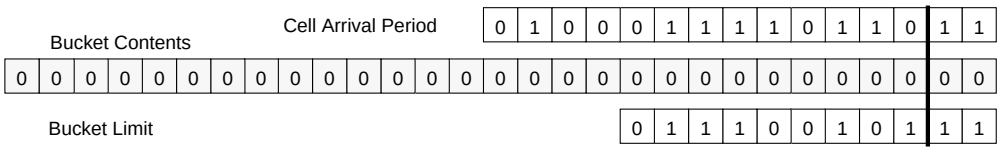


Figure A-6. Encoding of Bucket Parameters for 64 Kbps Circuit Emulation

Figure A-6 shows the encoding of the bucket parameters for this example. The actual bandwidth being enforced is 64.0048 Kb/s with a jitter of 10.01%. Note that the errors are small and in the user's favor.

A.6.2 Example B

This example involves a connection with a sustainable cell rate of STS-1 with a burst of twice the bandwidth for up to 400 milliseconds and an all-out burst of up to 200 microseconds. To enforce two different burst sizes, we use two buckets. One bucket uses the average connection bandwidth with a burst of twice the bandwidth for 400 ms. The other bucket takes the double bandwidth as its base bandwidth and allows an all-out burst for 200 μs.

An STS-1 connection (51.840 Mb/s at the physical layer) uses 116,830.19 cells per second. This is equivalent to a cell arrival period of 13.374112 cell times. This value is approximated by encoding the CAP as 0x357F (decimal 13695) using timescale 6 and with TSSH = 1 (13695/1024 = 13.374023). The burst of twice the bandwidth for 400 ms is 2 × 0.400 × 116,830.19 = 93,464 cells. When this is multiplied by the cell arrival period, the resulting bucket size is 1,249,998 cell times. Expressed in terms of timescale 6, the bucket size is 1,249,998 × 1024 = 1,279,997,956. The bucket contents (BKC) field has 29 bits which do not suffice to contain this value. Therefore, we choose to compromise precision in order to increase the bucket size. We now encode the CAP as 0x0D5F using timescale 5 (3423/256 = 13.371094). Recalculating the bucket size using this value produces 93,464 × 13.371094 = 1,249,715.9 Converting the bucket size to timescale 5 involves multiplying by 256 and rounding up, yielding 319,927,272. The bucket limit is obtained by subtracting the cell arrival period (3423) from the bucket size, yielding 319,923,849. Since the bucket limit is large, this example uses limit shift = 6. To determine the BKL after the shift, we must divide by 2^{18} (since LMS=6 provides a shift of 18 bits). The result is rounded up to yield 1221, or 0x4C5. These results are illustrated in Figure A-7.

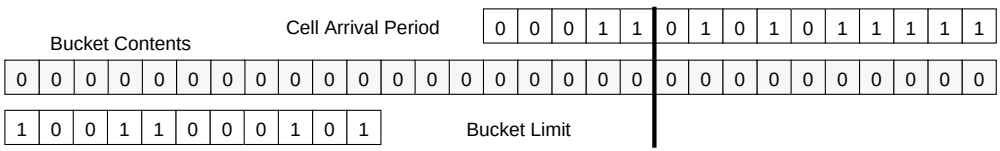


Figure A-7. Encoding of Bucket 1 Parameters

The second bucket uses a base rate of 2 × 116,830.19 = 233,660.38 cells per second. This is equivalent to a cell arrival period of 6.687056. This value is approximated by encoding the CAP as 0x6AFE using timescale 7 (27,390 / 4096 = 6.687012). The continuous burst for 200μs is 70.6415 cells. When this is multiplied by the cell arrival period, the resulting bucket size is 472.381 cell times, or in timescale 7 terms, 1,934,871. The bucket limit is 1,934,871. This value requires only 21 bits, so LMS = 4 can be used. Dividing by 2^{12} and rounding up produces a BKL value of 466, or 0x1D2. The encoding of these values is shown in Figure A-8.

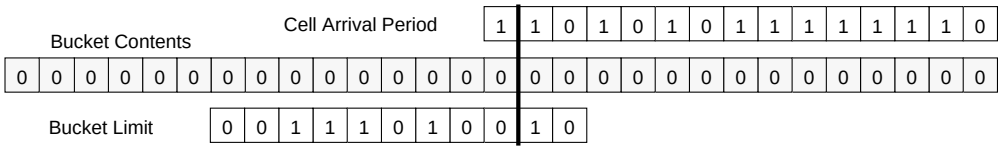


Figure A-8. Encoding of Bucket 2 Parameters

A.7 Bellcore Cell Relay Service Parameters

Bellcore [14] lists supported values of peak cell rate (PCR) and cell delay variation tolerance (CDVT) for constant bit rate (CBR) connections. For variable bit rate (VBR) connections, supported values of PCR, CDVT, sustainable cell rate (SCR), and maximum burst size (MBS) are provided.

The requirements call for a CDVT of 250 μs for all connections and an MBS of 210 cells for VBR connections. The CAP and BKL for CBR connections can be calculated with the formulas:

CAP = Frequency / (64 * PCR)

BKL = 0.00025 * Frequency / 64

Similarly, the CAP and BKL for VBR connections can be calculated with the formulas:

CAP = Frequency / (64 * SCR)

BKL = (0.00025 + 209 * (1/SCR - 1/PCR)) * Frequency / 64

In the following, tables A-5 and A-6 show precalculated CAP, BKL, and shift parameters assuming a ZCLK frequency of 100 MHz and GCR[TSSH] = 1.

Table A-5. Bucket Parameters for CBR Connections

PCR (cells/s)	CAP (Hex)	TSC	BKL (Hex)	LMS
0[1]	0x7FFF	0	0x187	0
173	0x2347	1	0x187	0
346	0x468F	2	0x61B	0
1038	0x5E14	3	0x30E	1
2076	0x2F0A	3	0x30E	1
4140	0x5E5A	4	0x187	2
8280	0x2F2D	4	0x187	2
16,560	0x5E5A	5	0x61B	2
119,910	0x341F	6	0x30E	3

[1] The minimum enforceable cell rate is approximately 11.9 cells/s.

Table A-6. Bucket Parameters for VBR Connections

PCR (cells/s)	CAP (Hex)	TSC	BKL (Hex)	LMS	SCR (cells/s)	CAP (Hex)	TSC	BKL (Hex)	LMS
173	0x2347	1	0x187	0	18	0x54C5	0	0x1F1	5
					87	0x4627	1	0x1C8	4
346	0x468F	2	0x61B	0	35	0x2B98	0	0x100	5
					173	0x2347	1	0x735	3
1038	0x5E14	3	0x30E	1	104	0x3AB0	1	0x2B2	4
					519	0x2F0A	2	0x134	4
2076	0x2F0A	3	0x30E	1	208	0x7560	2	0x564	4
					1038	0x5E14	3	0x268	4
3622	0x6BD9	4	0x187	2	363	0x4341	2	0x317	4
					1811	0x35EC	3	0x162	4
4140	0x5E5A	4	0x187	2	414	0x3AF8	2	0x2B6	4
					2070	0x2F2D	3	0x136	4
12,420	0x7DCE	5	0x61B	2	1242	0x4EA0	3	0x39E	4
					6210	0x3EE7	4	0x1A1	4
28,980	0x35EA	5	0x61B	2	2898	0x21B2	3	0x18E	4
					14,490	0x6BD5	5	0x2D9	4
33,120	0x2F2D	5	0x61B	2	3312	0x75F1	4	0x571	4
					16,560	0x5E5A	5	0x281	4
45,540	0x224F	5	0x61B	2	4554	0x55C6	4	0x3F7	4
					22,770	0x449E	5	0x1D9	4
96,000	0x411A	6	0x30E	3	9600	0x28B0	4	0x1E5	4
					48,000	0x208D	5	0x769	3
120,060	0x340E	6	0x30E	3	12,006	0x2089	4	0x185	4
					60,030	0x681D	6	0x30A	4
273,240	0x5B7E	7	0x187	4	27,324	0x392F	5	0x2B9	4
					136,620	0x2DBF	6	0x18D	4
353,207	0x46C7	7	0x187	4	35,321	0x2C3C	5	0x221	4
					176,604	0x2363	6	0x149	4

Appendix B
Maintenance Slot Calculations

B.1 Maintenance Slot Equations

The following variables are used in the maintenance slot calculations:

- R_L—Link rate measured in cells per second = Link rate in bps divided by 53 bytes/cell and 8 bits/byte
- C—Number of clocks per cell slot is a design constant (64)
- P—Period of a cell slot
- f—ZCLK frequency, a system design parameter
- I_S—Interval of maintenance slots measured in cell slots, a programmed value chosen by the user
- I_T—Interval of maintenance slots measured in time
- N_{CS}—Number of cell slots per second
- N_{FS}—Number of free cell slots second
- N_{MS}—Number of maintenance slots per second
- N_{ES}—Number of empty cell slots per second

The equations used in the maintenance slot calculations include:

- The period of an MC92520 cell slot is 64 ZCLK periods:

  $P = C / f$
- The number of cell slots per second is the inverse of a cell slot period:

  $N_{CS} = 1 / P = f / C$
- Free slots are cell slots that are not used for processing cells arriving from the PHY layer. The number of free slots per second is computed by subtracting the link cell rate from the total number of MC92520 cell slots:

  $N_{FS} = N_{CS} – R_L$
- Free slots can be used for one of two purposes. Each free slot is either declared a maintenance slot, or it is left empty to be used for processing inserted cells if necessary. The fraction of cell slots used as maintenance slots is determined by the

value of the maintenance period length (MPL) field (see Section B.4, “Maintenance Slot Parameters”) programmed by the user:
I_S = MPL + 1. The remaining free slots are left empty.

$N_{MS} = N_{CS} / I_S$

$N_{ES} = N_{FS} – N_{MS}$

- The time between maintenance slots is calculated by multiplying the number of cell slots between maintenance slots (maintenance slot interval) by the period of a cell slot:

  $I_T = I_S * P$

B.2 Maintenance Time Slot Example 1

We take as an example a configuration in which ZCLK is 25 MHz, and the link rate is STS-3c (149.76 Mb/s net bit rate after the SONET overhead is removed). Therefore,

- R_L = 149.76 E + 6 / (53 * 8) = 353,207 cells/sec
- f = 25 E + 6 Hz
- P = C / f = 64 / 25 E + 6 = 2.560 μs
- N_{CS} = f / C = 25 E + 6 / 64 = 390,625 cells/s

The MC92520, when running at 25 MHz, has 390,625 cell processing slots available per second. Of these, 353,207 are used for processing the cells received from the STS-3c link. Therefore, the number of free cell slots is:

- $N_{FS} = N_{CS} – R_L$ = 390,625 – 353,207 = 37,418 cell slots per second

At this point the user must calculate how many maintenance slots are needed for the maintenance tasks to be performed by the microprocessor (or DMA device). Note that each maintenance slot is the same length as a cell processing slot (P = 2.560 μs). Let us assume that at least 25,000 maintenance slots are needed per second. We must now calculate the maintenance slot interval which is the fraction of the cell processing slots used as maintenance slots.

- $N_{MS} = N_{CS} / I_S$

so

- $I_S = N_{CS} / N_{MS}$ = 390,625 / 25,000 = 15.63
- For a minimum of 25,000 maintenance slots, round I_S down to 15 and
- $N_{MS} = N_{CS} / I_S$ = 390,625 / 15 = 26,042 maintenance slots
- $I_T = I_S * P$ = 15 * 2.560 = 38.40 μs

The remaining free slots are left empty for insertion:

- $N_{ES} = N_{FS} – N_{MS}$ = 37,418 – 26,042 = 11,376 cell slots

If this number is not large enough for the expected number of inserted cells per second, we might reconsider the system design with the goal of reducing the number of maintenance slots needed.

B.3 Maintenance Time Slot Example 2

For another example, a configuration in which ZCLK is 100MHz, and the link rate is OC12/STM-4 (599.04 Mb/s net bit rate after the SONET overhead is removed). Therefore,

- R_L = 599.04 E + 6 / (53 * 8) = 1,412,830cells/sec
- f = 100MHz
- P = C / f = 64 / 25 E + 6 = 0.640 μs
- N_{CS} = f / C = 100 E + 6 / 64 = 1,562,500cells/s

The MC92520, when running at 100MHz, has 1,562,500 cell processing slots available per second. Of these, 1,412,830 are used for processing the cells received from the OC-12 link. Therefore, the number of free cell slots is:

- $N_{FS} = N_{CS} – R_L$ = 1,562,500 – 1,412,830 = 149,670 cell slots per second

At this point the user must calculate how many maintenance slots are needed for the maintenance tasks to be performed by the microprocessor (or DMA device). Note that each maintenance slot is the same length as a cell processing slot (P = 0.640 μs). Let us assume that at least 100,000 maintenance slots are needed per second. We must now calculate the maintenance slot Interval which is the fraction of the cell processing slots used as maintenance slots.

