NETWORK ACCESS PRODUCTS

Bt8233

ATM ServiceSAR
with xBR Traffic Management

Rockwell Semiconductor Systems

Advance Information

This document contains information on a product under development. The parametric information contains target parameters that are subject to change.

Bt8233

ATM ServiceSAR with xBR Traffic Management

The Bt8233 Service Segmentation and Reassembly Controller integrates ATM terminal functions, Peripheral Component Interconnect (PCI) Bus Master and Slave controllers, and a Universal Test and Operation Physical Interface (UTOPIA) interface with service specific functions in a single package. The *Service*SAR Controller generates and terminates Asynchronous Transfer Mode (ATM) traffic and automatically schedules cells for transmission. The Bt8233 is targeted at 155 Mbit/s throughput systems where the number of Virtual Circuit Channels (VCCs) is relatively large, or the performance of the overall system is critical. Examples of such networking equipment include Routers, Ethernet switches, ATM Edge switches, or Frame Relay switches.

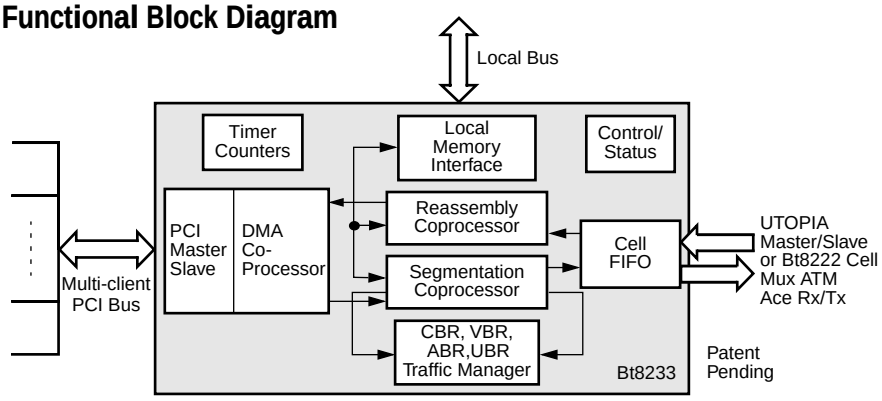
Service-Specific Performance Accelerators

The Bt8233 incorporates numerous service-specific features designed to accelerate and enhance system performance. For example, the Bt8233 implements Echo Suppression of Local Area Network (LAN) traffic via LAN Emulation Clients (LECID) filtering and supports Frame Relay Discard Eligibility (DE) to Cell Loss Priority (CLP) interworking.

Advanced xBR Traffic Management

The xBR Traffic Manager in the Bt8233 supports multiple ATM service categories. This includes Constant Bit Rate (CBR), Variable Bit Rate (VBR) (both single and dual leaky bucket), Unspecified Bit Rate (UBR) and Available Bit Rate (ABR). The Bt8233 manages each VCC independently. It dynamically schedules segmentation traffic to comply with up to 8+CBR user-configured scheduling priorities for the various traffic classes. Scheduling is controlled by a schedule table configured by the user and based on a user-specified time reference. ABR channels are managed in hardware according to user programmable ABR templates. These templates tune the performance of the Bt8233's ABR algorithms to the requirements of a specific system or network.

Functional Block Diagram



Service Specific Performance Accelerators

- LECID Filtering and Echo Suppression
- Dual Leaky Bucket based on CLP (Frame Relay)
- Frame Relay DE Interworking
- Internal Simple Network Management Protocol Management Information Base (SNMP MIB) counters

Flexible Architectures

- Multi-peer host
- Direct switch attachment via Reverse UTOPIA
- ATM Terminal
  - Host control
  - Local Bus control
- Optional local processor

xBR Traffic Management (TM)

- TM4.0 Service Classes
  - CBR
  - VBR (single, dual & CLP-based leaky buckets)
  - Real time VBR
  - ABR
  - UBR
  - Generic Flow Control (GFC) controlled & uncontrolled flows
- 8 Levels of priorities (8 + CBR)
- Dynamic per-VCC scheduling
- Multiple programmable ABR Templates (supplied by Rockwell or user)
- Scheduler driven by local system clock for low jitter CBR
- Internal Resource Management (RM) Operation and Maintenance (OAM) cell feedback path
- Virtual FIFO Rate Matching (Source Rate Matching)
- Tunnelling
  - Virtual Path (VP) Tunnels (Virtual Channel Identifier (VCI) interleaving on Protocol Data Units (PDU) boundaries)
  - CBR Tunnels (cells interleaved on UBR with an aggregate CBR limit)

Multi-Queue Segmentation Processing

The Bt8233's Segmentation Coprocessor generates ATM cells for up to 64K VCCs at a line rate of up to 200 Mbit/sec for simplex connections. The segmentation coprocessor formats cells on each channel according to segmentation VCC tables, utilizing up to 32 independent transmit queues and reporting segmentation status on a parallel set of up to 32 segmentation status queues. The segmentation coprocessor fetches client data from the host, formats ATM cells while generating and appending protocol overhead, and forwards these to the UTOPIA port. The segmentation coprocessor operates as a slave to the xBR Traffic Manager which schedules VCCs for transmission.

Multi-Queue Reassembly Processing

The Bt8233's Reassembly Coprocessor stores the payload data from the cell stream received by the UTOPIA port into host data buffers. Using a dynamic lookup method which supports Network to Network Interface (NNI) or User-Network Interface (UNI) addressing, the reassembly coprocessor processes up to 64K VCCs simultaneously. The host supplies free buffers on up to 32 independent free buffer queues. The reassembly coprocessor performs all Common Part Convergence Sublayer (CPCS) protocol checks and reports the results of these checks + other status data on 1 of 32 independent reassembly status queues.

High Performance Host Architecture with Buffer Isolation

The Bt8233 host interface architecture maximizes performance and system flexibility. The device's control and status queues enable Host/Segmentation Reassembly (SAR) communication via write operations alone. This lowers latency and PCI bus occupancy. Flexibility is achieved by supporting a scalable peer-to-peer architecture. Multiple host clients may be addressed by the SAR as separate physical or logical PCI peers. Segmentation and reassembly data buffers on the host system are identified by buffer descriptors in SRC shared (or host) memory which contain pointers to buffers. The use of buffer descriptors in this way allows for isolation of data buffers from the mechanisms that handle buffer allocation and linking. This provides a layer of indirection in buffer assignment and management that maximizes system architecture flexibility.

Designer Toolkit

Rockwell provides an evaluation system for the Bt8233, which provides a working reference design, an example software driver, and facilities for generating and terminating all service categories of ATM traffic. This system accelerates ATM system development by providing a rapid prototyping environment.

Print date: October 1997

Detailed Feature Summary

Multi-Queue Segmentation Processing

- 32 Transmit queues with optional priority levels
- 64K VCCs maximum(1)
- ATM Adaptation Layer (AAL5) CPCS generation
- AAL0 Null CPCS (optional use of PTI for PDU demarcation)
- ATM cell header generation
- Raw cell mode (52 octet)
- 200 MBit/sec half duplex
- 155 MBit/sec full duplex (w/ 2-cell PDUs)
- Variable length transmit First In First Out (FIFO) - Cell Delay Variation (CDV) - Host latency matching (1 to 9 cells)
- Symmetric Tx and Rx architecture
  - Buffer descriptors
  - Queues
- User defined field circulates back to the host (32 bits)
- Distributed host or SRC shared memory segmentation
- Simultaneous segmentation and reassembly
- Per-PDU control of CLP/PTI (UBR)
- Per-PDU control of AAL5 UU field
- Message & Streaming Status Modes
- Virtual Tx FIFO (PCI host)

Generous Implementation of OAM-PM Protocols

- Detection of all F4/F5 OAM flows
- Internal PM monitoring and generation for up to 128 VCCs
- Optional Global OAM Rx/Tx Queues
- In-Line OAM insertion & generation

Standards Compliance

- User Network Interface/Network to Network Interface (UNI/NNI) 3.1
- TM 4.0
- I.363
- I.610 /GR-1248
- AToM MIB (RFC1695)
- Interim Local Management Interface (ILMI) MIB
- American National Standards Institute (ANSI) T1.635
- GFC per I.361
- SNMP

Statistics and Write-Only Counters

- Global register counter of # of cells transmitted
- Global register counter of # of cells received on active channels
- Global T register counter of # of cells received on inactive channels
- Global register counter of # of AAL5 CPCS-PDUs discarded due to per-channel firewall, etc
- Reassembly (Rsm) per-VCC Service Discard counters (Frame Relay & LAN Emulation (LANE))
- One programmable interval timer (32 bits w/ interrupt)

High Performance Host Architecture with Buffer Isolation

- Write-Only Control and Status
- Read Multiple command for data transfer
- Up to 32 Host Clients Control and Status Queues
- Physical or logical clients
  - Enables peer-to-peer architecture
- Descriptor-based buffer chaining
- Scatter/Gather Direct Memory Access (DMA)
- Endian Neutral
- Non-word (byte) aligned host buffer addresses
- Automatically detects presence of Tx data or Rx free buffers
- Virtual FIFOs (PCI bursts treated as a single address)
- Hardware indication of Beginning of Message (BOM)
- Allows isolation of system resources

Standards-Based I/O

- 33 MHz PCI 2.0
- Physical Layer (PHY) Interfaces
  - UTOPIA Master (Level 1)
  - UTOPIA Slave (Level 1)
  - Bt8222 Cell Mux
- Flexible local memory architecture
- Optional local control interface
- Boundary Scan for board-level testing
- Source loopback, for diagnostics
- Glueless connection to Bt8222 ATM physical layer device

Designer Toolkit

- Evaluation System (Bt8233 EVS)
- Reference schematics
- Hardware Programming Interface - Bt8233HPI reference Source code (C)

Multi-Queue Reassembly Processing

- 32 reassembly queues
- 64K VCCs maximum(1)
- AAL5 CPCS Checking
- AAL0
  - Payload Type Identifier (PTI) Termination
  - Cell Count Termination
- Early Packet Discard (EPD), based on:
  - Receive buffer underflow
  - Receive status overflow
  - CLP with priority threshold
  - AAL5 max PDU length
  - Rx FIFO full
  - Frame Relay DE with priority threshold
  - LECID Filtering and Echo Suppression
  - Per-VCC firewalls
- Dynamic channel lookup (NNI or UNI addressing)
  - Supports full address space
  - Deterministic
  - Flexible VCI count per VPI
  - Optimized for signalling address assignment
- Message and Streaming Status Modes
- Raw cell mode (52 octet)
- 200 MBit/sec half duplex
- 155 MBit/sec full duplex (w/ 2-cell PDUs)
- Distributed host or SRC shared memory reassembly
- Eight programmable reassembly hardware time-outs (per-VCC assignable)
- Global max PDU length for AAL5
- Per-VCC buffer Firewall (memory usage limit)
- Simultaneous reassembly and segmentation
- Idle cell filtering
- 32K ABR VCCs

Electrical/Mechanical

- 208 pin Plastic Quad Flat Pack (PQFP)
- Pin-compatible with Bt8230 SRC
- Low Power 1.5 W (typical) @ full rate
- Industrial temperature range
- TTL level inputs
- Complementary Metal-Oxide Semiconductor (CMOS) level outputs

Note: (1) Depends on SRC shared memory size and device configuration.

Table of Contents

List of Figures

List of Tables

1.0 Bt8233 Product Overview

1.1 Introduction

The Bt8233 Service Segmentation and Reassembly Controller (*Service*SAR) delivers a wide range of advanced Asynchronous Transfer Mode (ATM), ATM Adaption Layer (AAL) and service specific features in a highly integrated CMOS package. Some of the Bt8233 service level features (for example, Virtual FIFO segmentation of circuit-based Constant Bit Rate (CBR) traffic, and Frame Relay Early Packet Discard (EPD) based on the Discard Eligibility (DE) field) provide system designers with capabilities of accelerating specific protocol interworking functions. Other service level functions, such as, Generic Flow Control (GFC) and echo suppression of multicast data frames on Emalated LAN (ELAN) channels, enable network level functionality or topologies.

In addition to meeting the requirements contained in User Network Interface (UNI) 3.1, the Bt8233 complies with ATM Forum Traffic Management specification TM4.0. The Bt8233 provides traffic shaping for all service categories:

- CBR
- Variable Bit Rate (VBR) (both single and dual leaky bucket)
- Unspecified Bit Rate (UBR)
- Available Bit Rate (ABR)

The internal xBR Traffic Manager automatically schedules each Virtual Circuit Channel (VCC) according to user assigned parameters.

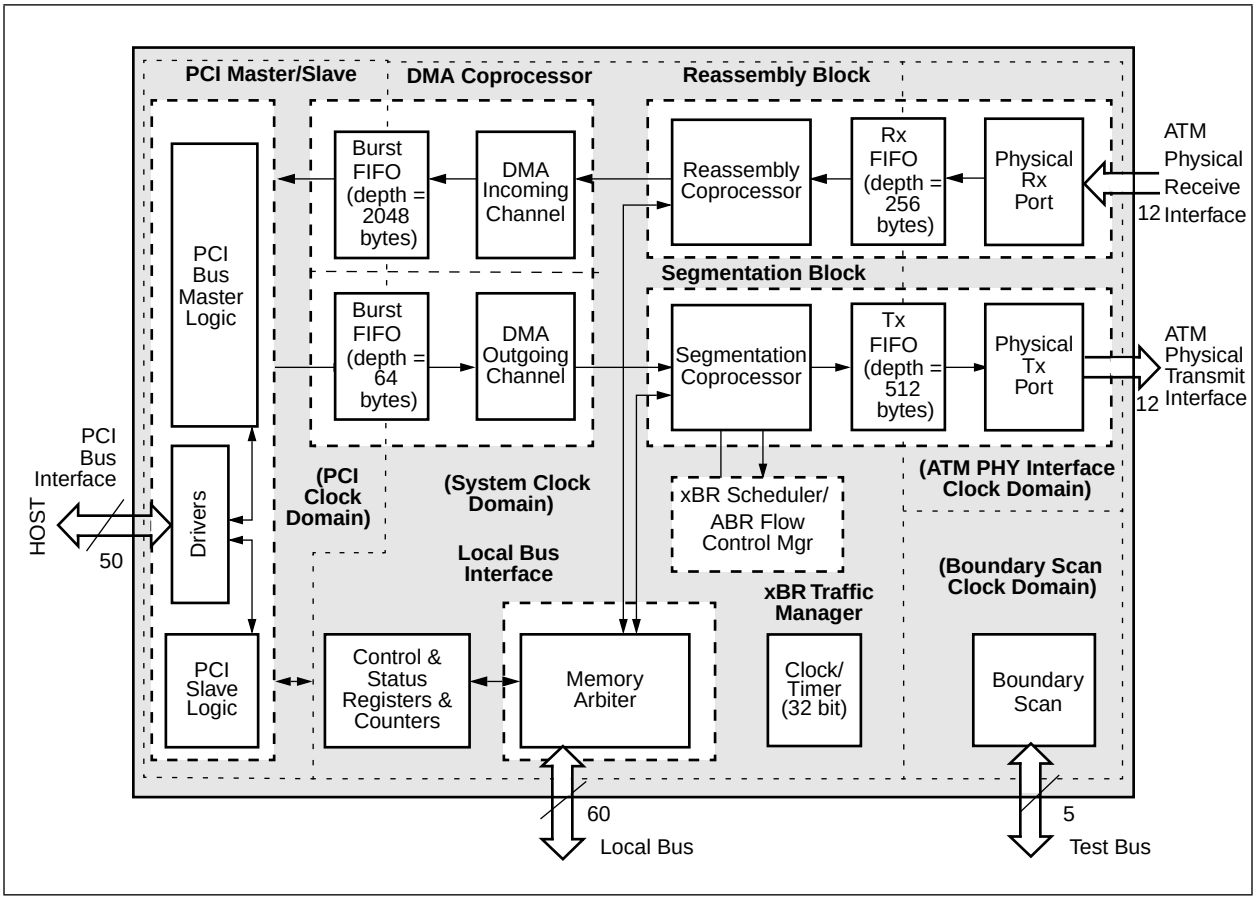
The Bt8233's architecture is designed to minimize and control host traffic congestion. The host manages the Bt8233 terminal using an efficient architecture that employs write-only control and status queues. For example, the host submits data for transmit by writing buffer descriptor pointers to one of 32 transmit queues. These entries may be thought of as task lists for the *Service*SAR to perform. The Bt8233 reports segmentation and reassembly status to the host by writing entries to segmentation and reassembly status queues, which the host then further processes. This architecture lessens the control burden on the host system and minimizes Peripheral Component Interconnect (PCI) bus utilization by eliminating reads across the PCI bus from host control activities.

The Bt8233 host interface provides for control of host congestion through the following mechanisms. First, each peer maintains separate control and status queues. Then, each VCC in a peer group may be limited to a specific maximum receive buffer utilization, further controlling congestion. EPD is supported for VCCs which exceed their resource allotments. On transmit, peers are assigned fixed or round-robin priority to ensure predictable servicing. For ABR, the host can implement a congestion notification algorithm with a simple one-word write to a SAR control register. The SAR will reduce the Explicit Rate (ER) field or set the Congestion Indication (CI/NI) bit in Turnaround RM cells based on user configuration.

The Bt8233 consists of five separate coprocessors each of which maintains state information in shared, off-chip memory. This memory is controlled by the SAR through the local bus interface, which arbitrates access to the bus between the various coprocessors. Although these coprocessors run off the same system clock, they operate asynchronously from each other. Communication between the coprocessors takes place through on-chip FIFOs or through queues in local memory. The five coprocessors are the Incoming Direct Memory Access (DMA), Outgoing DMA, Reassembly, Segmentation, and xBR Traffic Manager.

The Bt8233's on-chip coprocessor blocks are surrounded by high performance PCI and Universal Test and Operation Physical Interface (UTOPIA) ports for glueless interface to a variety of system components with full line rate throughput and low bus occupancy. Figure 1-1 illustrates these functional blocks.

Figure 1-1. Bt8233 Functional Block Diagram



1.2 Service Specific Performance Accelerators

The Bt8233 incorporates several service-specific features which accelerate system performance. Some of these service level features provide the possibility for designers to accelerate specific protocol interworking functions. Other service level features enable network level functionality. These features are outlined in Section 2.0, Bt8233 Architecture Overview, and fully are described in succeeding chapters.

1.2.1 UNI or NNI Addressing

The Bt8233 handles both UNI (8-bit VPI field) and Network to Network Interface (NNI) (12-bit VPI field) addresses.

1.2.2 Frame Relay Interworking

The VBR traffic category includes rate-shaping via the dual leaky bucket Generic Cell Rate Algorithm (GCRA) algorithm based on the Cell Loss Priority (CLP) bit, for use in Frame Relay.

1.2.3 Early Packet Discard

The EPD feature provides a mechanism to discard complete or partial CPCS-PDUs based upon service discard attributes or error conditions. The reassembly coprocessor performs EPD functions under the following conditions:

- Frame Relay packet discard based on the DE field in the received frame and the channel exceeding a user-defined priority threshold.
- Packet discard based on the CLP bit.
- LAN_LECID packet discard to implement echo suppression on multicast data frames on ELAN channels.
- Receive FIFO full condition/threshold.

1.2.4 CBR Traffic Handling

The segmentation coprocessor includes an internal rate-matching mechanism to match the internal rate (the local reference rate) of CBR segmentation to an external rate (the host rate).

The user can direct the Bt8233 to segment traffic from a fixed PCI address (i.e., a Virtual FIFO), for circuit-based CBR traffic.

The user can delineate up to 8 CBR pipes (or tunnels) in which to transmit multiple UBR or VBR channels. This allows proprietary management schemes to operate under a preallocated CBR bandwidth.

1.2.5 ABR Traffic Management

The ABR Flow Control Manager dynamically rate-shapes ABR traffic independently per-VCC, based upon network feedback. One or more ABR templates are used to govern the behavior of traffic.

- Both Relative Rate (RR) and ER algorithms are employed when computing a rate adjustment on an ABR VCC.
- Programmable ABR templates allow rate-shaping on groups of VCCs to be tuned for different network policies.
- The Bt8233 allows rate adjustments on Turnaround Resource Management (RM) cells, based on congestion in the host.
- The Bt8233 allows rate adjustments due to Use-It-Or-Lose-It behavior.
- The Bt8233 generates out-of-rate Forward RM cell(s) to restart scheduling of a VCC whose rate has dropped below the Schedule Table minimum rate.
- The Bt8233 optionally posts the current Allowed Cell Rate (ACR) on the segmentation status queue for the host monitoring functions.

1.2.6 VBR Traffic Management

The Bt8233 schedules each VBR VCC according to GCRA parameters stored in the individual VCC control tables. The internal xBR Traffic Manager schedules the transmitted data to maximize the permitted link utilization. The actual rate sent is accurate to within 0.15% of the negotiated rates over a range from 10 cells per second to full line rate of 155 Mbits/sec.

Three VBR modes are supported:

- Sustained Cell Rate (One Leaky Bucket).
- Peak and Sustained Cell Rate (Dual Leaky Bucket).
- CLP 0+1 shaping (supports committed/best effort services).

1.2.7 Virtual Path Networking

The Bt8233 can interleave segmentation of numerous VCCs (i.e., separate Virtual Circuit (VC) channels) as members of one Virtual Path (VP). VP based traffic shaping is supported. The entire VP is scheduled according to parameters for one VCC.

1.2.8 AAL for Proprietary Traffic

The Bt8233 incorporates an AAL0 traffic class for both segmentation and reassembly, which acts as an AAL level for proprietary use. Several options for packetization are implemented.

1.2.9 Optional Local Processing of ATM Management Traffic

The Bt8233's Local Processor Interface allows for an optional local processor to direct segmentation and reassembly of ATM management level traffic such as Operation and Maintenance (OAM) and PM cells, signalling and Interim Local Management Interface (ILMI). This off-loads network control traffic from the host, thereby focusing host processing power on the user application.

1.2.10 Internal SNMP MIB Counters

Bt8233 has three internal counters that measure cells received, cells discarded and AAL5 PDUs discarded (to meet ILMI and RFC1695 requirements).

1.3 Designer Toolkit

The Bt8233 ATM Evaluation System (EVS) provides complete evaluation capability for the Bt8233 Segmentation and Reassembly Controller (*Service*SAR). The EVS serves as a hardware and software reference design for development of customer-specific ATM applications. The Bt8233EVS was designed to provide a rapid prototyping environment to assist and speed customer development of new ATM products, thereby reducing product time to market.

The Bt8233EVS includes the following:

- Assists and speeds customer product development
- Provides hardware reference design
- Provides software reference design based on VxWorks
- Provides traffic generation and checking capability

Rockwell houses the Bt8233EVS in a PC-type chassis containing a PCI system board which serves as the host system for the Bt8233. The host system is designed to be an extremely flexible embedded systems platform for I/O applications with the PCI bus. Located on one of the host board's PCI expansion slots is the Bt8233EVM, a PCI card specifically designed to be a full-featured ATM controller implementing the full functionality of the Bt8233 *Service*SAR. The *Service*SAR resides at the heart of this PCI card.

The PCI interface between the host processor and the local system is controlled by Rockwell's Hardware Programming Interface (Bt8233HPI). The Bt8233HPI is a software driver to the Bt8233, on top of which a system designer can develop and place proprietary driver software. This interface allows users to easily port their applications to the Bt8233EVS. This software is written in C, and source code is available under license agreement.

The Bt8233EVS also includes a full set of design schematics and artwork for the Bt8233EVM PCI card.

2.0 Bt8233 Architecture Overview

2.1 Introduction

The Bt8233 *Service*SAR architecture enables it to efficiently handle high bandwidth throughput across the spectrum of two different ATM AALs and all ATM service categories. This chapter provides an overview of this architecture.

The first section describes the queue and data buffer system. Then, segmentation and reassembly functions are described from within the context of the queue structures. The last major operational description covers the xBR Traffic Manager. The remaining sections cover OAM/PM, the various device inputs and outputs, and the logic diagram with pin descriptions.

This architectural overview serves as a solid foundation for understanding the complete functionality of the Bt8233.

2.2 High Performance Host Architecture with Buffer Isolation

Once initialized and given a segmentation or reassembly task, the Bt8233 operates autonomously. Because the Bt8233 is a high performance subsystem, the Host/*Service*SAR architecture and the algorithms for task submission and status reporting have been optimized to minimize the control burden on the host system.

2.2.1 Multiple ATM Clients

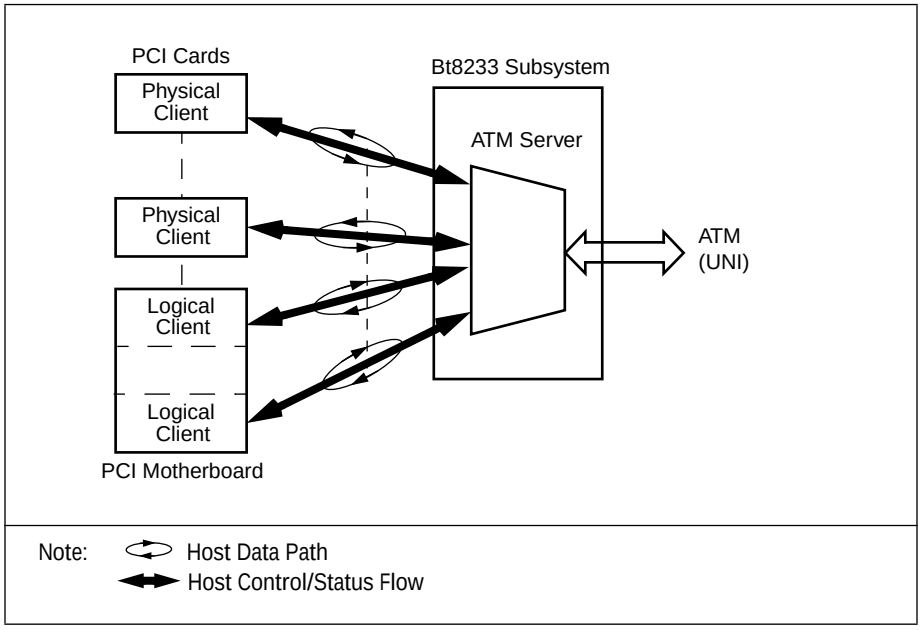
The Bt8233, functioning as an ATM UNI, provides a high throughput uplink to a broadband network. Most individual ATM service users (or clients) do not have the bandwidth requirements to equal the throughput capability of the Bt8233. As an application example, ATM clients may be Ethernet or Frame Relay ports. Therefore, many service clients are typically aggregated onto one ATM UNI. These clients may have very different needs and/or isolation requirements.

In order to fully capitalize on the high bandwidth of this service while meeting per-VCC Quality of Service (QoS) needs, the Bt8233 functions as an ATM server for up to 32 clients. In this way, the bandwidth requirements of the service user (the client) can be balanced with the service throughput capability of the Bt8233.

The Bt8233 provides multiple independent control and status communication paths. Each communication path, or flow, consists of a control queue and a status queue for both segmentation and reassembly. The host assigns each of these independent flows to system clients, or peers. These can be either physical or logical entities. As throughput requirements escalate, the host system can add processing power in the form of additional peers. This degree of freedom creates a scalable host environment. Multiple VCCs may be assigned to each client.

Each client interfaces to the Bt8233 independently. Due to its server architecture, the Bt8233 supplies the synchronization between asynchronous tasks requiring ATM services. Figure 2-1 illustrates this Client/Server model. It shows that clients may be multiple applications in a shared memory or separate physical entities. All communicate directly with the Bt8233.

Figure 2-1. Multiple Client Architecture Supports up to 32 Clients



2.2.2 Bt8233 Queue Structure

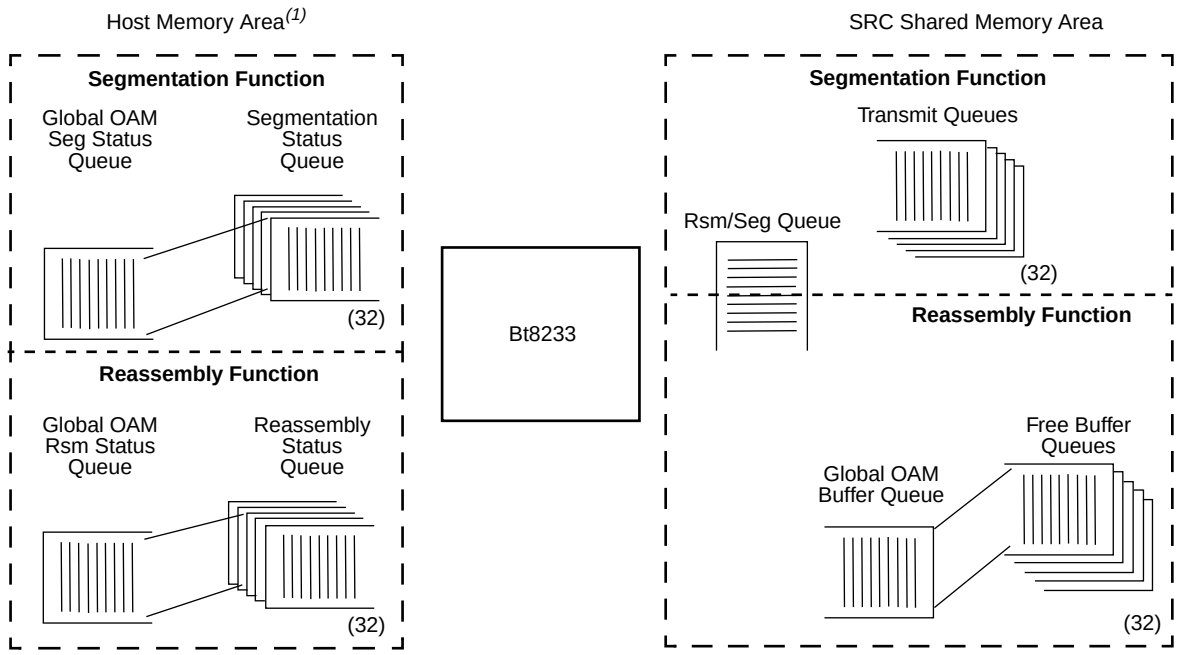
The flow of the reassembly, scheduling, and segmentation processes in the Bt8233 is monitored, coordinated, and controlled through the use of a full array of circular queues, serviced by the Bt8233 or by the host.

The following queues exist in SRC shared memory:

- Transmit Queues (up to 32 queues)
- Reassembly/Segmentation (Rsm Seg) Queue
- Free Buffer Queues (up to 32 queues)
  - includes the Global OAM Free Buffer Queue

Figure 2-2 illustrates the location of each queue.

Figure 2-2. Bt8233 Queue Architecture



Notes: (1). Status Queues may be optionally placed in SRC shared memory.

Transmit Queues are used by the host to submit chains of segmentation buffer descriptors to the Bt8233 for segmentation. The segmentation coprocessor then processes these Transmit Queue entries as part of the segmentation function.

The *Rsm/Seg Queue* is written to by the reassembly coprocessor and read by the segmentation coprocessor. The queue includes data on OAM-PM cells to be transmitted, as well as data on ABR-class received Backward_RM and Forward_RM cells. The Rsm/Seg Queue is a private queue for the SAR which the host cannot read or write.

The host furnishes data buffers to the reassembly processor by posting their location and availability to the *Free Buffer Queues*, one of which may be designated as the Global OAM Free Buffer Queue. The reassembly coprocessor uses

the Free Buffer Queue entries to allocate data buffers for received ATM cells during reassembly.

The following queues exist in host (or optionally, SRC shared) memory:

- Segmentation Status Queues (up to 32 queues)
  - Includes the Global OAM Segmentation Status Queue
- Reassembly Status Queues (up to 32 queues)
  - Includes the Global OAM Reassembly Status Queue

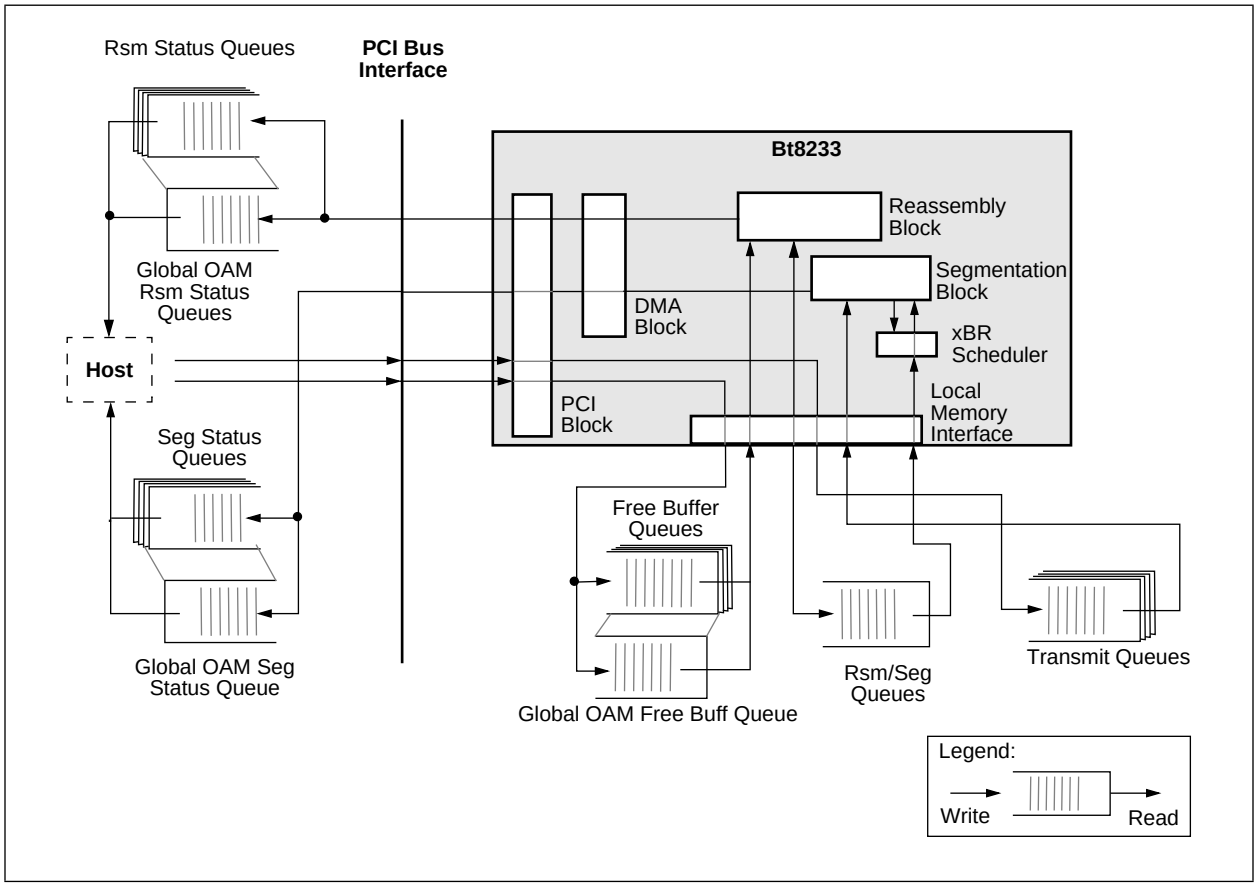
The Bt8233 reports segmentation status to the *Segmentation Status Queues*. One of these may be designated as the Global OAM Segmentation Status Queue. The host further processes these Segmentation Status Queue entries.

The Bt8233 reports reassembly status to the *Reassembly Status Queues*. One of these may be designated as the Global OAM Reassembly Status Queue. The host further processes these Reassembly Status Queue entries.

The queues described above provide the control information that fuels the reassembly and segmentation functions.

These queues, placed on asynchronous communication paths, directly associate the host with the Bt8233 during processing and associate each of the major functional blocks of the Bt8233 with each other. Figure 2-3 illustrates this associativity. The arrows indicate which system entity writes to the queue, and which entity reads the queue.

Figure 2-3. Interaction of Queues with Bt8233 Functional Blocks



2.2.3 Buffer Isolation Utilizing Descriptor-based Buffer Chaining

The Bt8233 employs buffer structures for reassembly and segmentation that maximize the flexibility of the system architecture by isolating the data buffers from the mechanisms that handle buffer allocation and linking. The data buffers, consequently, can contain only payload data, no control fields or other user fields. The user may store the data buffers separate from the buffer descriptors and implement a minimum data copy architecture.

Figure 2-4 illustrates how reassembly data buffers and buffer descriptors are chained together and manipulated by the *Service*SAR.

- The host creates a link between a reassembly data buffer and a buffer descriptor by writing in the buffer descriptor entry a pointer to the data buffer.
- The host then formats a Free Buffer Queue entry which includes pointers to both the data buffer and buffer descriptor, and writes this message to the Free Buffer Queue.
- The reassembly coprocessor reads this Free Buffer Queue entry and uses the pointer to the reassembly data buffer as the memory location to write the PDU being reassembled.
- As reassembly of that PDU progresses, the reassembly coprocessor chains together the necessary number of additional buffer descriptors (and thus their associated reassembly data buffers) to complete reassembly of the PDU.

Figure 2-4. Reassembly Buffer Isolation–Data Buffers Separated from Descriptors

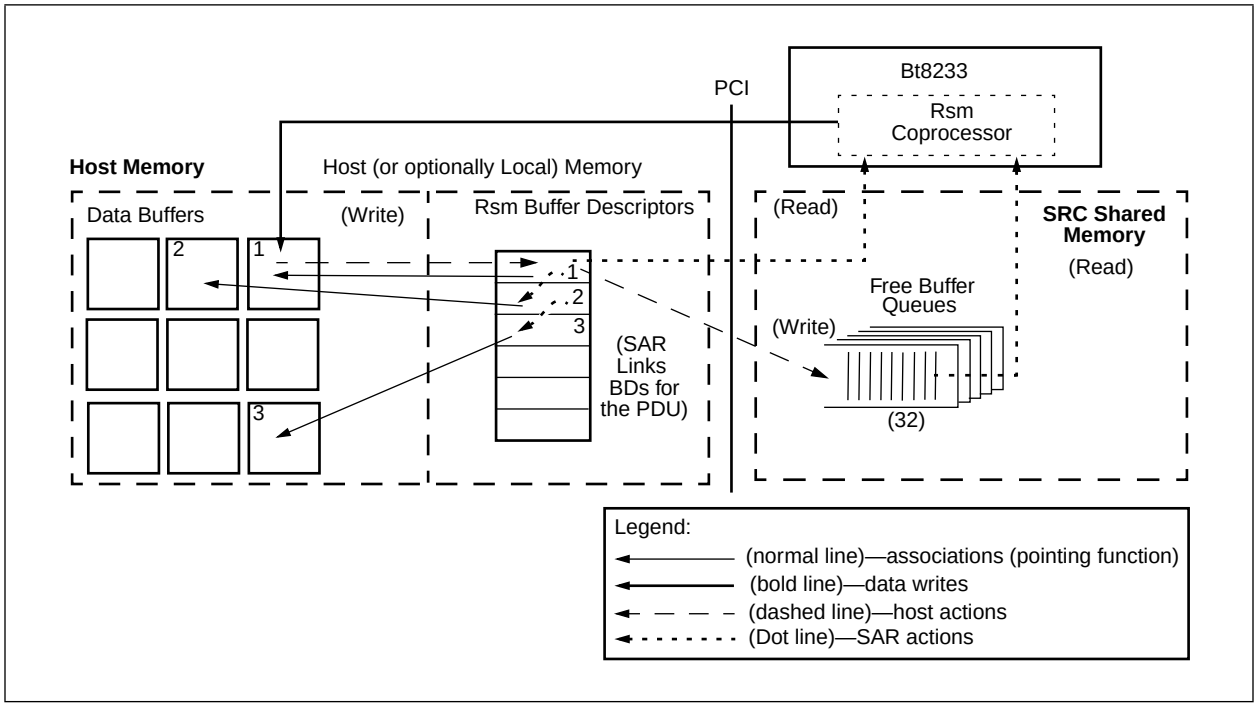
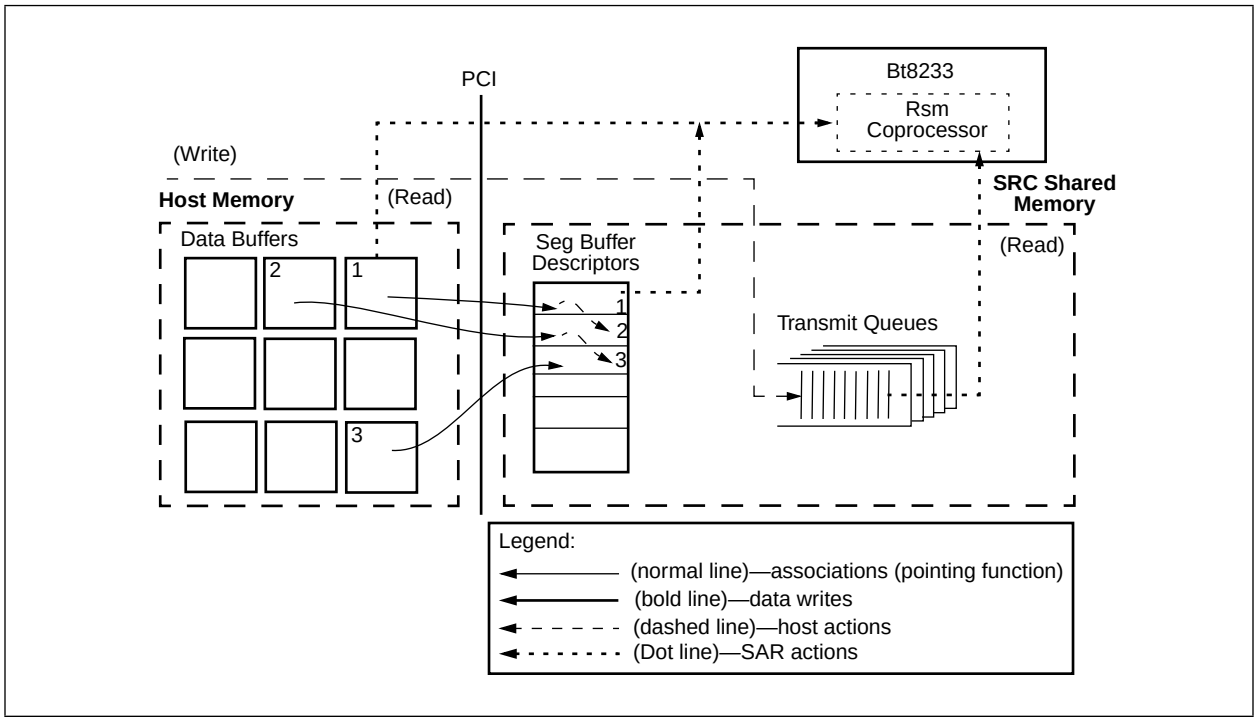


Figure 2-5 illustrates how the host submits linked data buffers (i.e., PDUs) to the Transmit Queue for segmentation. The process is as follows: the host links the buffer descriptors (in SRC shared memory) for the associated data buffers (in host memory) containing the PDU to be segmented. The host then formats a two-word Transmit Queue entry and writes this entry to the Transmit Queue. The location of the first buffer descriptor in the linked chain is contained in the Transmit Queue entry. The segmentation coprocessor automatically senses the presence of new Transmit Queue entries, reads them and schedules the new data for transmission. The Transmit Queue acts as a FIFO for segmentation task pointers.

Figure 2-5. Segmentation Buffer Isolation–Data Buffers Separated from Descriptors



2.2.4 Status Queue Relation to Buffers and Descriptors

The Status Queues employed by the Bt8233 are written by the SAR and read by the Host to further process the segmentation and reassembly data flow in progress or just completed. Each Status Queue entry includes data (such as error flags, status bits, etc.) which the Host uses in its succeeding process steps. The other principal function of the Status Queue entry is the inclusion of a pointer to the first buffer descriptor of the segmented or reassembled data buffer(s) which comprise a single PDU. This function establishes the association from the SAR to the Host of the successful or unsuccessful segmentation/reassembly of a PDU.

Figure 2-6 illustrates the association between the Segmentation Status Queues and segmentation data buffers and descriptors. The figure shows one three-buffer PDU on one virtual channel, represented by a single entry in one of the Transmit Queues and a single entry in one of the Segmentation Status Queues. The Host links the buffer descriptors pointing to the data buffers containing the PDU, makes a Host-only copy of the buffer descriptor, then writes the Transmit Queue entry. The SAR performs segmentation processing on the PDU and writes a Segmentation Status Queue entry informing the Host of the status of the segmentation process.

Figure 2-6. Segmentation Status Queues Related to Data Buffers and Descriptors

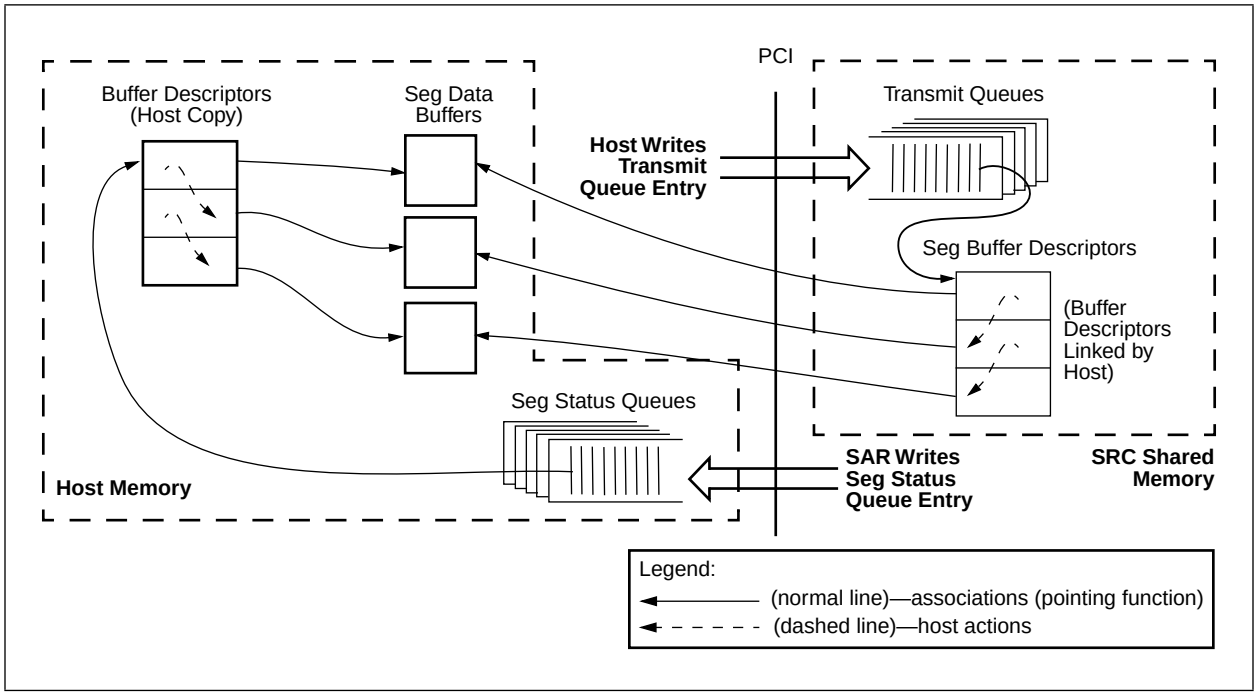
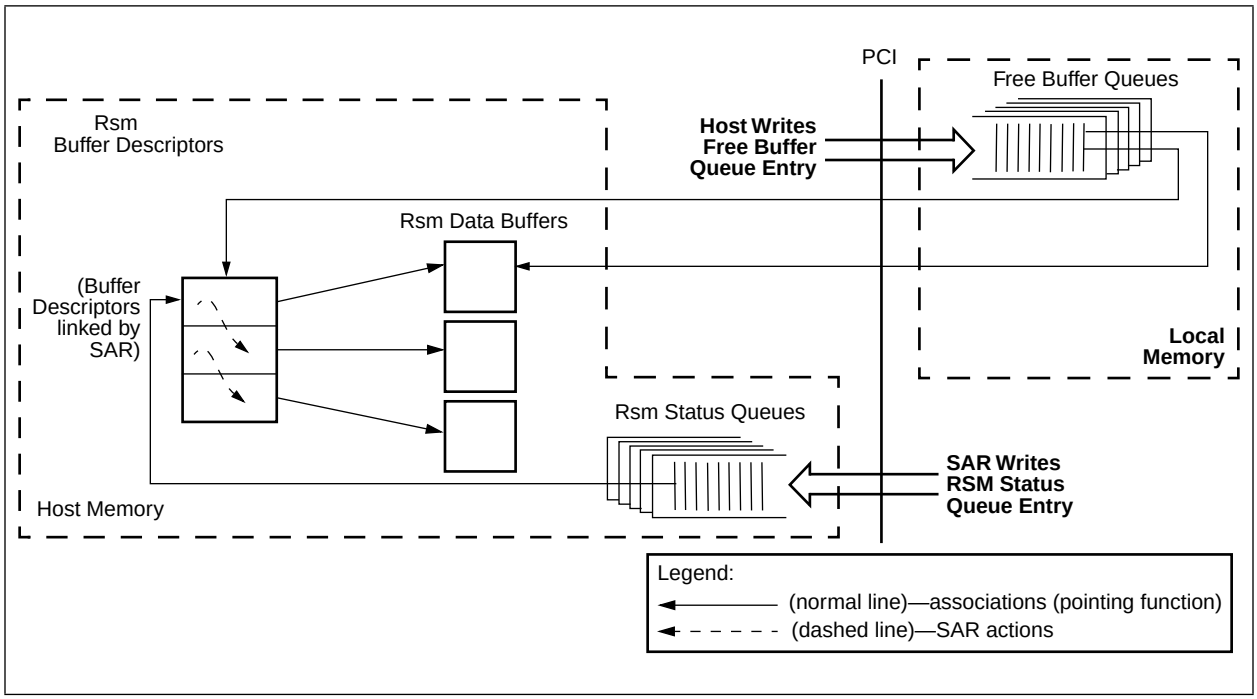


Figure 2-7 illustrates the association between the Reassembly Status Queues and reassembly data buffers and descriptors. The Host submits free buffers to the SAR by writing pointers to them in the Free Buffer Queue entries. The SAR links the buffer descriptors pointing to the three data buffers containing the reassembled PDU and writes the Rsm Status Queue entry containing the pointer to the first buffer descriptor for that PDU. The Host further processes the PDU utilizing that data.

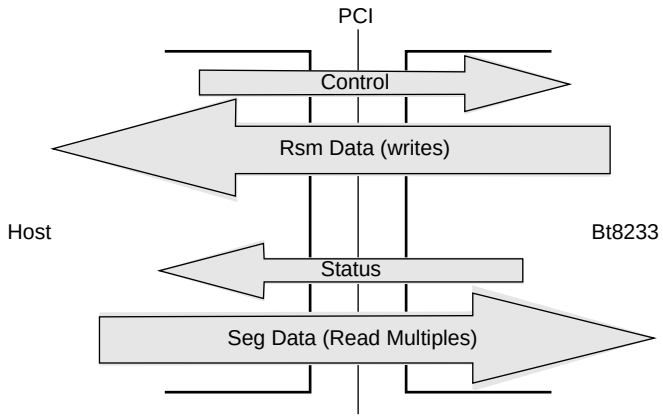
Figure 2-7. Reassembly Status Queues Related to Data Buffers and Descriptors



2.2.5 Write-Only Control/Status

The Bt8233 implements a “write-only” PCI control architecture as shown in Figure 2-8. The host manages the Bt8233 ATM terminal using write-only control and status queues. This architecture minimizes PCI bus utilization by eliminating reads from control activities. PCI writes utilize the bus much more efficiently than PCI reads. During a PCI write the Bus Master may ‘post’ the write data to an internal FIFO in the slave, terminate the transaction, and immediately release the bus. On the other hand, during PCI reads, the Bus Master retrieves the data from the slave while holding the bus. Since the data retrieval takes some time, reads increase the PCI bus utilization time for each transaction. The Bt8233 eliminates read operations except for burst reads to gather segmentation data.

Figure 2-8. Write-Only Control and Status Architecture



Note: Write-only architecture reduces PCI utilization dramatically, as reads take many more clock cycles.

2.2.6 Scatter/Gather DMA

The Bt8233’s DMA coprocessor works in close conjunction with the segmentation and reassembly coprocessors to gain access to the PCI bus, transfer the requested data, and notify the segmentation or reassembly coprocessor that the transfer is complete. The DMA coprocessor transfers all data using the read and write burst buffers in the PCI Bus Interface.

In general, two types of transactions are processed: 12– or 14– word burst accesses for data, or 1– to 4– word accesses for control and status messages.

For outgoing messages, the DMA coprocessor moves data from host memory to the segmentation coprocessor using a gather DMA method. For incoming messages, the DMA coprocessor moves data from the reassembly coprocessor to host memory using a scatter DMA method.

The DMA coprocessor is capable of handling transfers from the PCI bus with data that is not aligned on word boundaries. It will also selectively transfer data to comply with either a big endian or little endian host data structure.

2.3 Automated Segmentation Engine

The Bt8233 can segment up to 64K VCCs simultaneously. The segmentation coprocessor block independently segments each channel and multiplexes the VCCs onto the line with cell level interleaving. For each cell transmission opportunity, the xBR Traffic Manager tells the segmentation coprocessor which VCC to send.

The Bt8233 provides full support of the AAL5 protocol as well as a transparent or NULL adaptation layer, AAL0.

Each segmentation channel is specified as a single entry in the Segmentation VCC Table located in SRC shared memory. A VCC specifies a single Virtual Channel (VC) or VP in the ATM network. These VCC Table entries define the negotiated or contracted characteristics of the traffic for that channel They are initialized by the host either during system initialization or on-the-fly during operation. An initialized Segmentation VCC Table entry effectively establishes a connection on which data can be segmented.

NOTE: ABR VCCs occupy two table entries.

The Host submits data for segmentation by first linking buffer descriptors pointing to the buffers containing the PDU to be transmitted, and then submitting that chained message by writing to 1 of 32 independent circular Transmit Queues.

The segmentation coprocessor then operates autonomously, formatting the cells on each channel according to the host-defined Segmentation VCC Table entries for each channel. The segmentation coprocessor formats the ATM cell header for each cell based on the settings in the Segmentation VCC Table entry for that VCC. For AAL5 traffic, it also generates the PDU-specific fields in the trailer of the CPCS-PDU and places these in the last cell (the EOM cell) for the PDU. The formatted cells are passed through the Transmit FIFO to the PHY interface for transmission.

The system designer can set the depth of the Transmit FIFO from 1 to 9 cells deep in order to optimize the balance between Cell Delay Variation (CDV), which increases with longer Tx FIFO depth) and PCI latency protection (which decreases with shorter Tx FIFO depth.

The Bt8233 provides a method to segment traffic from a fixed PCI address (or Virtual FIFO). This is intended for circuit-based CBR traffic such as voice channel(s).

AAL0 is intended for client-proprietary use. For AAL0, the segmentation coprocessor segments the Service Data Unit (SDU) to ATM cell payload boundaries and generates ATM cell headers, but generates no other overhead fields.

The user has per-channel, per-PDU control of Raw Cell Mode segmentation, wherein the Segmentation Coprocessor reads the entire 52-octet ATM cell from the segmentation buffer and does not generate the ATM headers for the cells.

The Bt8233 reports segmentation status to the host on one of a set of 32 independent parallel Segmentation Status Queues. The Bt8233 writes a segmentation status queue entry on either PDU boundaries or buffer boundaries, selectable on a per-VCC basis. PDU boundary status reporting is called Message Mode, while buffer status reporting is called Streaming Mode.

2.4 Automated Reassembly Engine

The reassembly coprocessor processes cells received from the ATM Physical Interface Block. The coprocessor extracts the AAL SDU payload from the received cell stream and reassembles this information into buffers supplied by the host system.

Each active reassembly channel is specified as a single entry in the Reassembly VCC Table in SRC shared memory. Each entry defines the negotiated or contracted characteristics of the reassembly traffic for a particular channel, and each entry is initialized by the host during system initialization, or on-the-fly. The *Service*SAR uses the VCC Table to store temporary information to assist the reassembly process. An initialized Reassembly VCC Table entry effectively establishes a connection on which the Bt8233 can reassemble data.

Using a dynamic Channel Directory lookup method, the Bt8233 reassembles up to 64K VCCs simultaneously at a rate up to 200 Mbits/sec on simplex connections and 155 Mbits/sec on full duplex connections. The Channel Directory mechanism allows flexible preallocation of resources and provides deterministic channel identification over the full UNI or NNI VPI/VCI address space. The total number of VCCs supported is limited by the memory allocated to the VCC Table and the Channel Directory.

The reassembly coprocessor extracts the AAL SDU payload from the received cell stream and reassembles this information into system buffers allocated per-VCC. The Bt8233 supports AAL5 and AAL0 reassembly, and 52-octet raw cell mode.

For AAL5, the reassembly coprocessor extracts and checks all PDU protocol overhead.

The Bt8233 provides two methods of terminating an AAL0 PDU. PTI termination (wherein the PTI bit in the cell header is monitored for the End of Message (EOM) cell indication), and Cell Count termination (wherein the Bt8233 terminates the PDU when a user-defined number of cells have been received on that channel). The AAL0 PDU termination method is selectable on a per-VCC basis.

The user can, on a per-channel basis, establish Raw Cell Mode reassembly. In this mode the Header Error Check (HEC) octet is deleted to align the 53-octet cell to 32 bit boundaries, and the Rsm coprocessor reassembles the entire 52-octet ATM cell into the reassembly buffer.

The Bt8233 provides the user with generous per-channel control of the reassembly process, including:

- Assignment of priorities for reassembly buffer return processing.
- Cell filtering on inactive channels.
- Mechanisms to establish per-VCC firewalling by allocating buffer credits on a per-channel basis. This limits the possibility of one VCC consuming all of the memory resources.
- Per-VCC activation and control of a background hardware time-out function where the user selects 1 of 8 programmable time-out periods -- the background function then automatically detects partially reassembled PDUs and reports this status to the host so that these buffers can be recovered and re-allocated.
- Per-VCC monitoring of the length of the reassembled PDU, with status reporting if the length exceeds a set maximum length for that channel.

The Bt8233 implements an Early Packet Discard (EPD) feature to enable discarding of complete or partial CPCS-PDUs based upon service discard attributes or error conditions. The EPD function halts reassembly of the CPCS-PDU marked for discard until the next BOM cell and/or the error condition has cleared. The SAR writes a status queue entry with the appropriate status flags set, which indicate the reason for the discard. This function may be enabled for the following conditions:

- Frame Relay discard based on the frame's DE setting and the channel exceeding a user-defined priority threshold.
- CLP packet discard based on the received cell's CLP setting and exceeding channel priority threshold.
- LANE-LECID packet discard on ELAN channels, which implements echo suppression on multicast data frames.
- EPD on AAL5 channels when the reassembled PDU length exceeds the user-defined maximum PDU length for that VCC.
- EPD on channels encountering a free buffer queue empty (underflow) condition (meaning there are no available buffers in the free buffer queue that channel is assigned to).
- EPD on PDUs when a DMA Incoming FIFO Full condition occurs.
- EPD on channels encountering a reassembly status queue full (overflow) condition.

The system designer can set the reassembly status reporting for any channel to either Message Mode or Streaming Mode. In Message Mode, a status entry is written only when the last buffer in a message completes reassembly. In Streaming Mode, a status entry is written for each buffer as it completes reassembly.

2.5 Advanced xBR Traffic Management

The Bt8233 implements ATM's inherent robust traffic management capabilities for CBR, VBR, ABR, UBR, and GFC. The Bt8233 manages each VCC independently and dynamically. The user assigns each connection a service class, a priority level, and a rate if applicable. Then, the on-chip traffic controller, the xBR Traffic Manager optimizes usage of the line bandwidth according to the VCC's traffic parameters and control information stored in SRC shared memory. The xBR Traffic Manager guarantees the compliance of each VCC to its service contract with the ATM network at the UNI ingress point. It schedules all data traffic by acting as a master to the segmentation coprocessor.

One of the functional components of the xBR Traffic Manager is the xBR Scheduler. The xBR Traffic Manager assigns segmentation traffic from active VCCs to schedule slots, which the segmentation coprocessor then complies to by segmenting VCC traffic in the sequence/schedule dictated by the xBR Scheduler.

In addition to preemptive, true CBR, the Bt8233 provides eight segmentation priorities. The user configures these priorities for the remaining service categories, including the TM4.0 defined ABR class.

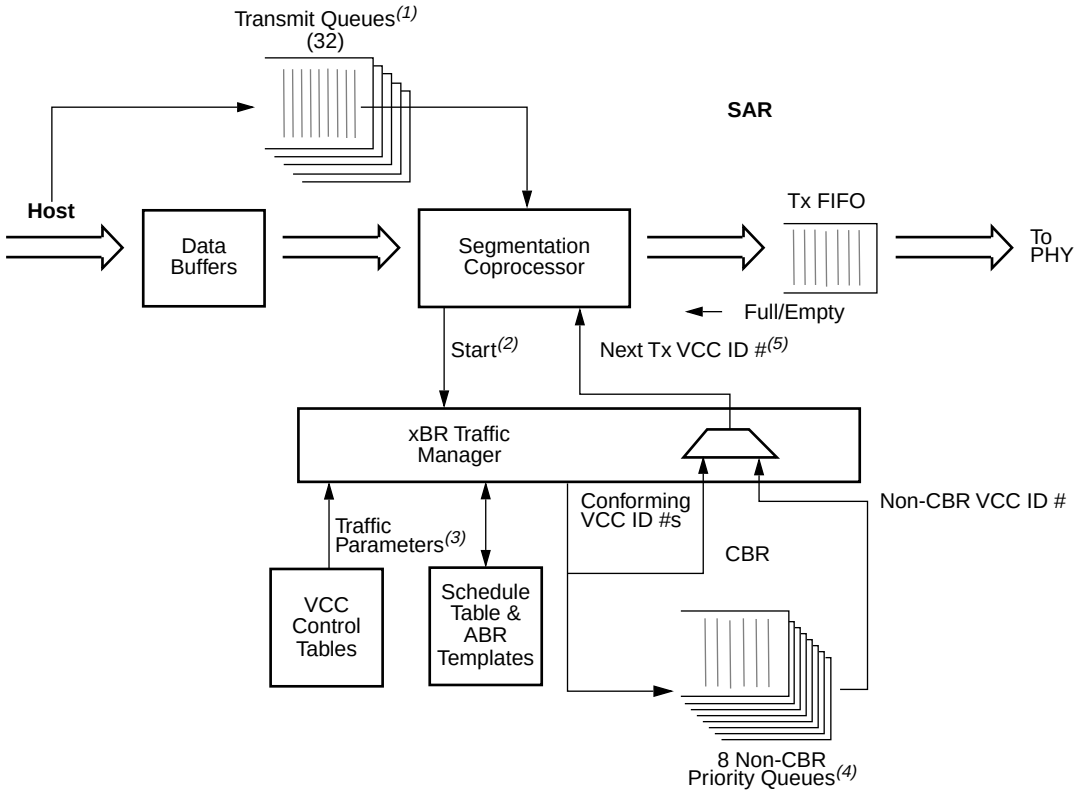
The Bt8233's xBR Traffic Manager implements three functional levels of traffic prioritizing. This is illustrated in Figure 2-9.

The host submits data to be sent by writing entries to the Transmit Queues for segmentation. The SAR processes these Transmit Queues either in round robin order (Transmit Queue 0 through 31),or in priority order (with Transmit Queue 31 having highest priority).

The Bt8233 then submits this traffic demand to the xBR Traffic Manager for scheduling. Traffic is asynchronously multiplexed based on the traffic class plus certain parameters from the Segmentation VCC Table entries (primarily the GCRA *I* and *L* parameters).

The conforming traffic to be transmitted is further groomed in internal priority queues. Each virtual channel is prioritized according to its assigned scheduling priority. CBR channels are given pre-assigned segmentation bandwidth, and channels for the remaining service categories scheduled according to their priority number (0 being the lowest priority and 7 being highest). This prioritization is shown in Figure 2-9 as a global set of queues, but is actually maintained on a per transmit opportunity basis. In this way, high priority traffic will be transmitted up to its GCRA limits, but will not block lower priority traffic when idle.

Figure 2-9. Multi-Level Model for Prioritizing Segmentation Traffic



Notes: (1). User selects priority or round robin service policy on Transmit Queues.
(2). The segmentation coprocessor notifies the Traffic Manager when a VCC becomes active, through reading new entries from the Transmit Queue.
(3). The user initializes traffic parameters on a per-VCC basis.
(4). Each conforming non-CBR VCC is assigned to a priority queue, while conforming CBR VCCs are simply slated for transmit.
(5). The highest priority conforming cell is transmitted.

2.5.1 CBR Traffic

The CBR service category requires guaranteed transmission rates as well as constrained CDV. The Bt8233 facilitates these needs when generating CBR traffic by pre-assigning specific schedule slots to CBR VCCs. For each CBR assigned cell slot, the Bt8233 generates a cell for that specific VCC unless data is not available.

The Bt8233 minimizes CDV by basing all traffic management on a local reference clock.

The Bt8233 also provides a mechanism to exactly match the scheduled rate of a CBR channel to the rate of its data source. The host may occasionally instruct the xBR Scheduler to skip one transmit opportunity on a channel in order to accomplish this rate-matching.

The Bt8233 manages CBR tunnels in the same manner as a CBR VCC. However, instead of one VCC, several UBR VCCs can be scheduled within this CBR tunnel, in round robin order.

Unused CBR and Tunnel timeslots are automatically made available to VCCs of other service categories by the xBR Scheduler.

2.5.2 VBR Traffic

The Bt8233 takes advantage of the asynchronous nature of ATM by reserving bandwidth for VBR channels at average cell transmission rates without pre-assigning ‘hardcoded’ schedule slots as with CBR traffic. This dynamic scheduling allows VBR traffic to be statistically multiplexed onto the ATM line, resulting in better utilization of the shared bandwidth resources. The xBR Scheduler supports multiple priority levels for VBR traffic. Through the combination of VBR parameters and priorities, it is possible to support real-time VBR services.

The outgoing cell stream for each VBR VCC is scheduled according to the GCRA algorithm. The GCRA *I* and *L* parameters control the per-VCC Peak Cell Rate (PCR) and Cell Delay Variation Tolerance (CDVT) of the outgoing cell stream on any channel. This guarantees compliance to policing algorithms applied at the network ingress point. The user can control the granularity of rate by dictating the number of schedule slots in the schedule table.

Channels may be rate-shaped as VCs or VPs according to one of three VBR definitions. VBR1 controls PCR and CDVT. VBR2 controls PCR and CDVT, and as well SCR. VBRC controls PCR and CDVT on all cells but controls SCR on only CLP=0 (i.e., high priority) cells.

2.5.3 ABR Traffic

The Bt8233 implements the ATM Forum ABR flow control algorithms. The Bt8233 acts as a fully compliant ABR Source and Destination as defined in the TM4.0 specification. The ABR service category effectively allows low cell loss transmission through the ATM network by regulating transmission based upon network feedback. The ABR algorithms regulate the rate of each VCC independently.

The Bt8233 employs an internal feedback control loop mechanism to enable the TM4.0 ATM source specification. The SAR utilizes the dynamic rate adjustment capability of the xBR Scheduler as the ATM Source's variable traffic rate shaper. The Bt8233 injects an in-rate stream of Forward RM cells for each ABR VCC. When these cells return to the Bt8233's receive port as Backward RM cells after a round trip through the network, the Bt8233 processes these cells and uses the data returned as feedback to dynamically adjust the rates on each ABR channel.

The Bt8233 also responds to an incoming ABR cell stream as an ABR Destination. The reassembly coprocessor processes received Forward RM cells. It turns around this incoming information to the segmentation coprocessor, which formats Backward RM cells containing this information and inserts these turnaround RM cells into the transmit cell stream.

The exact performance of the rate-shaper is governed by one or more ABR Templates in local memory. Each VCC is assigned to one of these templates. The templates control such behaviors as the size of the additive rate increase factors or multiplicative rate decrease steps. Each VCC's rate varies across the template independently.

2.5.4 UBR Traffic

The UBR service category is intended for non-real-time applications which do not require tightly constrained delay and delay variation, such as traditional computer communications applications like file transfer and e-mail.

Those VCCs which have not been assigned to one of the other service categories as covered above are scheduled as UBR traffic. All UBR channels within a priority are scheduled on a round-robin basis. To limit the bandwidth that a UBR priority consumes, the system designer should use a CBR tunnel in that priority level.

2.5.5 xBR Cell Scheduler

The xBR Scheduler schedules traffic according to a Dynamic Schedule Table maintained in SRC shared memory. The table contains a user-programmable number of schedule slots. The duration of a single slot is a user-programmable number of system clock cycles. The xBR Scheduler sequences through this table in a circular fashion to schedule traffic. By configuring the number of slots in the table and the duration of each slot, the system designer chooses a range of available rates. A specific rate for any channel is determined by how many slots in the table that channel is assigned. Schedule slots not reserved for CBR during table setup are used for the rest of the service categories.

The xBR Scheduler implements Rockwell's proprietary per-VCC rate shaping algorithms. The predecessor to the Bt8233, the Bt8230 SRC, proved the core algorithms. The Bt8233 extends their use to other service classes. The Bt8233 xBR Scheduler shapes all traffic classes, including CBR, single leaky bucket VBR, dual leaky bucket VBR, ABR and UBR. The host configures the Dynamic Schedule Table during system initialization, defining the table size in number of schedule slots and the length of each schedule slot in clock cycles. After setup, the Bt8233 dynamically manages the entire table.

The key features of the xBR Scheduler are:

1. Per-VCC rate control guarantees conformance to GCRA UPC/policing.
2. Dynamic reallocation of link bandwidth to active channels.
3. Dynamic, fair sharing of bandwidth on oversubscribed lines.
4. Multiple scheduling priorities.
5. Fine grained rate control.
6. Rate based on a user supplied reference clock.

Dynamic management provides on-the-fly reallocation of link bandwidth without host intervention. The Bt8233 fairly distributes the link bandwidth to channels based upon their QoS parameters and assigned transmission priority. As covered previously, the Bt8233 supports eight priorities in addition to preallocated CBR time slots.

The xBR Scheduler facilitates advanced network traffic management topologies. The Bt8233 rate shapes VCs or VPs. Additionally, CBR tunneling allows UBR or VBR traffic management schemes to operate under a preallocated CBR limit.

2.5.6 ABR Flow Control Manager

The ABR Flow Control Manager operates in conjunction with the xBR Scheduler to control the rate of ABR channels. The Bt8233 implements the TM4.0 specification in a template-controlled hardware state machine. Rockwell provides an initial set of templates which reside in SRC shared memory. The information within these templates defines conformant ABR behavioral responses to network and connection states. The Bt8233 generates ABR Source traffic, including internally generated RM cells, according to the template instructions. The reassembly coprocessor and the flow control manager collaboratively act as a fully compliant ABR destination terminal.

These templates provide three significant benefits to the user:

- First, since they control the flow control manager state machine, they can be optimized for specific applications.
- Second, the programmability of the templates insulates the hardware from changes in the relatively stable yet immature TM4.0 specification.
- Third, Rockwell provides the initial templates, which may be customized by the user later, shortening development time.

2.6 Generous Implementation of OAM-PM Protocols

The Bt8233 provides internal support for the detection and generation of OAM traffic, including PM-OAM.

The Bt8233 supports the F4 and F5 OAM flows according to I.610. It monitors up to 128 channels and generates in-rate PM-OAM cells.

The Bt8233 includes a Local Processor Interface, providing the capability of SRC shared memory segmentation and reassembly. The user can thus route OAM traffic, including PM traffic, to and from this optional local processor, thereby offloading ATM network management from the host. To facilitate this, the Bt8233 provides the option of user-defined global status queues for both segmentation and reassembly and a global buffer queue for reassembly, to which the user can assign SRC shared memory addresses. The Bt8233 will then process OAM traffic via the SRC shared processor. This will isolate the host from these management functions and focus host processing power on ATM user data traffic.

2.7 Standards-Based I/O

2.7.1 PCI Bus I/O

The PCI bus interface implements the full set of address, data, and control signals required to drive the PCI bus as a master. Additionally, the interface contains the logic required to support arbitration for the PCI bus. This interface is PCI Version 2.0 compliant. Future versions of the device will be PCI Version 2.1 compliant.

2.7.2 ATM PHY I/O

The Bt8233's ATM physical interface communicates with and controls the ATM link interface chip, which carries out all the transmission convergence and physical media-dependent functions defined by the ATM protocol. Three modes of operation are provided: standard UTOPIA, slave UTOPIA, and Bt8222 Cell Mux. Standard UTOPIA mode conforms to the UTOPIA Level 1 standard for an ATM Layer device. Slave UTOPIA mode reverses the control direction for use in place of a PHY on switch fabrics. Bt8222 Cell Mux mode is a FIFO interface, and allows the Bt8233 to interface in a glueless connection to the Bt8222 PHY chip. Using this mode, the Bt8233 can share the Bt8222 PHY in parallel with up to three other entities. This architecture enables the concentration of Bt8233-generated traffic with other cell-based traffic such as AAL1 Circuit Emulation.

2.7.3 SRC Shared Memory I/O

To simplify system implementations, the Bt8233 integrates a complete memory controller designed for direct interface to common Static RAMs (SRAMs). The Bt8233's memory controller operates at 33 MHz and can access up to 8 Mbytes of SRAM memory. The memory controller also arbitrates access to the internal control and status registers by the host and local processors. The memory banks can be configured to a variable number of sizes. All of this affords a wide degree of flexibility in SRC shared memory architecture.

2.7.4 Local Processor I/O

The Local Processor Interface in the Bt8233 allows an optional external CPU to be directly connected to the device to serve as a local controlling intelligence that can handle initialization, connection management, overall data management, error recovery, and OAM functions. The use of a local processor for these functions allows ATM message data to flow to and from host system memory in a substantially larger bandwidth. This is because as the local processor is handling the out of band functions described above.

The processor interface is loosely coupled, meaning that the processor connects to the Bt8233 through bidirectional transceivers and buffers for the address and data buses. This allows the processor fast access to B8233 memory and registers, but insulates the Bt8233 from processor instruction and data cache fills. It also allows the processor to control multiple Bt8233s or physical devices if desired.

2.7.5 Boundary Scan I/O and Loopbacks

The Bt8233 includes five pins for JTAG Boundary Scan, for board-level testing. The Bt8233 incorporates an internal loopback from the segmentation coprocessor to the reassembly coprocessor, to facilitate system diagnostics.

2.8 Electrical/Mechanical

The Bt8233 is a CMOS device packaged in a 208-pin Plastic Quad Flat Pack (PQFP). The Bt8233 operates within the standard industrial temperature range, and consumes 1.5 W typical at full processing rate. The device is designed to operate with TTL level inputs and CMOS level outputs.

2.9 Bt8230 Compatibility

The Bt8233 is fully pin-compatible with its predecessor, the Bt8230. This allows an easy upgrade path from a UNI 3.1 compliant system implemented with the Bt8230 to a more advanced system meeting the additional requirements of TM 4.0.

2.10 Logic Diagram and Pin Descriptions

A functionally partitioned logic diagram of the Bt8233 is shown in Figure 2-10. Pin descriptions, names and input/output assignments are detailed in Table 2-1.

Figure 2-10. Bt8233 Logic Diagram

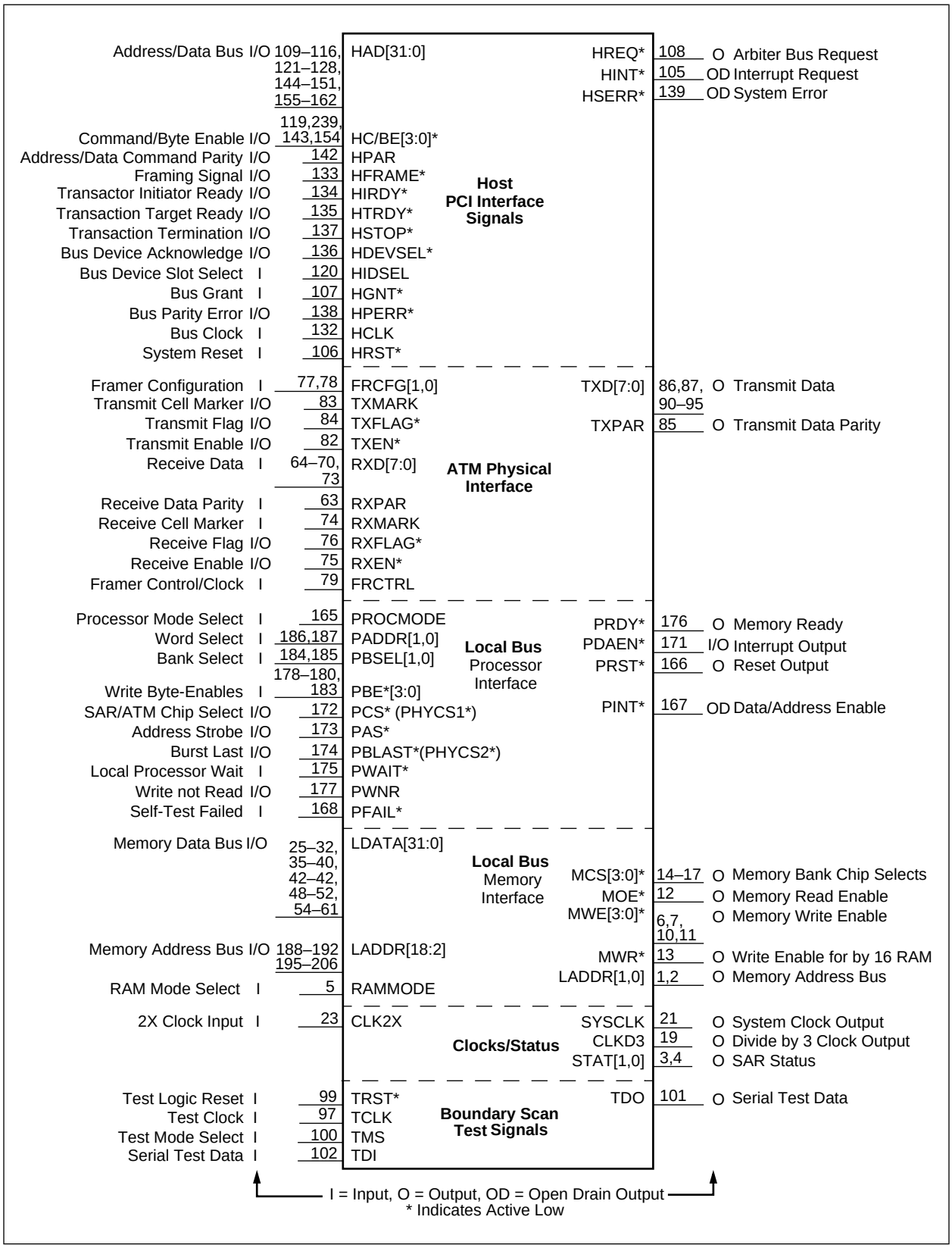


Table 2-1. Hardware Signal Definitions *(1 of 6)*

	Pin Label	Signal Name	I/O	Definition
Host PCI Interface Signals	HAD[31:0]	Multiplexed Address/Data Bus	I/O	Used by the PCI host or Bt8233 to transfer addresses or data over the PCI bus.
	HC/BE[3:0]*	Command/Byte Enable	I/O	Outputs a command (during PCI address phases) or byte enables (during data phases) for each bus transaction.
	HPAR	Address/Data Command Parity	I/O	Supplies the even parity computed over the HAD[31:0] and HC/BE[3:0]* lines during valid data phases; it is sampled (when the Bt8233 is acting as a target) or driven (when the Bt8233 acts as an initiator) one clock edge after the respective data phase.
	HFRAME*	Framing Signal	I/O	A high-to-low HFRAME* transition indicates that a new transaction is beginning (with an address phase). A low-to-high transition indicates that the next valid data phase will end the current transaction.
	HIRDY*	Transaction Initiator Ready	I/O	Used by the transaction initiator or bus master (either the Bt8233 or the PCI host) to indicate ready for data transfer. A valid data phase ends when both HIRDY* and HTRDY* are sampled asserted on the same clock edge.
	HTRDY*	Transaction Target Ready	I/O	Used by the transaction target or bus slave (either the Bt8233 or the PCI bus memory) to indicate that it is ready for a data transfer. A valid data phase ends when both HIRDY* and HTRDY* are sampled asserted on the same clock edge.
	HSTOP*	Transaction Termination	I/O	Driven by the current target or slave (either the Bt8233 or the PCI bus memory) to abort, disconnect, or retry the current transfer. The HSTOP* line is used by the PCI master in conjunction with the HTRDY* and HDEVSEL* lines to determine the type of transaction termination.
	HDEVSEL*	Bus Device Acknowledge	I/O	Driven by a target to indicate to the initiator that the address placed on the HAD[31:0] lines (together with the command on the HC/BE[3:0]* lines) has been decoded and accepted as a valid reference to the target's address space. Once asserted, it is held by the Bt8233 (when acting as a slave) until HFRAME* is deasserted; otherwise, it indicates (in conjunction with HSTOP* and HTRDY*) a target abort.
	HIDSEL	Bus Device Slot Select	I	Signals the Bt8233 that it is being selected for a configuration space access.
	HREQ*	Arbiter Bus Request	O	Asserted by the Bt8233 to request control of the PCI bus.
	HGNT*	Bus Grant	I	Asserted to indicate to the Bt8233 that it has been granted control of the PCI bus, and may begin driving the address/data and control lines after the current transaction has ended (indicated by HFRAME*, HIRDY* and HTRDY* all deasserted simultaneously).

