

# RS8235

## Endstation ATM ServiceSAR with xBR Traffic Management

The RS8235 Service Segmentation and Reassembly Controller integrates ATM terminal functions, PCI Bus Master and Slave controllers, and a UTOPIA interface with service specific functions in a single package. The *ServiceSAR* Controller generates and terminates ATM traffic as well as automatically scheduling cells for transmission. The RS8235 is targeted at cost-sensitive 155 Mbps throughput systems where the performance of the overall system is critical but the number of VCCs is not large.

Architecturally the RS8235 supports a similar feature set to the RS8234 ServiceSAR, but in a reduced footprint. It has been tuned for lower functionality systems.

The RS8235 directly connects to Mindspeed's RS8250 PHY for a total NIC solution. System designers can therefore offer a common traffic control and host interface to a new market segment.

### Advanced xBR Traffic Management

The xBR Traffic Manager in the RS8235 supports multiple ATM service categories. This includes CBR, VBR (both single and dual leaky bucket), UBR, GFR (Guaranteed Frame Rate) and ABR. The RS8235 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 RS8235's ABR algorithms to a specific system's or network's requirements.

### Multi-Queue Segmentation Processing

The RS8235's segmentation coprocessor generates ATM cells for up to 1 K VCCs at a line rate of up to 200 Mbps for simplex connections. The segmentation coprocessor formats cells on each channel according to segmentation VCC Tables, using up to four independent transmit queues and reporting segmentation status on a parallel set of up to four segmentation status queues.

-continued-

### Distinguishing Features

#### Service Specific Performance Accelerators

- LECID filtering and echo suppression
- Dual leaky bucket based on CLP (Frame Relay)
- Frame Relay DE interworking
- Internal SNMP MIB counters
- IP over ATM; supports both CLP0+1 and ABR shaping

#### Flexible Architectures

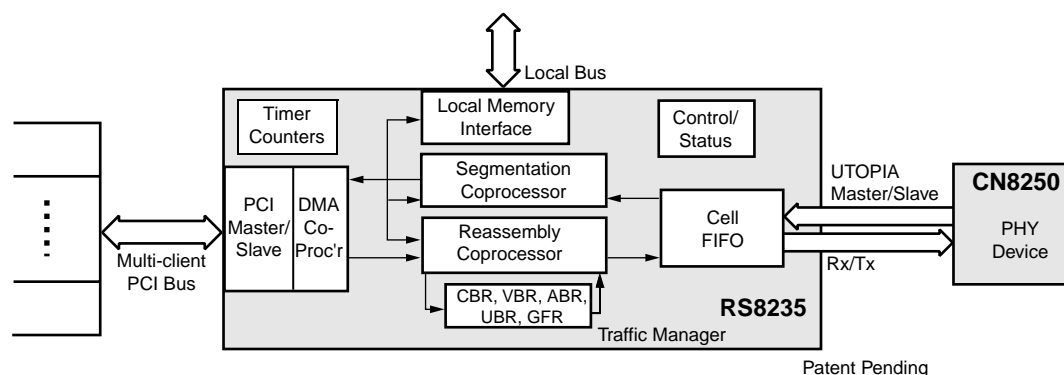
- Multi-peer host
- Direct switch attachment via reverse UTOPIA
- ATM terminal
  - Host control

#### xBR Traffic Management

- TM 4.1 Service Classes
  - CBR
  - VBR (single, dual & CLP-based leaky buckets)
  - Real time VBR
  - ABR
  - UBR
  - GFC (controlled & uncontrolled flows)
  - Guaranteed Frame Rate (GFR) (guaranteed MCR on UBR VCCs)
- 8 Levels of priorities (8 + CBR)

-continued-

### Functional Block Diagram



## Ordering Information

Model Number	Manufacturing Part Number	Product Revision	Package	Operating Temperature
RS8235	28235-15	A	208-pin QFP	0 °C to 85 °C

## Document Revision History

Document Number	Device Revision	Date	Comments
N8235DS1A	RS8235 Rev. A	—	Initial Release
N8235DSB	Rev. A	11/98	Additions and corrections to Chapter 14.
500407A	Rev. A	09/02	Revisions made. Changed format from Conexant to Mindspeed.

© 1998-2002, Mindspeed Technologies<sup>®</sup>, a Conexant business  
All Rights Reserved.

Information in this document is provided in connection with Mindspeed Technologies (Mindspeed) products. These materials are provided by Mindspeed as a service to its customers and may be used for informational purposes only. Mindspeed assumes no responsibility for errors or omissions in these materials. Mindspeed may make changes to specifications and product descriptions at any time, without notice. Mindspeed makes no commitment to update the information and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to its specifications and product descriptions.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Mindspeed's Terms and Conditions of Sale for such products, Mindspeed assumes no liability whatsoever.

THESE MATERIALS ARE PROVIDED AS IS WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, RELATING TO SALE AND/OR USE OF MINDSPEED PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, CONSEQUENTIAL OR INCIDENTAL DAMAGES, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. MINDSPEED FURTHER DOES NOT WARRANT THE ACCURACY OR COMPLETENESS OF THE INFORMATION, TEXT, GRAPHICS OR OTHER ITEMS CONTAINED WITHIN THESE MATERIALS. MINDSPEED SHALL NOT BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUES OR LOST PROFITS, WHICH MAY RESULT FROM THE USE OF THESE MATERIALS.

Mindspeed products are not intended for use in medical, lifesaving or life sustaining applications. Mindspeed customers using or selling Mindspeed products for use in such applications do so at their own risk and agree to fully indemnify Mindspeed for any damages resulting from such improper use or sale.

The following are trademarks of Conexant Systems, Inc.: Mindspeed Technologies<sup>®</sup>, the Mindspeed<sup>®</sup> logo, and "Build It First"<sup>®</sup>. Product names or services listed in this publication are for identification purposes only, and may be trademarks of third parties. Third-party brands and names are the property of their respective owners. For additional disclaimer information, please consult Mindspeed Technologies Legal Information posted at [www.mindspeed.com](http://www.mindspeed.com) which is incorporated by reference.

-continued from front-

The segmentation coprocessor retrieves 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 that schedules VCCs for transmission.

### Multi-Queue Reassembly Processing

The RS8235'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 that supports NNI or UNI addressing, the reassembly coprocessor processes up to 1 K VCCs simultaneously. The host supplies free buffers on up to four independent free buffer queues, and the reassembly coprocessor performs all CPCS protocol checks and reports the results of these checks as well as other status data on one of four independent reassembly status queues.

### High Performance Host Architecture with Buffer Isolation

The RS8235 host interface architecture maximizes performance and system flexibility. The device's control and status queues enable Host/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 SAR 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

Mindspeed provides an evaluation environment for the RS8235 that 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.

### Comparative Features of the RS8235 Endstation SAR Device

The RS8235 ServiceSAR offers designers streamlined functionality with respect to the RS8234 SAR. The RS8235 and the RS8234 are software compatible for the common features. Here is a Table of Feature Comparisons between the RS8235 and RS8234 SARs.

Feature	RS8235	RS8234
# of VCCs supported	1,000	32,000
Performance Monitoring	No	Yes
xBR Tunnels	No	Yes
# of each type of queue (i.e., transmit, free buffer, RSM status and SEG status)	4	32
AAL5	Yes	Yes
AAL3/4	No	Yes

08/15/02 5:00 pm

## *-continued Distinguishing Features-*

- Dynamic per-VCC scheduling
- Multiple programmable ABR templates (supplied by Mindspeed or user)
- Scheduler driven by local system clock for low jitter CBR
- Internal RM OAM cell feedback path
- Virtual FIFO rate matching (Source rate matching)
- Per-VCC MCR and ICR
- Tunneling
  - VP tunnels (VCI interleaving on PDU boundaries)
  - CBR tunnels (cells interleaved as UBR, VBR or ABR with an aggregate CBR limit)
- 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 signaling address assignment
- Message and streaming status modes
- raw cell mode (52 octet)
- 200 Mbps half duplex
- 155 Mbps full duplex (w/2-cell PDUs)
- Distributed host or SAR shared memory reassembly
- 8 programmable reassembly hardware time-outs (per-VCC assignable)
- Global max PDU length for AAL5
- Per-VCC buffer firewall (memory usage limit)
- Idle cell filtering

### **Multi-Queue Segmentation Processing**

- 4 transmit queues with optional priority levels
- 1 K VCCs maximum <sup>(1)</sup>
- AAL5 CPCS generation
- AAL0 Null CPCS (optional use of PTI for PDU demarcation)
- ATM cell header generation
- Raw cell mode (52 octet)
- 200 Mbps half duplex
- 155 Mbps full duplex (w/ 2-cell PDUs)
- Variable length transmit FIFO - 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 SAR shared memory segmentation
- Simultaneous segmentation and reassembly
- Per-PDU control of CLP/PTI (UBR)
- Per-PDU control of AAL5 UU field
- Message and streaming status modes
- Virtual Tx FIFO (PCI host)

### **Multi-Queue Reassembly Processing**

- 4 reassembly queues
- 1 K VCCs maximum <sup>(1)</sup>
- AAL5 CPCS checking
- AAL0
  - PTI termination
  - Cell count termination
- Early packet discard, 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

### **High Performance Host Architecture with Buffer Isolation**

- Write-only control and status
- Read multiple command for data transfer
- Up to 4 host clients control and status queues
- physical or logical clients
  - enables peer-to-peer architecture
- Descriptor-based buffer chaining
- Scatter/gather DMA
- Endian neutral (allows data word and control word byte swapping, for both big and little endian systems)
- 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 BOM
- Allows isolation of system resources
- Status queue interrupt delay

### **Designer Toolkit**

- Evaluation hardware and software
- Reference schematics
- Hardware Programming Interface - RS8235HPI reference sourcecode (C)

### **Generous Implementation of OAM Protocols**

- Detection of all F4/F5 OAM flows
- Optional global OAM Rx/Tx queues
- In-line OAM insertion and generation

### **Standards-Based I/O**

- 33 MHz PCI 2.1
- Serial EEPROM to store PCI configuration information
- PHY interfaces
  - UTOPIA master (Level 1)
  - UTOPIA slave (Level 1)
- Flexible SAR shared memory architecture
- Optional local control interface
- Boundary scan for board-level testing
- Source loopback for diagnostics
- Glueless connection to Mindspeed's ATM physical layer device, the RS8250

### **Standards Compliance**

- UNI/NNI 3.1
- TM 4.1
- Bellcore GR-1248
- ATM Forum B-ICI V2.0
- I.363
- I.610/GR-1248
- AToM MIB (RFC1695)
- ILMI MIB
- ANSI T1.635
- GFC per I.361
- SNMP
- PCI Revision 2.1
- IEEE 1149.1-1990
- IEEE 1149.1 Supplement B, 1994

### **Electrical/Mechanical**

- 208 QFP package
- 3.3 V power supply
- 5 V tolerant I/O pads
- 5 V-3.3 V PCI pads
- Low power 1.5 W (typical) @ full rate
- Commercial range
- TTL level inputs
- CMOS level outputs

### **Statistics and Write-only Counters**

- Global register counter of # of cells transmitted
- Global register counter of # of cells received on active channels
- Global 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 per-VCC service discard counters (frame relay and LANE)
- 1 programmable interval timer (32 bits w/ interrupt)

# Table of Contents

---

Table of Contents	v
List of Figures	xv
List of Tables	xvii
<b>1.0 Product Overview</b>	<b>1-1</b>
1.1 Introduction	1-1
1.2 Service Specific Performance Accelerators	1-3
1.3 Designer Toolkit	1-5
<b>2.0 Architecture Overview</b>	<b>2-1</b>
2.1 Introduction	2-1
2.2 High Performance Host Architecture With Buffer Isolation	2-2
2.2.1 Multiple ATM Clients	2-2
2.2.2 RS8235 Queue Structure	2-3
2.2.3 Buffer Isolation Utilizing Descriptor-Based Buffer Chaining	2-5
2.2.4 Status Queue Relation to Buffers and Descriptors	2-7
2.2.5 Write-Only Control/Status	2-10
2.2.6 Scatter/Gather DMA	2-10
2.2.7 Interrupts	2-11
2.3 Automated Segmentation Engine	2-12
2.4 Automated Reassembly Engine	2-14
2.5 Advanced xBR Traffic Management	2-16
2.5.1 CBR Traffic	2-17
2.5.2 VBR Traffic	2-18
2.5.3 ABR Traffic	2-18
2.5.4 UBR Traffic	2-19
2.5.5 GFR Traffic	2-19
2.5.6 xBR Cell Scheduler	2-19
2.5.7 ABR Flow Control Manager	2-20
2.6 Implementation of OAM Protocols	2-20
2.7 Standards-Based I/O	2-21

2.8	Electrical/Mechanical	2-22
2.9	Logic Diagram and Pin Descriptions	2-22
<b>3.0</b>	<b>Host Interface</b>	<b>3-1</b>
3.1	Overview	3-1
3.2	Multiple Client Architecture	3-2
3.2.1	Logical Clients	3-2
3.2.2	Resource Allocation	3-3
3.2.3	Resource Isolation	3-3
3.2.4	Peer-to-Peer Transfers	3-4
3.3	Write-Only Control and Status	3-5
3.3.1	Write-Only Control Queues	3-5
3.3.1.1	Control Variables	3-6
3.3.1.2	Queue Management	3-7
3.3.1.3	Underflow Conditions	3-8
3.3.2	Write-only Status Queues	3-8
3.3.2.1	Control Variables	3-8
3.3.2.2	Queue Management	3-9
3.3.2.3	Overflow Conditions	3-9
3.3.2.4	Status Queue Interrupt Delay	3-10
3.4	System Clocking	3-11
3.5	RS8235 Reset	3-11
<b>4.0</b>	<b>Segmentation Coprocessor</b>	<b>4-1</b>
4.1	Overview	4-1
4.2	Segmentation Functional Description	4-2
4.2.1	Segmentation VCCs	4-2
4.2.1.1	Segmentation VCC Table	4-2
4.2.1.2	VCC Identification	4-3
4.2.2	Submitting Segmentation Data	4-4
4.2.2.1	User Data Format	4-4
4.2.2.2	Buffer Descriptors	4-4
4.2.2.3	Host Linked Segmentation Buffer Descriptors	4-5
4.2.2.4	Transmit Queues	4-5
4.2.2.5	Partial PDUs	4-7
4.2.2.6	Virtual Paths	4-7
4.2.3	CPCS-PDU Processing	4-8
4.2.3.1	AAL5	4-8
4.2.3.2	AAL0	4-8
4.2.4	ATM Physical Layer Interface	4-9
4.2.5	Status Reporting	4-9
4.2.6	Virtual FIFOs	4-9
4.3	Segmentation Control and Data Structures	4-10
4.3.1	Segmentation VCC Table Entry	4-10

*Endstation ATM ServiceSAR with xBR Traffic Management*

4.3.2	Data Buffers . . . . .	4-12
4.3.3	Segmentation Buffer Descriptors . . . . .	4-13
4.3.4	Transmit Queues . . . . .	4-16
4.3.4.1	Entry Format . . . . .	4-16
4.3.4.2	Transmit Queue Management . . . . .	4-17
4.3.5	Segmentation Status Queues . . . . .	4-18
4.3.5.1	Entry Format . . . . .	4-18
4.3.5.2	Status Queue Management . . . . .	4-19
4.3.5.3	Status Queue Overflow . . . . .	4-19
4.3.6	Segmentation Internal SRAM Memory Map . . . . .	4-20
<b>5.0</b>	<b>Reassembly Coprocessor . . . . .</b>	<b>5-1</b>
5.1	Overview . . . . .	5-1
5.2	Reassembly Functional Description . . . . .	5-2
5.2.1	Reassembly VCCs . . . . .	5-2
5.2.1.1	Relation to Segmentation VCCs . . . . .	5-3
5.2.2	Channel Lookup . . . . .	5-3
5.2.2.1	Programmable Block Size for VCC Table/ VCI Index Table . . . . .	5-4
5.2.2.2	Setup . . . . .	5-5
5.2.2.3	Operation . . . . .	5-6
5.3	CPCS-PDU Processing . . . . .	5-7
5.3.1	AAL5 Processing . . . . .	5-8
5.3.1.1	AAL5 COM Processing . . . . .	5-8
5.3.1.2	AAL5 EOM Processing . . . . .	5-8
5.3.1.3	AAL5 Error Conditions . . . . .	5-9
5.3.2	AAL0 Processing . . . . .	5-9
5.3.2.1	Termination Methods . . . . .	5-9
5.3.2.2	AAL0 Error Conditions . . . . .	5-10
5.3.3	ATM Header Processing . . . . .	5-10
5.3.4	BOM Synchronization Signal . . . . .	5-10
5.4	Buffer Management . . . . .	5-12
5.4.1	Host vs. Local Reassembly . . . . .	5-12
5.4.2	Scatter Method . . . . .	5-12
5.4.3	Free Buffer Queues . . . . .	5-13
5.4.4	Linked Data Buffers . . . . .	5-14
5.4.5	Initialization of Buffer Structures . . . . .	5-16
5.4.5.1	Buffer Descriptors . . . . .	5-16
5.4.5.2	Free Buffer Queue Base Table . . . . .	5-16
5.4.5.3	Free Buffer Queue Entries . . . . .	5-16
5.4.5.4	Other Initialization . . . . .	5-16
5.4.6	Buffer Allocation . . . . .	5-17
5.4.7	Error Conditions . . . . .	5-17
5.4.8	Early Packet Discard . . . . .	5-18
5.4.8.1	General Description . . . . .	5-18
5.4.8.2	Frame Relay Packet Discard . . . . .	5-18

5.4.8.3	CLP Packet Discard	5-18
5.4.8.4	LANE-LECID Packet Discard — Echo Suppression on Multicast Data Frames	5-18
5.4.8.5	DMA FIFO Full	5-19
5.4.8.6	Error Conditions	5-19
5.4.9	Hardware PDU Time-out	5-20
5.4.9.1	Reassembly Time-out Process	5-20
5.4.9.2	Halting Time-out Processing	5-20
5.4.9.3	Timer Reset	5-20
5.4.9.4	Reassembly Time-out Condition	5-21
5.4.9.5	Time-out Period Calculation	5-21
5.4.10	Virtual FIFO Mode	5-21
5.4.10.1	Setup	5-21
5.4.10.2	Operation	5-21
5.4.10.3	Errors	5-21
5.4.11	Firewall Functions	5-22
5.4.11.1	Setup	5-22
5.4.11.2	Operation	5-22
5.4.11.3	Credit Return	5-23
<b>5.5</b>	<b>Global Statistics</b>	5-24
<b>5.6</b>	<b>Status Queue Operation</b>	5-25
5.6.1	Structure	5-25
5.6.1.1	Setup	5-26
5.6.1.2	Operation	5-27
5.6.1.3	Errors	5-27
5.6.1.4	Host Detection of Status Queue Entries	5-28
5.6.2	Status Queue Overflow or Full Condition	5-28
<b>5.7</b>	<b>Reassembly Control and Status Structures</b>	5-29
5.7.1	Channel Lookup Structures	5-29
5.7.2	Reassembly VCC Table	5-32
5.7.3	Reassembly Buffer Descriptor Structure	5-37
5.7.4	Free Buffer Queues	5-37
5.7.5	Reassembly Status Queues	5-39
5.7.6	LECID Table	5-42
5.7.7	Global Time-out Table	5-43
<b>6.0</b>	<b>Traffic Management</b>	6-1
<b>6.1</b>	<b>Overview</b>	6-1
6.1.1	xBR Cell Scheduler	6-3
6.1.2	ABR Flow Control Manager	6-4
<b>6.2</b>	<b>xBR Cell Scheduler Functional Description</b>	6-6
6.2.1	Scheduling Priority	6-6