- $N_{MS} = N_{CS} / I_S$

so

- $I_S = N_{CS} / N_{MS}$ = 1,562,500 / 100,000 = 15.63
- For a minimum of 100,000 maintenance slots, round I_S down to 15 and
- $N_{MS} = N_{CS} / I_S$ = 1,562,500 / 15 = 104,167 maintenance slots
- $I_T = I_S * P$ = 15 * 0.640 = 9.60 μs

The remaining free slots are left empty for insertion:

- $N_{ES} = N_{FS} – N_{MS}$ = 149,670 – 104,167 = 45,503 cell slots

If this number is not large enough for the expected number of inserted cells per second, we might reconsider the system design with the goal of reducing the number of maintenance slots needed.

B.4 Maintenance Slot Parameters

Table B-1 presents I_T, N_{MS}, and N_{ES} as a function of the ZCLK frequency and I_S for an STS-3c physical link in order to assist the user in choosing the operating frequency and the value of the maintenance period length.

Table B-1. Maintenance Slot Parameters: 24–25 MHz

Freq		25 MHz P = 2.56 µs 37418 free slots			24 MHz P= 2.67 µs 21792 free slots		
MPL	I_S	I_T (µs)	N_{MS}	N_{ES}	I_T (µs)	N_{MS}	N_{ES}
10	11	28.2	35512	1906	—	—	—
11	12	30.7	32553	4865	—	—	—
12	13	33.3	30049	7369	—	—	—
13	14	35.8	27902	9516	—	—	—
14	15	38.4	26042	11376	—	—	—
15	16	41.0	24415	13003	—	—	—
16	17	43.5	22979	14439	—	—	—
17	18	46.1	21702	15716	—	—	—
18	19	48.6	20560	16858	50.7	19737	2055
19	20	51.2	19532	17886	53.3	18751	3041
24	25	64.0	15626	21792	66.7	15001	6791
29	30	76.8	13021	24397	80.0	12501	9291
34	35	89.6	11161	26257	93.3	10714	11078
39	40	102.4	9766	27652	106.7	9375	12417
44	45	115.2	8681	28737	120.0	8333	13459
49	50	128.0	7813	29605	133.3	7500	14292
54	55	140.8	7102	30316	146.7	6818	14974
59	60	153.6	6510	30908	160.0	6250	15542
63	64	166.4	6104	31314	173.3	5859	15933

Table B-2 presents I_T, N_{MS}, and N_{ES} as a function of the ZCLK frequency and I_S for an OC-12 physical link to assist the user in choosing the value of the maintenance period length at 100MHz operation.

Table B-2. Maintenance Slot Parameters: 100MHz

Freq		100 MHz P = 0.64 μs 149,670 free slots		
MPL	I_S	I_T (μs)	N_{MS}	N_{ES}
10	11	7.04	142045	7625
11	12	7.68	130208	19462
12	13	8.32	120192	29478
13	14	8.96	111607	38063
14	15	9.60	104167	45503
15	16	10.24	97656	52014
16	17	10.88	91912	57758
17	18	11.52	86806	62864
18	19	12.16	82237	67433
19	20	12.80	78125	71545
24	25	16.0	62500	87170
29	30	19.2	52083	97587
34	35	22.4	44643	105027
39	40	25.6	39063	110607
44	45	28.8	34722	114948
49	50	32.0	31250	118420
54	55	35.2	28409	121261
59	60	38.4	26042	123628
63	64	40.96	24414	125256

Appendix C
VC Bundling

This section describes the use of the MC92520 for VC bundling. This involves bundling several VCCs that are routed identically through a series of switches into a single VPC. Bundling VCCs in this manner reduces the processing complexity at the intermediate switches. The MC92520 fully supports VC bundling. However, the programming of the connections at the bundling point is not totally obvious and is described in some detail in this appendix.

C.1 VP-VC Boundary

This section describes the treatment of the boundaries between the virtual channel region (where the VCCs are treated individually) and the virtual path region (where the bundle of VCCs is treated as one VPC.) This configuration is illustrated in Figure C-1.

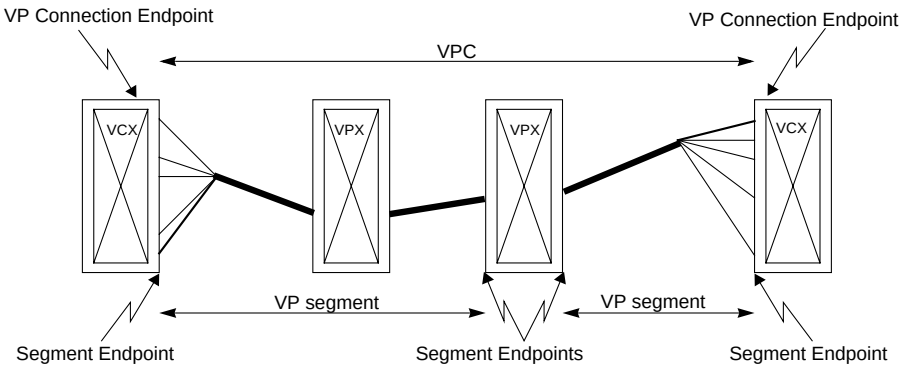


Figure C-1. Visibility of VCCs at the Endpoints of VPCs

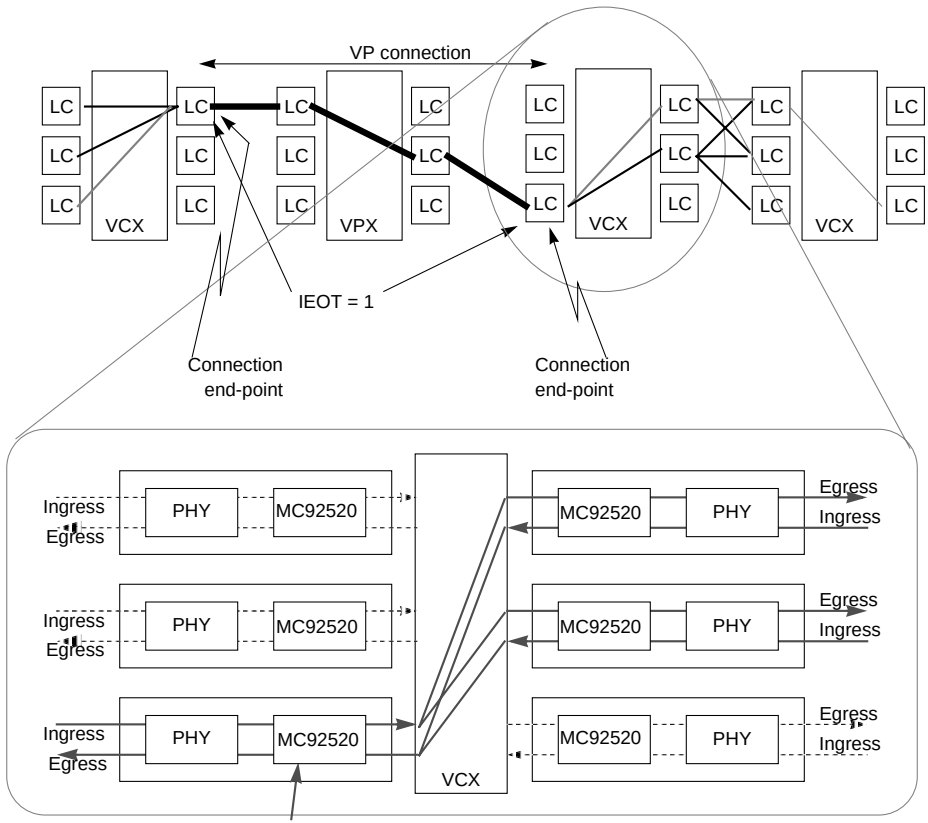
C.1.1 Bundling VCs into a VP

On the ingress side of the switch, all cells undergo a normal VC-level address-compression. Each VC connection is treated individually with its own entry in the external memory connection tables and a distinct CI pointing to its entry. The switch parameters must direct all the cells to the same line card after passing through the switch. On the egress side, the VCCs are still treated as separate connections, each with its own connection table entry. To

provide full address translation, the egress virtual path connection (EVPC) bit must be cleared for all VCCs. The VPI for all connections must be identical, and the VCIs must be distinct. The VP switches along the way do not change the VCI values, which remain constant until the opposite endpoint of the VPC.

C.1.2 Separating a VP into VCs

As shown in Figure C-2, the explosion of a VP into its component VCs must take place on the ingress side of the switch so that the virtual connections from which the virtual path is gathered can be routed to different VCC line cards. Each VCC has its own entry in the connection tables in the external memory and a distinct CI to point to its entry. Therefore, to produce a distinct CI for each VCC, the address compression must be done on the VPI/VCI.



VPI_VCI address compression on INGRESS for each link where ACM ≠ '01'

For F4-OAM traffic (i.e., VCI = 4 or VCI = 3):— The INGRESS is the connection/segment end-point, (i.e., IEOT=1 and possibly ISOT=1). – EVPC=1 (only VPI address translation on EGRESS)

For all other connections (user + F5-OAM): VPI_VCI address translation on EGRESS: EVPC=0.

Figure C-2. VP/VC Boundary Point

C.2 F4-Level OAM Processing at a VP/VC Boundary

As stated above, the VCCs are treated individually at the VP/VC boundary. However, the F4 flow of OAM cells refers to the VPC as a whole. We must treat these OAM cells as belonging to a VPC connection. For this purpose two additional entries are provided in the connection tables. The address compression on the F4-level OAM cells (VCI = 3 or 4) of the VPC must produce a CI that points to the special connection entry for the VCI value. The egress virtual path connection (EVPC) and ingress virtual path connection (IVPC) bits of these entries are set to indicate that VP treatment is necessary. The ingress end-to-end OAM termination (IEOT) bit of this entry should be set so that all F4-level OAM cells are removed from the cell flow.

C.3 Continuity Check

When performing an F4-level continuity check at the VP/VC boundary point, a user cell arriving on one of the VCCs causes the receive traffic bits of that VCC's entry in the flag table to be set. In order to determine if continuity was lost, the receive traffic bits from all of the VCCs should be collected by reading the entries of the flag table. Performing a logical OR on all of the bits consolidates them into a receive traffic indicator of the VPC as a whole.

C.3.1 OAM Block Test

This section describes how a block test at the VP/VC boundary point is performed. In order to perform an F4-level performance monitoring block test that originates or terminates at the VP/VC boundary point, a single entry in the OAM table must be used because the block test covers all of the cells belonging to the VPC. Because each VCC has its own entry in the connection tables, the OAM_ptr fields of all of the VCCs (as well as the special VPC entry—VCI = 3 for segment or VCI = 4 for end-to-end) must be identical and point to the common entry in the OAM table. The CI field of the OAM table entry points to the special VPC entry. At the originating end of the block test, the forward monitoring cell generation (FMCG) bit in the OAM table entry should be set to indicate that forward monitoring cells should be generated at this point. At the terminating end of the block test, the forward monitoring cell generation (FMCG) bit in the OAM table entry should be reset to prevent generation of FMCs.

C.4 F5-Level OAM Processing at a VP/VC Boundary

At a VP/VC boundary point, the F4-level OAM block test requires that, on the VP side of the switch, the OAM pointers of all of the VCC entries point to a common entry in the OAM table, as described above. In this situation it is not possible to run an F5-level block test on one of the VCCs while an F4-level block test is being run on the VPC as a whole because the F5-level block test would require a separate OAM table entry for the VCC. This is not a severe limitation because VCC visibility is not required on the VP side of the switch.

Instead, the F5-level OAM block test may be performed from the VC side of the switch. This situation is illustrated in Figure C-3.