Table 2-1. Hardware Signal Definitions *(2 of 6)*

	Pin Label	Signal Name	I/O	Definition
PCI Interface Signals	HINT*	Interrupt Request	OD	Signals an interrupt request to the PCI host and is tied to the INTA* line on the PCI bus.
	HPERR*	Bus Parity Error	I/O	Driven asserted by the Bt8233 (as a bus slave) or by a target addressed by the Bt8233 when it acts as a bus master to indicate a parity error on the HAD[31:0] and HC/BE[3:0]* lines. It is asserted when the Bt8233 is a bus slave or sampled when the Bt8233 is a bus master on the second clock edge after a valid data phase. The Bt8233 drives the HPERR* line only when acting as a slave.
	HSERR*	System Error	OD	Indicates a system error or a parity error on the HAD[31:0] and HC/BE[3:0]* lines during an address phase. This pin is handled in the same way as HPERR*, and is only driven by the Bt8233 when it acts as a bus slave.
	HCLK	Bus Clock	I	Supplies the PCI bus clock signal.
	HRST*	System Reset	I	Performs a hardware reset of the Bt8233 and associated peripherals when asserted. Must be asserted for 16 cycles of HCLK.
ATM Physical Interface	FRCFG[1,0]	Framer Configuration	I	Configuration pins FRCFG[1,0] determine what framer interface the Bt8233 supports. 00 = Bt8222 Cell Mux interface 01 = UTOPIA interface 10 = Slave UTOPIA interface 11 = Reserved, do not use
	TXD[7:0]	Transmit Data	O	Carries outgoing data bytes to the framer chip in all framer modes.
	TXPAR	Transmit Data Parity	O	Outputs the 8-bit odd parity computed over the TXD[7:0] lines in all framer modes.
	TXMARK	Transmit Cell Marker	I/O	In both UTOPIA and slave UTOPIA modes, the TXMARK line is asserted by the Bt8233 when the starting byte of a 53-byte cell is being output. In Bt8222 Cell Mux mode, this pin is asserted by the framer to indicate that it is expecting the starting byte of a 53-byte ATM cell to be transferred on the TXD[7:0] lines as the next valid data byte.
	TXFLAG*	Transmit Flag	I/O	In UTOPIA mode, TXFLAG* indicates that the transmit buffer in the downstream link interface chip is full, and no more data can be accepted. In slave UTOPIA mode, this pin indicates to the link interface chip that the Bt8233 transmit buffer is empty. In Bt8222 Cell Mux mode, TXFLAG* indicates that no more data is available in the transmit buffer of the Bt8233.
	TXEN*	Transmit Enable	I/O	Indicates that valid data has been placed on the TXD[7:0], TXPAR, and TXMARK lines in the current clock cycle when the Bt8233 is in UTOPIA or slave UTOPIA mode. This pin is an output in UTOPIA mode and an input in slave UTOPIA mode. In Bt8222 Cell Mux mode, the TXEN* input indicates that the downstream framer device is requesting the Bt8233 to transfer a byte of data on the TXD[7:0] lines.
	RXD[7:0]	Receive Data	I	Transfers incoming data bytes from the link interface or framer chip to the Bt8233 in all framer modes.

Table 2-1. Hardware Signal Definitions *(3 of 6)*

	Pin Label	Signal Name	I/O	Definition
ATM Physical Interface	RXPAR	Receive Data Parity	I	Should be driven with the 8-bit odd parity computed over the RXD[7:0] lines by the link interface or framer chip in all framer modes.
	RXMARK	Receive Cell Marker	I	Indicates that the current byte being transferred on the RXD[7:0] lines is the starting byte of a 53-byte cell.
	RXFLAG*	Receive Flag	I/O	In UTOPIA mode, RXFLAG* indicates that the receive buffer in the downstream link interface chip is empty, no more data can be transferred, and the RXD[7:0], RXPAR, and RXMARK lines are invalid. In slave UTOPIA mode, this pin indicates to the framer chip that the receive FIFO in the Bt8233 is full. In Bt8222 Cell Mux mode, the RXFLAG* output indicates that the Bt8233 internal FIFO buffer is full and any subsequent attempts to transfer data will be ignored.
	RXEN*	Receive Enable	I/O	In UTOPIA mode, RXEN* indicates that the Bt8233 is ready to receive data on the RXD[7:0], RXPAR, and RXMARK lines in the next clock cycle. This pin is an output in UTOPIA mode, and an input in slave UTOPIA mode. In Bt8222 Cell Mux mode, the RXEN* input is driven active by the framer to signify that valid data has been presented on the RXD[7:0], RXPAR, FRCTRL, and RXMARK lines.
	FRCTRL	Framer Control/Clock	I	In UTOPIA and slave UTOPIA mode, the FRCTRL line should be driven with a clock that is synchronous to that used by the framer device for interfacing to the Bt8233. The TXD[7:0], TXPAR, TXMARK, TXFLAG* TXEN*, RXD[7:0], RXPAR, RXMARK, RXFLAG*, and RXEN* lines must be synchronous to this clock in UTOPIA mode, and maintain the specified setup and hold times with reference to its rising edge. In Bt8222 Cell Mux mode, this pin marks a cell as invalid, and the Bt8233 discards the cell.
Local Processor Interface	PROCMODE	Processor Mode Select	I	When grounded, this input selects the local processor mode. When pulled to a logic high, the stand-alone mode is selected.
	PADDR[1,0]	Word Select Inputs	I	The PADDR[1,0] inputs are connected to the word select field of the CPU address bus (address bits [3, 2] for the Intel i80960CA processor, which can perform 4-word burst transactions). These inputs are used by the Bt8233 to allow single-cycle bursts to be performed without requiring very short memory access times.
	PBSEL[1,0]	Bank Select Inputs	I	Select one of four banks of memory to be accessed. They are decoded by the memory controller to generate the appropriate chip/bank selects to the external memory.

Table 2-1. Hardware Signal Definitions *(4 of 6)*

	Pin Label	Signal Name	I/O	Definition
Local Processor Interface	PBE[3:0]*	Write Byte-Enables	I	Supply byte enables for each local processor memory access. These pins are only relevant during writes by the local processor to SRC shared. Each byte enable line corresponds to a specific byte lane in the LDATA[31:0] data bus: PBE[0]* corresponds to LDATA[7:0], PBE[1]* to LDATA[15:8], PBE[2]* to LDATA[23:16], and PBE[3]* to LDATA[31:24].
	PCS* (PHYCS1*)	SAR Chip Select ATM PHY Chip Select (in standalone mode)	I/O	In local processor mode with PROCMODE tied low, PCS* is the SAR chip select input. In stand-alone mode, this pin is PHYCS1*, which may be connected to the chip select input of the Bt8222 PHY device.
	PAS*	Address Strobe	I/O	Indicates a local processor address cycle. In stand-alone mode, PAS* is used to drive the AS* pin of the Bt8222 device.
	PWNR	Write/not Read	I/O	The PWNR input indicates the direction of a local processor transfer. A logic I indicates a write while a logic 0 indicates a read. During stand-alone mode, this output provides the same function for the Bt8222.
	PWAIT*	Local Processor Wait	I	Used by the local processor or external logic to insert wait states for read or write transactions.
	PBLAST* (PHYCS2*)	Burst Last ATM PHY Chip Select (in stand-alone mode)	I/O	In local processor mode, this input is used by the processor to indicate the end of a transaction. During stand-alone mode, this output is a second chip select, PHYCS2*.
	PRDY*	Memory Ready	O	Signals that the memory or control register has accepted the data on a write, or that data is available to latch by the local processor on a read cycle.
	PDAEN*	Interrupt Output	I/O	Connected to the output enable input of the bidirectional transceivers and buffers used to isolate the Bt8233 data and address bus from the local processor. In standalone mode, this input is connected to the physical device's interrupt output(s).
	PFAIL*	Self-Test Failed	I	The local processor can indicate a failure of its internal self-test or initialization processes by asserting the PFAIL* input to the Bt8233.
	PINT*	Data/Address Enable	OD	Asserted by the Bt8233 to the local processor to signal an interrupt request in local processor mode.
	PRST*	Reset Output	O	Asserted by the Bt8233 to the local processor whenever the HRST* input is asserted, or when the LP_ENABLE bit in the CONFIG0 register is a logic low.

Table 2-1. Hardware Signal Definitions *(5 of 6)*

	Pin Label	Signal Name	I/O	Definition
SRC shared memory Interface	LDATA[31:0]	Memory Data Bus	I/O	Data I/O bus. Used for memory reads and writes as well as Control and Status Register access by the local processor.
	LADDR[18:2]	Memory Address Bus	I/O	Address I/O bus. Used for memory reads and writes as well as Control and Status Register access by the local processor.
	LADDR[1,0]	Memory Address Bus	O	The two least significant bits of address I/O bus. Used for memory reads and writes as well as CSR access by the local processor.
	MCS[3:0]*	Memory Bank Chip Selects	O	Selects one of four addressable banks of SRAM memory.
	MOE*	Memory Read Enable	O	Indicates that a read cycle is proceeding and the memory device output buffers should be enabled, driving data onto the LDATA[31:0] lines.
	MWE[3:0]*	Memory Write Enable	O	Memory byte write enables for by_4 or by_8 SRAMs. For by_16 devices, these outputs are byte enables that are active on writes and reads.
	MWR*	Enable for 16 RAM Write	O	Memory write enable for by_16 SRAMs.
	RAMMODE	RAM Mode Select	I	Selects RAM chips supported. 1 = by_16 memory devices. 0 = by_4 or by_8 memory devices.
Clocks and Status	CLK2X	2x Clock Input	I	Double frequency (from SYSCLK) TTL clock input. (66 MHz max)
	SYSCLK	System Clock Output	O	This divide by 2 of CLK2X is the internal system clock as well as the external system clock. (33 MHz maximum)
	CLKD3	Divide by 3 Clock Output	O	This output clock is a 50% duty cycle one-third divide of CLK2X; it may be used for the UTOPIA interface clock. (22 MHz max)
	STAT[1,0]	SAR Status	O	Bt8233 internal status outputs. Internal status controlled by the STAT_MODE[4:0] field in the CONFIG0 Register. (ADD IN FINAL DRAFT)

Table 2-1. Hardware Signal Definitions *(6 of 6)*

	Pin Label	Signal Name	I/O	Definition
Boundary Scan Test Signals	TRST*	Test Logic Reset	I	When a logic low, this signal asynchronously resets the boundary scan test circuitry and puts the test controller into the reset state. This state allows normal system operation. Tie to ground when boundary scan is not implemented
	TCLK	Test Clock	I	Generated externally by the system board or by the tester. Tie to ground when boundary scan is not implemented.
	TMS	Test Mode Select	I	Decoded to control test operations.
	TDO	Serial Test Data	O	Outputs serial test pattern data.
	TDI	Serial Test Data	I	Input for serial test pattern data.
Supply Voltage	PWR	Power	–	Sixteen pins are provided for supply voltage.
	GND	Ground	–	Twenty-two pins are provided for ground.

3.0 Host Interface

3.1 Overview

The Bt8233 SAR segments and reassembles user data packets at over 200 Mbits/second simplex, and over 155 Mbits/second full duplex. The actual segmentation and reassembly processes execute without run-time host control. However, the ATM host system supplies the data for transmission and buffers for received data. In addition to this control, the host processes status returned from the SAR. To take advantage of the Bt8233's high throughput, the host must process control and status information at a comparable rate.

NOTE: Application Example—An Ethernet Switch uses a Bt8233 based subsystem as an uplink to an OC–3 ATM backbone. Under worst case conditions, Ethernet packets (64 octets) map into two ATM cells. At OC–3 rates, the Bt8233 converts 176.6K packets/second to cells in each direction. Therefore, the host must process control and status information for a total of 353.2K packets/second. This packet rate equates to a packet service time of 2.83 microseconds/packet.

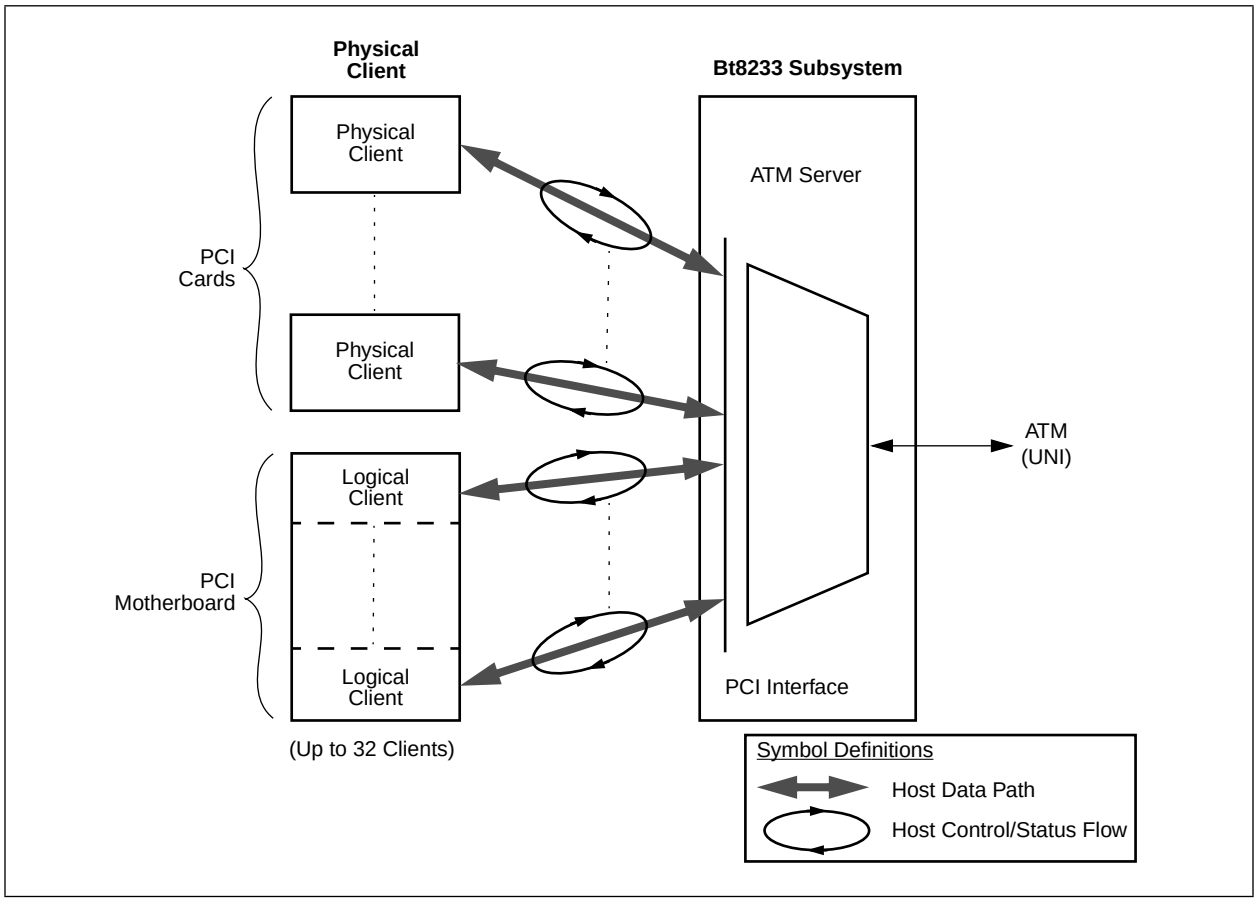
In cases such as the example above, the service rate places extraordinary performance requirements on the host system. High throughput systems require large numbers of processing cycles and efficient utilization of system buses.

The Bt8233 provides a flexible, high performance host interface architecture. With this interface, the Bt8233 facilitates a scalable, distributed host system. The interface also minimizes the impact of an ATM port on the host system's PCI bus.

3.2 Multiple Client Architecture

The Bt8233 provides multiple independent control and status communication paths. Each communication path, or flow, consists of a control queue and a status queue for both segmentation and reassembly. The host assigns each of these independent flows to system clients or peers. As throughput requirements escalate, the host system can add processing power in the form of additional peers. This degree of freedom creates a scalable host environment. Figure 3-1 illustrates this client/server relationship. The Bt8233 provides an ATM server for up to 32 clients.

Figure 3-1. Client/Server Model of the Bt8233



3.2.1 Logical Clients

As shown in Figure 3-1, the clients need not be physically distinct PCI peers. The host can also assign control and status queues to system software tasks, or logical clients. Since the queues offer individually distinct communication paths, each logical client interfaces to the Bt8233 independently. Due to its server architecture, the Bt8233 supplies the synchronization between asynchronous tasks requiring ATM services.

3.2.2 Resource Allocation

With either physical PCI peers, logical peers, or some combination of the two types, the Bt8233 multiplexes each peer's transmitted packets onto the line and routes incoming packets to the appropriate peer. The host system allocates shared resources such as host and SRC shared memory, VPI/VCI address space, and CBR time slots to peers and clients arbitrarily.

3.2.3 Resource Isolation

Since each peer is assigned to an independent control and status path, the Bt8233 isolates the resources of each peer. Not only does this simplify resource management, but queue error conditions caused by a single peer do not affect any other peers.

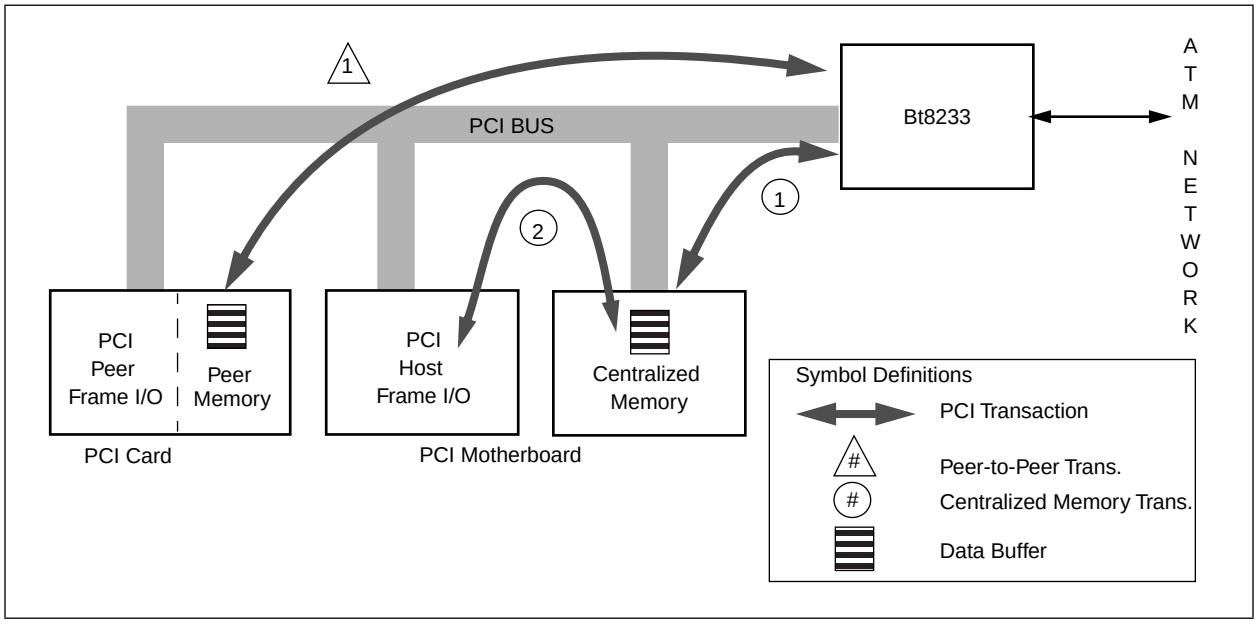
NOTE: Application Example—A system designer implements a Bt8233 terminal as an ATM uplink for a Service Access Multiplexer (SAM). The SAM is comprised of the ATM card and several Frame Relay adapter cards. The host assigns each Frame Relay adapter card to a set of Bt8233 control and status queues at initialization. During operation, one of the Frame Relay adapter cards experiences a hardware failure. The failure prevents the card's processor from servicing the Bt8233's Reassembly Status Queue. Eventually, the Bt8233 fills the queue and is unable to proceed -- this situation is referred to as queue overflow. The Bt8233 shuts down reassembly on VCCs which are assigned to the overflowed queue ONLY. Since the other cards in the system are assigned to other status queues, their VCCs remain unaffected by the failure.

3.2.4 Peer-to-Peer Transfers

The multiple queue architecture of the Bt8233 also enables peer-to-peer PCI transfers. The Bt8233 transfers ATM cells as a PCI master. Since the buffer control structures are independent for each peer, each identifies a unique address range in PCI memory space. The host defines the address range of each peer. The Bt8233 transfers data within this address range. An address range corresponds either to a region of centralized host memory, or to a set of peer resident buffers. Figure 3-2 shows the difference between these two options.

NOTE: Centralized memory buffers require store-forward operations, while the peer buffers enable peer-to-peer transfers. Thus, peer-to-peer transfers reduce the utilization of the PCI bus.

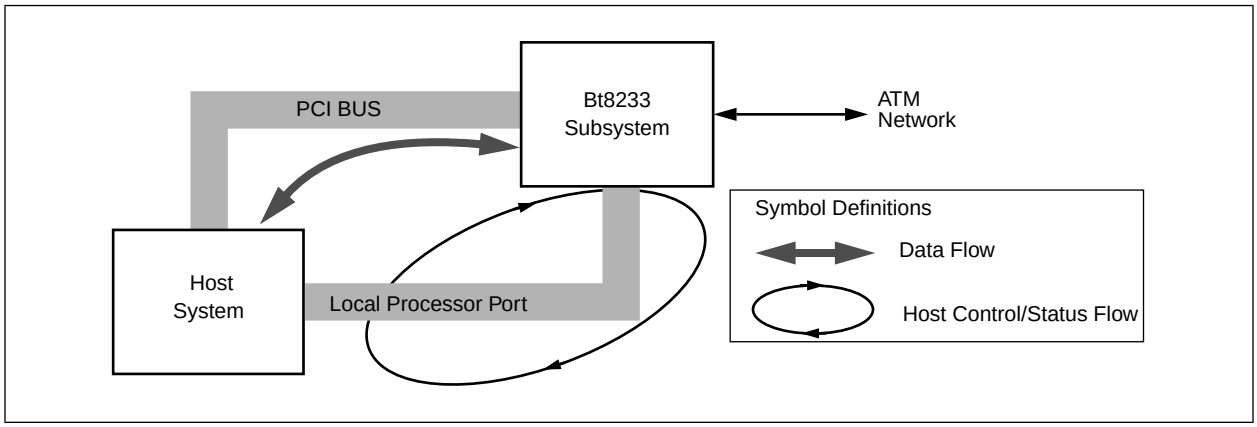
Figure 3-2. Peer-to-Peer vs. Centralized Memory Data Transfers



3.2.5 Local Processor Clients

The Bt8233 supports limited bandwidth SRC shared memory segmentation and reassembly. Any peer may use the local processor port instead of the PCI bus for control and status, as well as for data traffic. Hosts can use SRC shared memory for control and status, but transfer data across the PCI bus. This out-of-band control configuration diverts control overhead from the PCI bus, lessening the burden of ATM's high throughput and robust management requirements on the host system. Figure 3-3 shows an out-of-band control architecture.

Figure 3-3. Out-of-Band Control Architecture



3.3 Write-Only Control and Status

For host-based applications, the host manages the Bt8233 SAR using write-only control and status queues (see Table 3-1). This architecture minimizes PCI bus utilization by eliminating reads from the control path. PCI writes utilize the bus much more efficiently than PCI reads. During a PCI write, the target may post the write data in an internal FIFO, terminate the transaction, and immediately release the bus. On the other hand, during reads, the target retrieves the data while holding the bus. Since the data retrieval takes some time, reads increase the PCI bus utilization.

The Bt8233's write-only architecture uses reads only for segmentation data (PDU) fetches. All control and status transactions are writes. This section describes the management of write-only queues. The purpose and entries of each class of queue are described in later sections.

Table 3-1. Bt8233 Control and Status Queues

Type	Segmentation	Reassembly
Control	Transmit Queue	Free Buffer Queue
Status	Segmentation Status Queue	Reassembly Status Queue

3.3.1 Write-Only Control Queues

The host controls run-time segmentation and reassembly through write-only control queues. There are two types of control queues -- the segmentation Transmit Queues and the reassembly Free Buffer Queues. The host submits buffers for segmentation on the Transmit Queues and supplies buffers for received data on the Free Buffer Queues. Each type of queue is managed as a write-only control queue.

These queues reside in SRC shared memory at a location defined by a base register pointer. To allow multiple clients, the Bt8233 supports 32 queues of each type. The SAR and host manage each queue independently through queue management variables. The SAR stores its variables in internal registers called base tables. The host maintains its own variables within its driver. Each queue contains a programmable number of queue entries.

3.3.1.1 Control Variables

Table 3-2 describes the variables for write-only control queues.

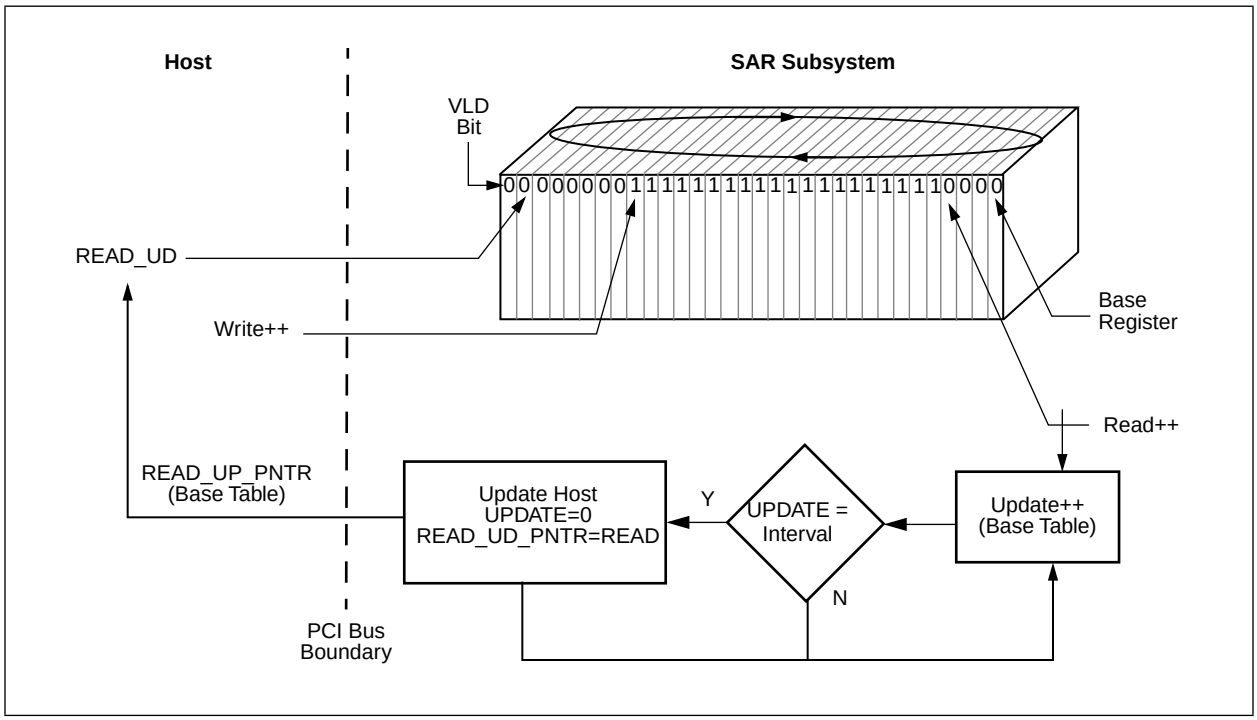
Table 3-2. Write-Only Control Queue Variables

Variable	Definition	Location	Initialization
WRITE	Current host position in queue	Host variable	0
READ_UD	Last known SAR position in queue as seen by host	Host word aligned variable	0
READ	Current SAR position in queue	SAR Base Table	0
INTERVAL	Number of queue entries processed by SRC before writing READ_UD	SAR register	Host defined
UPDATE	Number of queue entries since last write of READ_UD	SAR Base Table	0
READ_UD_PNTR	SAR pointer to READ_UD	SAR Base Table	&READ_UD

3.3.1.2 Queue Management

Figure 3-4 illustrates the control queue management algorithm. The host initializes all of the variables described in Table 3-2. Once the SAR is enabled, it maintains the READ pointer. When the SAR processes a queue entry, it advances the READ pointer (READ++). Since the queues are circular, the pointer will eventually wrap back to zero. It also advances an internal counter, UPDATE (UPDATE++). When INTERVAL queue entries have been processed, the SAR writes its current position in the queue, READ, to a host variable, READ_UD. The host determines the magnitude of INTERVAL at initialization. Larger numbers result in fewer overhead PCI accesses, but will also introduce larger latency between host updates, which reduces the effective size of the queue.

Figure 3-4. Write-Only Control Queue



The host also maintains a pointer into the queue, WRITE. When the host submits a new entry, it must first ensure that the SAR has already processed the entry location. The host compares WRITE to READ_UD. If (WRITE +1) modulo size_of_queue equals READ_ID, then host should halt writing of status queue entry.

This results in only being able to use N-1 queue entries. However if this is not done, then a full condition cannot be distinquished from an empty condition. The host must wait until READ_UD is modified by the SAR before proceeding. This algorithm ensures that the host does not overflow the control queue, without reading the queue itself.

Once it has verified its ownership of the entry, the host writes the entry and increments its write pointer (WRITE++). During this write, the host sets the valid bit (VLD) in the entry to one.

The Bt8233 “snoops” writes to the control queue areas. Once a write is detected to a specific queue, the SAR attempts to process the queue entry at READ. Before acting on the entry, the SAR checks for ownership of the entry, indicated by the VLD bit. Once the Bt8233 has processed the entry, it resets the VLD bit to zero.

3.3.1.3 Underflow Conditions

An underflow condition occurs when the SAR attempts to retrieve a queue entry, and the host has not yet supplied this entry. This condition only happens on the Free Buffer Queues. The SAR detects this condition by checking the queue entry VLD bit. Once detected, the SAR enters an “Underflow Detected” state on this queue only. Since this signifies that no data buffers are available for reassembly, the SAR initiates EPD on all channels assigned to this queue. Section 5.0, Reassembly Coprocessor, describes SAR handling of Free Buffer Queue underflow in detail.

3.3.2 Write-Only Status Queues

The SAR reports status to the host through write-only status queues. Both the segmentation and reassembly coprocessors use their own format of status queue. However, the Bt8233 manages all status queues with the same algorithm.

These queues reside in host memory, or optionally SRC shared memory, at a location defined within the Base Table for each queue. The host must assign word aligned (4 byte), status queue base addresses. To support multiple clients, the Bt8233 provides 32 queues of each type. The SAR and host manage each queue independently through queue management variables. The SAR stores its variables in internal base table registers. The host maintains its variables in its driver. Each queue contains a programmable number of queue entries.

3.3.2.1 Control Variables

Table 3-3 describes the variables for write-only status queue management.

Table 3-3. Write-only Status Queue Variables

Variable	Definition	Location	Initialization
WRITE	Current SAR position in queue	SAR Base Table	0
READ_UD	Last known Host position in queue as seen by SAR	SAR Base Table	0
READ	Current Host position in queue	Host Variable	0
INTERVAL	Number of queue entries processed by Host before writing READ_UD	Host Variable	Host defined
UPDATE	Number of queue entries since last write of READ_UD	Host Variable	0
READ_UD_PNTR	Host pointer to READ_UD	Host Variable	&READ_UD

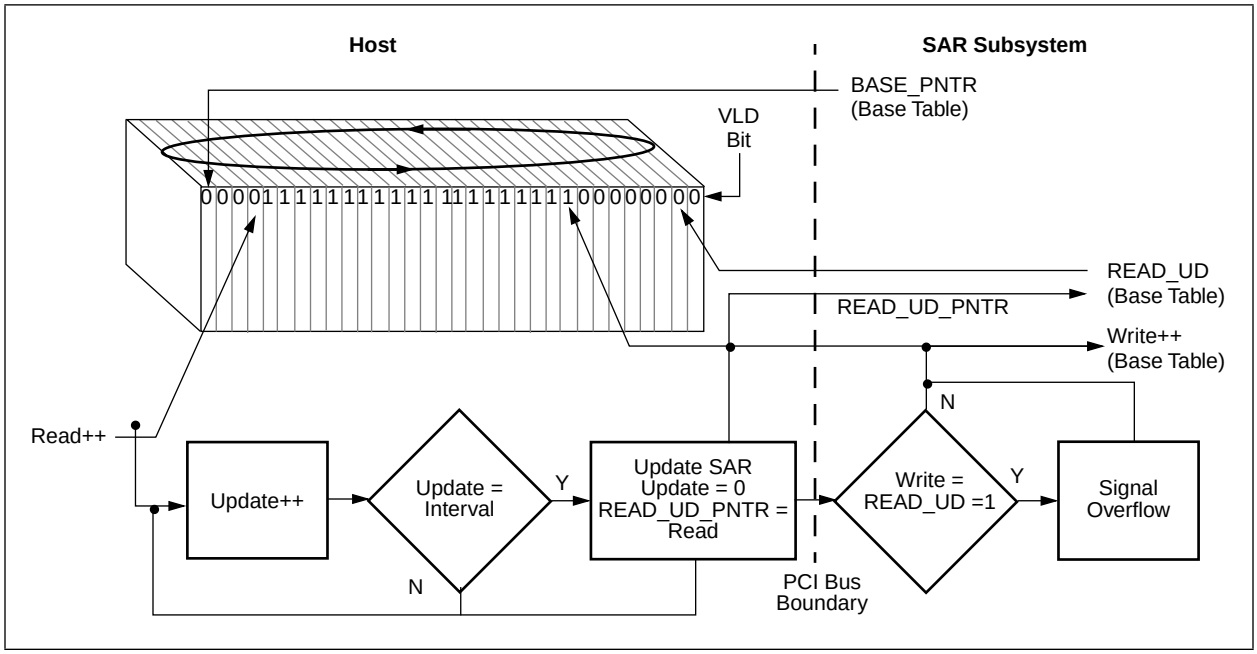
3.3.2.2 Queue Management

Figure 3-5 illustrates the status queue management algorithm. The host initializes all of the variables described in Table 3-3.

The SAR maintains its own write pointer, WRITE. The Bt8233 reports status to the host by writing a status queue entry. After it writes the entry, the Bt8233 increments its write pointer (WRITE++). This write also triggers a maskable interrupt.

The host either responds to this interrupt or periodically polls the status queue. The VLD bit in each queue entry enables polling. The SAR sets the valid (VLD) bit equal to one when it writes a status queue entry. The host resets it to zero after processing an entry.

Figure 3-5. Write-Only Status Queue



The host also maintains a pointer into the status queue, READ. Each time it services an entry, it increments a counter (UPDATE++). When this counter reaches a host specified threshold (INTERVAL), the host informs the SAR of its current queue position by writing READ_UD in the queue's base table register.

3.3.2.3 Overflow Conditions

An overflow condition occurs when the SAR attempts to write a status queue entry, but the status queue entry is unavailable. This condition may happen for both the Segmentation and Reassembly Status Queues. The Segmentation and Reassembly sections describe the handling of this event. In either case, the result is severe and therefore undesirable. The host control service rate should match or exceed the status queue reporting rate of the Bt8233.

The Bt8233 detects an overflow condition by comparing its current WRITE pointer to the READ_UD pointer; i.e., the last known Host READ position. If WRITE points to the entry immediately before the READ_UD (WRITE = READ_UD -1), the SAR detects the imminent overflow condition.

To inform the host of the event, the SAR sets the overflow indication bit (OVFL) in the exhausted status queue. Since it cannot report status, the Bt8233 segmentation and reassembly processing is temporarily halted for VCCs assigned to the overflowed status queue only. All other processes and queues remain operational.

4.0 Segmentation Coprocessor

4.1 Overview

ATM's cell transport mechanism enables large numbers of virtual channels or Virtual Circuit Connections (VCCs) to be multiplexed onto a single physical interface. The segmentation process converts user data (typically Ethernet, Token Ring, or Frame Relay packets) into ATM cells.

The Bt8233 can segment 64K VCCs simultaneously. The segmentation coprocessor block independently segments each channel and multiplexes the VCCs onto the line with cell level interleaving. The Bt8233 XBR Traffic Manager determines the order of execution of these independent processes. to ensure the requested QoS for every channel. For each cell transmission opportunity, the XBR Traffic Manager tells the segmentation coprocessor which VCC to send. Therefore, the segmentation coprocessor acts as a slave to the XBR Traffic Manager.

In addition to converting blocked user data into ATM cells, the Bt8233 generates all AAL overhead for AAL5, or optionally utilizes a null adaptation layer, AAL0. The Bt8233 also generates the ATM cell header, as defined by the host, for each VCC. Furthermore, the segmentation coprocessor and XBR Traffic Manager together provide service-specific features to enhance the performance of Frame Relay internetworking and LAN Emulation.

4.2 Segmentation Functional Description

4.2.1 Segmentation VCCs

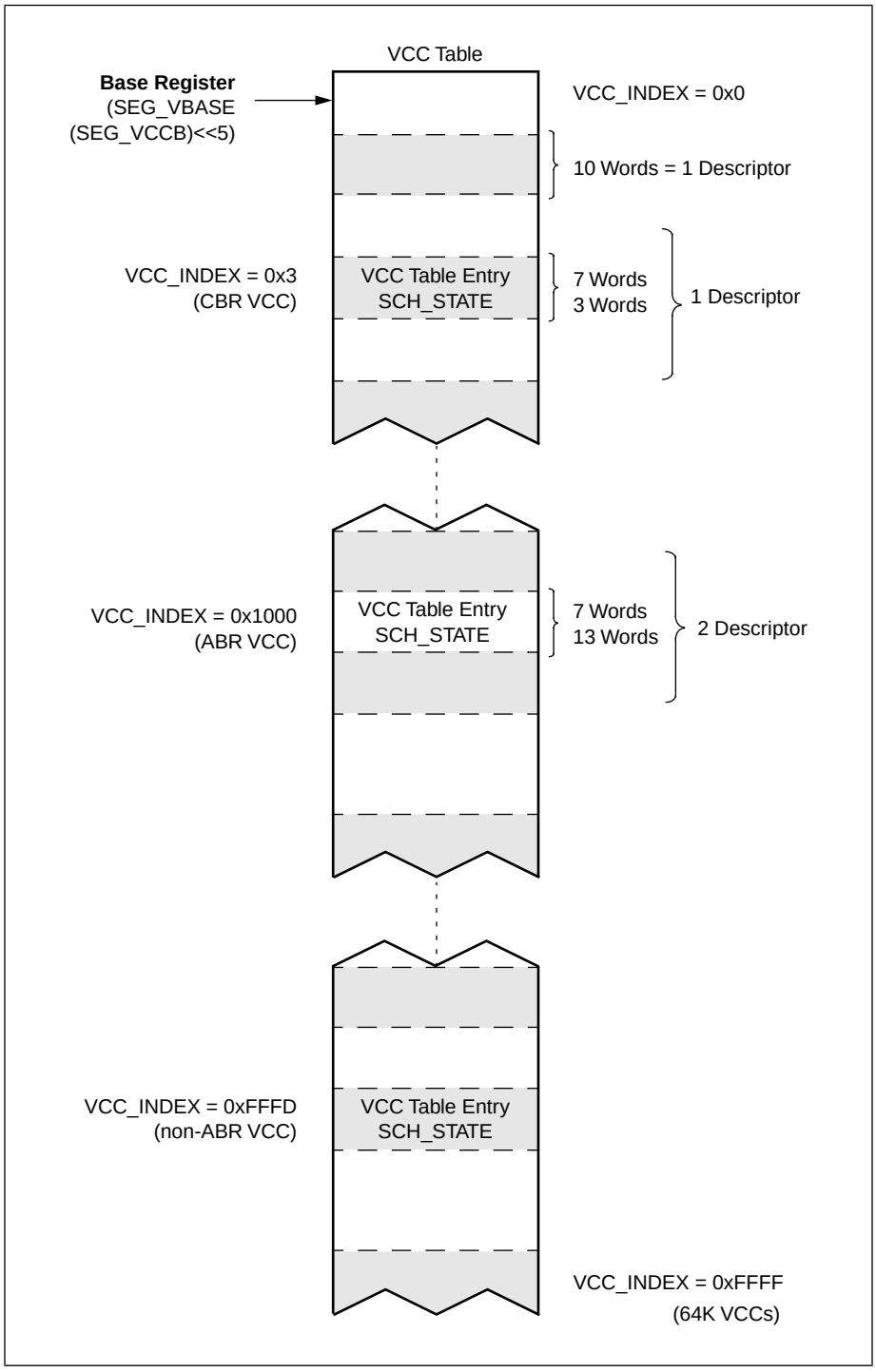
A VCC specifies a single VC or VP in the ATM network. The Bt8233 supports up to 64K segmentation VCCs, referenced by a unique index, VCC_INDEX. The VCC_INDEX identifies a location in the Segmentation VCC Table, an array of 10-word segmentation VCC channel descriptors.

Except for ABR service category VCCs, each segmentation VCC occupies a single descriptor in the table (10 words). Due to the large number of specified parameters for ABR traffic, ABR VCCs occupy two descriptors (20 words) in the Segmentation VCC Table. The VCC_INDEX for ABR VCCs points to the first of the two descriptors, and must be evenly divisible by two.

A Segmentation VCC Table channel descriptor consists of two parts: an AAL specific VCC Table Entry (seven words) and a service class specific Schedule State (SCH_STATE). For non-ABR VCCs, the remaining three words of the channel descriptor form the SCH_STATE entry. For an ABR VCC, the SCH_STATE entry contains 13 words, which require an additional descriptor location.

Figure 4-1 shows a VCC Table with a CBR VCC at VCC_INDEX 3, an ABR VCC located at VCC_INDEX 0x1000, and a non-ABR VCC at VCC_INDEX 0xFFFD. The host allocates the VCC Table in SRC shared memory and provides an address to the SAR in a base register field, SEG_VBASE(SEG_VCCB). Note that for the most efficient ABR performance, all ABR VCCs should be placed in the upper half of the VCC Table (i.e., with smaller VCC indexes). The only reason not to put ABR VCCs at the lowest address range is to place CBR VCCs in the first 32K of VCC addresses.

Figure 4-1. Segmentation VCC Table



The VCC Table Entry contains generic information common to all traffic classes. This includes a default ATM header, which the host may modify during the segmentation process. See Section 4.3.1, Segmentation VCC Table Entry for full details of the structure of a Seg VCC Table entry.

4.2.1.1 VCC Identification

The host allocates a region of SRC shared memory for the Segmentation VCC Table at system initialization, based on the maximum number of connections and the maximum number of ABR connections. The host informs the SAR of the location of the table through the internal base register, SEG_VBASE(SEG_VCCB).

Once a table has been established, the host assigns segmentation VCCs to entries in the table. The host describes the Seg VCC by initializing the Seg VCC Table Entry including the SCH_STATE portions of the assigned VCC. The VCC_INDEX, defined as the offset into the table in 10 word increments, uniquely identifies a segmentation channel. In all communication between the SAR and the host, a VCC_INDEX field specifies a VCC.

4.2.2 Submitting Segmentation Data

Once the host establishes a connection, it supplies data for segmentation. The host submits full or partial PDUs, either one at a time or in batches, for individual VCCs. The SAR accepts PDUs at any time, regardless of the state of the connection, and segments data on each VCC independently.

4.2.2.1 User Data Format

The Bt8233 accepts user PDUs as sequences of data buffers. SRC shared memory resident Segmentation Buffer Descriptor (SBDs) provide the address, length, and control information for buffers. The host forms PDU buffer sequences by linking buffer descriptors. The data buffers themselves contain only user data and reside in host (or optionally SRC shared) memory.

NOTE: Local data buffer segmentation should be limited to low bandwidth applications, such as Signalling, OAM and ILMI). Host data buffers contain the bulk of segmentation data, and can begin on any byte-aligned address in the SAR's PCI address space.

4.2.2.2 Buffer Descriptors

The host submits data using the following process sequence. First, the host allocates a Segmentation Buffer Descriptor (SBD) for each buffer in a message. SBDs reside in SRC shared memory, and must begin on word-aligned addresses. Then, the host describes the buffers in the SBD. This description includes the address and length of the buffer, as well as control fields for the SAR during buffer segmentation. These control fields specify the VCC_INDEX, the AAL type, and header override information for the buffer.

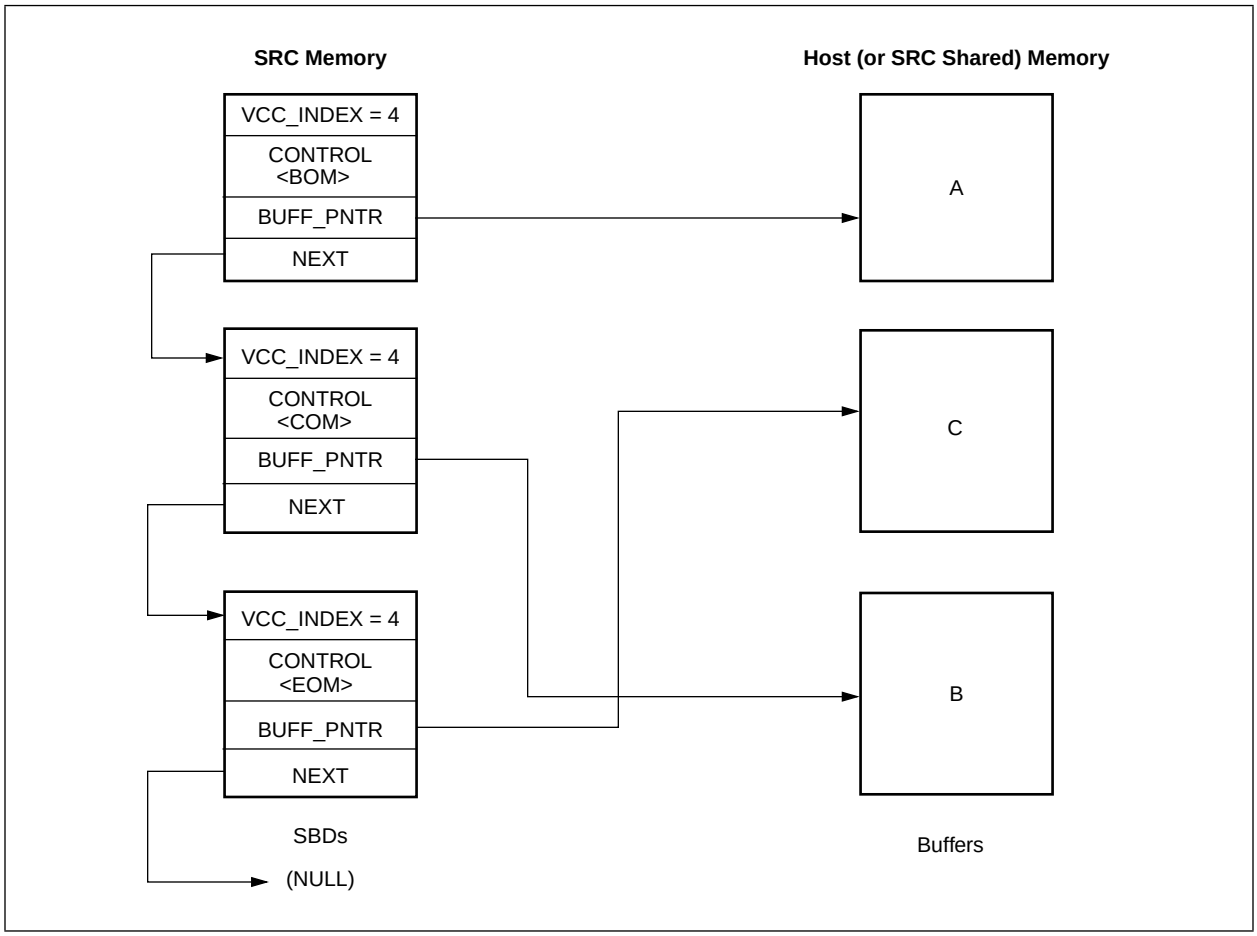
The host then creates PDU message sequences by concatenating buffers. The host forms the buffer sequence by linking a list of buffer descriptors using the SBD's NEXT field. Two bits in the SBD control fields delineate the beginning and end of messages. One bit, BOM, specifies that the buffer contains the beginning of a message. The second bit, EOM, notifies the SAR that the buffer contains the end of the current message. Table 4-1 shows the appropriate utilization of these bits in detail. The host identifies the end of the SBD chain by terminating the list with a NULL (0) NEXT pointer.

Table 4-1. Segmentation PDU Delineation

BOM	EOM	Buffer to Message Adaptation
0	0	Segment as Continuation of Message (COM).
0	1	Terminate current CPCS-PDU at end of buffer (EOM).
1	0	Restart CPCS-PDU generation (BOM). Wait for EOM for termination.
1	1	Buffer contains complete message. Restart/terminate CPCS-PDU.

NOTE: Application Example: Host Linked SBDs—Figure 4-2 shows an example of SBD chaining. The host has formed a 3-buffer message by linking three SBDs. In this example, the SAR will transmit a message sequence of buffer A, then buffer B, and finally buffer C, as the end of message.

Figure 4-2. SBD Chaining



4.2.2.3 Transmit Queues

Once the linked list of SBDs is complete, the host submits the chained message to the SAR for segmentation. The host uses one of the 32 Transmit Queues for this purpose. Each Transmit Queue is a circular queue of one-word entries. The host identifies the next available Transmit Queue Entry according to the write-only host interface specification described in Section 3.3.1, Write-Only Control Queues. The host processor writes a pointer to the SRC shared memory SBD onto the next available transmit queue entry. During this write, the host also sets the VLD bit to indicate the entry is valid.

The Bt8233 detects this write by snooping SRC shared memory accesses. When a write occurs to any of the Transmit Queues, the Bt8233 marks that ring as pending. Once every cell slot, the Bt8233 services one queue entry from one Transmit Queue. The system designer selects one of two service order priority schemes. With the SEG_CTRL(TX_RND) bit set to 1, the Bt8233 services queues in round robin order (i.e., one entry per queue in transmit queue sequence for all active queues). With the SEG_CTRL(TX_RND) bit set to 0, the Bt8233 services the queues in fixed priority order (i.e., entries from higher priority active queues are serviced before lower priority queue entries, with transmit queue 31 having highest priority).

In either case, the SAR services one Transmit Queue Entry by linking the new buffer descriptor chain to the VCC Table Entry identified by the VCC_INDEX in the first SBD. The VCC Table Entry includes pointers to buffer descriptors for segmentation. The SAR will link the new SBDs to the current chain of SBDs on a VCC. The host may submit data to a VCC while the SRC is segmenting a previously submitted message. Once the chain has been linked, the Bt8233 resets the transmit queue VLD bit to zero.

NOTE: Application Example: SAR Transmit Queue Processing—Figure 4-3 and Figure 4-4 illustrate the process of linking SBDs to a VCC Table Entry. As the new buffers are submitted, the VCC is processing a single buffer PDU (BOM/EOM). The Bt8233 accepts new PDUs while it is processing outstanding buffers.

Figure 4-3. Before SAR Transmit Queue Entry Processing

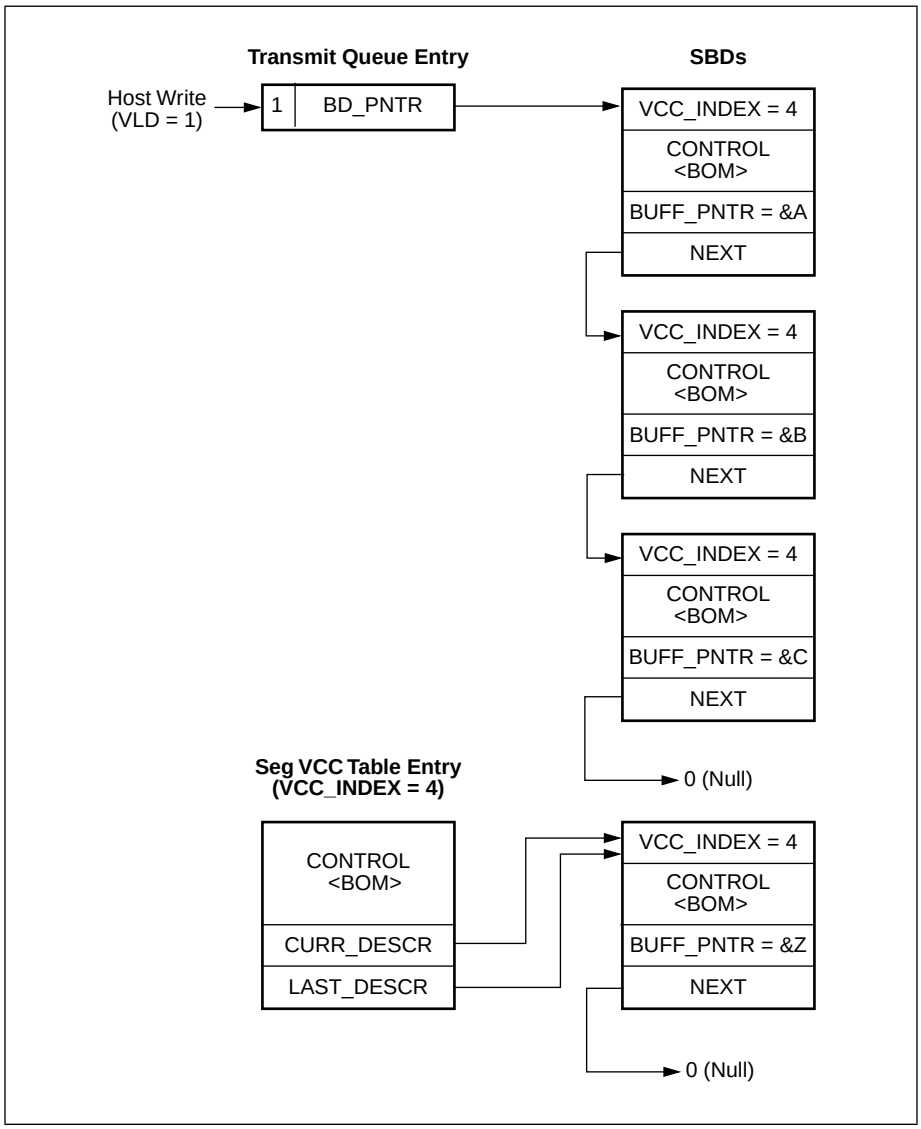
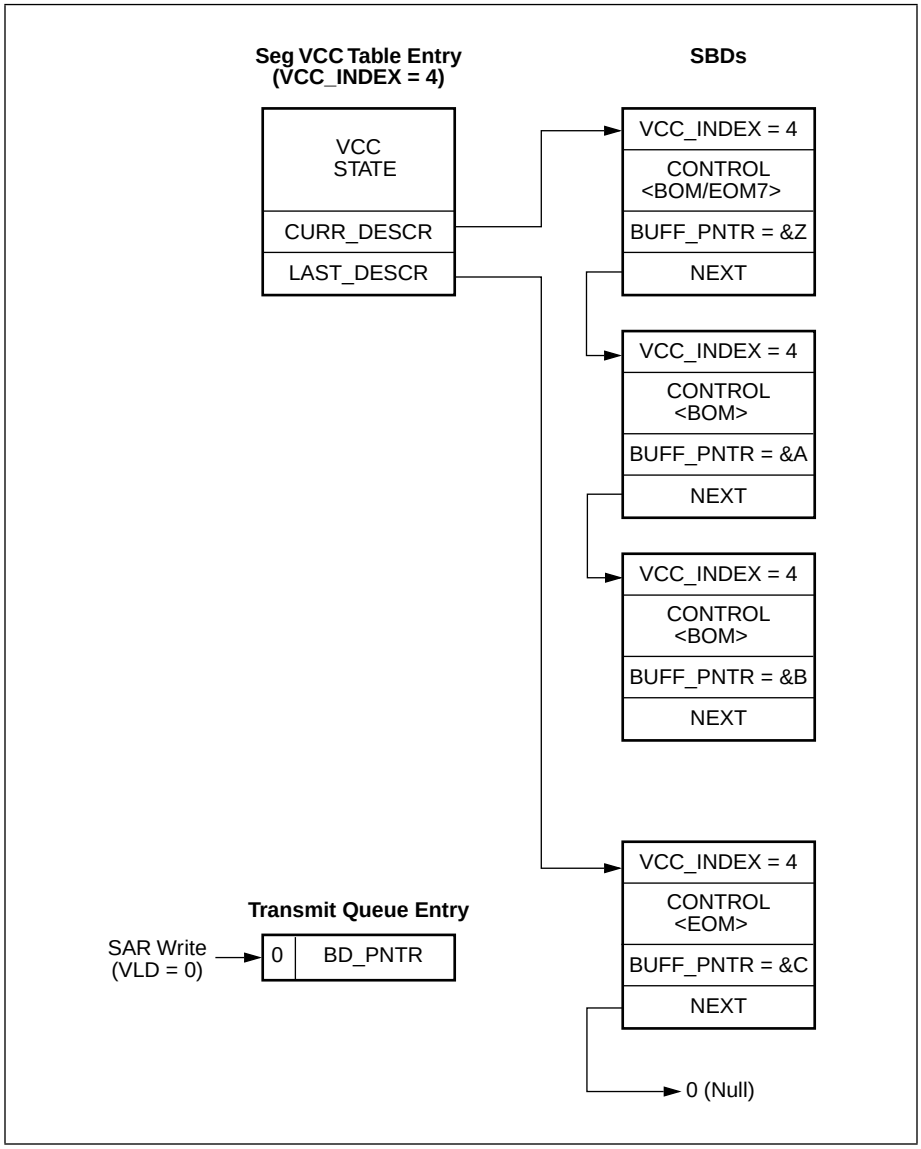


Figure 4-4. After SAR Transmit Queue Entry Processing



4.2.2.4 Partial PDUs

The host may submit partial PDUs to the Bt8233. In this case, the SAR transmits the data and halts on a cell boundary. The partial PDU buffers are not required to be aligned to a cell boundary by the host. The Bt8233 tracks the remaining segmentation data. If a partial cell remains, the Bt8233 will hold the buffer until it can complete a cell. Once the host submits an additional buffer, the Bt8233 resumes segmentation.

4.2.2.5 Virtual Paths

For network management simplicity, the host can create Virtual Path VCCs (i.e., it can segment many VCIs on one VP). The host supplies the VCI for the ATM header within each SBD. When using this method, the Bt8233 must be provided with contiguous linked SBDs that are of the same VCC_INDEX (i.e., the same VCI) for the length of the PDU. This allows the Bt8233 to multiplex VCI messages at the PDU level. For AAL5 segmentation, the host must guarantee that SBDs are linked with PDU multiplexing to preserve CPCS-PDU integrity.

4.2.3 CPCS-PDU Processing

The buffers submitted by the user contain only user data, the CPCS Service Data Unit (CPCS-SDU). The Bt8233 adds the CPCS-PDU protocol fields to the CPCS-SDU. The SAR supports two AAL levels -- the standard AAL5, and a transparent adaptation layer (AAL0).

Specific features also allow the generation of OAM cells. Chapter 7.0, OAM Functions covers OAM generation in detail.

4.2.3.1 AAL5

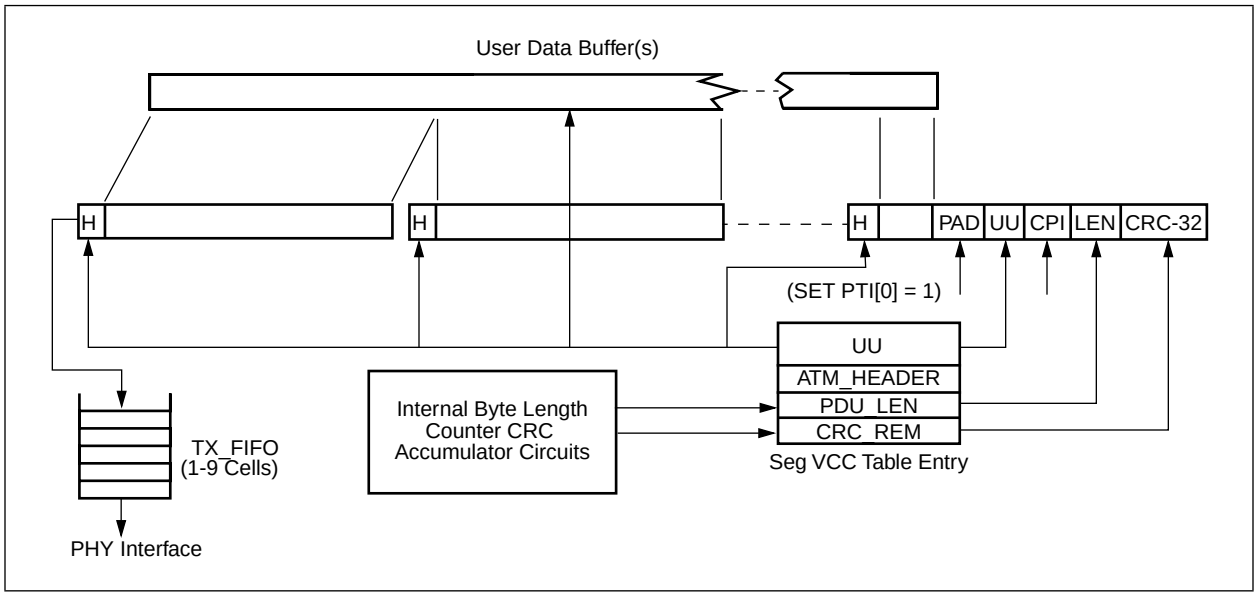
For AAL5, the SAR generates the CPCS-PDU trailer, and pads the CPCS-SDU to align the PDU to a cell boundary. The Bt8233 generates the Pad, Length, Common Part Indicator (CPI) and Cyclic Redundancy Check (CRC) fields according to I.363. The host supplies the CPCS User-to-User Indication (UU) field in the first buffer descriptor in a message and the Bt8233 transmits it following I.363.

The SAR generates the ATM header according to host-initialized settings in the VCC Table Entry. The Bt8233 terminates AAL5 PDUs by setting bit zero of the Payload Type Identifier field, PTI[0], equal to one.

The host aborts PDUs by activating an EOM SBD abort option. The host activates this by setting the following fields in the SBD entry to the following values: AAL_MODE = AAL5 (b00), and AAL_OPT = ABORT (b01).

Figure 4-5 shows the Bt8233's AAL5 PDU generation scheme. The SAR uses internal circuits to generate and store PDU length and CRC-32 in the Seg VCC Table. The Bt8233 transmits these fields within the EOM cell. The Pad and CPI fields are generated internally.

Figure 4-5. AAL5 CPCS–PDU Generation



4.2.3.2 AAL0

The Bt8233 also supports a transparent or NULL adaptation layer, AAL0. AAL0 maps CPCS-SDUs directly to CPCS-PDU payloads. The SAR pads the SDU to a 48-byte cell payload boundary, but generates no other overhead. The SAR generates the ATM header and PDU termination indications, as with AAL5.

4.2.4 ATM Physical Layer Interface

Once the segmentation coprocessor has formed an ATM cell, the Bt8233 transfers the cell to the Transmit FIFO. The user chooses the length of this FIFO, with possible sizes from one to nine cells. The FIFO depth is programmable, since there is a trade-off between absorbing PCI latency with a longer FIFO and introducing greater jitter. Section 6.2.3.3, CBR CDV in the Traffic Management chapter discusses this trade-off in greater depth. Once sent to the transmit FIFO, the cell passes through the FIFO to the PHY layer interface circuits.

4.2.5 Status Reporting

The Bt8233 informs the host of segmentation completion using Segmentation Status Queues. The host assigns each VCC to one of 32 status queues, enabling a multiple peer architecture as described in Section 3.2, Multiple Client Architecture. The Bt8233 reports status entry on either PDU or buffer boundaries, selectable on a per-VCC basis by setting the STM_MODE bit. PDU boundary status is referred to as Message Mode, while buffer status reporting is called Streaming Mode. Error conditions also generate status queue entries, although this is a rare occurrence within a Bt8233 subsystem's segmentation block. The Segmentation Status Queues operate according to the write-only host interface, defined in Section 3.3.2, Write-Only Status Queues.

The Bt8233 returns a user-supplied field (USER_PNTR) from the first SBD associated with the status entry. The SAR does not use this field for any internal purpose; it simply circulates the information back to the host. The value of USER_PNTR must uniquely describe the segmented buffer associated with the SBD. USER_PNTR may contain the address of the buffer or of a host data structure describing the buffer. To simplify host management, the Bt8233 also returns the VCC_INDEX of the VCC on which the buffer was transmitted.

4.2.6 Virtual FIFOs

In addition to gathering PDU data from buffers, the Bt8233 provides an optional method to segment from a fixed PCI address, or Virtual FIFO. The Bt8233 supports AAL0, CBR Virtual FIFO segmentation.

The host configures the channel for Virtual FIFO operation by setting the CURR_PNTR and RUN fields to zero in the Seg VCC Table entry. The host writes the FIFO address to the FIFO_PNTR field in the VCC Table entry. The host also initially sets the SCH_MODE field = CBR.

At this point the host can start writing cells to the external host Transmit FIFO. Once the FIFO is almost full, the host sets the RUN bit to a logic high. The SAR will then start reading from the FIFO. When the FIFO gets below almost empty, the host will set the SCH_OPT bit to a logic high. The SAR will then skip a cell transmit opportunity in order to allow the FIFO to refill. After the SAR skips a cell, it will reset the SCH_OPT bit to a logic low.

In this mode, the segmentation coprocessor reads 48 bytes of payload from the host FIFO and prepends the ATM_HEADER value in the VCC Table entry. The host does not use the Transmit Queue for SBDs for Virtual FIFOs. The Bt8233 transmits cell payloads from this location indefinitely, with no status reporting.

4.3 Segmentation Data Structures

4.3.1 Segmentation VCC Table Entry

Each Segmentation VCC Table Entry occupies one 10-word descriptor of the Segmentation VCC Table. The first seven words are generic, independent of traffic class. The last three or thirteen words provide additional parameters specific to service classes.

There are two basic formats for the generic part of the VCC Table Entry -- AAL5/AAL0 and Virtual FIFO. Each completely describes the state of the segmentation process for individual VCCs. Tables 4-2 and 4-3 describe the format of AAL5 and AAL 0 VCC Table Entries.

Table 4-2. Segmentation VCC Table Entry—AAL5/AAL0 Format

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	PM_INDEX							Rsvd				LAST_PNTR																				
1	Reserved								Rsvd			Reserved																				
2	ATM_HEADER																															
3	Reserved																Reserved															
4	CRC_REM																															
5	STM_MODE	STAT					PM_EN	Rsvd	Rsvd	Rsvd		CURR_PNTR																				
6	Rsvd	VPC	Rsvd					SCH_MODE				Rsvd	PRI			SCH_OPT	Reserved															
7-9/19	SCH_STATE																															

Notes: 1. = Written by host at VCC setup.
2. = May be dynamically modified during active segmentation.

Table 4-3. Segmentation VCC Table Entry–AAL5/0 Format Field Descriptions

Field Name	Description
PM_INDEX	Performance Monitoring index. 128 VCCs may be selected for automatic OAM PM generation. Each monitored VCC has a unique performance monitoring index. If this field is changed while the VCC is active, only the byte containing the field should be written. (See Chapter 6.0, Traffic Management).
LAST_PNTR	Points to last buffer descriptor currently linked to the VCC. The two least significant bits of the pointer are assumed to be 0 (word-aligned). The address of the buffer descriptor is (LAST_PNTR << 2) or (LAST_PNTR * 4).
ATM_HEADER	Used for each ATM cell for the VCC. The transmitted header may be modified by option bits in the current buffer descriptor.
CRC_REM	Accumulated value of AAL5 32-bit CRC, calculated on-the-fly by the SAR & appended to the PDU at EOM.
STM_MODE	Streaming Status Mode 0 Message Mode: a status entry is written (with a corresponding maskable interrupt) only when the last buffer in a message completes segmentation. The status entry written corresponds to the first buffer descriptor in the message. 1 Streaming Mode: a status entry is written (with a corresponding maskable interrupt) for each buffer as it completes segmentation.
STAT	Identifies the Segmentation Status Queue used for all status entries.
PM_EN	Enables performance monitoring for the VCC. If this bit is changed while the VCC is active, only the byte containing the bit should be written. (See Chapter 7.0, OAM Functions).
CURR_PNTR	Pointer to the current buffer descriptor for the VCC. This field is automatically updated by the SAR. The two least significant bits of the pointer are assumed to be zero (word-aligned). The address of the descriptor is (CURR_PNTR << 2) or (CURR_PNTR * 4).
VPC	On a VCC with SCH_MODE = ABR, indicates a VP (instead of a VCC) connection, and RM cells will be generated on VCI = 6.
SCH_MODE	Traffic class of VCC. See Chapter 5.0, Reassembly Coprocessor for details. 000 UBR - Unspecified Bit Rate 001 CBR - Constant Bit Rate 010 Reserved 100 VBR1 –Single Leaky Bucket VBR 101 VBR2 –Dual Leaky Bucket VBR with both buckets always active 110 VBRC–Dual Leaky Bucket VBR with bucket 1 applied only to CLP=0 cells 111 ABR –Available Bit Rate (as specified by TM 4.0)
PRI	Segmentation priority. The lowest priority is zero. This field is not active when SCH_MODE is CBR.
SCH_OPT	Schedule option. The use of this bit depends on the setting of the SCH_MODE field. VBR1, VBR2, VBRC, ER: Initializes bucket state to send maximum burst. SAR writes this bit to 0 after bucket state initialized. CBR: Indicates that the next cell slot opportunity should be skipped. This bit is written by the host and cleared by the SAR after the cell slot is skipped. If this bit is changed while the VCC is active, only the byte containing the bit should be written.
SCH_STATE	Specific scheduling state information. The contents of this field depend on the setting of the SCH_MODE field. It is not used when SCH_MODE is set to UBR or CBR. (The contents of this field are detailed in Chapter 6.0, Traffic Management.

Table 4-4 shows the format for Virtual FIFO VCC Table Entries, including setup values for restricted fields. Table 4-5 describes the field that differs from the AAL5/AAL0 format.

Table 4-4. Segmentation VCC Table Entry–Virtual FIFOs

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	PM_INDEX							Rsvd				LAST_PNTR																				
1	Reserved								Rsvd			BOM_PNTR																			
2	ATM_HEADER																															
3	FIFO_PNTR																															
4	CRC_REM																															
5	STM_MODE	STAT					PM_EN	Rsvd	RUN=1	Rsvd		CURR_PNTR= 0x00000																				
6	Reserved								SCH_MODE (=001 - CBR)			Rsvd	PRI			SCH_OPT	Reserved														
7-9/19	SCH_STATE																															
	Notes: 1. = Written by host at VCC setup. 2. = May be dynamically modified during active segmentation.																															

Table 4-5. Segmentation VCC Table Entry - Virtual FIFO Format Field Descriptions

Field Name	Description
FIFO_PNTR	Pointer to the PCI space data FIFO for CBR_FIFO scheduling mode.

4.3.2 Data Buffers

Data buffers contain CPCS-SDUs for segmentation and reside in host or SRC shared memory. The Bt8233 retrieves host data buffers from any byte aligned PCI address using the “READ Multiple” PCI command. SRC shared memory data buffers must be aligned on word boundaries. Buffers contain any number of bytes of ONLY user data, up to a maximum of 64 K.

4.3.3 Segmentation Buffer Descriptors

SBDs reside in SAR shared memory on word aligned addresses. (refer to Table 4-6 through Table 4-8). The host controls the allocation and management of SBDs. For each buffer to be segmented, the host utilizes one buffer descriptor from its pool of free descriptors. Table 4-9 provides buffer field descriptor descriptions.

Table 4-6. SBD Entry Format

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	UU								Rsvd	NEXT_PNTR																						Rsvd
1	USER_PNTR																															
2	BUFFER_PNTR																															
3	Rsvd	LOCAL	SET_CI	SET_CLP	HEADER_MOD	RPL_VCI	OAM_STAT	GEN_PDU	CRC10	AAL_MODE		AAL_OPT		CELL	BOM	EOM	LENGTH															
4	MISC_DATA																SEG_VCC_INDEX															

Table 4-7. MISC_DATA Field Bit Definitions with HEADER_MOD Bit Set

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Def.	GFC_DATA				Rsvd		WR_GFC	WR_PTI	WR_VCI	PTI_DATA			VCI_DATA			
Note: This definition of bits 31:16, MISC_DATA field, applies when the HEADER_MOD bit is set; used when generating OAM cells.																

Table 4-8. MISC_DATA Field Bit Definitions with RPL_VCI Bit Set

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Def.	NEW_VCI															
Note: This definition of bits 31:16, MISC_DATA field, applies when the RPL_VCI bit is set; used when identifying virtual channels under a VP VCC.																

Table 4-9. Buffer Descriptor Field Descriptions *(1 of 2)*

Field Name	Description
UU	AAL5 User-to-User indication. This field is copied to the VCC Table Entry UU field when BOM is set and AAL_MODE is AAL5.
NEXT_PNTR	Pointer to next buffer descriptor for the VCC. The two least significant bits of the pointer are assumed to be zero (word-aligned). The host links segmentation buffer descriptors by writing this field to ((ADDRESS of SDB)>>2) or ((ADDRESS of SBD)/4) before submitting the chain on the Transmit Queue. The NEXT_PNTR of the last buffer descriptor in a chain is set to NULL (=0).
USER_PNTR	User data field returned in status entry. This field may equal BUFFER_PNTR. The SAR circulates this field back to the host in the status entry without using it internally.
BUFFER_PNTR	Pointer to segmentation buffer in host or SRC shared memory space. Host or local location is determined by the LOCAL bit.
LOCAL	0 BUFFER_PNTR is a byte aligned PCI address. 1 BUFFER_PNTR is a word aligned address in SRC shared memory instead of host memory.
SET_CI	0 The Bt8233 generates Bit 1 of PTI[2:0] from the VCC Table Entry ATM_HEADER field. 1 Sets bit 1 of the ATM header PTI[2:0] field for all cells in buffer to 1.
SET_CLP	0 The Bt8233 generates the CLP bit from the VCC Table Entry ATM_HEADER field. 1 Sets the ATM header CLP bit for all cells in buffer to 1. Also used to control VBR CLP Dual Leaky Bucket mode.
HEADER_MOD	0 The Bt8233 ignores the WR_GFC, WR_PTI and WR_VCI bits in this buffer descriptor. 1 Activates the WR_GFC, WR_PTI, and WR_VCI bits for this buffer descriptor.
RPL_VCI	0 The Bt8233 generates the VCI field from the VCC Table Entry ATM_HEADER field. 1 The Bt8233 generates the VCI field from the NEW_VCI for all cells in the buffer. NEW_VCI will also be copied in to the VCI portion of the ATM_HEADER field in the VCC entry
OAM_STAT	Buffer will report status to the global OAM Status Queue SEG_CTRL(OAM_STAT_ID) instead of the STAT specified in the VCC Table Entry. For Message mode VCCs (VCC Table Entry STM_MODE = 0), the OAM_STAT bit of the last descriptor for the CPCS-PDU determines which queue to use (even though the first descriptor is returned in message mode). See Chapter 7.0, OAM Functions for more details.
GEN_PDU	1 Generate AAL5 trailer. This bit should be set to one for all AAL5 VCCs.
CRC10	Overwrite last ten bits of each cell with CRC10 calculation. Used for OAM cells.
AAL_MODE	Controls AAL segmentation mode 00 AAL5 01 AAL0 Read 48-octet ATM cell payload from segmentation buffer. Only formatting is to set PTI[0] on last cell of an EOM buffer. 10 Rsvd 11 Rsvd

***Table 4-9. Buffer Descriptor Field Descriptions** (2 of 2)*

Field Name	Description
AAL_OPT	Options for AAL_MODE For AAL_MODE = AAL0 00 Normal operation 01 SINGLE 10 Rsvd 11 Rsvd For AAL_MODE = AAL5 00 Normal operation 01 ABORT SINGLE LENGTH is ignored and 48 octets are read from buffer to form the payload of a single ATM cell. VCC Table Entry CRC_REM, BUFF_LENGTH, PDU_LENGTH are not affected. By using the LINK_HEAD bit in the transmit entry, the buffer descriptor is linked at start of buffer descriptor chain for VCC. This means that there are no concerns with mid-cell insertion, and that the cell will have low latency. This is intended for OAM cells. NOTE: This option MUST be set in the BD for any Tx Queue entry with LINK_HEAD set. To do otherwise may result in corrupted seg data. ABORT Send AAL5 ABORT cell (no data is read from segmentation buffer). A buffer that has both ABORT and BOM set is returned without sending an abort cell.
CELL	0 Bt8233 reads the 48 octet payload of ATM cells from memory and generates the ATM header internally. 1 The Bt8233 reads the entire 52-octet ATM cell from segmentation buffer. The ATM_HEADER stored in the VCC Table Entry is not used in this mode and AAL_MODE is ignored.
BOM	1 Buffer contains the beginning of a message. See Table 4-1.
EOM	1 Buffer contains the end of a message. See Table 4-1.
LENGTH	Number of bytes of data contained in the segmentation buffer. The Bt8233 places no restrictions on this value besides an absolute maximum size of 64 K.
GFC_DATA	Data for WR_GFC option.
WR_GFC	0 The Bt8233 generates the GFC field from the VCC Table Entry ATM_HEADER field. 1 The Bt8233 overwrites the ATM header GFC field for all cells in the buffer with GFC_DATA. Global GFC changes (active for all buffers of VCC) can be set in the VCC Table Entry ATM header. This bit is active only when HEADER_MOD is set.
WR_PTI	0 The Bt8233 generates the PTI field from the VCC Table Entry ATM_HEADER field. 1 The Bt8233 overwrites the ATM header PTI field for all cells in the buffer with PTI_DATA. The host may use this feature to generate F5 and PM OAM cells. See Chapter 7.0, OAM Functions. This bit is active only when HEADER_MOD is set. This bit disables PM TUC and BIP16 calculations.
WR_VCI	0 The Bt8233 generates the VCI field from the VCC Table Entry ATM_HEADER field. 1 The Bt8233 overwrites the ATM header VCI field for all cells in the buffer with (0x0000|VCI_DATA). (MSBs of VCI are set to zero). Used to generate F4 OAM cells. See Chapter 7.0, OAM Functions. This bit is active only when HEADER_MOD is set. This bit disables PM TUC and BIP16 calculations.
PTI_DATA	Data for WR_PTI option. Normally used to generate OAM cells.
VCI_DATA	Data for WR_VCI option. Normally used to generate OAM cells.
NEW_VCI	Data for the RPL_VCI. The Bt8233 overwrites the VCC Table Entry ATM_HEADER VCI field with this data. Therefore, the effect is permanent until the next buffer descriptor with RPL_VCI is processed.
SEG_VCC_INDEX	Identifies the VCC Entry in the VCC Table. The Bt8233 links this buffer descriptor to the identified VCC.