*Endstation ATM ServiceSAR with xBR Traffic Management*

6.2.1.1	8 Priority Levels + CBR	6-6
6.2.1.2	VCC Priority Assignment	6-6
6.2.2	Dynamic Schedule Table	6-6
6.2.2.1	Overview	6-6
6.2.2.2	Schedule Table Slots	6-7
6.2.2.3	Schedule Slot Formats	6-8
6.2.2.4	Some Scheduling Scenarios	6-9
6.2.3	CBR Traffic	6-9
6.2.3.1	CBR Rate Selection	6-10
6.2.3.2	Available Rates	6-10
6.2.3.3	CBR Cell Delay Variation (CDV)	6-12
6.2.3.4	CBR Channel Management	6-13
6.2.4	VBR Traffic	6-14
6.2.4.1	Mapping RS8235 VBR Service Categories to TM 4.1 VBR Service Categories	6-14
6.2.4.2	Rate Shaping vs. Policing	6-14
6.2.4.3	Single Leaky Bucket	6-15
6.2.4.4	Dual Leaky Bucket	6-15
6.2.4.5	CLP Based Buckets	6-15
6.2.4.6	Rate Selection	6-15
6.2.4.7	Real-Time VBR and CDV	6-15
6.2.5	UBR Traffic	6-15
6.2.6	CBR Tunnels (Pipes)	6-16
6.2.7	Guaranteed Frame Rate	6-17
6.2.8	PCR Control for Priority Queues	6-17
<b>6.3</b>	<b>ABR Flow Control Manager</b>	<b>6-18</b>
6.3.1	A Brief Overview of TM 4.1	6-18
6.3.2	Internal ABR Feedback Control Loop	6-18
6.3.2.1	Source Flow Control Feedback	6-18
6.3.2.2	Destination Behavior	6-19
6.3.2.3	Out of Rate Cells	6-20
6.3.3	Source and Destination Behaviors	6-20
6.3.4	ABR VCC Parameters	6-20
6.3.5	ABR Templates	6-20
<b>6.4</b>	<b>GFC Flow Control Manager</b>	<b>6-22</b>
6.4.1	A Brief Overview of GFC	6-22
6.4.2	The RS8235's Implementation of GFC	6-22
6.4.3	Configuring the Link for GFC Operation	6-23
<b>6.5</b>	<b>Traffic Management Control and Status Structures</b>	<b>6-24</b>
6.5.1	Schedule Table	6-24
6.5.2	CBR-Specific Structures	6-24
6.5.2.1	CBR Traffic	6-24
6.5.2.2	Tunnel Traffic	6-24
6.5.2.3	SCH_STATE Fields For CBR	6-25
6.5.3	VBR-Specific Structures	6-26
6.5.3.1	VBR SCH_STATE	6-26

6.5.3.2	VBR1 or VBR2 Schedule State Table	6-26
6.5.3.3	Bucket Table for VBR2 and VBRC	6-27
6.5.4	GFR-Specific Structures	6-28
6.5.4.1	GFR Schedule State Table	6-28
6.5.4.2	GFR MCR Limit Bucket Table	6-29
6.5.5	ABR-Specific Structures	6-30
6.5.5.1	ABR Schedule State Table	6-30
6.5.6	Scheduler Internal SRAM Registers	6-33
<b>7.0</b>	<b>OAM Functions</b>	<b>7-1</b>
7.1	OAM Overview	7-1
7.1.1	OAM Functions Supported	7-1
7.1.2	OAM Flows Supported	7-1
7.1.2.1	F4 OAM Flow	7-1
7.1.2.2	F5 OAM Flow	7-2
7.1.3	OAM Cell Format	7-2
7.2	Segmentation of OAM Cells	7-3
7.2.1	Key OAM-Related Fields for OAM Segmentation	7-3
7.2.1.1	Segmentation Buffer Descriptors	7-3
7.2.1.2	Low Latency Transmission	7-3
7.2.1.3	Segmentation Status Queue	7-3
7.2.1.4	F4 Flow	7-3
7.2.2	Error Condition During OAM Segmentation	7-4
7.3	Reassembly of OAM Cells	7-5
7.3.1	Key OAM-Related Fields for OAM Reassembly	7-5
7.3.1.1	Reassembly VCC State Table	7-5
7.3.1.2	Reassembly Status Queue	7-5
7.3.1.3	F4 Flow	7-5
7.3.2	OAM Reassembly Operation	7-5
7.3.3	Error Conditions During OAM Reassembly	7-6
7.4	OAM Control and Status Structures	7-6
<b>8.0</b>	<b>DMA Coprocessor</b>	<b>8-1</b>
8.1	Overview	8-1
8.2	DMA Read	8-1
8.3	DMA Write	8-1
8.4	Misaligned Transfers	8-2
8.5	Control Word Transfers	8-4
<b>9.0</b>	<b>Local Memory Interface</b>	<b>9-1</b>
9.1	Overview	9-1
9.2	Memory Bank Characteristics	9-2
9.3	Memory Size Analysis	9-5

<b>10.0</b>	<b>PCI Bus Interface</b> .....	10-1
10.1	Overview .....	10-1
10.2	Unimplemented PCI Bus Interface Functions .....	10-3
10.3	PCI Configuration Space .....	10-3
10.4	PCI Bus Master Logic .....	10-4
10.5	Burst FIFO Buffers .....	10-5
10.6	PCI Bus Slave Logic .....	10-6
10.7	Byte Swapping of Control Structures .....	10-6
10.8	Power Management .....	10-7
10.9	Interface Module to Serial EEPROM .....	10-8
10.9.1	EEPROM Format .....	10-8
10.9.2	Loading the EEPROM Data at Reset .....	10-9
10.9.3	Accessing the EEPROM .....	10-9
10.9.4	Using the Subsystem ID Without an EEPROM .....	10-10
10.10	PCI Host Address Map .....	10-10
<b>11.0</b>	<b>ATM UTOPIA Interface</b> .....	11-1
11.1	Overview of ATM UTOPIA Interface .....	11-1
11.2	Typical Interface Connection to the RS8250 PHY .....	11-2
11.3	ATM Physical Interface Logic .....	11-6
11.4	ATM Physical I/O Pins .....	11-7
11.5	UTOPIA Mode Cell Handshake Timing .....	11-9
11.6	UTOPIA Mode Octet Handshake Timing .....	11-11
11.7	Slave UTOPIA Mode .....	11-13
11.8	Loopback Mode .....	11-15
11.9	Receive Cell Synchronization Logic .....	11-16
11.10	Transmit Cell Synchronization Logic .....	11-17
<b>12.0</b>	<b>RS8235 Registers</b> .....	12-1
12.1	CSR Registers .....	12-1
12.1.1	Register Terminology .....	12-2
12.2	System Registers .....	12-3
12.2.1	0x00 - Real-Time Clock Register (CLOCK) .....	12-3
12.2.2	0x04 - Alarm Register 1 (ALARM1) .....	12-3
12.2.3	0x0c - System Status Register (SYS_STAT) .....	12-4
12.2.4	0x14 - Configuration Register 0 (CONFIG0) .....	12-5
12.2.5	0x1c - Interrupt Delay Register (INT_DELAY) .....	12-6

<b>12.3 Segmentation Registers</b> .....	12-7
12.3.1 0x80 - Segmentation Control Register (SEG_CTRL) .....	12-7
12.3.2 0x84 - Seg VCC Table & Schedule Table Base Address Register (SEG_VBASE) .....	12-8
12.3.3 0x88 - Segmentation PM Base Register (SEG_PMBASE) .....	12-8
12.3.4 0x8c - Segmentation Transmit Queue Base Register (SEG_TXBASE) .....	12-8
<b>12.4 Scheduler Registers</b> .....	12-9
12.4.1 0xa0 - Schedule Priority Register (SCH_PRI) .....	12-9
12.4.2 0xa8 - Schedule Size Register (SCH_SIZE) .....	12-10
12.4.3 0xac - Scheduler Control Register (SCH_CTRL) .....	12-10
12.4.4 0xb0 - Maximum ABR VCC_INDEX Register (SCH_ABR_MAX) .....	12-11
12.4.5 Schedule ABR Constant Register (SCH_ABR_CON) .....	12-11
12.4.6 0xb4 - Schedule ABR Constant Register (SCH_ABR_CON) .....	12-12
12.4.7 Schedule ABR Constant Register (SCH_ABR_CON) .....	12-12
12.4.8 0xb8 - ABR Decision Table Lookup Base Register (SCH_ABRBASE) .....	12-12
12.4.9 Schedule ABR Constant Register (SCH_ABR_CON) .....	12-12
12.4.10 0xbc - ABR Congestion Register (SCH_CNG) .....	12-12
12.4.11 Schedule ABR Constant Register (SCH_ABR_CON) .....	12-13
12.4.12 0xc0 - PCR Queue Interval 0 and 1 Register (PCR_QUE_INT01) .....	12-13
12.4.13 0xc4 - PCR Queue Interval 2 and 3 Register (PCR_QUE_INT23) .....	12-13
<b>12.5 Reassembly Registers</b> .....	12-14
12.5.1 0xf0 - Reassembly Control Register 0 (RSM_CTRL0) .....	12-14
12.5.2 0xf4 - Reassembly Control Register 1 (RSM_CTRL1) .....	12-16
12.5.3 0xf8 - Reassembly Free Buffer Queue Base Register (RSM_FQBASE) .....	12-17
12.5.4 0xfc - Reassembly Free Buffer Queue Control Register (RSM_FQCTRL) .....	12-17
12.5.5 0x100 - Reassembly Table Base Register (RSM_TBASE) .....	12-18
12.5.6 0x104 - Reassembly Time-out Register (RSM_TO) .....	12-18
12.5.7 0x108 - Reassembly/Segmentation Queue Base Register (RS_QBASE) .....	12-18
<b>12.6 Counters and Status Registers</b> .....	12-19
12.6.1 0x160 - ATM Cells Transmitted Counter (CELL_XMIT_CNT) .....	12-19
12.6.2 0x164 - ATM Cells Received Counter (CELL_RCVD_CNT) .....	12-19
12.6.3 0x168 - ATM Cells Discarded Counter (CELL_DSC_CNT) .....	12-19
12.6.4 0x16c - AAL5 PDUs Discarded Counter (AAL5_DSC_CNT) .....	12-20
12.6.5 0x1a4 - Host Status Write Register (HOST_ST_WR) .....	12-20
12.6.6 Host Interrupt Status Registers .....	12-20
12.6.6.1 0x1c0 - Host Interrupt Status Register 0 (HOST_ISTAT0) .....	12-21
12.6.6.2 0x1c4 - Host Interrupt Status Register 1 (HOST_ISTAT1) .....	12-22
12.6.6.3 0x1d0 - Host Interrupt Mask Register 0 (HOST_IMASK0) .....	12-23
12.6.6.4 0x1d4 - Host Interrupt Mask Register 1 (HOST_IMASK1) .....	12-24
<b>12.7 PCI Bus Interface Registers</b> .....	12-25
12.7.1 PCI Configuration Register .....	12-25
12.7.1.1 PCI Configuration Register Field Descriptions .....	12-26
12.7.1.2 PCI Command Register Description .....	12-28
12.7.1.3 PCI Status Register Description .....	12-28
12.7.1.4 PCI Special Status Register Description .....	12-29
12.7.1.5 EEPROM Register Description .....	12-30

<b>13.0</b>	<b>SAR Initialization - Example Tables</b> .....	13-1
13.1	Segmentation Initialization .....	13-2
13.1.1	Segmentation Control Registers .....	13-2
13.1.2	Segmentation Internal Memory Control Structures .....	13-3
13.1.3	Segmentation SAR Shared Memory Control Structures .....	13-4
13.2	Scheduler Initialization .....	13-6
13.2.1	Scheduler Control Registers .....	13-6
13.2.2	Scheduler Internal Memory Control Structures .....	13-6
13.2.3	Scheduler SAR Shared Memory Control Structures .....	13-7
13.3	Reassembly Initialization .....	13-9
13.3.1	Reassembly Control Registers .....	13-9
13.3.2	Reassembly Internal Memory Control Structures .....	13-11
13.3.3	Reassembly SAR Shared Memory Control Structures .....	13-12
13.4	General Initialization .....	13-14
<b>14.0</b>	<b>Electrical and Mechanical Specifications</b> .....	14-1
14.1	Timing .....	14-1
14.1.1	PCI Bus Interface Timing .....	14-1
14.1.2	ATM Physical Interface Timing—UTOPIA and Slave UTOPIA .....	14-3
14.1.3	System Clock Timing .....	14-5
14.1.4	RS8235 Memory Interface Timing .....	14-7
14.1.5	PHY Interface Timing (Standalone Mode) .....	14-11
14.2	Absolute Maximum Ratings .....	14-13
14.3	DC Characteristics .....	14-14
14.4	Mechanical Specifications .....	14-15
<b>Appendix A:</b>	<b>Boundary Scan</b> .....	A-1
A.1	Instruction Register .....	A-3
A.2	BYPASS Register .....	A-3
A.3	Boundary Scan Register .....	A-4
A.4	Boundary Scan Register Cells .....	A-4
A.5	Electrical Characteristics .....	A-8
A.6	Boundary Scan Description Language (BSDL) File .....	A-10



## List of Figures

Figure 1-1.	RS8235 Functional Block Diagram . . . . .	1-2
Figure 2-1.	Multiple Client Architecture . . . . .	2-3
Figure 2-2.	RS8235 Queue Architecture . . . . .	2-4
Figure 2-3.	Interaction of Queues with RS8235 Functional Blocks . . . . .	2-5
Figure 2-4.	Reassembly Buffer Isolation — Data Buffers Separated from Descriptors . . . . .	2-6
Figure 2-5.	Segmentation Buffer Isolation — Data Buffers Separated from Descriptors . . . . .	2-7
Figure 2-6.	Segmentation Status Queues Related to Data Buffers and Descriptors . . . . .	2-8
Figure 2-7.	Reassembly Status Queues Related to Data Buffers and Descriptors . . . . .	2-9
Figure 2-8.	Write-Only Control and Status Architecture . . . . .	2-10
Figure 2-9.	Multi-Level Model for Prioritizing Segmentation Traffic . . . . .	2-17
Figure 2-10.	RS8235 Logic Diagram (1 of 3) . . . . .	2-23
Figure 3-1.	Client/Server Model of the RS8235 . . . . .	3-2
Figure 3-2.	Peer-to-Peer vs. Centralized Memory Data Transfers . . . . .	3-4
Figure 3-3.	Write-Only Control Queue . . . . .	3-7
Figure 3-4.	Write-Only Status Queue . . . . .	3-9
Figure 4-1.	Segmentation VCC Table . . . . .	4-3
Figure 4-2.	Segmentation Buffer Descriptor Chaining . . . . .	4-5
Figure 4-3.	Before SAR Transmit Queue Entry Processing . . . . .	4-6
Figure 4-4.	After SAR Transmit Queue Entry Processing . . . . .	4-7
Figure 4-5.	AAL5 CPCS-PDU Generation . . . . .	4-8
Figure 5-1.	Reassembly — Basic Process Flow . . . . .	5-2
Figure 5-2.	Reassembly VCC Table . . . . .	5-3
Figure 5-3.	Direct Index Method for VPI/VCI Channel Lookup . . . . .	5-4
Figure 5-4.	Programmable Block Size Alternate Direct Index Method . . . . .	5-5
Figure 5-5.	CPCS-PDU Reassembly . . . . .	5-7
Figure 5-6.	AAL5 EOM Cell Processing — Fields to Status Queue . . . . .	5-8
Figure 5-7.	AAL5 Processing — CRC and PDU Length Checks . . . . .	5-9
Figure 5-8.	AAL0 PTI PDU Termination . . . . .	5-10
Figure 5-9.	Host and SAR Shared Memory Data Structures for Scatter Method . . . . .	5-13
Figure 5-10.	Free Buffer Queue Structure . . . . .	5-14
Figure 5-11.	Buffer Structures . . . . .	5-15
Figure 5-12.	Data Structure Locations for Status Queues . . . . .	5-25
Figure 5-13.	Status Queue Structure Format . . . . .	5-26
Figure 5-14.	VPI/VCI Channel Lookup Structure . . . . .	5-29
Figure 5-15.	Reassembly VCC Table Entry Lookup Mechanism . . . . .	5-32
Figure 5-16.	LECID Table . . . . .	5-42
Figure 6-1.	Non-ABR Cell Scheduling . . . . .	6-3
Figure 6-2.	ABR Flow Control . . . . .	6-4
Figure 6-3.	Schedule Table with Size = 100 . . . . .	6-7

Figure 6-4.	Schedule Slot Formats . . . . .	6-8
Figure 6-5.	One Possible Scheduling Priority Scheme with the RS8235 . . . . .	6-9
Figure 6-6.	Assigning CBR Cell Slots. . . . .	6-11
Figure 6-7.	Introduction of CDV at the ATM/PHY Layer Interface . . . . .	6-12
Figure 6-8.	Schedule Table with Slot Conflicts at Different CBR Rates . . . . .	6-12
Figure 6-9.	CDV Caused by Schedule Table Size at Certain CBR Rates . . . . .	6-13
Figure 6-10.	ABR Service Category Feedback Control . . . . .	6-18
Figure 6-11.	RS8235 ABR-ER Feedback Loop (Source Behavior) . . . . .	6-19
Figure 6-12.	RS8235 ABR-ER Feedback Generation (Destination Behavior) . . . . .	6-20
Figure 6-13.	Head and Tail Pointers. . . . .	6-33
Figure 7-1.	OAM Cell Format . . . . .	7-2
Figure 8-1.	Little Endian Aligned Transfer . . . . .	8-2
Figure 8-2.	Little Endian Misaligned Transfer. . . . .	8-3
Figure 8-3.	Big Endian Aligned Transfer. . . . .	8-3
Figure 8-4.	Big Endian Misaligned Transfer . . . . .	8-4
Figure 9-1.	RS8235 Memory Map . . . . .	9-1
Figure 9-2.	0.5 MB SRAM Bank Using by_8 Devices. . . . .	9-3
Figure 9-3.	1 MB SRAM Bank Using by_16 Devices . . . . .	9-4
Figure 11-1.	RS8250 and SAR (RS8235) Interface . . . . .	11-3
Figure 11-2.	RS8235/PHY Functional Timing with Inserted Wait States . . . . .	11-4
Figure 11-3.	RS8235/RS8250 Read/Write Functional Timing . . . . .	11-5
Figure 11-4.	Receive Timing in UTOPIA Mode with Cell Handshake . . . . .	11-9
Figure 11-5.	Transmit Timing in UTOPIA Mode with Cell Handshake . . . . .	11-10
Figure 11-6.	Receive Timing in UTOPIA Mode with Octet Handshake . . . . .	11-11
Figure 11-7.	Transmit Timing in UTOPIA Mode with Octet Handshake . . . . .	11-12
Figure 11-8.	Receive Timing in Slave UTOPIA Mode . . . . .	11-13
Figure 11-9.	Transmit Timing in Slave UTOPIA Mode . . . . .	11-14
Figure 11-10.	Source Loopback Mode Diagram . . . . .	11-15
Figure 14-1.	PCI Bus Input Timing Measurement Conditions . . . . .	14-2
Figure 14-2.	PCI Bus Output Timing Measurement Conditions . . . . .	14-3
Figure 14-3.	UTOPIA and Slave UTOPIA Input Timing Measurement Conditions . . . . .	14-4
Figure 14-4.	UTOPIA and Slave UTOPIA Output Timing Measurement Conditions . . . . .	14-5
Figure 14-5.	Input System Clock Waveform . . . . .	14-6
Figure 14-6.	Output System Clock Waveform . . . . .	14-6
Figure 14-7.	RS8235 Memory Read Timing . . . . .	14-9
Figure 14-8.	RS8235 Memory Write Timing . . . . .	14-10
Figure 14-9.	Synchronous PHY Interface Input Timing . . . . .	14-11
Figure 14-10.	Synchronous PHY Interface Output Timing. . . . .	14-12
Figure 14-11.	208-Pin Plastic Quad Flat Pack . . . . .	14-15
Figure 14-12.	RS8235 Pinout Configuration . . . . .	14-16
Figure A-1.	Test Circuitry Block Diagram . . . . .	A-2
Figure A-2.	Timing Diagram For Boundary Scan . . . . .	A-9