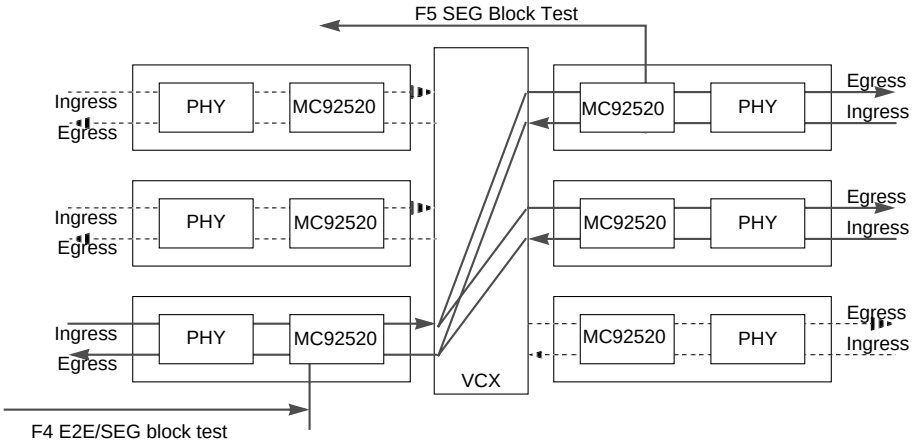


Figure C-3. F4- and F5-Level Block Tests at the Same Switch

C.5 Reserved VCI Values

At the VP/VC boundary, the MC92520 provides significant flexibility in copying or removing cells with reserved VCI values. Because the individual VCIs are visible at this point, the cells with each specific VCI value can be directed to a separate entry in the connection tables. In the connection entry, the ingress copy all (ICA) cells or ingress remove all (IRA) cells' bits may be set. This arrangement provides the ability to specify the treatment of the reserved VCI values of each VPC independently. If cells containing any one of a group of VCI values are to be treated in an identical manner, they may be routed to a single shared connection entry in which the ingress virtual path connection (IVPC) bit is set, thereby reducing the memory needed for this function.

NOTE:

Cells with reserved VCI values are not intended to be routed past the end of the VPC, so they should most likely be removed at this point.

Appendix D
MC92520 Applications

The following sections describe several applications that use the MC92520 in an ATM system.

D.1 Standard Architecture

Figure D-1 describes Motorola's proposed architecture for a line implementation.

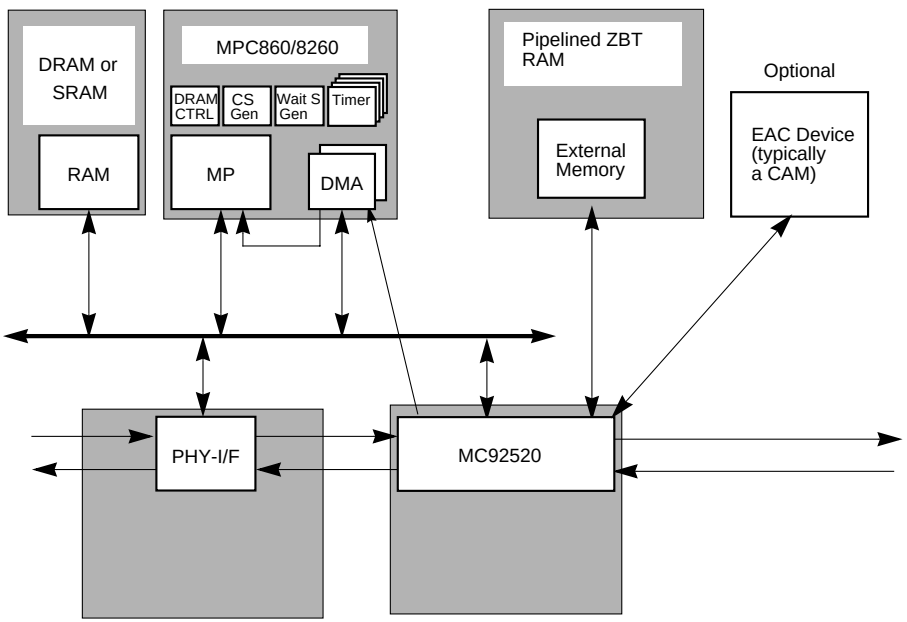


Figure D-1. Motorola's ATM Architecture

D.1.1 MC92520/MC92510

The MC92520 supports one or more PHY devices with an aggregate bandwidth of up to 662.5 Mbps. The MC92510 supports one or more PHY devices with an aggregate bandwidth of up to 331.75 Mbps.

D.1.2 PHY

For details, call your local Motorola representatives.

D.1.3 Line Card Microprocessor

The MPC8260 or the MPC860 may be used as the processor. The MPC860 contains a powerful CPU and many additional features such as:

- DRAM controller
- Chip select generator
- Wait state generator
- Timers
- Two external DMA channels (IDMA)
- Communication controller for many standards

Another recommended part is the MPC860SAR which is an MPC860 + SAR functionality and can be used for automating the SAR functionality. For more details see the *MPC860 User's Manual.*

NOTE:

Any microprocessor can be used as the processor. However, the processor interface of the MC92520 is optimized for use with the MPC860 or MPC8260.

D.1.4 External Memory

The external memory implementation must use pipelined ZBT RAMs for proper operation. The external memory array must be carefully designed, and the printed circuit board properly arranged, such that the stringent timing requirements of the external memory interface are met.

NOTE:

Refer to Section 4.4, "External Memory Interface," for additional information, as well as section Section 9.3.9, "External Memory Interface Timing."

D.1.5 External Address Compression CAM

The optional external address compression content-addressable memory (CAM) interface is designed to work gluelessly with the “Match Port” of Motorola MCM69C232 or MCM69C432 CAMs, or their equivalent. The CAM control port does not connect to the MC92520, rather it is controlled by the microprocessor.

D.1.6 Microprocessor RAM

This application can use any DRAM or SRAM that meets the application design requirements.

D.2 Multiple PHY Architecture

Figure D-2 shows a possible architecture where multiple (low-speed) PHY devices are connected to a single MC92520.

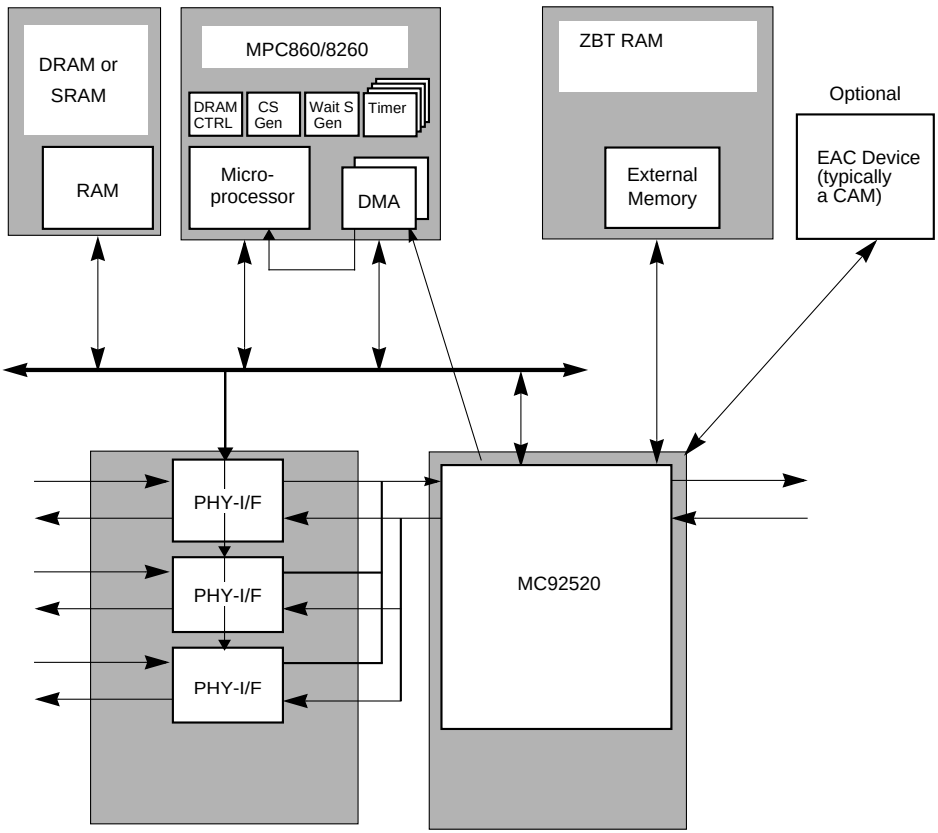


Figure D-2. Motorola‘s ATM Architecture for Multiple PHY Devices

D.3 DSLAM Access Network Architectures

Figure D-3 shows a possible access network architecture on which one MC92520 device interfaces a SONET PHY layer device on the network side and multiple PHY layer devices on the subscriber side. In this case, the ingress of the MC92520 corresponds to the upstream while the egress corresponds to the downstream. The switch interface of the MC92520 is interfaced to the queues.

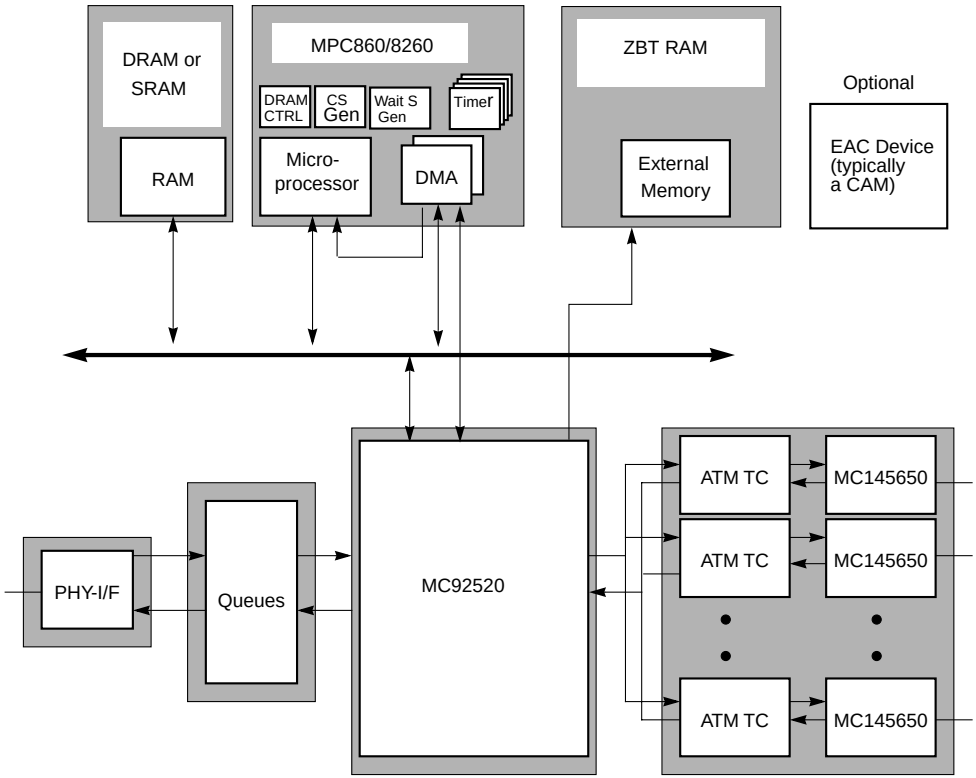


Figure D-3. Motorola's DSLAM Solution (Ingress Upstream/Egress Downstream)

Figure D-4 shows a possible access network architecture on which one MC92520 device interfaces a SONET PHY layer device on the network side and multiple PHY layer devices on the subscriber side, but in this case the ingress of the MC92520 corresponds to the downstream while the egress corresponds to the upstream. The switch interface of the MC92520 is again interfaced to the queues.