4.3.4 Transmit Queues

4.3.4.1 Entry Format The host submits chains of Segmentation Buffer Descriptors to the Bt8233 by writing a single word Transmit Queue Entry. Tables 4-10 through Table 4-13 describe the format of these entries.

Table 4-10. Transmit Queue Entry Format

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	VLD	LINK_HEAD	FIND_CHAIN	Rsvd						SEG_BD_PNTR																						Rsvd	

Table 4-11. Transmit Queue Entry Field Descriptions

Field Name	Description
VLD	0 Entry invalid. Waiting for the host to submit new data for segmentation. 1 Entry valid. The SAR will process the entry when its read pointer into the queue advances to this entry. Written to 1 by the host when submitting a new entry. The SAR clears this bit to zero when it has successfully linked the buffer descriptor chain to the VCC Table.
LINK_HEAD	0 The Bt8233 links the new descriptor chain at the end of the existing chain on the VCC. 1 The Bt8233 links the new descriptor chain at the head of the existing chain. If this bit is set, the buffer must contain data for at least one cell. Only a single buffer descriptor may be linked to a transmit queue entry when this bit is set. This bit is intended for use with the buffer descriptor SINGLE option to send in line management cells with reduced latency(1).
FIND_CHAIN	Indicates the SAR is searching for the end of the buffer descriptor chain. The host always writes this bit to zero.
SEG_BD_PNTR	Points to the first buffer descriptor in the new buffer descriptor chain. Bit 22:2 of the address are specified; the two least significant bits of the pointer are assumed to be zero (word-aligned).

Notes: (1). The SINGLE option must be set in the buffer descriptor for any Tx Queue entry with LINK_HEAD set. To do otherwise may result in corrupted seg data.

4.3.4.2 Queue Management

The Transmit Queues reside in a single continuous section of SRC shared memory. During initialization the host configures the number of entries per queue with the SEG_CTRL(TR_SIZE) field. The size ranges from 64 to 4096 entries per queue. The host also selects a priority scheme at initialization with the SEG_CTRL(TX_RND) bit. Both of these fields are static configurations and must not be changed during run time operation.

By initializing the SEG_TXBASE register, the host determines the base address of all active Transmit Queues. This register contains the base address of the first queue, the number of active queues, and the write-only update interval for all queues. Other internal registers, the Transmit Queue Base Table Entries, track the current state of the queue. Table 4-12 and Table 4-13 describe the fields of these queues.

The byte address of any Transmit Queue Entry is given by:

```
(SEG_TXBASE(SEG_TXB)*128)
+ <Transmit Queue Number> * <Decoded TQ_SIZE value>
+ <Entry number>*4
```

The host manages each Transmit Queue as an independent write-only control queue. Chapter 3.0, Host Interface describes the run time management of a write-only control queue. The Transmit Queue Base Table contains all of the queue control variables except for INTERVAL, which is located in the SEG_TX_BASE register.

Table 4-12. Transmit Queue Base Table Entry

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	READ_UD_PNTR																														Rsvd	LOCAL
1	Rsvd												UPDATE				Rsvd				READ											

Table 4-13. Transmit Queue Base Table Entry Field Descriptions

Field Name	Description
READ_UD_PNTR	Points to READ_UD used by host to prevent queue overflow. The SAR will write its read pointer into the queue to this address periodically. See Chapter 2.0, Bt8233 Architecture Overview for details.
LOCAL	0 READ_UD located in PCI address space. 1 READ_UD located in SRC shared memory.(1)
UPDATE	SAR position in update interval. Number of queue entries processed since last update of READ_UD.
READ	SAR read pointer. Represents the SAR's current position in the Transmit Queue.
Notes: (1). For write-only PCI host interfaces, this bit should be set low.	

4.3.5 Segmentation Status Queues

4.3.5.1 Entry Format

The Bt8233 reports segmentation to the host on one of 32 status queues. Each entry on the queue is 2 words. Table 4-14 and Table 4-15 describe the format of these entries.

Table 4-14. Segmentation Status Queue Entry

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	USER_PNTR																															
1	VLD	Rsvd	STOP	DONE	SINGLE	OVFL	I_EXP					I_MAN		Rsvd			SEG_VCC_INDEX															

Table 4-15. Segmentation Status Queue Entry Field Descriptions

Field Name	Description
USER_PNTR	Copy of the USER_PNTR from the Segmentation Buffer Descriptor. The SAR circulates this field from the SBD without using it internally. In Message mode, the SAR returns the USER_PNTR of the BOM buffer. In Streaming Mode, the SRC returns the USER_PNTR of all buffers.
VLD	0 Entry invalid. Indicates that the SAR has not written the entry, and the host should halt status processing. 1 Entry valid. Indicates that the host may process the entry. Written to 1 by the SAR. Written to 0 by the host.
STOP	VCC has stopped because no more segmentation data is available.
DONE	Set when buffer segmentation is complete, and buffer is released to host.
SINGLE	Set if SINGLE option set in the segmentation buffer descriptor.
OVFL	Overflow: status entry is last entry available. See Status Queue Overflow, below.
I_EXP	Current I_EXP rate parameter of the VCC. This field is written only when VCC_INDEX(SCH_MODE) = ABR. See Chapter 6.0, Traffic Management.
I_MAN	The two MSBs of the current I_MAN rate parameter for the VCC. This field is written only when VCC_INDEX(SCH_MODE)=ABR. See Chapter 6.0, Traffic Management.
SEG_VCC_INDEX	Segmentation VCC index on which the SAR transmitted the buffer or PDU.

4.3.5.2 Queue Management

At initialization, the host assigns the location and size of up to 32 queues by initializing internal registers, the Segmentation Status Queue Base Table Entries (refer to Table 4-16 and Table 4-17). The location and size of each queue is independently programmable via these base tables.

The SAR tracks its current position and the most recent known host position in the queues with fields in the base table entries. The host manages the queues as write-only status queues. The Status Queue Base Table Entry contains all of the SAR's write-only control variables.

Table 4-16. Segmentation Status Queue Base Table Entry

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	BASE_PNTR																														Rsvd	LOCAL
1	SIZE		Rsvd		WRITE												Rsvd				READ_UD											

Table 4-17. Segmentation Status Queue Base Table Entry Field Descriptions

Field Name	Description
BASE_PNTR	Points (Bits 31:2) to base of status queue. Bits 1:0 are always 0 (word aligned).
LOCAL	0 Status queue located in PCI address space. 1 Status queue located in SRC shared memory address space. This bit should be set to 0 for a write-only PCI Host architecture.
SIZE	Number of entries in this status queue: 00 64 01 256 10 1024 11 4096
WRITE	SAR write pointer. Represents the SAR's current position in the queue.
READ_UD	Last update of the host processor read pointer. This field is written by the host processor.

4.3.5.3 Status Queue Overflow

Since status queues contain a finite number of entries, it is possible that the SAR will exhaust the available entries. Although the SAR handles this condition, the host should attempt to prevent overflows.

The Bt8233 detects when it writes the last available entry in a status queue (WRITE=READ_UD-1), and alerts the host to this condition by setting the OVFL bit in the status entry. Until the host services the queue and increments the READ_UD pointer in the base table register, the Bt8233 inhibits segmentation on all channels that report on the overflowed status queue. All other channels are unaffected.

4.3.6 Segmentation Internal SRAM Memory Map

The Segmentation Internal SRAM is in the address range 0x1400-0x17FF.

The Segmentation Status Queue Base Table Registers (SEG_ST_QUn) are in the address range 0x1400-0x14FF.

The Internal Transmit Queue Base Table Registers (SEG_TQ_QUn) are in the address range 0x1500-0x15FF.

Other Internal segmentation and scheduler registers are in the address range 0x1600-0x17FF. See Table 4-18 for the segmentation Internal SRAM Memory Map.

Table 4-18. Segmentation Internal SRAM Memory Map

Address	Name	Description
Segmentation Status Queue Base Table Registers (SEG_SQ_QUn):		
0x1400–0x1407	SEG_SQ_QU0	Status Queue 0 Base Table.
0x1408–0x140F	SEG_SQ_QU1	Status Queue 1 Base Table
⋮	⋮	⋮
0x14F8–0x14FF	SEG_SQ_QU31	Status Queue 31 Base Table.
Internal Transmit Queue Base Table Registers (SEG_TQ_QUn):		
0x1500–0x1507	SEG_TQ_QU0	Transmit Queue 0 Base Table
0x1508–0x150F	SEG_TQ_QU1	Transmit Queue 1Base Table
⋮	⋮	⋮
0x15F8–0x15FF	SEG_TQ_QU31	Transmit Queue 31Base Table
Further Internal Segmentation and Scheduler Registers:		
0x1600–0x163F	Reserved	Not Implemented
0x1640–0x1643	(Reserved Name)	Initialize to 0xFFFFFFFF.
0x1644–0x1647	(Reserved Name)	Initialize to 0xFFFFFFFF.
0x1648–0x164B	(Reserved Name)	Initialize to 0xFFFFFFFF.
0x164C–0x164F	(Reserved Name)	Initialize to 0xFFFFFFFF.
0x1740–0x1743	(Reserved Name)	Initialize to 0xFFFFFFFF.
0x1744–0x1747	(Reserved Name)	Initialize to 0xFFFFFFFF.
0x1748–0x174B	(Reserved Name)	Initialize to 0xFFFFFFFF.
0x174C–0x174F	(Reserved Name)	Initialize to 0xFFFFFFFF.
0x1750–0x17FF	Reserved	Not Implemented

5.0 Reassembly Coprocessor

5.1 Overview

The reassembly coprocessor processes cells received from the ATM Physical Interface Block. The coprocessor extracts the AAL SDU payload from the received cell stream and reassembles this information into buffers supplied by the host system. The Bt8233 supports AAL0 and AAL5 reassembly as well as cell mode (1-cell PDUs through a Virtual FIFO, for CBR voice traffic).

The Bt8233 reassembles up to 64K VCCs simultaneously. Individual connections are identified through separate VPI and VCI Index Table structures. The VPI/VCI Index Table mechanism provides very fast, consistent channel identification over the full range of VPI/VCI addresses.

CPCS-PDU payload data, the CPCS-SDU, fills the host-supplied data buffers assigned to each VCC. The host assigns each VCC to one or two of 32 independent buffer pools, from which the Rsm coprocessor draws buffers as needed.

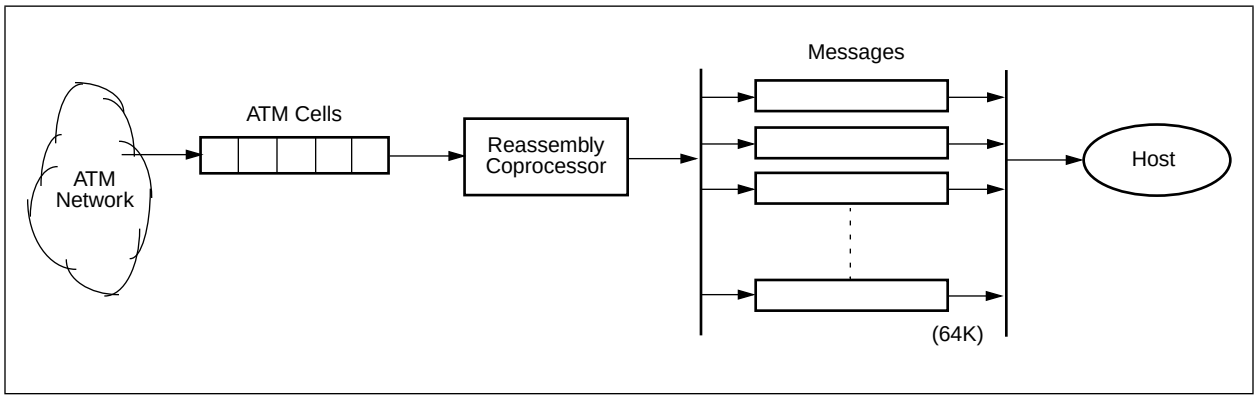
The Bt8233 extracts the CPCS-SDU from the CPCS-PDU, writes the SDU to host-supplied buffers, and performs all CPCS-PDU checks. The results of these checks, as well as AAL information, are passed to the host on one of 32 independent status queues.

NOTE: This chapter provides information on the functions and data structures of the reassembly coprocessor. For detailed information on how the Bt8233 handles PM cells, deals with OAM functions, and interacts with the segmentation coprocessor in handling traffic management and scheduling, refer to Chapter 6.0, Traffic Management, and Chapter 7.0, OAM Functions.

5.2 Reassembly Functional Description

Each cell received from the ATM Physical Interface Block belongs to any one of a possible 64 k virtual channels, or simultaneous messages. Due to the asynchronous nature of ATM, the cell contained in any incoming cell slot may belong to any VCC. Thus, the reassembly coprocessor must assign each arriving cell to the proper VCC, thereby demultiplexing the incoming messages (See Figure 5-1).

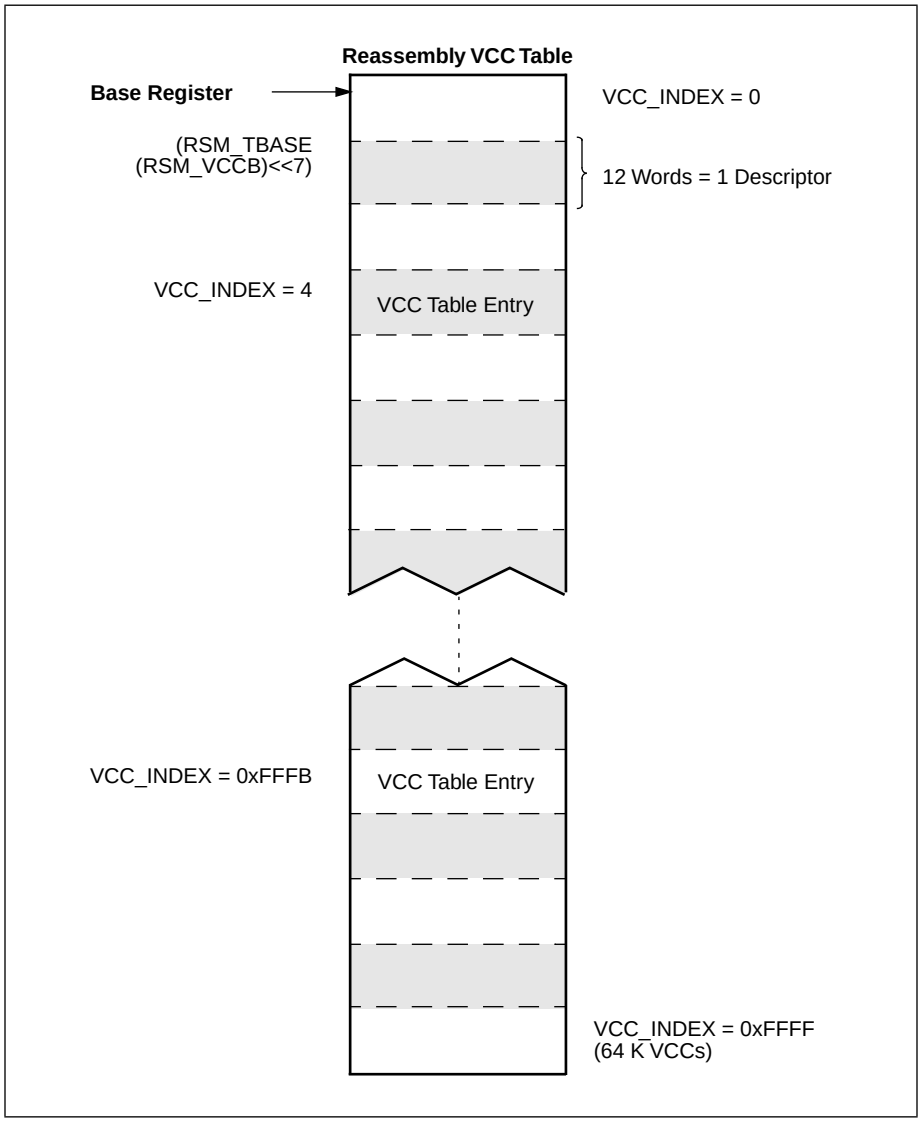
Figure 5-1. Reassembly, Basic Process Flow



5.2.1 Reassembly VCCs

As with segmentation VCCs, the Bt8233 supports up to 64K reassembly VCCs, referenced by VCC_INDEX, which identifies a location in the Reassembly VCC shown in Figure 5-2.

Figure 5-2. Reassembly VCC Table



Each entry in the Reassembly VCC Table consists of 12 words and describes a single VC. Each VC may be processed as either AAL5 or AAL0. AAL0 VCs may optionally be treated as Virtual FIFOs.

5.2.1.1 Relation to Segmentation VCCs

The Reassembly VCC index assignment is independent from the assignment of Segmentation VCC indexes. A full duplex connection may have a Segmentation VCC_INDEX of 0x100, while its receive channel has a Reassembly VCC_INDEX of 0x800. This is especially important when a VP is represented by a single Segmentation VCC_INDEX, but each of its VCs is represented by its own distinct Reassembly VCC Table entry.

For several operations, most notably ABR, the Bt8233 provides a method to associate a Reassembly VCC with a segmentation channel. The SEG_VCC_INDEX field in the Reassembly VCC Table allows one or many reassembly channels to correlate to one specific Segmentation VCC.

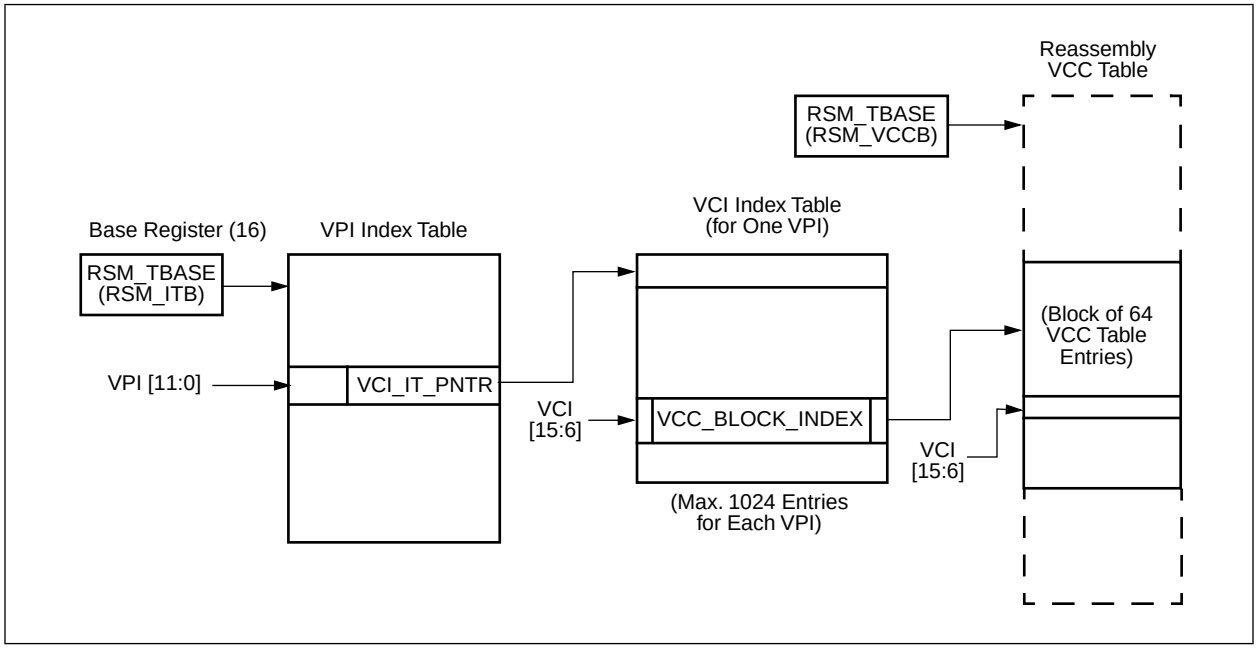
5.2.2 Channel Lookup

The Bt8233's reassembly coprocessor implements a VPI/VCI Index Table mechanism utilizing direct index lookup. This is done in order to assign each cell to a virtual channel, based on its VPI/VCI value. Each channel is thus identified by its internally generated index value, the VCC_INDEX.

This VPI/VCI Table Index mechanism dynamically maps VPI/VCIs to concatenated index values. It simplifies user channel assignment, provides flexible provisioning for received traffic, and provides fast, consistent lookup times regardless of the VPI/VCI values. In addition, using VPI/VCI table indexes minimizes the memory impact in preallocating large numbers of channels, by requiring that only the VCI Index Table entries be preallocated instead of the VCC Table entries.

Figure 5-3 illustrates the direct index channel lookup mechanism.

Figure 5-3. Direct Index Method for VP/VCI Channel Lookup



5.2.2.1 Setup

At system initialization, the user configures the Bt8233 to comply with either the 8-bit UNI VPI field or the 12-bit NNI VPI field. This is accomplished by setting the RSM_CTRL0 (VPI_MASK) bit to a logic high for UNI operation or a logic low for NNI operation. The configuration determines whether the Bt8233 will treat the upper nibble of the first header octet of each received cell as the GFC field (in the UNI VPI definition), or as an extension of the VPI address. This gives an address range for VPIs of either 256 entries (for UNI) or 4096 entries (for NNI). This sets the VPI Index Table size and dictates the number of VCI Index Tables to be allocated.

At system initialization, the user can also limit the valid range of both VPI and VCI addresses to be processed, to reduce the memory size of the lookup structures. VPIs are limited by VP_EN, and VCIs are limited by VCI_RANGE in the VPI Index Table.

VPI/VCI address pairs can now be preallocated in groups by mapping VCI Index Table entries to blocks in the Reassembly VCC Table.

Once the reassembly process has been initiated, additional channels, Switched Virtual Circuits (SVCs) can be dynamically allocated with simple “on-the-fly” index updates.

5.2.2.2 Operation

Upon reception of a cell, the reassembly coprocessor uses the VPI field as an index into the VPI Index Table, the base address of which is located at RSM_TBASE(RSM_ITB) * 0x80. The maximum allowed VPI value for UNI header operation is 255, and the maximum allowed VPI value for NNI operation is 4095, controlled by the RSM_CTRL0(VPI_MASK) field. If the VPI_MASK bit is a logic high (indicating UNI header operation), the four most significant bits of the ATM header are ANDed with “0000” before checking if the VPI value exceeds 255. If the VPI value exceeds 255, the reassembly coprocessor discards the cell and increments the CELL_DSC_CNT counter. Otherwise, the Rsm coprocessor reads the VPI Index Table entry.

The VCI_RANGE field in the VPI Index Table entry is used to set the maximum allowed value of VCI[15:6] values for that VPI, thus setting the size of the VCI Index Table for that VPI. If the value of VCI[15:6] of the received VCI field in the ATM header is greater than the VCI_RANGE field in the VPI Index Table entry, or VP_EN is a logic low, the reassembly coprocessor discards the cell and increments the CELL_DSC_CNT counter.

The VCI_IT_PNTR indicates the base address of the VPI's VCI Index Table. The Bt8233 then reads the appropriate entry in the VCI Index Table. The address of the VCI Index Table entry is derived as follows:

```
VCI_IT_PNTR * 4 + VCI[15:6] * 4.
```

The VCC_BLOCK_INDEX in the VCI Index Table entry selects a contiguous block of 64 Reassembly VCC State Tables, offset from the base address of the Rsm VCC Table. The VCC_INDEX value is derived by concatenating the VCC_BLOCK_INDEX value with the VCI[5:0] bits from the received cell header. Thus, VCI[5:0] from the received header points to the Rsm VCC State Table entry for that VCC.

```
VCC_INDEX = (VCC_BLOCK_INDEX<<6) + VCI[5:0]
```

The reassembly coprocessor reads the first word of the VCC Table entry. If VC_EN is a logic low, the cell is discarded and the CELL_DSC_CNT counter is incremented. Optionally, the counter is not incremented if the AAL_TYPE field has a value of 11. VC_EN allows idle cells to be filtered if the PHY layer has not already done so.

If the channel is active, the Bt8233 increments the CELL_RCVD_CNT counter.

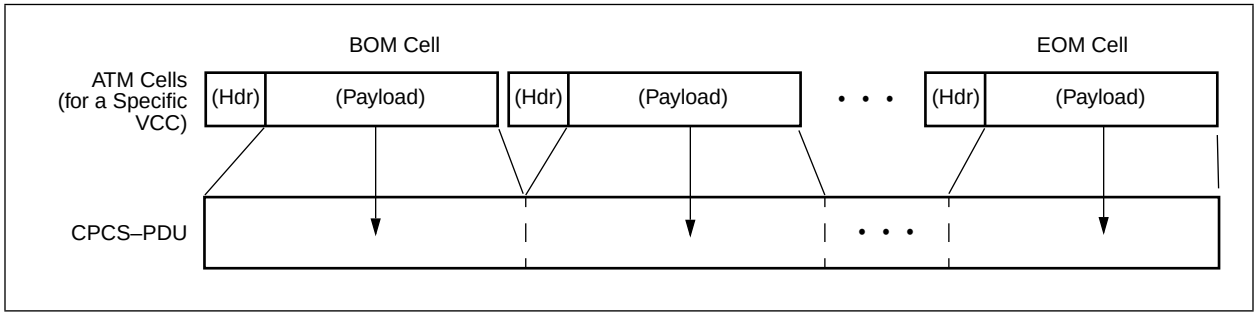
5.3 CPCS-PDU Processing

After the VCC has been identified via channel lookup, the reassembly coprocessor performs the appropriate CPCS-PDU processing according to the AAL_TYPE field in the Reassembly VCC Table.

The reassembly process is essentially the extraction and concatenation of consecutive ATM cell payloads on a specific VCC to form a CPCS-PDU. This processing is either reassembly into an AAL5 PDU, according to the specification in ANSI T1.635, or reassembly into a transparent AAL0 PDU. The exact process is governed by the AAL type detailed in the subsections below.

Figure 5-4 illustrates the basic process function.

Figure 5-4. CPCS_PDU Reassembly



5.3.1 Setup

Each active reassembly VCC must have a corresponding entry in the Reassembly VCC Table to describe its state. At channel setup time, the host allocates a Reassembly VCC Table entry and configures the VCC according to its provisional or negotiated characteristics. For Provisioned Virtual Connections (PVCs), the host performs these tasks upon system initialization. For Switched Virtual Connections (SVCs), it performs the tasks dynamically. This includes the AAL in use, its assigned buffer pools and allocation priority, and the associated Segmentation VCC Index (SEG_VCC_INDEX) for full duplex connections.

5.3.2 Operation

Once the reassembly process is activated, this Rsm VCC Table entry will be used to track the current state of the connection and direct the Bt8233 to perform specific functions as described throughout this chapter.

5.3.3 AAL5 Processing

Except for the EOM cell, all of the data within AAL5 cell payloads is user data. The reassembly coprocessor writes all user data to memory as described in Section 5.4, Buffer Management. The EOM cell contains both user data and CPCS-PDU overhead and delineates the end of an AAL5 PDU.

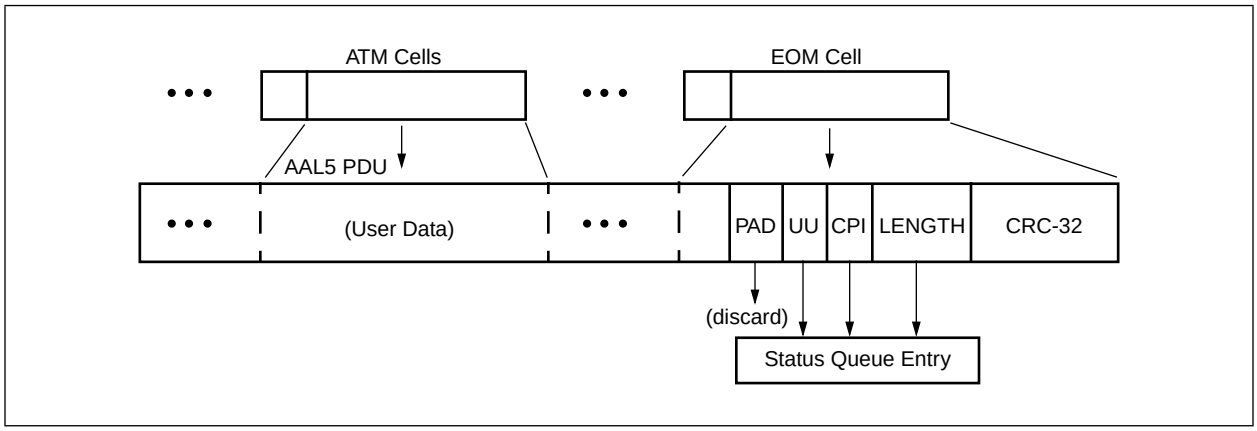
5.3.3.1 AAL5 COM Processing

During reassembly of the PDU, the reassembly coprocessor calculates a CRC-32 value on the received AAL5 PDU and counts the length of the PDU. The CRC-32 value is collected in an accumulator, and the LENGTH value is collected in a Length Counter. After each received cell is processed, the reassembly coprocessor writes the CRC-32 and LENGTH values to the Reassembly VCC State Table for that channel.

During reassembly of the AAL5 PDU, certain bytes of the PDU other than user data are written to a Status Queue entry for that PDU (see Figure 5-5). The Rsm coprocessor writes these specific fields to the Status Queue entry:

- User to User (UU) information
- Common Part Indicator (CPI) field
- LENGTH field

Figure 5-5. AAL5 EOM Cell Processing–Fields to Status Queue



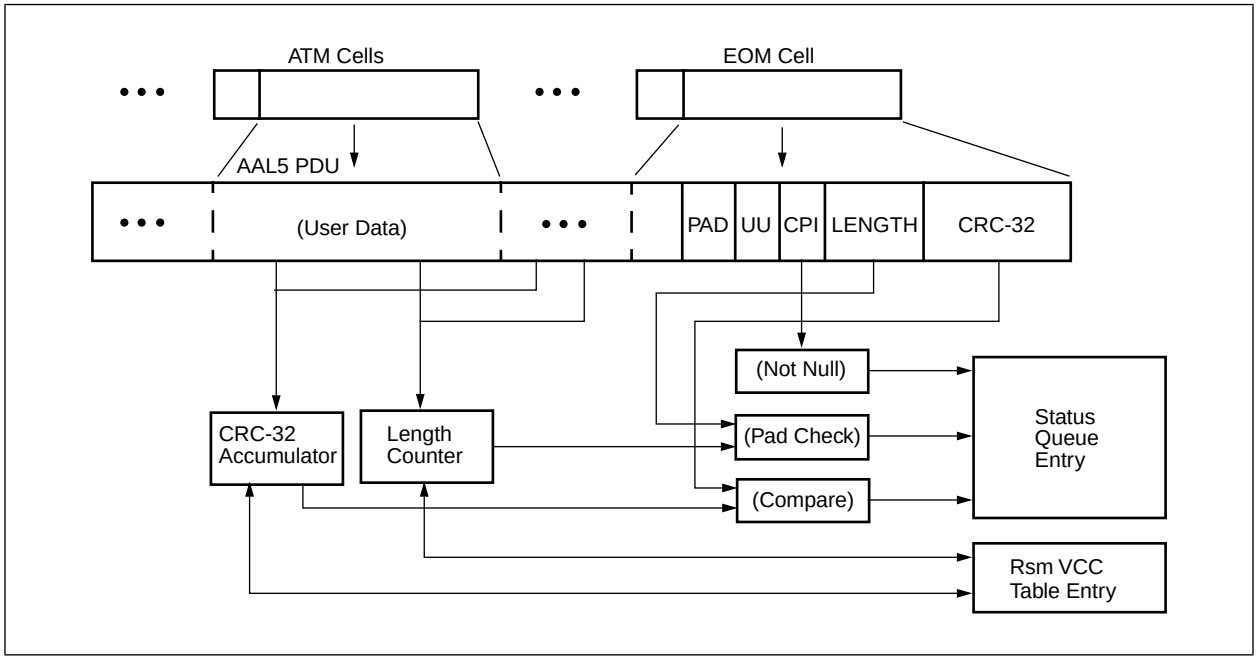
5.3.3.2 AAL5 EOM Processing

When the EOM cell is processed, the reassembly coprocessor performs the following checks:

- If the LENGTH field in the trailer of the AAL5 PDU is 0, the reassembly coprocessor sets the ABORT bit in the Status Queue entry to a logic high.
- The reassembly coprocessor compares the calculated CRCREM value to the CRC-32 value in the trailer of the AAL5 PDU. If different, the reassembly coprocessor sets the CRC_ERROR bit in the Status Queue entry to a logic high.
- The reassembly coprocessor compares the value collected in the Length Counter to the value in the LENGTH field in the trailer of the AAL5 PDU. If the number of Pad bytes is less than 0 or greater than 47, the PAD_ERROR bit in the Status Queue entry is set to a logic high.
- If the CPI field in the AAL5 trailer is not 0, the reassembly coprocessor sets the CPI_ERROR bit in the Status Queue entry to a logic high.

Figure 5-6 illustrates these process functions.

Figure 5-6. AAL5 Processing–CRC and PDU Length Checks



The Bt8233 reports all PDU termination events, with or without errors, in a status queue entry for that channel. See Section 5.6, Status Queue Operation, for full details.

5.3.3.3 AAL5 Error Conditions

The user can set a global variable for the reassembly coprocessor, RSM_CTRL0 (MAX_LEN), dictating maximum SDU delivery length. The maximum allowable length, in bytes, of any AAL5 CPCS-PDU including trailer and pad is:

min(RSM_CTRL0(MAX_LEN) * 1024, 65568).

During reassembly, this MAX_LEN value is checked to ensure that the PDU under reassembly does not exceed the maximum SDU delivery length.

If the Bt8233 receives a non-EOM cell, where TOT_PDU_LEN + 48 is greater than MAX_LEN * 1024 (or 65568), EPD is performed. The Bt8233 reports this condition via a Status Queue entry, with the LEN_ERROR and EPD status bits set. The AAL5_DSC_CNT counter is also incremented. Refer to Section 5.4.8, EPD, for details on how this process is handled.

For each EOM cell where TOT_PDU_LEN + 48 is greater than MAX_LEN * 1024 (or 65568), the PDU is completed, with BA_ERROR status bit set.

5.3.4 AAL0 Processing

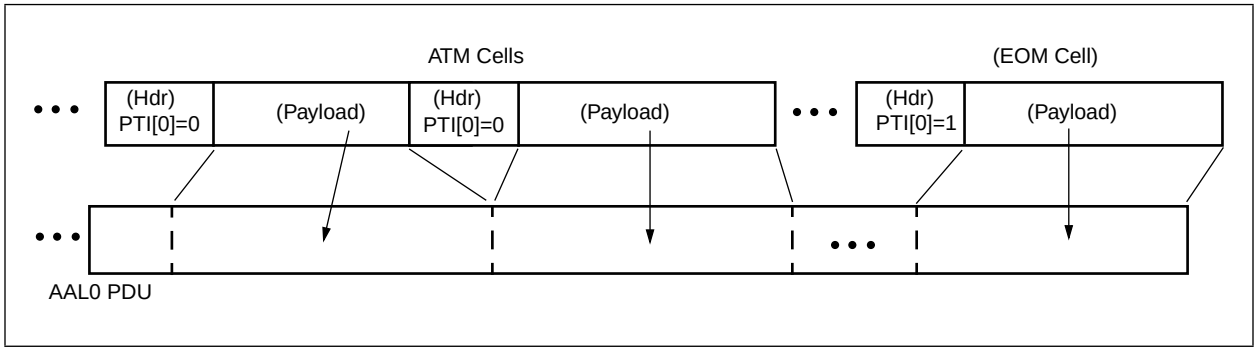
AAL0 is a transparent adaptation layer, allowing for pass-through of raw data cells during CPCS-PDU processing. AAL0 channels are intended to be used for AAL proprietary adaptation layers.

The Bt8233 provides two methods of terminating an AAL0 PDU -- Cell Count EOM and PTI Termination, as shown in Figure 5-7.

The TCOUNT field in the Rsm VCC State Table entry determines the method for each VCC.

- If TCOUNT is nonzero, Cell Count EOM PDU termination is enabled. PDUs will terminate when a fixed number of cells (TCOUNT) have been received. CCOUNT must be initialized to a value of 1 in this mode.
- If TCOUNT equals 0, then PTI Termination is enabled. In this case, a received cell with PTI[0] set to one indicates the end of the AAL0 message. The total maximum allowable length of an AAL0 PDU in this mode is: (CCOUNT * 2) bytes.

Figure 5-7. AAL0 PTI PDU Termination



The Bt8233 reports all PDU termination events, with or without errors, in a Status Queue entry for that channel. See Section 5.6, Status Queue Operation.

5.3.4.1 AAL0 Error Conditions

If the Bt8233 receives a non-EOM cell in PTI termination mode, where TOT_PDU_LEN + 48 is greater than CCOUNT * 2, EPD is performed. The Bt8233 reports this condition via a Status Queue entry, with the LEN_ERROR and EPD status bits set. Refer to Section 5.4.8, EPD, for details on how this process is handled.

For each EOM cell where TOT_PDU_LEN + 48 is greater than CCOUNT * 2, the PDU is completed, with BA_ERROR status bit set.

The Bt8233 processes error conditions for AAL0 (such as free buffer queue underflow, status queue overflow, and per-channel buffer firewall), in the same way as AAL5 CPCS-PDUs are processed.

5.3.5 ATM Header Processing

ATM level congestion indication and cell loss priority are mapped to the CPCS-PDU status queue entry in the following manner:

- LP: value of the ATM Header CLP bit ORed across all cells in a CPCS-PDU
- CI_LAST: value of ATM Header PTI[1] bit in last cell of CPCS-PDU
- CI: value of ATM Header PTI[1] bit ORed across all cells in a CPCS-PDU

5.3.6 BOM Synchronization Signal

The STAT[1:0] output pins may be programmed to provide an indication that a BOM cell is being written across the PCI bus. Additional external circuitry could snoop the BOM cell for a service level protocol header and perform appropriate lookup as the CPCS-PDU is being reassembled. To configure the STAT pins, set the STATMODE field in the CONFIG0 register to 0x0A. The STAT output truth table is defined in Table 5-1:

Table 5-1. STAT Output Pin Values for BOM Synchronization

	STAT[1]	STAT[0]
NOT BOM	0	0
AAL5 BOM	0	1
AAL0 BOM	1	0
NOT USED	1	1

The STAT output pins are valid during a SAR PCI master write address cycle. External circuitry detects a BOM cell transfer by detecting a logic high on either STAT pin during a SAR PCI master write address cycle. External circuitry can then snoop the subsequent data cycles of the BOM cell transfer to extract the appropriate protocol overhead.

5.4 Buffer Management

Once CPCS-PDU processing has been implemented, the cell payloads are written to data buffers. Each channel retrieves the location of its buffers from one of 32 Free Buffer Queues. The reassembly coprocessor tracks the location of the buffers from the VCC Table entry for that channel.

NOTE: The process cycle time of a read transaction across the PCI bus is much longer than a write transaction, due to the PCI bus being held in a busy state while the remote processor accesses and processes the read request. Therefore, in order to speed up processing flow during reassembly, the Bt8233 uses only control and status writes across the PCI bus between host and local systems.

Data buffers are supplied according to the mechanisms detailed below.

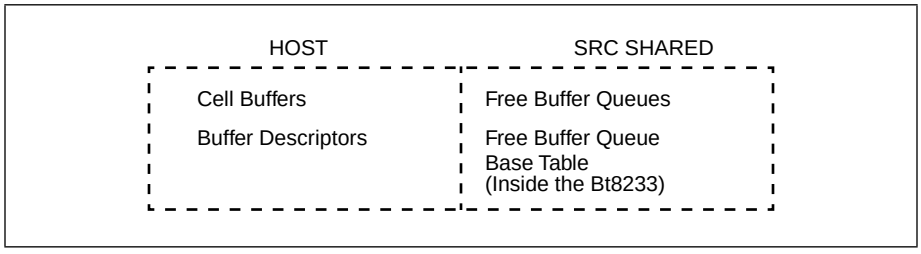
5.4.1 Host vs. Local Reassembly

Data buffers can reside in both host and SRC shared memory. The majority of user data traffic should be reassembled in host memory. SRC shared memory reassembly is intended for low bandwidth management and control functions, such as OAM and Signalling. This allows an optional local processor to off-load these network management functions from the host, focusing host processing power on the user application.

5.4.2 Scatter Method

The Bt8233 uses an intelligent scatter method to write cell payload data to host memory (see Figure 5-8). During reassembly to host memory, the reassembly coprocessor uses the DMA coprocessor to control the scatter function. The reassembly coprocessor controls the incoming DMA block during scatter DMA to host memory.

Figure 5-8. Host and Local Data Structures for Scatter Method



Four data structures are maintained: two in the host memory, one in SRC shared memory, and one in internal memory. The linked cell buffers (HCELL_BUFF) and Reassembly Buffer Descriptors reside in host memory, and the Free Buffer Queues (HFR_BUFF_QU) reside in SRC shared memory. The Free Buffer Queues also have an associated free buffer queue base table. This table is in internal memory. The Bt8233 allows for up to 32 independent free buffer queues.

5.4.3 Free Buffer Queues

5.4.3.1 Structure

The free buffer queue structure consists of a free buffer queue base table, two base address registers, RSM_FQBASE(FBQ0_BASE and FBQ1_BASE), and the corresponding free buffer queues.

The Reassembly VCC Table entry for any channel contains two 5-bit fields, BFR0 and BFR1. These fields identify the free buffer queues that have been assigned to this channel by the host during initialization of the VCC Table entry. BFR0 contains the BOM free buffer queue number, and BFR1 contains the COM free buffer queue number. Typically, the BFR0 number is for free buffer queues 0-15, and the BFR1 number is for free buffer queues 16-31.

The free buffer queue is configured in two banks. Bank 0 contains free buffer queues 0-15, and Bank 1 contains free buffer queues 16-31.

To disable this two tier buffer structure, the user can set BFR0 to equal BFR1

Depending on the type of arriving cell (whether BOM or COM), the corresponding BFRx buffer number is used as an index to the appropriate free buffer queue base table entry.

The base addresses for these banks are in RSM_FQBASE(FBQ0_BASE and FBQ1_BASE). The reassembly coprocessor calculates the address of the first entry for any of the 32 free buffer queues as follows:

```
FBQx_BASE + ((size of each free buffer queue) * BFRx MOD 16)
```

Each of the 32 free buffer queues is a circular queue whose entries are sequentially read by the SAR. The reassembly coprocessor calculates the index for each sequential read as follows:

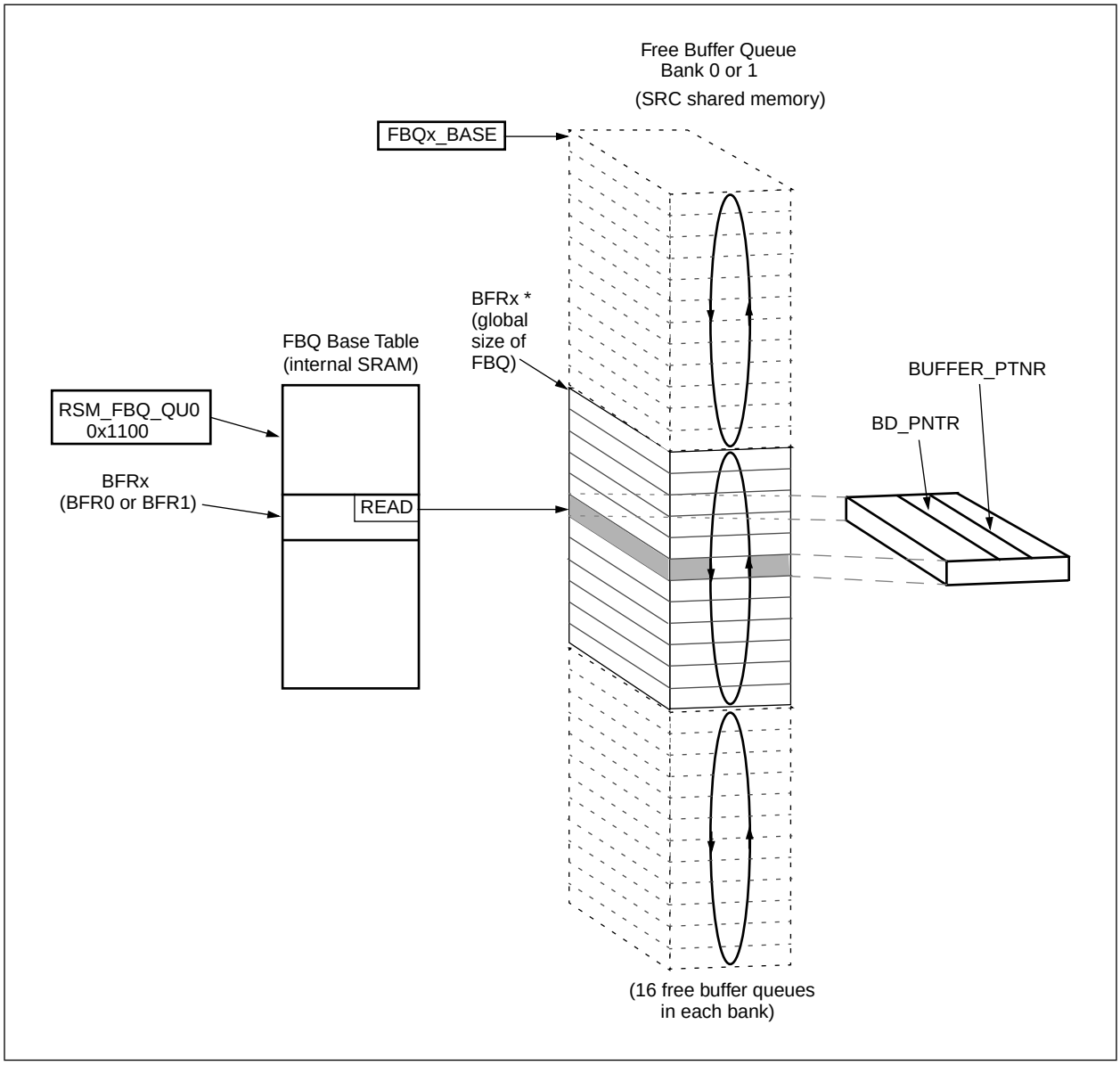
```
[index of first entry for the queue] + (READ index pointer *
        size of each free buffer queue entry)
```

The READ field in the base table entry for any free buffer queue is the current READ index pointer, and is continually updated with each read of that queue.

Each free buffer queue entry contains a pointer to a buffer descriptor (BD_PNTR) and a pointer to a data buffer (BUFFER_PNTR); when the free buffer queue entry is read it returns those pointers.

Figure 5-9 illustrates this structure. Refer to Chapter 3.0, Host Interface for more information on the operation of the Free Buffer Queue.

Figure 5-9. Free Buffer Queue Structure



5.4.4 Linked Data Buffers

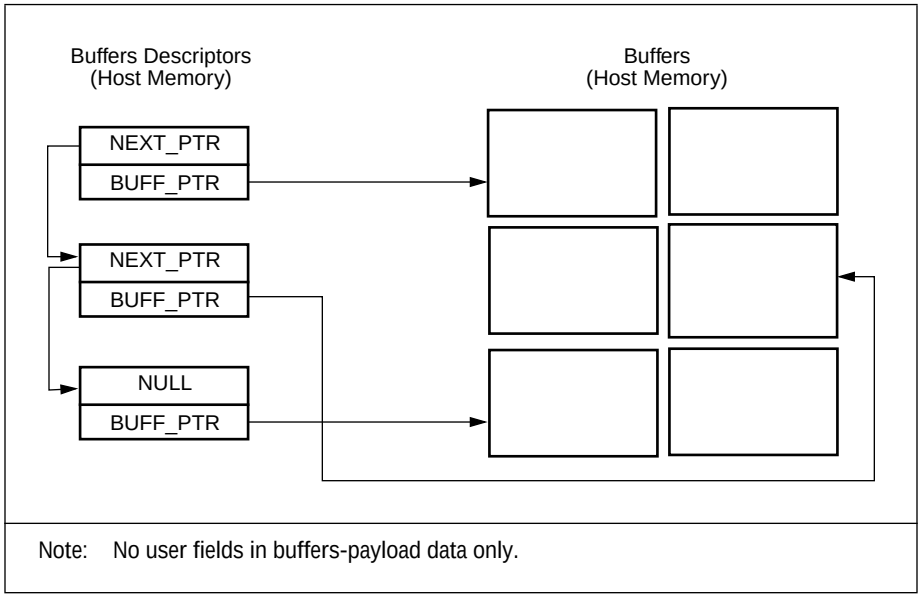
After the free buffer queue has returned pointers to a buffer descriptor and a cell buffer, the reassembly coprocessor writes payload data to the buffer.

The linked cell buffers contain the payload portions of the ATM cells. The buffers do not contain any control information. A pointer in a separate buffer descriptor structure links the buffers.

Thus, the pointer in the free buffer queue (BD_PNTR) points to a buffer descriptor which has a pointer (BUFF_PTR) pointing to a buffer. The use of this buffer locating mechanism offers a layer of indirection in buffer assignment that maximizes system architecture flexibility.

Figure 5-10 illustrates this structure.

Figure 5-10. Buffer Structures



Note: No user fields in buffers-payload data only.

The data buffers are linked by a pointer (NEXT_PNTR) in the first word of the buffer descriptor, which is written by the reassembly coprocessor when the current buffer is completed. The host writes the second word of the buffer descriptor to point to the next associated data buffer. The link pointer of the last buffer descriptor in a chain is written to NULL.

The link pointer (NEXT_PNTR) is not written if the LNK_EN bit in the Rsm VCC Table entry for that channel is a logic low.

NOTE: Only the data buffers are affected by big/little endian processing. The buffer control structures (i.e., the buffer descriptors, free buffer queue base table, and free buffer queues) are the same in both big and little endian modes.

5.4.5 Initialization of Buffer Structures

Before operation of the reassembly coprocessor is enabled, the host must initialize these buffer structures. The initialization in this section assumes that the firewall function is disabled (i.e., RSM_FQCTRL(FBQ0_RTN) = 0), and therefore all Free Buffer Queue entries are two words.

5.4.5.1 Buffer Descriptors

In every buffer descriptor entry, write the pointer to an available data buffer in the BUFF_PTR field. This assigns every data buffer to its own buffer descriptor.

5.4.5.2 Free Buffer Queue Base Table

Allocate the size (in number of entries) of each of the 32 free buffer queues in the RSM_FQCTRL register, (FBQ_SIZE) field, based on these values:

00 = 64
01 = 256
10 = 1024
11 = 4096

Initialize the free buffer queue update INTERVAL (i.e., how many buffers are taken off the free buffer queue before the Bt8233 writes the current READ index pointer to host memory). This is written to RSM_FQCTRL(FBQ_UD_INT).

Initialize the FORWARD, READ, UPDATE, and EMPT fields in each FB Queue Base Table entry to zeroes.

Initialize the READ_UD_PTR field in each base table entry with the appropriate address.

Write the appropriate length (in bytes) for the data buffers in that queue, in the LENGTH field of each free buffer queue base table entry.

Initialize BFR_LOCAL and BD_LOCAL to the appropriate host/SRC shared memory locations. Normally, both structures are in host memory.

5.4.5.3 Free Buffer Queue Entries

Write the base addresses of Free Buffer Queues Banks 0 and 1 in RSM_FQBASE(FBQ0_BASE and FBQ1_BASE).

For each allocated Free Buffer Queue entry, write the BD_PNTR and BUFFER_PTR corresponding to the buffer and buffer descriptor pair. Also write the VLD bit to a logic high.

For each unallocated Free Buffer Queue entry, write the VLD bit to a logic low.

5.4.5.4 Other Initialization

The user can globally disable free buffer underflow protection by setting RSM_CTRL(RSM_FBQ_DIS) to a logic high.

5.4.6 Buffer Allocation

The reassembly coprocessor performs buffer allocation when a new channel is being reassembled, or when a buffer on an existing channel in the process of being reassembled, is full.

The reassembly coprocessor reads the appropriate free buffer queue base table entry and free buffer queue entry. If the VLD bit is a logic low, a queue empty condition has occurred. If the VLD bit is a logic high, the reassembly coprocessor uses the assigned buffer to store payload data.

The VLD bit is then written to a logic low without corrupting the BD_PNTR value. The READ index pointer and UPDATE counter are incremented. If the UPDATE counter equals RSM_FQCTL(FBQ_UD_INT), then the READ index pointer is written to the location pointed to by READ_UD_PNTR, and the UPDATE counter is reset to zero.

When the host wants to return a buffer to a free buffer queue, the host WRITE index pointer is compared to the Bt8233 READ index pointer located at READ_UD_PNTR. If the WRITE index pointer + one is equal to the READ index pointer, an overflow condition has been detected and further processing is halted. Otherwise, the host writes and updates the free buffer queue entry with a new buffer pointer and buffer descriptor pointer and sets VLD bit to a logic high. The host then increments its WRITE index pointer.

5.4.7 Error Conditions

5.4.7.1 Free Buffer Queue Underflow or Empty Condition

An empty condition occurs when a buffer is needed, and there are no available buffers in the Free Buffer Queue. If the BFR1 queue is empty, and BFR1 does not equal BFR0, the Rsm coprocessor checks the BFR0 queue before declaring an empty condition.

If an empty condition occurs after the first buffer of a CPCS-PDU is written, the reassembly coprocessor will perform EPD on the channel and write a Status Queue entry with the EPD and Free Buffer Queue Underflow (UNDF) bits set to a logic high. EPD functions are described in Section 5.4.8, EPD. Also, if a buffer queue empty condition initially occurs at the beginning of a BOM cell, a Status Queue entry is written with UNDF set to a logic high and BD_PNTR null. In both cases, the RSM_HF_EMPT bit is set in the HOST_ISTAT1 and LP_ISTAT1 registers if the BD_LOCAL bit is a logic low in the Free Buffer Queue Base Table, or the RSM_LF_EMPT bit is set if BD_LOCAL is a logic high.

All cells of a PDU up to and including the next EOM cell are discarded. Upon receiving a BOM or SSM cell, the reassembly coprocessor checks the queue indicated by BFR0 for a valid free buffer. If a free buffer exists, the Rsm coprocessor stores the cell in the assigned buffer.

For AAL5 channels, the AAL5_DSC_CNT counter is incremented for each CPCS_PDU discarded during this error condition.

Channels that have outstanding buffers from an empty queue are not affected until they need a new buffer. Once the host has written more free buffers on the queue with VLD bit set to a logic high, the reassembly coprocessor will automatically recover from the empty condition.

5.4.8 EPD

The packet discard feature provides a mechanism to discard complete or partial CPCS-PDUs, based upon service discard attributes or error conditions.

5.4.8.1 General Description

The EPD function performs these basic functions:

- Halts reassembly of the CPCS-PDU marked for discard until the next BOM cell and the error condition has cleared
- Writes a Status Queue entry with the EPD bit set and other appropriate STATUS and PDU_CHECKS bits set, based on the reason for the discard.

5.4.8.2 Frame Relay Packet Discard

The frame relay discard attribute is contained in the BOM cell of a CPCS-PDU. If the FRD_EN bit in the Rsm VCC Table is a logic high, the frame relay packet discard function for that VCC is enabled, and the functions below are performed.

When the reassembly coprocessor receives a BOM cell on a VCC with this feature enabled, it checks the 1-bit DE field in the frame relay header. If this bit is a logic high, and the channel priority (DPRI in the Rsm VCC Table) is less than or equal to the global priority, RSM_CTRL (GDIS_PRI), the Rsm coprocessor discards the cell, marks the rest of the packet for discard, and increments the SERV_DIS counter in the VCC Table. All cells on that channel up to the next BOM are discarded.

If the SERV_DIS counter rolls over, the CNT_ROVR bit in the next status entry for this channel will be set to a logic high. The CNT_ROVR bit in the VCC Table holds this flag information until a status is sent.

5.4.8.3 CLP Packet Discard

If the CLPD_EN bit in the Rsm VCC Table is a logic high, the CLP Packet Discard function for that VCC is enabled, and the functions below are performed.

When the Rsm coprocessor receives a cell, and this function is enabled, it checks the 1-bit CLP field in the ATM header. If this bit is a logic high, and the channel priority (DPRI in the VCC Table) is less than or equal to the global priority, RSM_CTRL(GDIS_PRI), then the Rsm coprocessor discards the cell, marks the rest of the packet for discard, and increments the SERV_DIS counter in the VCC Table. All cells on that channel up to the next BOM are discarded.

If the SERV_DIS counter rolls over, the CNT_ROVR bit in the next status entry for this channel will be set to a logic high. The CNT_ROVR bit in the VCC Table holds this flag information until a status is sent.

5.4.8.4 LANE-LECID Packet Discard - Echo Suppression on Multicast Data Frames

The system designer can utilize this feature to discard superfluous traffic on the ATM network caused by LECIDs transmitting multicast frames (i.e., point-to-multipoint ELAN traffic).

If the LECID_EN bit in the Rsm VCC Table is a logic high, the LANE-LECID discard function for that VCC is enabled, and the functions below are performed.

The DPRI field is used as an index into the LECID table. This allows support for up to 32 LECIDs, each a unique identifier for a single LAN Emulation Client.

When the Rsm coprocessor receives a BOM cell with this function enabled, it checks the 16-bit LECID field in the LANE header against the value in the LECID table. If a match occurs, the Rsm coprocessor discards the cell, marks the rest of the packet for discard, and increments the SERV_DIS counter in the VCC Table.

If the SERV_DIS counter rolls over, the CNT_ROVR bit in the next status entry for this channel will be set to a logic high. The CNT_ROVR bit in the VCC Table holds this flag information until a status is sent.

5.4.8.5 DMA FIFO Full

The purpose of this function is to allow a graceful recovery from an incoming DMA FIFO full condition. Without this function the reassembly coprocessor is stalled when the FIFO is full, until recovery from the full condition. This causes the cells to be dropped indiscriminately on the upstream side of the reassembly block without any record of which VCCs the cells belonged to. Upon recovery from the full condition, cells belonging to corrupted PDUs continue to be processed, which wastes PCI bandwidth during the recovery phase. This function provides for a more efficient use of Host and SAR resources by allowing the reassembly block to process and drop cells during the full condition.

The reassembly block will mark all channels that receive a cell during the full condition for subsequent EPD. Upon recovery from the full condition, the reassembly block performs EPD on the appropriate channels as cells are received on those channels. In addition, cells will continue to be dropped on each channel until after an EOM cell is received for that channel. EPD processing is delayed until recovery from the full condition, since the status entry also requires the use of the incoming DMA FIFO.

This function is enabled by setting the FF_DSC bit in each VCC entry to a logic high.

NOTE: The user may want to disable this function if the free buffers, buffer descriptors, and Rsm status queues reside in SRC shared memory.

Similarly, RSM_CTRL1(OAM_FF_DSC) should be set to a logic high. OAM_FF_DSC is only valid if RSM_CTRL1(OAM_QU_EN) is set to a logic high.

NOTE: The user may want to disable this function if the global OAM buffers, buffer descriptors, and status queues reside in SRC shared memory.

EPD due to a FIFO full condition is indicated by the FFPD bit in the Rsm Status Queue entry being a logic high.

5.4.8.6 Error Conditions

Partially reassembled CPCS-PDU's will be recovered for the following error conditions:

- Non-EOM Max PDU Length exceeded
- Free Buffer Queue underflow
- Status Queue overflow.

5.4.9 Hardware PDU Time-Out

The Bt8233 automatically detects active CPCS-PDU time-out for reassembly channels. A PDU time-out occurs when a partially received PDU does not complete within a set time period. When it detects this time-out condition, the Bt8233 provides a status queue indication to the host. This indication allows the host to recover the buffers held by the partially completed PDU. The Bt8233 supports up to eight reassembly time-out periods.

5.4.9.1 Reassembly Time-Out Process

A background hardware process performs the reassembly time-out function. The process is activated at a user-selected interval. The process is globally enabled by setting the GTO_EN bit in the RSM_CTRL0 register. The RSM_TO register controls the process activity once enabled. The process is activated every RSM_TO_PER rising edges of SYSCLK on cell boundaries. Note that GTO_EN set to zero resets the internal time-out interrupt counter.

Each time the process is activated, it examines a single VCC, identified by TO_VCC_INDEX. This is a 16-bit variable located at address 0x1350 in internal SRAM. The host should initialize this register to zero at system initialization.

To enable hardware time-out on an individual VCC, the host must set TO_EN in the VCC Table Entry. The host also assigns one of eight time-out periods to each VCC by initializing the TO_INDEX field in the VCC Table Entry.

The Bt8233 checks the TO_EN bit and the active PDU indicator bit, ACT_PDU, to see if time-out processing is enabled and necessary, respectively, for the current connection. If either bit is zero, then TO_VCC_INDEX is incremented by one and compared to RSM_TO_CNT in the RSM_TO register. If TO_VCC_INDEX = RSM_TO_CNT, then TO_VCC_INDEX is reset to zero, and the time-out search is restarted at the beginning of the VCC Table.

If both bits are set, the Bt8233 increments CUR_TOCNT in the Rsm VCC Table entry. Then it compares CUR_TOCNT to the time-out value selected, TERM_TOCNTx, where x = TO_INDEX. TERM_TOCNT0 through TERM_TOCNT7 are located at address 0x1340 through 0x134c in internal SRAM. They must be initialized to appropriate values during system initialization.

If CUR_TOCNT = TERM_TOCNTx, then a time-out condition has occurred on the current VCC. The Bt8233 follows the procedure described in Section 5.4.9.4, Reassembly Time-Out Condition.

5.4.9.2 Halting Time-Out Processing

The host should set the TO_LAST bit in the Rsm VCC Table entry for the last VCC_INDEX for which time-out processing is enabled. When the Bt8233 detects this bit set to one, it halts time-out processing.

When time-out processing is halted, the time-out process will still be activated, but the VCC will not be checked for a time-out condition. The Bt8233 will simply increment TO_VCC_INDEX and compare it to RSM_TO_CNT. If they are equal, TO_VCC_INDEX is reset to zero, and the full time-out processing is re-enabled.

5.4.9.3 Timer Reset

The Bt8233 reassembly time-out process increments the CUR_TOCNT value. If it reaches a threshold value, a time-out condition has occurred. In AAL5 and AAL0, PTI termination modes, the reception of a non-EOM cell resets the counter.

5.4.9.4 Reassembly Time-Out Condition

The Bt8233 reports reassembly time-out conditions via the VCC's Reassembly Status Queue. The TO bit in the STATUS field of the Status Queue Entry will be set to one. In message mode, the BD_PNTR will point to the beginning of the partial buffer descriptor chain. In streaming mode, the BD_PNTR will point to the last buffer descriptor in the chain. The only other valid fields in the status queue entry will be VCC_INDEX and VLD.

Once status has been reported, the Bt8233 re-initializes the VCC Table entry to begin accepting a CPCS-PDU.

5.4.9.5 Time-Out Period Calculation

The following equation determines the time-out period of a VCC.

```
Period = SYSCLK period * RSM_TO_PER * RSM_TO_CNT *
         TERM_TOCNTx
```

RSM_TO_CNT must be greater than or equal to the maximum number of VCCs which require time-out processing.

5.4.10 Virtual FIFO Mode

This mode provides a logical FIFO port for cell data to host memory. Its principal use is for AAL–CBR voice traffic.

5.4.10.1 Setup

To enable this mode on any channel, set FIFO_EN in the Rsm VCC Table to a logic high. The user initializes the CBUFF_PNTR field in the Rsm VCC Table to the address of the FIFO port. The channel should also be configured for AAL0 fixed length termination mode, with a termination length of 1 cell and LNK_EN = 0.

5.4.10.2 Operation

Whenever a buffer is required, during reassembly in this mode, the CBUFF_PNTR address is used without accessing the free buffer queue.

No status entries are written in this mode since there is no way to maintain synchronization between status entries and cells in the FIFO under FIFO overflow conditions.

5.4.10.3 Errors

When the FIFO port is on the PCI bus, the CBUFF_PNTR address must be on a 64-byte boundary, and a decode of any address in the 64-byte block will access the FIFO. External circuitry must also ensure that only complete cells are written into the host FIFO.

The beginning of a cell transfer can be detected by the PCI address being 64-byte aligned.

5.4.11 Firewall Functions

Implementation of multiple Free Buffer Queues and EPD performs a firewalling functionality on a group basis.

The user can also set up per-VCC firewalling on a channel-by-channel basis. The firewall mechanism allows the user to allocate buffer credits on a per-channel basis.

5.4.11.1 Setup

Set RSM_FQCTRL(FBQ0_RTN) to a logic high. This sets free buffer queue block 0 to contain queues with four-word entries. This is used to support per-VCC firewall credit update.

Set the global firewall control bit to a logic high in register RSM_CTRL0, field (FWALL_EN), to globally enable firewall processing on a per-channel basis.

Set the following fields of the VCC Table entry for the channel being set up for firewall processing:

- Set the FW_EN bit to a logic high to enable firewall processing on that channel.
- Set RX_COUNTER[15:0] to assign the initial buffer credit for the channel.

Initialize the FORWARD fields in the Free Buffer Queue base tables to point to the entry where credit will initially be returned. Typically, this will be the first entry after the initial buffers placed on the queue. Write the FWD_VLD bit in all Free Buffer Queue entries to a logic low.

5.4.11.2 Operation

During reassembly on a channel enabled for firewall processing, whenever a buffer is taken off free buffer queues 0 through 15, the Rsm coprocessor decrements the RX_COUNTER[15:0] in the RsmVCC Table entry for that channel. This allows COM buffers to be placed on queues 16 through 31 and not be firewalled.

If the RX_COUNTER[15:0] for a channel is zero when a buffer is required, then the Rsm coprocessor declares a firewall condition. If the firewall condition occurs on a BOM or SSM, the Bt8233 writes a status queue entry with the FW bit set and a NULL in the BD_PNTR field.

If the firewall condition occurs on a COM or EOM, the Rsm coprocessor initiates EPD and writes a status queue entry with the FW and EPD bits set. It then discards cells on that channel until the channel has recovered from the firewall condition.

All AAL5 PDUs discarded under the firewall condition cause the AAL5_DSC_CNT counter to be incremented. Recovery occurs only on a BOM or SSM cell when the credit is rechecked.

5.4.11.3 Credit Return

The user returns credit, at the same time the buffer is recovered to the Free Buffer Queue, by writing the third word of the Free Buffer Queue. The VCC_INDEX is written to the channel to which credit is returned. The FWD_VLD bit is set to a logic high, and the QFC bit is set to a logic low. The Rsm coprocessor increments the RX_COUNTER[15:0] of the applicable channel. For proper operation of the update interval function, buffers must be returned at the same time as credits are returned.

Credits are returned to VCCs through Bank 0 Free Buffer Queues. In order to return buffer credits independently from buffer usage, the Bt8233 maintains a separate read pointer into free buffer queues which return credits. This pointer name is FORWARD, in the Free Buffer Queue Base Table Entry. The host determines the number of Bank 0 Free Buffer Queues which return credits by setting FWD_EN in the RSM_FQCTRL register.

The Bt8233 snoops writes to Free Buffer Queues that return firewall credits. When a write completes, the Bt8233 will begin processing firewall return credits on that queue. The third word of each entry will be read, and if FWD_VLD is set, a credit will be added to the VCC_INDEX indicated. The Bt8233 will continue to process credit return entries until FWD_VLD is zero. Multiple Free Buffer Queues may have credit return entries outstanding at one time. The Bt8233 will process the entries according to the priority set in FWD_RND in the RSM_FQCTRL register. If FWD_RND is a logic low, the Bt8233 will exhaust the credit returns on the highest number active queue before proceeding to other queues. Otherwise, it will service the queues in round-robin order.

Before the reassembly coprocessor is enabled, the host must initialize the FORWARD read pointer to the first entry where credit will be returned. Typically, this will be the first entry after the initial buffers placed on the queue.

5.5 Global Statistics

To meet the requirements of ILMI (ATM Forum) and AToM (RFC1695) documents, three register-based counters are implemented:

- CELL_RCVD_CNT - Number of cells received that map to active channels
- CELL_DSC_CNT - Number of cells received that map to inactive channels. This includes idle cells since those channels will be turned off.
- AAL5_DSC_CNT - Number of AAL5 CPCS-PDUs discarded due to per channel firewall, buffer queue underflow, FIFO full packet discard, status queue overflow or maximum CPCS-PDU length exceeded on non-EOM cells.

The first two counters are implemented as 32-bit counters, and the third is a 16-bit counter. All three are set to zero upon a reset and are not reset to zero upon a read of the counter by the host. The counters roll over and optionally cause an interrupt upon rollover.

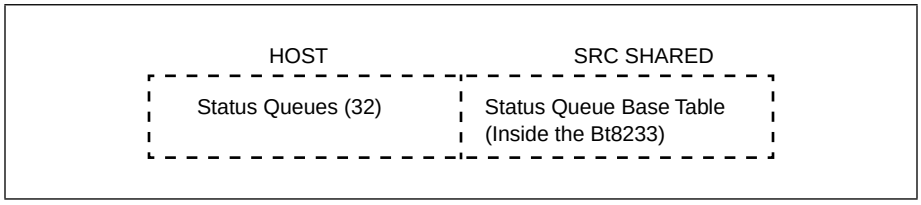
5.6 Status Queue Operation

The Bt8233 reports reassembly status to the host via the Reassembly Status Queue. The reassembly coprocessor normally writes a status queue entry when a complete CPCS-PDU has been reassembled. One field of the status queue entry (BD_PNTR) points to the first buffer descriptor in the linked list of buffer descriptors for that reassembled PDU. The rest of the fields of that status queue entry provide data on the status of the reassembled PDU, then used by the host in directing further processing.

5.6.1 Structure

Two data structures are maintained: one in host memory and one in SRC shared memory, as shown in Figure 5-11. The status queues (HSTAT_QU) reside in host memory, and the status queue base table resides in SRC shared memory, allowing for up to 32 independent circular status queues.

Figure 5-11. Data Structure Locations for Status Queues

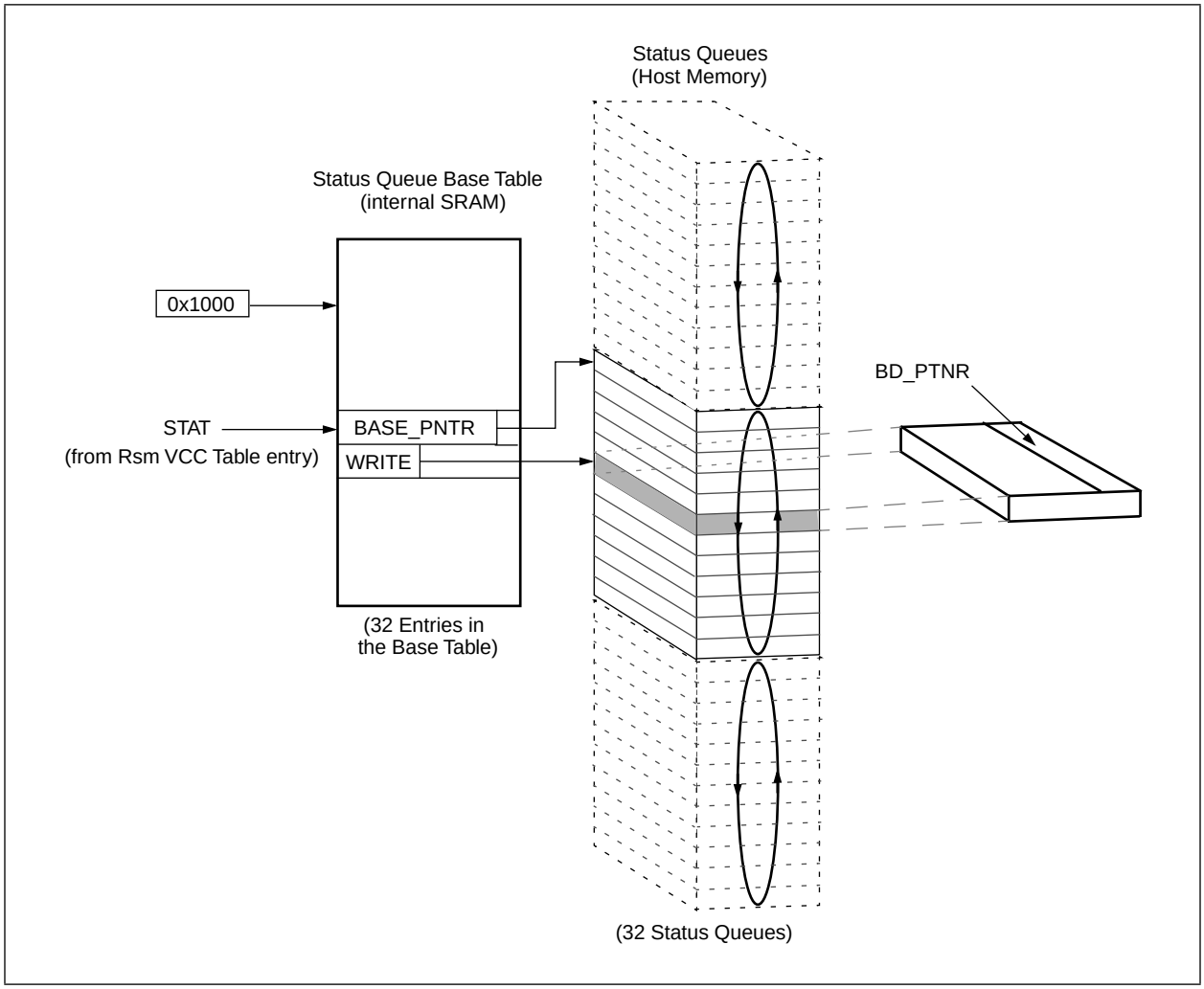


The status queue base table is located at 0x001000 in internal SRAM. The status queue base table contains status queue base information for up to 32 queues. The queues are accessed via a status pool number, the STAT field in the Rsm VCC Table. This field is used as an index to the correct status queue base table entry.

The BASE_PNTR field in the status queue base table entry points to the base address of the status queue associated with that status queue base table entry. The WRITE field is the index pointer maintained by the Bt8233, incremented each time a status queue entry is written, to point to the next status queue entry for that status queue.

Figure 5-12 illustrates this structure.

Figure 5-12. Status Queue Structure Format



5.6.2 Setup

At system initialization, set up the following fields in each of the Status Queue Base Table entries:

- Write the base address of the corresponding status queue in the BASE_PNTR field.
- Initialize the WRITE and READ_UD fields to zeroes.
- Set the SIZE field for the size of the corresponding status queue.
- Set the LOCAL bit to a status high if the status queue is in SRC shared memory; otherwise set the bit to a logic low.

In addition, initialize each status queue entry in all 32 status queues by setting the VLD bit to a logic low.

Initialize the READ pointers to zero for each status queue.

5.6.3 Operation

The reassembly coprocessor normally writes a status queue entry when a full CPCS-PDU has been reassembled.

Each time the Rsm coprocessor writes a status queue entry, it sets the VLD bit in the entry to a logic high and increments the WRITE pointer in the status queue base table entry for that status queue.

When the host processes the status queue, it reads entries based on the host READ pointer for that status queue. In order to maintain data coherency, the host reads only the VLD bit before reading any other word.

When the host finds the VLD bit set to a logic high, it directs the reassembly coprocessor to process the status queue entry, increment the host READ counter, and reset the VLD bit to a logic low. The host also periodically writes the READ counter value to the READ_UD field in the status queue base table entry for that queue.

When in message mode, the Rsm coprocessor writes a status entry at the completion of a CPCS-PDU. Optionally, a status entry can be written at both the beginning and end of a message. This allows the host to initiate protocol header processing in advance of receiving the complete message. The host can then traverse the linked cell buffers to collect the complete CPCS-PDU.

If the BINTR bit in the Rsm VCC State Table is a logic high, the Rsm coprocessor writes an additional status queue entry at the completion of the first buffer of a CPCS-PDU. This status queue entry is delineated by the BOM bit set and the EOM bit cleared. This allows the host to begin packet processing before reception of the complete CPCS-PDU. In this case only the BD_PNTR and VCC_INDEX fields are valid in that status queue entry.

The STM_MODE bit in the Rsm VCC Table is set to a logic high, activating Streaming Mode for that channel. In this mode, the Rsm coprocessor writes a status entry for each completed buffer. The BD_PNTR field in the status entry points to the corresponding buffer descriptor for that single buffer. Only the last status entry for that CPCS-PDU, with EOM bit a logic high, contains valid status data for that PDU.

The Rsm coprocessor writes a status queue entry for each received OAM cell.