## List of Tables

Table 2-1.	Hardware Signal Definitions (1 of 5)	2-26
Table 3-1.	RS8235 Control and Status Queues	3-5
Table 3-2.	Write-Only Control Queue Variables	3-6
Table 3-3.	Write-only Status Queue Variables	3-8
Table 4-1.	Segmentation PDU Delineation	4-4
Table 4-2.	Segmentation VCC Table Entry — AAL5/AAL0 Format	4-10
Table 4-3.	Segmentation VCC Table Entry — AAL5 and 0 Field Descriptions (1 of 2)	4-11
Table 4-4.	Segmentation VCC Table Entry — Virtual FIFOs	4-12
Table 4-5.	Segmentation VCC Table Entry — Virtual FIFO Format Field Descriptions	4-12
Table 4-6.	Segmentation Buffer Descriptor Entry Format	4-13
Table 4-7.	MISC_DATA Field Bit Definitions with HEADER_MOD Bit Set <sup>(1)</sup>	4-13
Table 4-8.	MISC_DATA Field Bit Definitions with RPL_VCI Bit Set (1)	4-13
Table 4-9.	Field Descriptions of the Segmentation Buffer Descriptor (1 of 3)	4-14
Table 4-10.	Transmit Queue Entry Format	4-16
Table 4-11.	Field Descriptions of the Transmit Queue Entry	4-16
Table 4-12.	Transmit Queue Base Table Entry	4-17
Table 4-13.	Transmit Queue Base Table Entry Field Descriptions	4-17
Table 4-14.	Segmentation Status Queue Entry	4-18
Table 4-15.	Segmentation Status Queue Entry Field Descriptions	4-18
Table 4-16.	Segmentation Status Queue Base Table Entry	4-19
Table 4-17.	Segmentation Status Queue Base Table Entry Field Descriptions	4-19
Table 4-18.	Segmentation Internal SRAM Memory Map	4-20
Table 5-1.	Programmable Block Size Values for Direct Index Lookup	5-5
Table 5-2.	STAT Output Pin Values for BOM Synchronization	5-11
Table 5-3.	Normal VPI Index Table Entry Format	5-29
Table 5-4.	VPI Index Table Entry Format with EN_PROG_BLK_SX(RSM_CTRL1) Enabled	5-29
Table 5-5.	VPI Index Table Entry Descriptions	5-30
Table 5-6.	Normal VCI Index Table Format	5-30
Table 5-7.	VCI Index Table Format with EN_PROG_BLK_SZ(RSM_CTRL1) Enabled	5-31
Table 5-8.	VCI Index Table Descriptions	5-31
Table 5-9.	Reassembly VCC Table Entry Format — AAL5	5-33
Table 5-10.	Reassembly VCC Table Entry Format — AAL0	5-34
Table 5-11.	PDU_FLAGS Field Bit Definitions	5-34
Table 5-12.	ABR_CTRL Field Bit Definitions	5-34
Table 5-13.	AAL_EN Field Bit Definitions	5-34
Table 5-14.	Reassembly VCC Table Descriptions (1 of 2)	5-35
Table 5-15.	Reassembly Buffer Descriptor Structure	5-37
Table 5-16.	Reassembly Buffer Descriptor Structure Definitions	5-37
Table 5-17.	Free Buffer Queue Base Table Entry Format	5-38

Table 5-18.	Free Buffer Queue Base Table Entry Descriptions . . . . .	5-38
Table 5-19.	Free Buffer Queue Entry Format . . . . .	5-38
Table 5-20.	Free Buffer Queue Entry Descriptions . . . . .	5-39
Table 5-21.	Reassembly Status Queue Base Table Entry Format . . . . .	5-39
Table 5-22.	Reassembly Status Queue Base Table Entry Descriptions . . . . .	5-40
Table 5-23.	Reassembly Status Queue Entry Format . . . . .	5-40
Table 5-24.	PDU_CHECKS Field Bits . . . . .	5-40
Table 5-25.	STATUS Field Bits . . . . .	5-40
Table 5-26.	STM Field Bits . . . . .	5-40
Table 5-27.	Reassembly Status Queue Entry Descriptions (1 of 2) . . . . .	5-41
Table 5-28.	LECID Table Entries . . . . .	5-42
Table 5-29.	LECID Table Field Definition . . . . .	5-43
Table 5-30.	Global Time-out Table Entry Format . . . . .	5-43
Table 5-31.	Global Time-out Table Entry Descriptions . . . . .	5-43
Table 6-1.	ATM Service Category Parameters and Attributes (1 of 2) . . . . .	6-1
Table 6-2.	Selection of Schedule Table Slot Size by System Requirements . . . . .	6-8
Table 6-3.	Schedule Slot Entry — CBR/Tunnel Traffic . . . . .	6-24
Table 6-4.	CBR_TUN_ID Field, Bit Definitions — CBR Slot . . . . .	6-24
Table 6-5.	CBR_TUN_ID Field, Bit Definitions — Tunnel Slot . . . . .	6-24
Table 6-6.	Schedule Slot Field Descriptions — CBR Traffic . . . . .	6-25
Table 6-7.	SCH_STATE for SCH_MODE = CBR . . . . .	6-25
Table 6-8.	CBR SCH_STATE Field Descriptions (1 of 2) . . . . .	6-25
Table 6-9.	SCH_STATE for SCH_MODE = VBR1 or VBR2 . . . . .	6-26
Table 6-10.	VBR1 and VBR2 SCH_STATE Field Descriptions (1 of 2) . . . . .	6-26
Table 6-11.	Bucket Table Entry . . . . .	6-27
Table 6-12.	Bucket Table Entry Field Descriptions . . . . .	6-27
Table 6-13.	SCH_STATE for SCH_MODE = GFR . . . . .	6-28
Table 6-14.	SCH_STATE Field Descriptions for SCH_MODE = GFR . . . . .	6-28
Table 6-15.	GFR MCR Limit Bucket Table Entry . . . . .	6-29
Table 6-16.	GFR MCR Bucket Table Entry Field Descriptions . . . . .	6-29
Table 6-17.	SCH_STATE for SCH_MODE = ABR . . . . .	6-30
Table 6-18.	ER SCH_STATE Field Descriptions (1 of 2) . . . . .	6-31
Table 6-19.	Scheduler Internal SRAM Memory Map . . . . .	6-33
Table 9-1.	Memory Bank Size . . . . .	9-2
Table 9-2.	Memory Size in Bytes . . . . .	9-5
Table 10-1.	EEPROM Fields . . . . .	10-8
Table 11-1.	Standalone Interface Pins . . . . .	11-2
Table 11-2.	ATM Physical Interface Mode Select . . . . .	11-6
Table 11-3.	UTOPIA Mode Signals . . . . .	11-7
Table 11-4.	Slave UTOPIA Mode Interface Signals . . . . .	11-8
Table 12-1.	RS8235 Control and Status Registers (1 of 2) . . . . .	12-1
Table 13-1.	Table of Values for Segmentation Control Register Initialization . . . . .	13-2
Table 13-2.	Table of Values for Segmentation Internal Memory Initialization . . . . .	13-3
Table 13-3.	Table of Values for SEG SAR Shared Memory Initialization (1 of 2) . . . . .	13-4
Table 13-4.	Table of Values for Scheduler Control Register Initialization . . . . .	13-6
Table 13-5.	Table of Values for Scheduler SAR Shared Memory Initialization (1 of 2) . . . . .	13-7

*Endstation ATM ServiceSAR with xBR Traffic Management*

Table 13-6.	Table of Values for Reassembly Control Register Initialization (1 of 2) . . . . .	13-9
Table 13-7.	Table of Values for Reassembly Internal Memory Initialization . . . . .	13-11
Table 13-8.	Table of Values for RSM SAR Shared Memory Initialization (1 of 2) . . . . .	13-12
Table 13-9.	Table of Values for General Control Register Initialization (1 of 2) . . . . .	13-14
Table 14-1.	PCI Bus Interface Timing Parameters (1 of 2) . . . . .	14-1
Table 14-2.	UTOPIA Interface Timing Parameters . . . . .	14-3
Table 14-3.	Slave UTOPIA Interface Timing Parameters . . . . .	14-4
Table 14-4.	System Clock Timing (1 of 2) . . . . .	14-5
Table 14-5.	SRAM Organization Loading Dependencies . . . . .	14-7
Table 14-6.	SAR Shared Memory Output Loading Conditions . . . . .	14-7
Table 14-7.	RS8234 Memory Interface Timing . . . . .	14-8
Table 14-8.	PHY Interface Timing (PROCMODE = 1) . . . . .	14-11
Table 14-9.	Absolute Maximum Ratings . . . . .	14-13
Table 14-10.	DC Characteristics . . . . .	14-14
Table 14-11.	RS8235 Pinout (1 of 6) . . . . .	14-17
Table A-1.	Boundary Scan Signals . . . . .	A-1
Table A-2.	IEEE Std. 1149.1 Instructions . . . . .	A-3
Table A-3.	Boundary Scan Register Cells (1 of 6) . . . . .	A-5
Table A-4.	Timing Specifications . . . . .	A-8



# 1.0 Product Overview

---

## 1.1 Introduction

The RS8235 Service Segmentation and Reassembly Controller (*ServiceSAR*) delivers a wide range of advanced ATM, AAL, and service-specific features in a highly integrated CMOS package.

Some of the RS8235's service-level features provide system designers with capabilities of accelerating specific protocol interworking functions. These features include 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.

Other service-level functions enable network level functionality or topologies. Two examples of these features include Generic Flow Control (GFC), and echo suppression of multicast data frames on Emulated LAN (ELAN) channels.

In addition to meeting the requirements contained in UNI 3.1, the RS8235 complies with ATM Forum Traffic Management specification TM4.0. The RS8235 provides traffic shaping for all service categories, including CBR, Variable Bit Rate (VBR) (both single and dual leaky bucket), Unspecified Bit Rate (UBR), Available Bit Rate (ABR), GFC (both controlled and uncontrolled flows), and Guaranteed Frame Rate (GFR) (i.e., guaranteed Minimum Cell Rate [MCR] on UBR Virtual Channel Connections [VCCs]). The internal xBR Traffic Manager automatically schedules each VCC according to user-assigned parameters.

The RS8235's architecture is designed to minimize and control host traffic congestion. The host manages the RS8235 terminal using an efficient architecture employing write-only control and status queues. For example, the host submits data for transmit by writing buffer descriptor pointers to one of four transmit queues. These entries can be thought of as task lists for the *ServiceSAR* to perform. The RS8235 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.

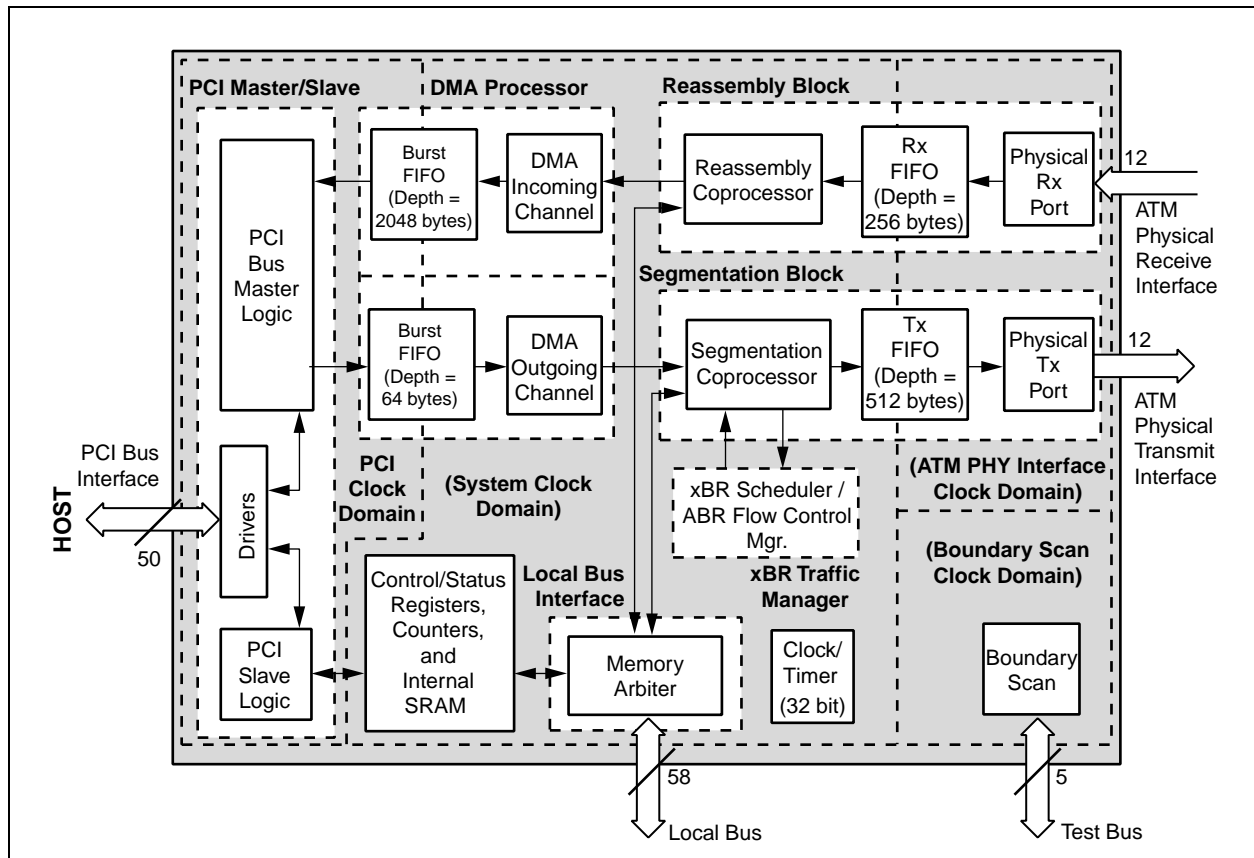
The RS8235 host interface provides for control of host congestion through the following mechanisms. First, each peer maintains separate control and status queues. Then, each VCC in a peer group can be limited to a specific maximum receive buffer utilization, further controlling congestion. Early Packet Discard 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) bit in Turnaround (Resource Management (RM) cells, based on user configuration.

The RS8235 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. These coprocessors, though they run off the same system clock, operate asynchronously from each other. Communication between the coprocessors takes place through on-chip FIFOs or through queues in SAR shared memory (i.e., memory local to the SAR and accessible both to the SAR and the Host). The five coprocessors are the Incoming Direct Memory Access (DMA), Outgoing DMA, Reassembly, Segmentation, and xBR Traffic Manager.

The RS8235's on-chip coprocessor blocks are surrounded by high performance PCI and 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. RS8235 Functional Block Diagram



## 1.2 Service Specific Performance Accelerators

The RS8235 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 [Chapter 2.0](#) and are fully described in succeeding chapters.

### *UNI or NNI Addressing*

The RS8235 handles both User-Network Interface (UNI) addresses (uses an 8-bit Virtual Path Identifier [VPI] field), and Network-to-Network Interface (NNI) addresses (uses a 12-bit VPI field).

### *Frame Relay Interworking*

The VBR traffic category includes rate-shaping via the dual leaky bucket Generic Cell Rate Algorithm (GCRA) based on the Cell Loss Priority (CLP) bit, for use in Frame Relay. The RS8235 also implements the Frame Relay discard attribute by performing early packet discard based on the frame's DE field and assigned discard priority.

### *IP Interworking*

The RS8235 facilitates ATM call control signalling procedures as defined in ATM Forum's UNI Signalling 4.0 Specification (SIG 4.0), to support IP over ATM environments. Some of the SIG 4.0 capabilities that are of interest to IP over ATM and which the RS8235 allows for are:

- ABR signalling for point-to-point calls
- Traffic parameter negotiation
- Frame discard support

### *Guaranteed Frame Rate*

The RS8235 can rate-shape ATM Adaptation Layer Type 5 (AAL5) Common Part Convergence Sublayer Protocol Data Units (CPCS-PDUs, i.e., "frames") in the UBR service category, by providing a guaranteed Minimum Cell Rate (MCR) for UBR VCCs.

### ***Early Packet Discard***

The Early Packet Discard (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.
- LANE-LECID packet discard to implement echo suppression on multicast data frames on ELAN channels.
- Packet discard when a firewall condition occurs on a VCC or group.
- Receive FIFO full condition/threshold.

### ***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 RS8235 to segment traffic from a fixed PCI address (i.e., a Virtual FIFO) for circuit-based CBR traffic.

The user can delineate up to eight CBR pipes (or tunnels) in which to transmit multiple UBR, VBR or ABR channels. In addition, the bandwidth of any single tunnel can be shared by up to four different priorities of traffic, in effect, establishing a multi-service tunnel. This allows proprietary management schemes to operate under preallocated CBR bandwidths.

### ***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 Explicit Rate (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.
- New per-VCC Minimum Cell Rate (MCR) and Initial Cell Rate (ICR) fields reduce the number of ABR templates needed in local memory.
- The RS8235 allows rate adjustments on Turnaround RM cells, based on congestion in the host.
- The RS8235 allows rate adjustments due to Use-It-Or-Lose-It behavior.
- The RS8235 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 RS8235 optionally posts the current Allowed Cell Rate (ACR) on the segmentation status queue for the host monitoring functions.

### ***VBR Traffic Management***

The RS8235 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 Mbps.

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)  
(This is the mode recommended by the Internet Engineering Task Force [IETF] as the most convenient model for IP over ATM interworking.)

### ***Virtual Path Networking***

The RS8235 can interleave segmentation of numerous VCCs (i.e., separate 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.

### ***AAL for Proprietary Traffic***

The RS8235 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.

### ***Internal Counters***

RS8235 has three internal Simple Network Management Protocol (SNMP) Management Information Base (MIB) counters that measure cells received, cells discarded, and AAL5 PDUs discarded (to meet ILMI and RFC1695 requirements).

## ***1.3 Designer Toolkit***

The RS8235 ATM evaluation environment provides evaluation capability for the RS8235 SAR. This environment serves as a hardware and software reference design for development of customer-specific ATM applications. The evaluation hardware and software was designed to provide a rapid prototyping environment to assist and speed customer development of new ATM products, thereby reducing product time to market.

This environment facilitates the following:

- Assists and speeds customer product development
- Provides hardware reference design
- Provides software reference design (based on ANSI "C")
- Provides traffic generation and checking capability

Comprising part of this development environment is the RS8235/825xEVM, a PCI card specifically designed to be a full-featured ATM controller implementing the full functionality of the RS8235 *ServiceSAR*. The *ServiceSAR* resides at the heart of this PCI card.

The PCI interface between the host processor and the local system is controlled by Mindspeed's Hardware Programming Interface (RS823xHPI), 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 RS8235. This software is written in C, and source code is available under license agreement.

The evaluation environment also includes a full set of design schematics, as well as artwork for the RS8235/825xEVM PCI card.



## 2.0 Architecture Overview

---

### 2.1 Introduction

The RS8235 *ServiceSAR* architecture enables it to efficiently handle high bandwidth throughput across the spectrum of three different ATM Adaptation Layers 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 Operation and Maintenance (OAM), 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 RS8235.

## **2.2 High Performance Host Architecture With Buffer Isolation**

Once initialized and given a segmentation or reassembly task, the RS8235 operates autonomously. Because the RS8235 is a high performance subsystem, the Host/ServiceSAR 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**

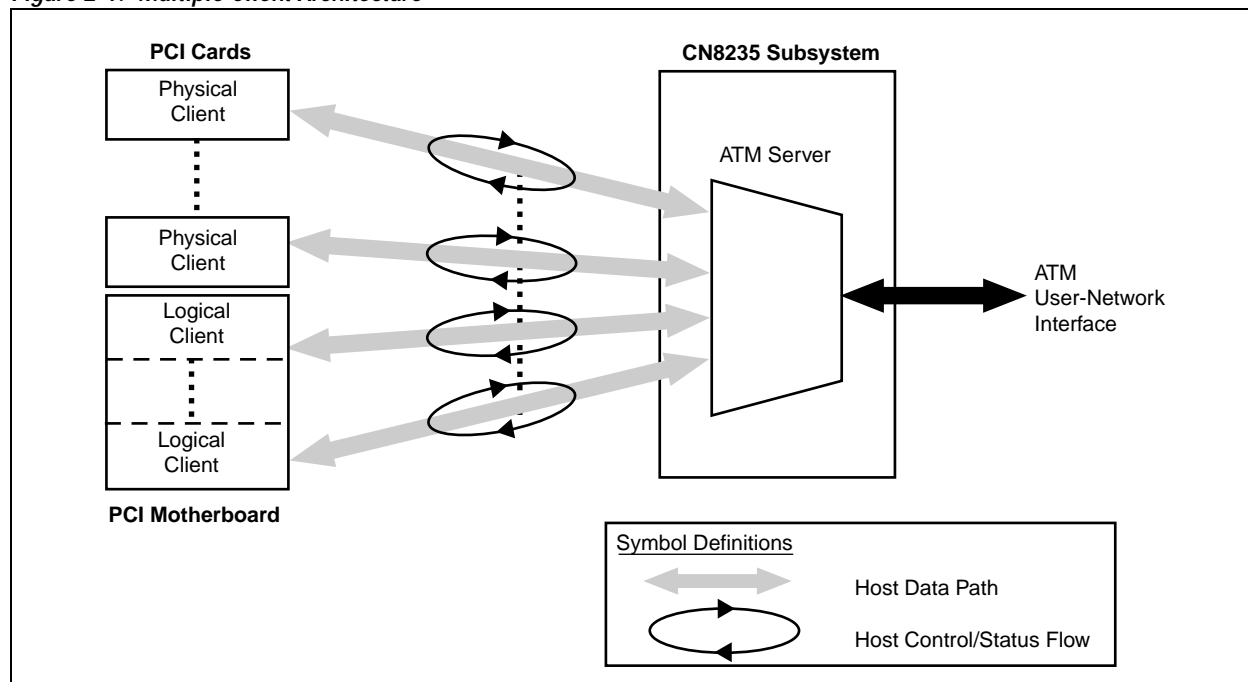
The RS8235, functioning as an ATM UNI, provides a high throughput uplink to a broadband network. Most individual ATM service users (or clients) do not have the bandwidth requirements to equal the throughput capability of the RS8235. As an application example, ATM clients can be Ethernet or Frame Relay ports. Therefore, several service clients are typically aggregated onto one ATM UNI. These clients can 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 RS8235 functions as an ATM server for up to four clients. In this way, the bandwidth requirements of the service user (the client) can be balanced with the service throughput capability of the RS8235 and the rest of the specific system.