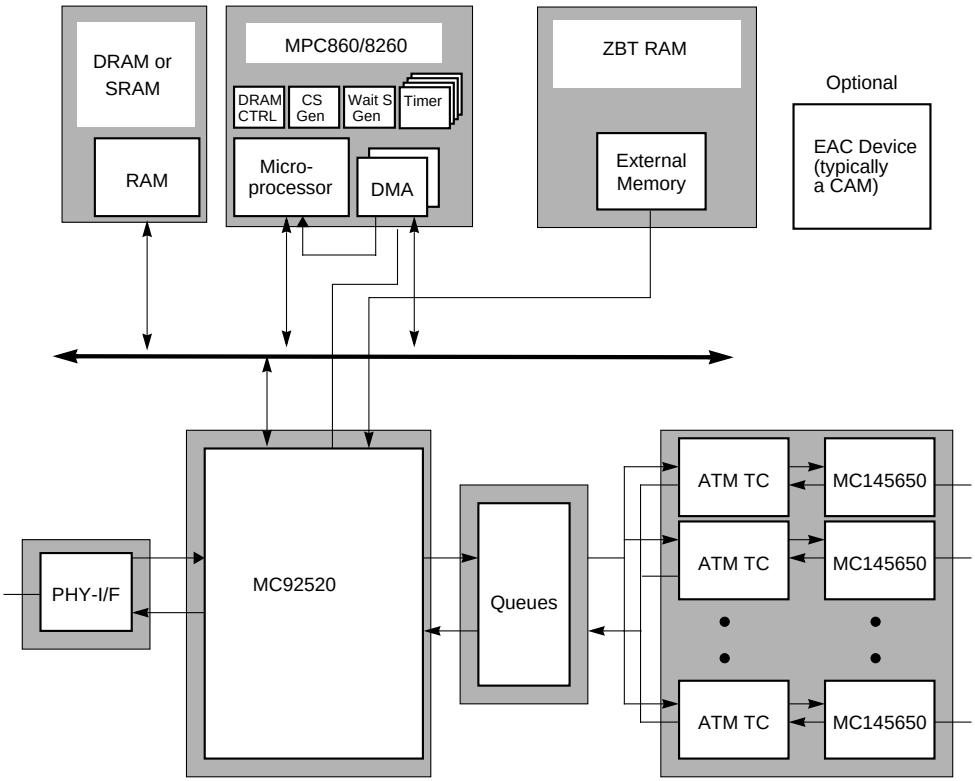


Figure D-4. Motorola's DSLAM Solution (Ingress Downstream/Egress Upstream)