Refer to Chapter 2.0, Bt8233 Architecture Overview for more detailed information on the operation of Status Queues.

5.6.4 Errors

The Rsm coprocessor also writes a status entry for several error conditions:

- Reassembly time-out
- Early packet discard
- Per channel firewall
- CPCS abort

To ensure that an error indication occurs even if no CPCS-PDUs are being reassembled on channels having free buffer queues in the empty state, a BOM cell will cause a status queue entry to be written. If a BOM cell is received, and no Early Packet Discard (EPD) have occurred on channels mapped to the empty free buffer queue, The following will occur:

- A status queue entry is written with the BOM and UNDF bits set to a logic high.
- Either the RSM_HF_EMPT bit in the HOST_ISTAT1 register or the RSM_LF_EMPT bit in the LP_ISTAT1 register is set to a logic high.

This status does not point to a linked list of buffer descriptors. This status will be written a maximum of once per free buffer queue empty condition.

5.6.5 Host Detection of Status Queue Entries

The host can use either a polling operation or an interrupt routine to detect new status queue entries.

To poll each status queue, the host continuously reads the VLD bit at the current READ position until it returns a logic high. The host then processes the status entry, writes the VLD bit to a logic low, and increments its current READ pointer. Periodically, the host writes the current READ index value into the READ_UD field of the Status Queue Base Table entry.

The host can also use an interrupt routine to process status queues. When the reassembly coprocessor writes a status queue entry into host memory, the HOST_ISTAT0 (RSM_HS_WRITE) bit is set to a logic high to prompt an interrupt. Upon receiving an interrupt, the host reads HOST_ST_WR (RSM_HS_WRITE[15:0]) to determine which host memory status queue(s) caused the interrupt. (Only status queues 0 through 15 are reported in this register.) A typical operation for the interrupt manager would be to only read HOST_ISTAT1 upon receiving an interrupt and periodically read HOST_ISTAT0 to ensure that no error conditions have occurred. Once the interrupt manager has determined which status queue(s) caused the interrupt, the host would start reading the appropriate status queues at their current read location. The host processes status entries until reading an entry with the VLD bit set to logic low. Again, the host periodically writes the current READ index value into the READ_UD field of the Status Queue Base Table entry.

5.6.6 Status Queue Overflow or Full Condition

A status queue overflow or full condition is entered when the last available status queue entry is written. The reassembly coprocessor detects the condition by comparing the WRITE and READ_UD index pointers in the corresponding status queue base table. Upon detecting a status overflow condition, the Rsm coprocessor sets the internal OVFL bit in the last status queue entry written to a logic high to indicate the condition. To prompt an interrupt, the Rsm processor also sets either the RSM_HS_FULL bit in the HOST_ISTAT1 register on the RSM_LS_FULL bit in the LP_ISTAT1 register to one.

The Rsm coprocessor also sets to one either the RSM_HS_FULL bit in the HOST_ISTAT1 register or the RSM_LS_FULL bit in the LP_ISTAT1 register, to prompt an interrupt.

While the reassembly coprocessor is in status-full condition, it discards all cells. If a Continuation of Message (COM) or EOM cell is received while the status queue is full, the channel is marked for status full packet discard. When an Single Segment Message (SSM), EOM, or OAM cell is received during a status-full condition, the cell is discarded, and the status queue is checked. If there is now room in the status queue, the status full condition is exited.

For multiple peer configurations, an interrupt manager can be configured by the user to detect the full condition and advise the peers to check if their queues have overflowed. Each peer would then check the OVFL bit in the last status queue entry written (pointed to by READ_UD - 1), to determine if that peer's status queue has filled. If the OVFL bit is not set to a logic high, the host should also check the entry pointed to by (READ_UD2 - 1) to determine if an overflow condition occurred during a host update of the READ_UD index pointer. Since the reassembly coprocessor recovers from the overflow condition automatically, the host does not have to determine which queue overflowed.

After a status group has exited a full condition, the Rsm coprocessor will perform EPD on channels marked for packet discard due to the status overflow condition, when a cell is received on any of those channels. Cells up to and including the next EOM will also be discarded.

Status Queue overflow protection can be globally disabled by setting RSM_CTRL0(RSM_STAT_DIS) to a logic high.

5.7 Reassembly Control and Status Structures

5.7.1 Channel Lookup Structures

The reassembly coprocessor uses a VPI/VCI Table Index mechanism employing direct index lookup. In this way, each arriving cell can be assigned to a virtual channel, based on its VPI/VCI value.

Figure 5-13 illustrates this lookup mechanism.

Table 5-2 and Table 5-3 describe the VPI Index Table format, and Figure 5-4 and Table 5-5 describe the VCI Index Table format.

Figure 5-13. VPI/VCI Channel Lookup Structure

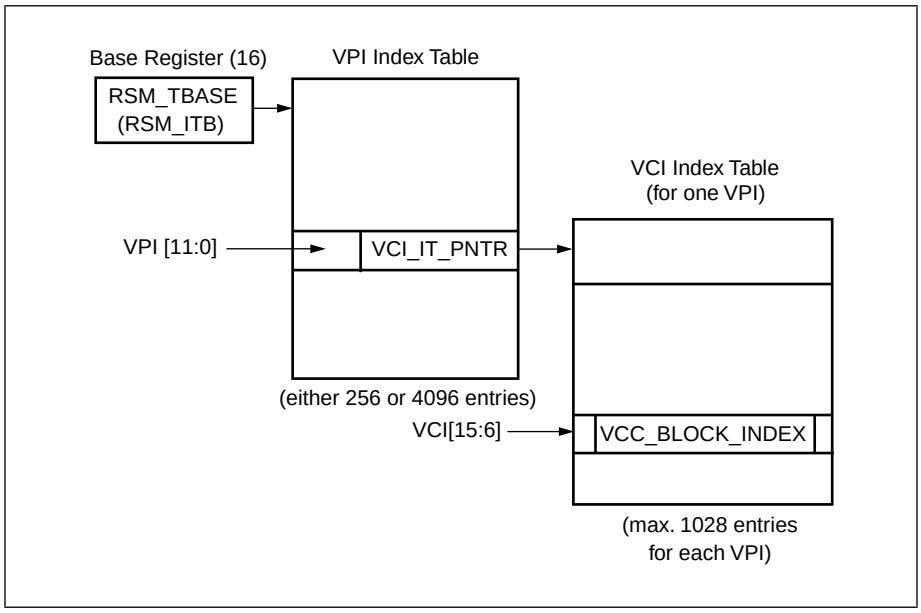


Table 5-2. VPI Index Table Entry Format

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	VP_EN	VCI_RANGE										VCI_IT_PNTR																				

Table 5-3. VPI Index Table Entry Descriptions

Field Name	Description/Function
VP_EN	Enables the VPI for lookup processing. If not enabled, cell is discarded, and the CELL_DSC_CNT counter is incremented.
VCI_RANGE	Determines the maximum VCI[15:6] value allowed. If cell VCI exceeds maximum, cell is discarded, and counted as CELL_DSC_CNT.
VCI_IT_PNTR	VCI Index Table Base Pointer. Points to the base of the VCI Index Table for the VPI by appending 2 least significant zero bits to form a byte address.

Table 5-4. VCI Index Table Format

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Reserved																VCC_BLOCK_INDEX										Reserved					

Table 5-5. VCI Index Table Descriptions

Field Name	Description/Function
VCC_BLOCK_INDEX	VCC block index. Index to the block of 64 Rsm VCC Table entries allocated to VCI[15:6]. VCC_BLOCK_INDEX is concatenated with the 6 least significant bits of the VCI to form the index into the Rsm VCC Table to access the VCC Table entry for that channel.

5.7.2 Reassembly VCC Table

Each Reassembly VCC Table entry occupies one 11-word descriptor of the Reassembly VCC Table.

There are 2 basic formats for the Reassembly VCC Table entry -- AAL5 and AAL0. Each completely describes the state of the reassembly process for individual VCCs.

Figure 5-14 illustrates the VCC Table entry lookup mechanism as a continuation from Figure 5-15.

Table 5-6 and Table 5-7 describe the format of AAL5 and AAL0 Reassembly VCC Table entries. Table 5-8 through Table 5-10 provide field bit definitions for PDU_FLAGs, ABR_CTRL, and AAL_EN, respectively. Table 5-11 provides Reassembly VCC Descriptions.

Figure 5-14. Reassembly VCC Table Entry Lookup Mechanism

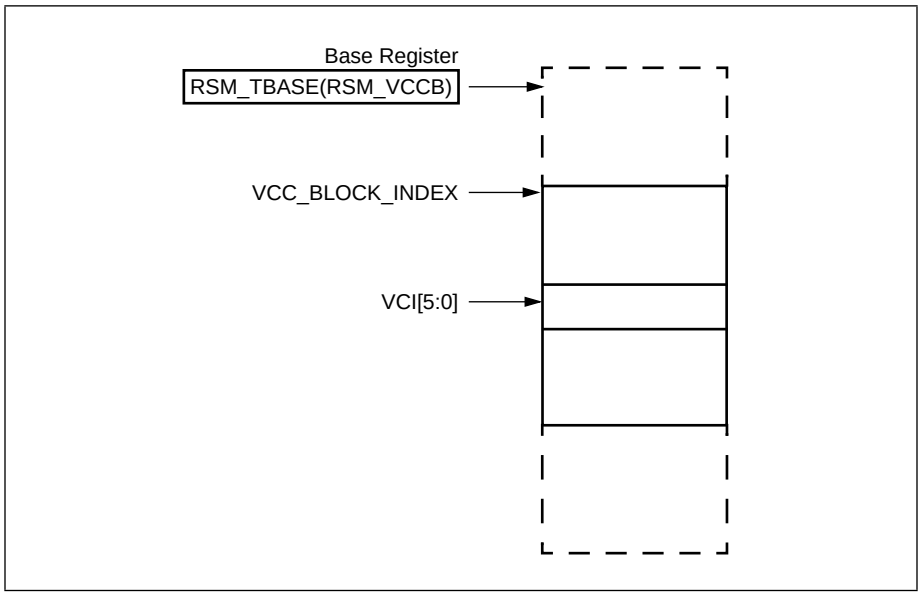


Table 5-6. Reassembly VCC Table Entry Format–AAL5

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	VC_EN	AAL_TYPE		DPRI					Reserved			FF_DSC	TO_INDEX			PM_INDEX							AAL_EN									
1	TO_LAST	TO_EN		CUR_TOCNT													Rsvd	Reserved					ABR_CTRL									
2	PDU_FLAGS										Reserved						Reserved															
3	CRCREM																															
4	Reserved																Rsvd	STAT					BFR1					BFR0				
5	Reserved																															
6	Reserved																														Rsvd	
7	Reserved																														Rsvd	Rsvd
8	SEG_VCC_INDEX																SERV_DIS															
9	Reserved																RX_COUNTER/VPC_INDEX															
10	Reserved																															
11	Reserved																															

Notes: 1. = Written by host at VCC setup.
2. = May be dynamically modified during active segmentation.

Table 5-7. Reassembly VCC Table Entry Format–AAL0

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	VC_EN	AAL_TYPE		DPRI					Reserved				TO_INDEX			PM_INDEX							AAL_EN									
1	TO_LAST	TO_EN		Reserved													RSVD	Reserved														
2	PDU_FLAGS								Reserved								TOT_PDU_LEN															
3	COUNT																TCOUNT															
4	CBUFF_LEN																RSVD	STAT					BFR1							BFR0		
5	Reserved																															
6	Reserved																														Rsvd	
7	Reserved																														RSVD	RSVD
8	SEG_VCC_INDEX																SERV_DIS															
9	Reserved																RX_COUNTER/VPC_INDEX															
10	Reserved																															
11	Reserved																															

Notes: 1. = Written by host at VCC setup.
2. = May be dynamically modified during active segmentation.

Table 5-8. PDU_FLAGS Field Bit Definitions

31	30	29	28	27	26	25	24	23	22
CNT_ROVR	SFPD_PND	EPD	CI	CLP	BFR_LOCAL	BD_LOCAL	ACT_PDU	BOM_BUF	FFPD_PND

Table 5-9. ABR_CTRL Field Bit Definitions

7	6	5	4	3	2	1	0
ER_EN	Reserved	Reserved	ABR_VPC	Reserved	Reserved	Reserved	Reserved

Table 5-10. AAL_EN Field Bit Definitions

9	8	7	6	5	4	3	2	1	0
PM_EN	FIFO_EN	LNK_EN	FW_EN	M52_EN	BINTR	STM_MODE	LECID_EN	FRD_EN	CLPD_EN

Table 5-11. Reassembly VCC Table Descriptions *(1 of 2)*

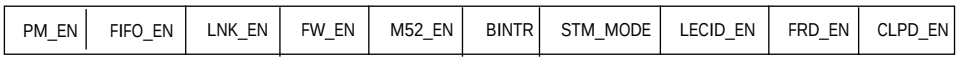
Field Name	Description/Function
VC_EN	Enables the VCC table entry. If disabled, the cell is discarded.
AAL_TYPE	When VC_EN is a logic high, configure channel to process specific AA; else, when VC_EN is a logic low, a value of '11' will cause the CELL_DSC_CNT counter not to be incremented 00–AAL5 01–AAL0 10–Reserved 11–Reserved
DPRI	Discard Priority value. Compared against global priority in CLP and Frame Relay discard modes to determine discard eligibility. In LANE-LECID echo suppression mode, this field is the index into the LECID table that holds channel LECID values.
FF_DSC	FIFO Full Discard. When a logic high, cells will be discarded when the Incoming DMA FIFO is almost full. This includes OAM cells when RSM_CTRL1(OAM_QU_EN) is a logic low.
TO_INDEX	Selects one of eight INIT_TOCNT values in internal SRAM.
PM_INDEX	Pointer to a PM OAM processing word. Index with reference to top of VPI Index table.
AAL_EN	Enable various cell processes. The AAL_EN field contains the following control bits:  PM_EN—Enable PM OAM processing on this channel. FIFO_EN—Enable Logical FIFO mode. LNK_EN—Enable writing of Buffer Descriptor NEXT field. FW_EN—Enable firewall processing. Used in conjunction with FWALL_EN bit in RSM_CTR0. M52_EN—If set high, all 52 octets of the cell are written to a cell buffer. BINTR—Enable interrupt after BOM buffer filled in message mode. STM_MODE—Enable streaming mode. LECID_EN—Enable LANE-LECID echo suppression. FRD_EN—Enable frame relay DE mode. CLPD_EN—Enable CLP discard mode.
PDU_FLAGS	Set various flags related to PDUs. The PDU_FLAGS field contains the following control bits:  CNT_ROVR—Indication that SERVICE_CNT counters have rolled over. The next status entry will indicate this condition. SFPD_PND—Status Full Packet Discard Pending. Set when status queue is full, and CPCS needs to be discarded when status entries available. EPD—Early Packet Discard Flag. Set when early packet discard occurs on a channel. Cleared when new packet starts and error condition cleared. CI—Congestion Indication. PTI[1] header bit ORed across the CPCS-PDU. CLP—Cell Loss Priority. CLP header bit ORed across the CPCS-PDU. BFR_LOCAL—Buffer Local. If high, the current cell buffer is located in SRC shared memory; otherwise, host memory. SAR maintains this bit. BD_LOCAL— Buffer Descriptor Local. If high, the buffer descriptors reside in SRC shared memory; otherwise, host memory. SAR maintains this bit. ACT_PDU—Active PDU. Indication that at least one buffer has been taken off of the free buffer queue for the current PDU being received. BOM_BUF—BOM buffer flag. Set high when filling the first buffer of a PDU. FFPD_PND—DMA FIFO Full Packet Discard Pending.

Table 5-11. Reassembly VCC Table Descriptions *(2 of 2)*

Field Name	Description/Function
ABR_CTRL	Set various control bits related to QFC. The QFC_CTRL field contains the following control bits: ER_EN | Reserved | ABR_VPC | Rsvd | Rsvd | Rsvd | Rsvd ER_EN—Enable ER operation. ABR_VPC—Indicates ER connection is VPC (0 indicates VCC). For VPC operation, all VCC entries in the VPC group except VCI=6, must be initialized with VPC_INDEX pointing to the VCC entry corresponding with VCI=6. The VCI=6 entry contains the integrated EFCI bit over the VPC group. Also, all entries in the VPC group including VCI=6 must set the ABR_VPC bit to a logic high.
TOT_PDU_LEN	Total PDU length in bytes.
TO_LAST	Indicates the last VCC table entry to process for time-out.
TO_EN	Enable time-out processing on the channel.
CRCREM	Cycle Redundancy Check Remainder. CRC32 remainder used in AAL5 only.
CCOUNT	Termination Cell count. Used in AAL0 to terminate packet. When PTI termination mode is enabled, the maximum total length of a CPCS-PDU is CCOUNT * 2 bytes. In fixed length mode, initialize to 1.
TCOUNT	Termination Count. Used in AAL0 to determine the number of cells in a packet. If this field is zero in AAL0 mode, then PTI termination mode is enabled.
STAT	Status Queue Pool Number.
BFR1	COM Free Buffer Queue Pool Number.
BFR0	BOM Free Buffer Queue Pool Number.
CBUFF_PNTR	Current Buffer Pointer. Pointer to the current unused position of the cell buffer where cell payload data may be written. In Logical FIFO Mode, the address of the FIFO.
SEG_VCC_INDEX	Channel index of corresponding segmentation channel. Used by ER and PM-OAM processing.
SERV_DIS	Service Discard Counter. Counts the number of CPCS-PDUs discarded due to either LANE_LECID echo suppression, CLP discard or frame relay DE discard.
RX_COUNTER/ VPC_INDEX	When ABR_VPC is a logic low, RX_COUNTER is the firewall mode credit counter. When ABR_VPC is a logic high, VPC_INDEX is used to control a VPC group. See ABR_VPC for description of this field.

5.7.3 Reassembly Buffer Descriptor Structure

Reassembly Buffer Descriptors usually reside in host memory (and optionally in SRC shared memory) on word-aligned addresses. The host controls the allocation and management of reassembly buffer descriptors.

During initialization, the host writes a pointer to an associated reassembly data buffer in each buffer descriptor entry in the BUFF_PTR field.

Table 5-12 and Table 5-13 describe the format of the reassembly buffer descriptors.

Table 5-12. Reassembly Buffer Descriptor Structure

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	NEXT_PTR																															
1	BUFF_PTR																															

Table 5-13. Reassembly Buffer Descriptor Structure Definitions

Field Name	Description/Function
NEXT_PTR	Pointer to the address of the next buffer descriptor.
BUFF_PTR	Pointer to the address of the data cell buffer. SAR does not access this word, so the user may place it anywhere in the buffer descriptor.

5.7.4 Free Buffer Queues

The host initializes the Free Buffer Queues in SRC shared memory, and during reassembly processing submits data buffers for reassembly to the Free Buffer Queues.

The Free Buffer Queue Base Table is located in Bt8233 internal memory. Table 5-14 and Table 5-15 describe the format of the Free Buffer Queue Base Table entries.

Table 5-16 and Table 5-17 describe the format of the Free Buffer Queue entries.

Table 5-14. Free Buffer Queue Base Table Entry Format

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Reserved								UPDATE								EMPT	BFR_LOCAL	Rsvd		READ											
1	LENGTH																Rsvd	BD_LOCAL	Rsvd		FORWARD											
2	READ_UD_PNTR																														Rsvd	
3	Reserved																															

Table 5-15. Free Buffer Queue Base Table Entry Descriptions

Field Name	Description/Function
UPDATE	Read Index Pointer Update Interval counter.
EMPT	Free Buffer Queue Empty flag. Used to indicate that an appropriate status entry has been written to indicate the empty condition.
BFR_LOCAL	If logic high, cell buffers located in SRC shared memory, otherwise in host memory.
BD_LOCAL	If logic high, buffer descriptor and READ_UD word are located in SRC shared memory, otherwise they are in host memory.
READ	Current READ index pointer.
LENGTH	Length in bytes of buffers in queue. Even though LENGTH is in bytes, the user must set the length to a mod-32 bit boundary.
FORWARD	Current Forward Processing Read index pointer.
READ_UD_PNTR	Word address in user memory to write READ pointer every UPDATE interval.

Table 5-16. Free Buffer Queue Entry Format

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	BUFFER_PNTR																															
1	BD_PNTR																														Rsvd	VLD
2[(1)]	Reserved														Rsvd	FWD_VLD	VCC_INDEX															
3[(1)]	(Not Used)																															
Notes: (1). These words are used only in Bank 0, if RSM_FBQCTL (FBQ0_RTN) = 1.																																

Table 5-17. Free Buffer Queue Entry Descriptions

Field Name	Description/Function
BUFFER_PNTR	Pointer to beginning of cell buffer.
BD_PNTR	Pointer to corresponding cell buffer descriptor.
VLD	Free Buffer Valid Bit. If high, location has a valid free buffer.
Rsvd	Always set to 0.
FWD_VLD	Forward Valid. If logic high, word contains valid buffer return information.
VCC_INDEX	Channel of corresponding buffer return.

5.7.5 Reassembly Status Queues

The Bt8233 reports reassembly status to the host on any one of 32 Reassembly Status Queues.

At initialization the host assigns the location and size of up to 32 Rsm Status Queues by initializing the Reassembly Status Queue Base Table entries in Bt8233 internal memory. The location and size of each Rsm Status Queue is independently programmable via these base table entries.

Table 5-18 and Table 5-19 describe the format of the Reassembly Status Queue Base Table entries.

Table 5-20 through Table 5-21 describe the formats of the Reassembly Status Queue entries.

Table 5-18. Reassembly Status Queue Base Table Entry Format

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	BASE_PNTR																														Rsvd	LOCAL
1	SIZE		Rsvd		WRITE												Rsvd				READ_UD											

Table 5-19. Reassembly Status Queue Base Table Entry Descriptions

Field Name	Description/Function
BASE_PNTR	Base Pointer. Pointer to the base word address of the status queue
LOCAL	If set high, queue is located in SRC shared memory, otherwise it is located in host memory.
SIZE	Size of the status queue. 00–64 01–256 10–1024 11–4096
WRITE	Current Write index pointer maintained by the SAR.
READ_UD	Periodic Read update index pointer maintained by the user.

Table 5-20. Reassembly Status Queue Entry Format with FWD_PM = 0

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	BD_PNTR																														“00”	
1	UU											CPI					CPCS_LENGTH															
2	Rsvd		PDU_CHECKS														VCC_INDEX															
3	VLD	Reserved										Reserved						STATUS							FWD_PM	OAM			STM			

Table 5-21. Reassembly Status Queue Entry Format with FWD_PM = 1

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	BD_PNTR																														“00”	
1	TRCC0																TRCC0+1															
2	Rsvd		PDU_CHECKS														VCC_INDEX															
3	VLD		Reserved						BIPV											OVFL				CNT_ROVR	FWD_PM	OAM			STM			

Table 5-22. PDU_CHECKS Fields Bits

29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD	BA_ ERROR	RSVD	LEN_ ERROR	PAD_ ERROR	CPI_ ERROR	RSVD	CRC_ ERROR	BOM_ BUF	FFPD_ PND	RSVD	LP	CI_ LAST	CI

Table 5-23. STATUS Field Bits

16	15	14	13	12	11	10	9	8
FFPD	EPD	FW	UNDF	OVFL	SFPD	TO	ABORT	CNT_ROVR

Table 5-24. STM Field Bits

3	2	1	0
EOM	BOM	STM_MODE	BFR1

Table 5-25. Reassembly Status Queue Entry Descriptions *(1 of 2)*

Field Name	Description/Function
BD_PNTR	Buffer Descriptor pointer. In message mode, pointer to the first buffer descriptor in the linked list. In streaming mode, pointer to an individual buffer descriptor.
UU	AAL5 CPCS-UU field.
CPI	AAL5 CPCS-CPI field. In AAL0 PTI terminated mode, the least significant bit contains the most significant bit of the total PDU length. The CPCS_LENGTH field contains the 16 least significant bits.
CPCS_LENGTH	AAL5 CPCS-LENGTH field. In AAL0 PTI terminated mode, it is the 16 least significant bits of the total PDU length. The CPI field contains the most significant bit.

Table 5-25. Reassembly Status Queue Entry Descriptions *(2 of 2)*

Field Name	Description/Function
PDU_CHECKS	Set various error flags related to PDUs. The PDU_CHECKS field contains the following control bits: Rsd | BA_ERROR | Rsd | LEN_ERROR | PAD_ERROR | CPI_ERROR | Rsd | CRC_ERROR | Rsd | Rsd | Rsd | LP | CI_LAST | CI BA_ERROR— Indicates that the total PDU length exceeds MAX_LENGTH field when AUU = 1. LEN_ERROR—Total CPCS-PDU length exceeds maximum length and not at end of message. PAD_ERROR—Pad field length is not correct. CPI_ERROR—CPI field is not all 0. CRC_ERROR—AAL5 CPCS or OAM CRC error. LP—Value of CLP header bit ORed across the CPCS-PDU. CI_LAST—Value of the PTI[1] header bit in the last cell of the CPCS-PDU. CI—Value of the PTI[1] header bit ORed across the CPCS-PDU.
VCC_INDEX	VCC index of channel.
VLD	Valid. Indication that status entry is valid. The host must set to zero after processing status.
STATUS	Sets various status bits. The STATUS field contains the following control bits: FFPD | EPD | FW | UNDF | OVFL | SFPD | TO | ABORT | CNT_ROVR FFPD—DMA FIFO Full Packet Discard. EPD—Early Packet Discard occurred. A partially reassembled CPCS-PDU has been discarded due to an error condition. FW—Firewall error condition occurred. May cause EPD. UNDF—Free Buffer Queue Underflow occurred. May cause EPD. OVFL—Last available Status Queue entry. SFPD—Status Full packet discard occurred. Will cause EPD. TO—Reassembly time-out condition occurred on this channel. EPD will not be set. ABORT—Abort function detected. EPD will not be set. CNT_ROVR—Indication that the SERVICE_CNT counter has rolled over on the channel.
OAM	If this field is nonzero, it indicates that an OAM or management cell has been received as follows: 001–F4 OAM 010–F4 OAM End to End 100–F5 OAM 101–F5 OAM End to End 110–PTI = 6 111–PTI = 7
STM	Sets various bits related to streaming mode. The STM field contains the following control bits: EOM | BOM | STM_MODE | BFR1 EOM—In streaming mode or BOM interrupt mode, this bit identifies that the buffer contains an EOM. BOM—In streaming mode or BOM interrupt mode, this bit identifies that the buffer contains a BOM. In message mode with BOM interrupt enabled, indicates that status entry points to only one buffer which contains a BOM. STM_MODE—Indication that streaming mode is enabled on the channel. BFR1—In streaming mode, indicates which free buffer queue the buffer came from. Logic high indicates COM(BFR1), logic low indicates BOM(BFR0).
FWD_PM	PM-OAM Forward Monitoring cell detected.
BIPV	BIP 16 violations. Same as BLER0+1 field in Backward Reporting PM-OAM cell.
TRCC0	PM total received cell count for CLP = 0.
TRCC0+1	PM total received cell count for CLP = 0+1.

5.7.6 LECID Table

The LECID Table illustrated in Figure 5-15, includes up to 32 unique identifiers for LECIDs. The DPRI field in the Reassembly VCC Table entry is used as the index into this table. Table 5-26 defines LECID Table fields.

Figure 5-15. LECID Table, Illustrated

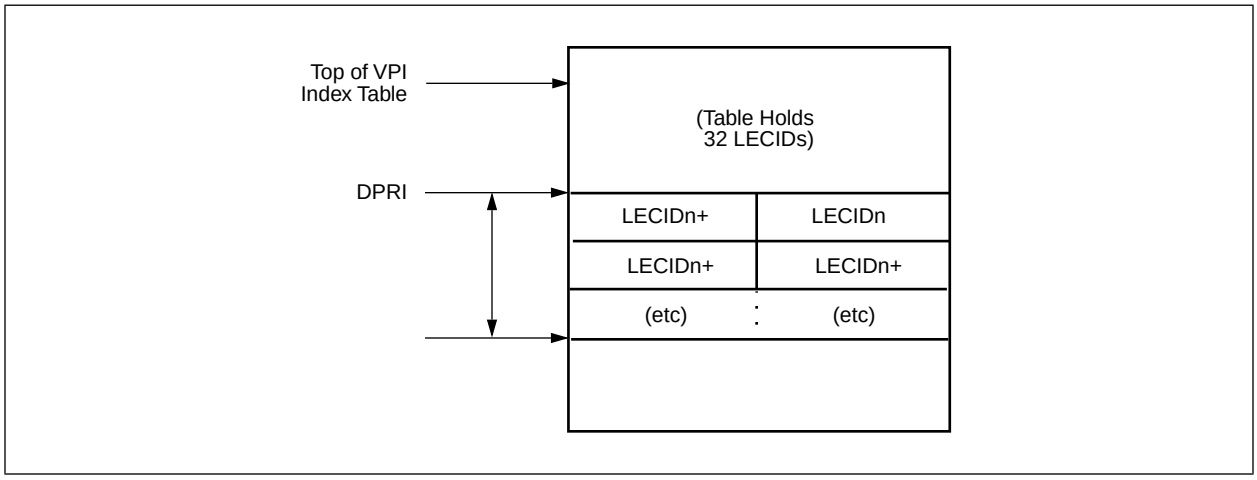


Table 5-26. LECID Table Entries

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	LECID1																LECID0															
...	...																...															
1-15	LECID31																LECID30															

Table 5-27. LECID Table Field Definition

Field Name	Function/Description
LECIDn	LAN Emulation Client Identifier, a unique identifier for a LAN Emulation Client (LECID). The LECID Table is capable of storing 32 LECIDs. The system designer can initialize this table to contain the LECIDs of LAN Emulation Clients that are transmitting multicast frames (point-to-multipoint Emulated LAN traffic) on an ATM network. Thus, with the LECID_EN bit set to a logic high on a channel, the Rsm Coprocessor looks for a match in the LECID Table with the LECID in each received LANE frame, and if a match, the frame is discarded. This implements echo suppression of superfluous multi-broadcast LANE traffic on the ATM network.

5.7.7 Global Time-Out Table

This table exists in internal SRAM starting at address 0x1340. The values in this table should be initialized during system initialization, before reassembly processing is started. These values set the selectable hardware PDU time-out values as described in Section 5.4.9, Hardware PDU Time-Out. See Table 5-28 for the Global Time-Out table entry format, and Table 5-29 for Global Time Out Table entry descriptions.

Table 5-28. Global Time-Out Table Entry Format

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	TERM_TOCNT1																TERM_TOCNT0															
1	TERM_TOCNT3																TERM_TOCNT2															
2	TERM_TOCNT5																TERM_TOCNT4															
3	TERM_TOCNT7																TERM_TOCNT6															
4	Reserved																TO_VCC_INDEX															

Table 5-29. Global Time-out Table Entry Descriptions

Field Name	Description/Function
TERM_TOCNTx	Time-out expiration count.
TO_VCC_INDEX	Time-out VCC_INDEX tracking variable.

5.7.8 Reassembly Internal SRAM Memory Map

The Reassembly Internal SRAM is in the address range 0x1000–0x13FF, as indicated in Table 5-30.

Table 5-30. Reassembly Internal SRAM Memory Map

Address	Name	Description
Internal Reassembly Status Queue Base Table Registers:		
0x1000–0x1007	RSM_SQ_QU0	Status Queue 0 Base Table.
0x1008–0x100F	RSM_SQ_QU1	Status Queue 1 Base Table
⋮	⋮	⋮
0x10F8–0x10FF	RSM_SQ_QU31	Status Queue 31 Base Table.
Internal Reassembly Free Buffer Queue Base Table Registers:		
0x1100–0x110F	RSM_FBQ_QU0	Free Buffer Queue 0 Base Table
0x1110–0x111F	RSM_FBQ_QU1	Free Buffer Queue 1Base Table
⋮	⋮	⋮
0x12F0–0x12FF	RSM_FBQ_QU31	Free Buffer Queue 31Base Table
Other Internal Reassembly Registers:		
0x1300–0x133F	Reserved	
0x1340–0x1353	GBL_TO	Global Time-out Table
0x1354–0x13FF	Reserved	

6.0 Traffic Management

6.1 Overview

The traffic management capabilities of ATM differentiate it from other communication technologies. The Bt8233 xBR Traffic Manager implements the complete set of ATM Service Categories as defined in the ATM Forum's Traffic Management (TM)4.0 Specification. These categories include CBR, Real-Time (RT) and Non-Real-Time (NRT) VBR, UBR, and ABR.

Table 6-1 provides a list of ATM attributes (traffic parameters, QoS parameters, and feedback characteristics) and identifies whether the Bt8233 supports these for each service category.

Table 6-1. ATM Service Category Parameters and Attributes *(1 of 2)*

Attribute	ATM Layer Service Category				
	CBR	**rt-VBR(1)**	**nrt-VBR(2)**	**UBR(3)**	**ABR**
Traffic Parameters:					
Peak Cell Rate (PCR)	Specified/Supported				
Cell Delay Variation Tolerance (CDVT), at PCR	Supported				
Sustainable Cell Rate (SCR)		Supported			
Maximum Burst Size (MBS)		Supported			
Cell Delay Variation Tolerance (CDVT), at SCR		Supported			
Minimum Cell Rate (MCR)					Supported
QoS Parameters:					
Peak-to-Peak Cell Delay Variation (CDV)	Supported		Not Supported		
Max Cell Transfer Delay (CTD)	Supported(4)		Not Supported(4)		

Table 6-1. ATM Service Category Parameters and Attributes *(2 of 2)*

Attribute	ATM Layer Service Category				
	CBR	**rt-VBR(1)**	**nrt-VBR(2)**	**UBR(3)**	**ABR**
Cell Loss Ratio (CLR)	Supported(4)			Not Supported(4)	Supported(4)
Other Attributes:					
Feedback(5)	Not Supported				Supported

Notes: (1). The real-time VBR service category is intended for real-time applications; i.e., those requiring tightly constrained delay and varied delay, as would be appropriate for voice and video applications.
(2). The non-real-time VBR service category is intended for non-real-time applications which have bursty traffic characteristics.
(3). The UBR service category is intended for non-real-time applications which do not require tightly constrained delay and varied delay. Examples of such applications are traditional computer communications applications such as file transfer and e-mail.
(4). This is a network parameter.
(5). Feedback refers to the several types of control cells called RM cells which are conveyed back to the source in order to control the source transmission rate in response to changing ATM layer transfer characteristics.

In order to supply these services, the xBR Traffic Manager directs the segmentation on each active VCC by controlling the segmentation coprocessor. By intelligently selecting when each VCC transmits a cell, the Traffic Manager guarantees that the output of the Bt8233 conforms to the negotiated traffic contract. This selection process (called Scheduling) executes dynamically, based upon per-VCC parameters.

The xBR Traffic Manager consists of two primary components. The first is the dynamic Cell Scheduler which provides for CBR, VBR, and UBR traffic. Second, for ABR service classes, the Bt8233 includes an additional state machine -- the ABR Flow Control Manager, which works in conjunction with the xBR Cell Scheduler.

Due to the complexities of ABR, a dedicated state machine is required to achieve full rate performance, one which reacts to feedback from the network and adjusts cell transmission accordingly. This specification models ABR to align with the TM4.0 ABR specification. The Bt8233 supports the rate-based flow control and service models specified for ABR in TM4.0.

The Bt8233 also provides Generic Flow Control (GFC). This XON/XOFF protocol complements the GFC algorithm by allowing switches to significantly overallocate port bandwidth.

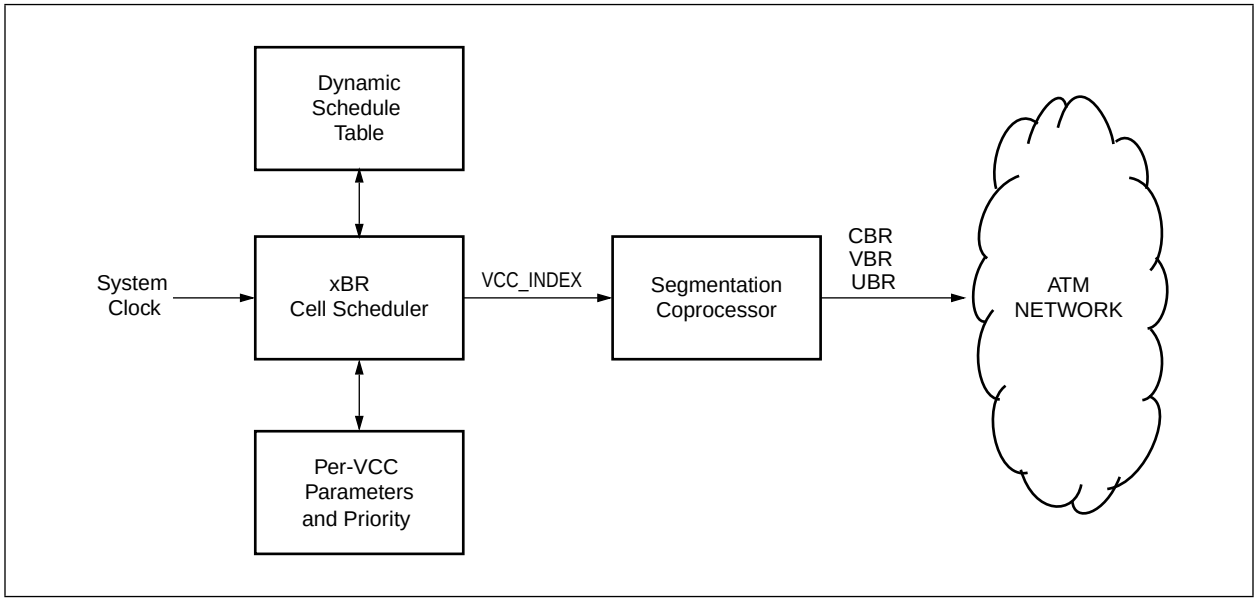
6.1.1 xBR Cell Scheduler

The Cell Scheduler maintains the QoS guarantees for each service category. It rate-shapes all segmentation traffic according to per-channel parameters. This provides each channel with appropriate transmission opportunities and guarantees conformance with network traffic contract policing algorithms applied to the outputs.

The Cell Scheduler selects individual channels based upon the Dynamic Schedule Table. A portion of this table may be reserved for CBR services. The VPs and VCs with CBR service reserve cell slots for transmission and are intended for highly regular data sources, such as voice circuits. The Bt8233 schedules all other traffic dynamically. The proprietary dynamic scheduling algorithm uses the remaining bandwidth to statistically multiplex all other service classes onto the line. The Bt8233 can manage VCC traffic on either a VC or a VP level. In addition, it can schedule traffic as a CBR tunnel, or pipe (i.e., several VCCs assigned to a single CBR scheduling priority, with individual VCCs within that tunnel scheduled on a round-robin basis). To enhance flexibility, the Bt8233 supports eight priorities of non-CBR traffic.

Figure 6-1 shows a high-level block diagram of the Cell Scheduler control flow for CBR, VBR, and UBR traffic. The Cell Scheduler tracks cell slots using the system clock and decides which VCC should send during each slot. This decision is based upon per-VCC parameters and the current condition of the Dynamic Schedule Table. Unlike ABR, these service categories are open-loop, and need no feedback from the network for run-time cell scheduling.

Figure 6-1. Non-ABR Cell Scheduling



6.1.2 ABR Flow Control Manager

The Bt8233 implements the ATM Forum ABR flow control algorithms, referred to in the aggregate as ABR by this document. The ABR service category effectively allows zero cell loss transmission through an ATM network, by regulating transmission based upon network feedback. The ABR algorithms regulate the rate of each VCC independently.

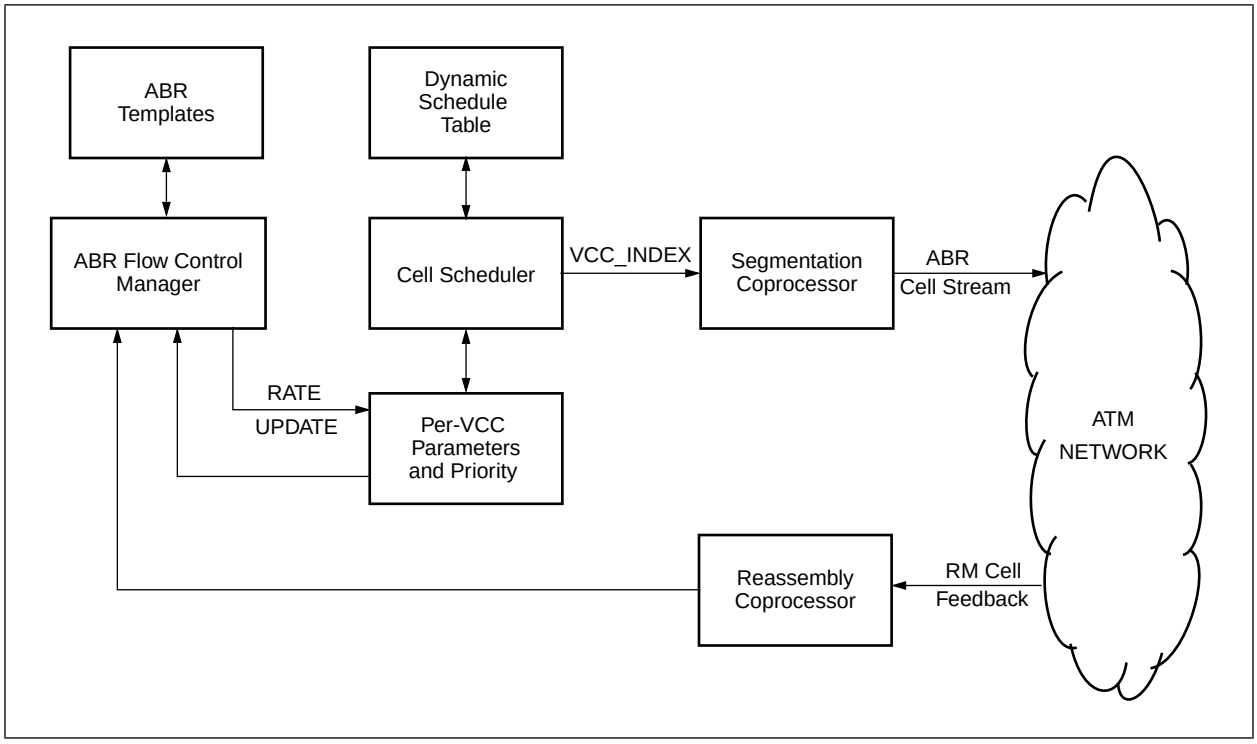
Figure 6-2 shows a high-level block diagram of the Bt8233's ABR feedback control loop.

Network feedback is received by the reassembly coprocessor and routed to the ABR Flow Control Manager. This state machine processes the feedback information according to per-VCC parameters and user-programmable ABR Templates, both located in SRC shared memory. These templates define ABR service parameters and policies for groups of VCCs.

Once the feedback is processed, the flow control manager provides the Cell Scheduler with an updated rate. Under the policy imposed by the template, the rate complies with the TM4.0 Source Behavior specifications.

Until the next update, the Cell Scheduler uses this rate to dynamically schedule the connection.

Figure 6-2. ABR Flow Control



For optimal performance, the ABR flow control manager implements the ABR algorithm in a hardware state machine. However, to provide flexibility against minor changes in the immature TM4.0 specification, the state machine is programmable through Rockwell-supplied ABR templates. These templates, resident in SRC shared memory, also provide a policy-tuning mechanism for interoperability and performance. Separate groups of VCCs can be assigned to application-optimized templates according to their path through the network.

NOTE: Application Example—Application Specific Templates—Each Template may have different flow control parameters. For instance, Initial Cell Rate (ICR) is inversely related to the Fixed Round Trip Transit (FRTT) time between two endpoints. WAN (Wide Area Network) connection FRTTs tend to be far larger than LAN FRTTs. Therefore, the FRTT limit imposed on WAN ICRs dwarfs that placed on LAN ICRs. LAN performance, especially response time, depends on a high ICR. By isolating WAN VCCs to a lower ICR template, the system designer meets the requirements of the WAN, but retains maximum LAN performance.

6.2 xBR Cell Scheduler Functional Description

6.2.1 Scheduling Priority

6.2.1.1 Eight Priority Levels + CBR

The Bt8233 supports eight scheduling priorities in addition to the optional CBR service category, with seven being the highest priority down to zero being the lowest priority. CBR channels are assigned a priority which is in effect higher than the eight scheduling priorities discussed here. From one to four of these priorities can be assigned to VBR and ABR service classes. The others are shared UBR priorities. The VBR/ABR priorities must be contiguous within the eight scheduling priorities, and thus accessible by an offset pointer. The host sets the offset of the VBR/ER priorities in SEG_CTRL(VBR_OFFSET). Figure 6-4 and Figure 6-5 illustrates how the value of VBR_OFFSET is determined.

6.2.1.2 VCC Priority Assignment

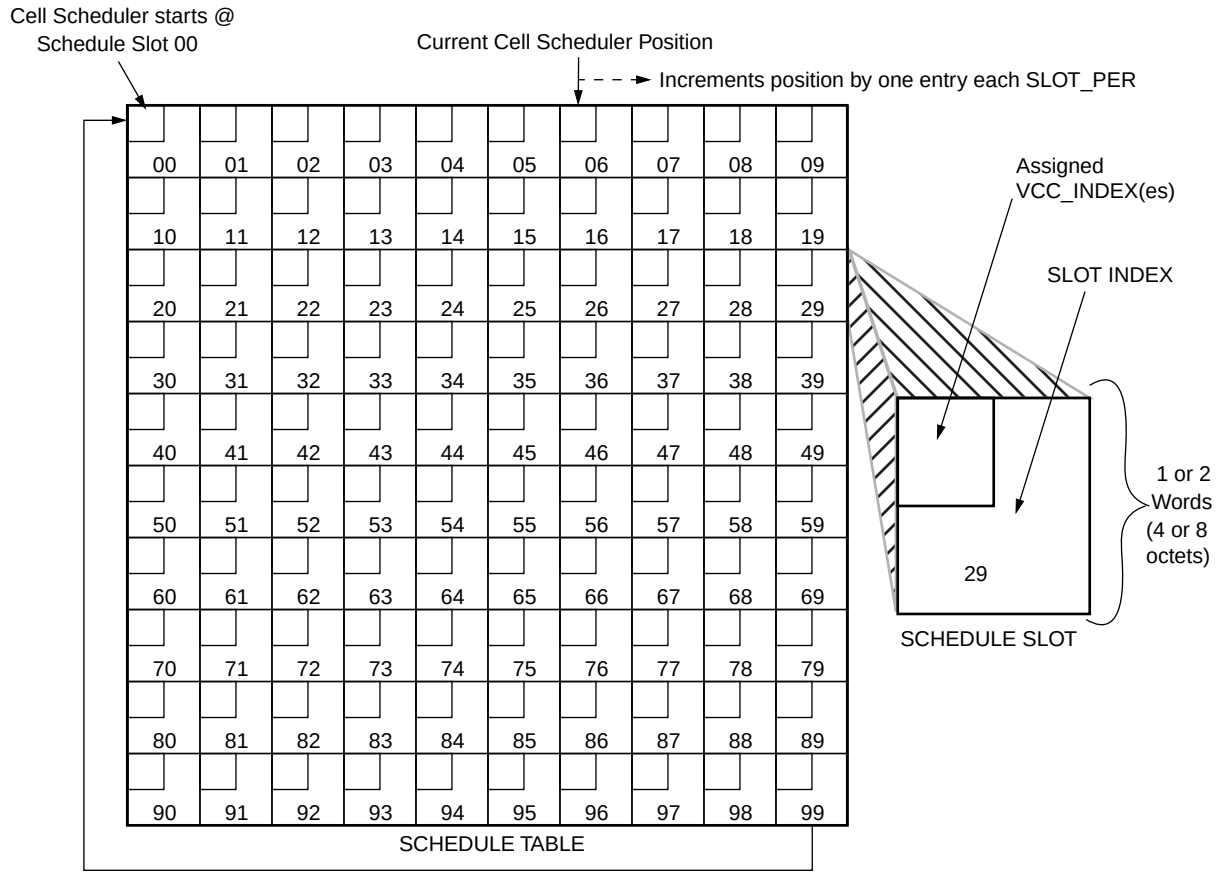
The host assigns the priority of all VCCs, except CBR VCCs, by setting the PRI field in the Segmentation VCC Table Entry. This priority should be set at connection setup and should not be changed dynamically.

6.2.2 Dynamic Schedule Table

The xBR Cell Scheduler schedules traffic according to the Dynamic Schedule Table. The table contains a programmable number of slots, determined by SCH_SIZE(TBL_SIZE). The duration of a single slot is a programmable number of system clocks, set as the number of system clocks per schedule slot, in SCH_SIZE(SLOT_PER). The Cell Scheduler sequences through this table in a circular fashion to schedule CBR, VBR, and ABR traffic. UBR traffic is handled as described in Section 6.2.5, UBR Traffic. By configuring the number of slots and the duration of each slot, the system designer chooses a range of available rates. This range of available rates is dictated by the rate at which a single slot is scheduled, the size of the table, and how many slots in the Dynamic Schedule Table each channel is assigned.

NOTE: Application Example: Schedule Table Setup—Figure 6-3 represents a sample schedule table. In this example the system designer has chosen 100 schedule slots. The duration of each slot is a programmable number of system clocks. The system designer may choose to schedule cells close to the full payload data rate of STS-3c at 353.2K cells/sec. At this cell rate, each cell slot will take 95 clocks. The scheduler begins at slot 00 and increments its position in the table every 95 clocks. After 9500 clocks, the scheduler position returns to 00.

Figure 6-3. Schedule Table with Size = 100



Note: Scheduler wraps back to 00 at end of table

As indicated in Table 6-2, four possible schedule slot formats exist. The user selects a format based on system requirements. If the system needs to generate CBR traffic, then the first 16 bits of each schedule table slot are reserved for a CBR Slot Entry. The number of distinct VBR and ABR priorities required, and the enabling of CBR traffic, govern the size requirements for each slot. The user controls the size and format of all schedule slots through the CBR_TUN and DBL_SLOT bits in the SEG_CTRL register. These factors, coupled with the size of the Schedule Table, determine the memory requirements for the Schedule Table.

Table 6-2. Selection of Schedule Table Slot Size by System Requirements

CBR Service	# VBR/ABR Priorities Required	Schedule Slot Size	CBR_TUN	DBL_SLOT	Available VBR/ABR Priority Levels
No	4	2 Words (64 bits)	0	1	VBR_OFFSET + 0, 1, 2 or 3
Yes	3	2 Words (64 bits)	1	1	VBR_OFFSET + 1, 2 or 3
No	2	1 Word (32 bits)	0	0	VBR_OFFSET + 0 or 1
Yes	1	1 Word (32 bits)	1	0	VBR_OFFSET + 1

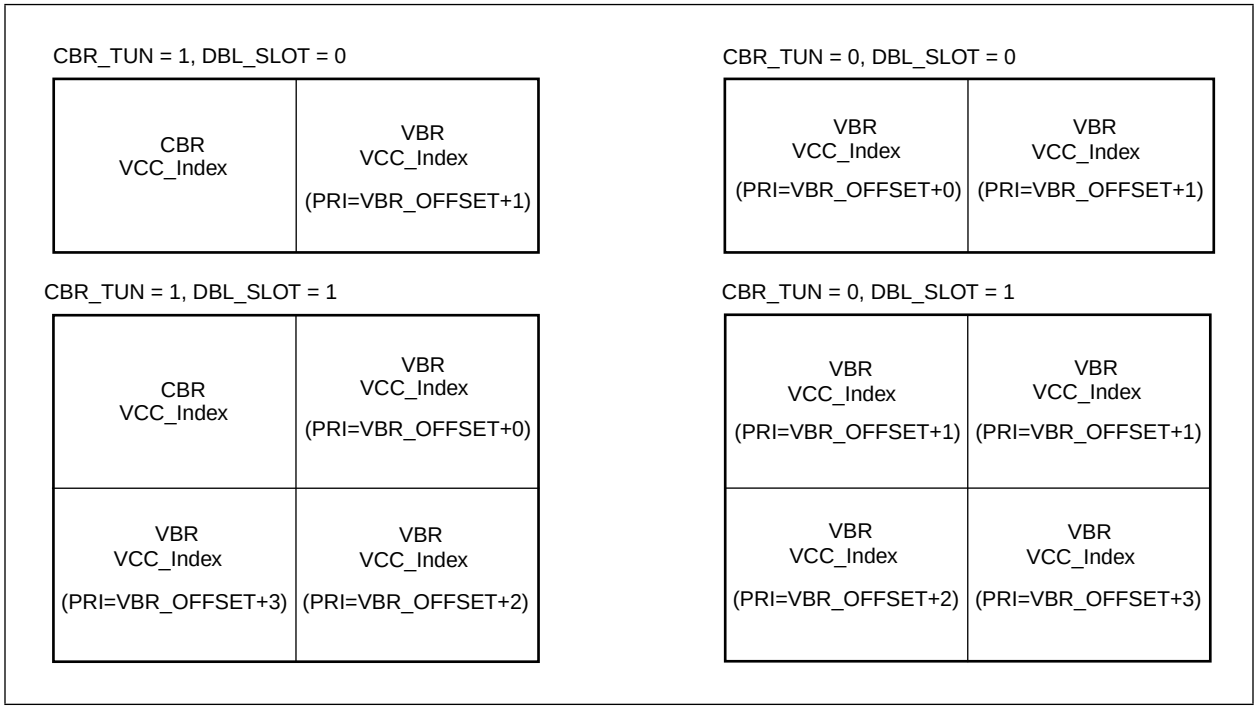
The VCC Index contents of each Schedule Table slot, based on these settings, is illustrated in Figure 6-4. The SAR may assign VBR VCC index(es) to any or all of the VBR VCC_Index fields in a schedule table slot. Each of the VBR fields in a schedule table slot that are assigned a VBR VCC_Index have a different scheduling priority (PRI), one from the other. Also, the SAR may assign the VCC index of the first of a linked list of VBR VCCs to any of the VBR VCC_Index fields.

The value of VBR_OFFSET is the difference between the highest VBR/ABR priority being actively scheduled and the largest VBR VCC_Index field in the scheduling slot (either 1 or 3). For example, if the system designer assigns VBR/ABR VCCs to three different scheduling priorities with the highest of those priorities being ‘5’ (i.e., PRI=5), then:

```
VBR_OFFSET = 5 - 3 = 2
```

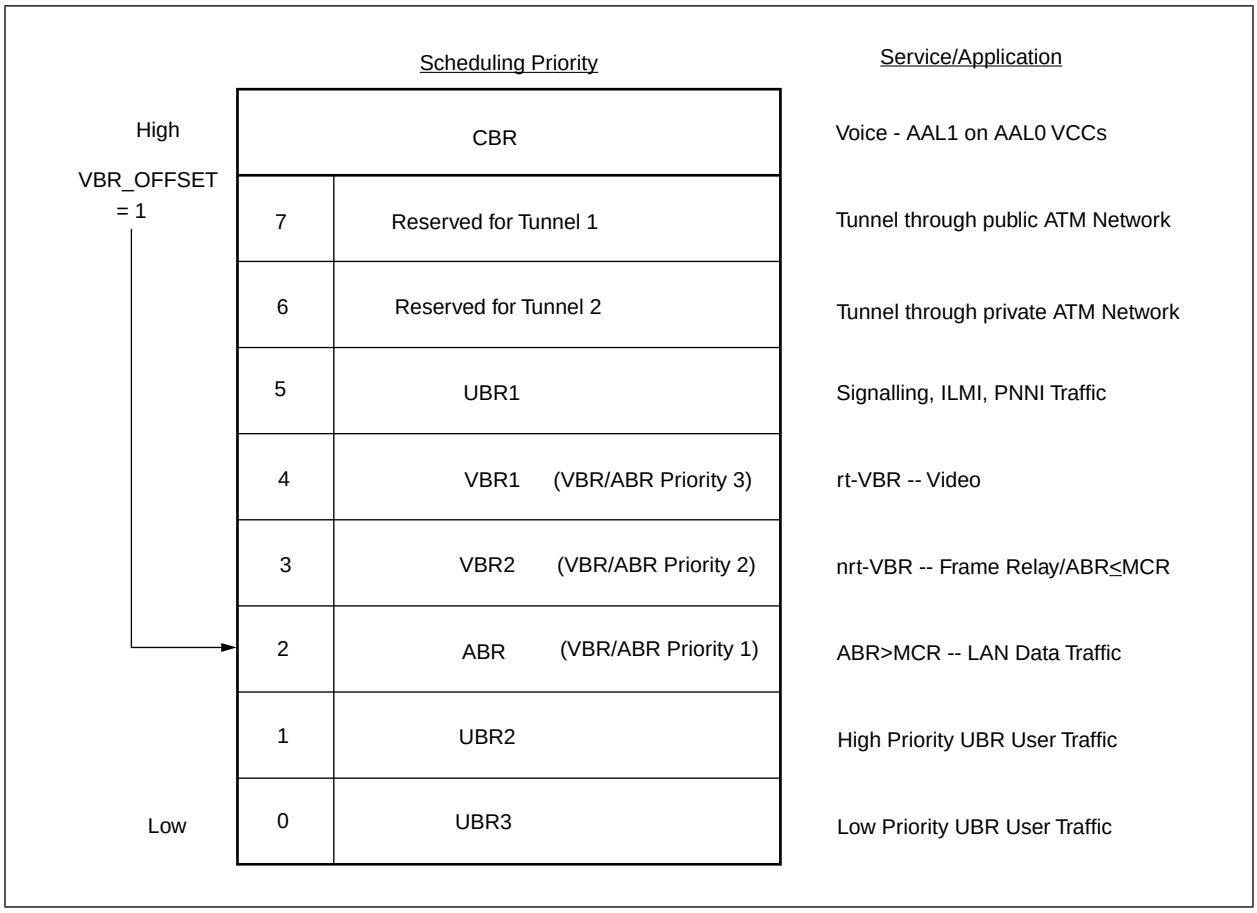
How these priorities are assigned to the VBR fields is illustrated below.

Figure 6-4. Schedule Slot Formats



NOTE: Application Example: Scheduler Priority Assignment—# of VBR/ABR-ER Priorities = 3 (CBR_TIN = 1, DBL_SLOT = 1). Highest VBR/ABR Scheduling priority is 4 (PRI = 4). VBR_OFFSET = 1. (See Figure 6-5.)

Figure 6-5. Typical Scheduling Priority Scheme with the Bt8233



As mentioned before, the VBR/ABR priorities must be contiguous in order to be accessible by VBR_OFFSET. To ensure that these priorities remain contiguous when accessed by VBR_OFFSET, the following condition must be met:

$$\text{VBR\_OFFSET} + (\#\ \text{of VBR/ABR priorities}) \leq 7$$

6.2.3 CBR Traffic

The CBR service category guarantees end-to-end bandwidth through the network. Certain data sources, such as voice circuits, require this guarantee, as well as constrained CDV. CDV is a measure of the "burstiness" of traffic. The ATM network supplies a minimum CDV for CBR channels by reserving cell transmission opportunities for the connections.

The Bt8233 generates CBR traffic by assigning specific slots in the Schedule Table to a CBR VCC. This connection will always send a single cell during its assigned slots. The System Clock serves as a time reference for CBR cell generation. The system designer programs the duration of schedule slots in system clocks in the SLOT_PER (slot period) field of the SCH_SIZE register.

6.2.3.1 CBR Rate Selection

Maximum Rate

For each CBR assigned cell slot, the Bt8233 generates one cell on the specified VCC (see Figure 6-6). The maximum or base rate of CBR channels is determined by the duration of a cell slot according to equation below, where *Rmax* is the maximum rate in cells per second.

$$Rmax = \frac{frequency(SYSCLK)}{SLOT - PER}$$

SLOT_PER should be nominally set to the number of clock cycles needed to transmit a full cell, and its minimum bound should be no less than 70.

NOTE: Application Example: Determining Maximum Rate—To achieve the maximum rate, the user assigns one VCC to every cell slot in the Schedule Table. This prevents any other VCC from being scheduled, since this channel uses all of the available slots.

frequency(SYSCLK) = 33 MHz
SLOT_PER = 93 Clocks/Slot
Rmax = 354.8K Cells/Second

6.2.3.2 Available Rates

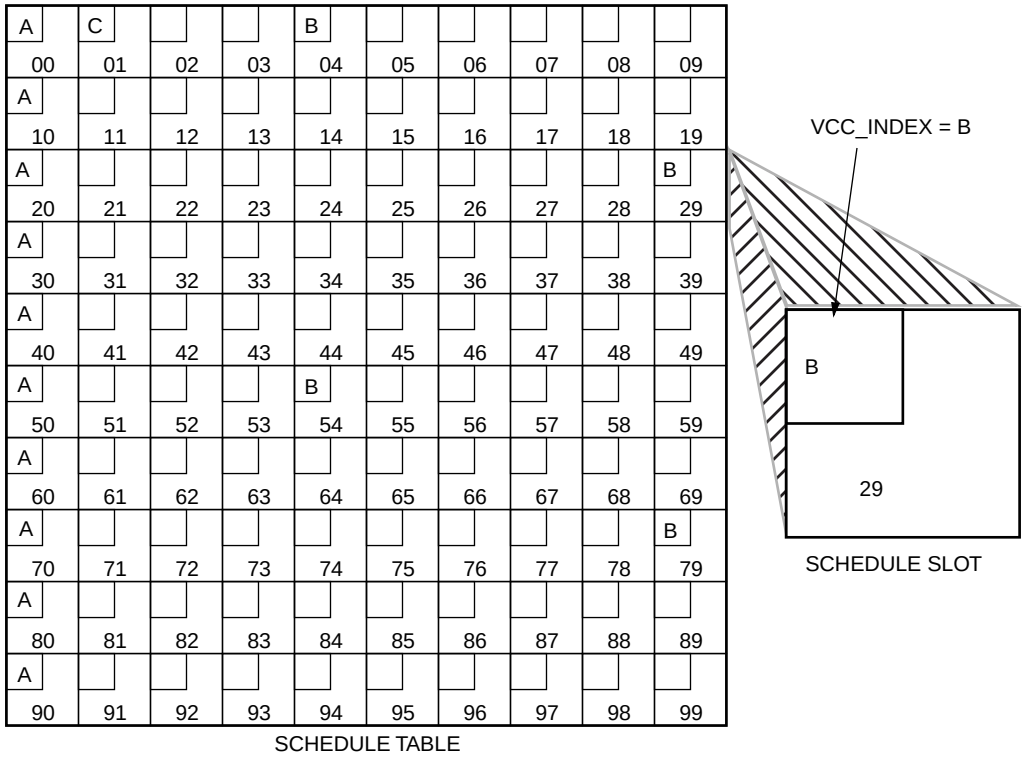
Once a maximum rate (R) has been selected, the size of the Schedule Table determines the rate granularity and minimum rate. The system designer specifies the number of table slots in SCH_SIZE(TBL_SIZE) The available CBR rates are:

```
R, R*(TBL_SIZE -1)/TBL_SIZE, R*(TBL_SIZE - 2)/TBL_SIZE, ...
        ..., R/TBL_SIZE
```

Minimum Rate The minimum rate available is therefore R/TBL_SIZE.

NOTE: Application Example: Assigning CBR Schedule Table Slots—The next figure shows an example Schedule Table with slots assigned to various CBR channels, each with a different rate. In this example, TBL_SIZE = 100 and SLOT_PER = 93. VCC_INDEX A occupies every tenth schedule slot. Therefore, they will transmit at R/10 or 35.4K cells/second. VCC_INDEX B occupies a slot every 25 cell slots and will transmit at a rate of R/25 or 14.2K cells/sec. VCC_INDEX C occupies only one schedule slot. Therefore, it will transmit at the minimum rate, R/TBL_SIZE or 3.54K cells/second. Note that not all cell slots have been assigned to CBR channels. During these slots, The Bt8233 will dynamically schedule traffic from the other service classes. However, the total bandwidth of channels A, B, and C will be reserved.

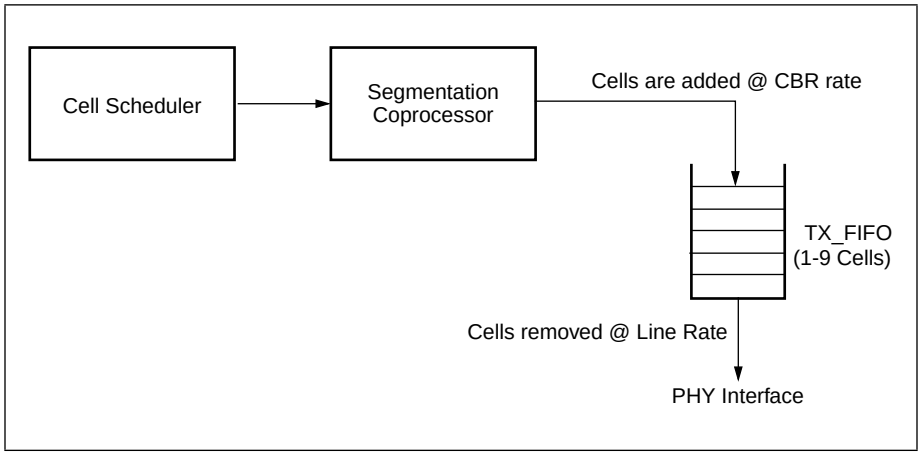
Figure 6-6. Assigning CBR Cell Slots



6.2.3.3 CBR CDV

By definition, CBR connections are sensitive to CDV[1]. The Bt8233's Traffic Manager minimizes CDV by basing all Traffic Management on the SYSCLK frequency. However, no system is without some CDV. In the case of terminals using the Bt8233, the dominant factor in CDV is the variation introduced between the segmentation coprocessor and the PHY layer device at the Transmit FIFO (TX_FIFO). Line overhead created by the framer in the PHY layer device causes this variation. Figure 6-7 illustrates this interface.

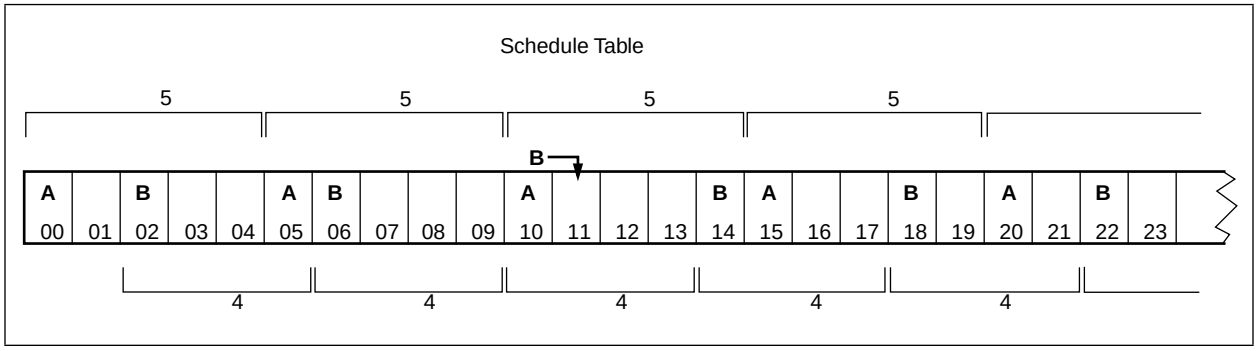
Figure 6-7. Introduction of CDV at the ATM/PHY Layer Interface



A possible source of Schedule-Table-dependent CDV is created when the host contracts for different CBR rates on more than one CBR channel, causing schedule slot conflicts in the Schedule Table. Figure 6-8 below illustrates an example of this. The figure shows a linear representation of a Schedule Table with 100 schedule slots. CBR channel A has reserved bandwidth for a rate of 70.8K cells/second, or every 5th cell slot. CBR channel B has reserved bandwidth for a rate of 88.7K cells/second, or every 4th cell slot. In this case there will be a schedule-slot reservation conflict between channels A and B every 20th cell slot, and one of the channel's slots will have to be reserved one slot later.

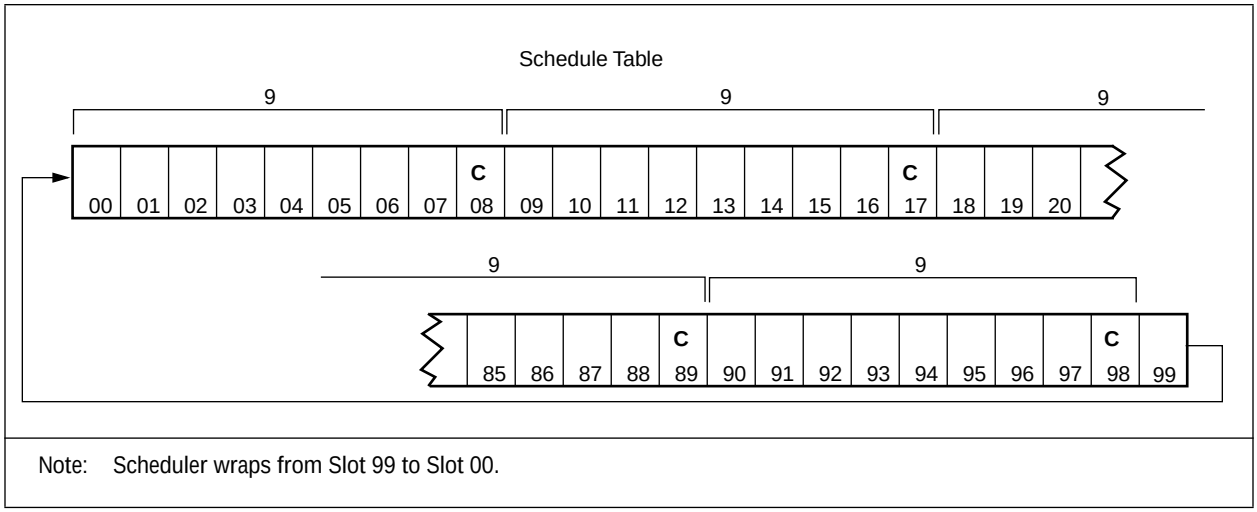
1. See ATM Forum UNI 3.1 or TM4.0 for a complete definition of CDV.

Figure 6-8. Schedule Slot Reservation Conflicts at Different CBR Rates



Another possible source of Schedule-Table-dependent CDV comes about for certain CBR rates whose schedule slot spacings do not evenly divide the table slot size. An example of this is illustrated in Figure 6-9 below. In this example, the beginning and end of a 100 slot Schedule Table is shown. The system designer has reserved bandwidth for CBR channel C at a rate of 32.25K cells/second, or every 9th cell slot. When the Scheduler wraps from the end of the table to the beginning of the table, the number of schedule slots between the last C channel slot at the end of the table and the first C channel slot at the beginning of the table is 10 slots, not 9.

Figure 6-9. CDV Caused by Schedule Table Size at Certain CBR Rates



Note: Scheduler wraps from Slot 99 to Slot 00.

The worst case CDV is determined by the following formula:

$$CDV_{max} = 1/(\text{CBR rate in cells/sec}) + TX\_FIFO\_LEN/(\text{line rate in cells/sec})$$

The first term in the formula above is introduced by CBR Rate Matching (see the CBR Rate Matching section below).

The second term in the formula above is introduced by the TX_FIFO itself. The FIFO introduces worst case CDV when one CBR cell is transmitted through an empty FIFO, and the next CBR cell from that channel is segmented to a full FIFO.

The system designer programs the TX_FIFO_LEN in the SEG_CTRL register. By reducing the TX_FIFO_LEN, the system designer reduces CDV. However, the TX_FIFO also absorbs PCI bus latency. To prevent PCI latency from impacting line utilization, set the TX_FIFO length according to the following formula:

$$TX\_FIFO\_LEN > 1 + (\text{worst case PCI latency})/(\text{line rate in cells/sec})$$

6.2.3.4 CBR Channel Management

The host initializes the Segmentation VCC Table Entry of a CBR VCC. The SCH_MODE of the VCC should be set to 001 (CBR). The Bt8233 thereafter ignores further processing based on the SCH_MODE field in the VCC Table Entry for CBR VCCs.

CBR PVC Provisioning

Once a Schedule Table has been created, the system assigns specific cell slots to CBR Provisioned Virtual Connections (PVCs). This process takes place before the segmentation process is initiated. Once segmentation has begun, the Bt8233 will dynamically allocate these cell slots to other channels until the data is supplied for the CBR VCC.

CBR SVC Setup and Teardown

It is also possible to set up and tear down CBR Switched Virtual Connections (SVCs) without disrupting ongoing segmentation. The user simply configures a new VCC Table Entry. Once the VCC is initialized, the host assigns schedule slots to the VCC according to its rate. The host then submits data as with any segmentation VCC.

CBR Rate Matching

In reality, the CBR schedule rate of a channel will not exactly match the rate of its data source. To compensate for this asynchronous data source behavior, the Bt8233 provides a rate matching mechanism for use when CBR traffic is mapped to a Virtual FIFO. (This method is needed only for Virtual FIFOs. For VCCs that are not Virtual FIFOs, the cell transmission is skipped automatically if there is no data available).

For an individual CBR channel mapped to a Virtual FIFO, the host directs the Bt8233 to skip one cell transmission opportunity whenever data is unavailable. The host requests this rate-matching adjustment by setting the SCH_OPT bit of the CBR channel. The next time the Bt8233 encounters a CBR slot for this VCC, it will not transmit data on that VCC. Then the Bt8233 indicates to the host that a slot has been skipped by clearing the SCH_OPT bit.

With this method, the host effectively synchronizes the Bt8233's scheduled rate to the external data source rate. The host must configure the CBR VCC rate slightly higher than the actual rate of the data source. Skipping cell transmission slots then compensates for the rate differential.

CBR Tunnels (Pipes) CBR VCCs are managed as a single VCC Table Entry. The Bt8233 schedules CBR tunnels in a similar manner. A CBR tunnel occupies schedule table slots in the same manner as a CBR VCC. However, instead of one VCC, several VCCs are scheduled within this CBR slot. The VCCs within a CBR tunnel may be either UBR or VBR (VBR1 or VBR2) traffic. In the case of a UBR tunnel the VCCs are serviced in round robin order. For a VBR tunnel, each VCC is shaped to its GCRA parameters within the CBR tunnel.

The host identifies a tunnel as a scheduling priority. Therefore, up to eight tunnels may be active at once. Of these 8 tunnels, up to 3 may be VBR tunnels. Individual VCCs are assigned to the tunnel by the host, by setting the PRI field in the VCC Table to the priority of the tunnel.

The host must assign the highest priority level(s) to CBR tunnel(s). If the system designer is establishing one CBR tunnel, its priority level must be 7. If two CBR tunnels are being established, their priority levels must be 6 and 7, and so on for more CBR tunnels.

NOTE: Figure 6-5, shown previously, shows priorities 6 and 7 used as CBR Tunnels for UBR Traffic. The two tunnels are assigned the two highest priorities. The Host assigns a fixed number of Schedule Table slots to the tunnel to reserve a fixed rate. Each time an assigned slot is encountered by the Cell Scheduler, it selects a VCC from a round-robin queue of active VCCs assigned to that priority. For example, 100 UBR VCCs with PRI = 7 might currently be segmenting data. Each will get 1/100th of the CBR bandwidth assigned to the tunnel. These channels will take priority over all other traffic classes. Tunneling enables system-level end-users to purchase CBR services from a WAN service provider. The purchaser may then dynamically manage the traffic within this leased CBR tunnel as either CBR or UBR services.

In this example, the user has configured the Bt8233 to manage two independent tunnels. One tunnel, priority 6, is through a private ATM network, perhaps a corporate ATM campus backbone. The other, priority 7, carries traffic through a public network. This topology allows the end-user to lease reserved CBR bandwidth from an administrative domain, but manage the usage of the tunnel in an arbitrary fashion.

6.2.4 VBR Traffic

The Bt8233 Cell Scheduler also supports multiple priority levels for VBR traffic. The VBR service class takes advantage of the asynchronous nature of ATM by reserving bandwidth for VBR channels at average cell transmission rates without hardcoding time slots, as with CBR traffic. This dynamic scheduling allows VBR traffic to be statistically multiplexed onto the ATM line, resulting in better utilization of the shared bandwidth resources.

6.2.4.1 Rate Shaping vs. Policing

The Cell Scheduler rate-shapes the segmentation traffic for up to 64K connections. The outgoing cell stream for each VCC is scheduled according to the GCRA algorithm. This guarantees compliance to policing algorithms applied at the network ingress point. Channels may be rate-shaped as VCs or VPs, according to one of three leaky bucket paradigms, set by the SCH_MODE bit in the channel's Segmentation VCC Table entry.

6.2.4.2 Single Leaky Bucket

The first and simplest bucket scheme is single leaky bucket. The user defines a single set of GCRA parameters -- *I* (Interval) and *L* (Limit). *I* is used to control the per-VCC Peak Cell Rate (PCR), and *L* is used to control the CDVT of the outgoing cell stream. The user enables this scheme by setting the SCH_MODE bits to '100' (VBR1).

6.2.4.3 Dual Leaky Bucket

The user may also select, on a per-VCC basis, to apply two leaky buckets to a single connection. The user enables this scheme by setting the SCH_MODE bits to '101' (VBR2).

When using VBR2 SCH_MODE, you are limited to 256 values for the *I*2 and *L*2 parameters. These parameters are stored as Bucket table entries. See Table 6-11 for the definition of a Bucket table entry. There is complete flexibility with regard to using these 256 values to specify SCR or PCR. In VBR2, *I*1 and *L*1 can specify either PCR and CVDT, or Sustainable Cell Rate (SCR) and Burst Tolerance (BT), with *I*2 and *L*2 used to specify the parameters not assigned to *I*1 and *L*1.

For example, you can configure:

```
I1 = PCR, L1 = CDVT, I2 = SCR, L2 = BT;
```

– or –

```
I1 = SCR, L1 = BT, I2 = PCR, L2 = CDVT.
```

6.2.4.4 CLP Based Buckets

The third option allows both buckets to apply to CLP=0 cells. CLP=0 means high priority cells; CLP=1 means cells are subject to discard. CLP=1 cells are scheduled from the second bucket only. Therefore the second bucket should correspond to PCR. This controls the PCR of the total cell stream, but only controls the Sustainable Cell Rate of CLP=0 cells. The user enables this scheme by setting the SCH_MODE bits to '110' (VBRC).

6.2.4.5 Rate Selection

The *I* and *L* parameters of the GCRA algorithm represent cell slots in the schedule table. Therefore, the VBR channels have the same maximum rate available as the CBR channels. However, because VBR channels are not constrained to repeat at Rmax/TBL_SIZE, they have a much finer rate granularity. The minimum rate for VBR channels is Rmax/(TBL_SIZE-1).