The RS8235 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 can be assigned to each client.

Each client interfaces to the RS8235 independently. Due to its server architecture, the RS8235 supplies the synchronization between asynchronous tasks requiring ATM services. [Figure 2-1](#) illustrates this client/server model. It shows that clients can be multiple applications in a shared memory, or separate physical entities. All communicate directly with the RS8235.

Figure 2-1. Multiple Client Architecture



## 2.2.2 RS8235 Queue Structure

The flow of the reassembly, scheduling, and segmentation processes in the RS8235 is monitored, coordinated, and controlled through the use of a full array of circular queues, serviced by the RS8235 or by the host.

The following queues exist in local memory:

- transmit queues (up to four queues)
- reassembly/segmentation queue
- free buffer queues (up to four queues) — includes the Global OAM free buffer queue

[Figure 2-2](#) illustrates the location of each queue.

Transmit queues are used by the host to submit chains of segmentation buffer descriptors to the RS8235 for segmentation. The segmentation coprocessor then processes these transmit queue entries as part of the segmentation function.

The Reassembly/Segmentation queue is written to by the reassembly coprocessor and read by the segmentation coprocessor. The queue includes data on OAM cells to be transmitted, as well as data on ABR-class received Backward\_RM and Forward\_RM cells. The RSM/SEGqueue 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 can 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.

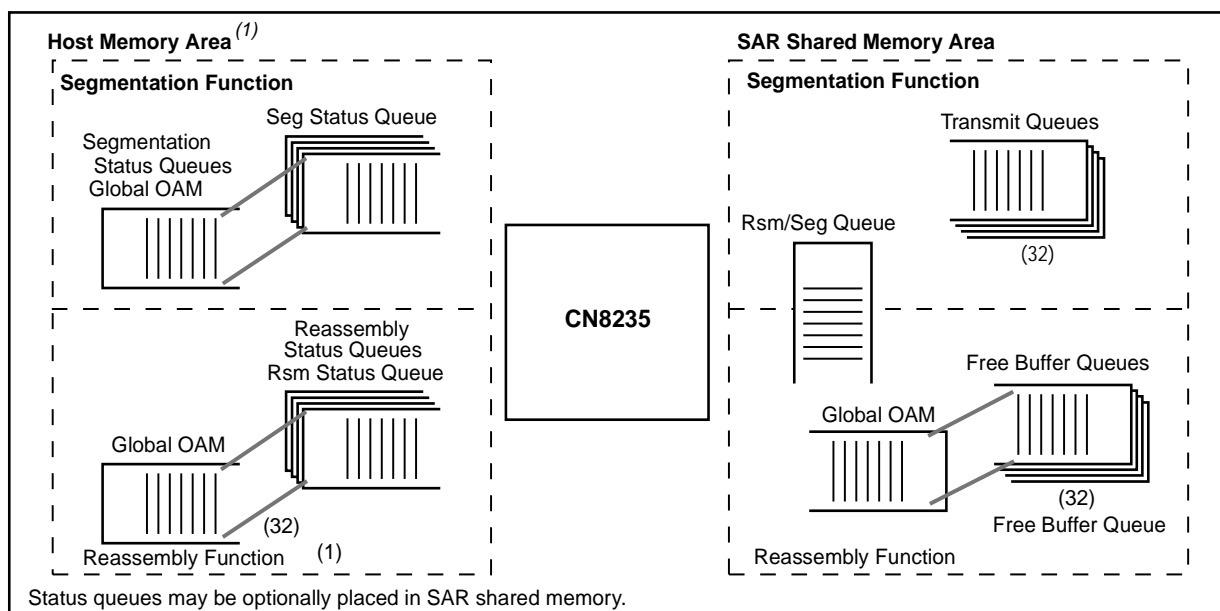
The following queues exist in host (or optionally, local) memory:

- segmentation status queues (up to four queues) — includes the Global OAM segmentation status queue
- reassembly status queues (up to four queues) — includes the Global OAM reassembly status queue

The RS8235 reports segmentation status to the segmentation status queues. One of these can be designated as the Global OAM segmentation status queue. The Host further processes these segmentation status queue entries.

The RS8235 reports reassembly status to the reassembly status queues. One of these can be designated as the Global OAM reassembly status queue. The Host further processes these reassembly status queue entries.

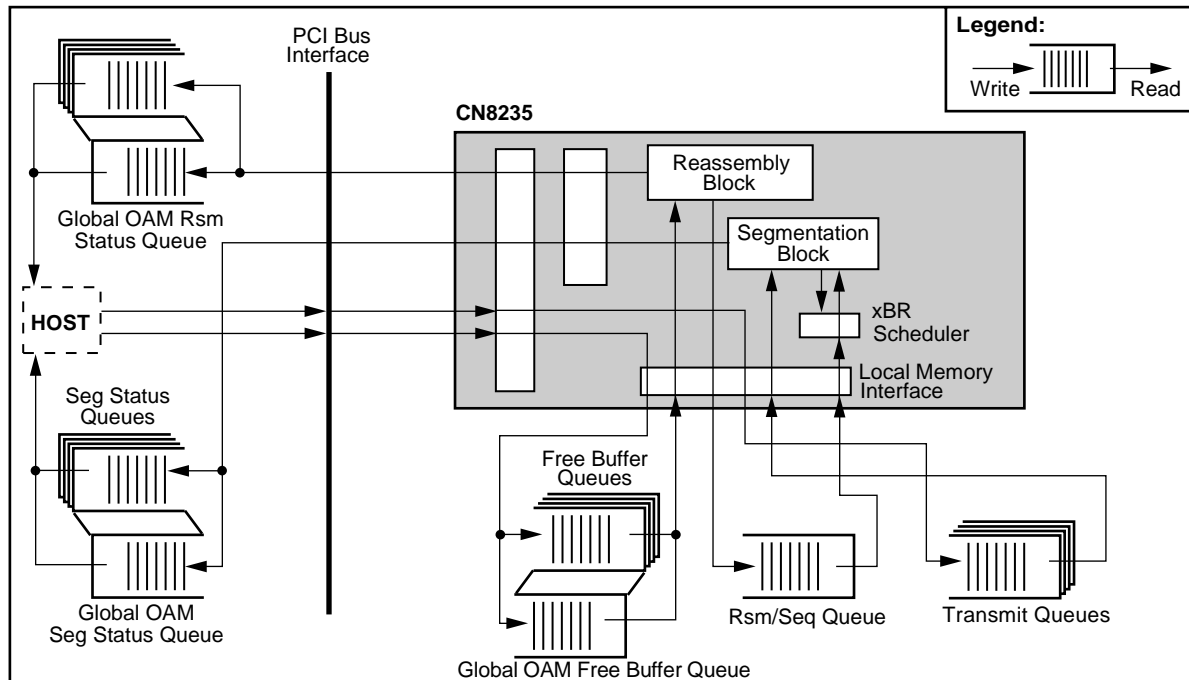
Figure 2-2. RS8235 Queue Architecture



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 RS8235 during processing, and associate each of the major functional blocks of the RS8235 with each other. Figure 2-3 illustrates these interactions. The arrows indicate which system entity writes to each queue, and which entity reads each queue.

Figure 2-3. Interaction of Queues with RS8235 Functional Blocks



### 2.2.3 Buffer Isolation Utilizing Descriptor-Based Buffer Chaining

The RS8235 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. This allows the data buffers to contain only payload data — no control fields or other user fields. The user can 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 *ServiceSAR*.

1. 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.
2. 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.
3. 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.
4. 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

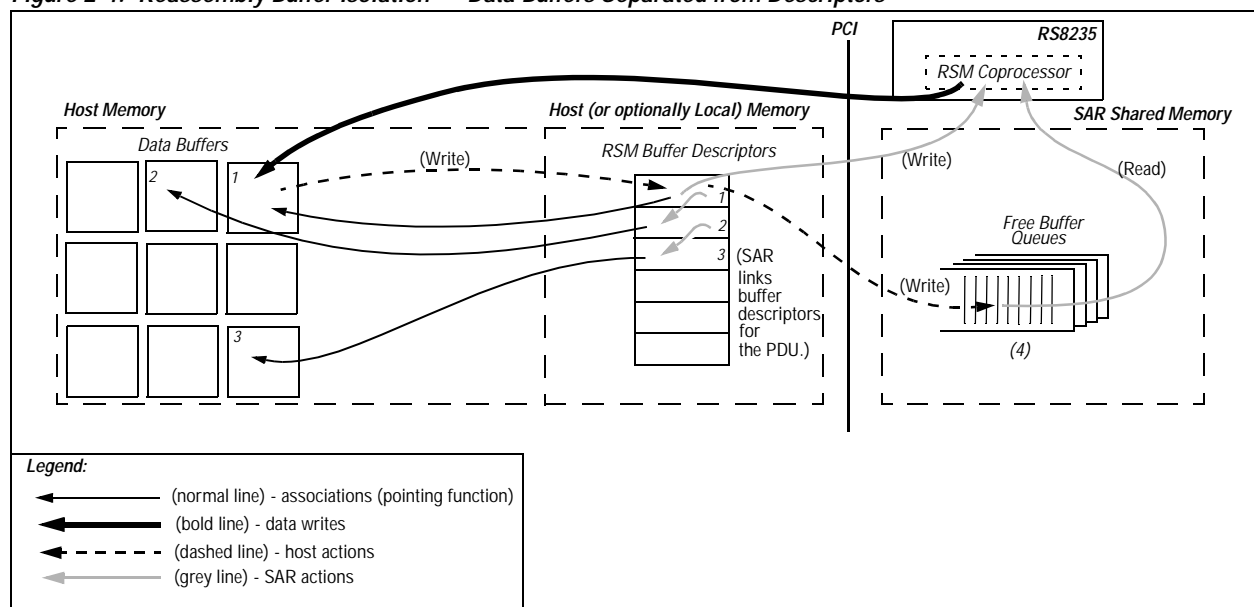
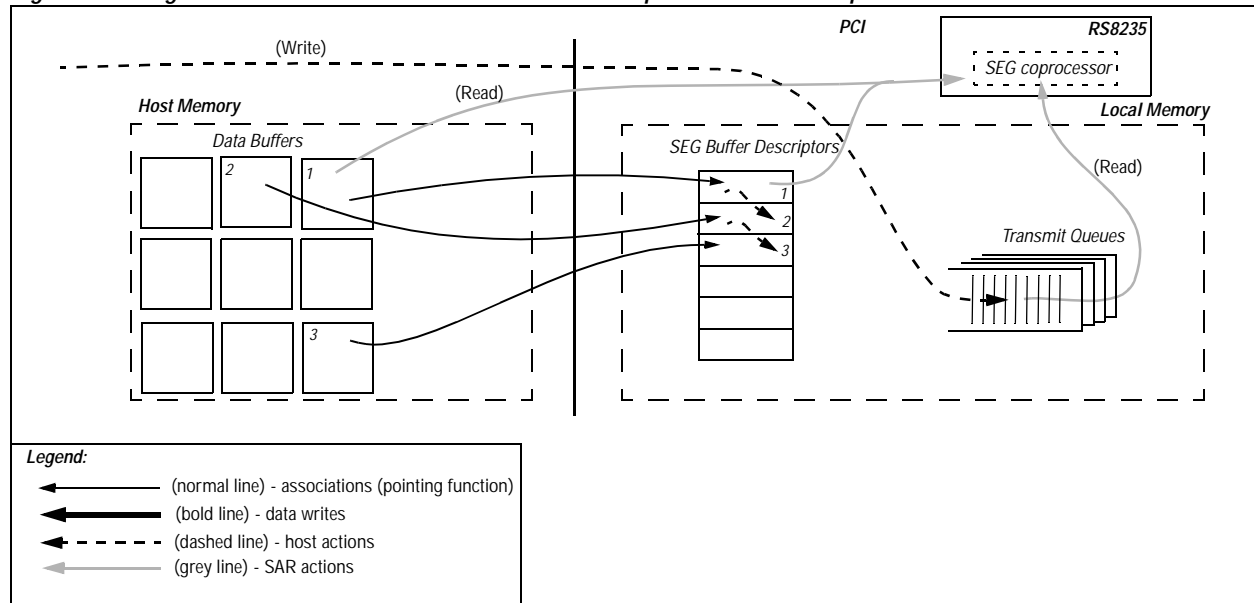


Figure 2-5 illustrates how the host submits linked data buffers (i.e., PDUs) to the transmit queue for segmentation. The process is as follows:

1. The host links the buffer descriptors (in SAR shared memory) for the associated data buffers (in host memory) containing the PDU to be segmented.
2. 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.
3. 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 RS8235 are written by the SAR and read by the host. These status queue entries provide the data needed by the host in order to further process the segmentation and reassembly data flow in progress or just completed. Each status queue entry thus includes data (such as error flags and status bits) which the host uses in its succeeding process steps. The SAR also includes, in each status queue entry, a pointer to the first buffer descriptor of the segmented or reassembled data buffer(s) which comprise a single PDU. This accomplishes the other principal function of a status queue entry: to establish the association from the SAR to the host of the successful or unsuccessful segmentation or 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 a single virtual channel, represented by a single entry in one of the transmit queues and a single entry in one of the SEG 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 SEG 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

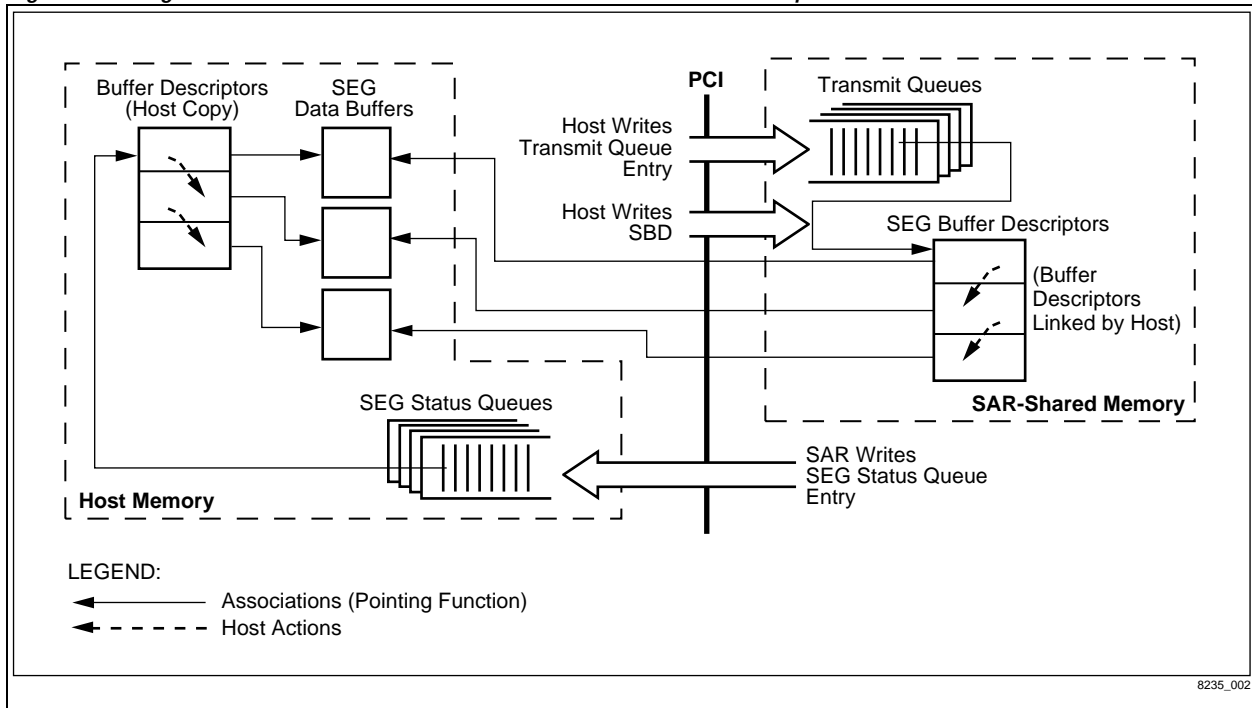
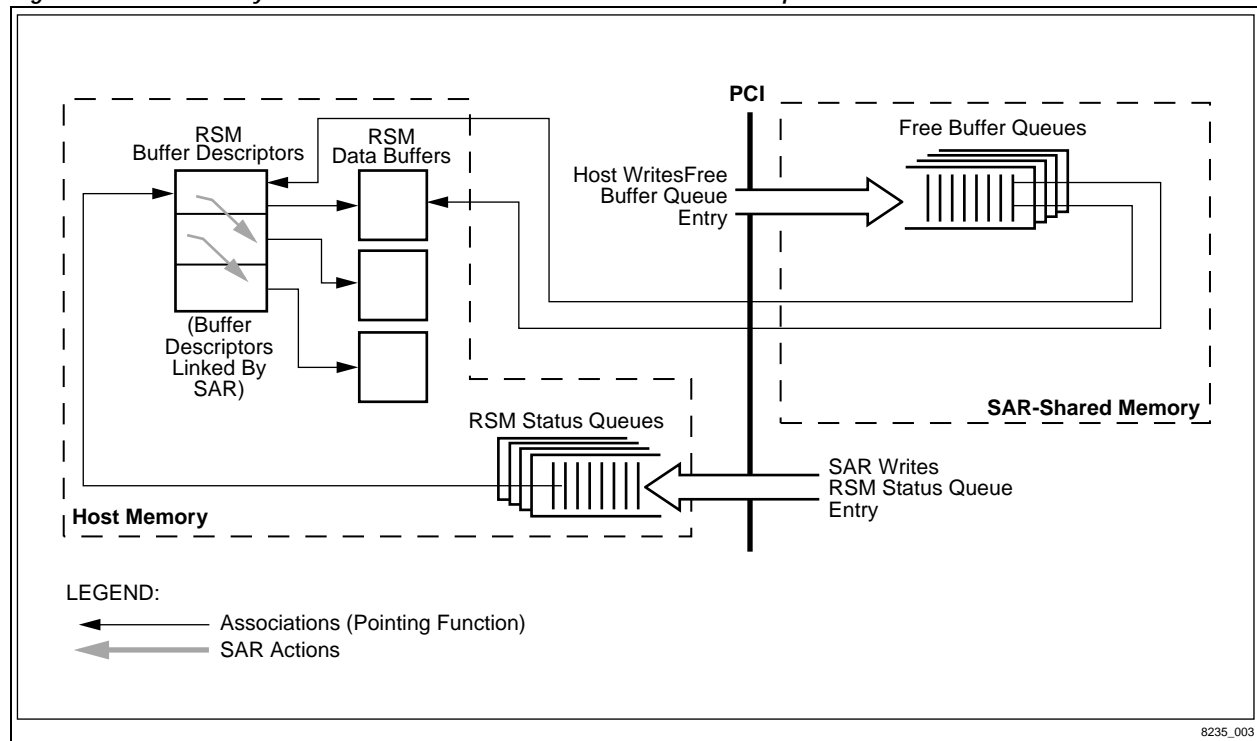


Figure 2-7 illustrates the association between the reassembly status queues and reassembly data buffers and descriptors. The host submits free buffers to the SAR by writing pointers to them in the free buffer queue entries. The SAR links the buffer descriptors which point to the three data buffers containing 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 using that data.