Appendix E
BSDL Code

```
-- BSDL file for design atmc_top_palm

-- Created by jig version 0.1b

-- Creation Date: Wed Apr  5 09:45:59 2000

   entity atmc_top_palm is

   generic (PHYSICAL_PIN_MAP: string:= “P352PBGA”);

   port (
                  ZCLKI:  in      bit;
                ZCLKO_2:  out     bit;
                  ZCLKO:  out     bit;
                   ACLK:  in      bit;
                 ARST_B:  in      bit;
                PLLVDD1:  linkage bit;
                    TPA:  linkage bit;
                PLLGND1:  linkage bit;
                    TDO:  out     bit;
                    TMS:  in      bit;
                    TDI:  in      bit;
                 TRST_B:  in      bit;
                    TCK:  in      bit;
                TEST_SE:  linkage bit;
                  AMODE:  in      bit;
              STXAVALID:  in      bit;
              STXADDR_0:  in      bit;
              STXADDR_1:  in      bit;
```

```
 STXADDR_2:  in      bit;
 STXADDR_3:  in      bit;
 STXADDR_4:  in      bit;
    STXCLK:  in      bit;
 STXDATA_0:  in      bit;
 COREGND12:  linkage bit;
 COREVDD12:  linkage bit;
 STXDATA_1:  in      bit;
 STXDATA_2:  in      bit;
 STXDATA_3:  in      bit;
 STXDATA_4:  in      bit;
 STXDATA_5:  in      bit;
 STXDATA_6:  in      bit;
 STXDATA_7:  in      bit;
 STXDATA_8:  in      bit;
 STXDATA_9:  in      bit;
STXDATA_10:  in      bit;
STXDATA_11:  in      bit;
STXDATA_12:  in      bit;
STXDATA_13:  in      bit;
STXDATA_14:  in      bit;
STXDATA_15:  in      bit;
  STXENB_B:  inout   bit;
   STXPRTY:  inout   bit;
    STXSOC:  inout   bit;
   STXCLAV:  inout   bit;
   SRXCLAV:  inout   bit;
    SRXSOC:  inout   bit;
   SRXPRTY:  inout   bit;
 SRXDATA_0:  inout   bit;
 SRXDATA_1:  inout   bit;
 COREGND11:  linkage bit;
 COREVDD11:  linkage bit;
 SRXDATA_2:  inout   bit;
 SRXDATA_3:  inout   bit;
 SRXDATA_4:  inout   bit;
```

```
 SRXDATA_5:  inout   bit;
 SRXDATA_6:  inout   bit;
 SRXDATA_7:  inout   bit;
 SRXDATA_8:  inout   bit;
 SRXDATA_9:  inout   bit;
SRXDATA_10:  inout   bit;
SRXDATA_11:  inout   bit;
SRXDATA_12:  inout   bit;
SRXDATA_13:  inout   bit;
SRXDATA_14:  inout   bit;
SRXDATA_15:  inout   bit;
  SRXENB_B:  in      bit;
    SRXCLK:  in      bit;
     MADD2:  in      bit;
     MADD3:  in      bit;
     MADD4:  in      bit;
     MADD5:  in      bit;
     MADD6:  in      bit;
     MADD7:  in      bit;
     MADD8:  in      bit;
     MADD9:  in      bit;
    MADD10:  in      bit;
    MADD11:  in      bit;
    MADD12:  in      bit;
    MADD13:  in      bit;
    MADD14:  in      bit;
    MADD15:  in      bit;
    MADD16:  in      bit;
    MADD17:  in      bit;
    MADD18:  in      bit;
    MADD19:  in      bit;
    MADD20:  in      bit;
    MADD21:  in      bit;
    MADD22:  in      bit;
    MADD23:  in      bit;
 COREGND10:  linkage bit;
```

```
COREVDD10:  linkage bit;
   MADD24:  in      bit;
   MADD25:  in      bit;
   MWSL_B:  in      bit;
   MWSH_B:  in      bit;
MENDCYC_B:  in      bit;
   MSEL_B:  in      bit;
    MWR_B:  in      bit;
     MCLK:  in      bit;
  MREQ0_B:  out     bit;
  MREQ1_B:  out     bit;
   MINT_B:  out     bit;
MDTACK0_B:  out     bit;
MDTACK1_B:  out     bit;
 COREGND9:  linkage bit;
 COREVDD9:  linkage bit;
  MDATA_0:  inout   bit;
  MDATA_1:  inout   bit;
  MDATA_2:  inout   bit;
  MDATA_3:  inout   bit;
  MDATA_4:  inout   bit;
  MDATA_5:  inout   bit;
  MDATA_6:  inout   bit;
  MDATA_7:  inout   bit;
  MDATA_8:  inout   bit;
  MDATA_9:  inout   bit;
 MDATA_10:  inout   bit;
 MDATA_11:  inout   bit;
 MDATA_12:  inout   bit;
 COREGND8:  linkage bit;
 COREVDD8:  linkage bit;
 MDATA_13:  inout   bit;
 MDATA_14:  inout   bit;
 MDATA_15:  inout   bit;
 MDATA_16:  inout   bit;
 MDATA_17:  inout   bit;
```

```
MDATA_18:  inout   bit;
MDATA_19:  inout   bit;
MDATA_20:  inout   bit;
MDATA_21:  inout   bit;
MDATA_22:  inout   bit;
MDATA_23:  inout   bit;
MDATA_24:  inout   bit;
MDATA_25:  inout   bit;
MDATA_26:  inout   bit;
COREGND7:  linkage bit;
COREVDD7:  linkage bit;
MDATA_27:  inout   bit;
MDATA_28:  inout   bit;
MDATA_29:  inout   bit;
MDATA_30:  inout   bit;
MDATA_31:  inout   bit;
TXFULL_B:  in      bit;
 TXENB_B:  out     bit;
TXADDR_4:  out     bit;
TXADDR_0:  out     bit;
TXADDR_1:  out     bit;
TXADDR_2:  out     bit;
TXADDR_3:  out     bit;
    TXSOC: out     bit;
  TXPRTY:  out     bit;
TXDATA_0:  out     bit;
TXDATA_1:  out     bit;
TXDATA_2:  out     bit;
TXDATA_3:  out     bit;
TXDATA_4:  out     bit;
TXDATA_5:  out     bit;
TXDATA_6:  out     bit;
TXDATA_7:  out     bit;
TXDATA_8:  out     bit;
TXDATA_9:  out     bit;
TXDATA_10: out     bit;
```

```
TXDATA_11:  out     bit;
TXDATA_12:  out     bit;
TXDATA_13:  out     bit;
TXDATA_14:  out     bit;
TXDATA_15:  out     bit;
RXEMPTY_B:  in      bit;
    RXSOC:  in      bit;
   RXPRTY:  in      bit;
 RXDATA_0:  in      bit;
 RXDATA_1:  in      bit;
 RXDATA_2:  in      bit;
 RXDATA_3:  in      bit;
 RXDATA_4:  in      bit;
 RXDATA_5:  in      bit;
 COREVDD6:  linkage bit;
 COREGND6:  linkage bit;
 RXDATA_6:  in      bit;
 RXDATA_7:  in      bit;
 RXDATA_8:  in      bit;
 RXDATA_9:  in      bit;
RXDATA_10:  in      bit;
RXDATA_11:  in      bit;
RXDATA_12:  in      bit;
RXDATA_13:  in      bit;
RXDATA_14:  in      bit;
RXDATA_15:  in      bit;
  RXENB_B:  out     bit;
 RXADDR_0:  out     bit;
 RXADDR_1:  out     bit;
 RXADDR_2:  out     bit;
 RXADDR_3:  out     bit;
 RXADDR_4:  out     bit;
  EACOE_B:  out     bit;
  EACSM_B:  out     bit;
   EACCLK:  out     bit;
EACDATA_0:  inout   bit;
```

```
 EACDATA_1:  inout   bit;
 EACDATA_2:  inout   bit;
 EACDATA_3:  inout   bit;
  COREVDD5:  linkage bit;
  COREGND5:  linkage bit;
 EACDATA_4:  inout   bit;
 EACDATA_5:  inout   bit;
 EACDATA_6:  inout   bit;
 EACDATA_7:  inout   bit;
 EACDATA_8:  inout   bit;
 EACDATA_9:  inout   bit;
EACDATA_10:  inout   bit;
EACDATA_11:  inout   bit;
EACDATA_12:  inout   bit;
EACDATA_13:  inout   bit;
EACDATA_14:  inout   bit;
EACDATA_15:  inout   bit;
EACDATA_16:  inout   bit;
EACDATA_17:  inout   bit;
EACDATA_18:  inout   bit;
EACDATA_19:  inout   bit;
EACDATA_20:  inout   bit;
EACDATA_21:  inout   bit;
EACDATA_22:  inout   bit;
EACDATA_23:  inout   bit;
EACDATA_24:  inout   bit;
EACDATA_25:  inout   bit;
EACDATA_26:  inout   bit;
EACDATA_27:  inout   bit;
EACDATA_28:  inout   bit;
EACDATA_29:  inout   bit;
EACDATA_30:  inout   bit;
EACDATA_31:  inout   bit;
    EMWR_B:  out     bit;
   EMWSL_B:  out     bit;
   EMWSH_B:  out     bit;
```

```
    EMADD_2:  inout   bit;
    EMADD_3:  inout   bit;
    EMADD_4:  inout   bit;
    EMADD_5:  inout   bit;
    EMADD_6:  inout   bit;
    EMADD_7:  inout   bit;
    EMADD_8:  inout   bit;
    EMADD_9:  inout   bit;
   EMADD_10:  inout   bit;
   COREVDD4:  linkage bit;
   EMADD_11:  inout   bit;
   COREGND4:  linkage bit;
   EMADD_12:  inout   bit;
   EMADD_13:  inout   bit;
   EMADD_14:  inout   bit;
   EMADD_15:  inout   bit;
   EMADD_16:  inout   bit;
   EMADD_17:  inout   bit;
   EMADD_18:  inout   bit;
 EMADD_B_19:  out     bit;
   EMADD_19:  inout   bit;
   COREVDD3:  linkage bit;
 EMADD_B_20:  out     bit;
   EMADD_20:  inout   bit;
 EMADD_B_21:  out     bit;
   EMADD_21:  inout   bit;
 EMADD_B_22:  out     bit;
   EMADD_22:  inout   bit;
   COREGND3:  linkage bit;
 EMADD_B_23:  out     bit;
   EMADD_23:  inout   bit;
     EMOE_B:  out     bit;
   EMDATA_0:  inout   bit;
   EMDATA_1:  inout   bit;
   EMDATA_2:  inout   bit;
   EMDATA_3:  inout   bit;
```

```
 COREVDD2:  linkage bit;
 EMDATA_4:  inout   bit;
 COREGND2:  linkage bit;
 EMDATA_5:  inout   bit;
 EMDATA_6:  inout   bit;
 EMDATA_7:  inout   bit;
 EMDATA_8:  inout   bit;
 EMDATA_9:  inout   bit;
EMDATA_10:  inout   bit;
EMDATA_11:  inout   bit;
EMDATA_12:  inout   bit;
EMDATA_13:  inout   bit;
EMDATA_14:  inout   bit;
EMDATA_15:  inout   bit;
EMDATA_16:  inout   bit;
EMDATA_17:  inout   bit;
EMDATA_18:  inout   bit;
 COREVDD1:  linkage bit;
EMDATA_19:  inout   bit;
 COREGND1:  linkage bit;
EMDATA_20:  inout   bit;
EMDATA_21:  inout   bit;
EMDATA_22:  inout   bit;
EMDATA_23:  inout   bit;
EMDATA_24:  inout   bit;
EMDATA_25:  inout   bit;
EMDATA_26:  inout   bit;
EMDATA_27:  inout   bit;
EMDATA_28:  inout   bit;
EMDATA_29:  inout   bit;
EMDATA_30:  inout   bit;
EMDATA_31:  inout   bit;
   GND_67:  linkage bit;
   GND_66:  linkage bit;
   GND_65:  linkage bit;
   GND_64:  linkage bit;
```

```
GND_63:  linkage bit;
GND_62:  linkage bit;
GND_61:  linkage bit;
GND_60:  linkage bit;
GND_59:  linkage bit;
GND_58:  linkage bit;
GND_57:  linkage bit;
GND_56:  linkage bit;
GND_55:  linkage bit;
GND_54:  linkage bit;
GND_53:  linkage bit;
GND_52:  linkage bit;
GND_51:  linkage bit;
GND_50:  linkage bit;
GND_49:  linkage bit;
GND_48:  linkage bit;
GND_47:  linkage bit;
GND_46:  linkage bit;
GND_45:  linkage bit;
GND_44:  linkage bit;
GND_43:  linkage bit;
GND_42:  linkage bit;
GND_41:  linkage bit;
GND_40:  linkage bit;
GND_39:  linkage bit;
GND_38:  linkage bit;
GND_37:  linkage bit;
GND_36:  linkage bit;
GND_35:  linkage bit;
GND_34:  linkage bit;
GND_33:  linkage bit;
GND_32:  linkage bit;
GND_31:  linkage bit;
GND_30:  linkage bit;
GND_29:  linkage bit;
GND_28:  linkage bit;
```

```
GND_27:  linkage bit;
GND_26:  linkage bit;
GND_25:  linkage bit;
GND_24:  linkage bit;
GND_23:  linkage bit;
GND_22:  linkage bit;
GND_21:  linkage bit;
GND_20:  linkage bit;
GND_19:  linkage bit;
GND_18:  linkage bit;
GND_17:  linkage bit;
GND_16:  linkage bit;
GND_15:  linkage bit;
GND_14:  linkage bit;
GND_13:  linkage bit;
GND_12:  linkage bit;
GND_11:  linkage bit;
GND_10:  linkage bit;
GND_09:  linkage bit;
GND_08:  linkage bit;
GND_07:  linkage bit;
GND_06:  linkage bit;
GND_05:  linkage bit;
GND_04:  linkage bit;
GND_03:  linkage bit;
GND_02:  linkage bit;
GND_01:  linkage bit;
GND_00:  linkage bit;
VDD_15:  linkage bit;
VDD_14:  linkage bit;
VDD_13:  linkage bit;
VDD_12:  linkage bit;
VDD_11:  linkage bit;
VDD_10:  linkage bit;
VDD_09:  linkage bit;
VDD_08:  linkage bit;
```

```
                VDD_07:  linkage bit;