6.4.5.6 Real-Time VBR and CDV

Real-time VBR traffic should be assigned the highest scheduling priority to minimize cell delay variation. The worst-case CDV for rt-VBR traffic is calculated as:

```
((# VBR VCCs at same-or-higher priority)+TX_FIFO_LEN)/(line
        rate in cells/sec).
```

6.2.5 UBR Traffic

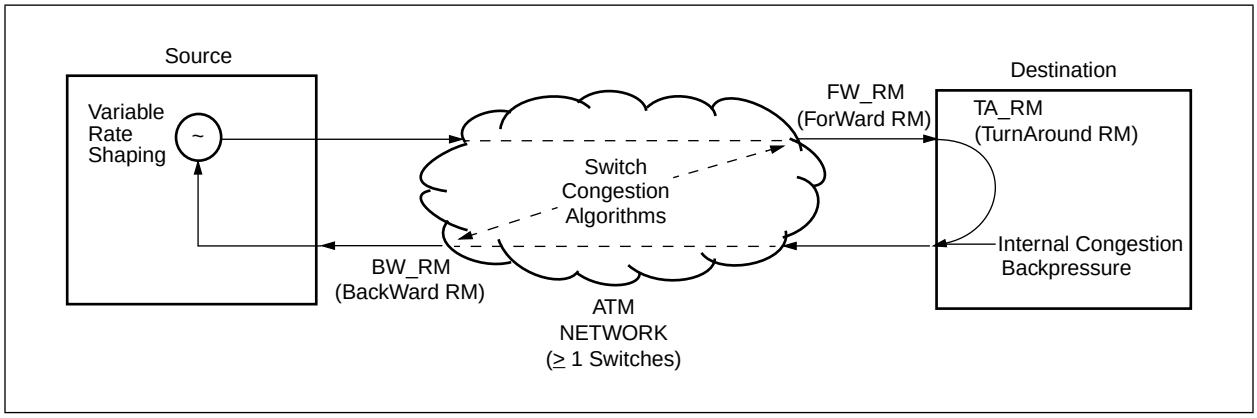
The remaining priorities of traffic are scheduled as UBR traffic. All UBR channels within a priority are scheduled on a round-robin basis. To limit the bandwidth that a UBR priority consumes, the user should use a CBR tunnel in that priority level.

6.3 ABR Flow Control Manager

6.3.1 A Brief Overview of TM4.0

This section briefly describes the TM4.0 ABR Flow Control algorithms. However, it is strongly recommended that the reader be familiar with the ATM Forum's TM4.0 specification before attempting to understand the Bt8233's ABR implementation. Figure 6-10 illustrates ABR service category feedback control.

Figure 6-10. ABR Service Category Feedback Control



6.3.2 Internal ABR Feedback Control Loop

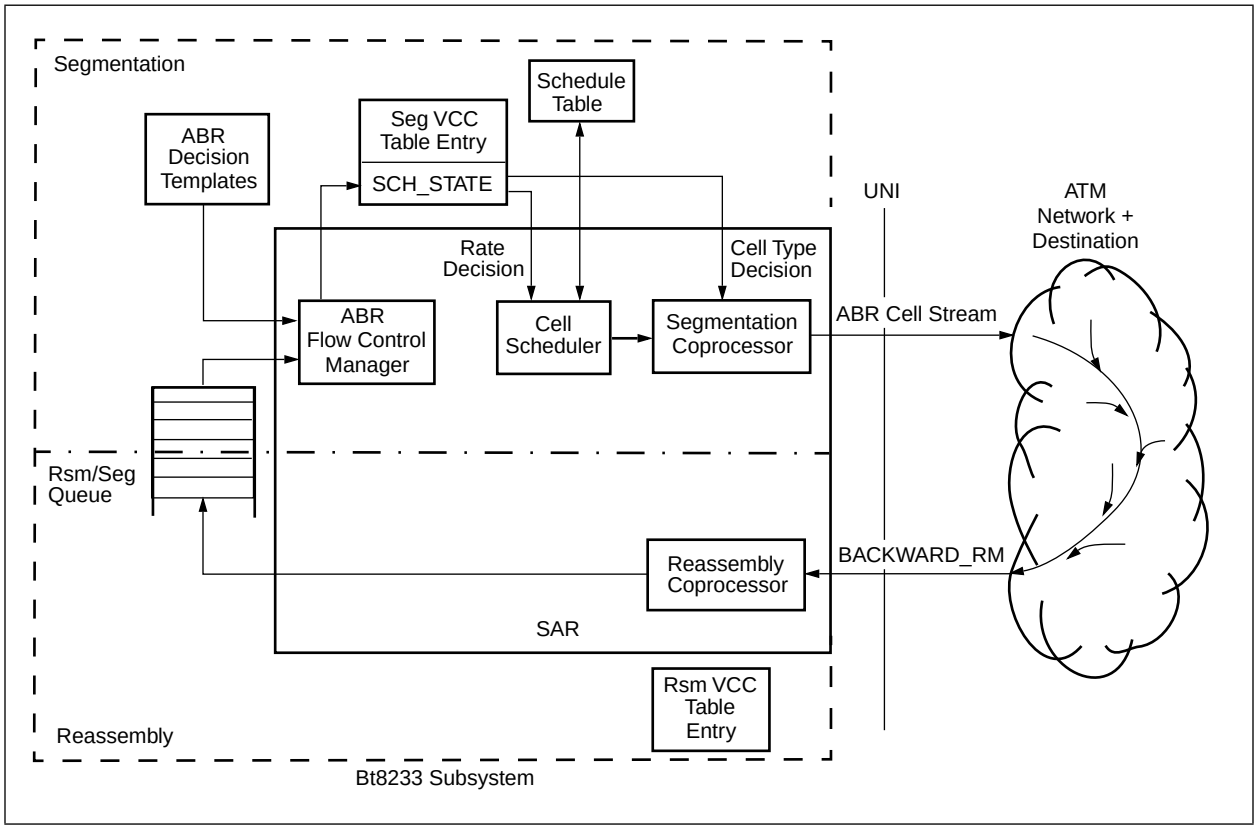
As a complete implementation of the UNI ATM layer, the Bt8233 acts as both an ABR Source and Destination, complying with all required TM4.0 ABR behaviors. The Bt8233 utilizes the dynamic rate adjustment capability of the XBR Cell Scheduler as the Source's variable rate shaper. An internal feedback mechanism supplies feedback from the received cell stream to the ABR Flow Control Manager, a special purpose state machine. This state machine translates the feedback, extracted from received Backward RM cells, to instructions for the XBR Cell Scheduler and segmentation coprocessor. It supports both binary and ER flow control methods.

6.3.2.1 Source Flow Control Feedback

Figure 6-11 illustrates the feedback loop which controls the Source cell stream. The Bt8233 injects an in-rate cell stream (i.e., CLP = 0) into the ATM network for each ABR VCC. Network elements modify the flow control fields in the cell stream's Forward RM cells. After a round trip through the network and destination node, these cells return to the Bt8233 receive port as Backward RM cells. The reassembly coprocessor processes incoming Backward RM cells, and communicates with the segmentation state machines via the RSM/SEG Queue in SRC shared memory. Upon receiving this feedback from the queue, the ABR flow control manager updates fields within the SCH_STATE portion of the Segmentation VCC Table Entry. The XBR Cell Scheduler and segmentation coprocessor use these fields to generate the ABR in-rate cell stream, closing the Source Behavior feedback loop.

The SAR also processes the Explicit Forward Congestion Indication (EFCI) bit in the data cell header(s) per the rules in TM4.0.

Figure 6-11. Bt8233 ABR-ER Feedback Loop (Source Behavior)

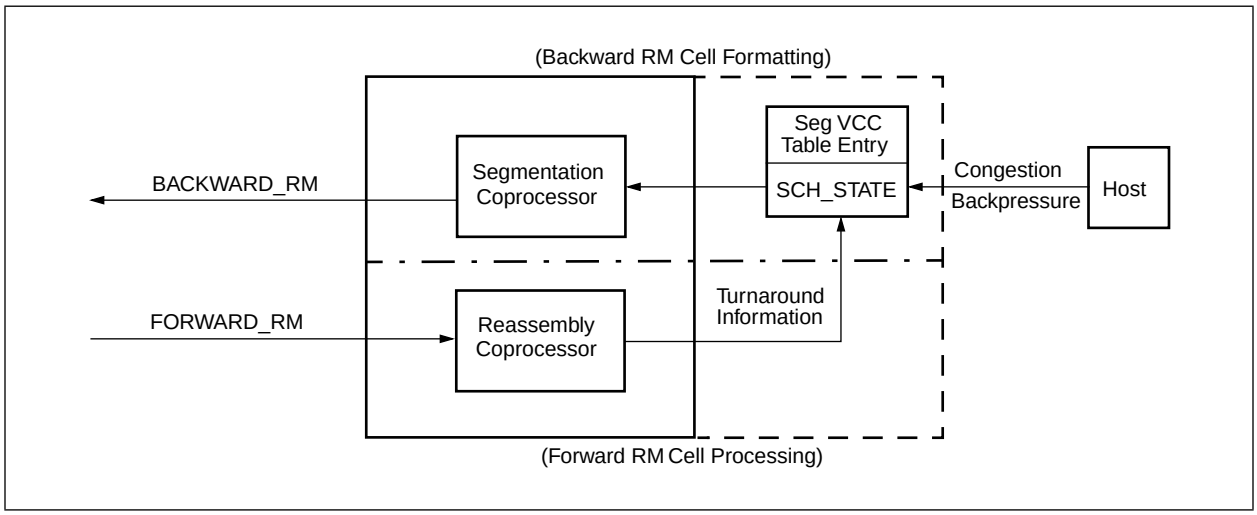


6.3.2.2 Destination Behavior

The Bt8233 also responds to an incoming ABR cell stream as an ABR Destination. As shown in Figure 6-12, the reassembly coprocessor processes received Forward RM cells. It “turns around” this incoming information to the segmentation coprocessor via the SCH_STATE part of the Segmentation VCC Table Entry. The Segmentation Processor then formats Backward RM cells containing this information and inserts these turnaround RM cells into the transmit cell stream.

The Bt8233 also provides a mechanism for the Destination host to apply backpressure to the Source. The host notifies the Bt8233 of internal system congestion. The Bt8233 passes this information to the Source via Backward RM cells, to flow control the transmitting Source. The SAR also processes the EFCI bit in the data cell header(s) per the rules in TMA.0.

Figure 6-12. Bt8233 ABR-ER Feedback Generation (Destination Behavior)



6.3.2.3 Out-of-Rate Cells In addition to in-rate cell streams, the Bt8233 can also generate out-of-rate (CLP = 1) cell streams. These CLP=1 streams compliment and enhance the information flow contained in the in-rate cell streams. An example of the use of this mechanism would be to send out of rate forward RM cells on a channel if the transmit rate on that channel has dropped below the schedule table minimum rate. This provides a mechanism to restart scheduling of an ABR VCC whose rate has dropped to zero or below the schedule table minimum rate.

6.3.3 Source and Destination Behaviors

ABR's Source and Destination Behavior Definitions are each listed in the ATM Forum's TM4.0 Specification. Refer to this specification for these definitions.

6.3.4 ABR VCC Parameters

The state of each VCC is stored individually in its SEG VCC Table Entry. The ABR parameters are stored in the SCH_STATE part of the Segmentation VCC Table Entry. Due to the large number of ABR parameters, the SCH_STATE of ABR VCCs require an additional location in the SEG VCC Table.

6.3.5 ABR Templates

The ABR Templates reside in SRC shared memory in a region referred to as the ABR Instruction Table. The ABR Instruction Table contains one or more templates. Individual VCCs are assigned to a single template. Each template supports a group of VCCs.

These ABR Templates are furnished by Rockwell, and are downloaded to the Bt8233 as a complete microcoded state machine.

6.4 GFC Flow Control Manager

6.4.1 A Brief Overview of GFC

Generic Flow Control (GFC) is a one-way control mechanism which allows the network equipment to control the input from an end station, for the class(es) of traffic defined as controlled. This mechanism does not allow the end station to exert any control on traffic from the network.

The usefulness of GFC is that it allows overbooking of the bandwidth on the input side of the network switch buffers. This allows a much higher degree of multiplexing than is otherwise possible, and significantly reduces the network-side costs of connections. By overbooking the input bandwidth, a high degree of sharing is possible, and the buffer system can be utilized by more end nodes than full bandwidth input would allow. GFC is used to coordinate access to that bandwidth when temporary conflicts occur.

GFC provides a link-level, short term, XON/XOFF-type flow control mechanism. This mechanism only works on the link from the end station to the first piece of network equipment. The GFC protocols are defined and described in ITU Recommendation I.361.

6.4.2 The Bt8233's Implementation of GFC

The Bt8233 implements the GFC one-queue mode. The reassembly coprocessor provides Auto Configure and Command Detection. The segmentation coprocessor provides Halt Processing and Per-transmit Queue SET_A control. It does not implement the optional Queue B.

Once the link has been configured for GFC operation (as described in the subsection below), a received HALT indication will cause the segmentation coprocessor to halt processing of all channels, both controlled and uncontrolled. This halt condition will continue until a cell is received without the HALT indication.

A received SET_A indication will increment the GFC credit counter by one. A GFC-controlled cell can only be sent when the GFC credit counter is equal to one. Transmission of a GFC-controlled cell decrements the credit counter by one. Each of the eight transmit priority queues can be configured for GFC control by setting the appropriate GFCn bit(s) in the Scheduling Priority (SCH_PRI) register. In this way the SAR can segment both GFC-controlled and GFC uncontrolled traffic simultaneously. GFC-controlled queues will be active only when the GFC credit counter is equal to one.

Note that CBR-traffic is not affected by the SET_A command since it is not mapped into a transmit priority queue. In addition, the segmentation coprocessor implements a credit borrow algorithm that provides better utilization of the line when receive and transmit cell streams are not synchronized. Up to one credit can be borrowed.

The user must control the transmitted GFC field via the HEADER_MOD and GFC_DATA fields in the buffer descriptor entries. For GFC-controlled channels, GFC_DATA = 0101; and for non-GFC-controlled channels, GFC_DATA = 0001.

6.4.2.1 Configuring the Link for GFC Operation

The following describes a sample sequence of how to auto-configure a link for GFC operation after the link has been initialized:

1. Host sets a software GFC initialization timer = 0.
2. Disable reassembly coprocessor by setting RSM_CTRL0(RSM_EN) = 0.
3. Set the framer chip to pass unassigned cells.
4. Enable the GFC link interrupt (GFC_LINK), by setting HOST_IMASK0 (EN_GFC_LINK) = 1.
5. Read HOST_ISTAT0 register twice, to clear it.
6. Enable the reassembly coprocessor, and set the GFC initialization timer to some user-assigned value.
7. Upon occurrence of an interrupt and before GFC initialization timer expiration, read HOST_ISTAT0. If GFC_LINK is a logic high, continue. If the timer expires before GFC_LINK is detected, do not enable the link for GFC processing.
8. Set SEG_CTRL(SEG_GFC) to a logic high.
9. Set the GFCn bit(s) in the SCH_PRI register to enable the appropriate priority queue(s) for GFC controlled operation.
10. Set the framer chip to generate unassigned cells with GFC field in cell headers set to the value of 0001.

6.5 Traffic Management Control and Status Structures

6.5.1 Schedule Table

At initialization, all words in the entire Schedule Table space should be written to 0xFFFFFFFF. Individual schedule slot entries can then be initialized as either CBR slots or Tunnel slots as needed. The two formats are indicated in Table 6-3.

Table 6-3. Schedule Slot Entry–CBR/Tunnel Traffic

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CBR	CBR_TUN_ID															Reserved for 1 VBR Pointer															
1(1)	Reserved for 2 VBR Pointers																															
Notes: (1). This word is present only when the DBL_SLOT field in SEG_CTRL is set, meaning 2 words per schedule slot.																																

6.5.1.1 CBR Traffic

A schedule slot is dedicated to a CBR connection by formatting the CBR bit to a logic high and the CBR_TUN_ID field in the slot entry as indicated in Table 6-4.

Table 6-4. CBR_TUN_ID Field, Bit Definitions–CBR Slot

Bit	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Def.	1	CBR_VCC_INDEX (14 bits)													

6.5.1.2 Tunnel Traffic

A schedule slot is dedicated to a tunnel by formatting the CBR bit to a logic high and the CBR_TUN_ID field of the slot entry as indicated in Table 6-5. Table 6-6 defines the Schedule Slot Field names.

Table 6-5. CBR_TUN_ID Field, Bit Definitions–Tunnel Slot

Bit	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Def.	0	Reserved											PRI		

Table 6-6. Schedule Slot Field Descriptions—CBR Traffic

Field Name	Description
CBR	Set high to indicate a CBR or tunnel slot; set low to indicate a NON CBR or tunnel slot.
PRI	Global priority used for tunnel.
CBR_VCC_INDEX	Segmentation VCC index for dedicated CBR schedule slot.

6.5.2 SCH_STATE

This section specifies the SCH_STATE part of the Segmentation VCC Table entry, which consists of words 7 through 9 for VBR, and words 7 through 19 for ER.

6.5.2.1 VBR1 or VBR2 Schedule State Table

Table 6-7. SCH_STATE for SCH_MODE = VBR1 or VBR2

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
7	BUCKET2 (MSB)				L1_EXP					L1_MAN									I1_EXP					I1_MAN							
8	BUCKET2 (LSB)				RESERVED																											
9	RESERVED																															

See Section 6.2.4, VBR Traffic, for key data on *I* and *L* values.

Table 6-8. VBR1 and VBR2 SCH_STATE Field Descriptions

Field Name	Description
BUCKET2	Index into Bucket table to give I2 and L2 for SCHED_MODE = VBR2 & VBRC. Bucket table base is given by SEG_BCKB register field. Entries in Bucket table have same format as L1_EXP, L1_MAN, I1_EXP and I1_MAN.
L1_EXP	GCRA L parameter exponent for bucket 1. L1_EXP must satisfy L1_EXP <= min(I1_EXP + 9,29) L1_EXP that does not satisfy L1_EXP >= I1_EXP - 9 will have an effective L1 value of 0.
L1_MAN	GCRA L parameter mantissa for bucket 1. Bucket 1 L value is $2^{(L1\_EXP - 10)}(1 + L1\_MAN \S 512)$
I1_EXP	GCRA I parameter exponent for bucket 1. I1_EXP must satisfy 10 <= I1_EXP <= 25
I1_MAN	GCRA I parameter mantissa for bucket 1. Bucket 1 I value is $2^{(I1\_EXP - 10)}(1 + I1\_MAN \S 512)$

6.5.2.2 ER Schedule State Table

Refer to Tables 6-9 through 6-11.

Table 6-9. SCH_STATE for SCH_MODE = ER

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
7	Rsvd				L_EXP					L_MAN									I_EXP					I_MAN									
8	Rsvd				PRESENT																											
9	Rsvd	MCR_LIM_NZ	DELTA_SIGN	DELTA_NZ	MCR_LIM_EXP					MCR_LIM_MAN									DELTA_EXP					DELTA_MAN									
10	CONG_ID					Rsvd	OOR_PRI			Rsvd	TM_EXP	BCK_OOR	FWD_OOR	SCH_OOR	TA_XMIT	TA_PND	CELL_INDEX																
11	FWD_INDEX																RM_TIME															
12	RATE_INDEX																EB_INDEX															
13	CRM																UNACK															
14	Reserved																NEXT_OOR															
15	Reserved																															
16[(1)]	TA_ID									TA_DIR	TA_BN	TA_CI	TA_NI	TA_RA	Rsvd		TA_ER															
17[(1)]	TA_CCR																TA_MCR															
18	FWD_ID									FWD_DIR	FWD_BN	FWD_CI	FWD_NI	FWD_RA	Rsvd		FWD_ER															
19	Reserved																FWD_MCR															

Notes: (1). Words 16 & 17 are written directly by the Rsm coprocessor (turnaround information).

Note: = Values are furnished by the ABR Templates.

***Table 6-10. ER SCH_STATE Field Descriptions** (1 of 2)*

Field Name	Description
L_EXP	GCRA L parameter exponent for ER rate.
L_MAN	GCRA L parameter mantissa for ER rate.
I_EXP	GCRA I parameter exponent for ER rate.
I_MAN	GCRA I parameter mantissa for ER rate.
CONG_ID	Congestion index for changing ER on turn around RM cells. The ER in the turn around cell is changed only if the FBQ_CNG[CONG_ID] bit is set in the SCH_CNG register. The normal setting for this field is the free buffer queue ID for the VCC.
OOR_PRI	Segmentation priority for out of rate RM cells.
TM_EXP	RM_TIME value has expired and is no longer valid.
BCK_OOR	Backward RM cell has been scheduled out of rate.
FWD_OOR	Forward RM cell has been scheduled out of rate.
SCH_OOR	VCC has halted due to low ACR and has out of rate forward RM cells enabled.
TA_XMIT	A backward RM cell has been transmitted after the last forward RM cell transmitted.
TA_PND	A backward RM cell is waiting for transmission.
CELL_INDEX	ER cell type decision block index for cell type decisions. The cell type decision block is located at byte address SCH_ABAB*128 + 8*CELL_INDEX
FWD_INDEX	ER cell type decision block index for cell type decisions after a forward RM cell is transmitted.
RM_TIME	Global slot count [21:6] at time of last forward RM transmission.
RATE_INDEX	ER rate decision block index. The rate decision block is located at byte address SCH_ABRD*128 + 32*RATE_INDEX.
EB_INDEX	ER exponent table index for rate index block. The exponent table is located at byte address SCH_ABRB*128 + EB_INDEX*128.
CRM	ER parameter
UNACK	Number of forward RM cells transmitted since last backward RM cell received. Initialize to zero.
NEXT_OOR	Next VCC index for linking out of rate RM cells.
TA_ID	ID field from most recent received forward RM cell. This field is written by the SAR.
TA_DIR	DIR field for turn around (backward) RM cell. This field is written by the SAR.
TA_BN	BN field for turn around (backward) RM cell. This field is written by the SAR.
TA_CI	CI field for turn around (backward) RM cell. This field is written by the SAR.
TA_NI	NI field for turn around (backward) RM cell. This field is written by the SAR.
TA_RA	RA field from most recent received forward RM cell. This field is written by the SAR.
TA_ER	ER field for turn around (backward) RM cell. This field is written by the SAR.
TA_CCR	CCR field from most recent received forward RM cell. This field is written by the SAR.

Table 6-10. ER SCH_STATE Field Descriptions *(2 of 2)*

Field Name	Description
TA_MCR	MCR field from most recent received forward RM cell. This field is written by the SAR.
FWD_ID	ID field for transmitted forward RM cells. This field is supplied by the user.
FWD_DIR	DIR field for transmitted forward RM cells. This field is supplied by the user.
FWD_BN	BN field for transmitted forward RM cells. This field is supplied by the user.
FWD_CI	CI field for transmitted forward RM cells. This field is supplied by the user.
FWD_NI	NI field for transmitted forward RM cells. This field is supplied by the user.
FWD_RA	RA field for transmitted forward RM cells. This field is supplied by the user.
FWD_ER	ER field for transmitted forward RM cells. This field is supplied by the user.
FWD_MCR	MCR field for transmitted forward RM cells. This field is supplied by the user.

6.5.3 Bucket Table

The Bucket Table has only 256 entries. Refer to Tables 6-11 and 6-12.

Table 6-11. Bucket Table Entry

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Rsvd				L2_EXP					L2_MAN									I2_EXP					I2_MAN							

Table 6-12. Bucket Table Entry Field Descriptions

Field Name	Description
L2_EXP	GCRA L parameter exponent for bucket 2. L2_EXP must satisfy L2_EXP <= min(I2_EXP + 9,29) L2_EXP that does not satisfy L2_EXP >= I2_EX - 9 will have an effective L2 value of zero.
L2_MAN	GCRA L parameter mantissa for bucket 2. Bucket 2 L value is $2^{(L2\_EXP - 10)}(1 + L2\_MAN \S 512)$
I2_EXP	GCRA I parameter exponent for bucket 2. I2_EXP must satisfy 10 <= I2_EXP <= 25
I2_MAN	GCRA I parameter mantissa for bucket 2. Bucket 2 I value is $2^{(I2\_EXP - 10)}(1 + I2\_MAN \S 512)$

7.0 OAM Functions

7.1 OAM Overview

OAM cells are ATM Layer management messages. These are generated by the host on the segmentation side of the Bt8233. They are detected and either monitored or processed on the reassembly side of the Bt8233.

ATM's OAM capabilities differentiate it from other less robustly managed communication technologies. The Bt8233 provides internal support for the detection and generation of OAM traffic, including PM OAM.

The Bt8233 supports the F4 and F5 OAM flows according to ITU-T Recommendation I.610. It also monitors the performance of up to 128 channels, generating PM-OAM cells according to the same specification.

7.1.1 OAM Functions Supported

Refer to ITU-T Recommendation I.610 for complete information on the structures and functions of OAM cell generation, detection and processing.

The Bt8233 internally supports the following functions as described in I.610:

- Full PM functions (designed for estimating transmission performance on any channel and for reporting performance estimations in the backward direction)
- Detecting of OAM cells, and routing them as directed by the host
- Generating OAM cells, as directed by the host

Implementation of the full range of functions in processing OAM cells will be done at the software level due to the low bandwidth of OAM traffic (less than 1% of the bandwidth of active connections).

7.1.2 OAM Flows Supported

7.1.2.1 F4 OAM Flow

The F4 OAM flow is the VP level, provided by OAM cells dedicated to ATM layer OAM functions for VPCs. The F4 flow is bidirectional.

OAM cells for the F4 flow have the same VPI value as the user cells of the VPC.

F4 flow OAM cells are identified **as** F4 flow cells by a pre-assigned VCI value of 3 (segment flow cell) or 4 (end-to-end flow cell). The same pre-assigned VCI value is used for both directions of the F4 flow. Refer to Table 7-2.

7.1.2.2 F5 OAM Flow

The F5 OAM flow is the Virtual Channel level, provided by OAM cells dedicated to ATM layer OAM functions for VCCs. The F5 flow is bidirectional.

OAM cells for the F5 flow have the same VPI value and VCI value as the user cells of the VCC.

F5 flow OAM cells are identified **as** F5 flow cells by a pre-assigned Payload Type Identifier (PTI) code value of 100 (segment flow cell) or 101 (end-to-end flow cell). The same pre-assigned PTI value is used for both directions of the F5 flow.

7.1.2.3 Performance Monitoring

Performance Monitoring is a set of functions which monitors and processes user information on a channel to produce maintenance information specific to that channel. This maintenance information is added to the in-rate data flow on that channel in the form of PM cells. The PM cells are added at the source of a connection or link and extracted at the sink of a connection or link. With this maintenance information, the user can estimate and analyze the transport integrity of that channel.

The PM flow is bidirectional, and PM cells are of two basic function types: forward monitoring cells (which carry the forward error detection information) and backward reporting cells (which carry the results of the performance monitoring checks).

The Bt8233 performs PM on up to 128 user-assigned channels. The SAR enables PM for a channel by setting the PM enable bits (PM_EN) in the Segmentation and Reassembly VCC State Table entries for that channel.

The Bt8233 performs PM processing on any channel by monitoring blocks of user cells on that channel. The size of this block of cells is set by the user and can have the value (N) of 128, 256, 512, or 1024 cells. The Bt8233 inserts a PM cell after every N user cells on that channel. A block size of 0 is valid when the SAR is a destination point only for PM processing. In this case, the segmentation coprocessor will only generate Backward reporting cells in response to Reassembly PM calculations.

PM as performed by the Bt8233 calculates the following:

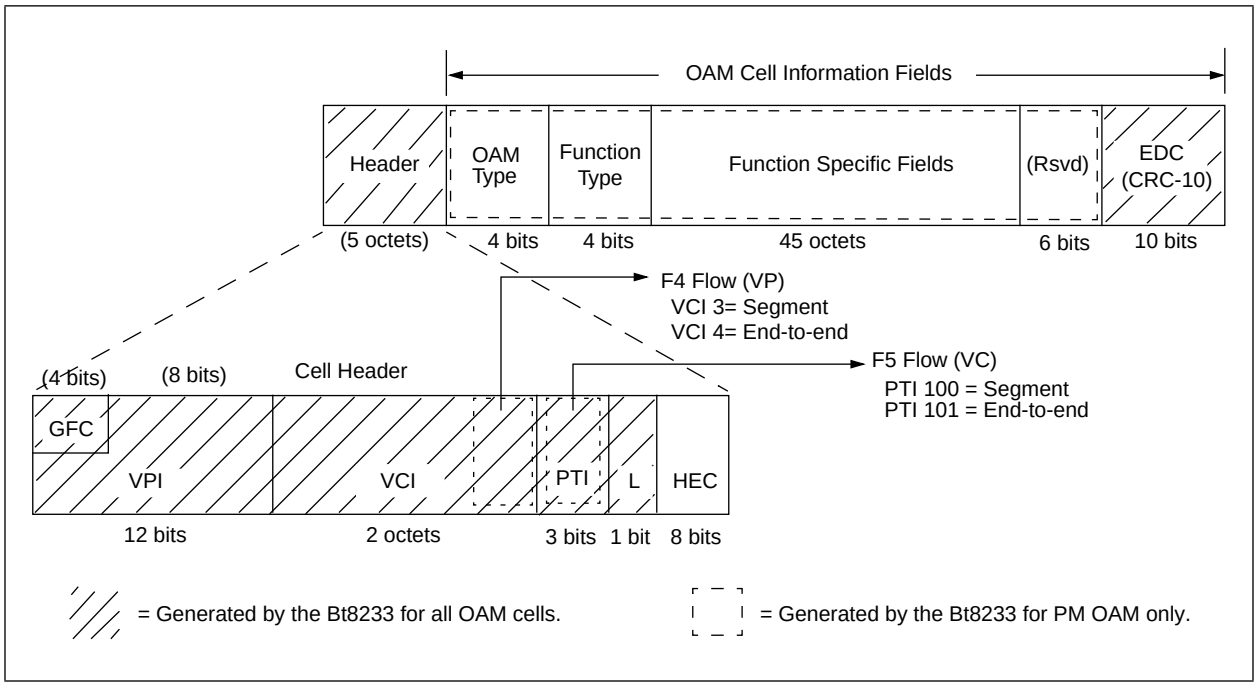
- Errored blocks (by means of a BIP-16 error detection code generated over the payloads of the user cells in the PM block)
- A count of misinserted PM forward monitoring cells

The user can activate PM on any channel either during connection establishment, or at any time after the connection has been established.

7.1.3 OAM Cell Format

Figure 7-1 illustrates the common OAM cell format, and identifies the fields specific to OAM.

Figure 7-1. OAM Cell Format



EDC = Error Detection Code (10 bits) -- this is a CRC-10 error detection code computed over the OAM cell information fields, excluding the EDC field.

The OAM Type and Function Type identifiers for Performance Management, as specified by I.610, are given in Table 7-1.

.

Table 7-1. OAM Type and Function Type Identifiers for Performance Management

OAM Type(1)	Coding	Function Type(2)	Coding
Performance Management	0010	Forward monitoring Backward reporting	0000 0001

Notes: (1). OAM Type indicates the type of management function performed by this cell; e.g., fault management, Performance Management, etc.
(2). Function Type indicates the actual function performed by this cell within the management type indicated by the OAM Type field.

7.1.4 Local vs. Host Processing of OAM

ATM carries with it significant network management overhead. The Bt8233 supports those robust OAM features described above. Due to the breadth of ATM applications and the continuing evolution of ATM management standards, it is essential that a range of flexibility be provided in the processing of network management overhead. To provide this flexibility, the Bt8233 includes an optional Local Processor Interface.

Since the Bt8233 is capable of SRC shared memory segmentation and reassembly, it can route OAM traffic including PM traffic to and from this Local Processor, thereby off-loading ATM network management from the host. Thus, host processing power is focused on the user applications specifically concerned with processing ATM user data traffic.

To accomplish this, the Bt8233 provides global OAM buffer and status queues (OAM_BFR_QU and OAM_STAT_QU, addresses for both assigned in the RSM_CTRL1 register). If the OAM_QU_EN bit in the RSM_CTRL1 register is set to a logic high, the Bt8233 routes OAM traffic to these global queues. With SRC shared memory addresses assigned to these global queues, the Bt8233 processes OAM traffic through the Local Processor, thereby freeing the host from these management functions. In addition, the global OAM segmentation status queue, SEG_CTRL(OAM_STAT_ID), is used if the OAM_STAT bit in the segmentation buffer descriptor is a logic high.

7.2 Segmentation of OAM Cells

The host (or local processor) places OAM cells in a single buffer, and thus allocates a single SBD for the OAM cell data buffer, not a linked list of SBDs.

The host (or local processor) then writes a pointer to that SBD in the next available Transmit Queue entry and sets the VLD bit to 1.

When the segmentation coprocessor processes that Transmit Queue entry, it submits the OAM cell data buffer to the xBR Traffic Manager and its thus scheduled for transmission.

7.2.1 Key OAM-Related Fields for OAM Segmentation

7.2.1.1 Segmentation Buffer Descriptors

There are several fields in the Segmentation Buffer Descriptor entry that are used to facilitate segmentation of OAM cells.

- Set the 2-bit AAL_OPT field to SINGLE (value = 01). This enables reading 48 octets from a single buffer to form a single ATM cell.
- Set the OAM_STAT bit to a logic high. The Bt8233 will now report status to the OAM-dedicated OAM_STAT_ID identified in the SEG_CTRL register, instead of the STAT specified in the Seg VCC Table entry.
- The user can set the single-bit HEADER_MOD field to a logic 1. This activates the WR_PTI and WR_VCI bits in the buffer descriptor, which signal the Bt8233 to overwrite the ATM header PTI and VCI fields for that cell with the values from the PTI_DATA and VCI_DATA fields. In this way, F4 and F5 flow OAM cells can be generated by the Bt8233.
- The VCI_DATA field set to a value of 3 (segment cell) or 4 (end-to-end cell) generates an F4 flow OAM cell.
- The PTI_DATA field set to a value of 100 (segment cell) or 101 (end-to-end cell) generates an F5 flow OAM cell.
- Set the AAL_MODE field to 01 (AAL0).
- Set both BOM and EOM bits to zero.
- Set the CRC10 bit to a logic high.

7.2.1.2 Low Latency Transmission

For low latency, the LINK_HEAD bit in the Transmit Queue entry should be set to a logic high. This tells the Bt8233 to link the buffer chain at the head of the existing chain for the corresponding VCC. This bit is intended for use with the Seg Buffer Descriptor's SINGLE option to send in-line OAM cells. Only a single Seg Buffer Descriptor may be linked to a Transmit Queue entry when this bit is set.

This bit must also be set if the OAM SBD is placed on the transmit queue after a partial PDU, to ensure correct segmentation.

7.2.1.3 Segmentation Status Queue

The SINGLE bit in the Seg Status Queue entry should be set to a logic high. This bit is set if the SINGLE option in the AAL_OPT field of the Seg Buffer Descriptor is set. This bit indicates a special buffer is in use, rather than the normal system-assigned buffers for normal CPCS-PDUs.

7.2.1.4 F4 Flow

For F4 flow operation, a separate VCC Table entry must be configured.

7.2.2 Error Condition During OAM Processing

Each OAM cell has a 10-bit Error Detection Code (EDC) field, for storing and transporting the calculated CRC-10 error detection code results (computed over the OAM cell information fields, excluding the EDC field). To enable this CRC-10 function, set the CRC10 bit in the Seg Buffer Descriptor entry to a logic high.

7.3 Reassembly of OAM Cells

To enable the reassembly coprocessor to detect and therefore further process OAM cells, set the OAM_EN bit in the Reassembly Control Register 1 (RSM_CTRL1) to a logic high.

The Bt8233 detects the following OAM cell flows:

- Segment F4 Flow
- End-to-end F4 Flow
- Segment F5 Flow
- End-to-end F5 Flow
- PTI = 6
- PTI = 7

The Bt8233 provides global OAM buffer and status queues (OAM_BFR_QU and OAM_STAT_QU, addresses for both assigned in the RSM_CTRL1 register). To activate these queues, set the OAM_QU_EN bit in the RSM_CTRL1 register to a logic high. The Bt8233 then routes OAM traffic to these global buffer and status queues.

NOTE: When OAM detection is enabled, the cell buffer size must be large enough to hold a complete cell.

7.3.1 Key OAM-Related Fields for OAM Reassembly

7.3.1.1 Reassembly VCC State Table

The Reassembly VCC State Table field, SEG_VCC_INDEX, should be written to point to the channel index of the corresponding segmentation channel. This is necessary for PM-OAM as well as ABR channels.

7.3.1.2 Reassembly Status Queue

The 3-bit OAM field in the Rsm Status Queue entry should be set to the value indicated in the Rsm Status Queue structure description for that field. A non-zero value in the OAM field indicates that the cell is an OAM cell.

If the CRC-10 error detection code computation on the OAM cell shows an error, the Bt8233 will set the CRC_ERROR bit in the Status Queue entry to a logic high.

7.3.1.3 F4 Flow

For F4 flow operation, a separate VCC Table entry must be configured.

7.3.2 OAM Reassembly Operation

Received OAM traffic should be detected and routed to the global OAM buffer and status queues (OAM_BFR_QU and OAM_STAT_QU).

The reassembly coprocessor treats OAM cells as one-cell PDUs. The Rsm coprocessor transfers the 48-octet OAM payload to the next available global data buffer and writes a Rsm Status Queue entry.

Once an OAM cell is detected, the Rsm coprocessor checks the cell to determine whether it is a PM cell. The coprocessor does this by seeing if the OAM_TYPE field in the cell is set to value 0010. PM detection is only performed on F4 and F5 OAM cells.

If the Rsm coprocessor finds the cell is a PM cell, it is processed following the guidelines described in Section 7.4, PM Processing below.

NOTE: OAM cells that get buffers from the global OAM Free Buffer Queue do not affect the per-VCC Firewall calculation.

7.3.3 Error Conditions During OAM Reassembly

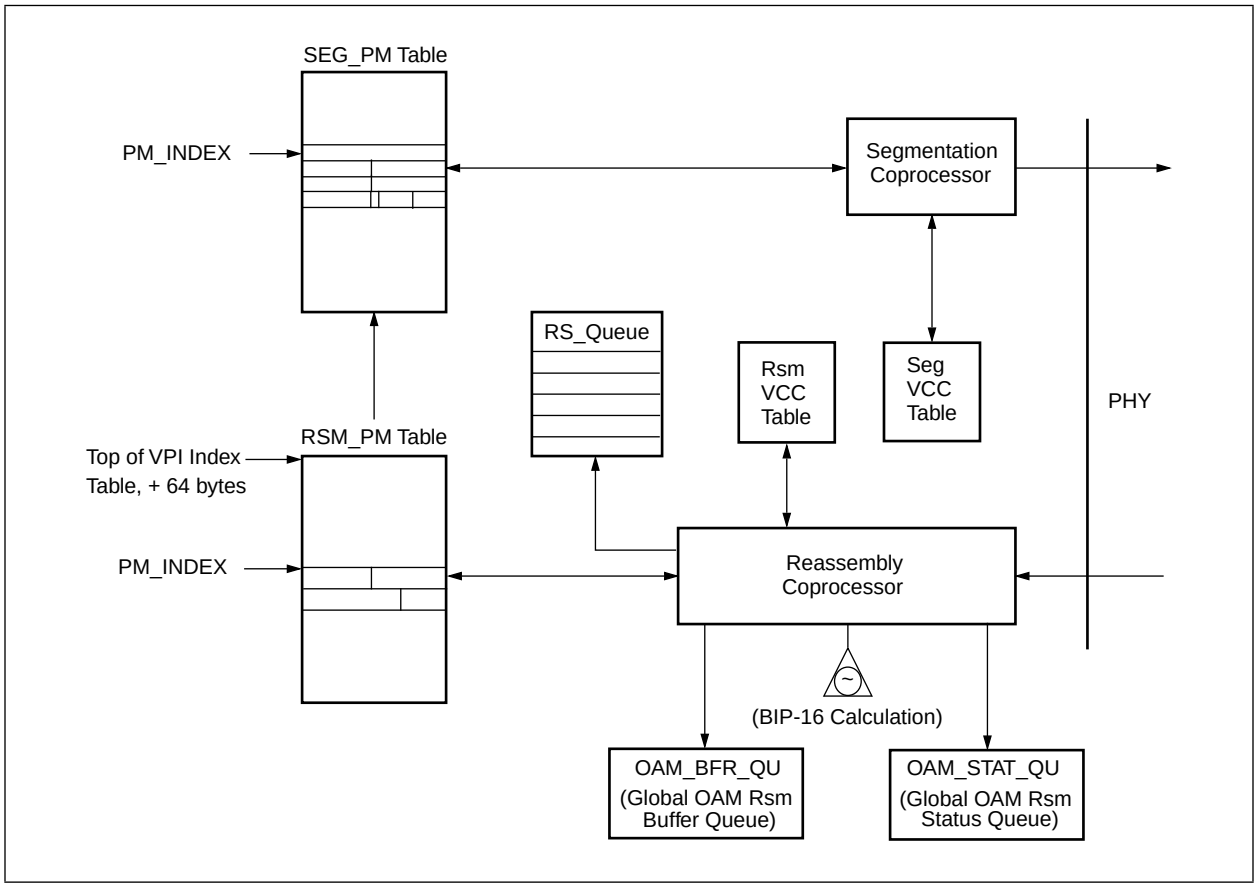
The Rsm coprocessor computes the CRC-10 error detection code each with received OAM cell's information fields. If an error is detected (i.e., the computed CRC value does not match the CRC-10 value written in the cell), the SAR sets the CRC_ERROR bit in the Status Queue entry to a logic high.

7.4 PM Processing

OAM PM can be enabled for up to 128 VCCs by setting the PM_EN bits in the Rsm VCC Table entry and the Seg VCC Table entry for that channel.

Figure 7-2 illustrates the functional blocks for PM processing.

Figure 7-2. Functional Blocks for PM Segmentation and Reassembly



For any channel on which PM processing is enabled, the Bt8233 performs these functions:

- The Rsm coprocessor reassembles received PM-OAM cells in the global reassembly OAM Buffer Queue (OAM_BFR_QU).
- A Reassembly Status Record is written to the global reassembly OAM Status Queue for each received PM-OAM cell. The status entry for a forward monitoring PM-OAM cell includes the BIPV calculation, TRCC0, and TRCC0+1. The TUC0 and TUC0+1 fields can be extracted directly from the cell payload.
- The Rsm coprocessor performs the BIP-16 calculation on each received data cell in the defined PM block and writes this data to the RSM_PM table entry set aside for that PM_INDEX.
- For each forward monitoring PM cell received, the Rsm coprocessor writes an entry for a backward reporting PM cell in the RS_Queue, causing the Bt8233 to generate and transmit a backward reporting PM cell.
- The segmentation coprocessor automatically generates a forward monitoring PM cell at the end of each PM block. It gets the data for these cells from the SEG_PM table entry for that PM_INDEX. This can be optionally disabled by setting the FWD_MON field equal to zero.

7.4.1 Initializing PM Operation

The user must initialize the following fields, specified in Table 7-2, before starting PM processing:

Table 7-2. PM-OAM Field Initialization For Any PM_INDEX

Register / Table	Field	Initialized Value	Notes
RSM_CTRL1 (Reassembly Control Register 1)	OAM_EN	0–1	
	OAM_QU_EN	0–1	
	OAM_BFR_QU	(User assigned)	
	OAM_STAT_QU	(User assigned)	
SEG_PMBASE (Seg PM Base Register)	SEG_PMB[15:0]	(User assigned)	Base address for SEG_PM table.

7.4.2 Setting Up Channels for PM Operation

When the Bt8233 receives a PM activation cell (OAM Type = 1000, Function Type = 0000), or when the user decides to activate PM processing on a channel, the host (or local) processor must enable PM on the applicable channel. It does this by setting the PM_EN bits in the Rsm and Seg VCC Table entries to logic high, and selecting an unused PM_INDEX (0-127).

The host will then initialize the corresponding SEG_PM and RSM_PM table entries. The format of each entry of the SEG_PM table is illustrated in Table 7-3 on page 151, and the format of each entry of the RSM_PM table is illustrated in Table 7-5 on page 153

The assigned PM_INDEX value for that channel will have to be written to both the Rsm VCC Table entry and the Seg VCC Table entry for that channel.

For F4 flows, each VCI channel in the VPI group must have the PM_EN bits in both the Rsm VCC Table entry and the Seg VCC Table entry set high and the PM_INDEX pointing to the same location. In addition, a Rsm and Seg VCC Table entry must be configured corresponding to VCI=3 or VCI=4.

To initialize backward reporting without forward monitoring on any channel, set the FWD_MON = 0.

Once the SEG_PM and RSM_PM table entries have been initialized, the processor will send an activation-confirmed OAM cell to the originator.

7.4.3 PM Operation

PM processing operates automatically on any channel until stopped by clearing the PM_EN fields in the Seg VCC Table entry and the Rsm VCC Table entry.

PM-OAM cells are not included in the NRM cell count, as part of ER processing. See Chapter 6.0, Traffic Management for details.

7.4.3.1 Generation of Forward Monitoring PM Cells

The segmentation coprocessor generates a forward monitoring PM cell at the end of each PM block, as defined by the BLOCK_SIZE field in the SEG_PM table for any PM_INDEX. It determines the point to generate a forward monitoring cell by following these processes:

- At the point of initialization of PM processing or when a forward monitoring cell is sent, the segmentation coprocessor sets the BLOCK_COUNT field to zero. It also increments Monitoring Cell Sequence Number (MSN), and re-initializes the BIP field to zero.
- As each data cell is segmented, the segmentation coprocessor increments the BLOCK_COUNT number and updates the BIP field.
- When the BLOCK_COUNT number reaches the block size specified by the BLOCK_SIZE field, signifying the end of the PM block, the segmentation coprocessor generates a new forward monitoring PM cell and starts these processes again.

7.4.3.2 Reassembly of Forward Monitoring PM Cells

When the Bt8233 receives a forward monitoring PM cell, the Rsm coprocessor reads the RSM_PM table word pointed to by the PM_INDEX field in the Rsm VCC Table entry. The location of the RSM_PM Table is above the LECID Table. The Block Error Result (BIPV), TRCC0, and TRCC0+1 fields are written to a special RSM-PM forward monitoring status queue entry. The TUC0 and TUC0+1 fields can be extracted directly from the RSM_PM cell payload.

When a new buffer is needed, the reassembly coprocessor uses the global OAM Buffer Queue if RSM_CTRL1(OAM_QU_EN) is a logic high. Otherwise, it uses the BFR0 pool identification number in the Rsm VCC Table to point to the appropriate free buffer queue.

A Rsm Status entry is written for each OAM cell reassembled.

The reassembly coprocessor uses the global OAM Status Queue if the RSM_CTRL1(OAM_QU_EN) bit is a logic high. Otherwise, it uses the STAT field in the Rsm VCC Table to determine which status queue to use for that channel.

7.4.3.3 Reassembly of Backward Reporting PM Cells

Backward reporting cells are reassembled in the same manner as non-PM OAM cells are reassembled.

7.4.3.4 Turnaround and Segmentation of Backward Reporting PM Cells

For each forward monitoring cell received, the Bt8233 also writes the BIPV, TRCC0, TRCC0+1, TUC0, and TUC0+1 fields to the RS_Queue, for further processing by the segmentation coprocessor. The segmentation coprocessor generates a backward reporting cell.

7.4.3.5 Turnaround of Backward Reporting PM Cells ONLY

To enable turnaround of backward reporting PM cells without generation of Forward monitoring PM cells, set the segmentation PM_EN bit in the Seg VCC Table entry to a logic high, and set the PM_INDEX, and set the FWD_MON field to zero.

7.4.4 Error Conditions During PM Processing

If OAM cells are not using the global reassembly OAM buffer pool, then the cells are treated as Single Segment Messages (SSMs) for purposes of the firewall protection. OAM cells using the global OAM buffer pool do not have per-channel protection.

If the RS_QUEUE fills, the PM-OAM information will be dropped, and the RS_QUEUE_FULL status indication will be set.

7.5 OAM Control and Status Structures

Refer to the Control and Data Structures sections of Chapters 4 (segmentation coprocessor) and 5 (reassembly coprocessor) for information on VCC Tables, Buffer Descriptors, the Transmit Queue, and Status Queues, as they apply to generating and processing OAM cells.

The base address of the SEG_PM table is given by the SEG_PMB field in the SEG_PMBASE register. The address of each entry is located at byte address:

```
SEG_PMB*128 + <PM_INDEX>*32
```

The RSM_PM table is located above the LECID table, which is located above the VPI table. The address of each entry is located at byte address:

```
if RSM_CTRL0(VPI_MASK)
RSM_ITB*128 + 1024 + 64 + <PM_INDEX>*16
else
RSM_ITB*128 + 16384 + 64 + <PM_INDEX>*16
```

7.5.1 SEG_PM Structure

Refer to Table 7-3 for the SEG_PM structure. For SEG_PM field descriptions, see Table 7-4.

Table 7-3. SEG_PM Structure

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	ATM_HEADER																															
1	FWD_TUC0																FWD_TUC)1															
2	FWD_PM	BLOCK_SIZE		FWD_MON	RSVD		BLOCK_COUNT										BIP															
3	Reserved								BLER								BCK_MSN								FWD_MSN							
4	BCK_TUC0																BCK_TUC01															
5	TRCC0																TRCC01															
6	Reserved																															
7	Reserved																															

Table 7-4. SEG_PM Field Descriptions

Field Name	Description
ATM_HEADER	ATM header to use for both backward reporting and forward monitoring cells.
FWD_TUC0	Total User Cell number with CLP = 0 for forward monitoring.
FWD_TUC01	Total User Cell number with CLP = 0,1 for forward monitoring.
BLOCK_SIZE	Size in cells of forward monitoring block: 00: block size = 128 01: block size = 256 10: block size = 512 11: block size = 1024
FWD_MON	Set to enable both forward monitoring and backward reporting. Clear to enable only backward reporting.
BLOCK_COUNT	Number of cells in current monitoring block.
BIP	BIP-16 for forward monitoring.
BLER	Block Error Result for backward reporting cells.
BCK_MSN	Monitoring Cell Sequence Number for backward reporting cells.
FWD_MSN	Monitoring Cell Sequence Number for forward monitoring cells.
BCK_TUC0	TUC0 field for backward reporting cells.
BCK_TUC01	TUC01 field for backward reporting cells.
TRCC0	Total Received Cell Count with CLP = 0 for backward reporting.
TRCC01	Total Received Cell Count with CLP = 0,1 for backward reporting.

7.5.2 RSM_PM Table

Refer to Table 7-5 for RSM_PM table entry. For RSM_PM table field descriptions, see Table 7-6.

Table 7-5. RSM_PM Table Entry

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Reserved																								MSN							
1	BIP16																BCNT															
2	TRCC0																TRCC0+1															
3	Reserved																															

Table 7-6. RSM_PM Table Field Descriptions

Field Name	Description
BIP16	The BIP-16 error detection code generated over the payloads of the user information cells in the PM block.
BCNT	Block Count. This field contains the calculated number of cells in the current PM block.
MSN	Monitoring Cell Sequence Number for forward monitoring PM cells. Backward reporting PM cells are not included in this sequence. This field allows for the detection of lost or mis-inserted PM cells containing forward monitoring information.
TRCC0	Total received cell count with CLP = 0.
TRCC0+1	Total received cell count.

8.0 DMA Coprocessor

8.1 Overview

The DMA coprocessor is intended to perform high-speed sustained data transfers to and from the host memory space. It is controlled by the segmentation and reassembly coprocessors.

The major functions of the DMA coprocessor are to transfer data from the host memory (through the PCI bus) to the segmentation coprocessor and to transfer data from the reassembly coprocessor to the host memory space (through the PCI bus).

In all modes of operation, the DMA coprocessor maintains a high level of performance. It uses burst transfers, when possible, to maximize utilization of the host bus bandwidth, performs byte switching to accommodate misaligned transfers, and carries out concurrent input and output transfers (alternating burst reads and burst writes) to support simultaneous input and output data streams.

8.2 DMA Read

For outgoing messages, DMA read cycles move data from host memory to the segmentation coprocessor using a gather DMA method. The maximum burst size is thirteen 32-bit words which corresponds to one cell. The burst size can be reduced by setting the MAX_BURST_LEN field in the PCI Configuration Register at a value of less than 13 words.

8.3 DMA Write

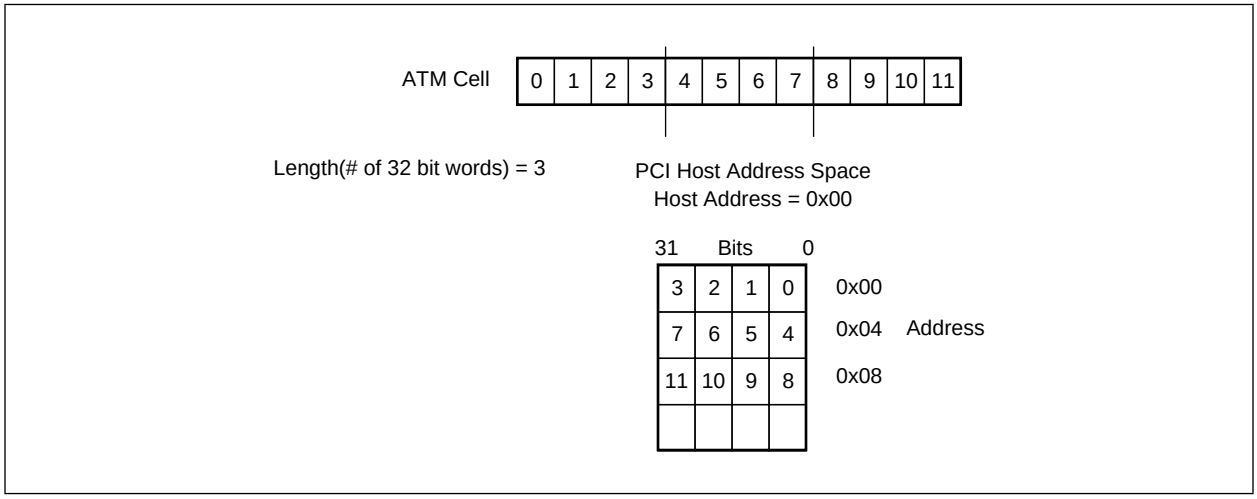
For incoming messages, DMA write cycles move data from the reassembly coprocessor to host memory using a scatter DMA method. The maximum burst size is fourteen 32-bit words which corresponds to one ATM cell and a status word appended to performance monitoring cells. The burst size can be reduced by setting the MAX_BURST_LEN field in the PCI Configuration Register at a value less than 13 words.

8.4 Misaligned Transfers

The reassembly and segmentation processors handle data internally on word addresses. The DMA coprocessor must be capable of handling transfers from the PCI bus without the same constraint (i.e., with data that is not aligned on word boundaries). In addition, the length of the transfer is specified in bytes, not 32-bit words, even though the data bus widths are all 32 bits.

To facilitate this, byte-switching logic is used within the Bt8233. When the Bt8233 specifies a host address with the LSBs = 00, it is implied that the data is byte-aligned. Figure 8-1 shows how a byte-aligned address would map into the PCI host address space for a little endian system. Selecting between big and little endian systems is done using the ENDIAN [bit 12] in Configuration Register 0 [CONFIG0;0x14].

Figure 8-1. LIttle Endian Aligned Transfer



When the Bt8233 specifies a host address with the LSBs not equal to 00, it is implied that the data is misaligned. Figure 8-2 shows how a misaligned address would map into the PCI host address space for a little endian system.

Figure 8-2. Little Endian Misaligned Transfer

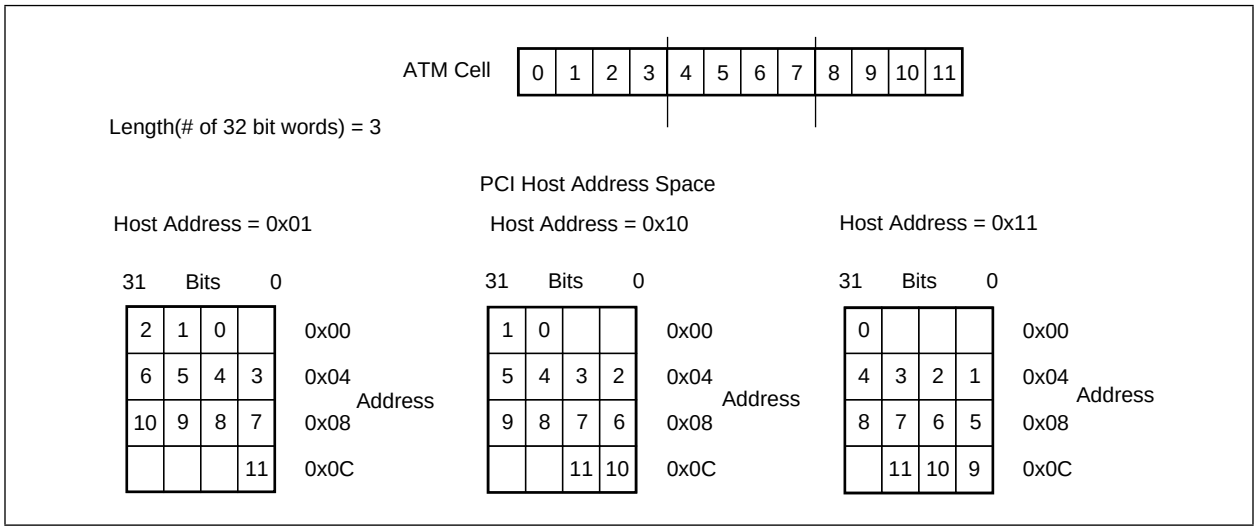
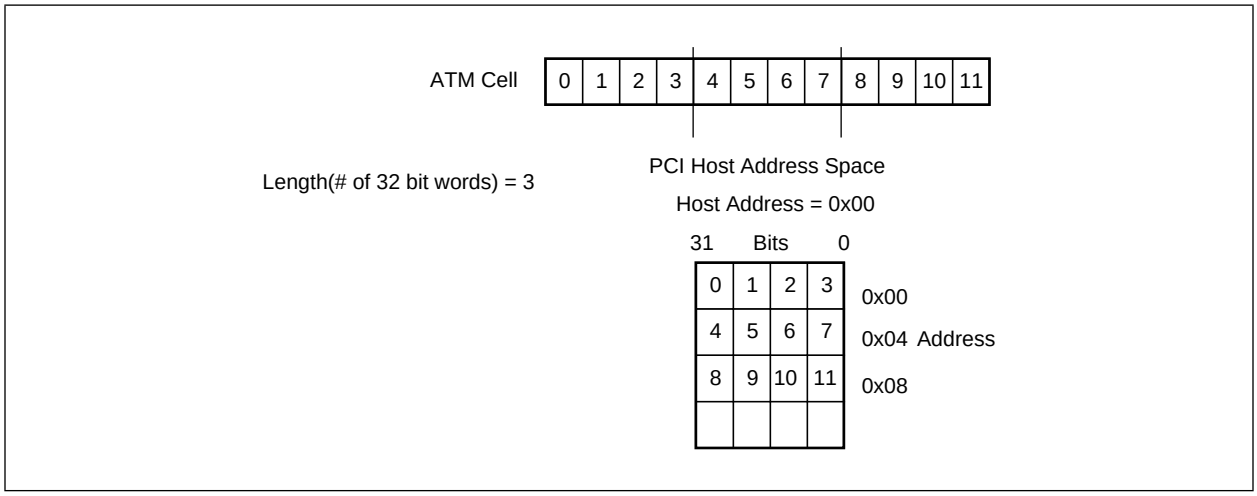


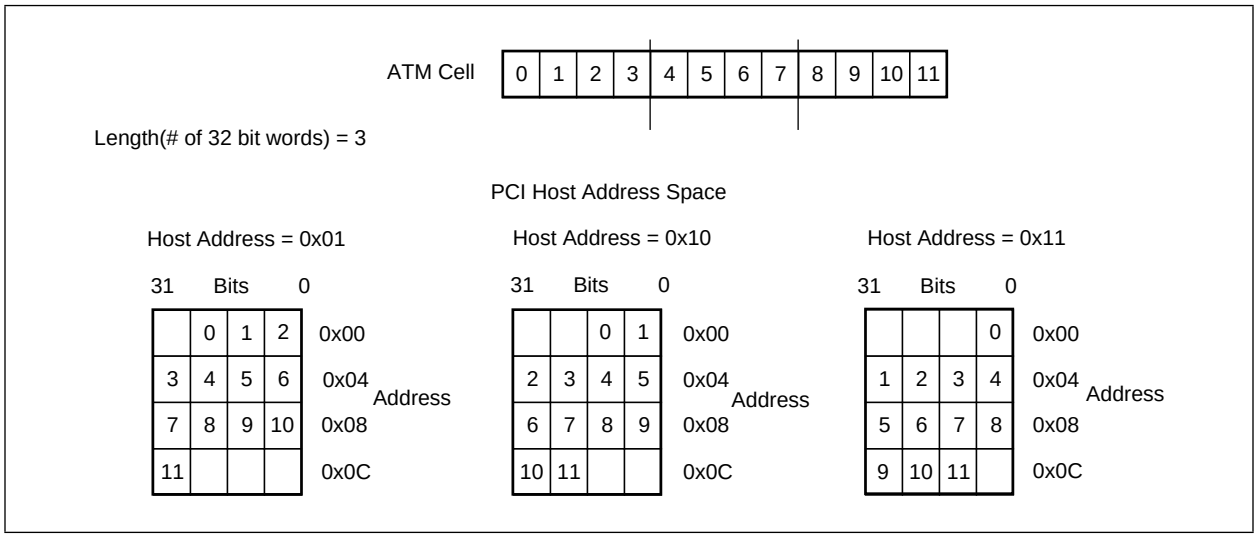
Figure 8-3 shows how a byte-aligned address would map into the PCI host address space for a big endian system.

Figure 8-3. Big Endian Aligned Transfer



When the Bt8233 specifies a host address with the LSBs ≠ 00, it is implied that the data is not aligned. Figure 8-4 shows how an unaligned address would map into the PCI host address space for a big endian system.

Figure 8-4. Big Endian Misaligned Transfer



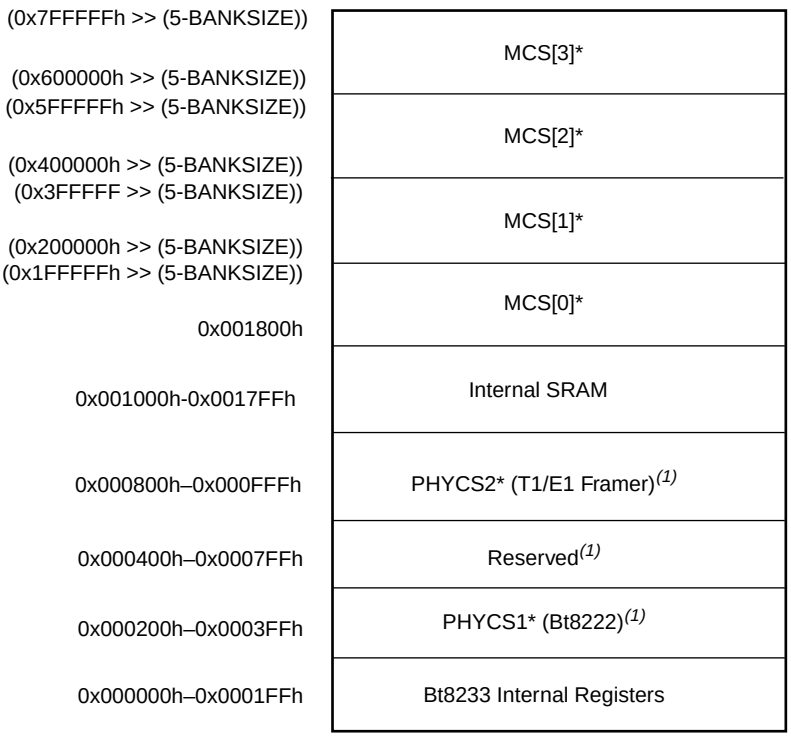
9.0 Local Memory Interface

9.1 Overview

To simplify system implementations, the Bt8233 integrates a complete memory controller designed for direct interface to common Static RAMs (SRAMs). The control and status registers, as well as the physical interface devices in the stand-alone mode of operation, are mapped into the bottom of the memory map. Consequently, accesses to these resources are also controlled by the memory controller. Figure 9-1 shows the Bt8233 address map.

Up to 8 MByte of external memory using SRAM devices can be accessed by the Bt8233. The amount of memory required is heavily dependent on the number of VCCs implemented, as well as the number of VCCs that are currently active. Memory requirements are discussed in detail in Section 9.3, Memory Size Analysis.

Figure 9-1. Bt8233 Memory Map



Note: All addresses in this illustration are byte addresses
Notes: (1). These device selects are available in standalone mode only; otherwise, this memory is mapped to MCS[0]*.

9.2 Memory Bank Characteristics

The external memory is organized in 1 to 4 banks of up to 2 MByte each. The system may use any number of banks to fulfill the memory requirements. The only requirement is that the banks must be of the same size and organization. The local processor selects between the banks via the PBSEL[1,0] inputs. BANKSIZE[2:0] [bits 9–7] in the CONFIG0 register denote the size of the memory banks and allow the Bt8233 to incorporate the various bank sizes into contiguous memory. Table 9-1 gives the coding of the BANKSIZE[2:0] control bits.

Table 9-1. Memory Bank Size

BANKSIZE	Bank Memory Organization	Total Bank Size (Bytes)	PBSEL[1,0] Connection	Typical Implementation
111	Reserved			
110	Reserved			
101	512 K x 32	2 M	A[22:21]	Four 512 K x 8
100	256 K x 32	1 M	A[21:20]	Two 256 K x 16, Eight 256 K x 4
011	128 K x 32	512 K	A[20:19]	Four 128 K x 8
010	64 K x 32	256 K	A[19:18]	Two 64 K x 16, Eight 64 K x 4
001	32 K x 32	128 K	A[18:17]	Four 32 K x 8
000	16 K x 32	64 K	A[17:16]	Two 16 K x 16, Eight 16 K x 4

The memory controller is designed to work with standard by_8 and by_4 SRAM devices as well as with by_16 devices. Grounding the RAMMODE input selects the by_4 or by_8 mode of operation, while pulling RAMMODE to a logic one selects by_16 operation. When by_16 operation is selected, the MWE[3:0]* outputs become byte enables for both reads and writes. Figure 9-2 shows a typical half MB bank implementation using by_8 SRAM devices. Figure 9-3 shows a typical 1 MB bank using by_16 RAM. To connect different sized RAM banks, simply use more or less address bits; all other control remains the same.

NOTE: The number and type of SRAM chips used affect the address and data bus capacitance and, therefore, the ac timing specifications and the required SRAM speed. Also note that the use of by_4 devices causes more address bus loading than the use of by_8 or by_16 devices. See Chapter 15.0, Electrical/Mechanical Specifications, for detailed timing information.

Figure 9-2. 0.5 MB SRAM Bank Utilizing by_8 Devices

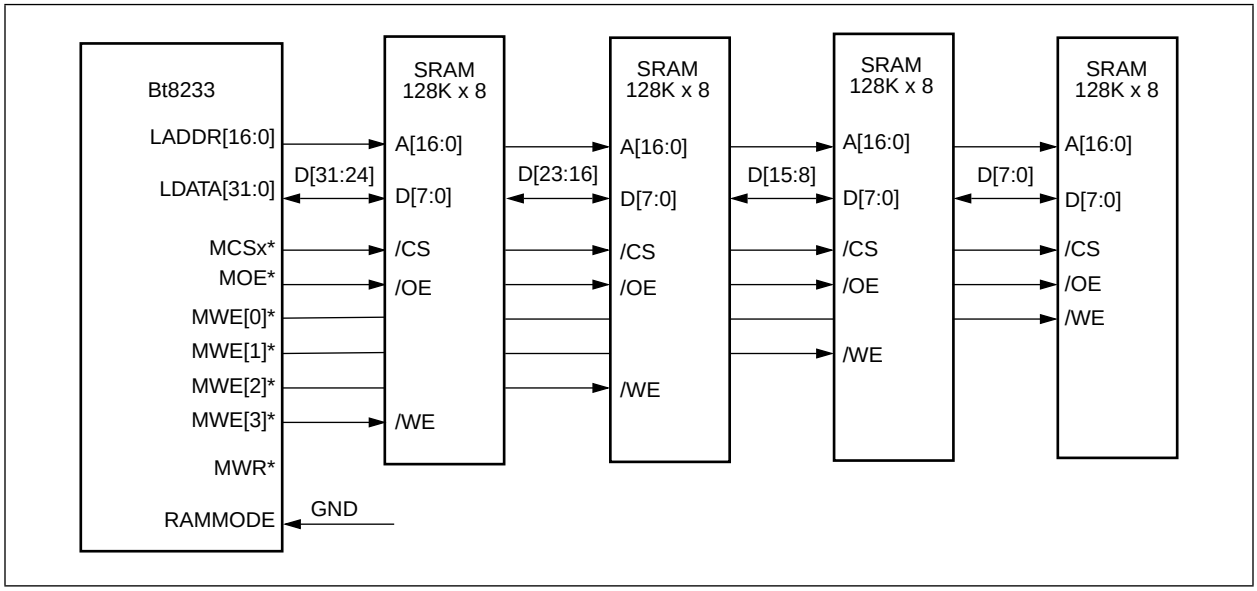
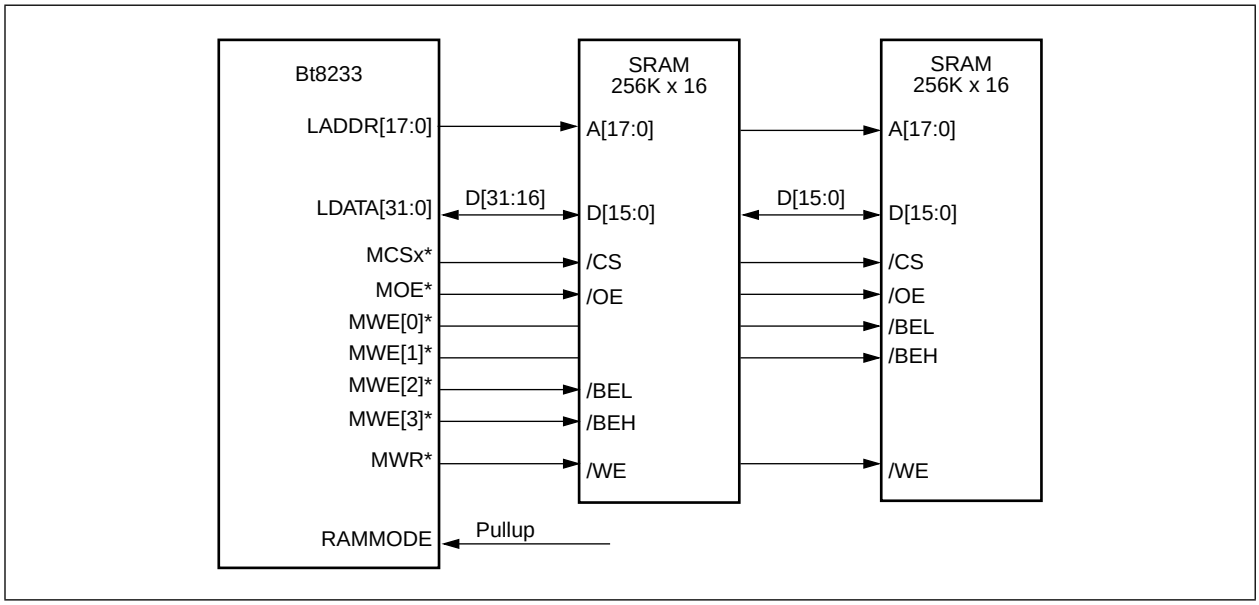


Figure 9-3. 1 MB SRAM Bank Utilizing by_16 Devices



The memory map contains space allocated to the Bt8222 physical interface IC, and to a future Rockwell T1/E1 framer. This mapping is valid only when the PROCMODE input pin is pulled high, indicating standalone operation with no local processor present. Standalone operation is detailed in Section 10.6, Standalone Operation, of Chapter 10.0, Local Processor Interface. When PROCMODE is logic low and the local processor is present, addresses 0x100h through 0xFFFh are available for general use and are mapped to MCS[0]*****.

The MEMCTRL bit in the CONFIG0 Register selects the number of wait states that the memory controller uses to access the SRAM. A logic 0 indicates zero wait state or single-cycle memory, while a logic 1 indicates one wait state or two-cycle memory. The power-on default is MEMCTRL = 1, selecting one wait state or two-cycle memory accesses.

Accesses made to the control registers and Internal SRAM by the local processor follow the convention for SRAM accesses, that is, either zero or one wait state depending on MEMCTRL programming. Subsequently, the local processor sees no functional timing differences between accesses to registers or SRAM. The internal register accesses from the PCI slave interface are always zero wait state.

SRAM access time requirements are directly proportional to the system clock speed, as well as the amount and organization of the memory. The required system clock speed for a given application is dependent on the physical line rate, number of VCCs, and the percentage of idle cells versus assigned cells. Memory access times and other requirements are specified at three typical implementations of 1, 2, and 4 banks of by_8 SRAM. In terms of address bus loading, one bank of by_8 SRAM equals one-half bank of by_16 or two banks of by_4. In this way, the system designer may choose the appropriate SRAM characteristics to suit the amount of memory and organization required for the application. See Chapter 15.0, Electrical/Mechanical Specifications for timing information.

9.3 Memory Size Analysis

Table 9.2 gives the segmentation memory size requirements for 1024 configured VCCs under the following assumptions:

Segmentation

1. Schedule Table is 2112 schedule slots, and each slot is a double word. This allows CBR and three-priority VBR schedule in 64 kbits/sec increments for an OC-3 connection.
2. 8 ABR Templates.
3. 32 OAM PM channels active.
4. Transmit queues configured for 256 entries.
5. No status queues in SRC local memory.
6. Each active channel has an average of one active segmentation buffer.
7. RS_Queue size is 1024.

Reassembly

1. 8 VPIs per 1024 channels.
2. 1K preallocation on VPI=0 only.
3. UNI VPI space.
4. 32 OAM PM channels active.
5. Free Buffer Queues configured for 256 entries.
6. No status queues in SRC local memory.
7. FBQ pools 0-15 have 4-word entries and pools 16-31 have 2-word entries.

Table 9-2. Memory Size in Bytes

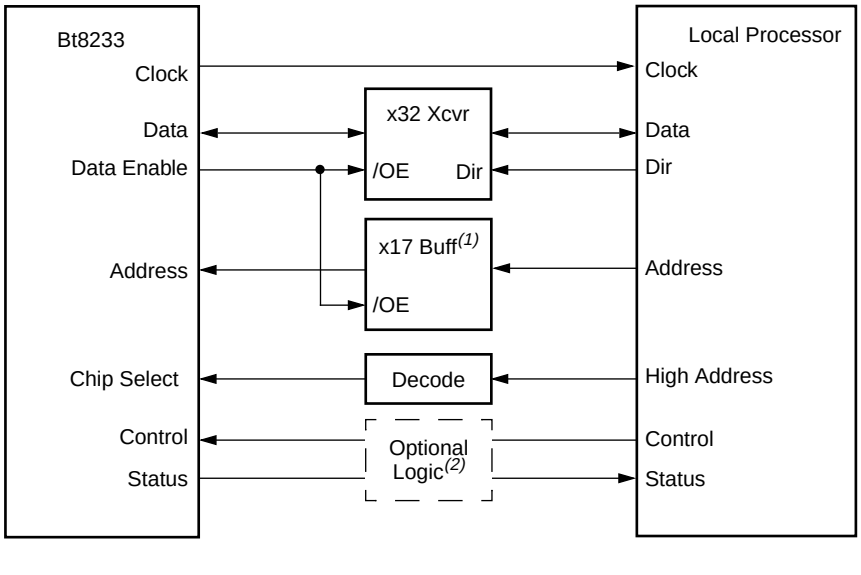
Data Structure	One Peer	16 Peers	32 Peers
Seg Schedule Table	16896	16896	16896
Seg SEG_PM	512	512	512
RS Queue	8192	8192	8192
Seg Transmit Queues	1024	16384	32768
Seg ER Tables	67200	67200	67200
Rsm Free Buffer Queues	4096	65536	98304
Rsm PM-OAM	256	256	256
Rsm VPI Index Table	1024	1024	1024
Total Fixed	99200	176000	225152
Seg VCC Table	40960	40960	40960
Seg Buffer Descriptors	20480	20480	20480
Rsm VCC Table	49152	49152	49152
Rsm VCI Index Table	92	92	92
Total Incremental	110684	110684	110684
Grand Total	209884	286684	335836

10.0 Local Processor Interface

10.1 Overview

The Bt8233 integrated circuit may be used in conjunction with an external processor that performs initialization, link management, monitoring, and control functions. The local processor interface consists of a loosely coupled architecture where it interfaces to the Bt8233 through bidirectional transceivers and buffers. These transceivers and buffers are controlled by the local processor and the Bt8233, as shown in Figure 10-1. The architecture allows the processor access to all of the Bt8233 SRC shared memory and control registers, while insulating the Bt8233 from processor instruction and data cache fills. This also allows the local processor the option to control multiple Bt8233 and/or physical interface devices.

Figure 10-1. Bt8233—Local Processor Interface



Notes: (1). Required to address full 8 MByte range, typically fewer address lines are used.
(2). Required for non-i960 processors.

The processor interface is a generic synchronous interface based on the Intel i960CA 32-bit architecture and is completely compatible with the i960CA/CF and the new i960Jx processors. Other synchronous and asynchronous processors (e.g., from Motorola, AMD, IDT) can be interfaced using external circuitry. The only requirement is that the processor have a 32-bit bus and that the control signals be synchronized to SYSCLK.

To access the Bt8233 SRC shared memory or control registers, the processor must arbitrate with the Bt8233 for access to the memory controller. Due to the requirements of reassembly and segmentation access to SRAM, and the implications of PCI bus utilization, the local processor has the lowest priority in the memory arbitration scheme. Since the local processor is typically used for low bandwidth supervision and maintenance functions, this should be acceptable.

When the SRC shared processor accesses the Bt8233's control registers, Internal SRAM, or SRC shared memory, a local processor memory request is generated internal to the Bt8233. The memory arbiter then coordinates this request with requests from other memory consumers and grants the memory bus to the local processor at the appropriate time. The local processor is held off during this process by the insertion of a variable number of wait states, accomplished by the i960 withholding READY* or RDYRCV*. Once the local processor is granted the memory system, the transceivers are enabled to allow the local processor's address and data to access the SRAM or control registers. The conclusion of the data transaction is signaled by the assertion of PRDY*. Wait states may be inserted by the processor at any time by asserting PWAIT**. The last data cycle in a burst is indicated by the PBLAST* signal. In this manner, non-i960 processor half-speed buses or slow transceivers can be accounted for.

The LP_BWAIT bit in the CONFIG0 Register will automatically add a single wait state between the first access in a burst and subsequent accesses. This can be used to simplify the design of memory controllers for processors that do not produce a wait output and which require more time between data cycles in a burst.

10.2 Interface Pin Descriptions

The local processor bus interface consists of the control, address, and status signals described in Table 10-1. Reference Table 2-1 and Figure 10-7, i80960CA/CF interface description for further information on these interface pins.