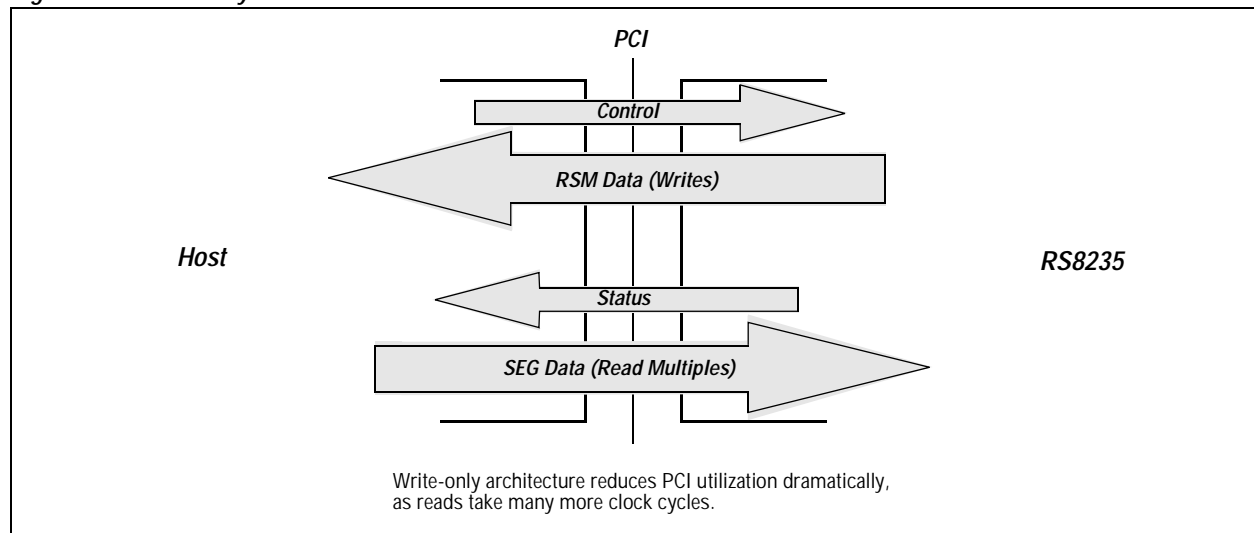
Figure 2-7. Reassembly Status Queues Related to Data Buffers and Descriptors



### 2.2.5 Write-Only Control/Status

Figure 2-8 illustrates the RS8235's write-only PCI control architecture. The host manages the RS8235 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 can 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 RS8235 eliminates read operations except for burst reads to gather segmentation data.

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



### 2.2.6 Scatter/Gather DMA

The RS8235's Direct Memory Access (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 13-word burst accesses for data
- 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.

## 2.2.7 Interrupts

The RS8235 informs the host of segmentation and reassembly activity by means of maskable interrupts sent to the host processor, triggered by writes to the segmentation and reassembly status queues.

The user can configure the SAR to generate these status queue entries and interrupts at PDU boundaries (called Message Mode), or at data buffer boundaries (called Streaming Mode).

The RS8235 can also be configured with a status queue interrupt delay, which can be enabled in order to reduce the interrupt processing load on the host. This has value when the SAR resides in an environment in which the host is not dedicated to datacom processing.

## 2.3 Automated Segmentation Engine

The RS8235 can segment up to 1 K 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 RS8235 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 SAR shared memory. A VCC specifies a single Virtual Circuit (VC) or Virtual Path (VP) in the ATM network. These VCC Table entries define the negotiated or contracted characteristics of the traffic for that channel, and 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 which point to the buffers containing the PDU to be transmitted, and then submitting that chained message to the SAR by writing to one of four 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 formatting functions include the following:

- The segmentation coprocessor formats the ATM cell header for each cell based on the settings in the Segmentation VCC Table entry for that VCC.
- The segmentation coprocessor also generates the CPCS-PDU header and trailer fields in the first and last cell of the segmented PDU.
- For AAL5 traffic, the SEG coprocessor also generates the PDU-specific fields in the trailer of the CPCS-PDU, and places these in the last cell (the End of Message [EOM] cell) for the PDU.
- 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 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 one to nine cells deep in order to optimize the balance between Cell Delay Variation (CDV) (which increases with longer transmit FIFO depth) and PCI latency protection (which decreases with shorter transmit FIFO depth).

The RS8235 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).

The RS8235 reports segmentation status to the host on one of a set of four independent parallel segmentation status queues. The RS8235 writes segmentation status queue entries 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 located in SAR shared memory. One RSM VCC Table entry defines the negotiated or contracted characteristics of the reassembly traffic for a particular channel. Each table entry is initialized by the host during system initialization, or on-the-fly. The SAR uses the RSM VCC Table to store temporary information to assist the reassembly process. An initialized Reassembly VCC Table entry effectively establishes a connection on which the RS8235 can reassemble data.

Using a dynamic Channel Directory lookup method, the RS8235 reassembles up to 1k VCCs simultaneously at a rate up to 200 Mbps on simplex connections and 155 Mbps on full duplex connections. The Channel Directory mechanism allows flexible preallocation of resources and provides deterministic channel identification over the full UNI or NNI Virtual Path Identifier/Virtual Channel Identifier (VPI/VCI) address space. The total number of VCCs supported is limited by the memory allocated to the RSM 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 RS8235 supports AAL5 and AAL0 reassembly, as well as 52-octet Raw Cell mode.

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

The RS8235 provides two methods of terminating an AAL0 PDU: Payload Type Identifier (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 RS8235 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 RS8235 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 one of eight 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 RS8235 implements an early packet discard feature to enable discarding of complete or partial CPCS-PDUs based upon service discard attributes or error conditions. The early packet discard function halts reassembly of the CPCS-PDU marked for discard until the next Beginning of Message (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 can be enabled for the following conditions:

- Frame Relay discard based on the frame's Discard Eligibility (DE) setting and the channel exceeding a user-defined priority threshold.
- CLP packet discard based on the received cell's CLP bit setting and exceeding channel priority threshold.
- LANE-LECID packet discard on ELAN channels which implements echo suppression on multicast data frames.
- Early packet discard on AAL5 channels when the reassembled PDU length exceeds the user-defined maximum PDU length for that VCC.
- Early packet discard 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).
- Early packet discard on PDUs when a DMA Incoming FIFO Full condition occurs.
- Early packet discard 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 RS8235 implements ATM's inherent robust traffic management capabilities for CBR, VBR, ABR, UBR, GFR and GFC. The RS8235 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 SAR 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 reserved CBR bandwidth, the RS8235 provides eight segmentation priorities. The user configures these priorities for the remaining service categories, including the TM 4.1 defined ABR class.

The RS8235's xBR Traffic Manager implements multiple functional levels of traffic prioritizing. This is illustrated in [Figure 2-9](#) below.

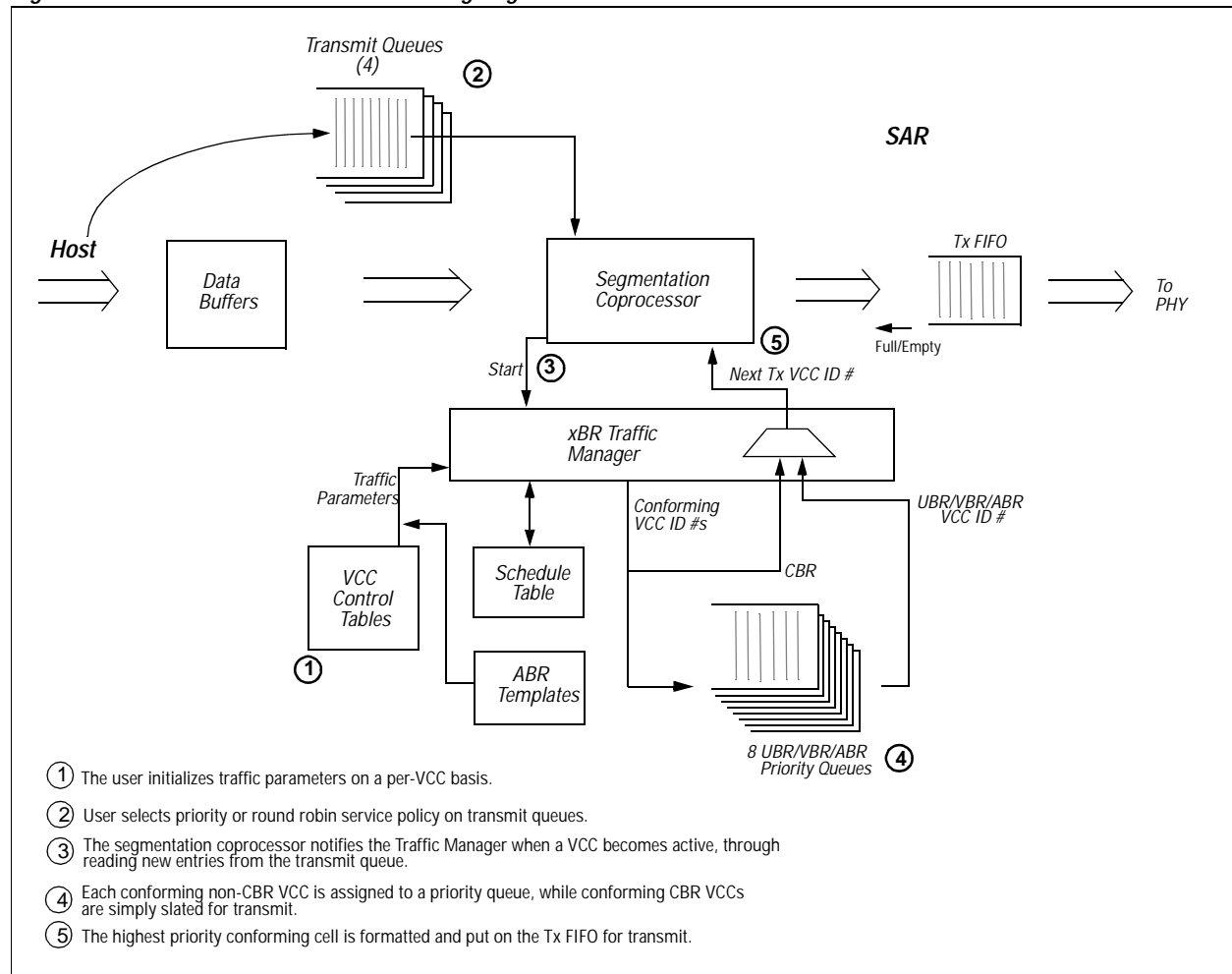
The host submits data to be sent by writing entries to the transmit queues for segmentation. The SAR processes these transmit queues either in round robin order (transmit queue zero through four, looped back to zero), or in priority order (with transmit queue four having highest priority). This scheme gives the user or system designer some control of the delay between the host submitting traffic and the SAR starting to process that traffic. For instance, the user could assign CBR traffic to the highest priority transmit queue in order to minimize any delay in processing and scheduling that traffic.

The RS8235 then submits this traffic demand to the xBR Traffic Manager for scheduling. Traffic is scheduled based on the traffic class plus certain parameters from the segmentation VCC Table entries (primarily the GCRA *I* and *L* parameters). And if the service category is ABR, the SAR also uses certain parameters from the ABR templates to help determine that traffic's placement on the Schedule Table.

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 (priority zero being the lowest priority and priority seven 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.

The RS8235 asynchronously multiplexes traffic based on the above schemes, as the transmit FIFO empties.

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



### 2.5.1 CBR Traffic

The CBR service category requires guaranteed transmission rates as well as constrained CDV. The RS8235 facilitates these needs when generating CBR traffic by pre-assigning specific schedule slots to CBR VCCs. For each CBR assigned cell slot, the RS8235 generates a cell for that specific VCC unless data is not available. The RS8235 also minimizes CDV by basing all traffic management on a local reference clock.

The RS8235 provides a mechanism to exactly match the scheduled rate of a CBR channel to the rate of its data source. The host might occasionally instruct the xBR Scheduler to skip one transmit opportunity on a channel in order to accomplish this rate-matching.

The RS8235 manages CBR tunnels in the same manner as a CBR VCC. However, instead of one VCC, several UBR, VBR or ABR VCCs can be scheduled within this CBR tunnel, in round robin order.

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

## 2.5.2 VBR Traffic

The RS8235 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 can be rate shaped as VCs or VPs according to one of three VBR definitions. VBR1 controls PCR and CDVT. VBR2 controls PCR and CDVT, as well as Sustained Cell Rate (SCR) and Burst Tolerance (BT). VBRC (also called VBR3) 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 RS8235 implements the ATM Forum ABR flow control algorithms. The RS8235 acts as both a fully compliant ABR Source and Destination, as defined in the TM 4.1 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 RS8235 employs an internal feedback control loop mechanism to enable the TM 4.1 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 RS8235 injects an in-rate stream of Forward RM cells for each ABR VCC. When these cells return to the RS8235’s receive port as Backward RM cells after a round trip through the network, the RS8235 processes these cells and uses the data returned as feedback to dynamically adjust the rates on each ABR channel.

The RS8235 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 SAR shared 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 GFR Traffic

Guaranteed Frame Rate is a new service category defined by the ATM Forum to provide a MCR QoS guarantee for AAL5 CPCS-PDUs not exceeding a specified frame length. A GFR service connection is treated as UBR with a guaranteed MCR.

The RS8235 implements GFR by scheduling/shaping the connections using both the VBR1 scheduling procedure (for the MCR rate value) and a UBR priority queue, thereby providing fair sharing for all GFR connections to excess bandwidth.

## 2.5.6 xBR Cell Scheduler

The xBR Scheduler slates traffic for transmission according to a Dynamic Schedule Table maintained in SAR 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 Mindspeed's proprietary per-VCC rate shaping algorithms. The predecessor to the RS8235, the Bt8230 SAR, proved the core algorithms. The RS8235 extends their use to other service classes. The RS8235 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 RS8235 dynamically manages the entire table.

Some 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 RS8235 fairly distributes the link bandwidth to channels based upon their QoS parameters and assigned transmission priority. As covered previously, the RS8235 supports eight priorities in addition to preallocated CBR time slots.

The xBR Scheduler facilitates advanced network traffic management topologies. The RS8235 rate shapes VCs or VPs. Additionally, CBR tunneling allows UBR, VBR and/or ABR traffic management schemes to operate under a preallocated CBR limit.

### 2.5.7 ABR Flow Control Manager

The ABR Flow Control Manager operates in conjunction with the xBR Scheduler to control the rate of ABR channels. The RS8235 implements the TM 4.1 specification in a template-controlled hardware state machine. Mindspeed provides an initial set of templates which reside in SAR shared memory. The information within these templates defines conformant ABR behavioral responses to network and connection states. The RS8235 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.

1. Since they control the Flow Control Manager state machine, they can be optimized for specific applications.
2. The programmability of the templates insulates the hardware from changes in the relatively stable, yet immature, TM 4.1 specification.
3. Mindspeed provides the initial templates, which can be customized by the user later, shortening development time.

## 2.6 Implementation of OAM Protocols

The RS8235 provides internal support for the detection and generation of OAM traffic. The RS8235 supports the F4 and F5 OAM flows according to I.610.

## 2.7 Standards-Based I/O

### *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, and contains the logic required to support arbitration for the PCI bus. This interface is PCI Version 2.1 compliant.

The PCI bus interface also includes an I<sup>2</sup>C Interface module that allows the PCI core to connect to a serial EEPROM. This 128-byte EEPROM is used to store specific PCI configuration information, loaded into the PCI Configuration space at reset. This allows several user-configurable features:

- User control of the size of the memory block for PCI addresses
- Enabling byte swapping of control words across the PCI bus
- Loading of Subsystem ID and Subsystem Vendor ID

### *ATM PHY I/O*

The RS8235's ATM physical interface communicates with and controls the ATM link interface device (which carries out all the transmission convergence) and physical media dependent functions defined by the ATM protocol. Two modes of operation are provided: standard UTOPIA and slave UTOPIA. 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.

### *SAR Shared Memory I/O*

To simplify system implementations, the RS8235 integrates a complete memory controller designed for direct interface to common Static RAMs (SRAMs). The RS8235's memory controller operates at 33 MHz and can access up to 8 MB 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 SAR shared memory architecture.

### *Boundary Scan I/O and Loopbacks*

The RS8235 includes five pins for Joint Test Action Group (JTAG) Boundary Scan, for board-level testing. The RS8235 incorporates an internal loopback from the segmentation coprocessor to the reassembly coprocessor to facilitate system diagnostics.

## 2.8 Electrical/Mechanical

The RS8235 is a CMOS device packaged in a 208-pin Plastic Quad Flat Pack (PQFP) format. It operates from a 3.3 V power supply and within the standard commercial range.

The device inputs are tolerant of 5 V signal levels, so external 5 V devices can be used. Any I/O (except for PCI) that requires a pullup must be tied through a resistor to 3.3 V and not 5 V.

## 2.9 Logic Diagram and Pin Descriptions

A functionally partitioned logic diagram of the RS8235 is shown in [Figure 2-10](#). Pin descriptions, names and input/output assignments are detailed in [Table 2-1](#).



Figure 2-10. RS8235 Logic Diagram (2 of 3)

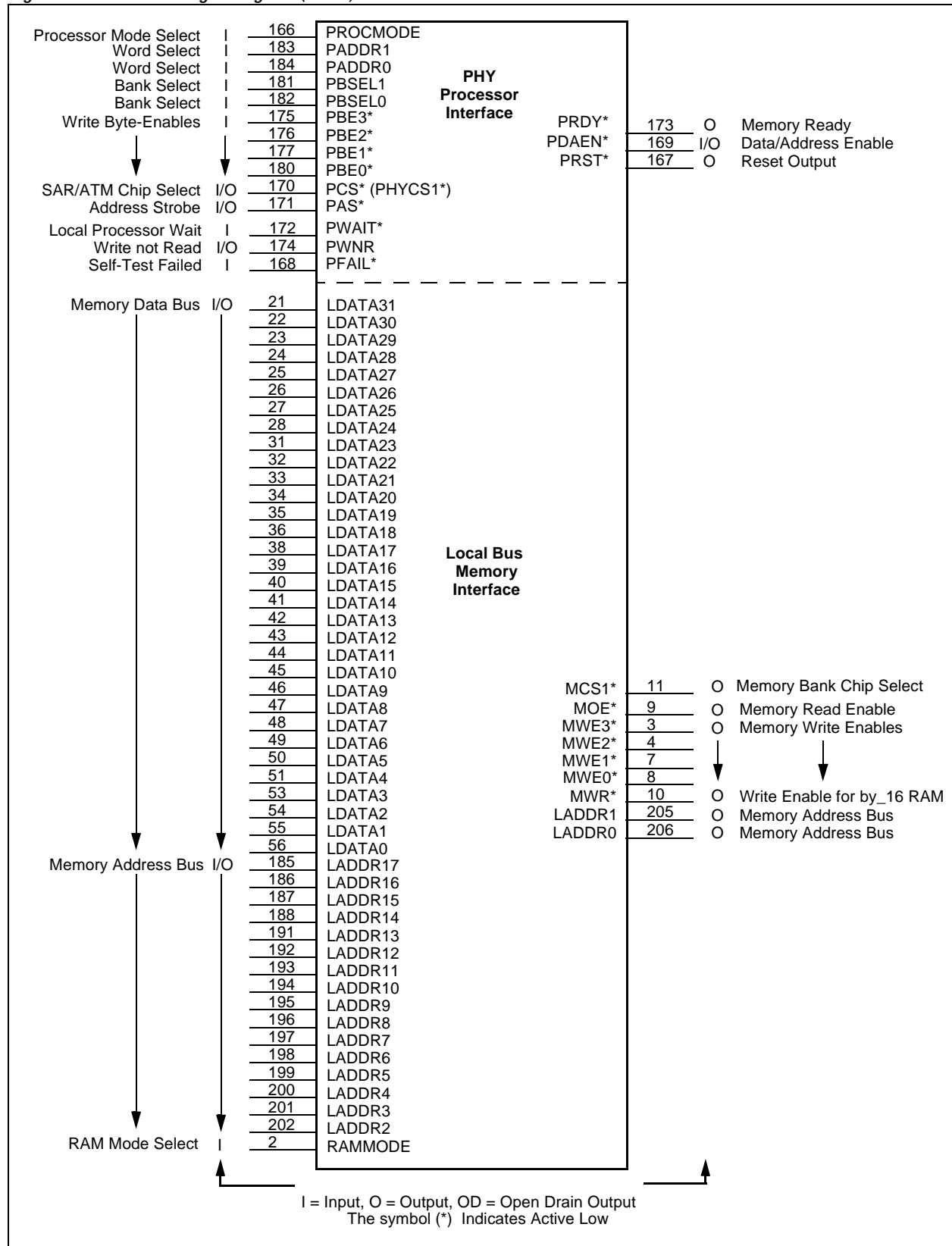


Figure 2-10. RS8235 Logic Diagram (3 of 3)

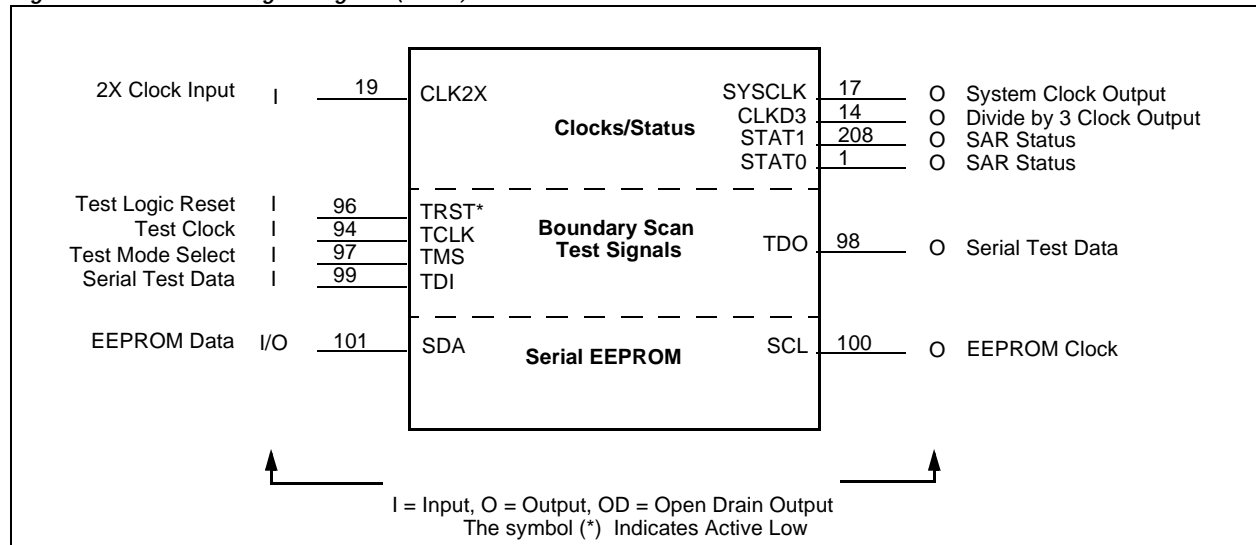


Table 2-1. Hardware Signal Definitions (1 of 5)

	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 RS8235 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 RS8235 is acting as a target) or driven (when the RS8235 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 RS8235 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 RS8235 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 RS8235 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 Ack	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 RS8235 (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 RS8235 that it is being selected for a configuration space access.
	HREQ*	Arbiter Bus Request	O	Asserted by the RS8235 to request control of the PCI bus.
HGNT*	Bus Grant	I	Asserted to indicate to the RS8235 that it has been granted control of the PCI bus, and can 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 5)

	Pin Label	Signal Name	I/O	Definition
Host 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 RS8235 (as a bus slave) or by a target addressed by the RS8235 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 RS8235 is a bus slave or sampled when the RS8235 is a bus master on the second clock edge after a valid data phase. The RS8235 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 RS8235 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 RS8235 and associated peripherals when asserted. Must be asserted for 16 cycles of HCLK.
	PCI5V	Bus Signalling	I	Must be tied high. This pad has an internal pullup resistor to VDD.