                VDD_06:  linkage bit;
                VDD_05:  linkage bit;
                VDD_04:  linkage bit;
                VDD_03:  linkage bit;
                VDD_02:  linkage bit;
                VDD_01:  linkage bit;
                VDD_00:  linkage bit );

use STD_1149_1_1994.all;

attribute COMPONENT_CONFORMANCE of atmc_top_palm: entity is “STD_1149_1_1993”;

attribute PIN_MAP of atmc_top_palm: entity is PHYSICAL_PIN_MAP;

constant P352PBGA: PIN_MAP_STRING :=
              “ZCLKI:    A3, “ &
            “ZCLKO_2:    C4, “ &
              “ZCLKO:    B3, “ &
               “ACLK:    C5, “ &
             “ARST_B:    A4, “ &
            “PLLVDD1:    D5, “ &
                “TPA:    B4, “ &
            “PLLGND1:    C6, “ &
                “TDO:    A5, “ &
                “TMS:    B5, “ &
                “TDI:    C7, “ &
              “TRST_B:    A6, “ &
                “TCK:    D7, “ &
            “TEST_SE:    B6, “ &
               “AMODE:    C8, “ &
           “STXAVALID:    A7, “ &
           “STXADDR_0:    D8, “ &
           “STXADDR_1:    B7, “ &
           “STXADDR_2:    C9, “ &
           “STXADDR_3:    A8, “ &
```

```
"STXADDR_4:    B8, " &
   "STXCLK:    A9, " &
"STXDATA_0:   C10, " &
"COREGND12:    B9, " &
"COREVDD12:   D10, " &
"STXDATA_1:   A10, " &
"STXDATA_2:   C11, " &
"STXDATA_3:   B10, " &
"STXDATA_4:   D12, " &
"STXDATA_5:   A11, " &
"STXDATA_6:   C12, " &
"STXDATA_7:   B11, " &
"STXDATA_8:   A12, " &
"STXDATA_9:   C13, " &
"STXDATA_10:  B12, " &
"STXDATA_11:  C14, " &
"STXDATA_12:  A13, " &
"STXDATA_13:  D13, " &
"STXDATA_14:  B13, " &
"STXDATA_15:  C15, " &
  "STXENB_B:  A14, " &
   "STXPRTY:  D15, " &
    "STXSOC:  B14, " &
   "STXCLAV:  C16, " &
   "SRXCLAV:  A15, " &
    "SRXSOC:  B15, " &
   "SRXPRTY:  A16, " &
"SRXDATA_0:   C17, " &
"SRXDATA_1:   B16, " &
"COREGND11:   D17, " &
"COREVDD11:   A17, " &
"SRXDATA_2:   C18, " &
"SRXDATA_3:   B17, " &
"SRXDATA_4:   A18, " &
"SRXDATA_5:   B18, " &
"SRXDATA_6:   C19, " &
```

```
“SRXDATA_7:   A19, “ &
“SRXDATA_8:   D18, “ &
“SRXDATA_9:   B19, “ &
“SRXDATA_10:  C20, “ &
“SRXDATA_11:  A20, “ &
“SRXDATA_12:  B20, “ &
“SRXDATA_13:  A21, “ &
“SRXDATA_14:  C21, “ &
“SRXDATA_15:  D20, “ &
“SRXENB_B:    B21, “ &
“SRXCLK:      A22, “ &
“MADD2:       C22, “ &
“MADD3:       D22, “ &
“MADD4:       B22, “ &
“MADD5:       A23, “ &
“MADD6:       C23, “ &
“MADD7:       B23, “ &
“MADD8:       A24, “ &
“MADD9:       B24, “ &
“MADD10:      A25, “ &
“MADD11:      C26, “ &
“MADD12:      D24, “ &
“MADD13:      C25, “ &
“MADD14:      E24, “ &
“MADD15:      D26, “ &
“MADD16:      E23, “ &
“MADD17:      D25, “ &
“MADD18:      F24, “ &
“MADD19:      E26, “ &
“MADD20:      E25, “ &
“MADD21:      G24, “ &
“MADD22:      F26, “ &
“MADD23:      G23, “ &
“COREGND10:   F25, “ &
“COREVDD10:   H24, “ &
“MADD24:      G26, “ &
```

```
                “MADD25:   H23, “ &
                “MWSL_B:   G25, “ &
                “MWSH_B:   J24, “ &
              “MENDCYC_B:  H26, “ &
                “MSEL_B:   H25, “ &
                 “MWR_B:   J26, “ &
                  “MCLK:   K24, “ &
               “MREQ0_B:   J25, “ &
               “MREQ1_B:   K23, “ &
                “MINT_B:   K26, “ &
             “MDTACK0_B:   L24, “ &
             “MDTACK1_B:   K25, “ &
              “COREGND9:   M23, “ &
              “COREVDD9:   L26, “ &
               “MDATA_0:   M24, “ &
               “MDATA_1:   L25, “ &
               “MDATA_2:   M26, “ &
               “MDATA_3:   N24, “ &
               “MDATA_4:   M25, “ &
               “MDATA_5:   P24, “ &
               “MDATA_6:   N26, “ &
               “MDATA_7:   N23, “ &
               “MDATA_8:   N25, “ &
               “MDATA_9:   R24, “ &
              “MDATA_10:   P26, “ &
              “MDATA_11:   R23, “ &
              “MDATA_12:   P25, “ &
              “COREGND8:   T24, “ &
              “COREVDD8:   R26, “ &
              “MDATA_13:   R25, “ &
              “MDATA_14:   T26, “ &
              “MDATA_15:   U24, “ &
              “MDATA_16:   T25, “ &
              “MDATA_17:   U23, “ &
              “MDATA_18:   U26, “ &
              “MDATA_19:   V24, “ &
```

```
“MDATA_20:   U25, “ &
“MDATA_21:   V26, “ &
“MDATA_22:   V25, “ &
“MDATA_23:   W24, “ &
“MDATA_24:   W26, “ &
“MDATA_25:   V23, “ &
“MDATA_26:   W25, “ &
“COREGND7:   Y24, “ &
“COREVDD7:   Y26, “ &
“MDATA_27:   Y25, “ &
“MDATA_28:  AA26, “ &
“MDATA_29:  AA24, “ &
“MDATA_30:   Y23, “ &
“MDATA_31:  AA25, “ &
“TXFULL_B:  AB24, “ &
 “TXENB_B:  AB23, “ &
“TXADDR_4:  AB25, “ &
“TXADDR_0:  AC26, “ &
“TXADDR_1:  AC24, “ &
“TXADDR_2:  AC25, “ &
“TXADDR_3:  AD26, “ &
   “TXSOC:  AD25, “ &
  “TXPRTY:  AE26, “ &
“TXDATA_0:  AF24, “ &
“TXDATA_1:  AD23, “ &
“TXDATA_2:  AE24, “ &
“TXDATA_3:  AD22, “ &
“TXDATA_4:  AF23, “ &
“TXDATA_5:  AC22, “ &
“TXDATA_6:  AE23, “ &
“TXDATA_7:  AD21, “ &
“TXDATA_8:  AF22, “ &
“TXDATA_9:  AE22, “ &
“TXDATA_10: AD20, “ &
“TXDATA_11: AF21, “ &
“TXDATA_12: AC20, “ &
```

```
“TXDATA_13:  AE21, “ &
“TXDATA_14:  AD19, “ &
“TXDATA_15:  AF20, “ &
“RXEMPTY_B:  AC19, “ &
    “RXSOC:  AE20, “ &
   “RXPRTY:  AD18, “ &
 “RXDATA_0:  AF19, “ &
 “RXDATA_1:  AE19, “ &
 “RXDATA_2:  AF18, “ &
 “RXDATA_3:  AD17, “ &
 “RXDATA_4:  AE18, “ &
 “RXDATA_5:  AC17, “ &
 “COREVDD6:  AF17, “ &
 “COREGND6:  AD16, “ &
 “RXDATA_6:  AE17, “ &
 “RXDATA_7:  AC15, “ &
 “RXDATA_8:  AF16, “ &
 “RXDATA_9:  AD15, “ &
“RXDATA_10:  AE16, “ &
“RXDATA_11:  AF15, “ &
“RXDATA_12:  AD14, “ &
“RXDATA_13:  AE15, “ &
“RXDATA_14:  AD13, “ &
“RXDATA_15:  AF14, “ &
  “RXENB_B:  AC14, “ &
 “RXADDR_0:  AE14, “ &
 “RXADDR_1:  AD12, “ &
 “RXADDR_2:  AF13, “ &
 “RXADDR_3:  AC12, “ &
 “RXADDR_4:  AE13, “ &
  “EACOE_B:  AD11, “ &
  “EACSM_B:  AF12, “ &
   “EACCLK:  AE12, “ &
“EACDATA_0:  AF11, “ &
“EACDATA_1:  AD10, “ &
“EACDATA_2:  AE11, “ &
```

```
“EACDATA_3:  AC10, “ &
 “COREVDD5:  AF10, “ &
 “COREGND5:   AD9, “ &
“EACDATA_4:  AE10, “ &
“EACDATA_5:   AF9, “ &
“EACDATA_6:   AE9, “ &
“EACDATA_7:   AD8, “ &
“EACDATA_8:   AF8, “ &
“EACDATA_9:   AC9, “ &
“EACDATA_10:  AE8, “ &
“EACDATA_11:  AD7, “ &
“EACDATA_12:  AF7, “ &
“EACDATA_13:  AE7, “ &
“EACDATA_14:  AF6, “ &
“EACDATA_15:  AD6, “ &
“EACDATA_16:  AC7, “ &
“EACDATA_17:  AE6, “ &
“EACDATA_18:  AF5, “ &
“EACDATA_19:  AD5, “ &
“EACDATA_20:  AC5, “ &
“EACDATA_21:  AE5, “ &
“EACDATA_22:  AF4, “ &
“EACDATA_23:  AD4, “ &
“EACDATA_24:  AE4, “ &
“EACDATA_25:  AF3, “ &
“EACDATA_26:  AE3, “ &
“EACDATA_27:  AF2, “ &
“EACDATA_28:  AD1, “ &
“EACDATA_29:  AC3, “ &
“EACDATA_30:  AD2, “ &
“EACDATA_31:  AB3, “ &
    “EMWR_B:  AB4, “ &
   “EMWSL_B:  AC2, “ &
   “EMWSH_B:  AA3, “ &
   “EMADD_2:  AB1, “ &
   “EMADD_3:  AB2, “ &
```

```
    “EMADD_4:    Y3, “ &
    “EMADD_5:   AA1, “ &
    “EMADD_6:    Y4, “ &
    “EMADD_7:   AA2, “ &
    “EMADD_8:    W3, “ &
    “EMADD_9:    Y1, “ &
   “EMADD_10:    W4, “ &
   “COREVDD4:    Y2, “ &
   “EMADD_11:    V3, “ &
   “COREGND4:    W1, “ &
   “EMADD_12:    W2, “ &
   “EMADD_13:    V1, “ &
   “EMADD_14:    U3, “ &
   “EMADD_15:    V2, “ &
   “EMADD_16:    U4, “ &
   “EMADD_17:    U1, “ &
   “EMADD_18:    T3, “ &
 “EMADD_B_19:    U2, “ &
   “EMADD_19:    R4, “ &
   “COREVDD3:    T1, “ &
 “EMADD_B_20:    R3, “ &
   “EMADD_20:    T2, “ &
 “EMADD_B_21:    R1, “ &
   “EMADD_21:    P3, “ &
 “EMADD_B_22:    R2, “ &
   “EMADD_22:    N3, “ &
   “COREGND3:    P1, “ &
 “EMADD_B_23:    P4, “ &
   “EMADD_23:    P2, “ &
     “EMOE_B:    M3, “ &
   “EMDATA_0:    N1, “ &
   “EMDATA_1:    M4, “ &
   “EMDATA_2:    N2, “ &
   “EMDATA_3:    L3, “ &
   “COREVDD2:    M1, “ &
   “EMDATA_4:    M2, “ &
```

```
 “COREGND2:    L1, “ &
 “EMDATA_5:    K3, “ &
 “EMDATA_6:    L2, “ &
 “EMDATA_7:    K4, “ &
 “EMDATA_8:    K1, “ &
 “EMDATA_9:    J3, “ &
“EMDATA_10:    K2, “ &
“EMDATA_11:    J1, “ &
“EMDATA_12:    J2, “ &
“EMDATA_13:    H3, “ &
“EMDATA_14:    H1, “ &
“EMDATA_15:    J4, “ &
“EMDATA_16:    H2, “ &
“EMDATA_17:    G3, “ &
“EMDATA_18:    G1, “ &
 “COREVDD1:    G2, “ &
“EMDATA_19:    F1, “ &
 “COREGND1:    F3, “ &
“EMDATA_20:    G4, “ &
“EMDATA_21:    F2, “ &
“EMDATA_22:    E1, “ &
“EMDATA_23:    E3, “ &
“EMDATA_24:    E4, “ &
“EMDATA_25:    E2, “ &
“EMDATA_26:    D1, “ &
“EMDATA_27:    D3, “ &
“EMDATA_28:    D2, “ &
“EMDATA_29:    C1, “ &
“EMDATA_30:    C2, “ &
“EMDATA_31:    B1, “ &
   “GND_67:  AF26, “ &
   “GND_66:  AF25, “ &
   “GND_65:   AF1, “ &
   “GND_64:  AE25, “ &
   “GND_63:   AE2, “ &
   “GND_62:   AE1, “ &
```

```
“GND_61:  AD24, “ &
“GND_60:   AD3, “ &
“GND_59:  AC23, “ &
“GND_58:  AC18, “ &
“GND_57:  AC13, “ &
“GND_56:   AC8, “ &
“GND_55:   AC4, “ &
“GND_54:   W23, “ &
“GND_53:    V4, “ &
“GND_52:   T16, “ &
“GND_51:   T15, “ &
“GND_50:   T14, “ &
“GND_49:   T13, “ &
“GND_48:   T12, “ &
“GND_47:   T11, “ &
“GND_46:   R16, “ &
“GND_45:   R15, “ &
“GND_44:   R14, “ &
“GND_43:   R13, “ &
“GND_42:   R12, “ &
“GND_41:   R11, “ &
“GND_40:   P23, “ &
“GND_39:   P16, “ &
“GND_38:   P15, “ &
“GND_37:   P14, “ &
“GND_36:   P13, “ &
“GND_35:   P12, “ &
“GND_34:   P11, “ &
“GND_33:   N16, “ &
“GND_32:   N15, “ &
“GND_31:   N14, “ &
“GND_30:   N13, “ &
“GND_29:   N12, “ &
“GND_28:   N11, “ &
“GND_27:    N4, “ &
“GND_26:   M16, “ &
```

```
“GND_25:   M15, “ &
“GND_24:   M14, “ &
“GND_23:   M13, “ &
“GND_22:   M12, “ &
“GND_21:   M11, “ &
“GND_20:   L16, “ &
“GND_19:   L15, “ &
“GND_18:   L14, “ &
“GND_17:   L13, “ &
“GND_16:   L12, “ &
“GND_15:   L11, “ &
“GND_14:   J23, “ &
“GND_13:    H4, “ &
“GND_12:   D23, “ &
“GND_11:   D19, “ &
“GND_10:   D14, “ &
“GND_09:    D9, “ &
“GND_08:    D4, “ &
“GND_07:   C24, “ &
“GND_06:    C3, “ &
“GND_05:   B26, “ &
“GND_04:   B25, “ &
“GND_03:    B2, “ &
“GND_02:   A26, “ &
“GND_01:    A2, “ &
“GND_00:    A1, “ &
“VDD_15:  AC21, “ &
“VDD_14:  AC16, “ &
“VDD_13:  AC11, “ &
“VDD_12:   AC6, “ &
“VDD_11:  AA23, “ &
“VDD_10:   AA4, “ &
“VDD_09:   T23, “ &
“VDD_08:    T4, “ &
“VDD_07:   L23, “ &
“VDD_06:    L4, “ &
```

```
                “VDD_05:   F23, “ &
                “VDD_04:    F4, “ &
                “VDD_03:   D21, “ &
                “VDD_02:   D16, “ &
                “VDD_01:   D11, “ &
                “VDD_00:    D6 “ ;