Table 10-1. Processor Interface Pins

Signal	Dir(1)	Description
PROCMODE	I	Processor interface mode select input—A logic low on this input enables the local processor mode of operation.
PCS*	I	Processor interface chip select—A logic low on this signal in conjunction with a logic low on PAS* at the rising edge of SYSCLK initiates a memory request to the memory controller.
PAS*	I	Processor address strobe— A logic low on this signal in conjunction with a logic low on PCS* latches the value of PWNR, PBSEL[1,0], PADDR[1,0], and PBE[3:0]* at the rising edge of SYSCLK.
PWNR	I	Processor write/read select—A logic one on this input indicates a write cycle, a logic zero indicates a read cycle. Latched at rising edge of SYSCLK when PAS* and PCS* are active.
PADDR[1,0]	I	Word select address inputs—Indicates the word address for a single cycle access, or the first word for a multi-cycle burst access. Latched at rising edge of SYSCLK when PAS* and PCS* are active.
PBSEL[1,0]	I	Bank select inputs—Decode to select MCS[3:0]*. Latched at rising edge of SYSCLK when PAS* and PCS* are active.
PBE[3:0]*	I	Byte select inputs—Active low. Allows individual bytes of selected word to be written. Not active on reads. Latched at rising edge of SYSCLK when PAS* and PCS* active. PBE[3]* controls writes to LDATA[31:24], PBE[2]* controls LDATA[23:16], etc.
PWAIT*	I	Processor wait input—Allows the processor to insert a variable number of wait states to extend memory transaction. Must be active on rising edge of SYSCLK with PRDY* active to insert wait cycle. May be used to interface to half speed or slow processor bus or to allow the use of slow transceivers. If the insertion of wait states is not required, set this input to a logic high. This signal may only be active, logic low, when PBLAST* is a logic high.
PBLAST*	I	Processor burst last input—Indicates the last word of a cycle. Must be active on rising edge of SYSCLK with PRDY* active to indicate last cycle. If burst accesses and wait cycles generated by PWAIT* are not required, this signal should be set to a logic low.
PRDY*	O	Processor interface ready signal—A logic low on this signal at rising edge of SYSCLK indicates that the present cycle has been completed. If a read cycle, the data is valid to latch by the processor; if a write cycle, the data has been written and may be removed from the bus. When PRDY* is active, wait states may be inserted with PWAIT*, or a single or burst cycle may be terminated by PBLAST*(2).
PFAIL*	I	The local processor can indicate a failure of its internal self-test or initialization processes by asserting the PFAIL* input to the Bt8233.

Note: The processor system is responsible for controlling the direction of the bidirectional data bus transceiver. In the i960 architecture, this may be controlled by the DT/R* signal.

Notes: (1). Direction given with respect to the Bt8233.
(2). This output corresponds to the READY* or RDYRCV* input in the i960 architecture.

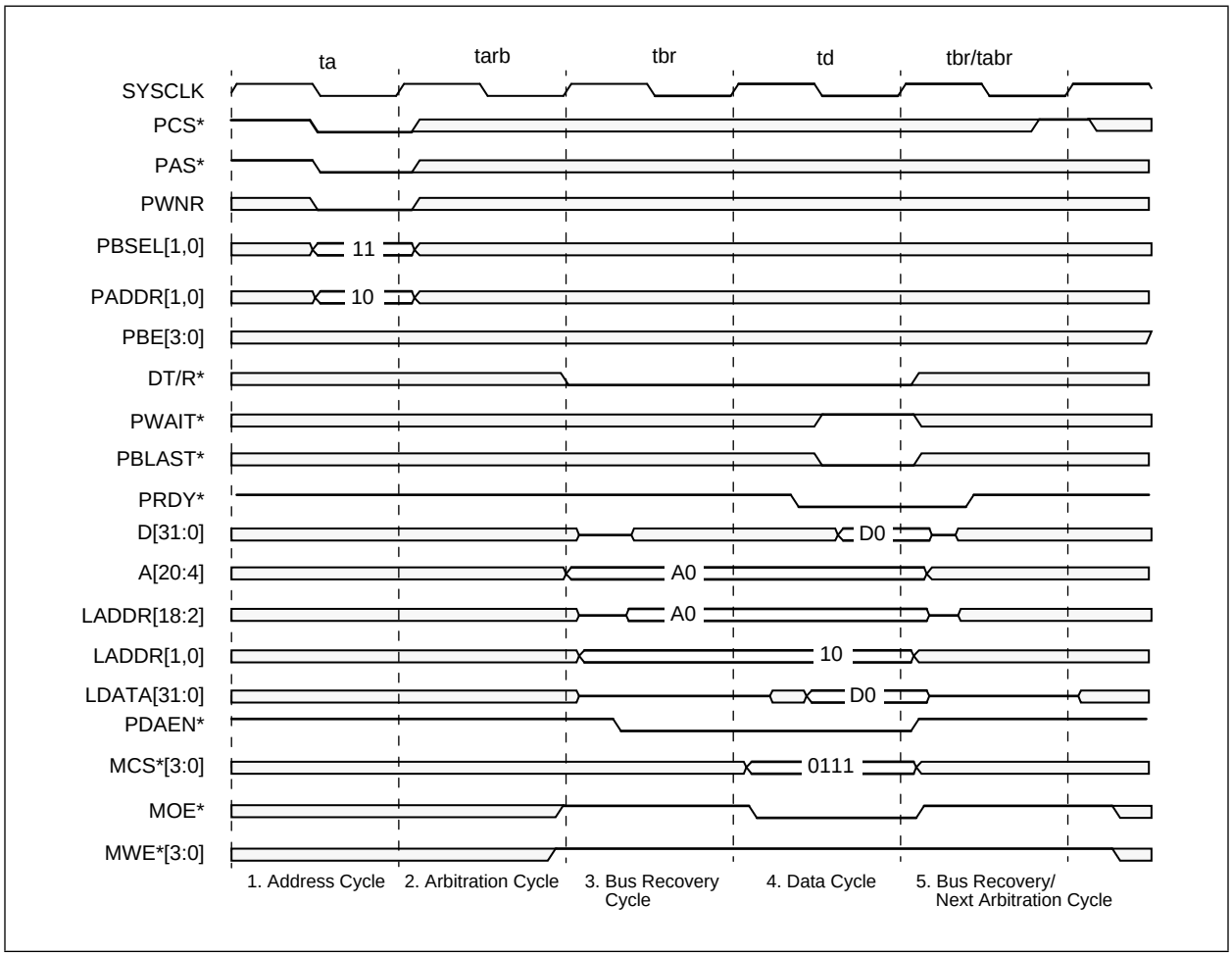
10.3 Bus Cycle Descriptions

Throughout the bus cycle descriptions, cycle refers to a single SYSCLK cycle ending with a rising edge. An arbitration cycle is one in which the memory requests from the local processor and internal memory consumers are compared, and the one with the highest priority is granted the memory access on the next cycle. A memory access that was previously arbitrated can occur on an arbitration cycle. Once the local processor has successfully acquired the memory controller, it holds the bus until it is relinquished by the assertion of PBLAST* on the last data cycle. Therefore, local processor burst transfers will always be completed and can theoretically be of arbitrary length. In practice, however, burst transfers should be limited to four or less. The maximum arbitration delay for a local processor access is on the order of 20 cycles; however, it will typically be from one to four cycles. This parameter is heavily influenced by the SYSCLK frequency, line rate, number of VCCs, idle cell ratio, and SRAM access speed. Therefore, a system design in which local processor accesses must occur within a fixed time period is not recommended.

10.3.1 Single Read Cycle, Zero Wait State Example

Figure 10-2 illustrates a single read cycle with zero wait states. During the address cycle (cycle 1) at the rising edge of SYSCLK with PCS* and PAS* active, a memory request is generated by the processor interface circuitry. Also at this time, the read/write select, bank select, and word select inputs (PWNR, PBSEL[1,0], and PADDR[1,0]) are internally latched. The byte enables (PBE[3:0]*) are don't cares during reads. During cycle 2, this local processor memory request is processed by the memory arbitration circuitry. If no other memory consumers request an access on the same cycle, the local processor is granted access on cycle 3. However, to take into account bus transceiver turnaround, cycle 3 is always a wait or bus recovery state. This gives sufficient time for the address from the processor to access the SRAM. For zero wait state SRAM, unless a wait state is inserted by the processor, the data is available to be latched into the processor on cycle 4, which is indicated by the assertion of PRDY*. Cycle 5 is an arbitration cycle for the internal memory consumers which may have requested access during the processor access. It also serves as a bus recovery cycle for the processor. Once the PCS*, PAS*, PWNR, PBSEL[1,0], and PADDR[1,0] are sampled at cycle 1, they are don't cares for the remainder of the access. Additionally, DT/R* is an output supplied by the local processor to indicate the direction of the data transceivers, and the Bt8233 PDAEN* signal is active to enable data and address.

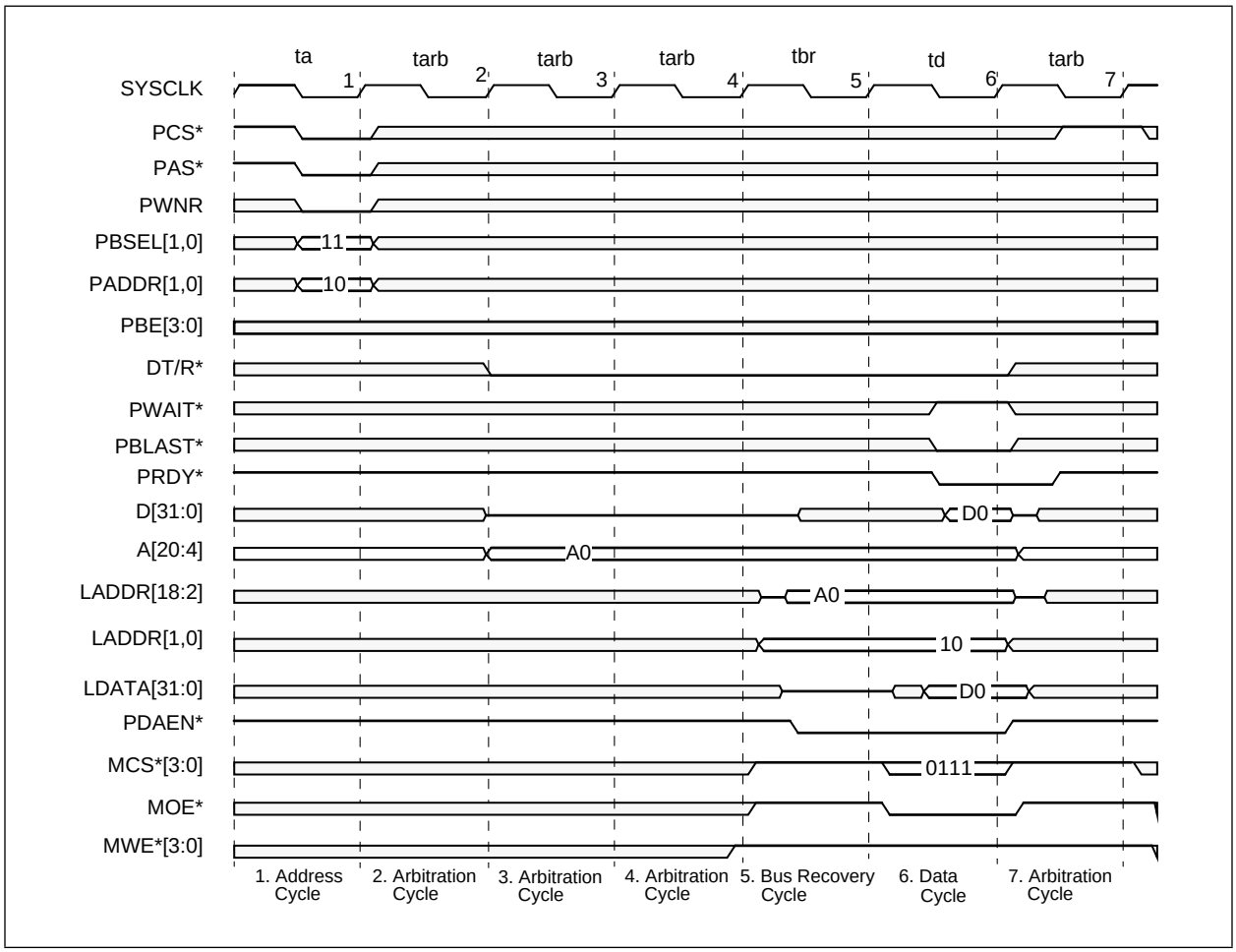
Figure 10-2. Local Processor Single Read Cycle



10.3.2 Single Read Cycle, Wait States Inserted by Memory Arbitration

Figure 10-3 illustrates a local processor single read cycle with arbitration wait states. This example is similar to the preceding one, except that here the local processor is not able to access the RAM immediately because of higher priority memory requests on cycles 2 and 3. On cycle 4, the memory controller allows the local processor access to the address and data bus, and the transaction takes place at the end of cycle 6.

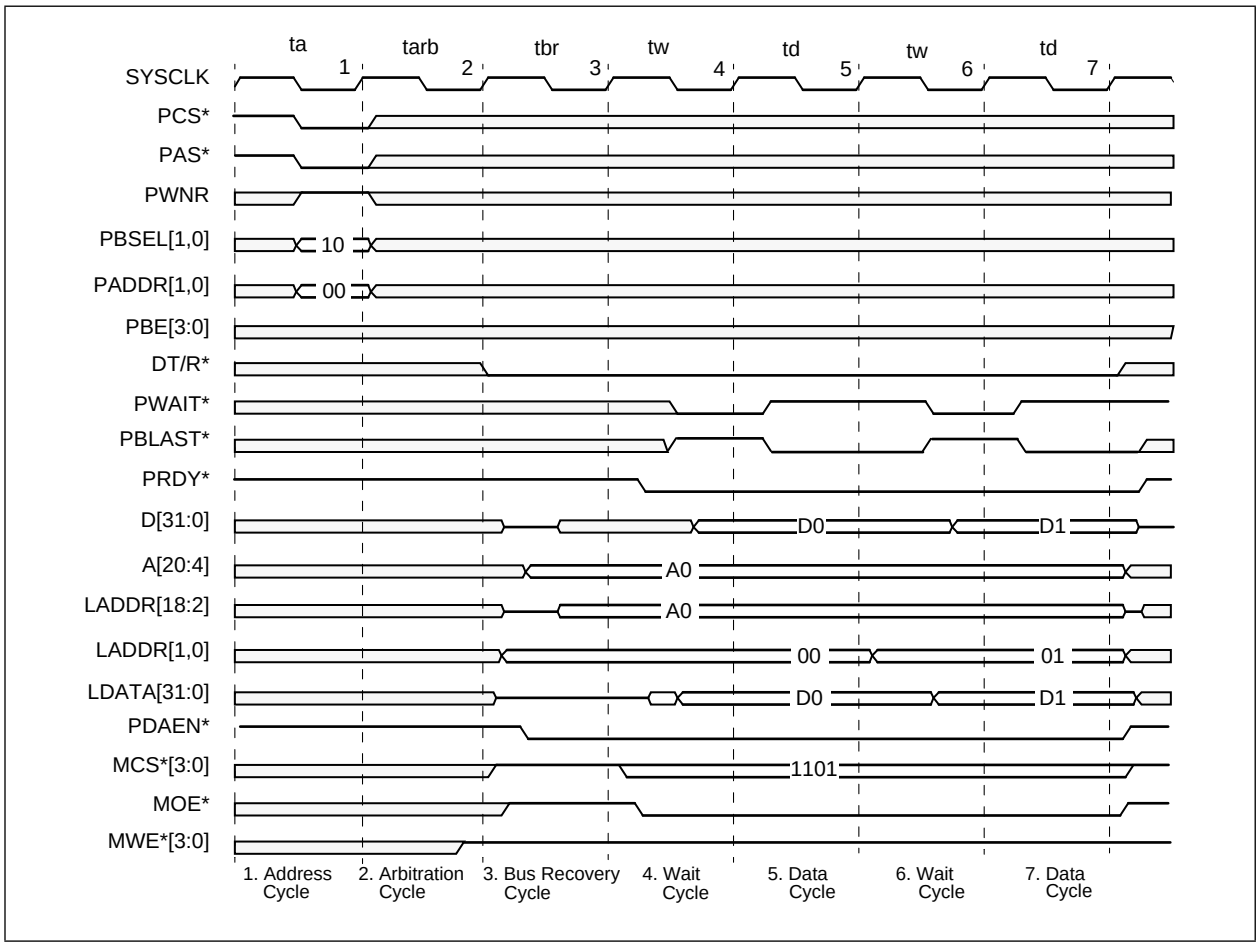
Figure 10-3. Local Processor Single Read Cycle with Arbitration Wait States



10.3.3 Double Read Burst with Processor Wait States

In Figure 10-4 the processor inserts wait states on cycle 4 and cycle 6 to allow additional time for the reads to occur. At the rising edge of SYSCLK on cycle 4 and cycle 6, the combination of PWAIT* low and PRDY* low extends the read by one more cycle. The local processor word select inputs (PADDR[1,0]) are latched at cycle 1, and the Bt8233 word select address lines, LADDR[1,0], are incremented automatically at the beginning of cycle 6.

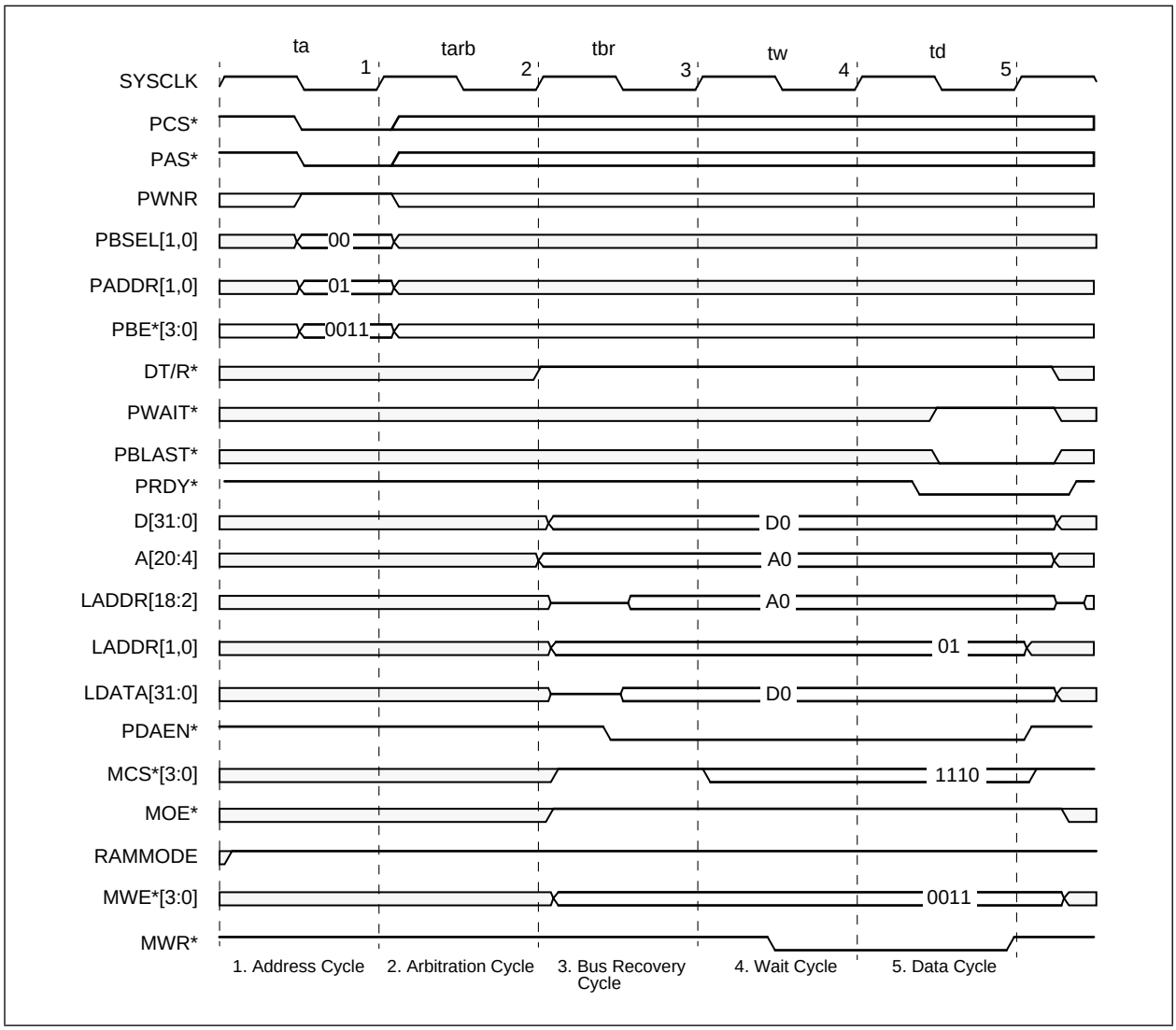
Figure 10-4. Local Processor Double Read with Wait States Inserted



10.3.4 Single Write With One-Wait-State Memory

In Figure 10-5, the local processor performs a single write into one-wait-state memory. There is no arbitration delay. Note that RAMMODE is a logic high, indicating that by_16 RAM is used. Here the PBE[3:0]* inputs are latched at cycle 1 and are used to select the byte enables that are active during the cycle when MWR* is active. In this case, the two most significant bits are active, indicating a 16-bit write to the two most significant bytes.

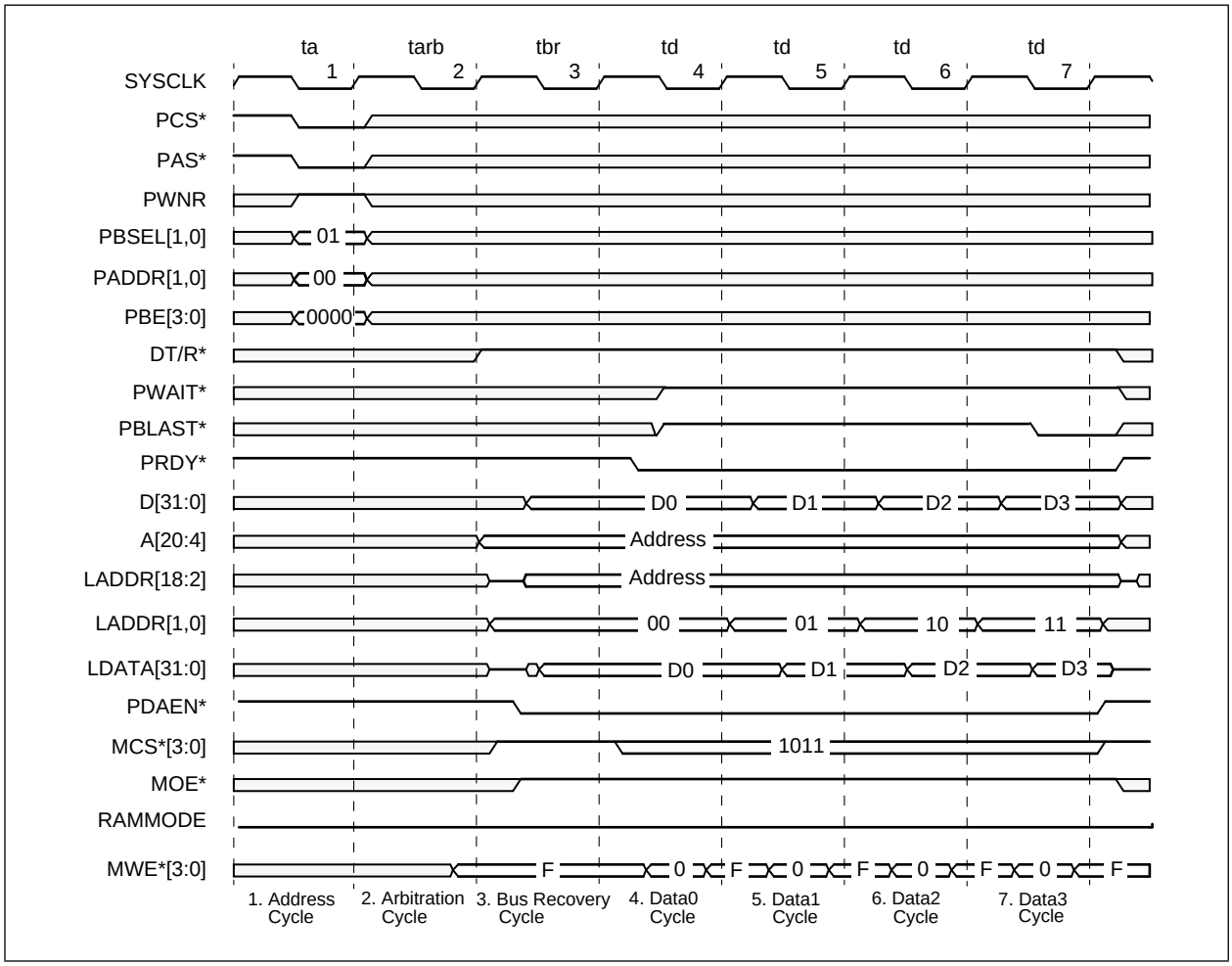
Figure 10-5. Local Processor Single Write with One Wait State by_16 SRAM



10.3.5 Quad Write Burst, No Wait States

In Figure 10-6, a quad burst write access to zero-wait-state memory is shown. RAMMODE is logic low, selecting by_8 or by_4 RAM mode. Here PBE[3:0]* is latched on cycle 1, indicating that the write is active on all bytes, and the MWE*[3:0] outputs are active as write strobes while MWR* is not used. The SRC shared memory word select addresses, LADDR[1,0], are incremented automatically by the Bt8233 on each successive write cycle. Although the i960 architecture has the limitation that a quad word transfer must start on a quad word boundary, the Bt8233 does not have that limitation; the PADDR[1,0] bits may be any value and are incremented as long as the burst transfer proceeds.

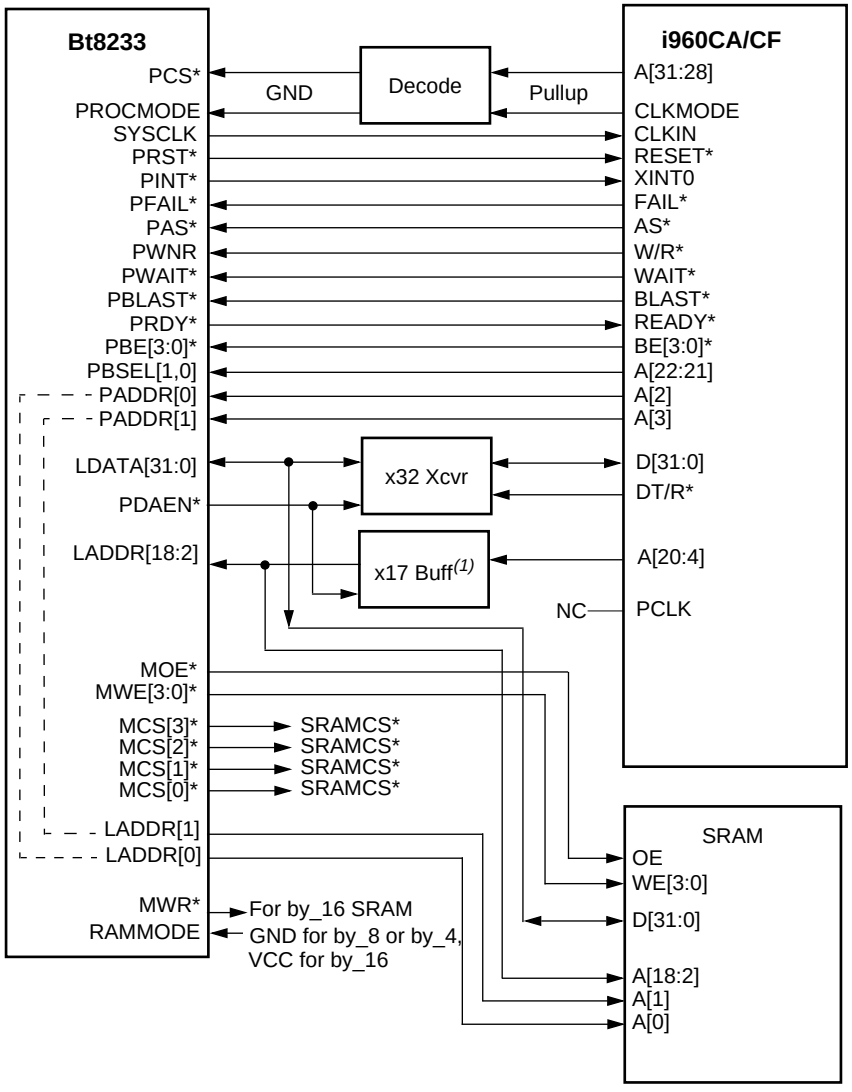
Figure 10-6. Local Processor Quad Write, No Wait States



10.4 Processor Interface Signals

Figure 10-7 illustrates the signal interface between the Bt8233 device and the i960CA/CF processor. The memory region decoded for PCS* should be set for N_{RAD} and N_{WAD} = 2, N_{RDD} and N_{WDD} = 0 or 1 depending on the use of zero or one wait state SRAM, and N_{XDA} = 1. In addition, external ready control must be enabled, and burst may be enabled or disabled at the system designers option. Pulling up the i960 CLKMODE input to a logic 1 selects the divide-by-one clock mode, making i960 PCLK synchronous to SYSCLK.

Figure 10-7. i960CA/CF to the Bt8233 Interface



Note (1): Required for full 8 MByte address range

This configuration is for addressing the entire 8 MB of SRAM. In the majority of systems, the SRAM requirements will be considerably less. The implications of this are that the PBSEL[1,0] inputs may be driven by lower order address lines, and there will be less than 17 address lines to buffer. Therefore, in most applications, the data transceivers may utilize two x16 parts, such as a 74ABT16245, and the address buffer may utilize a single x16 74ABT16244.

NOTE: The i960CA/CF signals a failure of its internal self-test upon reset or power-up by asserting its FAIL* output. This line is connected to the PFAIL* pin of the Bt8233, and the status of this pin is reflected in the Host Interrupt Status Register [HOST_ISTAT0;0xC0].

10.5 Local Processor Operating Mode

The major difference between the i80960Jx processor and the i80960CA is that the Jx utilizes a multiplexed address/data bus structure while the CA/CF is non-multiplexed. In the Bt8233 system, however, the demultiplexing of address/data takes place on the processor side of the address buffers and does not affect the Bt8233. Otherwise, the Jx has the same bus control signals as the CA/CF with the exception of the WAIT* signal, which the Jx does not possess. The insertion of wait states, if required, must be accomplished by an external memory controller which, in any case, is required for a Jx implementation.

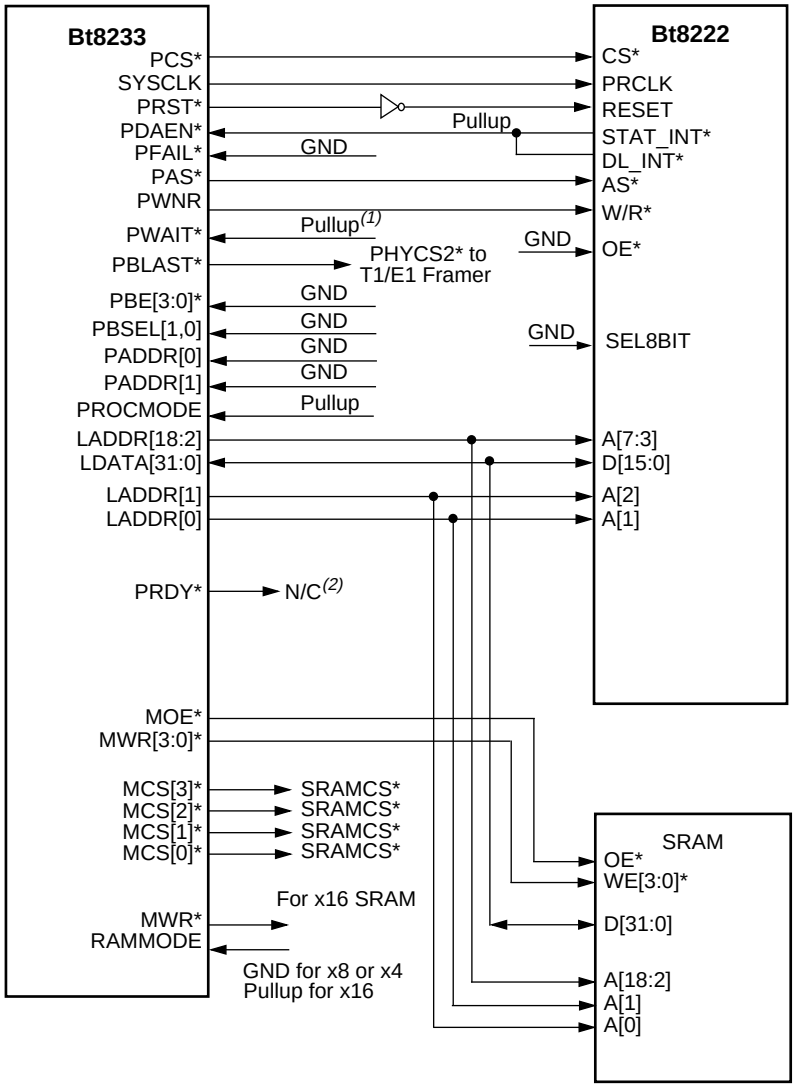
10.6 Standalone Operation

Standalone interface pins and descriptions are given in Table 10-2. Figure 10-8 shows the signal interface between the Bt8233 and the Bt8222 ATM receiver/transmitter device with no local processor. The PCS*, PAS*, and PWNR pins are now outputs providing chip select, address strobe, and write/read control to the Bt8222. PDAEN* is now an input connected to the interrupt sources of the Bt8222. PBLAST* is a second chip select which may be used to connect a future Rockwell T1/E1 framer, since the Bt8222 does not contain this function. The PRDY* output is active and indicates the cycles in which the data transaction occurs. The PWAIT* input is active and can be used to prolong the cycle as shown in Figure 10-9. Physical interface devices other than the Bt8222 can be connected by using PWAIT* to extend the read or write cycle and by using external logic to translate the Bt8233 control signals.

Table 10-2. Standalone Interface Pins

Signal	Dir(1)	Description
PROCMODE	I	Processor interface mode select input. A logic 1 enables standalone operation without a local processor.
PCS*	O	Chip select output for PHY device number 1. Synchronous to SYSCLK.
PBLAST*	O	Chip select output for PHY device number 2. Synchronous to SYSCLK.
PAS*	O	PHY address strobe. Synchronous to SYSCLK.
PWNR	O	PHY write/read select. A logic 1 on this output indicates a write cycle; a logic 0 indicates a read cycle. Synchronous to SYSCLK.
PRDY*	O	PHY interface ready signal. A logic low on this signal at rising edge of SYSCLK indicates that the data cycle has been completed
PWAIT*	I	PHY wait input. Allows external logic to insert wait states to extend data cycles. Only active when PRDY* is active.
PDAEN*	I	PHY interrupt input, active low, level sensitive(2).
PADDR[1,0]	I	Not used, pull to logic 0.
PBSEL[1,0]	I	Not used, pull to logic 0.
PBE[3:0]*	I	Not used, pull to logic 0.
Notes: (1). Direction given with respect to the Bt8233. (2). See the HOST_ISTAT0 Register for details.		

Figure 10-8. Bt8233 to the 8222 Microprocessor Interface (Standalone Operation)



Notes: (1). Notes: May be driven by external circuitry to extend cycles.
(2). May be used by external circuitry.

Figure 10-9. Bt8233/PHY Functional Timing with Inserted Wait States

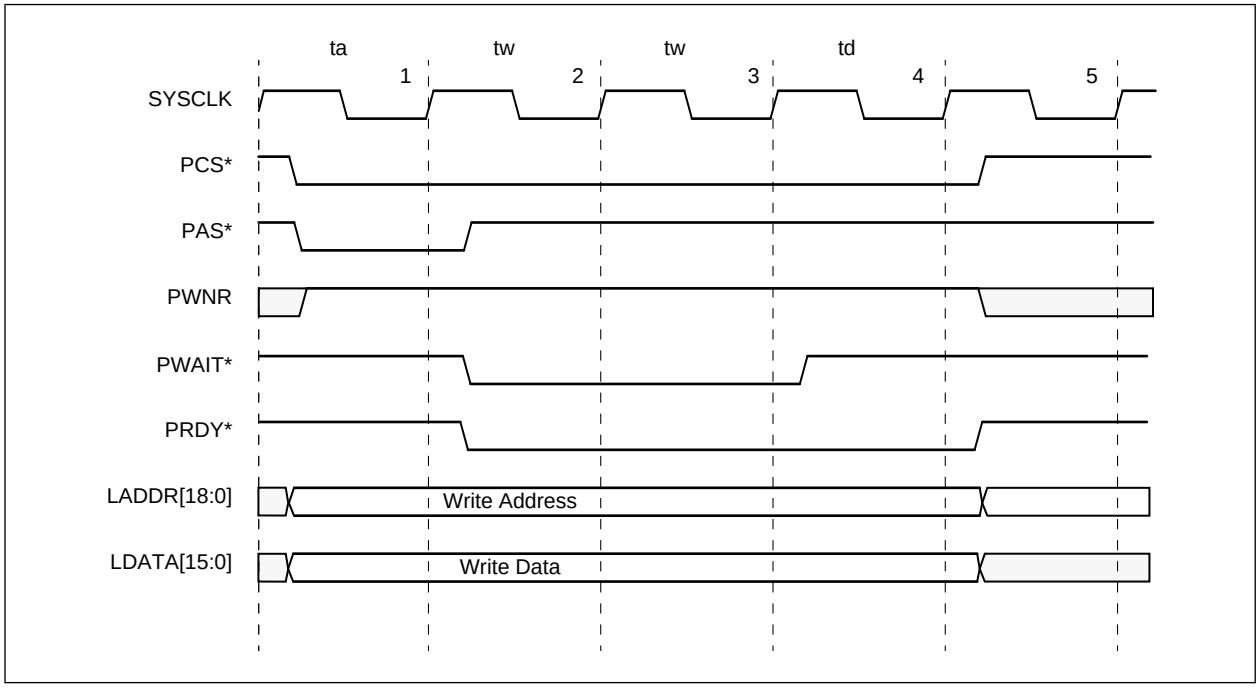
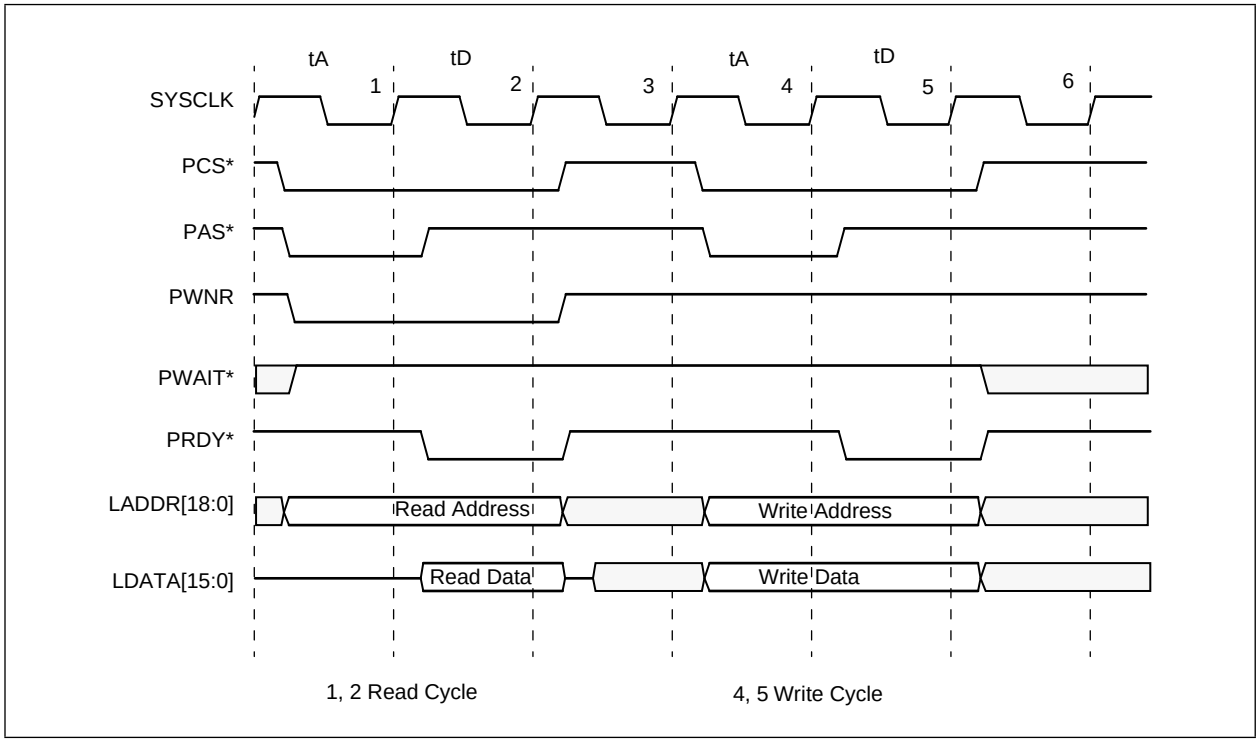


Figure 10-10 shows a read and write Bt8222 access. At cycle 1 (the rising edge of SYSCLK) the Bt8222 samples PCS*, PAS*, and PWNR low, indicating a read cycle. By the next rising edge of SYSCLK at cycle 2, the data is output by the Bt8222 to be latched by the Bt8233. The same procedure occurs for a write except that at cycle 4, PWNR is sampled high. The Bt8222 then latches the data to the appropriate internal register on the next SYSCLK rising edge at cycle 5.

Figure 10-10. Bt8233/Bt8222 Read/Write Functional Timing



10.7 System Clocking

The Bt8233 derives all of its timing from a 2x clock input, CLK2X. This clock is internally divided by 2 to create the system clock, SYSCLK. The system clock is used internal to the device and is output to the system to provide the clock to an external processor or PHY device. All processor interface signals are synchronous to SYSCLK. In addition, a CLK2X output asymmetrically divided by 3 (CLKD3), is provided and may be used as the clock for the UTOPIA ATM physical interface. Alternatively, SYSCLK can be used as the clock source for the UTOPIA ATM physical interface if the frequency is 25 MHz or less. In either case, the clock signal is looped externally to the FRCTRL input. For example, if CLK2X is 66 MHz, then CLKD3 is 22 MHz and is suitable for the UTOPIA interface. If CLK2X is 50 MHz, then SYSCLK is 25 MHz and is suitable for the UTOPIA interface. The CLK2X frequency required for a given application is a function of the physical line rate, number of VCCs, active concurrent VCCs, and the SRAM cycle time.

10.8 Real-Time Clock Alarm

A real-time clock counter and alarm registers are built into the Bt8233. The real-time clock consists simply of a 7-bit prescaler (configured via the DIVIDER field in the CONFIG0 Register) that accepts the SYSCLK input. The prescaler outputs a constant (nominally 1 MHz) pulse train and a 32-bit read/write counter (the Real Time Clock Register [CLOCK;0x00]) that counts the number of pulses output by the prescaler since the system was initialized. When the prescaler is set to generate a 1 MHz pulse train, the CLOCK counter counts in 1 µs intervals. An interrupt is generated when the CLOCK counter overflows, i.e., more than 2^{32} pulses have occurred since it was cleared to zero. If this happens, the CLOCK counter simply wraps around to zero and starts counting over. The control processor or host software is responsible for noting the overflow.

One simple real-time alarm is implemented in the Bt8233. This consists of the Alarm Register 1 [ALARM1;0x04], which is continuously compared to the Clock Register [CLOCK;0x00]. When a match is detected, the corresponding interrupt is generated to the local or host processors. Either processor can then respond to this interrupt and reload a new value into the ALARM1 Register.

10.9 Bt8233 Reset

The Bt8233 must be reset by the host processor prior to system initialization for proper operation. This can be done in one of two ways: by asserting the external HRST* pin (which is normally connected to the system power-up reset circuitry), or by setting the GLOBAL_RESET bit in the Configuration Register 0 [CONFIG0;0x14]. The HRST* pin must be deasserted, and GLOBAL_RESET must be cleared before beginning the Bt8233 initialization process.

Asserting the HRST* input pin automatically causes the local processor reset pin to be driven active, resetting the local processor. The reset to the local processor will stay active until the LP_ENABLE bit in the CONFIG0 Register is set to a logic high. When using the GLOBAL_RESET bit, the processor must manually set or clear the appropriate bits in the CONFIG0 Register.

11.0 PCI Bus Interface

11.1 Overview

The PCI bus interface is compliant with PCI Local Bus Specification, Revision 2.0. With the exception of HRST* and HINT*, this interface is completely synchronous to the PCI bus clock (HCLK). All inputs are sampled at the rising edge of HCLK, and all outputs are driven by the Bt8233 to be valid before the next rising edge of HCLK.

The maximum PCI bus clock rate supported by the Bt8233 is 33 MHz. The PCI bus interface logic is clocked directly from the PCI bus clock, while the remainder of the Bt8233 logic runs off separate clocks. Synchronizing registers and FIFOs are implemented in the PCI bus interface in order to transfer data between the PCI bus clock (HCLK) and the system (SYSCLK) clock domains.

The PCI bus drivers are shared between master and slave bus interface functional blocks. The PCI bus master logic (within the device) arbitrates via the PCI bus arbiter (external to the device) for access to the PCI bus. Access to the PCI bus automatically implies access to the bus drivers, since no other master can be concurrently communicating with the slave logic. The bus master logic contends for the bus on a transaction by transaction basis.

The PCI bus interface responds to read and write requests by the host CPU, allowing access to chip resources by host software. The Bt8233 is also capable of acting as DMA bus master on the PCI bus. As a result, the PCI bus interface implements the full set of address, data, and control signals required to drive the bus as master, and contains the logic required to support arbitration for the PCI bus. Note that the DMA coprocessor and the PCI bus interface are closely linked and, hence, are shown as one unit.

The PCI bus interface functional blocks are:

- I/O drivers and receivers that drive the pins connected to the PCI bus signals.
- PCI bus master logic that allows the bus interface to acquire mastership of the PCI bus and act as a transaction initiator. The bus master logic also contains a command decoder that interprets access commands generated by the DMA coprocessor and a burst controller for controlling the duration of each read or write burst. In addition, the bus master logic contains address counters that allow it to restart and retry burst transfers if required by the transaction target.
- Burst FIFO buffers that store and transfer bursts of data words between the DMA coprocessor and the PCI bus master logic.

- PCI bus slave logic that responds to transactions initiated by other masters on the PCI bus with the Bt8233 as a target. The bus slave logic also synchronizes data passed back and forth across the clock boundary between the PCI bus interface and the internal chip logic.
- Configuration registers holding initialization parameters and PCI bus status information.
- Logic that allows the host CPU to read/write the internal Bt8233 registers via the PCI slave port.
- Logic to enable read/write access to the SRC shared memory space from the host CPU, again via the PCI slave port.

11.2 Unimplemented PCI Bus Interface Functions

The PCI bus interface on the Bt8233 does not implement all the transaction types defined by the PCI bus specification. Only those sections of the protocol that are necessary for slave and DMA memory accesses are implemented. In particular, the following transaction types are not implemented:

- 64-bit transfers, as well as the DAC command.
- Snooping and cache support. Memory Read Line, Memory Write, and Invalidate commands are internally aliased to the Memory Read and Memory Write commands as per the PCI specification.
- Locked and exclusive accesses: the PCI LOCK* line is not driven by the Bt8233, and the PCI slave interface does not handle locked accesses by other bus masters in any special manner.
- I/O accesses (the I/O Read and I/O Write commands).
- Interrupt acknowledge cycles, including the Interrupt Acknowledge command.
- The Special Cycle command and special cycle transactions.
- Burst transfers that do not have simple, sequentially-incrementing addresses for consecutive data phases. The PCI master logic always performs sequentially incrementing burst transfers. The two LSBs of the PCI address lines (AD[1,0]) must be zero during the address phase of any transfer made to the PCI slave logic (indicating sequentially incrementing burst addresses). If AD[1,0] is not equal to zero, the slave logic will signal a type A or B target disconnect after the first data phase, forcing the external master to perform a single-word transfer as per the PCI specification.

11.3 PCI Bus Master Logic

The PCI bus master logic block is responsible for accepting read and write commands from the DMA coprocessor (passed via the burst FIFO buffers) and in turn acquiring mastership of the PCI bus and generating transactions to perform the actual data transfers. The bus master logic contains the following:

- Command decoder which interprets commands issued from the DMA coprocessor.
- Burst controller (that counts off read and write cycles in each burst on the PCI bus
- This controller also latches and drives the address and command during the address phase of each transfer.
- Arbitration logic that acquires control of the PCI bus.
- Bus state machine that sequences and controls transfers.

The bus master implements a software-configurable maximum burst length counter. This counter is started at the beginning of each read or write burst transaction and counts the actual number of words transferred during the burst. When it reaches the value set in the MAX_BRST_LEN field of the PCI configuration space, the burst is terminated and a new address phase is begun.

The addressed slave can request a disconnect or a retry during a read or a write transfer using the defined PCI protocol sequence. In this case, the bus master logic will terminate the current burst, maintain its bus request, and restart the transfer at the point of termination. Disconnects and retries are not regarded as errors.

Five possible sources of error are present during any PCI bus master transaction. If any of the following five errors occur, the bus master logic will permanently terminate the transaction, flag an error, and cease to process any more commands.

1. Target Abort—The PCI transaction will terminate if the addressed target signals a target abort. In this case, the RTA and MERROR bits in the PCI Configuration Register space will be set, and the PCI_BUS_STATUS[4] bit in the SYS_STAT Register will be set.
2. Master Abort—If the addressed target does not respond with an HDEVSEL* assertion, a master abort is flagged. In this case, the RMA and MERROR bits in the PCI Configuration Register space will be set, and the PCI_BUS_STATUS[3] bit in the SYS_STAT Register will be set.
3. Parity Error—If the data parity checked during read transfers is inconsistent with the state of the HPAR signal, a parity error is signaled. In this case, the DPR and MERROR bits in the PCI Configuration Register space will be set, and the PCI_BUS_STATUS[2] bit in the SYS_STAT Register will be set.

4 Interface Disabled—If the driver or application software on the PCI host CPU has disabled the Bt8233 PCI bus master logic (using the M_EN bit in the Command field of the PCI bus configuration registers), any attempt to perform a DMA transaction to the PCI bus will result in an error. In this case, the MERROR and INTF_DIS bits in the PCI configuration space will be set, and the PCI_BUS_STATUS[1] bit in the SYS_STAT Register will be set.

5 Internal Failure—Upon a synchronization error between the DMA coprocessor and the PCI master logic, an internal failure will be flagged. In this case, the MERROR and INT_FAIL bits in the PCI configuration space will be set, and the PCI_BUS_STATUS[0] bit in the SYS_STAT Register will be set.

The above errors permanently effect system level operation. Because of this, the system should be re-initialized, since full system level recovery is unlikely. The bus protocol errors can be cleared either by a software reset of the associated status flag or flags, i.e., RTA, RMA or DPR, or with a reset of the PCI bus master logic using the HRST* input pin. For example, a master abort error can be cleared by writing a logic 1 to the RMA status bit in the PCI Configuration Register space, causing the status bit to be cleared. Internal failures (attempting to initiate a master transaction with the interface disabled, or loss of synchronization with the DMA controller) can only be reset by applying the global reset, CONFIG0(GLOBAL_RESET), or by asserting the HRST* signal.

Next, the MERROR bit must be cleared. The MERROR bit in the PCI Configuration Register drives the PCI_BUS_ERROR interrupt. To clear this interrupt, a logic high must be written to the MERROR bit location. The MERROR bit can also be cleared by a logic low on the HRST* input pin.

The local processor can clear the error bits by setting CONFIG0 (PCI_ERR_RESET) to a logic high. After the errors have been cleared, the SAR should be re-initialized.

Several fields are provided in the PCI configuration space to aid in recovering from a PCI master error. The PCI host software can determine that an error occurred by checking the MERROR bit. It also can determine if the transaction was a read or write by inspecting the MRD bit and the read or write address at which an error occurred by reading the CUR_MSTR_RD_ADDR or CUR_MSTR_WR_ADDR fields.

The PCI Read and PCI Read MULTIPLE commands issued by the PCI block are under the control of the PCI_READ_MULTI bit in the CONFIG0 register.

11.4 Burst FIFO Buffers

Two small FIFO buffers are implemented to do the following: support PCI burst-mode operation, allow synchronization between the Bt8233 internal logic and the PCI bus interface, and carry commands from the DMA coprocessor to the PCI bus logic. The incoming FIFO is 512 x 32 bits, the outgoing FIFO is 16 x 36 bits.

11.5 PCI Bus Slave Logic

The PCI slave logic permits the host CPU on the PCI bus to access and modify Bt8233 resources (the external SRC shared memory, internal memory, and internal registers). Since the control processor also has access to these resources, the PCI slave logic must arbitrate for access prior to performing any read or write transaction. The slave logic also contains the PCI configuration registers, which control the PCI slave and master interfaces and may be read or written at any time by the PCI host. The slave logic implements the synchronizers required for rate-matching between the PCI bus clock and the internal Bt8233 system clock. Also, small FIFOs are used to speed up burst reads and writes performed by the host processor to local resources. This is accomplished by buffering perfected read data and absorbing latency during consecutive writes.

In general, the PCI slave interface functions as a normal memory-mapped PCI target. It responds to Memory Read, Memory Write, Configuration Read, and Configuration Write commands from any initiator on the PCI bus. The slave interface will only respond to Memory Read and Memory Write commands if the MS_EN bit of the Command field in the PCI Configuration Register has been set.

The PCI slave logic does not implement special cycle commands or respond to special cycles on the PCI bus. If a master performs a special cycle on the PCI bus, the following occurs:

- The slave logic never asserts HDEVSEL*.
- Parity errors during the address phase of the special cycle command will be reported to be asserting HSERR* in the normal fashion, if SE_EN and PE_EN in the command register are both set.
- Parity errors during the data phase are ignored.

11.6 PCI Host Address Map

The address map of the Bt8233 resources seen by the PCI bus are the same as those seen by the host processor. The base address of the Bt8233 resource mapping is defined in the BT8233_MEM_BASE field located in the PCI configuration space.

Burst reads of the CSR registers will only return valid data for the first address. Subsequent data words read during a burst read will be indeterminate.

11.7 PCI Configuration Space

In accordance with the PCI bus specification Revision 2.0, the Bt8233 PCI bus interface implements a 128-byte configuration register space. These configuration registers can be used by the host processor to initialize, control, and monitor the SAR bus interface logic. The complete definitions of these registers and the relevant fields within them are given in the PCI bus specification, and will not be repeated here. The descriptions and definitions of these register fields as implemented in the Bt8233 are shown in Chapter 12, Registers.

12.0 ATM Physical Interface

12.1 Overview of ATM PHY Interface

The ATM physical interface contains receive and transmit framer interface logic and receive error generation logic. The ATM physical interface block also interfaces with the segmentation and reassembly coprocessors.

The ATM physical interface for the Bt8233 accepts 52 octet cells from the segmentation coprocessor and transmits them to the PHY device while inserting a dummy HEC. The interface also receives 53 octet cells from the PHY Device, removes the HEC, and saves them in a FIFO to be used by the reassembly coprocessor.

The ATM physical interface is responsible for communicating with and controlling the ATM link interface chip, which carries out all the transmission convergence and physical media dependent functions defined by the ATM protocol. The block performs the following functions:

- Receives and transmits ATM physical interface logic. The ATM physical interface accommodates the Rockwell Bt8222 framer device, a UTOPIA-compatible framer, or a Rockwell-conceived slave UTOPIA interface. The ATM physical interface is responsible for converting between these devices and the internal data interfaces. The Slave UTOPIA interface connects the Bt8233 to a cell switched backplane.
- Receives cell synchronization logic, which validates cell boundaries in the incoming byte stream, strips off the HEC byte from the ATM header, and formats the remaining 52 bytes into thirteen 32-bit words before passing them to the incoming cell FIFO. The receive cell synchronization logic ensures that only complete cells are passed down to the remainder of the reassembly controller.
- Transmits cell synchronization logic, which converts the 32-bit data read from the transmit cell FIFO into an octet stream, generates appropriate cell delineation pulses for use by the transmit ATM physical interface, and inserts the blank HEC byte into the ATM header of each cell before to transferring it to the framer interface.
- Generates and checks odd parity on the octet transmit and receive data buses.

12.2 ATM Physical Interface Logic

The Bt8233 ATM physical interface logic consists of the I/O drivers required to communicate with the external framer device, together with adaptation logic required to convert between either the UTOPIA, Bt8222, or Slave UTOPIA interface protocol and the internal byte streams. Configuration pins (FRCFG[1,0]) determine whether the UTOPIA, Bt8222 interface, or slave UTOPIA protocol will be used.

12.3 ATM Physical I/O Pins

The desired operational mode is indicated to the Bt8233 by appropriately driving the FRCFG[1,0] input according to Table 12-1.

Table 12-1. ATM Physical Interface Mode Select

FRCFG[1,0]	ATM Physical Interface Mode
0 0	Bt8222
0 1	UTOPIA
1 0	Slave UTOPIA
1 1	Reserved–Do Not Use

The interpretation of the ATM physical interface pins on the Bt8233 and the actual signals generated or received by the framer in Bt8222 mode is shown in the Table 12-2. The Bt8222 framer needs to be configured to supply a Start of Cell indication. This is done by setting CELL_VAL[15] bit to a logic high in the Bt8222 device.

Table 12-2. Bt8233 Interface Signals in Bt8222 Mode *(1 of 2)*

Bt8233 Signal	Bt8222 Signal	Active Polarity	Bt8233 Direction
TXD[7:0]	FDAT_IN[7:0]	—	Out
TXPAR	FDAT_IN[8]	—	Out
TXMARK	FCTRL_OUT[16]	High	In
TXFLAG*	FCTRL_IN[0]	Low	Out
TXEN*	FCTRL_OUT[12]	Low	In
RXD[7:0]	FDAT_OUT[7:0]	—	In
RXPAR	FDAT_OUT[8]	—	In

Table 12-2. Bt8233 Interface Signals in Bt8222 Mode *(2 of 2)*

Bt8233 Signal	Bt8222 Signal	Active Polarity	Bt8233 Direction
RXMARK	FCTRL_OUT[10]	High	In
RXFLAG*	FCTRL_IN[4]	Low	Out
RXEN*	FCTRL_OUT[0]	Low	In
FRCTRL	FCTRL_OUT[4]	High	In

The interpretation of the ATM physical interface pins on the Bt8233 and the actual signals generated or received by the framer in UTOPIA mode are shown in Table 12-3. Note that both the TxClk and RxClk signals of the UTOPIA interface may be derived from the CLKD3 output of the Bt8233 (which must also be connected to the FRCTRL input).

Table 12-3. UTOPIA Mode Signals

Bt8233 Signal	PHY Signal	Active Polarity	Bt8233 Direction
TXD[7:0]	TxData[7:0]	—	Out
TXPAR	TxPrty	—	Out
TXMARK	TxSOC	High	Out
TXEN*	TxEnb*	Low	Out
TXFLAG*	TxFull*	Low	In
RXD[7:0]	RxData[7:0]	—	In
RXPAR	RxPrty	—	In
RXMARK	RxSOC	High	In
RXEN*	RxEnb*	Low	Out
RXFLAG*	RxEmpty*	Low	In
FRCTRL	RxClk/TxClk	Rising Edge	In

The interpretation of the ATM physical interface pins on the Bt8233 and the actual signals generated or received by the framer in slave UTOPIA mode are shown in Table 12-4.

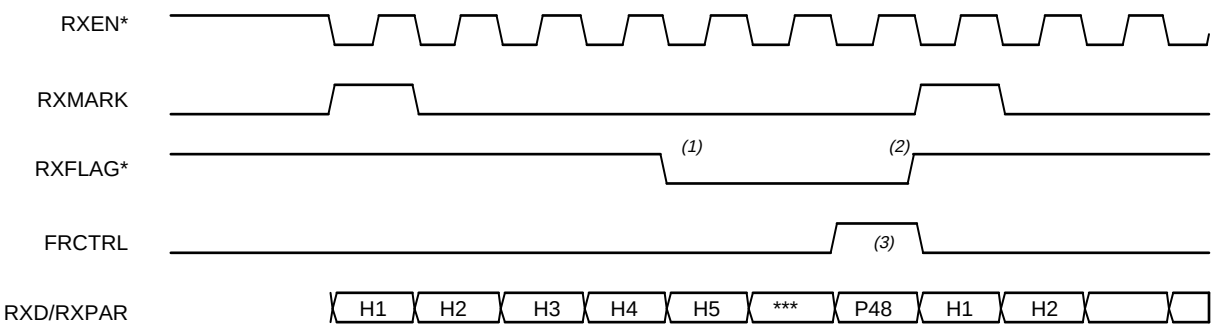
Table 12-4. Slave UTOPIA Mode Interface Signals

Bt8233 Signal	PHY Signal	Active Polarity	Bt8233 Direction
TXD[7:0]	TxData[7:0]	—	Out
TXPAR	TxPrty	—	Out
TXMARK	TxSOC	High	Out
TXEN*	TxEnb*	Low	In
TXFLAG*	TxEmp*	Low	Out
RXD[7:0]	RxData[7:0]	—	In
RXPAR	RxPrty	—	In
RXMARK	RxSOC	High	In
RXEN*	RxEnb*	Low	In
RXFLAG*	RxFull*	Low	Out
FRCTRL	RxClk/TxClk	Rising Edge	In

12.4 Bt8222 Mode Timing

As illustrated in Figure 12-1, received data is presented on the RXD[7:0], RXPAR, FRCTRL, and RXMARK by the framer and strobed into the Bt8233 using the RXEN* line. The adaptation logic computes the 8-bit odd parity over the RXD[7:0] lines and compares it to RXPAR. If in error, FR_PAR_ERR [bit 24] is set in the HOST_ISTAT0/LP_ISTAT0 registers. No data is discarded upon a parity error unless the RSM_PHALT bit in the RSM_CTRL Register is set to a logic high. If so, the reassembly coprocessor halts upon a parity error. The Bt8233 asserts the RXFLAG* line to indicate that its internal receive FIFO does not have room for another cell, and no more cells can be accepted; this normally results in cell loss. RXFLAG* is deasserted when there is room for a complete cell in the receive FIFO. The RXMARK input delimits the start of a cell. The FRCTRL signal is used by the adaptation logic to signal to the receive cell synchronizer that the current cell is errored or invalid and should be discarded. Even if the cell is invalid, a complete cell must be transferred or a synchronization error will occur. This signal is sampled on the last octet of each cell. The FR_RMODE bit in the CONFIG0 Register must be set to a logic high in this mode.

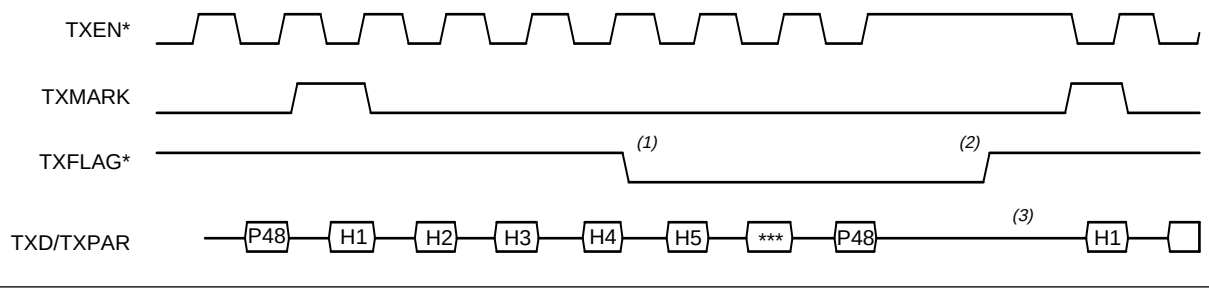
Figure 12-1. Receive Timing in Bt8222 Mode



Notes: (1). Notes: (1). RXFLAG* goes active as soon as there is no room for another cell in the receive FIFO. However, there is still room for the current cell.
(2). RXFLAG* goes inactive when there is room for a complete cell in the receive FIFO. Free running RXEN* is required when RXFLAG* is active.
(3). If FRCTRL is sampled high on the last octet of the cell, the cell is discarded.

Figure 12-2 illustrates how the TXEN* input from the framer is used as a data enable for the TXD[7:0] and TXPAR outputs and the TXMARK input. TXEN* also three-states the TXD and TXPAR outputs when there is a logic high so that multiple ATM SARs can be connected to the Bt8222 device. If another cell is not supplied by the transmit cell synchronization logic, the TXFLAG* output is asserted to indicate that a cell cannot be supplied in the next cell slot. The flag will be asserted after the first 4 bytes of the last cell are transmitted. As before, the TXPAR line carries the 8-bit odd parity computed over TXD[7:0]. The TXMARK line is asserted by the framer to indicate the start of each output cell.

Figure 12-2. Transmit Timing in Bt8222 Mode

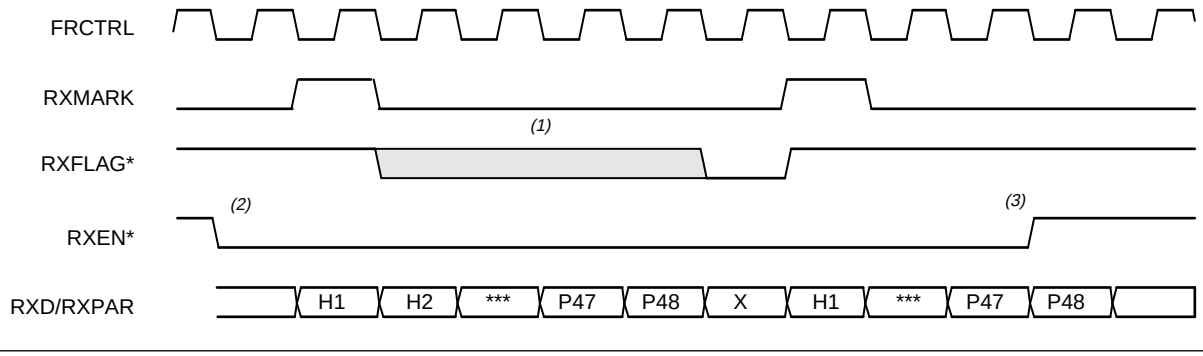


Notes: (1). TXFLAG* goes active when there is no longer a complete cell available in the transmit FIFO. This signal switches asynchronously to TXEN*.
(2). TXFLAG* goes inactive when a complete cell is in the transmit FIFO. This signal switches asynchronously toTXEN*.
(3). TXD and TXPAR are three-stated when TXEN* is a logic high.

12.5 UTOPIA Mode Cell Handshake Timing

If high, The UTOPIA_MODE bit in the CONFIG0 register selects cell-level handshaking. Received data is latched from the RXD[7:0] and RXPAR lines on the rising edge of FRCTRL after RXEN* is sampled active (see Figure 12-3). The 8-bit odd parity computed over the RXD[7:0] lines is compared to the RXPAR input. If in error, the FR_PAR_ERR bit is set in the HOST_ISTAT0/LP_ISTAT0 registers. No data is discarded upon a parity error unless the RSM_PHALT bit in the RSM_CTRL register is set to a logic high. If so, the reassembly coprocessor halts upon a parity error. The RXMARK signals to the Bt8233 the start of cell. The RXFLAG* input is the physical layer FIFO empty signal. When it is active, a complete cell is not present in the physical receive FIFO. The physical layer device sets RXFLAG* inactive when it has a complete cell to transfer. The Bt8233 sets RXEN* to a logic low if it can accept a complete cell. On the clock cycle after the last octet of a cell is transferred, the Bt8233 samples the RXFLAG* input. If low, the physical device does not have a cell to transfer. If RXFLAG* is high, the physical device has another cell to transfer, and the Bt8233 will immediately start receiving the next cell if it can accept a complete cell. The FR_RMODE bit in the CONFIG0 register should be set to a logic low in this mode.

Figure 12-3. Receive Timing in UTOPIA Mode with Cell Handshake

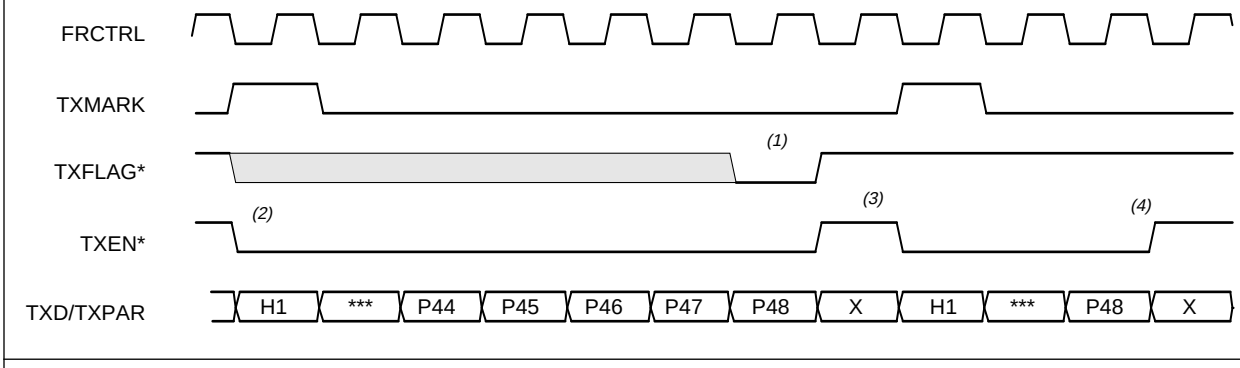


Notes: (1). Once a cell transfer is started, RXFLAG* is not sampled until the end of the cell.
(2). RXEN* goes active only if there is room in the receive FIFO for a complete cell.
(3). RXEN* goes inactive at cell boundaries if the receive FIFO cannot accept another cell.

Transmit data is driven on TXD[7:0] on the rising edge of FRCTRL when TXEN* is asserted. TXEN* is only asserted when there is data in the Bt8233 transmit FIFO. Simultaneously, the 8-bit odd parity computed over the TXD[7:0] lines is driven on to the TXPAR output. The TXMARK line is driven by the framer device to indicate start of cell. If the TXFLAG* input is asserted by the framer device, the framer device is full, and another cell is not transmitted to the physical framer. (See Figure 12-4.)

In UTOPIA mode, the FRCTRL input can be connected to the Bt8233 CLKD3 output, a 50% duty cycle clock derived by dividing CLK2X by 3.

Figure 12-4. Transmit Timing in UTOPIA Mode with Cell Handshake

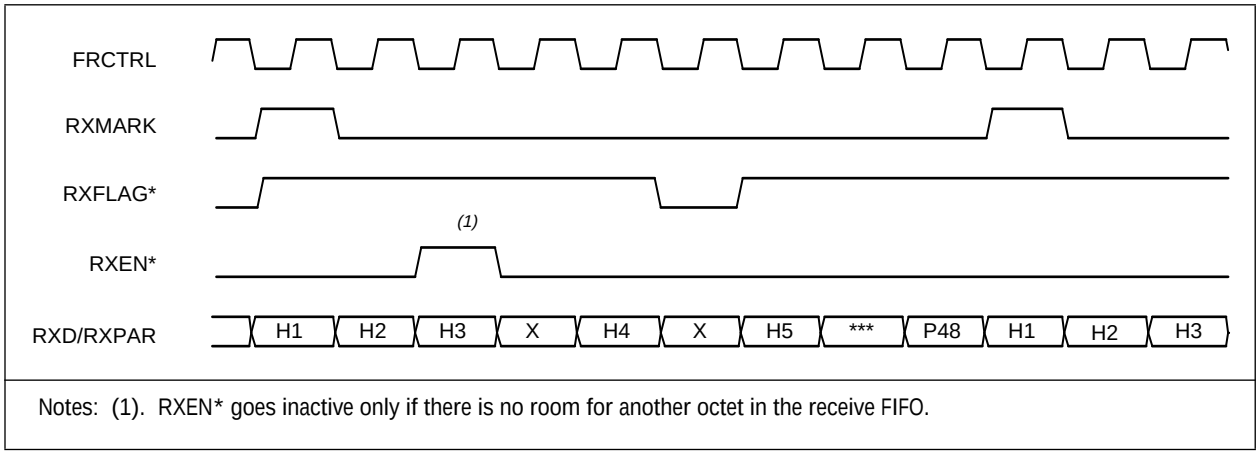


Notes: (1). Once transfer of a cell is started, TXFLAG* is sampled only on the last octet of a cell.
(2). TXEN* goes active if TXFLAG* is inactive at previous rising clock edge and a complete cell is in the transmit FIFO.
(3). TXEN* goes inactive due to TXFLAG* being active on previous cycle.
(4). TXEN* goes inactive since a complete cell is not in the transmit FIFO.

12.6 UTOPIA Mode Octet Handshake Timing

If low, the UTOPIA_MODE bit in the CONFIG0 register selects octet-level handshaking. Received data is latched from the RXD[7:0] and RXPAR lines on the rising edge of FRCTRL after RXEN* is sampled active (see Figure 12-3). The 8-bit odd parity computed over the RXD[7:0] lines is compared to the RXPAR input. If in error, FR_PAR_ERR in the HOST_ISTAT0/LP_ISTAT0 registers is set. No data is discarded upon a parity error unless the RSM_PHALT bit in the RSM_CTRL register is set to a logic high. If so, the reassembly coprocessor halts upon a parity error. The RXMARK signals the start of cell to the Bt8233. The RXFLAG* input is the physical layer FIFO empty signal. When it is active, no data is present in the physical receive FIFO. The physical layer device sets RXFLAG* inactive when it has an octet to transfer. The Bt8233 sets RXEN* to a logic low if it can accept an octet in the next clock cycle. The FR_RMODE bit in the CONFIG0 register should be set to a logic low in this mode.

Figure 12-5. Receive Timing in UTOPIA Mode with Octet Handshake

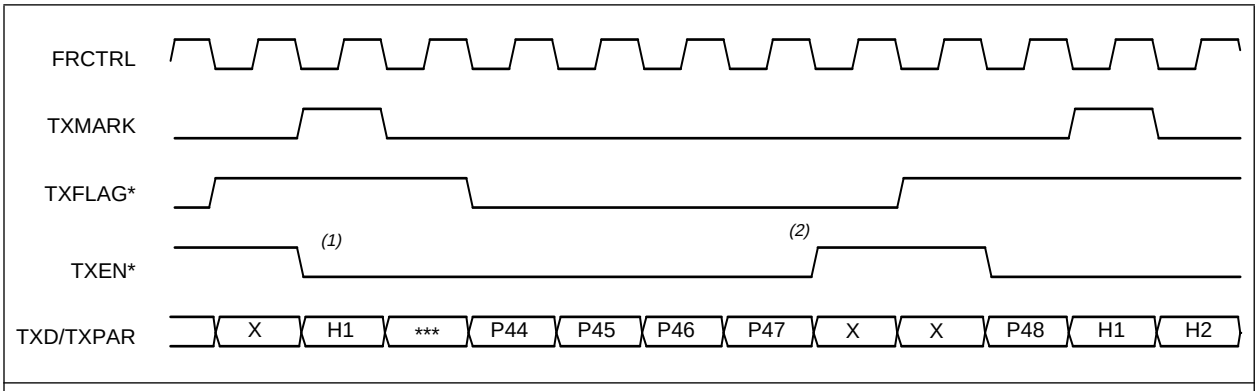


Notes: (1). RXEN* goes inactive only if there is no room for another octet in the receive FIFO.

Transmit data is driven on TXD[7:0] on the rising edge of FRCTRL when TXEN* is asserted. TXEN* is only asserted when there is data in the Bt8233 transmit FIFO. Simultaneously, the 8-bit odd parity computed over the TXD[7:0] lines is driven on to the TXPAR output. The TXMARK line is driven by the framer device to indicate start of cell. If the TXFLAG* input is asserted by the framer device, the framer device is full, and can accept only one to four more octets. (See Figure 12-4).

In UTOPIA mode, the FRCTRL input can be connected to the Bt8233 CLKD3 output, which is a 50% duty cycle clock derived by dividing the CLK2X input by 3.

Figure 12-6. Transmit Timing in UTOPIA Mode with Octet Handshake



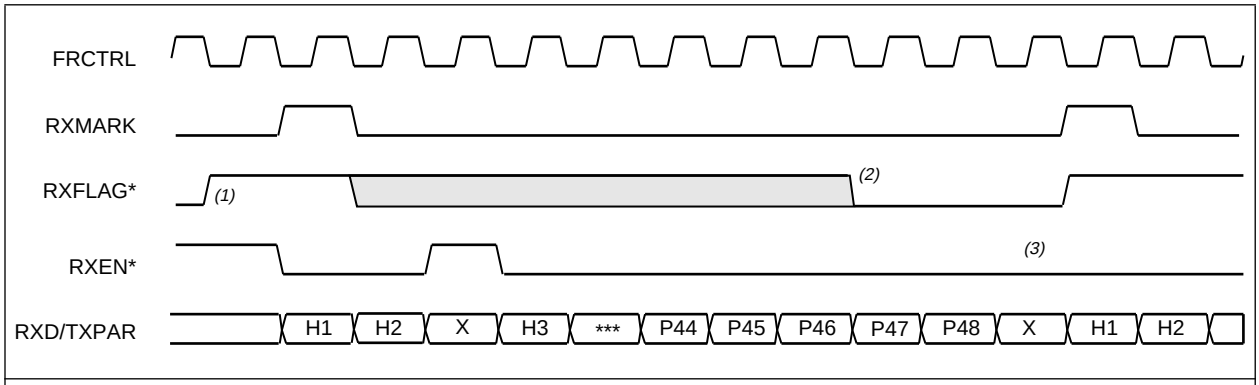
Notes: (1). TXEN* goes active if TXFLAG* is inactive at previous rising clock edge and a complete cell is in the transmit FIFO.
(2). TXEN* goes inactive one to four clock cycles after TXFLAG* goes active.

12.7 Slave UTOPIA Mode

The slave UTOPIA mode is similar to the UTOPIA mode, except the direction of the enable signals and FIFO flags are reversed. This allows a switch fabric or backplane to directly control the physical port. The transmit and receive enable signals are generated by the physical layer instead of the Bt8233. The TxFull* signal is changed to the TxEmpty* signal and is an output of the Bt8233. The RxEmpty* signal is changed to the RxFull* signal and is also an output of the Bt8233. This mode only supports a cell-level handshake protocol.

Received data is latched from the RXD[7:0] and RXPAR lines on the rising edge of FRCTRL when RXEN* is active (see Figure 12-3). The 8-bit odd parity computed over the RXD[7:0] lines is compared to the RXPAR input. If in error, the FR_PAR_ERR bit is set in the HOST_ISTAT0/LP_ISTAT0 registers. No data is discarded upon a parity error unless the RSM_PHALT bit in the RSM_CTRL register is set to a logic high. If so, the reassembly coprocessor halts upon a parity error. The RXMARK signals to the Bt8233 the start of cell. The RXFLAG* output is the receive FIFO full signal. When it is active, the Bt8233 cannot accept another cell. The Bt8233 sets RXFLAG* inactive when it has room in the receive FIFO for another cell. The physical device sets RXEN* to a logic low if it can transfer an octet. The FR_RMODE bit in the CONFIG0 register should be set to a logic low in this mode.