Table 2-1. Hardware Signal Definitions (3 of 5)

	Pin Label	Signal Name	I/O	Definition
ATM Physical Interface	FRCFG[1:0]	Framer Configuration	I	Configuration pins FRCFG[1,0] determine what framer interface the RS8235 supports. 00 = Reserved; do not use 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.
	TXSOC	Transmit Start of Cell	I/O	In both UTOPIA and slave UTOPIA modes, the TXSOC line is asserted by the RS8235 when the starting byte of a 53-byte cell is being output.
	TXCLAV*	Transmit Flag	I/O	In UTOPIA mode, TXCLAV* 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 RS8235 transmit buffer is empty.
	TXEN*	Transmit Enable	I/O	Indicates that valid data has been placed on the TXD[7:0], TXPAR, and TXSOC lines in the current clock cycle when the RS8235 is in UTOPIA or slave UTOPIA mode. This pin is an output in UTOPIA mode and an input in slave UTOPIA mode.
	RXD[7:0]	Receive Data	I	Transfers incoming data bytes from the link interface or framer chip to the RS8235 in all framer modes.
	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.
	RXSOC	Receive Start of Cell	I	Indicates that the current byte being transferred on the RXD[7:0] lines is the starting byte of a 53-byte cell.
	RXCLAV*	Receive Flag	I/O	In UTOPIA mode, RXCLAV* 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 RXSOC lines are invalid. In slave UTOPIA mode, this pin indicates to the framer chip that the receive FIFO in the RS8235 is full.
	RXEN*	Receive Enable	I/O	In UTOPIA mode, RXEN* indicates that the RS8235 is ready to receive data on the RXD[7:0], RXPAR, and RXSOC lines in the next clock cycle. This pin is an output in UTOPIA mode and an input in slave UTOPIA mode.
	RXCLK/TXCLK	Framer Control/Clock	I	In UTOPIA and slave UTOPIA mode, the RXCLK/TXCLK line should be driven with a clock that is synchronous to that used by the framer device for interfacing to the RS8235. The TXD[7:0], TXPAR, TXSOC, TXCLAV*, TXEN*, RXD[7:0], RXPAR, RXSOC, RXCLAV*, 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.

Table 2-1. Hardware Signal Definitions (4 of 5)

	Pin Label	Signal Name	I/O	Definition
PHY Processor Interface	PROCMODE	Processor Mode Select	I	Always tie to a logic high.
	PADDR[1:0]	Word Select Inputs	I	Always tie to ground.
	PBSEL[1:0]	Bank Select Inputs	I	Always tie to ground.
	PBE[3:0]*	Write Byte-Enables	I	Always tie to ground.
	PCS*	ATM PHY Chip Select	I/O	This pin can be connected to the chip select input of the Mindspeed PHY device.
	PAS*	Address Strobe	I/O	PAS* is used to drive the AS* pin of the Mindspeed PHY device.
	PWNR	Write/not Read	I/O	A logic one indicates a write while a logic zero indicates a read.
	PWAIT*	Processor Wait	I	Used by the external logic to insert wait states for read or write transactions.
	PRDY*	Memory Ready	O	Signals the cycle a data transfer occurs.
	PDAEN*	Data/Address Enable	I/O	This input is connected to the physical device's interrupt output(s).
	PFAIL*	Self-Test Failed	I	Always tie to a logic high.
	PRST*	Reset Output	O	Asserted by the RS8235 whenever the HRST* input is asserted, or when the LP_ENABLE bit in the CONFIG0 register is a logic low.
Local Bus Memory Interface	LDATA[31:0]	Memory Data Bus	I/O	Data I/O bus. Used for memory reads and writes.
	LADDR[17:2]	Memory Address Bus	I/O	Address I/O bus. Used for memory reads and writes.
	LADDR[1:0]	Memory Address Bus	O	The two least significant bits of address I/O bus. Used for memory reads and writes.
	MCS[1]*	Memory Bank Chip Select	O	Selects one addressable bank 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 Byte Write Enables	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*	Write Enable	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

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

	Pin Label	Signal Name	I/O	Definition
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 can be used for the UTOPIA interface clock. (22 MHz max)
	STAT[1:0]	SAR Status	O	RS8235 internal status outputs. Internal status controlled by the STAT_MODE[4:0] field in the CONFIG0 Register.
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. This pad has an internal pullup resistor to VDD.
	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. This pad has an internal pullup resistor to VDD.
	TDO	Serial Test Data	O	Outputs serial test pattern data.
	TDI	Serial Test Data	I	Input for serial test pattern data. This pad has an internal pullup resistor to VDD.
Serial EEPROM	SDA	EEPROM Data	I/O	Serial data.
	SCL	EEPROM Clock	O	Serial clock.
Supply Voltage	VDD	Power	–	Twenty pins are provided for supply voltage.
	VSS	Ground	–	Twenty-one pins are provided for ground.
	VGG	ESD Protection - Voltage Clamp	I	Enables ElectroStatic Discharge (ESD) protection. When the SAR is connected to 3.3 V power supply yet there are 5.0 V devices on the board, tie this pin to 5.0 V for ESD Protection. Otherwise, tie to 3.3 V.

## 3.0 Host Interface

---

### 3.1 Overview

The RS8235 segments and reassembles user data packets at 200 Mbps simplex, and over 155 Mbps 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 RS8235's high throughput, the host must process control and status information at a comparable rate.

#### **APPLICATION EXAMPLE**

An Ethernet Switch uses a RS8235 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 RS8235 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.

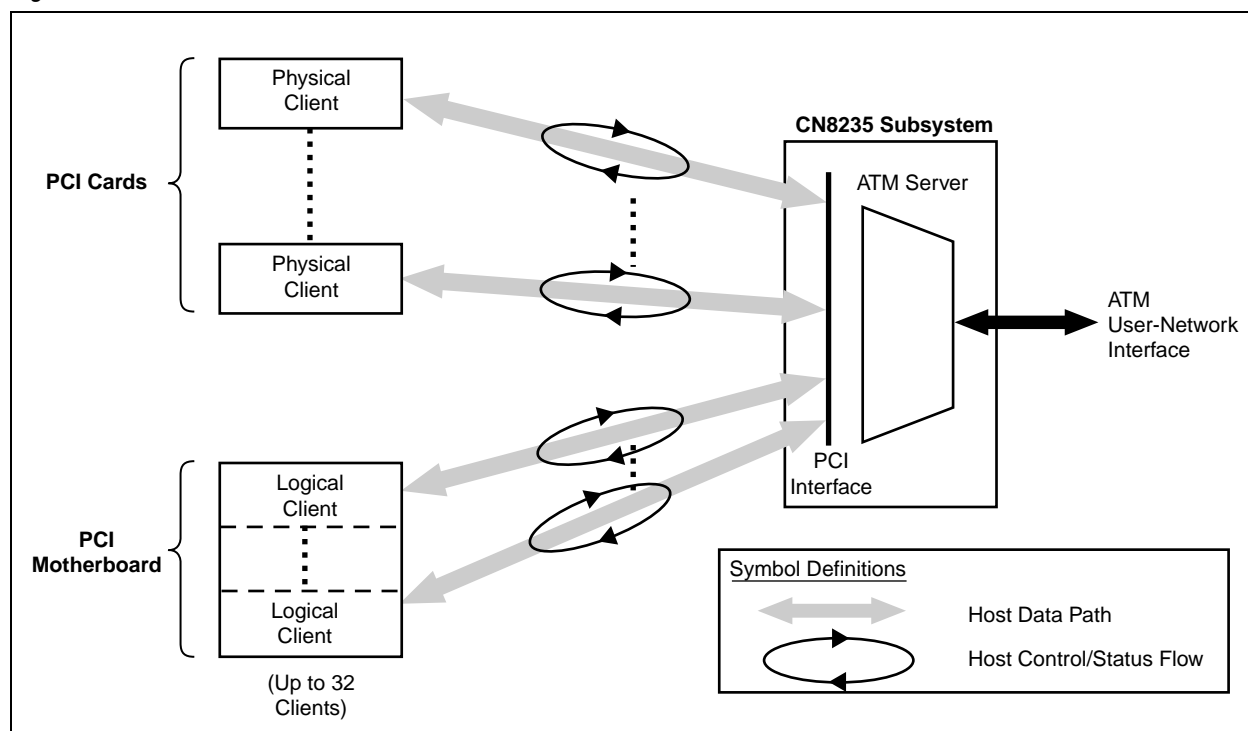
In cases such as the example above, the service rate places extraordinary performance requirements on the host system. High throughput systems require large numbers of processing cycles and efficient utilization of system buses.

The RS8235 provides a flexible, high performance host interface architecture. With this interface, the RS8235 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 RS8235 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. The RS8235 provides an ATM server for up to four clients. [Figure 3-1](#) illustrates this client/server relationship.

Figure 3-1. Client/Server Model of the RS8235



### 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 RS8235 independently. Due to its server architecture, the RS8235 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 RS8235 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 SAR 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 RS8235 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.

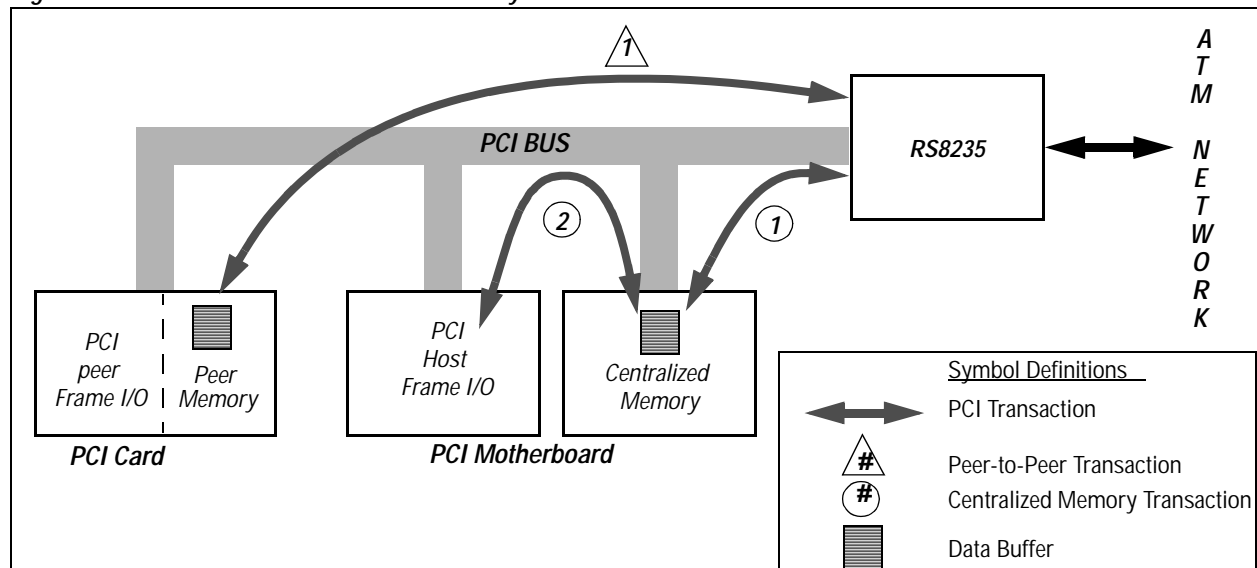
#### **APPLICATION EXAMPLE**

A system designer implements a RS8235 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 RS8235 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 RS8235's reassembly status queue. Eventually, the RS8235 fills the queue and is unable to proceed -- this situation is referred to as queue overflow. The RS8235 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 RS8235 also enables peer-to-peer PCI transfers. The RS8235 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 RS8235 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 that 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



### 3.3 Write-Only Control and Status

For host-based applications, the host manages the RS8235 SAR using write-only control and status queues. 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 can “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 RS8235’s write-only architecture uses reads only for segmentation data (PDU) returns. 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 chapters.

[Table 3-1](#) defines the RS8235 control and status queues.

*Table 3-1. RS8235 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 of PDU data for segmentation on the transmit queues, and supplies empty 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 SAR shared memory at a location defined by a base register pointer. To allow multiple clients, the RS8235 supports four 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

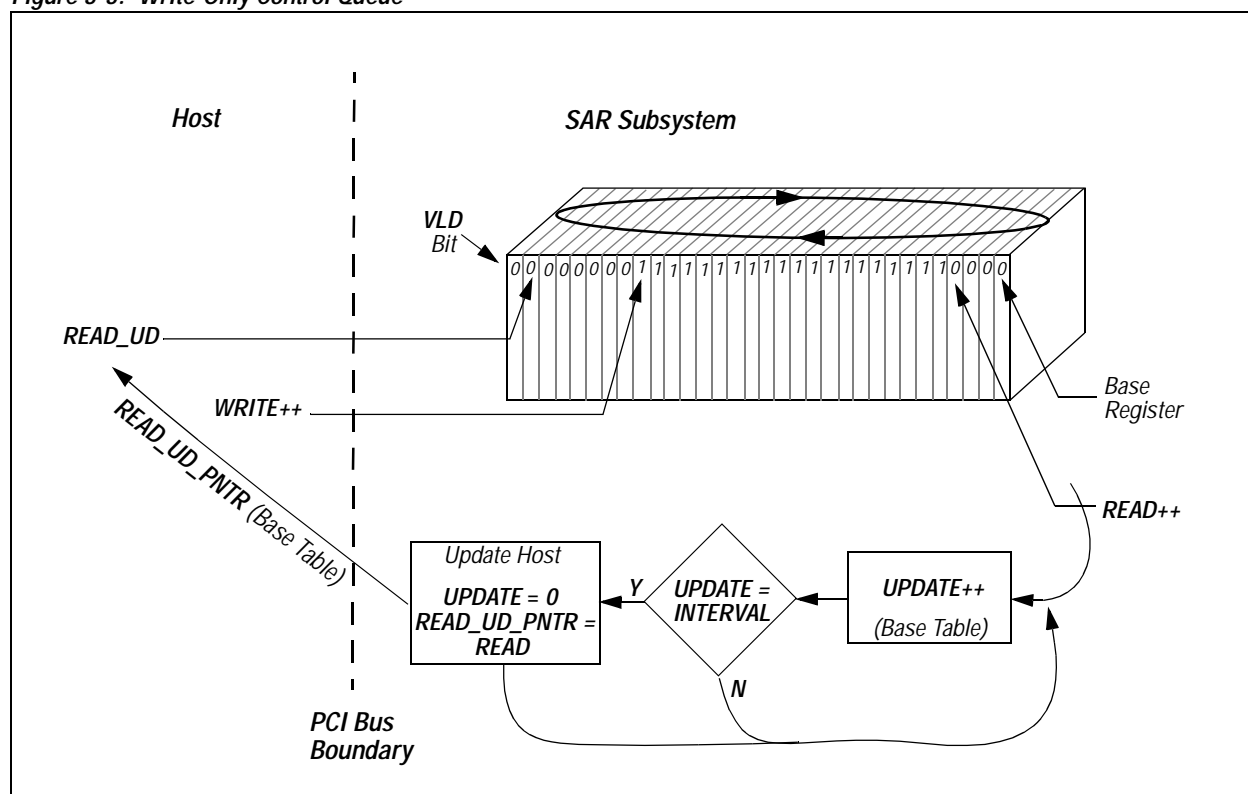
Variable	Definition	Location	Initialization
WRITE	Current host position in queue	Host variable	0
READ_UD	Last known SAR position in queue as seen by host	Host word-aligned variable	0
READ	Current SAR position in queue	SAR Base Table	0
INTERVAL	Number of queue entries processed by SAR before writing READ_UD	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 3-3 illustrates the control queue management algorithm. The host initializes all of the variables described in

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 3-3. 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\_UD, then the host should halt writing to the queue. This results in being able to use only N-1 queue entries. However, if this is not done, then a full condition cannot be distinguished 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 RS8235 “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 RS8235 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 happens only 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. [Chapter 5.0](#) 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 RS8235 manages all status queues with the same algorithm.