attribute TAP_SCAN_CLOCK of  A_TCK : signal is (1.00e+07, BOTH);
attribute TAP_SCAN_IN    of  A_TDI : signal is true;
attribute TAP_SCAN_MODE  of  A_TMS : signal is true;
attribute TAP_SCAN_OUT   of  A_TDO : signal is true;
attribute TAP_SCAN_RESET of A_TRST_B : signal is true;

attribute INSTRUCTION_LENGTH of atmc_top_palm : entity is 4;

attribute INSTRUCTION_OPCODE of atmc_top_palm : entity is
  “PRIVATE_5  (0111),” &
  “PRIVATE_4  (1010),” &
  “PRIVATE_3  (1011),” &
  “PRIVATE_2  (1101),” &
  “PRIVATE_1  (1110),” &
  “PRIVATE_0  (1000),” &
  “EXTEST     (0000),” &
  “IDCODE     (0001),” &
  “SAMPLE     (0010),” &
  “HIGHZ      (1001),” &
  “BYPASS     (1111),” &
  “CLAMP      (1100)” ;

attribute INSTRUCTION_CAPTURE of atmc_top_palm : entity is “0001”;

attribute INSTRUCTION_PRIVATE of atmc_top_palm : entity is
  “PRIVATE_5 ,” &
  “PRIVATE_4 ,” &
  “PRIVATE_3 ,” &
  “PRIVATE_2 ,” &
```

```
  “PRIVATE_1 ,” &
  “PRIVATE_0 “ ;

attribute IDCODE_REGISTER of atmc_top_palm : entity is
  “0000” &              -- version number
  “00101000” &          -- part number, part 1
  “00001010” &          -- part number, part 2
  “00000001110” &       -- manufacturer
  “1”;                  -- required by standard

attribute BOUNDARY_LENGTH of atmc_top_palm : entity is 302;

attribute BOUNDARY_REGISTER of atmc_top_palm : entity is

-- num cell  port           function   safe [cccel disval rslt]
“ 301 (BC_4,ZCLKI          , CLOCK       , X),” &
“ 300 (BC_1,*              , CONTROL     , 1),” &
“ 299 (BC_1,ZCLKO_2        , OUTPUT3     , 1,  300,    1,    Z),” &
“ 298 (BC_1,*              , CONTROL     , 1),” &
“ 297 (BC_1,ZCLKO          , OUTPUT3     , 1,  298,    1,    Z),” &
“ 296 (BC_4,ACLK           , CLOCK       , X),” &
“ 295 (BC_4,ARST_B         , OBSERVE_ONLY, X),” &
“ 294 (BC_4,AMODE          , OBSERVE_ONLY, X),” &
“ 293 (BC_4,STXAVALID      , OBSERVE_ONLY, X),” &
“ 292 (BC_4,STXADDR_0      , OBSERVE_ONLY, X),” &
“ 291 (BC_4,STXADDR_1      , OBSERVE_ONLY, X),” &
“ 290 (BC_4,STXADDR_2      , OBSERVE_ONLY, X),” &
“ 289 (BC_4,STXADDR_3      , OBSERVE_ONLY, X),” &
“ 288 (BC_4,STXADDR_4      , OBSERVE_ONLY, X),” &
“ 287 (BC_4,STXCLK         , CLOCK       , X),” &
“ 286 (BC_4,STXDATA_0      , OBSERVE_ONLY, X),” &
“ 285 (BC_4,STXDATA_1      , OBSERVE_ONLY, X),” &
“ 284 (BC_4,STXDATA_2      , OBSERVE_ONLY, X),” &
“ 283 (BC_4,STXDATA_3      , OBSERVE_ONLY, X),” &
“ 282 (BC_4,STXDATA_4      , OBSERVE_ONLY, X),” &
“ 281 (BC_4,STXDATA_5      , OBSERVE_ONLY, X),” &
```

```
" 280 (BC_4,STXDATA_6      , OBSERVE_ONLY, X)," &
" 279 (BC_4,STXDATA_7      , OBSERVE_ONLY, X)," &
" 278 (BC_4,STXDATA_8      , OBSERVE_ONLY, X)," &
" 277 (BC_4,STXDATA_9      , OBSERVE_ONLY, X)," &
" 276 (BC_4,STXDATA_10     , OBSERVE_ONLY, X)," &
" 275 (BC_4,STXDATA_11     , OBSERVE_ONLY, X)," &
" 274 (BC_4,STXDATA_12     , OBSERVE_ONLY, X)," &
" 273 (BC_4,STXDATA_13     , OBSERVE_ONLY, X)," &
" 272 (BC_4,STXDATA_14     , OBSERVE_ONLY, X)," &
" 271 (BC_4,STXDATA_15     , OBSERVE_ONLY, X)," &
" 270 (BC_1,*              , CONTROL     , 1)," &
" 269 (BC_7,STXENB_B       , BIDIR       , 1,  270,    1,    Z)," &
" 268 (BC_7,STXPRTY        , BIDIR       , 1,  270,    1,    Z)," &
" 267 (BC_7,STXSOC         , BIDIR       , 1,  270,    1,    Z)," &
" 266 (BC_1,*              , CONTROL     , 1)," &
" 265 (BC_7,STXCLAV        , BIDIR       , 1,  266,    1,    Z)," &
" 264 (BC_1,*              , CONTROL     , 1)," &
" 263 (BC_7,SRXCLAV        , BIDIR       , 1,  264,    1,    Z)," &
" 262 (BC_1,*              , CONTROL     , 1)," &
" 261 (BC_7,SRXSOC         , BIDIR       , 1,  262,    1,    Z)," &
" 260 (BC_7,SRXPRTY        , BIDIR       , 1,  262,    1,    Z)," &
" 259 (BC_1,*              , CONTROL     , 1)," &
" 258 (BC_7,SRXDATA_0      , BIDIR       , 1,  259,    1,    Z)," &
" 257 (BC_7,SRXDATA_1      , BIDIR       , 1,  259,    1,    Z)," &
" 256 (BC_7,SRXDATA_2      , BIDIR       , 1,  259,    1,    Z)," &
" 255 (BC_7,SRXDATA_3      , BIDIR       , 1,  259,    1,    Z)," &
" 254 (BC_7,SRXDATA_4      , BIDIR       , 1,  259,    1,    Z)," &
" 253 (BC_7,SRXDATA_5      , BIDIR       , 1,  259,    1,    Z)," &
" 252 (BC_7,SRXDATA_6      , BIDIR       , 1,  259,    1,    Z)," &
" 251 (BC_7,SRXDATA_7      , BIDIR       , 1,  259,    1,    Z)," &
" 250 (BC_1,*              , CONTROL     , 1)," &
" 249 (BC_7,SRXDATA_8      , BIDIR       , 1,  250,    1,    Z)," &
" 248 (BC_7,SRXDATA_9      , BIDIR       , 1,  250,    1,    Z)," &
" 247 (BC_7,SRXDATA_10     , BIDIR       , 1,  250,    1,    Z)," &
" 246 (BC_7,SRXDATA_11     , BIDIR       , 1,  250,    1,    Z)," &
" 245 (BC_7,SRXDATA_12     , BIDIR       , 1,  250,    1,    Z)," &
```

```
" 244 (BC_7,SRXDATA_13     , BIDIR       , 1,  250,    1,    Z)," &
" 243 (BC_7,SRXDATA_14     , BIDIR       , 1,  250,    1,    Z)," &
" 242 (BC_7,SRXDATA_15     , BIDIR       , 1,  250,    1,    Z)," &
" 241 (BC_4,SRXENB_B       , OBSERVE_ONLY, X)," &
" 240 (BC_4,SRXCLK         , CLOCK       , X)," &
" 239 (BC_4,MADD2          , OBSERVE_ONLY, X)," &
" 238 (BC_4,MADD3          , OBSERVE_ONLY, X)," &
" 237 (BC_4,MADD4          , OBSERVE_ONLY, X)," &
" 236 (BC_4,MADD5          , OBSERVE_ONLY, X)," &
" 235 (BC_4,MADD6          , OBSERVE_ONLY, X)," &
" 234 (BC_4,MADD7          , OBSERVE_ONLY, X)," &
" 233 (BC_4,MADD8          , OBSERVE_ONLY, X)," &
" 232 (BC_4,MADD9          , OBSERVE_ONLY, X)," &
" 231 (BC_4,MADD10         , OBSERVE_ONLY, X)," &
" 230 (BC_4,MADD11         , OBSERVE_ONLY, X)," &
" 229 (BC_4,MADD12         , OBSERVE_ONLY, X)," &
" 228 (BC_4,MADD13         , OBSERVE_ONLY, X)," &
" 227 (BC_4,MADD14         , OBSERVE_ONLY, X)," &
" 226 (BC_4,MADD15         , OBSERVE_ONLY, X)," &
" 225 (BC_4,MADD16         , OBSERVE_ONLY, X)," &
" 224 (BC_4,MADD17         , OBSERVE_ONLY, X)," &
" 223 (BC_4,MADD18         , OBSERVE_ONLY, X)," &
" 222 (BC_4,MADD19         , OBSERVE_ONLY, X)," &
" 221 (BC_4,MADD20         , OBSERVE_ONLY, X)," &
" 220 (BC_4,MADD21         , OBSERVE_ONLY, X)," &
" 219 (BC_4,MADD22         , OBSERVE_ONLY, X)," &
" 218 (BC_4,MADD23         , OBSERVE_ONLY, X)," &
" 217 (BC_4,MADD24         , OBSERVE_ONLY, X)," &
" 216 (BC_4,MADD25         , OBSERVE_ONLY, X)," &
" 215 (BC_4,MWSL_B         , OBSERVE_ONLY, X)," &
" 214 (BC_4,MWSH_B         , OBSERVE_ONLY, X)," &
" 213 (BC_4,MENDCYC_B      , OBSERVE_ONLY, X)," &
" 212 (BC_4,MSEL_B         , OBSERVE_ONLY, X)," &
" 211 (BC_4,MWR_B          , OBSERVE_ONLY, X)," &
" 210 (BC_4,MCLK           , CLOCK       , X)," &
" 209 (BC_1,*              , CONTROL     , 1)," &
```

```
" 208 (BC_1,MREQ0_B        , OUTPUT3     , 1,  209,    1,    Z)," &
" 207 (BC_1,MREQ1_B        , OUTPUT3     , 1,  209,    1,    Z)," &
" 206 (BC_1,MINT_B         , OUTPUT3     , 1,  209,    1,    Z)," &
" 205 (BC_1,*              , CONTROL     , 1)," &
" 204 (BC_1,MDTACK0_B      , OUTPUT3     , 1,  205,    1,    Z)," &
" 203 (BC_1,*              , CONTROL     , 1)," &
" 202 (BC_1,MDTACK1_B      , OUTPUT3     , 1,  203,    1,    Z)," &
" 201 (BC_1,*              , CONTROL     , 1)," &
" 200 (BC_7,MDATA_0        , BIDIR       , 1,  201,    1,    Z)," &
" 199 (BC_7,MDATA_1        , BIDIR       , 1,  201,    1,    Z)," &
" 198 (BC_7,MDATA_2        , BIDIR       , 1,  201,    1,    Z)," &
" 197 (BC_7,MDATA_3        , BIDIR       , 1,  201,    1,    Z)," &
" 196 (BC_7,MDATA_4        , BIDIR       , 1,  201,    1,    Z)," &
" 195 (BC_7,MDATA_5        , BIDIR       , 1,  201,    1,    Z)," &
" 194 (BC_7,MDATA_6        , BIDIR       , 1,  201,    1,    Z)," &
" 193 (BC_7,MDATA_7        , BIDIR       , 1,  201,    1,    Z)," &
" 192 (BC_1,*              , CONTROL     , 1)," &
" 191 (BC_7,MDATA_8        , BIDIR       , 1,  192,    1,    Z)," &
" 190 (BC_7,MDATA_9        , BIDIR       , 1,  192,    1,    Z)," &
" 189 (BC_7,MDATA_10       , BIDIR       , 1,  192,    1,    Z)," &
" 188 (BC_7,MDATA_11       , BIDIR       , 1,  192,    1,    Z)," &
" 187 (BC_7,MDATA_12       , BIDIR       , 1,  192,    1,    Z)," &
" 186 (BC_7,MDATA_13       , BIDIR       , 1,  192,    1,    Z)," &
" 185 (BC_7,MDATA_14       , BIDIR       , 1,  192,    1,    Z)," &
" 184 (BC_7,MDATA_15       , BIDIR       , 1,  192,    1,    Z)," &
" 183 (BC_1,*              , CONTROL     , 1)," &
" 182 (BC_7,MDATA_16       , BIDIR       , 1,  183,    1,    Z)," &
" 181 (BC_7,MDATA_17       , BIDIR       , 1,  183,    1,    Z)," &
" 180 (BC_7,MDATA_18       , BIDIR       , 1,  183,    1,    Z)," &
" 179 (BC_7,MDATA_19       , BIDIR       , 1,  183,    1,    Z)," &
" 178 (BC_7,MDATA_20       , BIDIR       , 1,  183,    1,    Z)," &
" 177 (BC_7,MDATA_21       , BIDIR       , 1,  183,    1,    Z)," &
" 176 (BC_7,MDATA_22       , BIDIR       , 1,  183,    1,    Z)," &
" 175 (BC_7,MDATA_23       , BIDIR       , 1,  183,    1,    Z)," &
" 174 (BC_1,*              , CONTROL     , 1)," &
" 173 (BC_7,MDATA_24       , BIDIR       , 1,  174,    1,    Z)," &
```

```
" 172 (BC_7,MDATA_25       , BIDIR       , 1,  174,    1,    Z)," &
" 171 (BC_7,MDATA_26       , BIDIR       , 1,  174,    1,    Z)," &
" 170 (BC_7,MDATA_27       , BIDIR       , 1,  174,    1,    Z)," &
" 169 (BC_7,MDATA_28       , BIDIR       , 1,  174,    1,    Z)," &
" 168 (BC_7,MDATA_29       , BIDIR       , 1,  174,    1,    Z)," &
" 167 (BC_7,MDATA_30       , BIDIR       , 1,  174,    1,    Z)," &
" 166 (BC_7,MDATA_31       , BIDIR       , 1,  174,    1,    Z)," &
" 165 (BC_4,TXFULL_B       , OBSERVE_ONLY, X)," &
" 164 (BC_1,*              , CONTROL     , 1)," &
" 163 (BC_1,TXENB_B        , OUTPUT3     , 1,  164,    1,    Z)," &
" 162 (BC_1,*              , CONTROL     , 1)," &
" 161 (BC_1,TXADDR_4       , OUTPUT3     , 1,  162,    1,    Z)," &
" 160 (BC_1,TXADDR_0       , OUTPUT3     , 1,  162,    1,    Z)," &
" 159 (BC_1,TXADDR_1       , OUTPUT3     , 1,  162,    1,    Z)," &
" 158 (BC_1,TXADDR_2       , OUTPUT3     , 1,  162,    1,    Z)," &
" 157 (BC_1,TXADDR_3       , OUTPUT3     , 1,  162,    1,    Z)," &
" 156 (BC_1,TXSOC          , OUTPUT3     , 1,  164,    1,    Z)," &