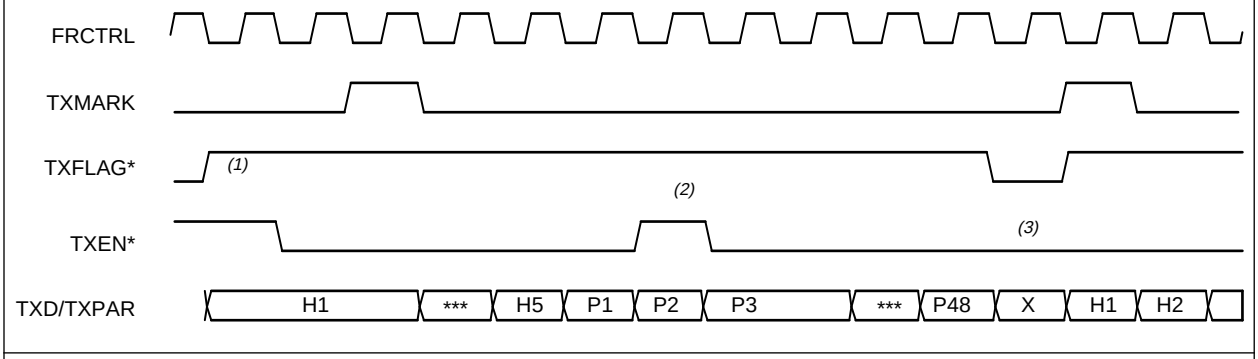
Figure 12-7. Receive Timing in Slave UTOPIA Mode



Notes: (1). RXFLAG* will go inactive when there is room in the receive FIFO for a complete cell.
(2). RXFLAG* will go active when there is no longer room in the receive FIFO for another complete cell.
(3). (RXEN* need not be inactive when RXFLAG* is active.

Transmit data is driven on TXD[7:0] on the rising edge of FRCTRL after TXEN* is sampled asserted (see Figure 12-4). Simultaneously, the 8-bit odd parity computed over the TXD[7:0] lines is driven on to the TXPAR output. The TXMARK line is driven by the framer device to indicate start of cell. If the TXFLAG* output is asserted by the Bt8233, the transmit FIFO does not contain a complete cell.

Figure 12-8. Transmit Timing in Slave UTOPIA Mode



Notes: (1). TXFLAG* goes inactive when a complete cell is in the transmit FIFO.
(2). Example of physical device deactivating TXEN*.
(3). Physical device may keep TXEN* active while TXFLAG* is active.

12.8 Loopback Mode

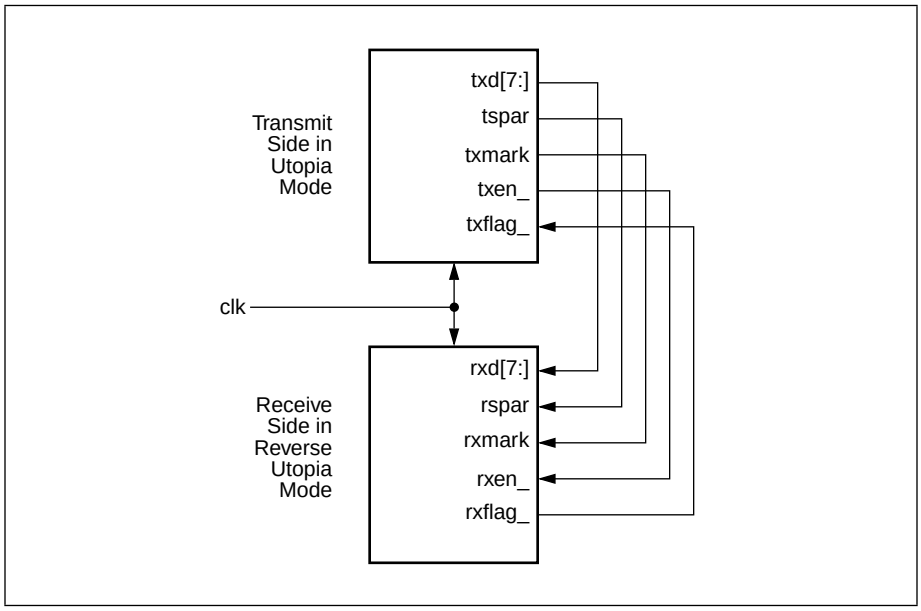
The physical interface can be internally looped by setting the FR_LOOP bit in the CONFIG0 register to a logical high. This mode uses the internal system clock for operation; therefore, a framer clock is not needed during loopback operation.

When the signal “fr_src_loop = 1”, the interface will loop back the data and control signals, so the path from the segmentation coprocessor to the reassembly coprocessor can be tested. The internal connections of the PHY Device interface signals are connected, as Figure 12-9 shows. The Transmit Side is put into the Utopia Mode and the Receive Side is put into the Reverse Utopia Mode.

In this mode the outputs of the chip, txd, txpar, txmark, txen_, and rxflag_, are tri-stated so there are no output conflicts with other devices connected to these signals.

NOTE: A chip reset is required after the changing of the FR_LOOP bit, because this changes the source of the clock to the Physical Interface circuitry.

Figure 12-9. Source Loopback Mode Diagram



12.9 Receive Cell Synchronization Logic

The receive cell synchronization logic accepts a stream of octets (together with error and cell boundary indications) from the receive ATM physical interface and performs the following functions:

- Maintains a sequence counter that marks the various components of an ATM cell: the 5-byte ATM header, the 1-byte HEC field within the header, and the 48-byte payload. The sequence counter is also used by the ATM physical interface to check cell boundary synchronization.
- Extracts and discards the HEC byte from each 53-byte ATM cell, leaving 52 bytes of cell data. In Bt8222 mode, the Bt8222 will check the HEC byte.
- Formats consecutive 4-byte segments into 32-bit words; thus, the header forms the first word, the first 4 bytes of the payload form the next word, and so on. A total of 13 32-bit words are created from each 52-byte cell after the HEC byte has been removed. The bytes within each word are left-justified (big-endian format), i.e., the first byte received is the MSB of the word.
- Ensures that a complete cell (exactly 52 bytes) is always written to the FIFO. If a synchronization error occurs, the FR_SYNC_ERR bit in the HOST_ISTAT0/LP_ISTAT0 registers is set. The ATM physical interface attempts to resynchronize with the data stream.
- Sets the RSM_OVFL bit in the HOST_ISTAT0/LP_ISTAT0 registers if an octet could not be transferred due to the receive FIFO being full.

12.10 Transmit Cell Synchronization Logic

The transmit cell synchronization logic copies cell data from the transmit cell FIFO to the transmit ATM physical interface while performing the following functions:

- Reads 32-bit words from the transmit cell FIFO and converts them to a stream of octets, with the MSB of each 32-bit word corresponding to the first byte derived from that word (big-endian format).
- Maintains a sequence counter that delineates the various components of each ATM cell (4-byte header, 48-byte payload) in the outgoing byte stream.
- Inserts a blank (all-zero) HEC byte, used as a placeholder, into the outgoing byte stream representing each ATM cell. The HEC placeholder is inserted after the first 4 bytes (the ATM header) have been transferred. In Bt8222 mode, the Bt8222 device will calculate and insert the correct HEC value.
- Generates appropriate cell delineation pulses to the transmit ATM physical interface logic. These pulses are used to generate the TXMARK* output and generate to verify synchronization with the framer device.
- If the ATM physical transmit interface runs out of cells to transmit, the SEG_UNFL bit is set in the HOST_ISTAT0/LP_ISTAT0 registers.

The transmit cell synchronization logic supplies a continuous stream of octets to the transmit ATM physical interface unit, with cell delineation pulses at the starting byte of every cell. Only complete 53-byte cells are supplied to the ATM physical interface. If the transmit cell FIFO is empty, the transmit cell synchronization logic indicates that no more data can be transferred to the framer.

13.0 Bt8233 Registers

13.1 CSR Registers

Control and status registers of the Bt8233 are listed in Table 13-1. Detailed register descriptions are provided in this section. Byte enables are ignored by the Bt8233 when writing to control and status registers.

Table 13-1. Bt8233 Control and Status Registers *(1 of 2)*

Address	Name	Type	Description
0x00	CLOCK	R/W	Real Time Clock Register
0x04	ALARM1	R/W L	Alarm Register 1
0x08	Reserved	—	Not Implemented
0x0C	SYS_STAT	R/O	System Status Register
0x10	Reserved	—	Not Implemented
0x14	CONFIG0	R/W	Basic Configuration And Control Register 0
0x18–0x7c	Reserved	—	Not Implemented
0x80	SEG_CTRL	R/W	Segmentation Control Register
0x84	SEG_VBASE	R/W	Seg VCC Table and Schedule Table Base Address Register
0x88	SEG_PMBASE	R/W	Seg PM Table and Bucket Table base address register
0x8c	SEG_TXBASE	R/W	Seg Transmit queue enable and base address register
0x90–0x9c	Reserved	—	Not Implemented
0xa0	SCH_PRI	R/WB	Priority Queue Control
0xa4	Reserved	—	Not Implemented
0xa8	SCH_SIZE	R/W	Schedule size and slot minimum drain rate Register
0xac	Reserved	—	Not Implemented
0xb0	SCH_ABR_MAX	R/W	Maximum ABR VCC_INDEX Register
0xb4	SCH_ABR_CON	R/W	Schedule ABR Constant Register
0xb8	SCH_ABRBASE	R/W	ABR decision table lookup base register.
0xbc	SCH_CNG	R/W	ABR congestion notification by free buffer queue
0xc0–0xeC	Reserved	—	Not Implemented
0xf0	RSM_CTRL0	R/W	Reassembly Control Register 0

Table 13-1. Bt8233 Control and Status Registers *(2 of 2)*

Address	Name	Type	Description
0xf4	RSM_CTRL1	R/W	Reassembly Control Register 1
0xf8	RSM_FQBASE	R/W	Reassembly Free Buffer Queue Base Register
0xfC	RSM_FQCTRL	R/W	Reassembly Free Buffer Queue Control Register
0x100	RSM_TBASE	R/W	Reassembly Table Base Register
0x104	RSM_TO	R/W	Reassembly Time-out Register
0x108	RS_QBASE	R/W	Reassembly/Segmentation Queue Base Address Register
0x10C–0x15C	Reserved	—	Not Implemented
0x160	CELL_XMIT_CNT	R/O	ATM Cells Transmitted Counter
0x164	CELL_RCVD_CNT	R/O	ATM Cells Received Counter
0x168	CELL_DSC_CNT	R/O	ATM Cells Discarded Counter
0x16C	AAL5_DCS_CNT	R/O	AAL5 PDUs Discarded Counter
0x170–0x19C	Reserved	—	Not Implemented
0x1a0	HOST_MBOX	R/W L	Host Mailbox Register
0x1a4	HOST_ST_WR	R/O	Host Status Write Register
0x1a8-0x1ac	Reserved	—	Not Implemented
0x1b0	LP_MBOX	R/W H	Local Processor Mailbox Register
0x1b4-0x1bc	Reserved	—	Not Implemented
0x1c0	HOST_ISTAT0	R/O	Host Processor Interrupt Status Register 0
0x1c4	HOST_ISTAT1	R/O	Host Processor Interrupt Status Register 1
0x1c8-0x1cc	Reserved	—	Not Implemented
0x1d0	HOST_IMASK0	R/W H	Host Interrupt Mask Register 0
0x1d4	HOST_IMASK1	R/W H	Host Interrupt Mask Register 1
0x1d8–0x1dc	Reserved	—	Not Implemented
0x1e0	LP_ISTAT0	R/O	Local Processor Interrupt Status Register 0
0x1e4	LP_ISTAT1	R/O	Local Processor Interrupt Status Register 1
0x1e8–0x1ec	Reserved	—	Not Implemented
0x1f0	LP_IMASK0	R/W L	Local Processor Interrupt Mask Register 0
0x1f4	LP_IMASK1	R/W L	Local Processor Interrupt Mask Register 1
0x1f8–0x1fc	Reserved	—	Not Implemented

13.1.1 Register Terminology

The access type terminology given below applies to all registers in this section. Each register description is prefaced with the appropriate abbreviated access types.

Abbreviation	Description
R/W	Read and write access for both host and local processors.
R/W H	Read and write access for host, read only access for local processor.
R/W L	Read and write access for local, read only access for host processor.
R/W R	Read and write access for both processors with restrictions.
R/O–W/O B	Part of this register is read only, part write only by both processors.
R/O	Read only access for both processors.

13.2 System Registers

13.2.1 Real-Time Clock Register (CLOCK)

Access types: R/W
This register contains the 32-bit real time clock. It is incremented by the system clock (SYSCLK) which has been divided by the DIVIDER[6:0] field in the CONFIG0 register. It may be written to any value by each processor, and may generate an interrupt on the RTC_OVFL status bit in the LP_ISTAT0 register when it overflows from 0xFFFFFFFF to 0x0.

Bit	Field Size	Name	Description
31–0	32	CLOCK[31:0]	Real-time clock value.

13.2.2 Alarm Register 1 (ALARM1)

Access types: R/W L
This register contains a 32-bit value which is compared against the CLOCK register. When the two registers are equal, the ALARM1 status bit in the LP_ISTAT0 register is latched and can be enabled to cause an interrupt to the local processor. To implement a periodic interrupt, add a constant value to this register after each interrupt.

Bit	Field Size	Name	Description
31–0	32	ALARM1[31:0]	ALARM1 comparison value.

13.2.3 System Status Register (SYS_STAT)

Access types: R/O

The System Status Register provides read-only system status. This register reflects the device ID and version information for the part, as well as pin programmable options that otherwise may not be visible to the processors. It also contains expanded information for the status located in the HOST_ISTAT0 and LP_ISTAT0 registers.

Bit	Field Size	Name	Description
31–17	15	Reserved	Not implemented at this time.
16–12	5	PCI_BUS_STATUS[4:0]	The status bits are as follows: 4: Target Abort 3: Master Abort 2: Parity Error 1: Interface Disabled 0: Internal Failure Reflects corresponding error bits in the PCI Configuration register. Bits are reset by either a write to the PCI Configuration register by the host, or by setting CONFIG0 (PCI_ERR_RESET) bit.
11	1	RAMMODE	Reflects the state of the RAMMODE input pin.
10	1	PROCMODE	Reflects the state of the PROCMODE input pin.
9, 8	2	FRCFG[1,0]	Reflects the state of the FRCFG[1,0] input pins.
7–4	4	VERSION [3:0]	Version number for the Bt8233.
3–0	4	DEVICE[3:0]	Device ID for the Bt8233, set to1.

13.2.4 Configuration Register 0 (CONFIG0)

Access types: R/W

This register provides all of the control and configuration bits that are not associated with the reassembly and segmentation coprocessors. The majority of these bits are configuration (which occurs at initialization time) and are not changed dynamically. The assertion of the HRST* system reset pin will clear all of the bits in the CONFIG0 register except for MEMCTRL, which will be set high.

Bit	Field Size	Name	Description
31	1	LP_ENABLE	When set, this bit causes the PRST* output pin to be high. This may be used to reset the local processor.
30	1	GLOBAL_RESET	When set, this bit causes reset of the segmentation and reassembly coprocessors as well as all latched status.
29	1	PCI_MSTR_RESET	When set, this bit resets the PCI master logic. Once active, this bit must stay active for 16 cycles of the HCLK input signal.
28	1	PCI_ERR_RESET	When set, resets all PCI error bits in the PCI configuration, including RMA, RTA, DPR, INTF_DIS, INT_FAIL and MERROR. This also re-enables PCI master operation.
27–26	2	Reserved	Always set to zero.
25	1	PHY2_EN	Enable the Second Phy device memory space in standalone operation.
24	1	INT_LBANK	When set, allows only byte 0 and 1 writes to address space 0x1000 - 0x10ff and 0x1400 - 0x14ff. This allows endian neutral access of the Status Queue Base Table READ_UD field by the host or local processor.
23	1	Reserved	Always set to zero.
22	1	PCI_READ_MULTI	When this bit is set, the SAR's PCI Master implements the PCI Read Multiple Command. Otherwise, the PCI Master implements the PCI Read Command.
21	1	PCI_ARB	Selects PCI master arbitration scheme. When a logic high, enables round-robin between read and write requests. When a logic low, reads have priority over writes.
20–16	5	STATMODE[4:0]	Selects which internal status to output on the STAT[1,0] output pins.
15	1	FR_RMODE	Controls reassembly start of cell processing. When set low, processing starts after the first two words of a cell are received. When set high, a complete cell must be in reassembly FIFO before cell is processed. n 8222 mode, this bit must set high.
14	1	FR_LOOP	When set, this bit enables loopback of cells at the ATM physical interface.

Bit	Field Size	Name	Description
13	1	UTOPIA_MODE	Selects byte or cell UTOPIA handshake mode. 0 = Octet handshake 1 = Cell handshake
12	1	ENDIAN	Selects between Little and Big Endian host data structures. 0 = Little Endian 1 = Big Endian
11	1	LP_BWAIT	Selects zero or one wait states between consecutive data cycles during local processor burst accesses. Set to logic low for standalone operation mode.
10	1	MEMCTRL	Selects zero or one wait states SRC shared memory (1 or 2 cycle).
9–7	3	BANKSIZE[2:0]	Selects size of memory banks for contiguous memory support. See Section 9.2, Memory Bank Characteristics, for further explanation.
6–0	7	DIVIDER[6:0]	Prescaler for SYSCLK which advances the counter in the CLOCK Register. SYSCLK is divided by the divider value; if zero, divided by 128.

13.3 Segmentation Registers

13.3.1 Segmentation Control Register (SEG_CTRL)

Access types: R/W
This register contains general control bits for the segmentation coprocessor. The assertion of the HRST* system reset pin or GLOBAL_RESET bit in the CONFIG0 register will cause the clearing of the SEG_ENABLE control bit.

Bit	Field Size	Name	Description
31	1	SEG_ENABLE	Segmentation Enable—Enables segmentation coprocessor. If disabled, the segmentation coprocessor will halt on a cell boundary.
30	1	SEG_RESET	Segmentation Reset —resets segmentation coprocessor and pointers.
29–27	3	VBR_OFFSET	Offset from schedule slot priority to general priority. (VBR_OFFSET + (# VBR/ABR priorities) $\leq$ 7).
26	1	SEG_GFC	Enable segmentation GFC processing. The segmentation machine is disabled when the SAR receives cells with GFC halt set. GFC priority queues (set in the SCH_PRI register) are active for one cell for each received cell with GFC SET_A bit = 1.
25	1	DBL_SLOT	Each schedule slot occupies two words.
24	1	CBR_TUN	Use first entry in each schedule slot for CBR/tunnel traffic.
23–16	8	Reserved	Program and read as 0.
15–12	4	TX_FIFO_LEN	Depth of transmit FIFO in cells. Valid range is 1-9. To ensure optimum performance, the depth of the FIFO should be at least 3.
11	1	CLP0_EOM	Set CLP in ATM header to 0 on last cell of CPCS-PDU.

Bit	Field Size	Name	Description
10–6	5	OAM_STAT_ID	Status queue ID for buffer descriptors with OAM_STAT set.
5	1	SEG_ST_HALT	Enables a status queue entry for a VCC halted with a partially segmented buffer.
4	1	SEG_LS_DIS	Segmentation Local Status Disable—Disable Segmentation check for SAR shared memory status queue full condition. If this bit is not set, the segmentation coprocessor will not segment any cells for a VCC assigned to a full SAR shared memory status queue. This bit may be used to disable overflow checking when the queues are sized large enough to prevent overflow.
3	1	SEG_HS_DIS	Segmentation Host Status Disable—Disable Segmentation check for PCI memory status queue full condition. If this bit is not set, the segmentation coprocessor will not segment any cells for a VCC assigned to a full PCI memory status queue. This bit may be used to disable overflow checking when the queues are sized large enough to prevent overflow.
2	1	TX_RND	Set for Transmit queue servicing is round robin. Clear for transmit queue servicing in priority order (31 is highest priority).
1–0	2	TR_SIZE[1:0]	Number of entries in each transmit queue 00 64 01 256 10 1024 11 4096

13.3.2 Segmentation Virtual Channel Connection Base Address Register (SEG_VBASE)

Access types: R/W

The SEG_VBASE register sets the base address in SRC shared memory for the segmentation VCC table and the schedule table. Both base addresses are 128-byte aligned and only the 16 most significant bits of the address are specified in the SEG_VBASE register.

Bit	Field Size	Name	Description
31–16	16	SEG_SCHB[15:0]	Base address for the schedule table.
15–0	16	SEG_VCCB[15:0]	Base address for the segmentation VCC table.

13.3.3 Segmentation PM Base Register (SEG_PMBASE)

Access types: R/W
The SEG_PMBASE register sets the base address in SRC shared memory for the VBR Bucket table and the performance monitoring table. Both base addresses are 128-byte aligned and only the 16 most significant bits of the address are specified in the SEG_PMBASE register.

Bit	Field Size	Name	Description
31–16	16	SEG_BCKB[15:0]	Base address for the VBR Bucket table.
15–0	16	SEG_PMB[15:0]	Base address for the performance monitoring table.

13.3.4 Segmentation Transmit Queue Base Register (SEG_TXBASE)

Access types: R/W
The SEG_TXBASE register sets the base address in SRC shared memory for the transmit queues and enables the individual queues. The base address is 128-byte aligned and only the 16 most significant bits of the address are specified in the SEG_TXBASE register.

Bit	Field Size	Name	Description
31–16	16	SEG_TXB[15:0]	Base address for the transmit queues.
15–13	3	Reserved	Program and read as 0.
12–5	8	XMIT_INTERVAL[7:0]	Interval for transmit queue READ_UD_PNTR update.
4–0	5	TX_EN	Transmit queues 0-TX_EN are enabled.

13.4 Scheduler Registers

13.4.1 Schedule Priority Register (SCH_PRI)

Access types: R/W
This register specifies the use of each global priority pointer.

Bit	Field Size	Name	Description
31–24	8	Reserved	Program and read as 0.
23	1	Reserved	Program and read as 0.
22	1	TUN_ENA7	Enable tunnel on global priority pointer 7.
21	1	GFC7	Enable GFC on priority pointer 7.
20	1	Reserved	Program and read as 0.
19	1	TUN_ENA6	Enable tunnel on global priority pointer 6.
18	1	GFC6	Enable GFC on priority pointer 6.
17	1	Reserved	Program and read as 0.
16	1	TUN_ENA5	Enable tunnel on global priority pointer 5.
15	1	GFC5	Enable GFC on priority pointer 5.
14	1	Reserved	Program and read as 0.
13	1	TUN_ENA4	Enable tunnel on global priority pointer 4.
12	1	GFC4	Enable GFC on priority pointer 4.
11	1	Reserved	Program and read as 0.
10	1	TUN_ENA3	Enable tunnel on global priority pointer 3.
9	1	GFC3	Enable GFC on priority pointer 3.
8	1	Reserved	Program and read as 0.
7	1	TUN_ENA2	Enable tunnel on global priority pointer 2.
6	1	GFC2	Enable GFC on priority pointer 2.
5	1	Reserved	Program and read as 0.
4	1	TUN_ENA1	Enable tunnel on global priority pointer 1.
3	1	GFC1	Enable GFC on priority pointer 1.
2	1	Reserved	Program and read as 0.
1	1	TUN_ENA0	Enable tunnel on global priority pointer 0.
0	1	GFC0	Enable GFC on priority pointer 0.

13.4.2 Schedule Size Register (SCH_SIZE)

Access types: R/W
The SCH_SIZE register sets the size of the schedule table in schedule slots and the period of a schedule slot in system clocks.

Bit	Field Size	Name	Description
31–16	16	TBL_SIZE[15:0]	Size of schedule table in schedule slots.
15–14	2	Reserved	Program and read as 0.
13–0	14	SLOT_PER[13:0]	Number of system clocks per schedule slot. The value written to the register should be SLOT_PER - 1. Minimum bound for SLOT_PER value = 70.

13.4.3 Schedule Maximum ABR Register (SCH_ABR_MAX)

Access types: R/W
The SCH_ABR_MAX register defines the configured maximum ABR VCC index.

Bit	Field Size	Name	Description
31–16	16	Reserved	Program and read as 0.
15–0	16	VCC_MAX	Maximum ABR VCC Index.

13.4.4 Schedule ABR Constant Register (SCJ_ABR_CON)

Access types: R/W
The SCH_ABR_CON register sets the ABR TRM and ADTF timeout. Time-out values are in units of 64 cell slots.

Bit	Field Size	Name	Description
31–16	16	ER_TRM	ER TRM parameter. Parameter value is sysclk period* SLOT_PER *ER_TRM *64.
15–0	16	ER_ADTF	ER ADTF parameter. Parameter value is sysclk period *SLOT_PER* ER_ADTF * 64.

13.4.5 ABR Decision Table Lookup Base Register (SCH_ABR_BASE)

Access types: R/W
The SCH_ABR_BASE register sets the base address in SRC shared memory for the ABR decision table. This address is 128-byte aligned, and only the 16 most significant bits of the address are specified in the SCH_ABR_BASE register.

Bit	Field Size	Name	Description
31–29	3	Reserved	Program and read as 0.
28	1	OOR_ENA	Enable ER out of rate forwared RM cell generation.
27–16	12	PPR_INT	Set OOR_INT = ((PER_VCC rate for out of rate cells) / ((SCH_SIZE(SLOT_PER) +1) *sysdk period * (SCH_ABR_MAX (VCC_MAX) /2))) -1.
15–0	16	SCH_ABRB[15:0]	Base address for the ABR decision table.

13.4.6 ABR Congestion Register (SCH_CNG)

Access types: R/W
The SCH_CNG register sets each reassembly free buffer queue to a congested or non-congested state for transmitted reverse RM cells.

Bit	Field Size	Name	Description
31–0	32	FBQ_CNG[31:0]	Congestion state for each free buffer queue.

13.5 Reassembly Registers

13.5.1 Reassembly Control Register 0 (RSM_CTRL0)

Access types: R/W

The reassembly control register 0, contains the general control bits for the reassembly coprocessor. The assertion of the HRST* system reset pin or the GLOBAL_RESET bit in the CONFIG0 register will clear the RSM_ENABLE control bit.

Bit	Field Size	Name	Description
31	1	RSM_ENABLE	Reassembly enable. Initiates an incoming transfer if set, and halts it if reset. If this bit is reset while the reassembly coprocessor is running, it temporarily suspends the activities of the reassembly coprocessor logic. Suspension takes place on a cell boundary, i.e., between the completion of all processing and transfers required for the current cell and the start of processing for the next cell. The hold can be removed, and the transfer resumed by setting the RSM_ENABLE bit. This bit will also be set low internally on certain reassembly error conditions. These include parity error with PHALT_EN. In this case, the error condition should be corrected and the RSM_ENABLE bit set high to resume operation.
30	1	RSM_RESET	Reassembly reset. Forces a hardware reset of the reassembly coprocessor when asserted. It must be deasserted before the reassembly coprocessor will resume normal operation.
29	1	Reserved	Not implemented at this time.
28	1	VPI_MASK	VPI Mask enable. Used to select UNI/NNI header operation in the direct index method. When a logic high, the 4 MSBs of the header are masked for UNI operation. This also controls the Index Table size. A UNI table is 256 entries and a NNI table is 4096.
27–24	4	Reserved	Not implemented at this time.
23–18	6	Reserved	Not implemented at this time.
17	1	RSM_PHALT	Reassembly coprocessor halt on parity error detect. The reassembly coprocessor will halt the incoming channel logic if a parity error is detected and the RSM_PHALT bit is set.
16	1	Reserved	Not implemented at this time.

Bit	Field Size	Name	Description
15	1	FWALL_EN	Firewall enable. Enables free buffer queue firewalling of user cells. If set, this bit enables the per connection free buffer queue firewall. Each connection that firewall is active in must have the FW_EN bit set to a logic high in the VCC table.
14	1	RSM_FBQ_DIS	Free Buffer Queue Underflow Protection Disable. When a logic high, the reassembly coprocessor ignores the VLD bit in the free buffer queue when a new buffer is required. The periodic writing of the read index pointer to host/SRC shared memory is also disabled.
13	1	RSM_STAT_DIS	Status Queue Overflow Protection Disable. When a logic high, the reassembly coprocessor ignores the READ_UD pointer.
12	1	GTO_EN	Global Time-out Enable. When a logic high, the automatic reassembly time-out function is enabled.
11–5	7	MAX_LEN	Maximum Length. MAX_LEN * 1024 is the maximum allowable size in bytes of an AAL5 CPCS-PDU including overhead.
4–0	5	GDIS_PRI	Global Discard Priority. Used by the Frame Relay and CLP discard functions. If the individual channel priority number is less than or equal to GDIS_PRI, PDUs on that channel may be discarded.

13.5.2 Reassembly Control Register 1 (RSM_CTRL1)

Access types: R/W
The reassembly control register 1, contains additional general control bits for the reassembly coprocessor.

Bit	Field Size	Name	Description
31–24	8	Reserved	Not implemented at this time.
23–20	4	Reserved	Not implemented at this time.
19–17	3	Reserved	Not implemented at this time.
16–13	4	Reserved	Not implemented at this time.
12	1	OAM_FF_DSC	OAM FIFO Full Discard. When a logic high and OAM_QU_EN is a logic high, an OAM cell will be discarded if the incoming DMA FIFO is almost full.
11	1	OAM_EN	Operating and maintenance enable. Enables detection and processing of OAM cells.
10	1	OAM_QU_EN	OAM Buffer/Status Queue Enable. When a logic high, an OAM cell will use the global OAM free buffer queue and status queue instead of per channel resources.
9–5	5	OAM_BFR_QU	OAM Free Buffer Queue. When OAM_QU_EN is a logic high, OAM cells will use buffers from the free buffer queue identified by OAM_BFR_QU.
4–0	5	OAM_STAT_QU	OAM Status Queue. When OAM_QU_EN is a logic high, OAM cells will post status to the status queue identified by OAM_STAT_QU.

13.5.3 Reassembly Free Buffer Queue Base Register (RSM_FQBASE)

Access types: R/W
This register determines the base address of both banks of contiguous free buffer queue spaces. The base address is a 16-bit number. Since both banks reside in SRC shared memory(23-bits of byte addressing), the structures can start on 128 byte boundaries. Bank 0 has additional boundary requirements if the buffer return mechanism is enabled.

Bit	Field Size	Name	Description
31–16	16	FBQ1_BASE	Free Buffer Queue Bank 1 base address.
15–0	16	FBQ0_BASE	Free Buffer Queue Bank 0 base address.

13.5.4 Reassembly Free Buffer Queue Control Register (RSM_FQCTRL)

Access types: R/W
This register contains Free Buffer Queue control information.

Bit	Field Size	Name	Description
31–16	16	Reserved	Not implemented at this time.
15–14	2	FBQ_SIZE	Free Buffer Queue Size. Selects the size of all Free Buffer Queues. 0–64 1–256 2–1024 3–4096
13	1	FWD_RND	Buffer Return Processing Priority Selection. When a logic low, buffer return entries are processed from queues in priority fashion with queue 15 having the highest priority. When a logic high, round-robin arbitration is used.
12	1	FBQ0_RTN	Free Buffer Queue0 Buffer Return Enable. When a logic high, bank 0 is enabled to process buffer return for firewall operation. When this bit is set, queue entries 0–15 are four words independent of the value of FWD_EN; otherwise they are two words.
11–8	4	FWD_EN	Forward Processing Enable. Selects the number of Free Buffer queues in bank 0 that have buffer return processing enabled. Starting with Free Buffer Queue 0, a value of 0 in FWD_EN selects only 1 queue and a value of 15 selects 16 queues.
7–0	8	FBQ_UD_INT	Free Buffer Queue Update Interval. This value determines how many buffers are taken off the Free Buffer Queue before the reassembly coprocessor writes the current read index pointer to host or SRC shared memory.

13.5.5 Reassembly Table Base Register (RSM_TBASE)

Access types: R/W
This register consists of a 16-bit address pointer that points to the beginning of the VPI Index table and a 16-bit address pointer that points to the beginning of the Rsm VCC Table. Since both tables reside in SRC shared memory, the tables start on 128-byte boundaries. The size of the VPI Index Table used in the direct index method is dependent upon the setting of RSM_CTRL0(VPI_MASK).

Bit	Field Size	Name	Description
31–16	16	RSM_VCCB	Reassembly VCC table base address.
15–0	16	RSM_ITB	VPI Index Table Base Address.

13.5.6 Reassembly Time-Out Register (RSM_TO)

Access types: R/W

Bit	Field Size	Name	Description
31–16	16	RSM_TO_PER	Reassembly Time-out Interrupt Period. The value in this register determines the number of sysclk periods for each time-out interrupt. A value of zero, divides by 65538.
15–0	16	RSM_TO_CNT	Reassembly Time-out Counter. The value in this register plus 1 determines the number of time-out interrupts that occur in each pass through the Rsm VCC Table.

13.5.7 Reassembly/Segmentation Queue Base Register (RS_QBASE)

Access types: R/W, L
This register contains the 128-byte aligned base address of the reassembly/segmentation queue structure.

Bit	Field Size	Name	Description
31–18	14	Reserved	Not Implemented at this time.
17–16	2	RS_SIZE[1:0]	Size of the RS Queue. Each RS Queue entry is 8 octets in length. 00–256 01–1024 10–4096 11–16384
15–0	16	RS_QBASE[15:0]	Base address of the reassembly/segmentation queue.

13.6 Control/Status Registers

13.6.1 ATM Cells Transmitted Counter (CELL_XMIT_CNT)

Access types: R/O

It counts the number of user data cells transmitted by the segmentation coprocessor. The counter is reset to 0 by an assertion of either the HRST* system reset pin or the GLOBAL_RESET bit in the CONFIG0 register. Optionally, an interrupt may be programmed when the counter rolls over.

Bit	Field Size	Name	Description
31–0	32	CELL_XMIT_CNT	Count of user data cells transmitted.

13.6.2 ATM Cells Received Counter (CELL_RCVD_CNT)

Access types: R/O

It counts the number of assigned cells received by the reassembly coprocessor. The counter is reset to 0 by an assertion of either the HRST* system reset pin or the GLOBAL_RESET bit in the CONFIG0 register. Optionally, an interrupt may be programmed when the counter rolls over.

Bit	Field Size	Name	Description
31–0	32	CELL_RCVD_CNT	Count of assigned received cells.

13.6.3 ATM Cells Discarded Counter (CELL_DSC_CNT)

Access types: R/O

It counts the number of unassigned cells received by the reassembly coprocessor. The counter is reset to 0 by an assertion of either the HRST* system reset pin or the GLOBAL_RESET bit in the CONFIG0 register. Optionally, an interrupt may be programmed when the counter rolls over.

Bit	Field Size	Name	Description
31–0	32	CELL_DSC_CNT	Count of unassigned cells dropped.

13.6.4 AAL5 CPCS-PDU Discard Counter (AAL5_DSC_CNT)

Access types: R/O
It counts the number of AAL5 CPCS-PDUs discarded due to buffer firewall, buffer underflow, or status overflow. The counter is reset to 0 by an assertion of either the HRST* system reset pin or the GLOBAL_RESET bit in the CONFIG0 register. Optionally, an interrupt may be programmed when the counter rolls over.

Bit	Field Size	Name	Description
31–0	16	Reserved	Not implemented at this time.
15–0	16	AAL5_DSC_CNT	AAL5 PDUs discarded by the reassembly coprocessor.

13.6.5 Host Mailbox Register (HOST_MBOX)

Access types: R/W L
This register implements a mailbox for communication between the host and local processors. The register is written by the local processor and read by the host to pass messages in that direction. Writes to this register may interrupt the host while reads may interrupt the local processor.

Bit	Field Size	Name	Description
31–0	32	HOST_MBOX[31:0]	Messages flow from local processor to host.

13.6.6 Host Status Write Register (HOST_ST_WR)

Access types: R/O
This register indicates if a reassembly or segmentation host located status queue has been written. Only queues 0 through 15 are supported. All bits are latched until read by the host. The RSM_HS_WRITE[15:0] bits are ORed together into HOST/LP_ISTAT0(RSM_HS_WRITE), and the SEG_HW_WRITE[15:0] bits are ORed together into HOST/LP_ISTAT0(SEG_HS_WRITE).

Bit	Field Size	Name	Description
31–16	16	RSM_HS_WRITE[15:0]	Indication that a host located reassembly status queue entry has been written. Only queues 0 through 15 are supported.
15–0	16	SEG_HS_WRITE[15:0]	Indication that a host located segmentation status queue entry has been written. Only queues 0 through 15 are supported.

13.6.7 Local Processor Mailbox Register (LP_MBOX)

Access types: R/W H

This register implements a mailbox for communication between the host and local processors. LP_MBOX is written by the host processor and read by the local processor to pass messages in that direction. Writes to this register may interrupt the local processor, while reads may interrupt the host processor.

Bit	Field Size	Name	Description
31–0	32	LP_MBOX[31:0]	Local processor mailbox register. Messages flow from host processor to local processor.

13.6.8 Host Interrupt Status Registers 0 (HOST_ISTAT0)

Access types: R/O

These two registers contain all interruptible status bits for the host processor. The corresponding interrupt enables are located in the HOST_IMASKx registers. Status types are defined as:

L	Level-sensitive status—A logic one on the status bit will cause an interrupt when enabled by the corresponding IMASK bit. Reading the status does not clear the status or interrupt. The source of the condition causing the status must be cleared before the status or interrupt is cleared.
E	Event driven status—A 0 ->1 transition on the status bit causes an interrupt when enabled. Reading the status register clears the status bit and the interrupt.
DE	Dual Event status—A 0 -> 1 and 1 -> 0 transition on the status bit can be enabled to cause an interrupt. Reading the status register clears the status bit and the interrupt.
Note: Only host reads will reset the status bits in the HOST_ISTAT0 register.	

Bit	Field Size	Type	Name	Description
31	1	L	PFAIL	Reflects inverted state of processor PFAIL* input.
30	1	L	PHY_INTR	In standalone operation, this bit reflects the inverted state of the PDAEN* input. PHY_INTR may be connected to a PHY interrupt source.
29	1	–	Reserved	Read as 0.
28	1	E	HOST_MBOX_ WRITTEN	This bit is set upon a write to the HOST_MBOX register by the local processor and cleared by a read of the HOST_MBOX register.
27	1	E	LP_MBOX_READ	This bit is set upon the read of the LP_MBOX register by the local processor.
26	1	–	Reserved	Read as 0.

Bit	Field Size	Type	Name	Description
25-24	2	–	Reserved	Read as 0.
23	1	–	Reserved	Read as 0. Reserved for future status page expansion.
22	1	L	HSTAT1	This bit is set when any bit in HOST_ISTAT1 is set.
21–19	3	–	Reserved	Read as 0.
18	1	E	GFC_LINK	Set when three consecutive received cells have GFC SET_A, SET_B, or HALT bits set.
17	1	L	RSM_RUN	Set when the reassembly machine is running. Will be high when the Rsm coprocessor is processing a cell.
16	1	L	RSM_HS_WRITE	Indicates reassembly host status has been written by Bt8233 to status queues 0 through 15. For queue number, read HOST_ST_WR which must be read in order to clear status bit.
15	1	E	RSM_LS_WRITE	Indicates reassembly local status has been written by Bt8233.
14–12	3	–	Reserved	Read as 0.
11	1	L	SEG_RUN	Set when the segmentation machine is running. Will be high when SEG_ENABLE bit in SEG_CTRL is high or when processing the last cell after SEG_ENABLE is low.
10	1	L	SEG_HS_WRITE	Indicates segmentation host status has been written by the Bt8233 to status queues 0 through 15. For queue number, read HOST_ST_WR which must be read in order to clear status bit.
9	1	E	SEG_LS_WRITE	Indicates that a segmentation local status queue has been written by the Bt8233.
8–4	5	–	Reserved	Read as 0.
3	1	E	AAL5_DSC_RLOVR	Set on the occurrence of an AAL5_DSC_CNT rollover.
2	1	E	CELL_DSC_RLOVR	Set on the occurrence of a CELL_DSC_CNT rollover.
1	1	E	CELL_RCVD_RLOVR	Set on the occurrence of a CELL_RCVD_CNT rollover.
0	1	E	CELL_XMT_RLOVR	Set on the occurrence of a CELL_XMIT_CNT rollover.

13.6.9 Host Interrupt Status Register 1 (HOST_ISTAT1)

Bit	Field Size	Type	Name	Description
31	1	L	PCI_BUS_EROR	This bit is set if the MERROR bit in the PCI configuration register is set. The MERROR bit is reset by either writing a logic one to the MERROR bit in the PCI configuration register, or setting the CONFIG0(PCI_ERR_RESET) bit to a logic high.
30–27	4	–	Reserved	Read as 0.
26	1	E	DMA_AFULL	Set when the incoming DMA burst FIFO becomes almost full.
25	1	E	FR_PAR_ERR	Set on the occurrence of a parity error on the reassembly ATM physical interface.
24	1	E	FR_SYNC_ERR	Set on the occurrence of a synchronization error on the reassembly ATM physical interface.
23–16	8	–	Reserved	Read as 0.
15	1	E	RS_QUEUE_FULL	Reassembly/segmentation queue full condition.
14	1	E	RSM_OVFL	Reassembly overflow. Indicates that a cell was lost due to a FIFO full condition.
13	1	E	RSM_HS_FULL	Set on the occurrence of a host status queue full condition.
12	1	E	RSM_LS_FULL	Set on the occurrence of a local status queue full condition.
11	1	E	RSM_HF_EMPT	Set on the occurrence of a host free buffer queue empty condition.
10	1	E	RSM_LF_EMPT	Set on the occurrence of a local free buffer queue empty condition.
9–3	7	–	Reserved	Read as 0.
2	1	E	SEG_UNFL	Segmentation underflow indicates that a scheduled cell could not be sent due to lack of PCI bandwidth.
1	1	E	SEG_HS_FULL	Indicates that the segmentation host status queue is full.
0	1	E	SEG_LS_FULL	Indicates that the segmentation local status queue is full.

13.6.10 Host Interrupt Mask Register 0 (HOST_IMASK0)

Access types: R/W H
This register contains the interrupt enables that correspond to the status bits in the HOST_ISTAT0 register. The assertion of the HRST* system reset pin will clear all of the HOST_IMASK0 interrupt enables.

Table 13-2. Host Interrupt Mask Register 0

Bit	Field Size	Name	Description
31	1	EN_PFAIL	Enables interrupt when PFAIL status is a logic 1
30	1	EN_PHY_INTR	Enables interrupt when PHY_INTR status is a logic 1.
29	1	Reserved	Set to 0.
28	1	EN_HOST_MBOX _WRITTEN	Enables interrupt when HOST_MBOX WRITTEN status is a logic 1.
27	1	EN_LP_MBOX_READ	Enables interrupt when LP_MBOX_READ status is a logic 1.
26	1	Reserved	Set to 0.
25–24	2	Reserved	Set to 0.
23	1	Reserved	Set to 0. Reserved for future status page expansion.
22	1	EN_HSTAT1	Global interrupt enable for HOST_ISTAT1 status register. Individual interrupts in HOST_ISTAT1 are enabled in HOST_IMASK1.
21–19	3	Reserved	Set to 0.
18	1	EN_GFC_LINK	Enables interrupt when GFC_LINK status is a logic 1
17	1	EN_RSM_RUN	Enables interrupt when RSM_RUN status is a logic 1.
16	1	EN_RSM_HS_WRITE	Enables interrupt when RSM_HS_WRITE status is a logic high.
15	1	EN_RSM_LS_WRITE	Enables interrupt when RSM_LS_WRITE status is a logic high.
14–12	3	Reserved	Set to 0.
11	1	EN_SEG_RUN	Enables interrupt when SEG_RUN status is a logic high.
10	1	EN_SEG_HS_WRITE	Enables interrupt when SEG_HS_WRITE status is a logic high.
9	1	EN_SEG_LS_WRITE	Enables interrupt when SEG_LS_WRITE status is a logic high.
8–4	5	Reserved	Set to 0.
3	1	EN_AAL5_DSC_RLOVR	Enables an interrupt when AAL5_DSC_RLOVR status is a logic high.
2	1	EN_CELL_DSC_RLOVR	Enables an interrupt when CELL_DSC_RLOVR status is a logic high.
1	1	EN_CELL_RCVD_RLOVR	Enables an interrupt when CELL_RCVD_RLOVR status is a logic high.
0	1	EN_CELL_XMIT_RLOVR	Enables an interrupt when CELL_XMIT_RLOVR status is a logic high.

13.6.11 Host Interrupt Mask Register 1 (HOST_IMASK1)

Access types: R/W H

This register contains the interrupt enables that correspond to the status in the HOST_ISTAT1 register. The assertion of the HRST* system reset pin will clear all of the HOST_IMASK1 interrupt enables.

Bit	Field Size	Name	Description
31	1	EN_PCI_BUS_ERROR	Enables interrupt when PCI_BUS_ERROR status is a logic 1.
30–27	4	Reserved	Set to 0.
26	1	EN_DMA_AFULL	Enabled interrupt when DMA_AFULL status is a logic 1.
25	1	EN_FR_PAR_ERR	Enables interrupt when FR_PAR_ERR status is a logic 1.
24	1	EN_FR_SYNC_ERR	Enables interrupt when FR_SYNC_ERR status is a logic 1.
23–16	8	Reserved	Set to 0.
15	1	EN_RSQUEUE_FULL	Enables interrupt when RSQUEUE_FULL status is a logic 1.
14	1	EN_RSM_OVFL	Enables interrupt when RSM_OVFL status is a logic 1.
13	1	EN_RSM_HS_FULL	Enables interrupt when RSM_HS_FULL status is a logic high.
12	1	EN_RSM_LS_FULL	Enables interrupt when RSM_LS_FULL status is a logic high.
11	1	EN_RSM_HF_EMPT	Enables interrupt when RSM_HF_EMPT status is a logic high.
10	1	EN_RSM_LF_EMPT	Enables interrupt when RSM_LF_EMPT status is a logic high.
9–3	7	Reserved	Set to 0.
2	1	EN_SEG_UNFL	Enables interrupt when SEG_UNFL status is a logic high.
1	1	EN_SEG_HS_FULL	Enables interrupt when SEG_HS_FULL status is a logic high.
0	1	EN_SEG_LS_FULL	Enables interrupt when SEG_LS_FULL status is a logic high.

13.7 Local Processor Interrupt Status Registers

Access types: R/O
The Local Processor Interrupt Status Registers contain all the interruptible status bits for the local processor. The corresponding interrupt enables are located in the LP_IMASKx registers. Status types are defined in the following table:

L	Level sensitive status—A logic 1 on the status bit will cause an interrupt when enabled by the corresponding IMASK bit. Reading the status does not clear the status or interrupt. The source of the condition causing the status must be cleared before the status or interrupt is cleared.
E	Event driven status—A 0 -> 1 transition on the status bit causes an interrupt when enabled. Reading the status register clears the status bit and the interrupt.
DE	Dual event status—A 0 -> 1 and 1 -> 0 transition on the status bit can be enabled to cause an interrupt. Reading the status register clears the status bit and the interrupt.
Note:	Only local processor reads will reset the status bits in the LP_ISTAT0 register.

13.7.12 Local Processor Interrupt Status Register 0 (LP_ISTAT0)

Bit	Field Size	Type	Name	Description
31	1	E	RTC_OVFL	Clock register overflow.
30	1	E	ALARM1	Set when ALARM1 register matches CLOCK register.
29	1	–	Reserved	Read as 0.
28	1	E	LP_MBOX_WRITTEN	This bit is set upon a write to the LP_MBOX register by the host processor.
27	1	E	HOST_MBOX_READ	This bit is set upon the read of the HOST_MBOX register by the host processor.
26	1	–	Reserved	Read as 0.
25–24	2	–	Reserved	Read as 0.
23	1	–	Reserved	Read as 0. Reserved for future status page expansion.
22	1	L	LSTAT1	This bit is set when any bit in LP_ISTAT1 is set.
21–19	3	–	Reserved	Read as 0.
18	1	E	GFC_LINK	Set when three consecutive received cells have GCF SET_A, SET_B or HALT bits set.
17	1	L	RSM_RUN	Set when the reassembly machine is running. Will be high when the Rsm coprocessor is processing a cell.

Bit	Field Size	Type	Name	Description
16	1	L	RSM_HS_WRITE	Indicates reassembly host status has been written by the Bt8233 to status queues 0 through 15. For queue number, read HOST_ST_WR which must be read in order to clear status bit.
15	1	E	RSM_LS_WRITE	Indicates that a reassembly local status queue has been written by the Bt8233.
14–12	3	–	Reserved	Read as 0.
11	1	L	SEG_RUN	Set when the segmentation machine is running. Will be high when SEG_ENABLE bit in SEG_CTRL is high or when processing the last cell after SEG_ENABLE is set low.
10	1	L	SEG_HS_WRITE	Indicates segmentation host status has been written by the Bt8233 to status queues 0 through 15. For queue number, read HOST_ST_WR, which must be read in order to clear status bit.
9	1	E	SEG_LS_WRITE	Indicates that a segmentation local status queue has been written by the Bt8233.
8–4	5	–	Reserved	Read as 0.
3	1	E	AAL5_DSC_RLOVR	Set on the occurrence of an AAL5_DSC_CNT rollover.
2	1	E	CELL_DSC_RLOVR	Set on the occurrence of a CELL_DSC_CNT rollover.
1	1	E	CELL_RCVD_RLOVR	Set on the occurrence of a CELL_RCVD_CNT rollover.
0	1	E	CELL_XMIT_RLOVR	Set on the occurrence of a CELL_XMIT_CNT rollover.

13.7.13 Local Processor Interrupt Status Register 1 (LP_ISTAT1)

Bit	Field Size	Type	Name	Description
31	1	L	PCI_BUS_EROR	This bit is set if the MERROR bit in the PCI configuration register is set. The MERROR bit is reset by either writing a logic one to the MERROR bit in the PCI configuration register, or setting the CONFIG0(PCI_ERR_RESET) bit to a logic high.
30–27	4	–	Reserved	Read as 0.
26	1	E	DMA_AFULL	Set when the incoming DMA burst FIFO becomes almost full.
25	1	E	FR_PAR_ERR	Set on the occurrence of a parity error on the reassembly ATM physical interface.
24	1	E	FR_SYNC_ERR	Set on the occurrence of a synchronization error on the reassembly ATM physical interface.
23–16	8	–	Reserved	Read as 0.
15	1	E	RS_QUEUE_FULL	Reassembly-segmentation queue full condition.
14	1	E	RSM_OVFL	Reassembly overflow. Indicates that a cell was lost due to a FIFO full condition.
13	1	E	RSM_HS_FULL	Set on the occurrence of a host status queue full condition.
12	1	E	RSM_LS_FULL	Set on the occurrence of a local status queue full condition.
11	1	E	RSM_HF_EMPT	Set on the occurrence of a host free buffer queue empty condition.
10	1	E	RSM_LF_EMPT	Set on the occurrence of a local free buffer queue empty condition.
9-3	7	–	Reserved	Read as 0.
2	1	E	SEG_UNFL	Segmentation underflow indicates that a scheduled cell could not be sent due to lack of PCI bandwidth.
1	1	E	SEG_HS_FULL	Indicates that the segmentation host status queue is full.
0	1	E	SEG_LS_FULL	Indicates that the segmentation local status queue is full.

13.7.14 Local Processor Interrupt Mask Register 0 (LP_IMASK0)

Access type: R/W L
This register contains the interrupt enables that correspond to the status in the LP_ISTAT0 register. The assertion of the HRST* system reset pin clears all of the LP_IMASK0 interrupt enables.

Bit	Field Size	Name	Description
31	1	EN_RTC_OVFL	Enables an interrupt when RTC_OVFL status is a logic high.
30	1	EN_ALARM1	Enables an interrupt when ALARM1 status is a logic one.
29	1	Reserved	Set to 0.
28	1	EN_LP_MBOX_WRITTEN	Enables an interrupt when LP_MBOX_WRITTEN status is a logic 1
27	1	EN_HOST_MBOX_READ	Enables an interrupt when HOST_MBOX_READ status is a logic 1.
26	1	Reserved	Set to 0.
25–24	2	Reserved	Set to 0.
23	1	Reserved	Set to 0. Reserved for future status page expansion.
22	1	EN_LSTAT1	Global interrupt enable for LP_ISTAT1 status register. Individual interrupts of LP_ISTAT1 are enabled in LP_IMASK1.
21–19	3	Reserved	Set to 0.
18	1	EN_GFC_LINK	Enables an interrupt when GFC_LINK status is a logic 1.
17	1	EN_RSM_RUN	Enables an interrupt when RSM_RUN status is a logic 1
16	1	EN_RSM_HS_WRITE	Enables an interrupt when RSM_HS_WRITE status is a logic high.
15	1	EN_RSM_LS_WRITE	Enables an interrupt when RSM_LS_WRITE status is a logic high.
14–12	3	Reserved	Set to 0.
11	1	EN_SEG_RUN	Enables an interrupt when SEG_RUN status is a logic high.
10	1	EN_SEG_HS_WRITE	Enables an interrupt when SEG_HS_WRITE status is a logic high.
9	1	EN_SEG_LS_WRITE	Enables an interrupt when SEG_LS_WRITE status is a logic high.
8–4	5	Reserved	Set to 0.
3	1	EN_AAL5_DSC_RLOVR	Enables an interrupt when AAL4_DSC_RLOVR status is a logic high.
2	1	EN_CELL_DSC_RLOVR	Enables an interrupt when CELL_DSC_RLOVR status is a logic high.
1	1	EN_CELL_RCVD_RLOVR	Enables an interrupt when CELL_RCVD_RLOVR status is a logic high.
0	1	EN_CELL_XMIT_RLOVR	Enables an interrupt when CELL_XMIT_RLOVR status is a logic high.

13.7.15 Local Processor Interrupt Mask Register 1 (LP_IMASK1)

Access types: R/W L
This register contains the interrupt enables that correspond to the statuses in the LP_ISTAT1 register. The assertion of the HRST* system reset pin will clear all of the LP_IMASK1 interrupt enables.

Bit	Field Size	Name	Description
31	1	EN_PCI_BUS_ERROR	Enables an interrupt when PCI_BUS_ERROR status is a logic 1.
30–27	4	Reserved	Set to 0.
26	1	EN_DMA_AFULL	Enables an interrupt when DMA_AFULL status is a logic 1.
25	1	EN_FR_PAR_ERR	Enables an interrupt when FR_PAR_ERR status is a logic 1.
24	1	EN_FR_SYNC_ERR	Enables an interrupt when FR_SYNC_ERR status is a logic 1.
23–16	8	Reserved	Set to 0.
15	1	EN_RSQUEUE_FULL	Enables an interrupt when RSQUQUQ_FULL status is a logic 1.
14	1	EN_RSM_OVFL	Enables an interrupt when RSM_OVFL status is a logic 1.
13	1	EN_RSM_HS_FULL	Enables an interrupt when RSM_HS_FULL status is a logic high.
12	1	EN_RSM_LS_FULL	Enables an interrupt when RSM_LS_FULL status is a logic high.
11	1	EN_RSM_HS_EMPT	Enables an interrupt when RSM_HS_EMPT status is a logic high.
10	1	EN_RSM_LS_EMPT	Enables an interrupt when RSM_LS_EMPT status is a logic high.
9–3	7	Reserved	Set to 0.
2	1	EN_SEG_UNFL	Enables an interrupt when SEG_UNFL status is a logic high.
1	1	EN_SEG_HS_FULL	Enables an interrupt when SEG_HS_FULL status is a logic high.
0	1	EN_SEG_LS_FULL	Enables an interrupt when SEG_LS_FULL status is a logic high.

13.8 PCI Bus Interface Configuration Registers

In accordance with the PCI Bus Specification Revision 2.0, the SAR PCI bus interface implements a 128-byte configuration register space. These configuration registers are used by the host processor to initialize, control, and monitor the PCI bus interface logic. The complete definitions of these registers and the relevant fields within them is given in the PCI bus specification. The implementation of these registers in the Bt8233 is indicated is shown in Table 13-3. Descriptions of fields are given in Table 13-4.

Table 13-3. PCI Configuration Register Definition

Byte Addr	Configuration Register Definition																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DEVICE_ID																VENDOR_ID															
0x04	STATUS																COMMAND															
0x08	0X020300																								REV_ID							
0x0C	0X0000																LAT_TIMER								0X00							
0x10	BT8233_MEM_BASE																0X000000															
0x14-0x38	0X00000000																															
0x3C	0X050201																								ILINE							
0X40	0X00000																				SPECIAL STATUS				MAX_BURST_LEN							
0X44	CUR_MSTR_RD_ADDR																															
0X48	CUR_MSTR_WR_ADDR																															
0X4C-0X7C	0X00000000																															

Table 13-4. PCI Configuration Registers Field Descriptions *(1 of 2)*

<table>
<tr><th>Field Name</th><th>Description/Function</th></tr>
<tr><td>DEVICE_ID</td><td>16-bit device identifier. Serves to uniquely identify the SAR to the host operating system. Set to 0x8233</td></tr>
<tr><td>VENDOR_ID</td><td>16-bit vendor identifier code, allocated on a global basis by the PCI SIG. Set to 0x109E.</td></tr>
<tr><td>STATUS</td><td>PCI bus interface status register. The PCI host may monitor its operation using the STATUS field. This field is further divided into subfields as shown below. These bits may be reset by writing a logic high to the appropriate bit.
<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27–25</td><td>24</td><td>23–16</td></tr>
<tr><td>DPE</td><td>SSE</td><td>RMA</td><td>RTA</td><td>0x0</td><td>DPR</td><td>0x80</td></tr>
</table>
DPE Detected Parity Error
SSE Signaled System Error
RMA Received Master Abort
RTA Received Target Abort
DPR Reported Data Parity Error</td></tr>
<tr><td>COMMAND</td><td>PCI bus interface control/command register. The PCI host may configure the SAR bus interface logic using the COMMAND field. This field is further divided into subfields as shown below. Active HRST* input causes all bits to be a logic 0.
<table>
<tr><td>15–10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5–3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0x00</td><td>FB_EN</td><td>SE_EN</td><td>0x0</td><td>PE_EN</td><td>0x0</td><td>M_EN</td><td>MS_EN</td><td>0x0</td></tr>
</table>
FB_EN Fast back-to-back enable across target
SE_EN SERR* pin output enable
PE_EN Parity report enable
M_EN Master enable. M_EN must be asserted (i.e., set to 1) before the Bt8233 can act as master on the PCI bus.
MS_EN Memory space enable. MS_EN must be asserted (i.e., set to 1) before the Bt8233 address space (registers & memory) can be accessed across the PCI Interface.</td></tr>
<tr><td>REV_ID</td><td>Revision ID code for the Bt8233 chip.</td></tr>
<tr><td>LAT_TIMER</td><td>Latency timer. Value after HRST* active is 0x0. All bits are writable. The suggested value is 0x10 in order to allow the complete transfer of a cell.</td></tr>
</table>

Table 13-4. PCI Configuration Registers Field Descriptions *(2 of 2)*

<table>
<tr><th>Field Name</th><th>Description/Function</th></tr>
<tr><td>BT8233_MEM_BASE</td><td>Base address of PCI address space occupied by the Bt8233 (as seen and assigned by the host processor). Value after HRST* active is 0x00.</td></tr>
<tr><td>ILINE</td><td>Interrupt line identifier.</td></tr>
<tr><td>SPECIAL_STATUS</td><td>Device status not defined by the PCI specification. The field is further subdivided into subfields as shown below. Detailed descriptions of the above subfields can be found in the PCI bus specification. Note that the configuration registers are accessed starting from byte address 0 in the configuration space allotted to an adapter card containing the SAR chip. Access to the configuration registers is available only to the PCI host CPU, and is independent of all other SAR logic.
<table>
<tr><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>INTF_DIS</td><td>INT_FAIL</td><td>MERROR</td><td>MRD</td></tr>
</table>
INTF_DIS — If the M_EN bit in the COMMAND field is a logic low, any attempt by the Bt8233 to perform a DMA transaction to the PCI bus will result in an error. INTF_DIS and MERROR bits will both be set at a logic high. This bit may be reset by writing a logic high to itself.
INT_FAIL — Set to a logic 1 when the an internal PCI/DMA synchronization error has occurred. The MERROR bit will also be set to a logic high. This bit may be reset by writing a logic high to itself.
MERROR — Indicates that the PCI bus master has encountered a fatal error and therefore has halted operation. Set when either RTA, RMA, DPR, INTF_DIS, or INT_FAIL errors occur. This bit may be reset by writing a logic high to itself.
MRD — If logic high, indicates that the errored transaction was a read and the address of the read is located in the CUR_MSTR_RD_ADDR field. If logic low, indicates a write with the corresponding address located in the CUR_MSTR_WR_ADDR field (read-only).</td></tr>
<tr><td>MAX_BURST_LEN</td><td>Sets the maximum number of words that can be transferred in a single transaction by the SAR PCI bus master. A value of zero is invalid and can lead to erroneous and unpredictable results. When HRST* input pin is active, this register is loaded with 16 decimal.</td></tr>
<tr><td>CUR_MSTR_RD _ADDR</td><td>Current read target address used by PCI bus master (read only).</td></tr>
<tr><td>CUR_MSTR_WR _ADDR</td><td>Current write target address used by PCI bus master (read only).</td></tr>
</table>

14.0 SAR Initialization–Example Tables

NOTE: The following tables provide an sample SAR initialization of control registers, internal memory control structures and external memory control structures.

14.1 Segmentation Initialization

14.1.1 Segmentation Control Registers

Before segmentation is enabled, the host must allocate and initialize all of the segmentation control registers. Table 14-1 lists the initial value(s) for each field.

Table 14-1. Table of Values for Segmentation Control Register Initialization *(1 of 2)*

Register	Field	Initialized Value	Notes
SEG_CTRL (Segmentation Control Register.	SEG_ENABLE	0–1	Must be set to a logic low until initialization of all segmentation structures is complete. Set to a logic high to commence segmentation process.
	SEG_RESET	0	Use CONFIG0(GLOBAL_RESET) to initialize the SAR.
	VBR_OFFSET	0	Schedule slot priority equals general priority.
	SEG_GFC	0	Disable segmentation GFC processing.
	DBL_SLOT	1	Enable two-word schedule slot entries.
	CBR_TUN	1	Enable CBR traffic scheduling.
	TX_FIFO_LEN	0x4	Set Transmit FIFO depth to 4 cells.
	CLP0_EOM	0	Disable CLP on EOM processing.
	OAM_STAT_ID	0x10	OAM global status queue set to 16.
	SEG_ST_HALT	0	Disable status queue entry for a VCC halted with a partially segmented buffer.
	SEG_LS_DIS	0	Enable local status queue full check.
	SEG_HS_DIS	0	Enable host status queue full check.
	TX_RND	1	Round-robin transmit queue processing selected.
	TR_SIZE	0x0	Transmit Queue size set to 64 entries.

***Table 14-1. Table of Values for Segmentation Control Register Initialization** (2 of 2)*

Register	Field	Initialized Value	Notes
SEG_VBASE (Seg Virtual Channel Connection Base Address Register.)	SEG_SCHB	0x1A5	Schedule Table starts at 0xD280 in SRC shared memory.
	SEG_VCCB	0x17D	Seg VCC Table starts at 0xBE80 in SRC shared memory.
SEG_PMBASE (Seg PM Base Address Register.	SEG_BCKB	0x219	VBR Bucket Table starts at 0x10C80 in SRC shared memory.
	SEG_PMB	0x1AD	PM Table starts at 0xD680 in SRC shared memory.
SEG_TXBASE (Segmentation Transmit Queue Base Register.	SEG_TXB	0x13D	Transmit Queues start at 0x9E80 in I SRC shared memory.
	XMIT_INTERVAL	0x20	Transmit Queue update interval set to 32.
	TX_EN	0x8	Transmit Queues 0 through 8 are enabled.

14.1.2 Segmentation Internal Memory Control Structures

Before segmentation is enabled, the host must allocate and initialize all of the segmentation Internal Memory Control Structures. Table 14-2 lists the initial value(s) for each field.

Table 14-2. Table of Values for Seg Internal Memory Initialization

Table	Field	Initialized Value	Notes
SEG_SQ_BASE Table Entry 0 (Seg Status Queue Base Table Entry 0.)	BASE_PNTR	0x1C00	Base address of Status Queue 0 is 0x7000.
	LOCAL	0	Status Queue 0 resides in host memory.
	SIZE	0x0	Size of Status Queue 0 is 64 entries.
	WRITE	0x0	MUST be initialized to 0.
	READ_UD	0x0	MUST be initialized to 0.
	Rsvd	0x0	MUST be initialized to 0.
SEG_TQ_BASE Table Entry 0 (Seg Transmit Queue BAse Table Entry 0.)	READ_UD_PNTR	0x40	Location of READ_UD is at 0x100.
	LOCAL	0	READ_UD located in host memory.
	UPDATE	0x0	MUST be initialized to 0.
	READ	0x0	MUST be initialized to 0.
	Rsvd	0x0	MUST be initialized to 0.

14.1.3 Segmentation SRC Shared Memory Control Structures

Before segmentation is enabled, the host must allocate and initialize all of the segmentation SRC shared memory control structures. Table 14-3 lists the initial value(s) for each field.

Table 14-3. Table of Values for Seg SRC Shared Memory Initialization *(1 of 2)*

Table	Field	Initialized Value	Notes
SEG VCC Table Entry 0 - (Words 0 - 6)	PM_INDEX	0x0	PM Table Index equal to 0.
	LAST_PNTR	0x0	MUST be initialized to 0.
	ATM_HEADER	0x00100100	ATM Header, VPI=0x1, VCI=0x10.
	CRC_REM	0xFFFF_FFFF	MUST be initialized to 0xFFFF_FFFF for AAL5 channels.
	STM_MODE	0	Status message mode enabled.
	STAT	0x2	Channel will use Status Queue number 2.
	PM_EN	1	PM OAM processing enabled.
	CURR_PNTR	0x0	MUST be initialized to 0.
	VPC	0	VCC connection.
	SCH_MODE	0x4	Channel configured for VBR traffic.
	PRI	0x3	Schedule priority is 3.
	SCH_OPT	1	Send maximum burst.
	Rsvd	0x0	MUST be initialized to 0.
SEG Buffer Descriptors			(No initialization required.)
SEG Transmit Queue Entries.	VLD	0	MUST be initialized to 0.
	LINK_HEAD	0	MUST be initialized to 0.
	FND_CHAIN	0	MUST be initialized to 0.
	SEG_BD_PNTR	0x0	MUST be initialized to 0.
	Rsvd	0x0	MUST be initialized to 0.
SEG Status Queue Entries	USER_PNTR	0x0	MUST be initialized to 0.
	VLD	0	MUST be initialized to 0.
	STOP	0	MUST be initialized to 0.
	DONE	0	MUST be initialized to 0.
	SINGLE	0	MUST be initialized to 0.
	OVFL	0	MUST be initialized to 0.
	I_EXP	0x0	MUST be initialized to 0.
	I_MAN	0x0	MUST be initialized to 0.
	SEG_VCC_INDEX	0x0	MUST be initialized to 0.
	Rsvd	0x0	MUST be initialized to 0.

***Table 14-3. Table of Values for Seg SRC Shared Memory Initialization** (2 of 2)*

Table	Field	Initialized Value	Notes
SEG_PM Table Entry 0	ATM_HEADER	0x00100108	VPI=0x1, VCI=0x10, PTI=4 (F5 segment).
	FWD_TUC0	0x0	MUST be initialized to 0.
	FWD_TUC01	0x0	MUST be initialized to 0.
	BCK_MSN	0x0	MUST be initialized to 0.
	BCK_TUC0	0x0	MUST be initialized to 0.
	BCK_TUC01	0x0	MUST be initialized to 0.
	TRCC0	0x0	MUST be initialized to 0.
	TRCC0+1	0x0	MUST be initialized to 0.
	BIP	0x0	MUST be initialized to 0.
	FWD_MON	1	Forward Monitoring cell generation enabled.
	BLOCK_SIZE	10	PM OAM block size of 512.
	FWD_MSN	0x3	Initial sequence number is 3.
	Rsvd	0x0	MUST be initialized to 0.

14.2 Scheduler Initialization

14.2.1 Scheduler Control Registers

Before segmentation is enabled, the host must allocate and initialize all of the Scheduler control registers. Table 14-4 lists the initial value(s) for each field.

Table 14-4. Table of Values for Scheduler Control Register Initialization

Register	Field	Initialized Value	Notes
SCH_PRI (Schedule Priority Register)	TUN_ENA7-0	0	No tunnels enabled.
	GFC7-0	0	No GFC priorities enabled.
SCH_SIZE (Schedule Size Register)	TBL_SIZE	0x80	Schedule Table consists of 128 entries.
	SLOT_PER	0x5B	Schedule slot period is 91 SYSCLK periods.
SCH_ABR_MAX (Schedule Maximum ABR Register)	VCC_MAX	0x63	Enable 50 channels of ABR processing.
SCH_ABR_CON (Schedule ABR Constant Register)	ABR_TRM	0x23C	Set TRM to TM4.0 default of 100 msec.
	ABR_ADTF	0xB2E	Set ADTF to TM4.0 default of 0.5 second.
SCH_ABRBASE (ABR Decision Table Lookup Base Reg)	OOR_ENA	1	Enable out of rate ABR RM cells.
	OOR_INT	0x1C9D	Produces an out of rate interval of 1 cell per second.
	SCH_ABRB	0x1D1	ABR Table starts at 0xE880 in SRC shared memory.
SCH_CNG (ABR Congestion Register)	FBQ_CNG	0x0	No congestion experienced.

14.2.2 Scheduler Internal Memory Control Structures

There are no internal memory scheduler structures that need to be initialized.

14.2.3 Scheduler SRC Shared Memory Control Structures

Before segmentation is enabled, the host must allocate and initialize all of the scheduler SRC shared memory control structures. Table 14-5 lists the initial value(s) for each field.

Table 14-5. Table of Values for Scheduler SRC Shared Memory Initialization *(1 of 2)*

Table	Field	Initialized Value	Notes
Schedule Table (CBR slot)	CBR	1	Slot will schedule a CBR channel.
	CBR_VCC_INDEX	0x0	Schedule SEG_VCC_INDEX = 0 channel as CBR
	Rsvd	0xF	Set all reserved bits to 1.
Schedule Table (Tunnel slot)	CBR	0	Slot will schedule a tunnel.
	PRI	0x3	Tunnel Priority Queue = 3.
	Rsvd	0xF	Set all reserved bits to 1.
Schedule Table (DEFAULT)	Rsvd	0xF	Set all reserved bits to 1.
Seg VCC Table Entry for VBR (Words 7 - 8)	BUCKET2	0x4	Offset of 4 into Bucket Table for VBR2 parameters.
	L1_EXP	0xB	
	L1_MAN	0x0	GCRA L = 2.
	I1_EXP	0xB	
	I1_MAN	0x100	GCRA I = 3.
	Rsvd	0x0	MUST be initialized to 0.
Bucket Table Entry	L2_EXP	0xA	
	L2_MAN	0x0	GCRA L = 1.
	I2_EXP	0xB	
	I2_MAN	0x0	GCRA I = 2.
	Rsvd	0x0	MUST be initialized to 0.

Table 14-5. Table of Values for Scheduler SRC Shared Memory Initialization *(2 of 2)*

Table	Field	Initialized Value	Notes
Seg VCC Table Entry for ER (Words 7–19)	COND_ID	0x1	Channel associated with RSM Free Buffer Queue 1 for Host congestion indication.
	OOR_PRI	0x2	Out of rate RM cells assigned to priority queue 2
	CRM	0x14	TM4.0 CRM parameter set to 20.
	FWD_ID	0x01	Forward RM cell ID field set to 1.
	FWD_DIR	0	Forward RM cell DIR field set to 0.
	FWD_BN	0	Forward RM cell BN field set to 0.
	FWD_CI	0	Forward RM cell CI field set to 0.
	FWD_NI	0	Forward RM cell NI field set to 0.
	FWD_RA	0	Forward RM cell RA field set to 0.
	FWD_ER	0x64BF	Forward RM cell ER field set to approximately 360000 cells per second.
	Rsvd	0x0	MUST be initialized to 0.
	(ALL OTHER FIELDS)	—	Direct output of ABR template.
ABR Cell Decision Block	(ALL FIELDS)	—	Direct output of ABR template.
ABR Rate Decision Block	(ALL FIELDS)	—	Direct output of ABR template.
Exponent Table	(ALL FIELDS)	—	Direct output of ABR template.

14.3 Reassembly Initialization

14.3.1 Reassembly Control Registers

Before reassembly is enabled, the host must allocate and initialize all of the reassembly control registers. Table 14-6 lists the initial value(s) for each field.

Table 14-6. Table of Values for Rsm Control Register Initialization *(1 of 2)*

Register	Field	Initialized Value	Notes
RSM_CTRL0 (Reassembly Control Register 0)	RSM_ENABLE	0–1	Must be set to a logic low until initialization of all reassembly structures is complete. Set to a logic high to commence reassembly process.
	RSM_RESET	0	Use CONFIG0(GLOBAL_RESET) to initialize the SAR.
	VPI_MASK	1	UNI VPI space selected.
	RSM_PHALT	0	Rsm halt on receive parity error disabled.
	FWALL_EN	1	Firewall function enabled.
	RSM_FBQ_DIS	0	Free Buffer Queue underflow protection enabled.
	RSM_STAT_DIS	0	Status Queue overflow protection enabled.
	GTO_EN	1	Enable internal time-out interrupt.
	MAX_LEN	0x10	AAL5 max CPCS-PDU length = 16384 bytes.
	GDPRI	0x4	Global Service Discard Priority = 4.
	Rsvd	0x0	MUST be initialized to 0.
RSM_CTRL1 (Reassembly Control Register 1)	OAM_FF_DSC	1	OAM cells discarded when Incoming DMA FIFO almost full.
	OAM_EN	1	OAM detection enabled.
	OAM_QU_EN	1	Global OAM FB and STAT queues enabled.
	OAM_BFR_QU	0x4	Global OAM Free Buffer Queue = 4.
	OAM_STAT_QU	0x4	Global OAM Status Queue = 1.
	Rsvd	0x0	MUST be initialized to 0.
RSM_FQBASE (Rsm Free Buffer Queue Base Register)	FBQ1_BASE	0xB0	Free Buffer Queue Bank 1 starts at 0x5800 in SRC shared memory.
	FBQ0_BASE	0x30	Free Buffer Queue Bank 0 starts at 0x1800 in SRC shared memory.

Table 14-6. Table of Values for Rsm Control Register Initialization *(2 of 2)*

Register	Field	Initialized Value	Notes
RSM_FQCTRL (Rsm Free Buffer Queue Control Register)	FBQ_SIZE	0x0	Free Buffer Queue size is 64 entries.
	FWD_RND	1	Free Buffer Queues are processed in round-robin fashion for credit return.
	FBQ0_RTN	0	Free Buffer Queue Bank 0 selected for buffer return processing.
	FWD_EN	0x4	Free Buffers Queues 0 through 4 enabled for buffer return processing.
	FBQ_UD_INT	0x20	Free Buffer Queue Update interval set to 32.
	Rsvd	0x0	MUST be initialized to 0.
RSM_TBASE (Rsm Table Base Register)	RSM_VCCB	0x105	Rsm VCC Table starts at 0x8280 in SRC shared memory.
	RSM_ITB	0xF0	VPI Index Table starts at 0x7800 in SRC shared memory.
RSM_TO (Rsm Time-out Register)	RSM_TO_PER	0x100	Internal time-out interrupt every 256 SYSCLK periods.
	RSM_TO_CNT	0x10	Rsm VCC Table entries 0 through 16 enabled for time-out processing.
RS_QBASE (Rsm/Seg Queue Base Register)	RS_SIZE	0x0	Rsm/Seg Queue size is 256 entries.
	RS_QBASE	0x12D	Rsm/Seg Queue starts at 0x9680 in SRC shared memory.
	Rsvd	0x0	MUST be initialized to 0.

14.3.2 Reassembly Internal Memory Control Structures

Before reassembly is enabled, the host must allocate and initialize all of the reassembly Internal Memory Control Structures. Table 14-7 lists the initial value(s) for each field.

Table 14-7. Table of Values for Rsm Internal Memory Initialization

Table	Field	Initialized Value	Notes
RSM_SQ_BASE Table Entry 0 (Rsm Status Queue Base Table Entry 0)	BASE_PNTR	0x1800	Base address of Rsm Status Queue 0 is 0x6000.
	LOCAL	0	Status Queue 0 resides in host memory.
	SIZE	0x0	Size of Status Queue 0 is 64 entries.
	WRITE	0x0	MUST be initialized to 0.
	READ_UD	0x0	MUST be initialized to 0.
	Rsvd	0x0	MUST be initialized to 0.
RSM_FBQ_BASE Table Entry 0 (Rsm Free Buffer Queue BAse Table Entry 0)	READ_UD_PNTR	0x0	Location of READ_UD is at 0x0.
	BD_LOCAL	0	Buffer descriptors and READ_UD located in host memory.
	BFR_LOCAL	0	Buffers located in host memory.
	EMPT	0	MUST be initialized to 0.
	UPDATE	0x0	MUST be initialized to 0.
	READ	0x0	MUST be initialized to 0.
	FORWARD	0x20	32 free buffers initially put on Free Buffer Queue.
	LENGTH	0x200	Buffer Lengths in Free Buffer Queue 0 are 200 bytes.
	Rsvd	0x0	MUST be initialized to 0.
Global Time-out Table	TERM_TOCNT0	0x400	Provides a 133 msec time-out period.
	TERM_TOCNT1	0xFFFF	Provides a 8.5 sec time-out period.
	TERM_TOCNT2	0x400	Provides a 133 msec time-out period.
	TERM_TOCNT3	0x400	Provides a 133 msec time-out period.
	TERM_TOCNT4	0x400	Provides a 133 msec time-out period.
	TERM_TOCNT5	0x400	Provides a 133 msec time-out period.
	TERM_TOCNT6	0x400	Provides a 133 msec time-out period.
	TERM_TOCNT7	0x400	Provides a 133 msec time-out period.
	TO_VCC_INDEX	0x0	MUST be initialized to 0.
	Rsvd	0x0	MUST be initialized to 0.

14.3.3 Reassembly SRC Shared Memory Control Structures

Before reassembly is enabled, the host must allocate and initialize all of the reassembly SRC shared memory control structures. Table 14-8 lists the initial value(s) for each field.