These queues reside in host memory, or optionally SAR 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 RS8235 provides four 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

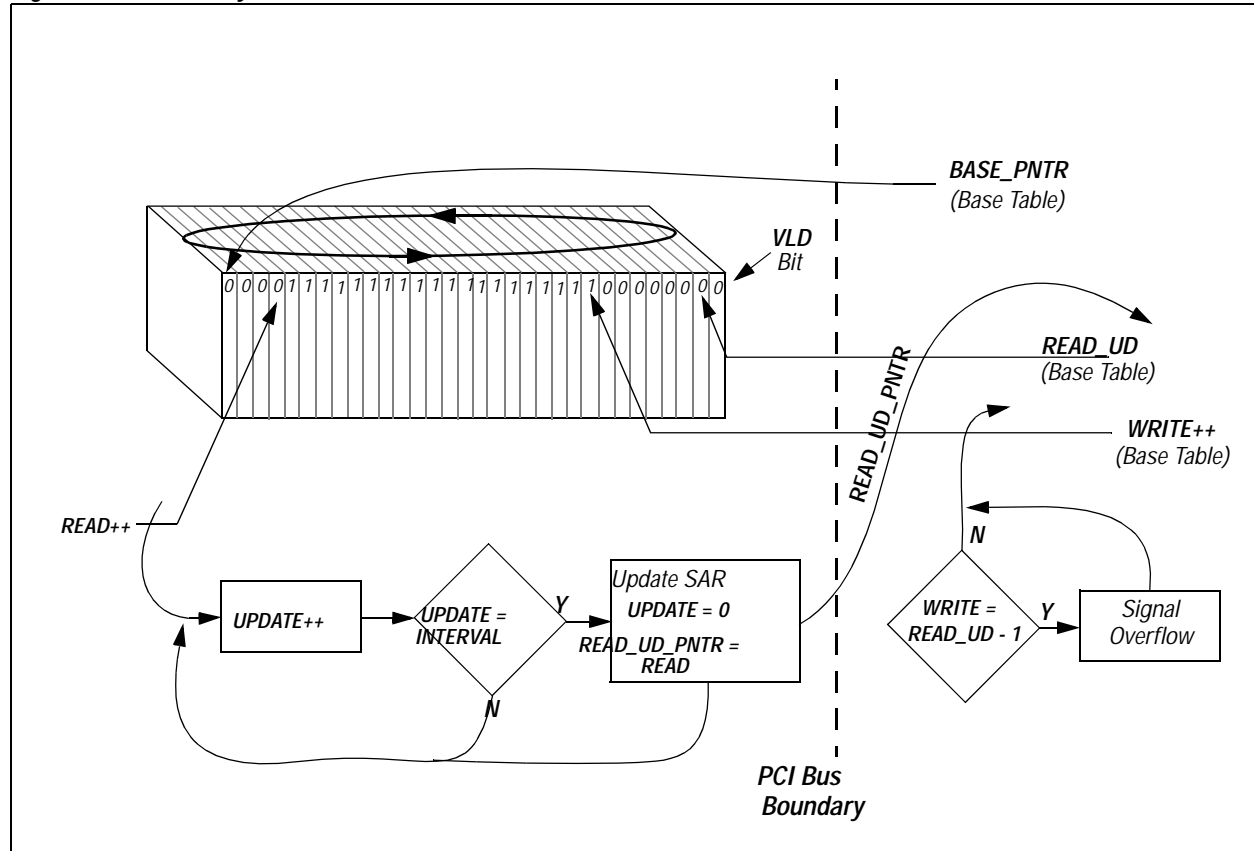
### 3.3.2.2 Queue Management

Figure 3-4 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 RS8235 reports status to the host by writing a status queue entry. After it writes the entry, the RS8235 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 VLD bit equal to one when it writes a status queue entry. The host resets it to zero after processing an entry.

Figure 3-4. Write-Only Status Queue



The host also maintains a pointer (READ) into the status queue. 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 can happen for both the segmentation and reassembly status queues. The Segmentation and Reassembly chapters describe the handling of this event. In either case, the result is severe and therefore undesirable. The host control service rate of the status queue should match or exceed the status queue reporting rate of the RS8235.

The RS8235 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 RS8235 segmentation and reassembly processing is temporarily halted for VCCs assigned to the overflowed status queue only. All other processes and queues remain operational.

#### 3.3.2.4 Status Queue Interrupt Delay

Status Queue Interrupt Delay has been added in order to reduce the interrupt processing load on the host. This is valuable in a Network Interface Card (NIC)-based solution, where the SAR resides in an environment in which the host is not dedicated to datacom processing. Both a timer holdoff mechanism and an event counter mechanism are implemented and work in parallel. The timer holdoff mechanism uses the ALARM1 and CLOCK register resources to implement an interval timer. Interrupts due to status queue writes, either host or local, are delayed until the timer expires. The event counter mechanism delays the assertion of the interrupt due to status queue writes until a fixed number of status queue writes have occurred. Both mechanisms work in parallel (not in series) if enabled, so that either mechanism needs to expire before the interrupt propagates to the output pin. Note that interrupts due to conditions other than status queue writes are not delayed. (See the table in [Section 12.6](#) for specific information on the Interrupt Delay (INT\_DELAY) register.)

##### *Timer Holdoff Mechanism*

The timer holdoff mechanism is enabled by setting INT\_DELAY (EN\_TIMER) to a logic high. The ALARM1 register is set to a value that will hold off the interrupt for a specified period of time. The user initializes the CLOCK register to zero. When the value in the CLOCK register is greater than the value in the ALARM1 register, status queue interrupts will be allowed to propagate to the appropriate interrupt pins, HINT\* or PINT\*. The CLOCK register is set to zero once an interrupt has propagated to the output pin, thus closing the status queue write interrupt window. The timer mechanism cannot be used in both the PINT\* and HINT\* circuits at the same time. The timer mechanism is configured via the INT\_DELAY (TIMER\_LOC) bit.

##### *Event Counter Mechanism*

The event counter mechanism is enabled by setting INT\_DELAY (EN\_STAT\_CNT) to a logic high. An internal counter is implemented that counts the number of status queue write events. The number of events before opening the interrupt window is programmable via the INT\_DELAY(STAT\_CNT) field. The window will be closed for STAT\_CNT number of events. When the internal counter has reached the value of STAT\_CNT, the interrupt window will be opened, which will allow the interrupt to propagate to the output pin. The counter is reset when the status registers are read and the interrupt output goes inactive.

## 3.4 System Clocking

The RS8235 derives all of its timing from a 2x clock input, CLK2X. This clock is internally divided by two to create the system clock, SYSCLK. This 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, CLKD3 (CLK2X asymmetrically divided by three, an output clock), is provided, and can 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 would be looped externally to the RXCLK/TXCLK 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.

## 3.5 RS8235 Reset

The RS8235 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)
- 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 RS8235 initialization process.



## 4.0 Segmentation Coprocessor

---

### 4.1 Overview

ATM's cell transport mechanism enables large numbers of virtual channels or 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 RS8235 can segment 1 K VCCs simultaneously. The segmentation coprocessor block independently segments each channel and multiplexes the VCCs onto the line with cell level interleaving. The RS8235 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 RS8235 generates all AAL overhead for AAL5, or optionally utilizes a null adaptation layer, AAL0. The RS8235 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 RS8235 supports up to approximately 1 K segmentation VCCs, referenced by a unique index, VCC\_INDEX. The actual number of VCCs that can be supported by the device is limited by the amount of local memory available (1 MByte). The 1 K segmentation VCCs are thus an approximation. 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.

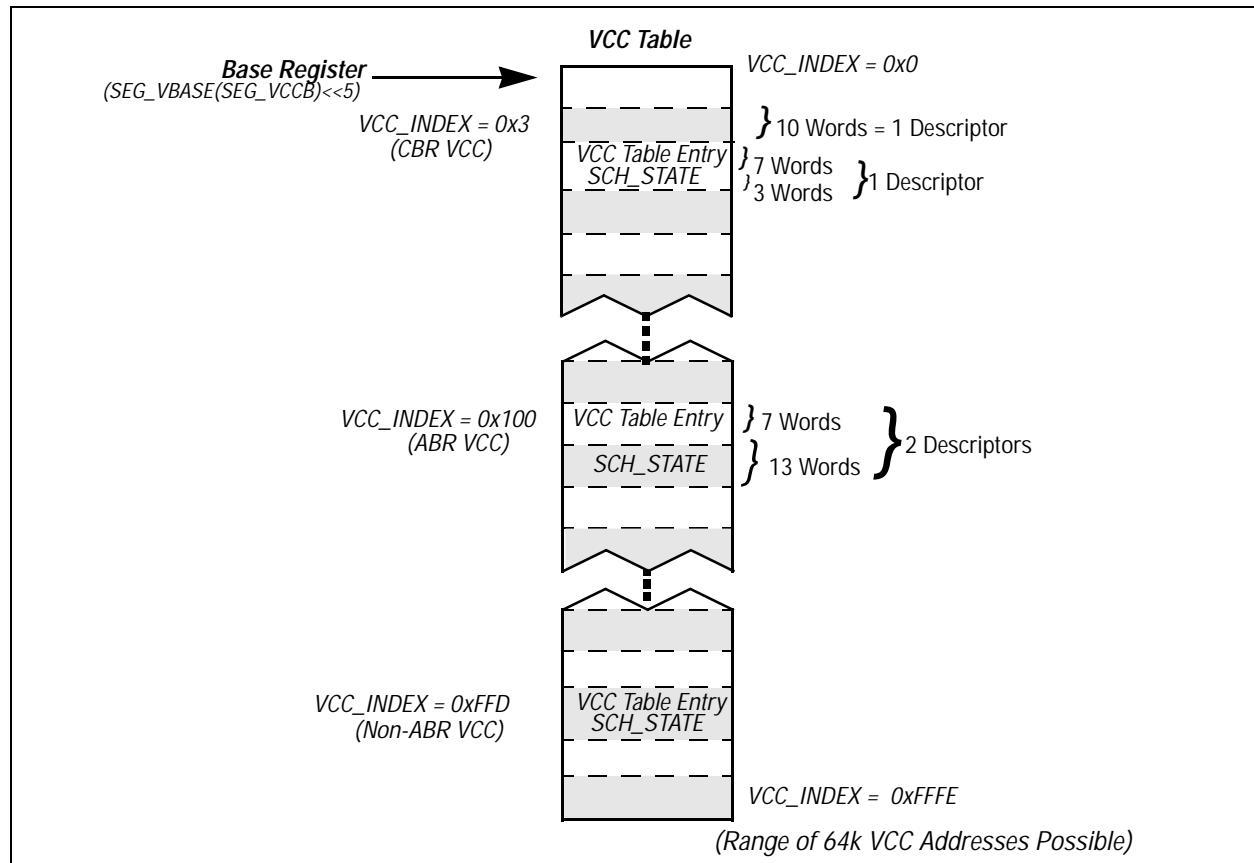
A Segmentation VCC Table channel descriptor consists of two parts: an AAL specific VCC Table Entry (seven words) and a service class specific Schedule State (SCH\_STATE). For non-ABR VCCs, the remaining three words of the channel descriptor form the SCH\_STATE entry. For an ABR VCC, the SCH\_STATE entry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

#### 4.2.1.1 Segmentation VCC Table

Figure 4-1 shows a VCC Table with a CBR VCC at VCC\_INDEX 3, an ABR VCC located at VCC\_INDEX 0x100, and a non-ABR VCC at VCC\_INDEX 0xFFD. The host allocates the VCC Table in SAR shared memory and provides an address to the SAR in a base register field, SEG\_VBASE(SEG\_VCCB). 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 32 K of VCC addresses. Valid VCC indexes for CBR traffic range from 0x0000 to 0x7FFE, due to the limit imposed by a 15-bit address. A VCC index of 0xFFFF for non-CBR and 0x7FFF for CBR indicate a NULL VCC.

While the RS8235 will accept any VCC Index within the range of 64 K VCC Indexes, the actual number of segmentation VCCs allowed by the SAR is limited by the amount of local memory available in which to allocate and create SEG VCC Tables, transmit queues, etc.

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 can modify during the segmentation process. See [Section 4.3.1](#) for full details of the structure of a SEG VCC Table entry.

#### 4.2.1.2 VCC Identification

The host allocates a region of SAR 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 RS8235 accepts user PDUs as sequences of data buffers. SAR shared memory resident segmentation buffer descriptors (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 SAR shared) memory. Host data buffers contain the bulk of segmentation data, and can begin on any byte-aligned address in the SAR's PCI address space. (Note: SAR shared memory data buffer segmentation should be limited to low bandwidth applications, such as Signalling, OAM, and ILMI).

### 4.2.2.2 Buffer Descriptors

The host submits data using the following process sequence. First, the host allocates an SBD for each buffer in a message. SBDs reside in SAR 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. The host identifies the end of the SBD chain by terminating the list with a NULL (0) NEXT pointer. [Table 4-1](#) shows the appropriate utilization of these bits in detail.

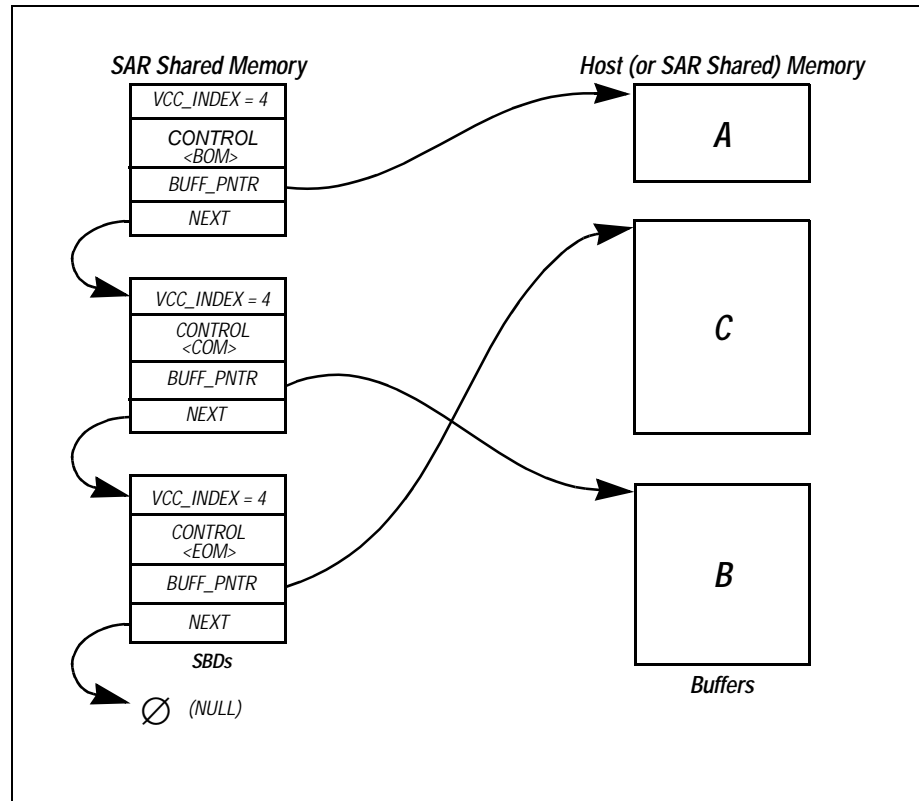
**Table 4-1. Segmentation PDU Delineation**

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.

#### 4.2.2.3 Host Linked Segmentation Buffer Descriptors

Figure 4-2 shows an example of SBD chaining. The host has formed a three-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. Segmentation Buffer Descriptor Chaining



#### 4.2.2.4 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 four 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. The host processor writes a pointer to the SAR 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 RS8235 detects this write by “snooping” SAR shared memory accesses. When a write occurs to any of the transmit queues, the RS8235 marks that queue as pending. Once every cell slot, the RS8235 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 one, the RS8235 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 zero, the RS8235 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 four 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 can submit data to a VCC while the SAR is segmenting a previously submitted message. Once the chain has been linked, the RS8235 resets the transmit queue VLD bit to zero.

**SAR Transmit Queue Processing**

Figure 4-3 and Figure 4-4 illustrate the process of linking SBDs to a VCC Table entry. Note that as the new buffers are submitted, the VCC is processing a single buffer PDU (BOM/EOM). The RS8235 accepts new PDUs while it is processing outstanding buffers.

Figure 4-3. Before SAR Transmit Queue Entry Processing

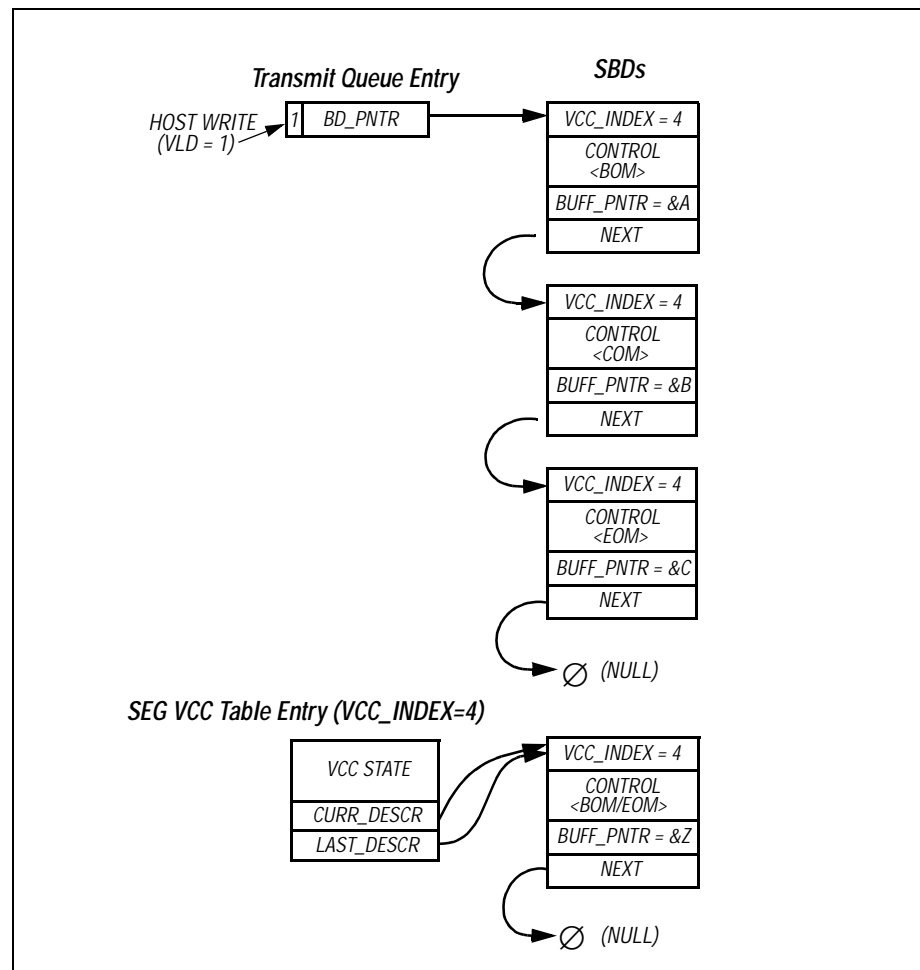
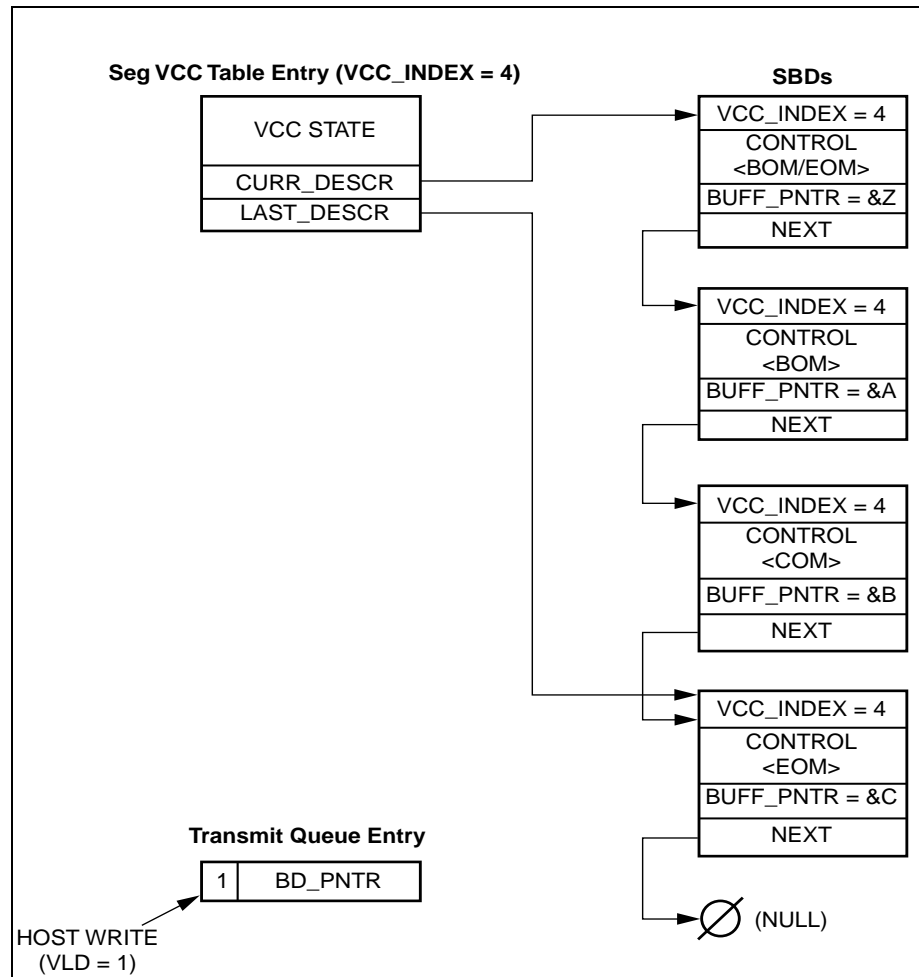


Figure 4-4. After SAR Transmit Queue Entry Processing



#### 4.2.2.5 Partial PDUs

The host can submit partial PDUs to the RS8235. 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 RS8235 tracks the remaining segmentation data. If a partial cell remains, the RS8235 will hold the buffer until it can complete a cell. Once the host submits an additional buffer, the RS8235 resumes segmentation.