" 155 (BC_1,TXPRTY         , OUTPUT3     , 1,  164,    1,    Z)," &
" 154 (BC_1,*              , CONTROL     , 1)," &
" 153 (BC_1,TXDATA_0       , OUTPUT3     , 1,  154,    1,    Z)," &
" 152 (BC_1,TXDATA_1       , OUTPUT3     , 1,  154,    1,    Z)," &
" 151 (BC_1,TXDATA_2       , OUTPUT3     , 1,  154,    1,    Z)," &
" 150 (BC_1,TXDATA_3       , OUTPUT3     , 1,  154,    1,    Z)," &
" 149 (BC_1,TXDATA_4       , OUTPUT3     , 1,  154,    1,    Z)," &
" 148 (BC_1,TXDATA_5       , OUTPUT3     , 1,  154,    1,    Z)," &
" 147 (BC_1,TXDATA_6       , OUTPUT3     , 1,  154,    1,    Z)," &
" 146 (BC_1,TXDATA_7       , OUTPUT3     , 1,  154,    1,    Z)," &
" 145 (BC_1,*              , CONTROL     , 1)," &
" 144 (BC_1,TXDATA_8       , OUTPUT3     , 1,  145,    1,    Z)," &
" 143 (BC_1,TXDATA_9       , OUTPUT3     , 1,  145,    1,    Z)," &
" 142 (BC_1,TXDATA_10      , OUTPUT3     , 1,  145,    1,    Z)," &
" 141 (BC_1,TXDATA_11      , OUTPUT3     , 1,  145,    1,    Z)," &
" 140 (BC_1,TXDATA_12      , OUTPUT3     , 1,  145,    1,    Z)," &
" 139 (BC_1,TXDATA_13      , OUTPUT3     , 1,  145,    1,    Z)," &
" 138 (BC_1,TXDATA_14      , OUTPUT3     , 1,  145,    1,    Z)," &
" 137 (BC_1,TXDATA_15      , OUTPUT3     , 1,  145,    1,    Z)," &
```

```
" 136 (BC_4,RXEMPTY_B      , OBSERVE_ONLY, X)," &
" 135 (BC_4,RXSOC          , OBSERVE_ONLY, X)," &
" 134 (BC_4,RXPRTY         , OBSERVE_ONLY, X)," &
" 133 (BC_4,RXDATA_0       , OBSERVE_ONLY, X)," &
" 132 (BC_4,RXDATA_1       , OBSERVE_ONLY, X)," &
" 131 (BC_4,RXDATA_2       , OBSERVE_ONLY, X)," &
" 130 (BC_4,RXDATA_3       , OBSERVE_ONLY, X)," &
" 129 (BC_4,RXDATA_4       , OBSERVE_ONLY, X)," &
" 128 (BC_4,RXDATA_5       , OBSERVE_ONLY, X)," &
" 127 (BC_4,RXDATA_6       , OBSERVE_ONLY, X)," &
" 126 (BC_4,RXDATA_7       , OBSERVE_ONLY, X)," &
" 125 (BC_4,RXDATA_8       , OBSERVE_ONLY, X)," &
" 124 (BC_4,RXDATA_9       , OBSERVE_ONLY, X)," &
" 123 (BC_4,RXDATA_10      , OBSERVE_ONLY, X)," &
" 122 (BC_4,RXDATA_11      , OBSERVE_ONLY, X)," &
" 121 (BC_4,RXDATA_12      , OBSERVE_ONLY, X)," &
" 120 (BC_4,RXDATA_13      , OBSERVE_ONLY, X)," &
" 119 (BC_4,RXDATA_14      , OBSERVE_ONLY, X)," &
" 118 (BC_4,RXDATA_15      , OBSERVE_ONLY, X)," &
" 117 (BC_1,*              , CONTROL     , 1)," &
" 116 (BC_1,RXENB_B        , OUTPUT3     , 1,  117,    1,    Z)," &
" 115 (BC_1,RXADDR_0       , OUTPUT3     , 1,  117,    1,    Z)," &
" 114 (BC_1,RXADDR_1       , OUTPUT3     , 1,  117,    1,    Z)," &
" 113 (BC_1,RXADDR_2       , OUTPUT3     , 1,  117,    1,    Z)," &
" 112 (BC_1,RXADDR_3       , OUTPUT3     , 1,  117,    1,    Z)," &
" 111 (BC_1,RXADDR_4       , OUTPUT3     , 1,  117,    1,    Z)," &
" 110 (BC_1,*              , CONTROL     , 1)," &
" 109 (BC_1,EACOE_B        , OUTPUT3     , 1,  110,    1,    Z)," &
" 108 (BC_1,EACSM_B        , OUTPUT3     , 1,  110,    1,    Z)," &
" 107 (BC_1,EACCLK         , OUTPUT3     , 1,  110,    1,    Z)," &
" 106 (BC_1,*              , CONTROL     , 1)," &
" 105 (BC_7,EACDATA_0      , BIDIR       , 1,  106,    1,    Z)," &
" 104 (BC_7,EACDATA_1      , BIDIR       , 1,  106,    1,    Z)," &
" 103 (BC_7,EACDATA_2      , BIDIR       , 1,  106,    1,    Z)," &
" 102 (BC_7,EACDATA_3      , BIDIR       , 1,  106,    1,    Z)," &
" 101 (BC_7,EACDATA_4      , BIDIR       , 1,  106,    1,    Z)," &
```

```
" 100 (BC_7,EACDATA_5      , BIDIR      , 1,  106,    1,    Z)," &
"  99 (BC_7,EACDATA_6      , BIDIR      , 1,  106,    1,    Z)," &
"  98 (BC_7,EACDATA_7      , BIDIR      , 1,  106,    1,    Z)," &
"  97 (BC_1,*              , CONTROL    , 1)," &
"  96 (BC_7,EACDATA_8      , BIDIR      , 1,   97,    1,    Z)," &
"  95 (BC_7,EACDATA_9      , BIDIR      , 1,   97,    1,    Z)," &
"  94 (BC_7,EACDATA_10     , BIDIR      , 1,   97,    1,    Z)," &
"  93 (BC_7,EACDATA_11     , BIDIR      , 1,   97,    1,    Z)," &
"  92 (BC_7,EACDATA_12     , BIDIR      , 1,   97,    1,    Z)," &
"  91 (BC_7,EACDATA_13     , BIDIR      , 1,   97,    1,    Z)," &
"  90 (BC_7,EACDATA_14     , BIDIR      , 1,   97,    1,    Z)," &
"  89 (BC_7,EACDATA_15     , BIDIR      , 1,   97,    1,    Z)," &
"  88 (BC_1,*              , CONTROL    , 1)," &
"  87 (BC_7,EACDATA_16     , BIDIR      , 1,   88,    1,    Z)," &
"  86 (BC_7,EACDATA_17     , BIDIR      , 1,   88,    1,    Z)," &
"  85 (BC_7,EACDATA_18     , BIDIR      , 1,   88,    1,    Z)," &
"  84 (BC_7,EACDATA_19     , BIDIR      , 1,   88,    1,    Z)," &
"  83 (BC_7,EACDATA_20     , BIDIR      , 1,   88,    1,    Z)," &
"  82 (BC_7,EACDATA_21     , BIDIR      , 1,   88,    1,    Z)," &
"  81 (BC_7,EACDATA_22     , BIDIR      , 1,   88,    1,    Z)," &
"  80 (BC_7,EACDATA_23     , BIDIR      , 1,   88,    1,    Z)," &
"  79 (BC_1,*              , CONTROL    , 1)," &
"  78 (BC_7,EACDATA_24     , BIDIR      , 1,   79,    1,    Z)," &
"  77 (BC_7,EACDATA_25     , BIDIR      , 1,   79,    1,    Z)," &
"  76 (BC_7,EACDATA_26     , BIDIR      , 1,   79,    1,    Z)," &
"  75 (BC_7,EACDATA_27     , BIDIR      , 1,   79,    1,    Z)," &
"  74 (BC_7,EACDATA_28     , BIDIR      , 1,   79,    1,    Z)," &
"  73 (BC_7,EACDATA_29     , BIDIR      , 1,   79,    1,    Z)," &
"  72 (BC_7,EACDATA_30     , BIDIR      , 1,   79,    1,    Z)," &
"  71 (BC_7,EACDATA_31     , BIDIR      , 1,   79,    1,    Z)," &
"  70 (BC_1,*              , CONTROL    , 1)," &
"  69 (BC_1,EMWR_B         , OUTPUT3    , 1,   70,    1,    Z)," &
"  68 (BC_1,EMWSL_B        , OUTPUT3    , 1,   70,    1,    Z)," &
"  67 (BC_1,EMWSH_B        , OUTPUT3    , 1,   70,    1,    Z)," &
"  66 (BC_1,*              , CONTROL    , 1)," &
"  65 (BC_7,EMADD_2        , BIDIR      , 1,   66,    1,    Z)," &
```

```
“  64 (BC_7,EMADD_3        , BIDIR      , 1,   66,    1,    Z),” &
“  63 (BC_7,EMADD_4        , BIDIR      , 1,   66,    1,    Z),” &
“  62 (BC_7,EMADD_5        , BIDIR      , 1,   66,    1,    Z),” &
“  61 (BC_7,EMADD_6        , BIDIR      , 1,   66,    1,    Z),” &
“  60 (BC_7,EMADD_7        , BIDIR      , 1,   66,    1,    Z),” &
“  59 (BC_7,EMADD_8        , BIDIR      , 1,   66,    1,    Z),” &
“  58 (BC_7,EMADD_9        , BIDIR      , 1,   66,    1,    Z),” &
“  57 (BC_7,EMADD_10       , BIDIR      , 1,   66,    1,    Z),” &
“  56 (BC_7,EMADD_11       , BIDIR      , 1,   66,    1,    Z),” &
“  55 (BC_7,EMADD_12       , BIDIR      , 1,   66,    1,    Z),” &
“  54 (BC_1,*              , CONTROL    , 1),” &
“  53 (BC_7,EMADD_13       , BIDIR      , 1,   54,    1,    Z),” &
“  52 (BC_7,EMADD_14       , BIDIR      , 1,   54,    1,    Z),” &
“  51 (BC_7,EMADD_15       , BIDIR      , 1,   54,    1,    Z),” &
“  50 (BC_7,EMADD_16       , BIDIR      , 1,   54,    1,    Z),” &
“  49 (BC_7,EMADD_17       , BIDIR      , 1,   54,    1,    Z),” &
“  48 (BC_7,EMADD_18       , BIDIR      , 1,   54,    1,    Z),” &
“  47 (BC_1,*              , CONTROL    , 1),” &
“  46 (BC_1,EMADD_B_19     , OUTPUT3    , 1,   47,    1,    Z),” &
“  45 (BC_7,EMADD_19       , BIDIR      , 1,   54,    1,    Z),” &
“  44 (BC_1,EMADD_B_20     , OUTPUT3    , 1,   47,    1,    Z),” &
“  43 (BC_7,EMADD_20       , BIDIR      , 1,   54,    1,    Z),” &
“  42 (BC_1,EMADD_B_21     , OUTPUT3    , 1,   47,    1,    Z),” &
“  41 (BC_7,EMADD_21       , BIDIR      , 1,   54,    1,    Z),” &
“  40 (BC_1,EMADD_B_22     , OUTPUT3    , 1,   47,    1,    Z),” &
“  39 (BC_7,EMADD_22       , BIDIR      , 1,   54,    1,    Z),” &
“  38 (BC_1,EMADD_B_23     , OUTPUT3    , 1,   47,    1,    Z),” &
“  37 (BC_7,EMADD_23       , BIDIR      , 1,   54,    1,    Z),” &
“  36 (BC_1,EMOE_B         , OUTPUT3    , 1,   70,    1,    Z),” &
“  35 (BC_1,*              , CONTROL    , 1),” &
“  34 (BC_7,EMDATA_0       , BIDIR      , 1,   35,    1,    Z),” &
“  33 (BC_7,EMDATA_1       , BIDIR      , 1,   35,    1,    Z),” &
“  32 (BC_7,EMDATA_2       , BIDIR      , 1,   35,    1,    Z),” &
“  31 (BC_7,EMDATA_3       , BIDIR      , 1,   35,    1,    Z),” &
“  30 (BC_7,EMDATA_4       , BIDIR      , 1,   35,    1,    Z),” &
“  29 (BC_7,EMDATA_5       , BIDIR      , 1,   35,    1,    Z),” &
```

```
"  28 (BC_7,EMDATA_6       , BIDIR       , 1,   35,    1,    Z)," &
"  27 (BC_7,EMDATA_7       , BIDIR       , 1,   35,    1,    Z)," &
"  26 (BC_1,*              , CONTROL     , 1)," &
"  25 (BC_7,EMDATA_8       , BIDIR       , 1,   26,    1,    Z)," &
"  24 (BC_7,EMDATA_9       , BIDIR       , 1,   26,    1,    Z)," &
"  23 (BC_7,EMDATA_10      , BIDIR       , 1,   26,    1,    Z)," &
"  22 (BC_7,EMDATA_11      , BIDIR       , 1,   26,    1,    Z)," &
"  21 (BC_7,EMDATA_12      , BIDIR       , 1,   26,    1,    Z)," &
"  20 (BC_7,EMDATA_13      , BIDIR       , 1,   26,    1,    Z)," &
"  19 (BC_7,EMDATA_14      , BIDIR       , 1,   26,    1,    Z)," &
"  18 (BC_7,EMDATA_15      , BIDIR       , 1,   26,    1,    Z)," &
"  17 (BC_1,*              , CONTROL     , 1)," &
"  16 (BC_7,EMDATA_16      , BIDIR       , 1,   17,    1,    Z)," &
"  15 (BC_7,EMDATA_17      , BIDIR       , 1,   17,    1,    Z)," &
"  14 (BC_7,EMDATA_18      , BIDIR       , 1,   17,    1,    Z)," &
"  13 (BC_7,EMDATA_19      , BIDIR       , 1,   17,    1,    Z)," &
"  12 (BC_7,EMDATA_20      , BIDIR       , 1,   17,    1,    Z)," &
"  11 (BC_7,EMDATA_21      , BIDIR       , 1,   17,    1,    Z)," &
"  10 (BC_7,EMDATA_22      , BIDIR       , 1,   17,    1,    Z)," &
"   9 (BC_7,EMDATA_23      , BIDIR       , 1,   17,    1,    Z)," &
"   8 (BC_1,*              , CONTROL     , 1)," &
"   7 (BC_7,EMDATA_24      , BIDIR       , 1,    8,    1,    Z)," &
"   6 (BC_7,EMDATA_25      , BIDIR       , 1,    8,    1,    Z)," &
"   5 (BC_7,EMDATA_26      , BIDIR       , 1,    8,    1,    Z)," &
"   4 (BC_7,EMDATA_27      , BIDIR       , 1,    8,    1,    Z)," &
"   3 (BC_7,EMDATA_28      , BIDIR       , 1,    8,    1,    Z)," &
"   2 (BC_7,EMDATA_29      , BIDIR       , 1,    8,    1,    Z)," &
"   1 (BC_7,EMDATA_30      , BIDIR       , 1,    8,    1,    Z)," &
"   0 (BC_7,EMDATA_31      , BIDIR       , 1,    8,    1,    Z)";

end atmc_top_palm;
```

INDEX

INDEX

D

E

INDEX

F

G

H

I

J

INDEX

INDEX

INDEX

S

INDEX

INDEX

W

Z