Table 14-8. Table of Values for Rsm SRC Shared Memory Initialization *(1 of 2)*

Table	Field	Initialized Value	Notes
VPI Index Table Entry 0	VP_EN	1	VPI=0 enabled. NOTE: All entries in the VPI Index table must be initialized. If VPI is disabled, set VP_EN=0.
	VCI_RANGE	0x10	Allowable VCI range is 0 to 0x43F.
	VCI_IT_PNTR	0x2014	VCI Index Table is located at 0x8050 in SRC shared memory.
VCI Index Table Entry 0	VCC_BLOCK_PNTR	0x0	Initial block of 64 VCC Table entries is 0.
	Rsvd	0x0	MUST initialized to 0.
RSM VCC Table Entry 0	FF_DSC	1	Enable FIFO Full early packet discard.
	VC_EN	1	Table Entry is enabled. NOTE all VCC Table entries that have a path through the VPI/VCI lookup space must be initialized. If Entry is disabled, set VC_EN=0.
	AAL_TYPE	0x0	Channel configured for AAL5.
	DPRI	0x2	Channel service discard priority set to 2.
	TO_INDEX	0x1	Point to TERM_TOCNT1 global time-out value.
	PM_INDEX	0x2	Channel used entry 2 of PM_OAM table.
	AAL_EN	0x082	Message Status mode and Frame Relay Discard enabled.
	TO_LAST	0	Not last vcc entry processed for time-out.
	TO_EN	1	Time-out processing enabled on this channel.
	CUR_TOCNT	0x0	MUST be initialized to 0.
	ABR_CTRL	0x0	No ABR service enabled on this channel.
	PDU_FLAGS	0x002	MUST be initialized to 2.
	TOT_PDU_LEN	0x0	MUST be initialized to 0.
	CRC_REM	0xFFFF_FFFF	MUST be initialized to 0xFFFF_FFFF for AAL5 channels.
	STAT	0x1	Channel uses Status Queue 1.
	BFR1	0x11	Channel uses Free Buffer Queue 17 for COM buffers.
	BFR0	0x1	Channel uses Free Buffer queue 1 for BOM buffers.
	SEG_VCC_INDEX	0x4	Corresponding seg channel = 4 for PM_OAM and ABR service.
	SERV_DIS	0x0	MUST be initialized to 0.
	RX_COUNTER	0x100	Initial firewall credit = 256.
	Rsvd	0x0	MUST be initialized to 0.

Table 14-8. Table of Values for Rsm SRC Shared Memory Initialization *(2 of 2)*

Table	Field	Initialized Value	Notes
RSM Buffer Descriptors	NEXT_PTR	0x0	Initialization Optional.
	BUFF_PNTR	0x2000	Buffer Address of 0x2000.
RSM Free Buffer Queue Entries.	VLD	0–	Free Buffer Queue may be initialized with several free buffers. Valid entries should have VLD=1 and invalid entries should have VLD=0.
	BUFFER_PNTR	0x2000	Buffer address of 0x2000.
	BD_PNTR	0x80	Buffer Descriptor address of 0x200.
	FWD_VLD	0	MUST be initialized to 0.
	VCC_INDEX	0	MUST be initialized to 0.
	Rsvd	0x0	MUST be initialized to 0.
RSM Status Queue Entries	VLD	0	MUST be initialized to 0.
	(ALL OTHER ENTRIES)	0	MUST be initialized to 0.
LECID Table	LECID0-31	0x20	LANE LECID = 32.
RSM_PM Table Entry 0	BCNT	0x0	MUST be initialized to 0.
	BIP16	0x0	MUST be initialized to 0.
	MSN	0x4	First expected MSN in forward monitoring cell is 4.
	Rsvd	0x0	MUST be initialized to 0.
	TRCC0	0x0	MUST be initialized to 0.
	TRCC01	0x0	MUST be initialized to 0.

14.4 General Initialization

14.4.1 General Control Registers

Before the SAR is enabled, the host must allocate and initialize all of the general SAR control registers. Table 14-9 lists the initial value(s) for each field.

Table 14-9. Table of Values for General Control Register Initialization *(1 of 2)*

Register	Field	Initialized Value	Notes
CONFIG0 (Configuration Register 0)	LP_ENABLE	0	No SRC shared processor used.
	GLOBAL_RESET	0–1	This must be toggled to a logic high and back to a logic low after completion of all initialization but before Rsm and Seg coprocessors are enabled.
	PCI_MSTR_RESET	0	Use GLOBAL_RESET to reset SAR.
	PCI_ERR_RESET	0	MUST be initialized to 0.
	PHY2_EN	0	Only one PHY device used.
	INT_LBANK	0–1	Should be set to 0 during initialization but set to one after system reset.
	PCI_READ_MULTI	1	PCI Read Multiple Command used.
	PCI_ARB	1	Round-robin arbitration of internal read/write PCI master.
	STATMODE	0x0	Selects BOM sync hardware mode.
	FR_RMODE	0	Early Rsm header processing enabled.
	FR_LOOP	0	Internal ATM physical interface disabled.
	UTOPIA_MODE	1	Cell handshake mode selected.
	ENDIAN	1	Big Endian mode selected.
	LP_BWAIT	0	Selects zero wait states between consecutive data cycles during SRC shared processor.
	MEMCTRL	0	Selects zero wait states SRC shared memory.
	BANKSIZE	0x3	512 KB banks selected.
	DIVIDER	0x0	Divide by 128 selected for CLOCK prescaler.
	Rsvd	0x0	MUST be initialized to 0.
HOST_ST_WR (Host Status Write Register)	RSM_HS_WRITE	—	Read twice after SAR reset.
	RSM_LS_WRITE	—	Read twice after SAR reset.
HOST_ISTAT1 (Host Interrupt Status Register 1)	(ALL)	—	Read twice HOST_ST_WR reset.

***Table 14-9. Table of Values for General Control Register Initialization** (2 of 2)*

Register	Field	Initialized Value	Notes
HOST_ISTAT0 (Host Interrupt Status Register 0)	(ALL)	—	Read twice HOST_ISTAT1 reset.
LP_ISTAT1 (Local Interrupt Status Register 1)	(ALL)	—	Read twice SAR reset.
LP_ISTAT0 (Local Interrupt Status Register 0)	(ALL)	—	Read twice LP_ISTAT1 reset.
HOST_IMASK1 (Host Interrupt Mask Register 1)	(ALL)	0x8700FC07	Enable all errors to cause an interrupt.
HOST_IMASK0 (Host Interrupt Mask Register 0)	(ALL)	0x4040000F	Enable interrupts in errors, counter rollovers and framer interrupt.
LP_IMASK1 (Local Interrupt Mask Register 1)	(ALL)	0x0	Local processor not used.
LP_IMASK0 (Local Interrupt Mask Register 0)	(ALL)	0x0	Local Processor not used.
PCI Configuration Register	COMMAND	0x00D6	Enable all functions of PCI interface.
	LAT_TIMING	0x10	Latency timer = 16 clock periods.
	Bt8233_MEM_BASE	0x01	Base address of SAR device in PCI memory space is 0x0100_0000.
	ILINE	0x00	Interrupt vector = 0.
	MAX_BUR_LEN	0x10	Max burst length = 16.

15.0 Electrical/Mechanical Specifications

15.1 Timing

15.1.1 PCI Bus Interface Timing

All PCI bus interface signals are synchronous to the PCI bus clock, HCLK, except for HRST* and HINT*. Table 15-1 provides the PCI bus interface timing parameters. Figure 15-1 and Figure 15-2 illustrate this timing.

Table 15-1. PCI Bus Interface Timing Parameters *(1 of 2)*

Symbol	Parameter	Min	Max	Units
t_{cyc}	HCLK Cycle Time(1)	30	–	ns
t_{high}	HCLK High Time(1)	11	19	ns
t_{low}	HCLK Low Time(1)	11	19	ns
t_{su}	HAD Input Setup Time to HCLK(1)	7	–	ns
	HC/BE Input Setup Time to HCLK(1)	7	–	ns
	HPAR Input Setup Time to HCLK(1)	7	–	ns
	HFRAME* Input Setup Time to HCLK(1)	7	–	ns
	HIRDY* Input Setup Time to HCLK(1)	7	–	ns
	HTRDY* Input Setup Time to HCLK(1)	7	–	ns
	HSTOP* Input Setup Time to HCLK(1)	7	–	ns
	HDEVSEL* Input Setup Time to HCLK(1)	7	–	ns
	HIDSEL Input Setup Time to HCLK(1)	7	–	ns
	HGNT* Input Setup Time to HCLK(1)	10	–	ns
	HPERR* Input Setup Time to HCLK(1)	7	–	ns
t_h	Input Hold Time from HCLK–All Inputs(1)	0	–	ns

Table 15-1. PCI Bus Interface Timing Parameters *(2 of 2)*

Symbol	Parameter	Min	Max	Units
t_{val}	HCLK to HAD Valid Delay(2)	2	11	ns
	HCLK to HC/BE Valid Delay(2)	2	11	ns
	HCLK to HPAR Valid Delay(2)	2	11	ns
	HCLK to HFRAME* Valid Delay(2)	2	11	ns
	HCLK to HIRDY* Valid Delay(2)	2	11	ns
	HCLK to HSTOP* Valid Delay(2)	2	11	ns
	HCLK to HDEVSEL Valid Delay(2)	2	11	ns
	HCLK to HPERR* Valid Delay(2)	2	11	ns
	HCLK to HREQ Valid Delay(2)	2	12	ns
	HCLK to HSERR* Valid Delay(2)	2	11	ns
	HCLK to STAT Valid Delay(3)	2	20	ns
t_{on}	Float to Active Delay—All Three-state Outputs(2)	2	–	ns
t_{off}	Active to Float Delay—All Three-state Outputs(2)	–	28	ns
$t_{rst-off}$	Reset Active to Output Float Delay	–	40	ns

Notes: (1). See Figure 15-1 for waveforms and definitions.
(2). (See Figure 15-2 for waveforms and definitions. The maximum output delays are measured with a 50 pF load, and the minimum delays are measured with a 0 pF load.
(3). Applicable when STAT outputs configured as BOM cell synchronization signals.

Figure 15-1. PCI Bus Input Timing Measurement Conditions

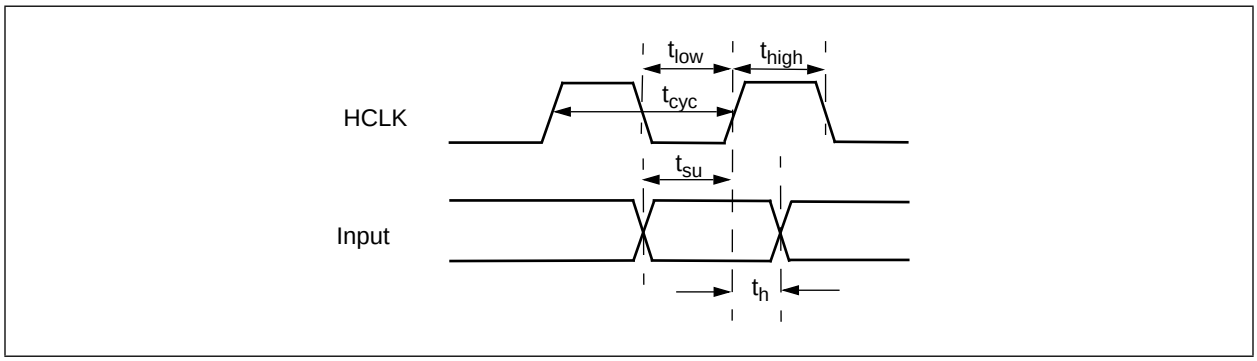
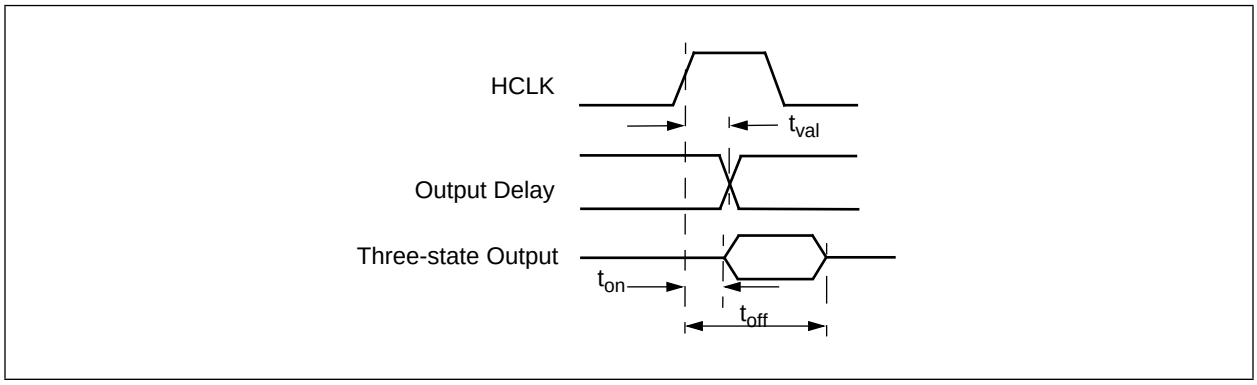


Figure 15-2. PCI Bus Output Timing Measurement Conditions



15.1.2 ATM Physical Interface Timing—UTOPIA and Slave UTOPIA

All ATM physical interface signals are synchronous to the interface clock, FRCTRL, except for TXFLAG* and RXFLAG* in the slave UTOPIA mode. Timing parameters for the UTOPIA interface are provided in Table 15-2. Table 15-3 provides the timing parameters for the slave UTOPIA interface. Timing diagrams for both interfaces are provided in Figure 15-3 and Figure 15-4.

Table 15-2. UTOPIA Interface Timing Parameters

Symbol	Parameter	Min	Max	Units
t_{cyc}	FRCTRL Cycle Time(1)	30	–	ns
t_{high}	FRCTRL High Time(1) (% of tcyc)	40	60	%
t_{low}	FRCTRL Low Time(1) (% of tcyc)	40	60	%
t_{su}	RXD Input Setup Time to FRCTRL(1)	10	–	ns
	RXPAR Input Setup Time to FRCTRL(1)	10	–	ns
	RXMARK Input Setup Time to FRCTRL(1)	10	–	ns
	RXFLAG* Input Setup Time to FRCTRL(1)	10	–	ns
	TXFLAG* Input Setup Time to FRCTRL(1)	10	–	ns
t_h	RXD Input Hold Time from FRCTRL(1)	1	–	ns
	RXPAR Input Hold Time from FRCTRL(1)	1	–	ns
	RXMARK Input Hold Time from FRCTRL(1)	1	–	ns
	RXFLAG* Input Hold Time from FRCTRL(1)	1	–	ns
	TXFLAG* Input Hold Time from FRCTRL(1)	1	–	ns
t_{val}	FRCTRL to TXD Valid Delay(2)	2	18	ns
	FRCTRL to TXPAR Valid Delay(2)	2	18	ns
	FRCTRL to TXMARK Valid Delay(2)	2	18	ns
	FRCTRL to TXEN* Valid Delay(2)	2	20	ns
	FRCTRL to RXEN* Valid Delay(2)	2	20	ns

Notes: (1). See Figure 15-3 for waveforms and definitions.
(2). See Figure 15-4 for waveforms and definitions. The output delays are measured with a 25 pF load.

Table 15-3. Slave UTOPIA Interface Timing Parameters

Symbol	Parameter	Min	Max	Units
t_{cyc}	FRCTRL Cycle Time(1)	30	–	ns
t_{high}	FRCTRL High Time(1) (% of tcyc)	40	60	%
t_{low}	FRCTRL Low Time(1) (% of tcyc)	40	60	%
t_{su}	RXD Input Setup Time to FRCTRL(1)	6	–	ns
	RXPAR Input Setup Time to FRCTRL(1)	3	–	ns
	RXMARK Input Setup Time to FRCTRL(1)	8	–	ns
	RXEN* Input Setup Time to FRCTRL(1)	10	–	ns
	TXEN* Input Setup Time to FRCTRL(1)	3	–	ns
t_h	RXD Input Hold Time from FRCTRL(1)	1	–	ns
	RXPAR Input Hold Time from FRCTRL(1)	1	–	ns
	RXMARK Input Hold Time from FRCTRL(1)	1	–	ns
	RXEN* Input Hold Time from FRCTRL(1)	1	–	ns
	TXEN* Input Hold Time from FRCTRL(1)	1	–	ns
t_{val}	FRCTRL to TXD Valid Delay(2)	2	16	ns
	FRCTRL to TXPAR Valid Delay(2)	2	16	ns
	FRCTRL to TXFLAG* Valid Delay(2)	2	20	ns
	FRCTRL to RXFLAG* Valid Delay(2)	2	20	ns
	FRCTRL to TXMARK Valid Delay(2)	2	17	ns

Notes: (1). See Figure 15-3 for waveforms and definitions.
(2). See Figure 15-4 for waveforms and definitions. The output delays are measured with a 25 pF load.

Figure 15-3. UTOPIA and Slave UTOPIA Input Timing Measurement Conditions

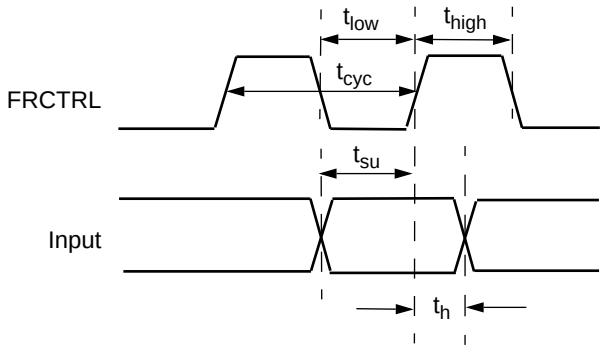
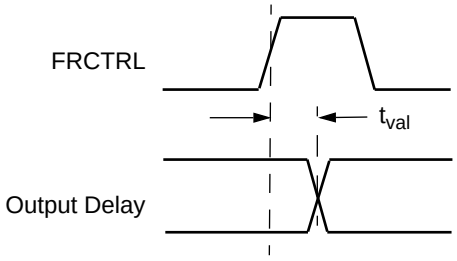


Figure 15-4. UTOPIA and Slave UTOPIA Output Timing Measurement Conditions



15.1.3 ATM Physical Interface Timing—Bt8222 Mode

All ATM physical interface signals are synchronous to the receive strobe, RXEN*, and the transmit strobe, TXEN*. Table 15-4, Figure 15-5, and Figure 15-6 provide the timing parameters for the Bt8222 interface.

Table 15-4. Bt8222 Interface Timing Parameters

Symbol	Parameter	Min	Max	Units
t_{cyc}	RXEN*/TXEN* Cycle Time(1)	50	–	ns
t_{high}	RXEN*/TXEN* High Time(1) (% of tcyc)	42	58	%
t_{low}	RXEN*/TXEN* Low Time(1) (% of tcyc)	42	58	%
t_{su}	RXD Input Setup Time to RXEN*(1)	13	–	ns
	RXPAR Input Setup Time to RXEN*(1)	10	–	ns
	RXMARK Input Setup Time to RXEN*(1)	8	–	ns
	FRCTRL Input Setup Time to RXEN*(1)	7	–	ns
	TXMARK Input Setup Time to TXEN*(1)	3	–	ns
t_h	RXD Input Hold Time from RXEN*(1)	20	–	ns
	RXPAR Input Hold Time from RXEN*(1)	20	–	ns
	RXMARK Input Hold Time from RXEN*(1)	20	–	ns
	FRCTRL Input Hold Time from RXEN*(1)	20	–	ns
	TXMARK Input Hold Time from TXEN*(1)	20	–	ns
t_{val}	TXEN* to TXD Valid Delay(2)	6	36	ns
	TXEN* to TXPAR Valid Delay(2)	6	39	ns
	TXEN* to TXFLAG* Valid Delay(2, 3)	4	21	ns
	RXEN* to RXFLAG* Valid Delay(2)	4	23	ns
t_{on}	Float to Active Delay—All 3s Outputs (TXD/TXPAR)(2)	4	16	ns
t_{off}	Active to Float Delay—All 3s Outputs (TXD/TXPAR)(2)	2	10	ns

Notes: (1). See Figure 15-5 for waveforms and definitions.
(2). See Figure 15-6 for waveforms and definitions. The output delays are measured with a 50 pF load.
(3). TXFLAG* going inactive is asynchronous to TXEN*.

Figure 15-5. Bt8222 Input Timing Measurement Conditions

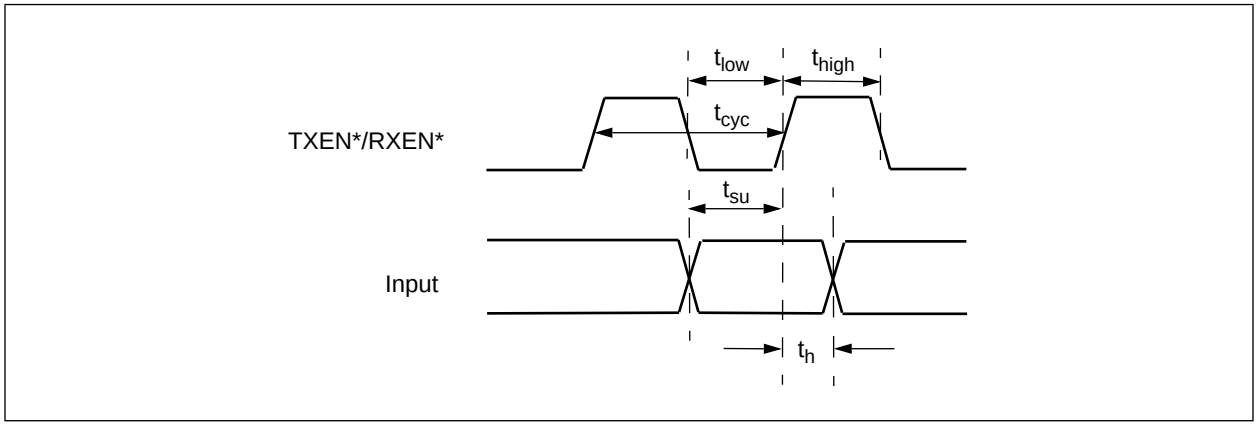
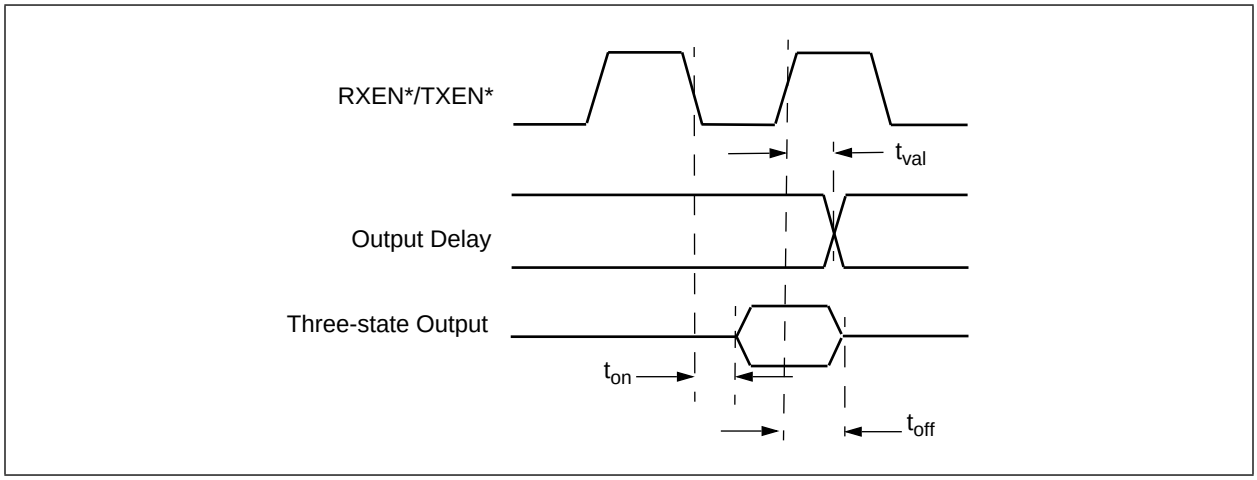


Figure 15-6. Bt8222 Output Timing Measurement Conditions



15.1.4 System Clock Timing

The system clock timing consists of the CLK2X input and the SYSCLK and CLKD3 outputs. The CLK2X input and SYSCLK outputs are used to generate the internal and external system clocks as well as SRC shared memory and processor read and write timing. The CLKD3 output may be used as the ATM Physical Interface clock. There is no defined skew relationship between the CLK2X input and SYSCLK and CLKD3 outputs. Table 15-5 specifies the timing parameters of the three clocks. Figure 15-7 and Figure 15-8 illustrates this timing.

Table 15-5. System Clock Timing

Symbol	Parameter	Min	Max	Units
Input Clocks(1)				
t_{cf}	CLK2X Frequency	0	66	MHZ
t_c	CLK2X Period	15.15		ns
t_{cd}	CLK2X Duty Cycle	40	60	%
t_{cr}	CLK2X Rise Time	0	6	ns
t_{cf}	CLK2X Fall Time	0	6	ns
Output Clocks(2)				
t_{sf}	SYSCLK Frequency	$t_{CF}/2$		MHz
t_s	SYSCLK Period	$2t_C$		ns
t_{sh}	SYSCLK High Time	$(t_S/2) – 2$	$(t_S/2) + 2$	ns
t_{sl}	SYSCLK Low Time	$(t_S/2) – 2$	$(t_S/2) + 2$	ns
t_{sr}	SYSCLK Rise Time	1	4	ns
t_{sf}	SYSCLK Fall Time	1	4	ns
t_{df}	CLKD3 Frequency	$t_{CF}/3$		MHz
t_d	CLKD3 Period	$3t_C$		
t_{dh}	CLKD3 High Time	$(t_D/2) – 2$	$(t_D/2) + 2$	ns
t_{dl}	CLKD3 Low Time	$(t_D/2) – 2$	$(t_D/2) + 2$	ns
t_{dr}	CLKD3 Rise Time	1	4	ns
t_{df}	CLKD3 Fall Time	1	4	ns

Notes: (1). See Figure 15-7 for waveforms and definitions.
(2). See Figure 15-8 for waveforms and definitions. The outputs are measured with a load of 35 pf.

Figure 15-7. Input System Clock Waveform

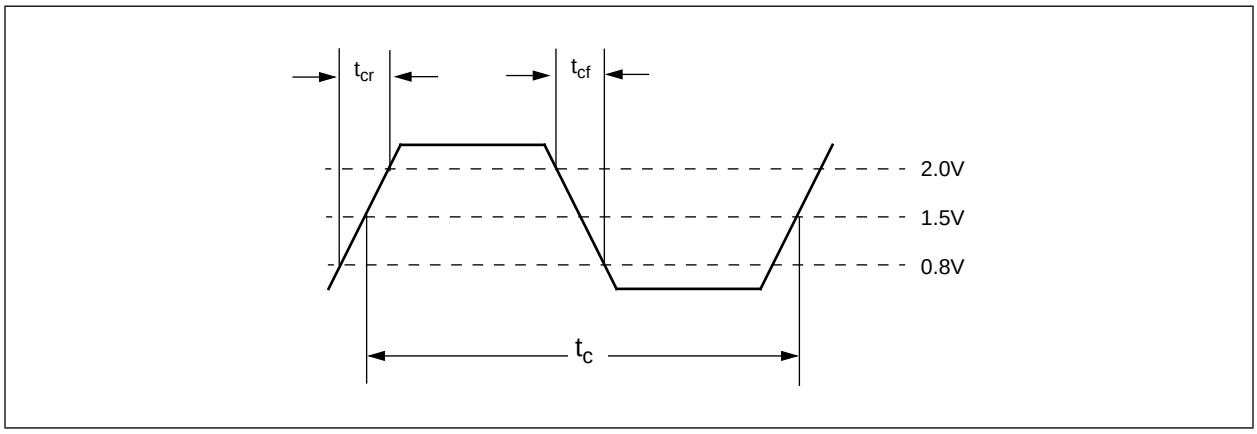
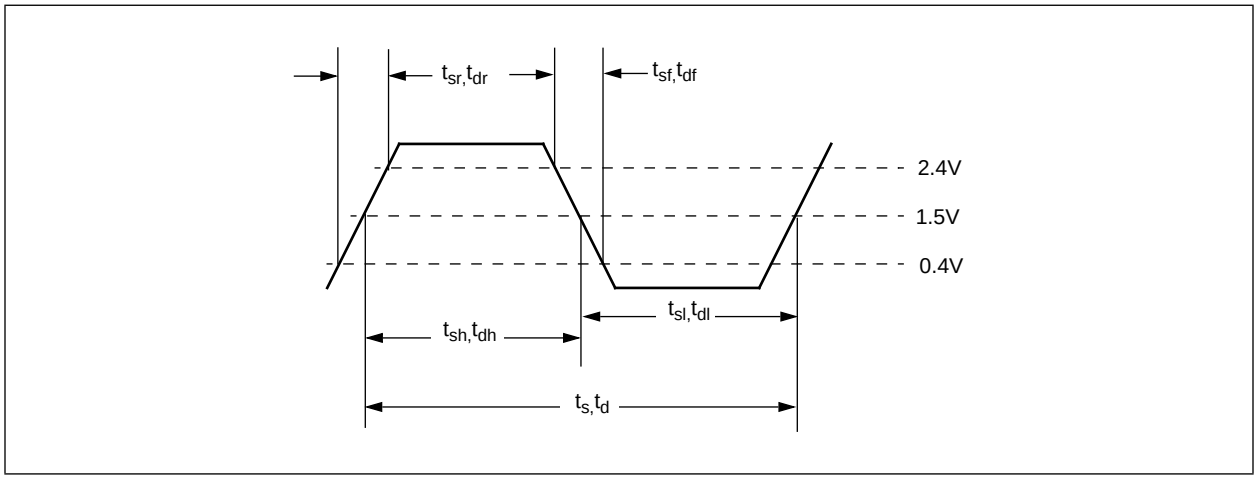


Figure 15-8. Output System Clock Waveform



15.1.5 Bt8233 Memory Interface Timing

Memory access times and other timing requirements are specified at three typical implementations of 1, 2, and 4 banks of by_8 SRAM. Table 15-6 gives the number of loads per bank for the different SRAM organizations. Table 15-7 gives the capacitive loading used in the timing specifications given for the three typical implementations. Bt8233 Memory Interface Timing is given in Table 15-8. See Section 9.2 for details of memory bank organization.

Table 15-6. SRAM Organization Loading Dependencies

Signal	Loads/Bank(1)		
	by_16 SRAM	by_8 SRAM	by_4 SRAM
LADDR[18:0](1)	2	4	8
LDATA[31:0](1)	1	1	1
MWR*(1)	2	N/A	N/A
MOE*(1)	2	4	8
MCSx*(1, 2)	2	4	8
MWEx*(1)	1	1	2

Notes: (1). Typical input loading for SRAM is 7 pf. For exact values, consult the SRAM databook.
(2). Only connected to one bank by definition.

Table 15-7. SRC Shared Memory Output Loading Conditions

Signal	4 banks of by_8 SRAM	2 banks of by_8 SRAM	1 bank of by_8 SRAM	Units
Memory Interface Loading(1)				
LADDR[18:0](1)	150	100	50	pf
MOE*	150	100	50	pf
MWR*(2)	–	50	35	pf
LDATA[31:0]	50	35	25	pf
MWE[3:0]*	50	35	25	pf
MCS[3:0]*	50	35	25	pf

Notes: (1). In general, the LADDR loading is the most critical parameter, one bank of by_8 SRAM has the same address loading as 2 banks of by_16 and 1/2 bank of by_4 SRAM. For example, use the timing from the 2 banks of by_8 column for 4 banks of by_16 SRAM, or 1 bank of by_4 SRAM.
(2). For by_16 SRAM, the WE* input has the same loading as the address bus and OE*; therefore, 16 loads specified by the 4 banks of by_8 is not applicable, since the maximum number of by_16 SRAMs supported is 8.

Table 15-8. Bt8233 Memory Interface Timing

Symbol	Parameter	4 banks of by_8 SRAM		2 banks of by_8 SRAM		1 bank of by_8 SRAM		Units
		Min	Max	Min	Max	Min	Max	
Memory Read Timing								
t_{rc}	READ Cycle Time(2)	T		T		T		ns
t_{aov}	LADDR[18:0] Output Valid(1)	1	10	1	8	1	7	ns
t_{mcsov}	MCS[3:0]* Output Valid(1)	1	10	1	8	1	7	ns
t_{mweov}	MWE* Byte Enables Output Valid(1) (RAMMODE = 1)	1	10	1	8	1	7	ns
t_{moel}	MOE* Low(1)		17		16		15	ns
t_{moeh}	MOE* High(1)		10		8		7	ns
t_{dod}	LDATA Output Disable(1)		7		6		6	ns
t_{dis}	LDATA Input Setup(1)	5		5		5		ns
t_{dh}	LDATA Input Hold(1)	0		0		0		ns
t_{doe}	LDATA Driven by SAR(1)	15		14		14		ns
Memory Write Timing								
t_{wc}	WRITE Cycle Time(2)	T		T		T		ns
t_{w}	MWR*, MWE[3:0]* Width(2, 4)	T/2		T/2		T/2		ns
t_{aov}	LADDR[18:0] Output Valid(4)	1	10	1	8	1	7	ns
t_{mcsov}	MCS[3:0]* Output Valid(4)	1	10	1	8	1	7	ns
t_{mweov}	MWE* Byte Enables Output Valid(1) (RAMMODE = 1)	1	10	1	8	1	7	ns
t_{dov}	LDATA Output Valid(4)	1	10	1	10	1	10	ns
t_{mwel}	MWE[3:0]* Low(4)		16		16		16	ns
t_{mweh}	MWE[3:0]* High(4)		1		1		1	ns
t_{mwrl}	MWR[3:0]* Low(4)		16		16		16	ns
t_{mwrh}	MWR[3:0]* High(4)		1		1		1	ns

Notes: (1). See Figure 15-9 for waveforms.
(2). T = t_S for single cycle memory (no wait states), T = $2t_S$ for two cycle memory, (one wait state).
(3). Insert one clock cycle for two cycle memory (one wait state).
(4). See Figure 15-10 for waveforms.
(5). SYSCLK shown for reference only.

Figure 15-9. Bt8233 Memory Read Timing

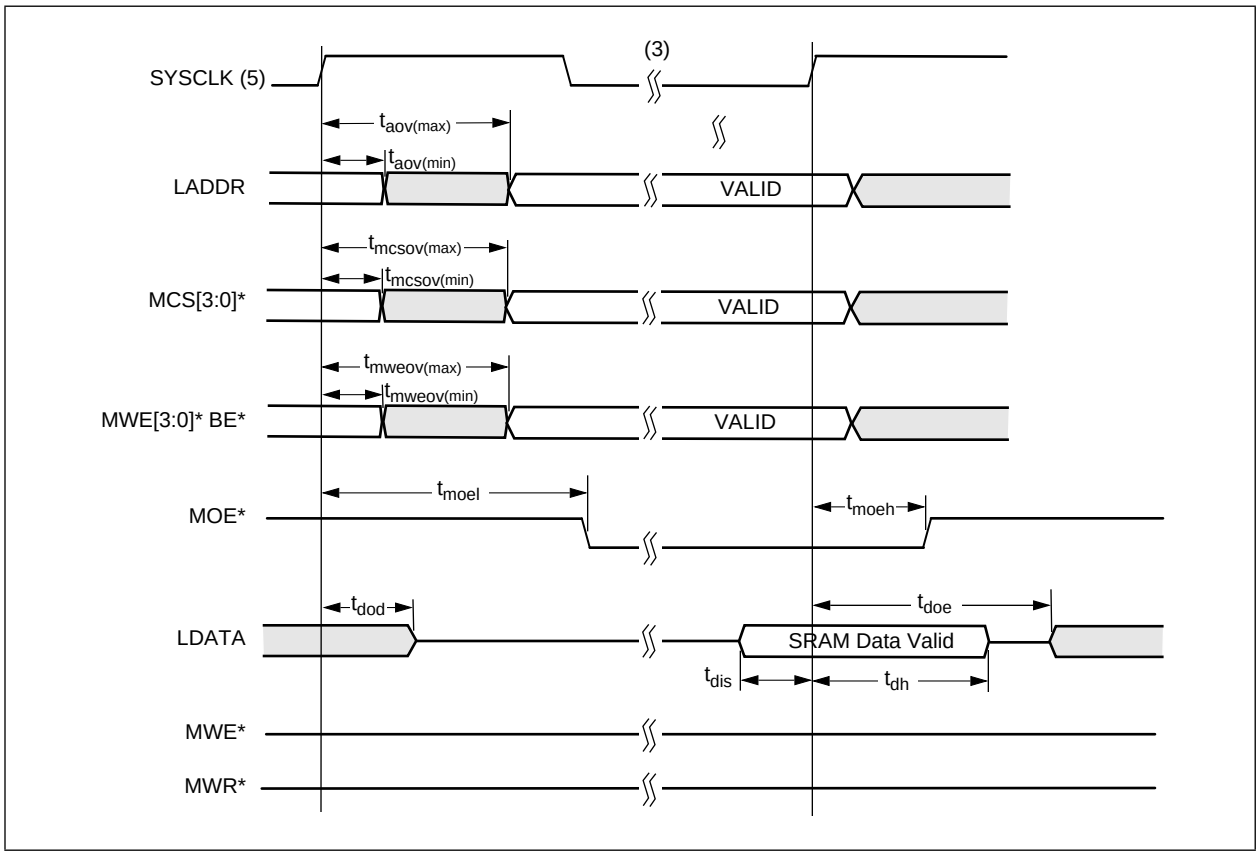
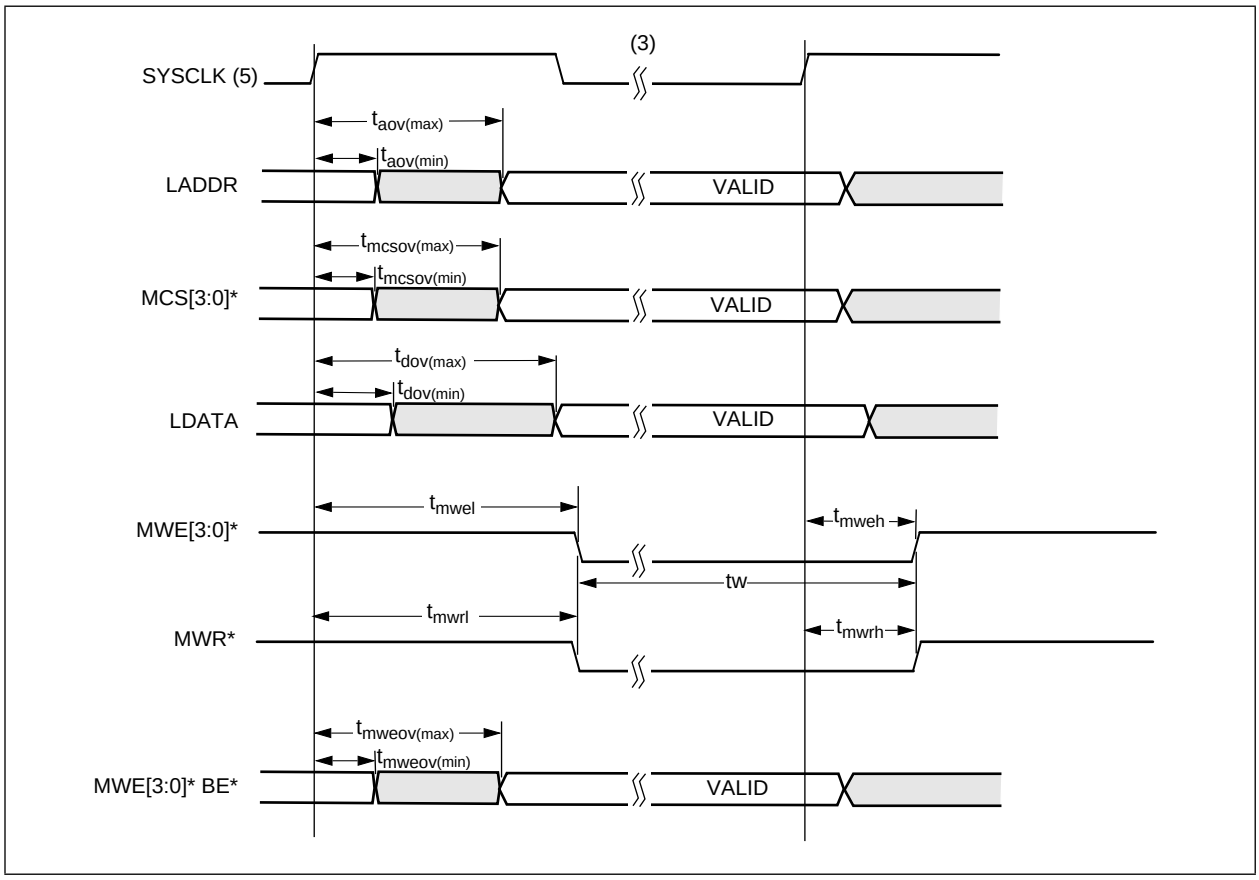


Figure 15-10. Bt8233 Memory Write Timing



15.1.6 PHY Interface Timing (Standalone Mode)

The standalone mode of operation is entered when the PROCMODE input is at a logic high indicating that no local processor is present. In this mode, the Bt8233 changes its memory map to include the ATM physical interface device. The interface is fully synchronous to SYSCLK and is designed to interface directly to the Bt8222 ATM Receiver/Transmitter. Timing is given in Table 15-9, Figure 15-11, and Figure 15-12. See Section 10.6 for details.

Table 15-9. PHY Interface Timing (PROCMODE = 1)

Symbol	Parameter	Min	Max	Units
Synchronous Inputs				
t_{is}	PWAIT* Input Setup(1)	14		ns
	LDATA Input Setup(1)	5		ns
t_{ih}	PWAIT* Input Hold(1)	0		ns
	LDATA Input Hold(1)	0		ns
Synchronous Outputs				
t_{ov}	PRDY* Output Valid Delay(2)	5	15	ns
	PAS Output Valid Delay(2)	5	15	ns
	PCS* Output Valid Delay(2)	5	15	ns
	PBLAST* Output Valid Delay(2)	5	15	ns
	PWNR Output Valid Delay(2)	5	15	ns
	LDATA* Output Valid Delay(2)	1	10	ns
	LADDR Output Valid Delay(2)	1	10	ns

Notes: (1). See Figure 15-11 for waveforms and definitions.
(2). See Figure 15-12 for waveforms and definitions. The outputs are measured with a load of 35 pf.

Figure 15-11. Synchronous PHY Interface Input Timing

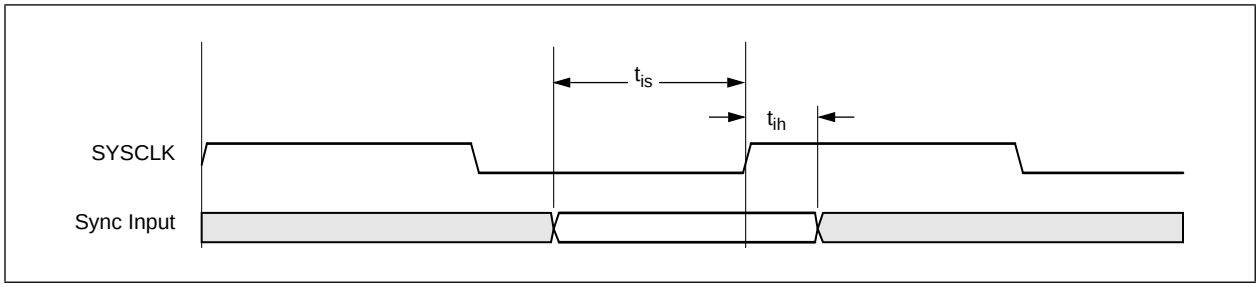
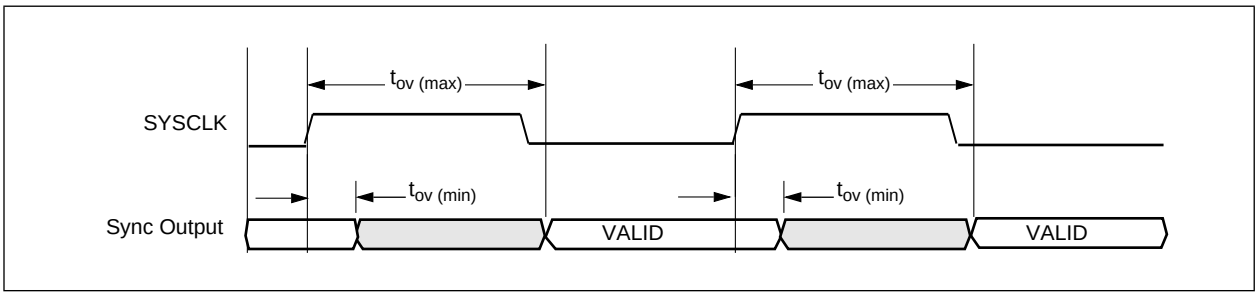


Figure 15-12. Synchronous PHY Interface Output Timing



15.1.7 Local Processor Interface Timing

Timing for the local processor interface can be broken into two sections. The first is the synchronous to SYSCLK interface of processor control signals to and from the Bt8233. The second is the interface to the SRC shared SRAM memory and the Bt8233 control and status registers. This interface involves both SRAM and transceiver and buffer timing parameters that are not specified and are left up to the system designer.

All of the synchronous interface signals are inputs to the Bt8233 except for the system clock (SYSCLK) and the ready output of the Bt8233 (PRDY*). The output loading of these signals is 35 pf for the timing given in Table 15-10. Memory Interface Timing is given in Table 15-11.

Table 15-10. Synchronous Processor Interface Timing (1 of 2)

Symbol	Parameter	Min	Max	Units
	Synchronous Inputs			
t_{is}	PCS* Input Setup(1)	8		ns
	PAS* Input Setup(1)	10		ns
	PBLAST* Input Setup(1)	10		ns
	PWAIT* Input Setup(1)	10		ns
	PADDR[1,0] Input Setup(1)	10		ns
	PBSEL[1,0] Input Setup(1)	8		ns
	PBE[3:0]* Input Setup(1)	8		ns
	PWNR Input Setup(1)	10		ns

Table 15-10. Synchronous Processor Interface Timing *(2 of 2)*

Symbol	Parameter	Min	Max	Units
t_{ih}	PCS* Input Hold(1)	0		ns
	PAS* Input Hold(1)	0		ns
	PBLAST* Input Hold(1)	0		ns
	PWAIT* Input Hold(1)	0		ns
	PADDR[1,0] Input Hold(1)	0		ns
	PBSEL[1,0] Input Hold(1)	0		ns
	PBE[3:0]* Input Hold(1)	0		ns
	PWNR Input Hold(1)	0		ns
Synchronous Outputs				
t_{ov}	PRDY* Output Valid Delay(2)	5	15	ns

Notes: (1). See Figure 15-13 for waveforms and definitions.
(2). See Figure 15-14 for waveforms and definitions. The outputs are measured with a load of 35 pf.

Figure 15-13. Synchronous Local Processor Input Timing

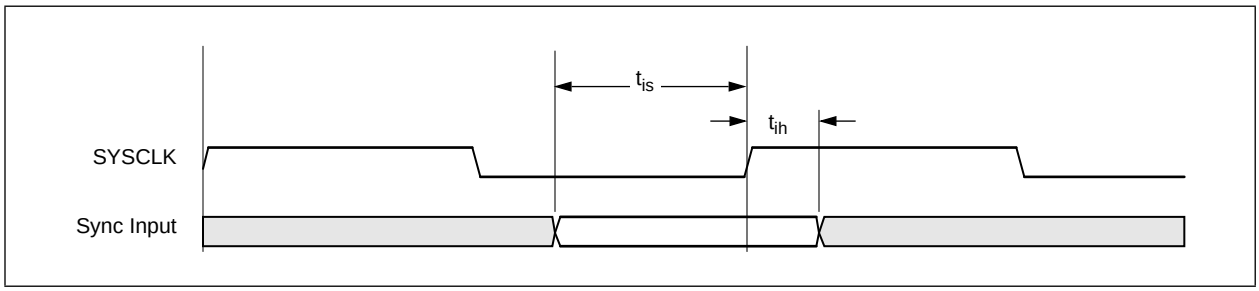


Figure 15-14. Synchronous Local Processor Output Timing

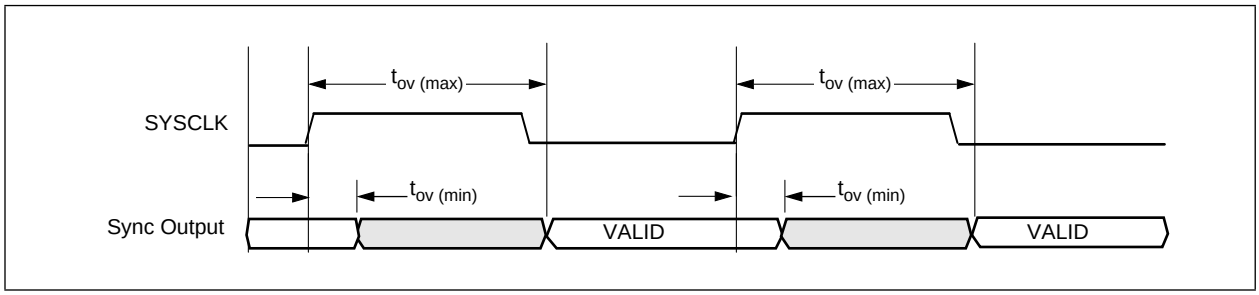


Table 15-11. Local Processor Memory Interface Timing

Symbol	Parameter	Min	Max	Units
t_{pdaenl}	SYSCLK to PDAEN* Low, Bus Recovery Cycle(1)		12	ns
t_{pdaenh}	SYSCLK to PDAEN* High, Bus Recovery Cycle(1)	2		ns
t_{lod}	LADDR[18:2], LDATA Output Disable to PDAEN* Low, Bus Recovery Cycle(1)	4		ns
t_{loe}	PDAEN* High to LADDR[18:2], LDATA Output Enable, Bus Recovery Cycle(1)	9		ns
t_{mcs}	SYSCLK to MCS*[3:0] Valid(1)	1	6	ns
t_{lav}	SYSCLK to LADDR[1,0] Valid(1, 2)	1	7	ns
t_{oel}	MOE* Active from SYSCLK(1, 3)	8	20	ns
t_{oeh}	MOE* Inactive from SYSCLK(1, 3)	1	10	ns
t_{wl}	MWR*, MWE[3:0] Active from SYSCLK(1, 3)		16	ns
t_{wh}	MWR*, MWE[3:0]* Inactive to SYSCLK(1, 3)		1	ns
t_{be}	MWE[3:0]* Byte Enables Valid from SYSCLK (RAMMODE = 1)(1)	1	7	ns
t_{crd}	CSR Read Data Output Valid	2	20	ns
t_{cwds}	CSR Write Data Setup to SYSCLK	8		ns
t_{cwdh}	CSR Write Data Hold from SYSCLK	0		ns
t_{las}	LADDR Setup to SYSCLK	12		ns

Notes: (1). See Figure 15-15 and Figure 15-16 for waveforms and definitions.
(2). t_{lav} is valid for second and subsequent accesses during burst transfers. See functional timing diagrams.
(3). In the case of two-cycle memory, or when inserting wait states by PWAIT*, MOE*, MWE[3:0]*, and MWR* are extended across 2 or more clock cycles with the same relative timing to SYSCLK, see the functional timing diagrams in Section 10.6 .

Figure 15-15. Local Processor Read Timing

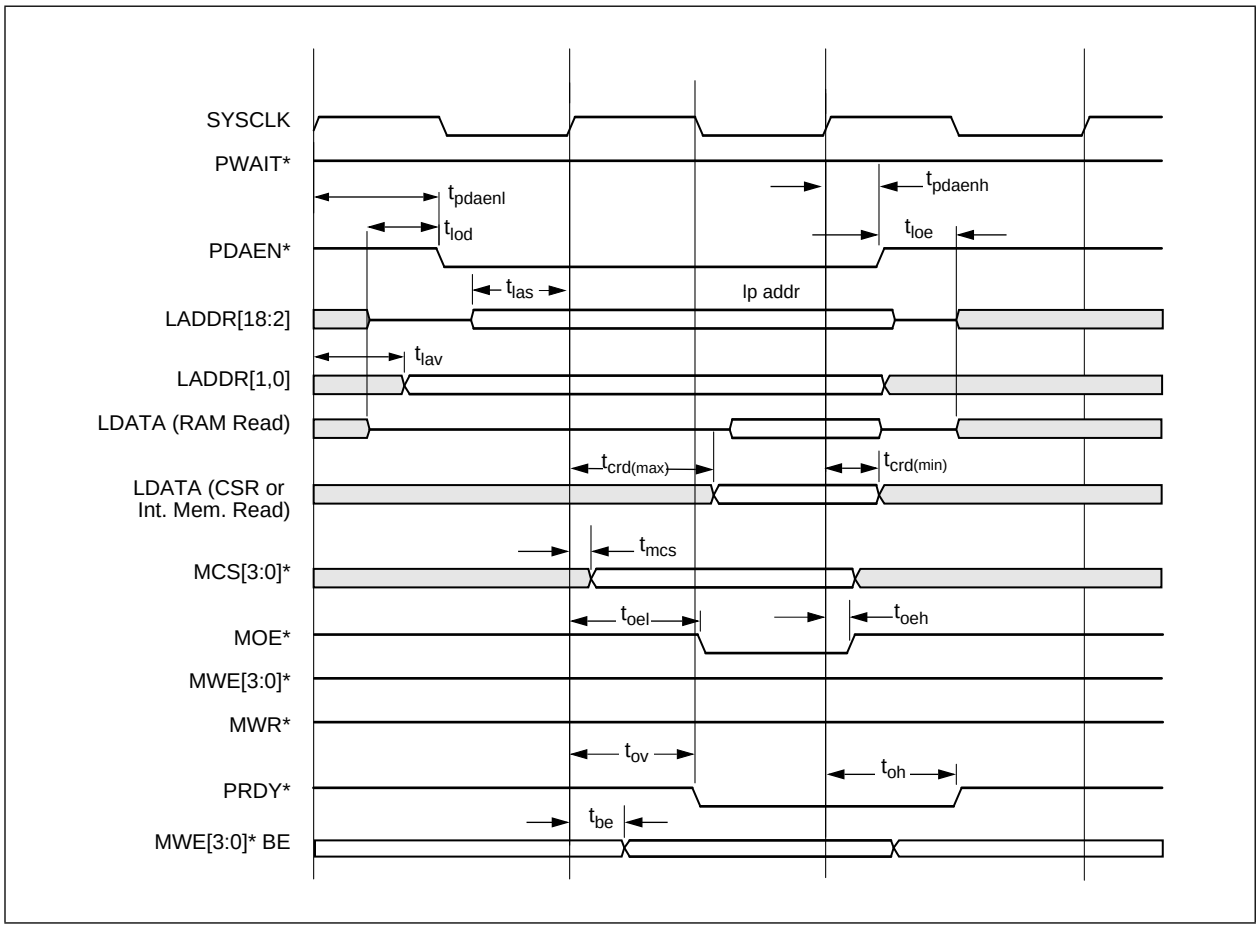
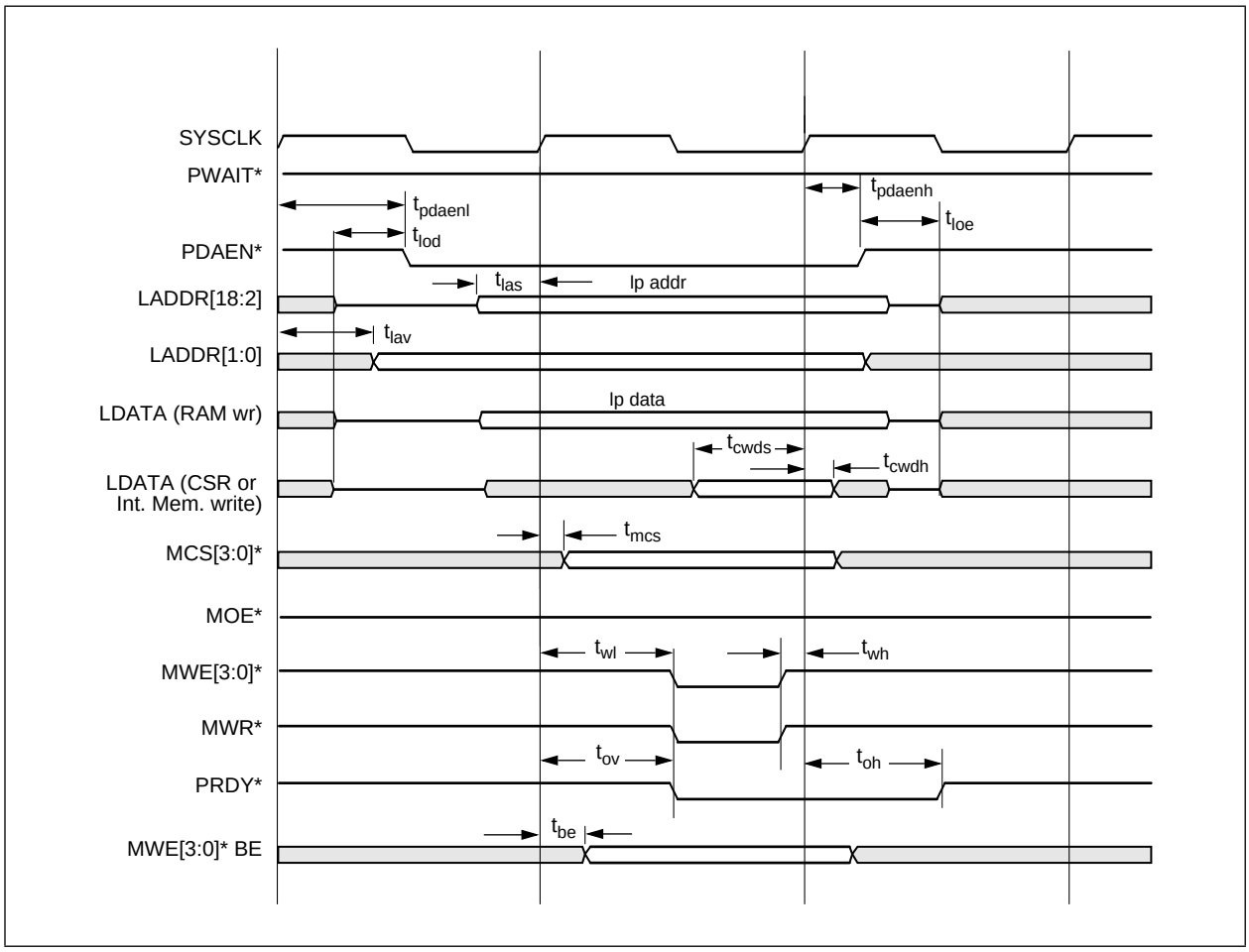


Figure 15-16. Local Processor Write Timing



15.2 Absolute Maximum Ratings

Stresses above those listed as absolute maximum ratings (Table 15-12) may cause permanent damage to the device. This is a stress rating only. Functional operation of the device at these or any other conditions above those indicated in other sections of this document is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. This device should be handled as an ESD-sensitive device. Voltage on any signal pin that exceeds the power supply voltage by more than +0.5 V can induce destructive latchup.

Table 15-12. Absolute Maximum Ratings

Parameter	Value	Unit
Supply Voltage	–0.5 to +6.0	V
Input Voltage	–0.5 to (Vdd + 0.5)	V
Output Voltage	–0.5 to (Vdd + 0.5)	V
Operating Temperature—No Air Flow(1)	TBD	C
Operating Temperature—400 Linear Feet per Minute (1)	TBD	C
Storage Temperature	–40 to 125	C
Operating Supply Voltage	4.75 to 5.25	V
Maximum Current @ Max Clock Frequencies(1)	TBD	mA
Note (1): This is preliminary information pending full device characterization.		

15.3 DC Characteristics

Table 15-13 specifies the DC Characteristics.

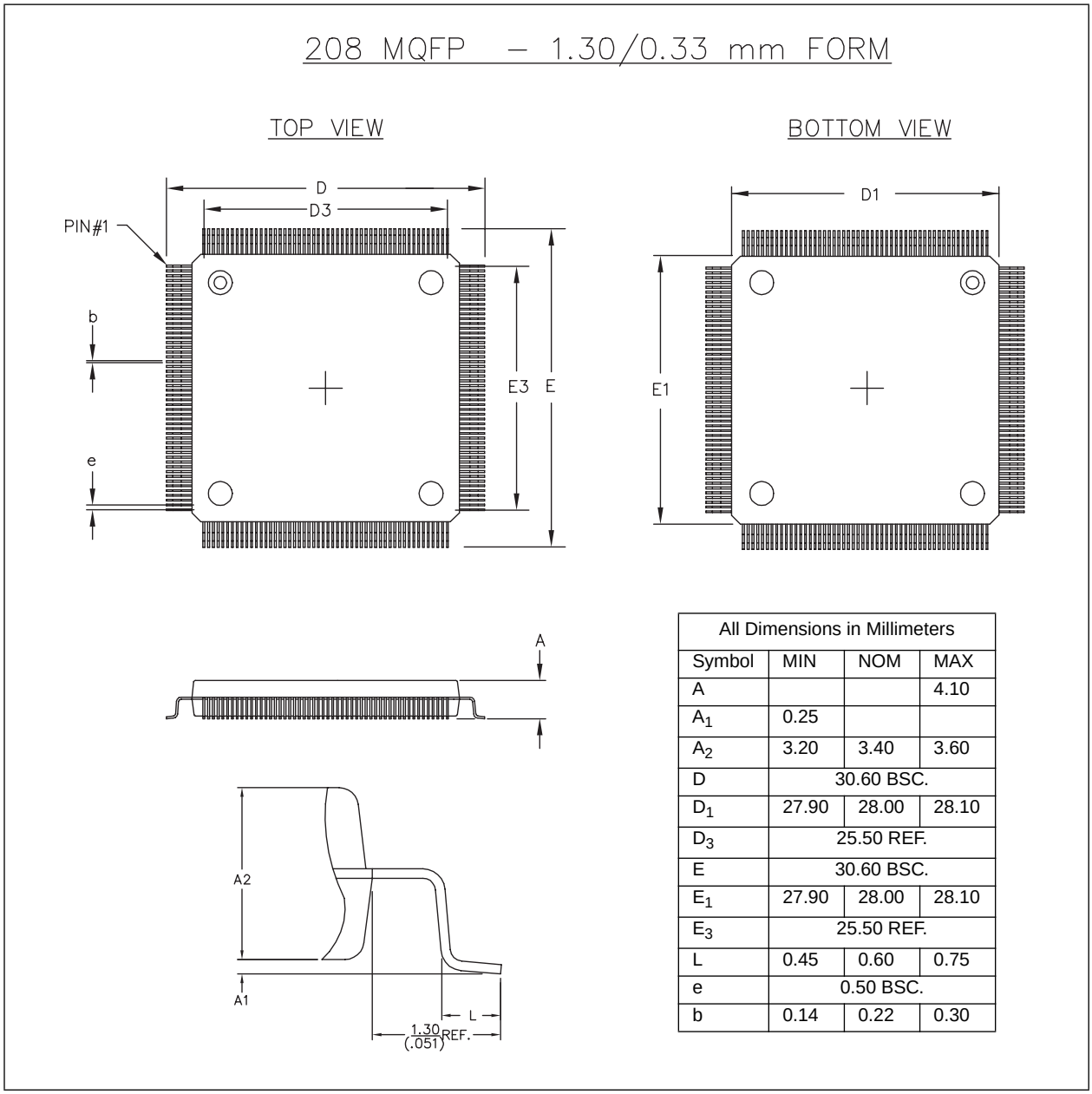
Table 15-13. DC Characteristics

Parameter	Conditions	Min	Max	Unit
Output Voltage High	I_{OH} = –0.8 mA	Vdd –0.1		V
Output Voltage Low	I_{OL} = 0.8 mA		Vss + 0.1	V
Input Voltage High		2.0	Vdd + 0.5	V
Input Voltage Low		–0.5	0.8	V
Input Leakage Current	Vin = Vdd or GND	–10	10	uA
Three-state Output Leakage Current	VOUT = Vdd or GND	–10	10	uA
Input Capacitance			7	pF
Output Capacitance			7	pF
Note: All outputs are CMOS drive levels and can be used with CMOS or TTL logic.				

15.4 Mechanical Specifications

The Bt8233 208-pin PQFP package is illustrated in Figure 15-17. Figure 15-18 is a pinout configuration of the Bt8233, and a pin listing is given in Table 15-14.

Figure 15-17. 208-Pin Plastic Quad Flat Pack (PQFP)



All Dimensions in Millimeters			
Symbol	MIN	NOM	MAX
A			4.10
A_1	0.25		
A_2	3.20	3.40	3.60
D		30.60 BSC.	
D_1	27.90	28.00	28.10
D_3		25.50 REF.	
E		30.60 BSC.	
E_1	27.90	28.00	28.10
E_3		25.50 REF.	
L	0.45	0.60	0.75
e		0.50 BSC.	
b	0.14	0.22	0.30

Figure 15-18. Bt8233 Pinout Configuration

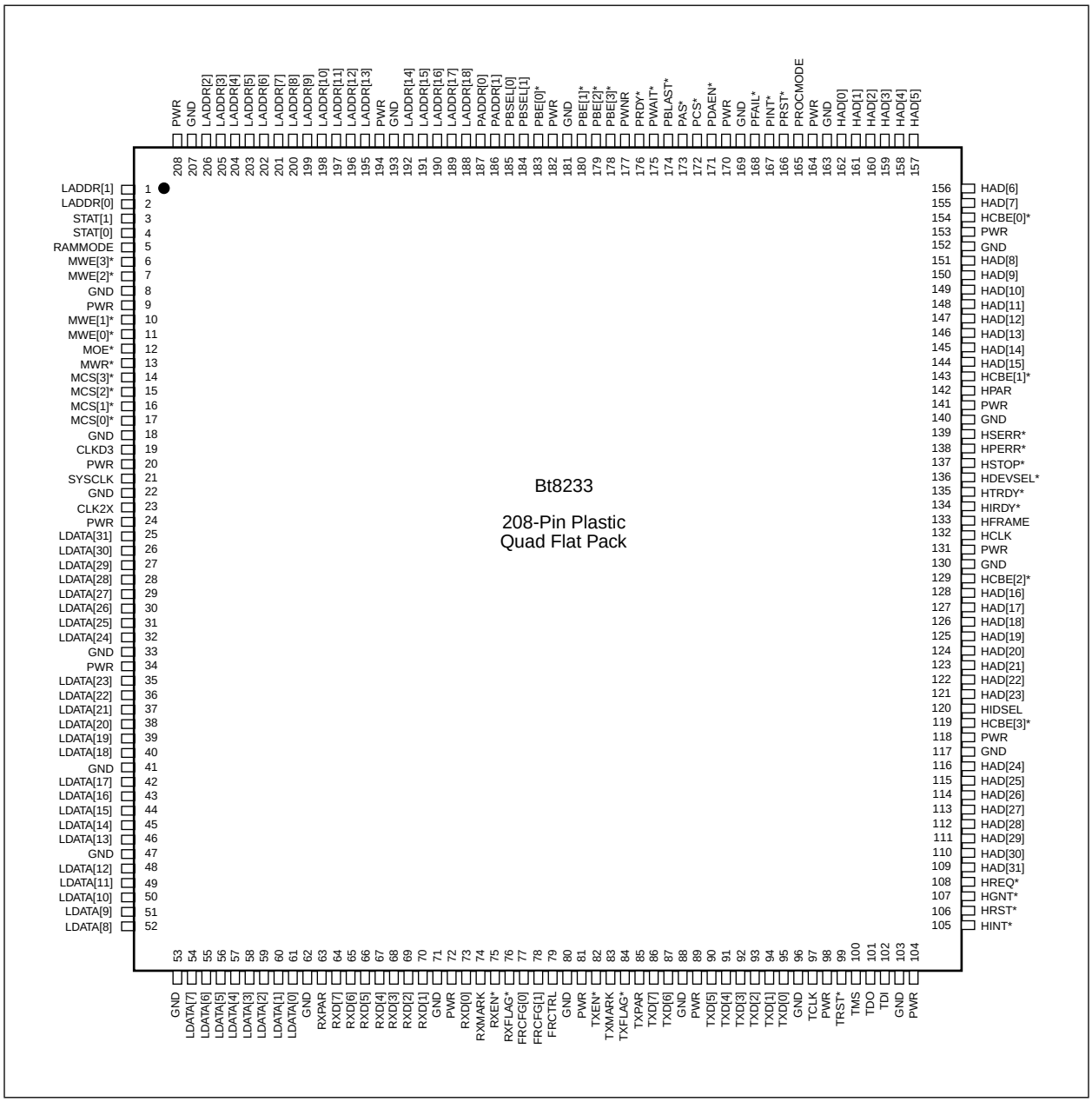


Table 15-14. Pin Descriptions *(1 of 2)*

Pin	Pin Label	I/O	Pin	Pin Label	I/O	Pin	Pin Label	I/O
1	LADDR[1]	I/O	36	LDATA[22]	I/O	71	GND	–
2	LADDR[0]	I/O	37	LDATA[21]	I/O	72	PWR	–
3	STAT[1]	O	38	LDATA[20]	I/O	73	RXD[0]	I
4	STAT[0]	O	39	LDATA[19]	I/O	74	RXMARK	I
5	RAMMODE	I	40	LDATA[18]	I/O	75	RXEN*	I/O
6	MWE[3]*	O	41	GND	–	76	RXFLAG*	I/O
7	MWE[2]*	O	42	LDATA[17]	I/O	77	FRCFG[0]	I
8	GND	–	43	LDATA[16]	I/O	78	FRCFG[1]	I
9	PWR	–	44	LDATA[15]	I/O	79	FRCTRL	I
10	MWE[1]*	O	45	LDATA[14]	I/O	80	GND	–
11	MWE[0]*	O	46	LDATA[13]	I/O	81	PWR	–
12	MOE*	O	47	GND	–	82	TXEN*	I/O
13	MWR*	O	48	LDATA[12]	I/O	83	TXMARK	I/O
14	MCS[3]*	O	49	LDATA[11]	I/O	84	TXFLAG*	I/O
15	MCS[2]*	O	50	LDATA[10]	I/O	85	TXPAR	O
16	MCS[1]*	O	51	LDATA[9]	I/O	86	TXD[7]	O
17	MCS[0]*	O	52	LDATA[8]	I/O	87	TXD[6]	O
18	GND	–	53	GND	–	88	GND	–
19	CLKD3	O	54	LDATA[7]	I/O	89	PWR	–
20	PWR	–	55	LDATA[6]	I/O	90	TXD[5]	O
21	SYSCLK	O	56	LDATA[5]	I/O	91	TXD[4]	O
22	GND	–	57	LDATA[4]	I/O	92	TXD[3]	O
23	CLK2X	I	58	LDATA[3]	I/O	93	TXD[2]	O
24	PWR	–	59	LDATA[2]	I/O	94	TXD[1]	O
25	LDATA[31]	I/O	60	LDATA[1]	I/O	95	TXD[0]	O
26	LDATA[30]	I/O	61	LDATA[0]	I/O	96	GND	–
27	LDATA[29]	I/O	62	GND	–	97	TCLK	I
28	LDATA[28]	I/O	63	RXPAR	I	98	PWR	–
29	LDATA[27]	I/O	64	RXD[7]	I	99	TRST*	I
30	LDATA[26]	I/O	65	RXD[6]	I	100	TMS	I
31	LDATA[25]	I/O	66	RXD[5]	I	101	TDO	O
32	LDATA[24]	I/O	67	RXD[4]	I	102	TDI	I
33	GND	–	68	RXD[3]	I	103	GND	–
34	PWR	–	69	RXD[2]	I	104	PWR	–
35	LDATA[23]	I/O	70	RXD[1]	I	105	HINT*	OD

Table 15-14. Pin Descriptions *(2 of 2)*

Pin	Pin Label	I/O	Pin	Pin Label	I/O	Pin	Pin Label	I/O
106	HRST*	I	141	PWR	–	176	PRDY*	O
107	HGNT*	I	142	HPAR	I/O	177	PWNR	I/O
108	HREQ*	O	143	HC/BE[1]*	I/O	178	PBE[3]*	I
109	HAD[31]	I/O	144	HAD[15]	I/O	179	PBE[2]*	I
110	HAD[30]	I/O	145	HAD[14]	I/O	180	PBE[1]*	I
111	HAD[29]	I/O	146	HAD[13]	I/O	181	GND	–
112	HAD[28]	I/O	147	HAD[12]	I/O	182	PWR	–
113	HAD[27]	I/O	148	HAD[11]	I/O	183	PBE[0]*	I
114	HAD[26]	I/O	149	HAD[10]	I/O	184	PBSEL[1]	I
115	HAD[25]	I/O	150	HAD[9]	I/O	185	PBSEL[0]	I
116	HAD[24]	I/O	151	HAD[8]	I/O	186	PADDR[1]	I
117	GND	–	152	GND	–	187	PADDR[0]	I
118	PWR	–	153	PWR	–	188	LADDR[18]	I/O
119	HCBE[3]*	I/O	154	HCBE[0]*	I/O	189	LADDR[17]	I/O
120	HIDSEL	I	155	HAD[7]	I/O	190	LADDR[16]	I/O
121	HAD[23]	I/O	156	HAD[6]	I/O	191	LADDR[15]	I/O
122	HAD[22]	I/O	157	HAD[5]	I/O	192	LADDR[14]	I/O
123	HAD[21]	I/O	158	HAD[4]	I/O	193	GND	–
124	HAD[20]	I/O	159	HAD[3]	I/O	194	PWR	–
125	HAD[19]	I/O	160	HAD[2]	I/O	195	LADDR[13]	I/O
126	HAD[18]	I/O	161	HAD[1]	I/O	196	LADDR[12]	I/O
127	HAD[17]	I/O	162	HAD[0]	I/O	197	LADDR[11]	I/O
128	HAD[16]	I/O	163	GND	–	198	LADDR[10]	I/O
129	HCBE[2]*	I/O	164	PWR	–	199	LADDR[9]	I/O
130	GND	–	165	PROCMODE	I	200	LADDR[8]	I/O
131	PWR	–	166	PRST*	O	201	LADDR[7]	I/O
132	HCLK	I	167	PINT*	OD	202	LADDR[6]	I/O
133	HFRAME*	I/O	168	PFAIL*	I	203	LADDR[5]	I/O
134	HIRDY*	I/O	169	GND	–	204	LADDR[4]	I/O
135	HTRDY*	I/O	170	PWR	–	205	LADDR[3]	I/O
136	HDEVSEL*	I/O	171	PDAEN*	I/O	206	LADDR[2]	I/O
137	HSTOP*	I/O	172	PCS*	I/O	207	GND	–
138	HPERR*	I/O	173	PAS*	I/O	208	PWR	
139	HSERR*	OD	174	PBLAST*	I/O			
140	GND	–	175	PWAIT*	I			

Please mail or fax this form to:
Rockwell Semiconductor Systems
Manager, Technical Publications Dept.
(619) 597–4338

READER RESPONSE PAGE

We welcome your evaluation of this publication. Your comments and suggestions will help us make improvements to all current and future documents. Please rate the following characteristics of the publication you received:

Poor Excellent

1. Overall usefulness of the publication.
❍ ❍ ❍ ❍ ❍ ❍
2. Organization of the publication.
❍ ❍ ❍ ❍ ❍ ❍
3. Completeness and thoroughness of the material.
❍ ❍ ❍ ❍ ❍ ❍
4. Clarity and accuracy of the material.
❍ ❍ ❍ ❍ ❍ ❍
5. Usefulness of the diagrams and illustrations.
❍ ❍ ❍ ❍ ❍ ❍
6. Quantity of diagrams and Illustrations.
❍ ❍ ❍ ❍ ❍ ❍
7. Ease of finding specific information.
❍ ❍ ❍ ❍ ❍ ❍
8. Publication page layout and format.
❍ ❍ ❍ ❍ ❍ ❍

9. I received the publication(s): (Please mark one).
❑ Within one week after ordering it
❑ Within two weeks after ordering it
❑ Within three or more weeks after ordering it

10. The publication delivery time was: (Please mark one).
❑ Earlier than I needed for my needs
❑ Just in time for my needs
❑ Too late for my needs

11. In what format(s) would you prefer to receive future documents? (Check all that apply.)
❑ Printed books
❑ Internet/PDF delivery
❑ CD-ROM
❑ Other: _____

12. How would you rank the importance of the following publication features? (1=Very important)
____ Publication design and layout
____ Accuracy of content
____ Detailed diagrams and illustrations
____ Receiving the publication in a timely manner

13. We would like to hear your thoughts on how we could improve this publication. Please write your comments or suggestions below:

_____

_____

_____

Thank you for taking the time to evaluate this publication.

Document title:_____

Revision No.: _____

Your Name: _____

Title: _____

Company: _____

E-mail: _____

Phone: _____

Because Communication Matters™

URL Address
www.nb.rockwell.com

E-mail Address
literature@nb.rockwell.com

For more information:
Call 1-800-854-8099

International information:
Call 1-714-221-6966

Headquarters
Rockwell Semiconductor Systems Inc.
4311 Jamboree Road, P.O. Box C
Newport Beach, CA 92658-8902
Phone: (714) 221-4600
Fax: (714) 221-6375

European Headquarters
Rockwell Semiconductor Systems S.A.R.L.
Les Taissounieres B1
1680 Route des Dolines BP 283
06905 Sophia Antipolis Cedex
France
Phone: 00.33.4.93.00.33.35
Fax: 00.33.4.93.00.33.03

Asia Pacific Headquarters
1, Kallang Sector, #07-04/06
Kolam Ayer Industrial Park
Singapore, 1334
Phone: 011-65-841-3801
Fax: 011-65-841-3802

US Southwest
Phone: (714) 222-9119
Fax: (714) 222-0620

US Los Angeles
Phone: (805) 376-0559
Fax: (805) 376-8180

US South Central
Phone: (972) 479-9310
Fax: (972) 479-9317

US Southeast
Phone: (770) 393-1630
Fax: (770) 395-1419

US Florida/South America
Phone: (813) 799-8406
Fax: (813) 799-8306

US Northwest
Phone: (408) 249-9696
Fax: (408) 249-6518

US North Central
Phone: (630) 773-3454
Fax: (630) 773-3907

US Northeast
Phone: (508) 692-7660
Fax: (508) 692-8185

Europe Mediterranean
Phone: (39-2) 93179911
Fax: (39-2) 93179913

Europe North
Phone: (44-1) 344 486444
Fax: (44-1) 344 486555

Europe South
Phone: (33-1) 49 06 39 80
Fax: (33-1) 49 06 39 90

Europe Central
Phone: (49-89) 829-1320
Fax: (49-89) 834-2734

Australia
Phone: (61-2) 9805 5555
Direct Fax: (61-2) 9888 9372

Hong Kong
Phone: (852) 2 827-0181
Fax: (852) 2 827-6488

Japan
Phone: (81-3) 5371 1551
Fax: (81-3) 5371 1501

Korea
Phone: (82-2) 565-2880
Fax: (82-2) 565-1440

Singapore
Phone: (65) 733-2511
Fax: (65) 733-0635

Taiwan
Phone: (886-2) 720-0282
Fax: (886-2) 757-6760

L8233_A
Printed in USA

Rockwell Semiconductor Systems