#### 4.2.2.6 Virtual Paths

For network management simplicity, the host can create VP 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 RS8235 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 RS8235 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 RS8235 adds the CPCS-PDU protocol fields to the CPCS-SDU. The SAR supports two AAL levels: AAL5, and a transparent adaptation layer (AAL0).

Specific features also allow the generation of OAM cells. Chapter 7 covers OAM generation in detail.

#### 4.2.3.1 AAL5

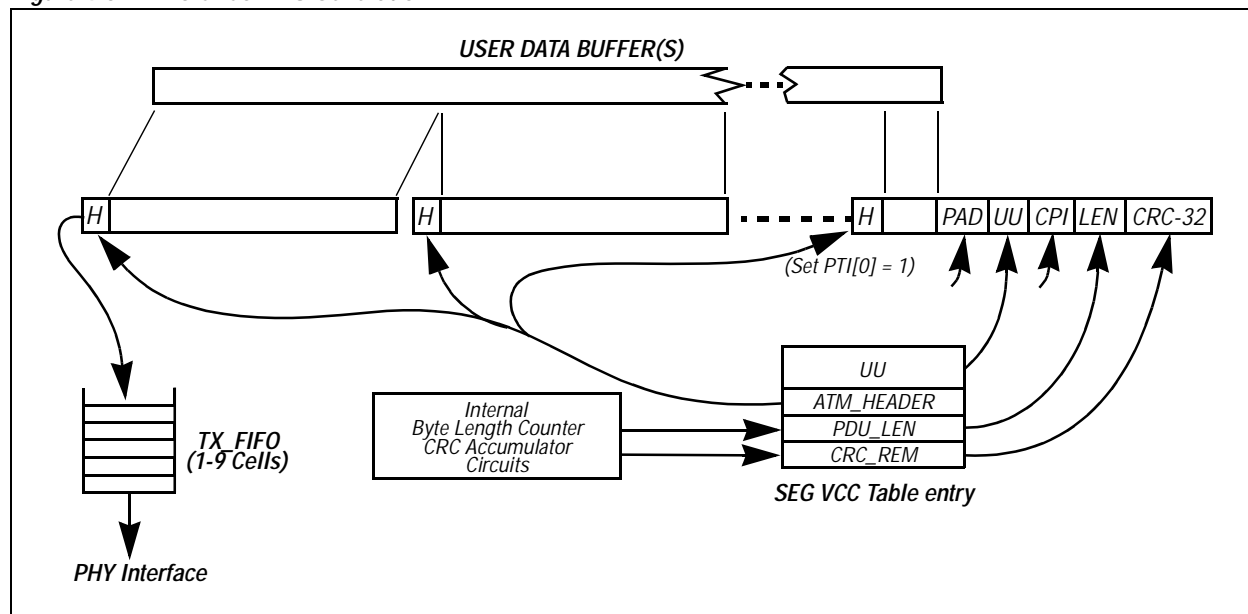
For AAL5, the SAR generates the CPCS-PDU trailer, and pads the CPCS-SDU to align the PDU to a cell boundary. The RS8235 generates the PAD, Length (LEN), Common Part Indicator (CPI), and Cyclic Redundancy Check (CRC) fields according to I.363. The host supplies the CPCS User-to-User Indication (UU) field in the first buffer descriptor in a message, and the RS8235 transmits it, following I.363.

The SAR generates the ATM header according to host-initialized settings in the VCC Table Entry. The RS8235 terminates AAL5 PDUs by setting bit zero of the Payload Type Identifier field, PTI[0] = 1.

The host aborts PDUs by activating an EOM SBD abort option. The host activates this by setting the following fields in the SEG buffer descriptor entry to these values: AAL\_MODE = AAL5 (b00), and AAL\_OPT = ABORT (b01).

Figure 4-5 shows the RS8235'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 RS8235 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 RS8235 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 in the same manner as it does with AAL5.

## 4.2.4 ATM Physical Layer Interface

Once the segmentation coprocessor has formed an ATM cell, the RS8235 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](#) 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 RS8235 informs the host of segmentation completion using segmentation status queues. The host assigns each VCC to one of four status queues, enabling a multiple peer architecture as described in [Section 3.2](#). The RS8235 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, though this is a rare occurrence within a RS8235 subsystem's segmentation block. The segmentation status queues operate according to the write-only host interface, defined in [Section 3.3.2](#).

The RS8235 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 might contain the address of the buffer or of a host data structure describing the buffer. To simplify host management, the RS8235 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 RS8235 provides an optional method to segment from a fixed PCI address, or Virtual FIFO. The RS8235 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 (i.e., attaches to the beginning) the ATM\_HEADER value in the VCC Table entry. The host does not use the transmit queue for Virtual FIFOs. The RS8235 transmits cell payloads from this location indefinitely, with no status reporting.

## 4.3 Segmentation Control and 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 13 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 format completely describes the state of the segmentation process for individual VCCs. [Table 4-2](#) and [Table 4-3](#) describe the format of AAL5 and AAL 0 VCC Table Entries.

#### KEY:

	= Written by Host at VCC setup
	= May be dynamically modified during active segmentation

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	Rsvd							Rsvd				LAST_PNTR																				
1	Reserved							Rsvd				Reserved																				
2	ATM_HEADER																															
3	Reserved																Reserved															
4	CRC_REM																															
5	STM_MODE	Rsvd				STAT	Rsvd	Rsvd	Rsvd	Rsvd	CURR_PNTR																					
6	Rsvd	VPC	SCH_GFR/CBR_W_TUN		Rsvd	GFR_PRI			SCH_MODE			Rsvd	PRI		SCH_OPT		Reserved															
7-9/19	SCH_STATE																															

Table 4-3. Segmentation VCC Table Entry — AAL5 and 0 Field Descriptions (1 of 2)

Field Name	Description
LAST_PNTR	Points to last buffer descriptor currently linked to the VCC. The two least significant bits of the pointer are assumed to be zero (word-aligned). The address of the buffer descriptor is (LAST_PNTR << 2) or (LAST_PNTR x 4).
ATM_HEADER	Used for each ATM cell for the VCC. The transmitted header can 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 and 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.
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 x 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_GFR/ CBR_W_TUN	If SCH_MODE = GFR, this bit set to a logic high indicates that the VCC is currently scheduled on a GFR_PRI priority queue for segmentation. The SCH bit set to a logic high indicates that the VCC is currently scheduled on a VBR priority queue. If SCH_MODE = CBR, this bit set to a logic high specifies that an unused CBR schedule slot should be used as a tunnel slot, with tunnel priorities specified in word 7 of the VCC table entry.
GFR_PRI	Specifies the UBR priority level for a GFR service connection. GFR_PRI must be < PRI.
SCH_MODE	Traffic class of VCC. (See Chapter 6 for details.) 000 UBR - Unspecified Bit Rate 001 CBR - Constant Bit Rate 010 Reserved 011 GFR - Guaranteed Frame Rate. 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 4.1)
PRI	Segmentation priority. The lowest priority is zero. The highest priority is seven. This field is not active when SCH_MODE is CBR. When SCH_MODE is GFR, this specifies the VBR priority.
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. The SAR writes this bit to zero 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.

Table 4-3. Segmentation VCC Table Entry — AAL5 and 0 Field Descriptions (2 of 2)

Field Name	Description
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. (The contents of this field are detailed in Chapter 6 - 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	Rsvd				Rsvd				LAST_PNTR																												
1	Reserved							Rsvd				BOM_PNTR																									
2	ATM_HEADER																																				
3	FIFO_PNTR																																				
4	CRC_REM																																				
5	STM_MODE	Rsvd				STAT	Rsvd	Rsvd	RUN=1	Rsvd	CURR_PNTR= 0x00000																										
6	Reserved								SCH_MODE (=001 - CBR)				Rsvd	PRI	SCH_OPT	Reserved																					
7-9/19	SCH_STATE																																				

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 SAR shared memory. The RS8235 retrieves host data buffers from any byte aligned PCI address using the “READ Multiple” PCI command. SAR shared data buffers must be aligned on word boundaries. Buffers contain any number of bytes of only user data, up to a maximum of approximately 64k.

### 4.3.3 Segmentation Buffer Descriptors

SBDs reside in SAR shared memory on word aligned addresses. 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 4-6 through Table 4-9 display the SEG buffer descriptor formats.

Table 4-6. Segmentation Buffer Descriptor 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 <sup>(1)</sup>

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			
<b>NOTE(S):</b>																
<sup>(1)</sup> This definition of bits 31:16, MISC_DATA field, applies when the HEADER_MOD bit is set and RPL_VCI = 0; used when generating OAM cells.																

Table 4-8. MISC\_DATA Field Bit Definitions with RPL\_VCI Bit Set <sup>(1)</sup>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Def.	NEW_VCI															
<b>NOTE(S):</b>																
<sup>(1)</sup> 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. Field Descriptions of the Segmentation Buffer Descriptor (1 of 3)

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 SDB)/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 can 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 SAR shared memory space. Host or SAR shared memory 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 SAR shared memory instead of host memory.
SET_CI	0 - The RS8235 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 one.
SET_CLP	0 - The RS8235 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 one. Also used to control VBR CLP Dual Leaky Bucket mode.
HEADER_MOD	0 - The RS8235 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 RS8235 generates the VCI field from the VCC Table Entry ATM_HEADER field. 1 - The RS8235 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 <a href="#">Chapter 7.0</a> for more details.
GEN_PDU	1 - Generate AAL5 trailer.
CRC10	Overwrite last 10 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. Field Descriptions of the Segmentation Buffer Descriptor (2 of 3)

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, and 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 buffer descriptor for any Tx Queue entry with LINK_HEAD set. To do otherwise can 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 - RS8235 reads the 48 octet payload of ATM cells from memory and generates the ATM header internally. 1 - The RS8235 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 <a href="#">Table 4-1.</a> )
EOM	1 - Buffer contains the end of a message. (See <a href="#">Table 4-1.</a> )
LENGTH	Number of bytes of data contained in the segmentation buffer. Local memory: non-EOM buffer length must be multiple of 4 bytes (mod 4).The RS8235 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 RS8235 generates the GFC field from the VCC Table Entry ATM_HEADER field. 1 - The RS8235 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 RS8235 generates the PTI field from the VCC Table Entry ATM_HEADER field. 1 - The RS8235 overwrites the ATM header PTI field for all cells in the buffer with PTI_DATA. The host can use this feature to generate F5 OAM cells. See Chapter 7. This bit is active only when HEADER_MOD is set. This bit disables BIP16 calculations.
WR_VCI	0 - The RS8235 generates the VCI field from the VCC Table Entry ATM_HEADER field. 1 - The RS8235 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 <a href="#">Chapter 7.0.</a> ) This bit is active only when HEADER_MOD is set. This bit disables 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.

Table 4-9. Field Descriptions of the Segmentation Buffer Descriptor (3 of 3)

Field Name	Description
NEW_VCI	Data for the RPL_VCI. The RS8235 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 RS8235 links this buffer descriptor to the identified VCC.

### 4.3.4 Transmit Queues

**4.3.4.1 Entry Format** The host submits chains of SBDs to the RS8235 by writing a single word transmit queue entry. The [Table 4-10](#) and [Table 4-11](#) 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. Field Descriptions of the Transmit Queue Entry

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 one 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 RS8235 links the new descriptor chain at the end of the existing chain on the VCC. 1 - The RS8235 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 can 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. NOTE: It is mandatory that the SINGLE option is set in the buffer descriptor for any Tx Queue entry with LINK_HEAD set. To do otherwise can result in corrupted seg data.
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. Bits 22:2 of the address are specified; the two least significant bits of the pointer are assumed to be zero (word-aligned).

#### 4.3.4.2 Transmit Queue Management

The transmit queues reside in a single continuous section of SAR 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 runtime 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. A set of other internal registers, the Transmit Queue Base Table entries, track the current state of the queues. [Table 4-12](#) and [Table 4-13](#) below describe the fields of these queues.

The byte address of any transmit queue entry is given by:

$$\begin{aligned} &(\text{SEG\_TXBASE}(\text{SEG\_TXB}) \times 128) \\ &+ \langle \text{transmit queue number} \rangle \times \langle \text{decoded TQ\_SIZE value} \rangle \\ &+ \langle \text{entry number} \rangle \times 4 \end{aligned}$$

The host manages each transmit queue as an independent write-only control queue. [Chapter 3.0](#) describes the runtime 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\_TXBASE 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 for details.)
LOCAL	0 - READ_UD located in PCI address space. 1 - READ_UD located in SAR shared memory. Note: For write-only PCI host interfaces, this bit should be set low.
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.

### 4.3.5 Segmentation Status Queues

**4.3.5.1 Entry Format** The RS8235 reports segmentation to the host on one of four status queues. Each entry on the queue is two 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 SAR 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 can process the entry. Written to one by the SAR. Written to zero 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 is 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.)
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.)
SEG_VCC_INDEX	Segmentation VCC index on which the SAR transmitted the buffer or PDU.

#### 4.3.5.2 Status Queue Management

At initialization, the host assigns the location and size of up to four queues by initializing internal registers, the Segmentation status queue base table entries. 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](#) and [Table 4-17](#) describe the status queue base table structure.

**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 zero (word aligned).
LOCAL	0 - Status queue located in PCI address space. 1 - Status queue located in SAR shared memory address space. This bit should be set to zero for a write-only PCI host architecture.
SIZE	Number of entries in this status queue: 00 = 64 01 = 256 10 = 1,024 11 = 4,096
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 RS8235 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 RS8235 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 0x1400–0x141F.

The internal transmit queue base table registers (SEG\_TQ\_QUn) are in the address range 0x1500–0x151F.

Other internal segmentation and scheduler registers are in the address range 0x1600–0x17FF.

**Table 4-18. Segmentation Internal SRAM Memory Map**

Address	Name	Description
<b>Segmentation Status Queue Base Table Registers (SEG_SQ_QUn):</b>		
0x1400–0x1407	SEG_SQ_QU0	Status Queue 0 Base Table
0x1408–0x140F	SEG_SQ_QU1	Status Queue 1 Base Table
0x1410–0x1417	SEG_SQ_QU2	Status Queue 2 Base Table
0x1418–0x141F	SEG_SQ_QU3	Status Queue 3 Base Table
0x1420–0x14FF	Reserved	Not Implemented
<b>Internal Transmit Queue Base Table Registers (SEG_TQ_QUn):</b>		
0x1500–0x1507	SEG_TQ_QU0	Transmit Queue 0 Base Table
0x1508–0x150F	SEG_TQ_QU1	Transmit Queue 1 Base Table
0x1510–0x1517	SEG_TQ_QU2	Transmit Queue 2 Base Table
0x1518–0x151F	SEG_TQ_QU3	Transmit Queue 3 Base Table
0x1520–0x15FF	Reserved	Not Implemented
<b>Further Internal Segmentation and Scheduler Registers:</b>		
0x1600–0x17FF	Reserved	Not Implemented

## 5.0 Reassembly Coprocessor

---

### 5.1 Overview

The reassembly (RSM) 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 RS8235 supports AAL5 and AAL0 reassembly, as well as cell mode (1-cell PDUs through a Virtual FIFO, for CBR voice traffic).

The RS8235 reassembles up to approximately 1 K 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 four independent buffer pools, from which the RSM coprocessor draws buffers as needed.

The RS8235 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 four independent status queues.

**NOTE:** This chapter provides information on the functions and data structures of the reassembly coprocessor.

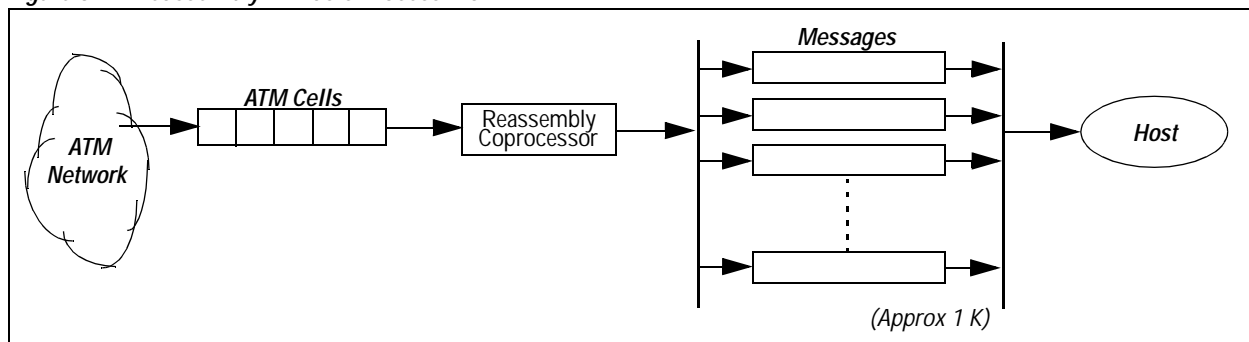
For detailed information on how the RS8235 deals with Operation, Administration and Maintenance (OAM) functions and interacts with the segmentation coprocessor in handling traffic management and scheduling, refer to [Chapter 6.0 Traffic Management](#) and [Chapter 7.0 OAM Functions](#).

## 5.2 Reassembly Functional Description

Each cell received from the ATM Physical Interface block belongs to any one of a possible 1 K virtual channels, or simultaneous messages. Due to the asynchronous nature of ATM, the cell contained in any incoming cell slot can belong to any VCC. Thus, the reassembly coprocessor must assign each arriving cell to the proper VCC, thereby demultiplexing the incoming messages.

Figure 5-1 illustrates the basic reassembly process flow.

Figure 5-1. Reassembly — Basic Process Flow



### 5.2.1 Reassembly VCCs

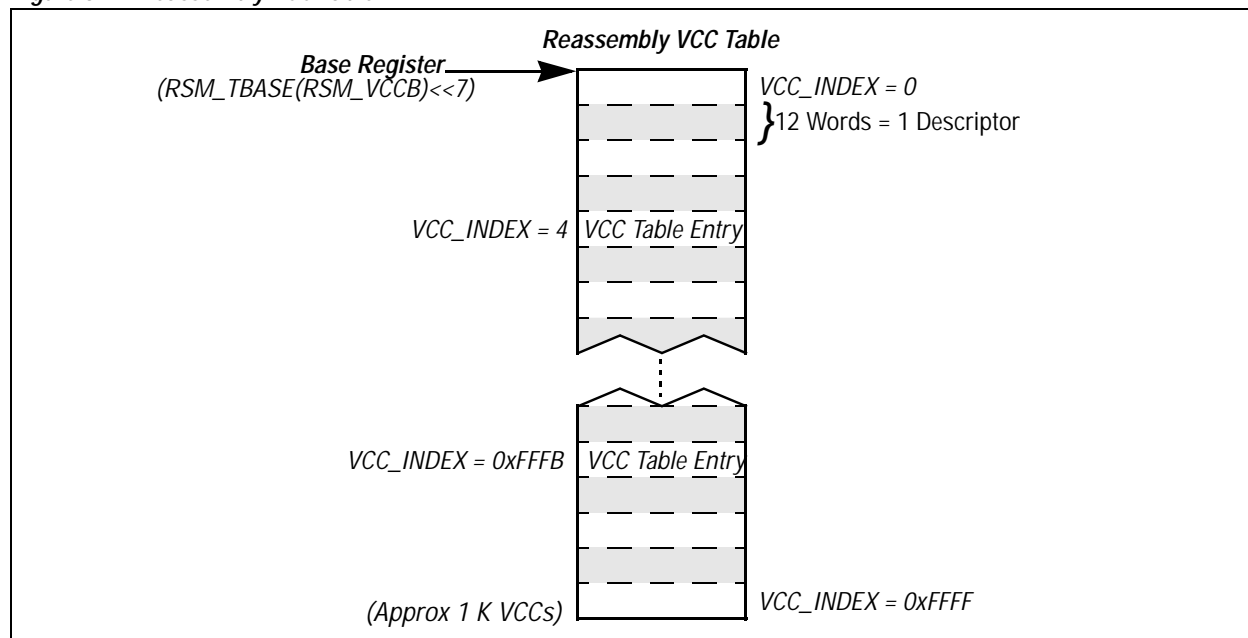
As with segmentation VCCs, the RS8235 supports up to 1 K reassembly VCCs, referenced by VCC\_INDEX, which identifies a location in the reassembly VCC Table.

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

While the RS8235 will accept any RSM VCC Index within the range of 64 K VCC Indexes, the actual number of reassembly VCCs allowed by the SAR is limited by the amount of local memory available in which to allocate and create RSM VCC Tables, free buffer queues, RSM status queues, etc.

Figure 5-2 displays how entries in the RSM VCC Table are indexed by VCC\_INDEX.

Figure 5-2. Reassembly VCC Table



#### 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 can 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 RS8235 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.

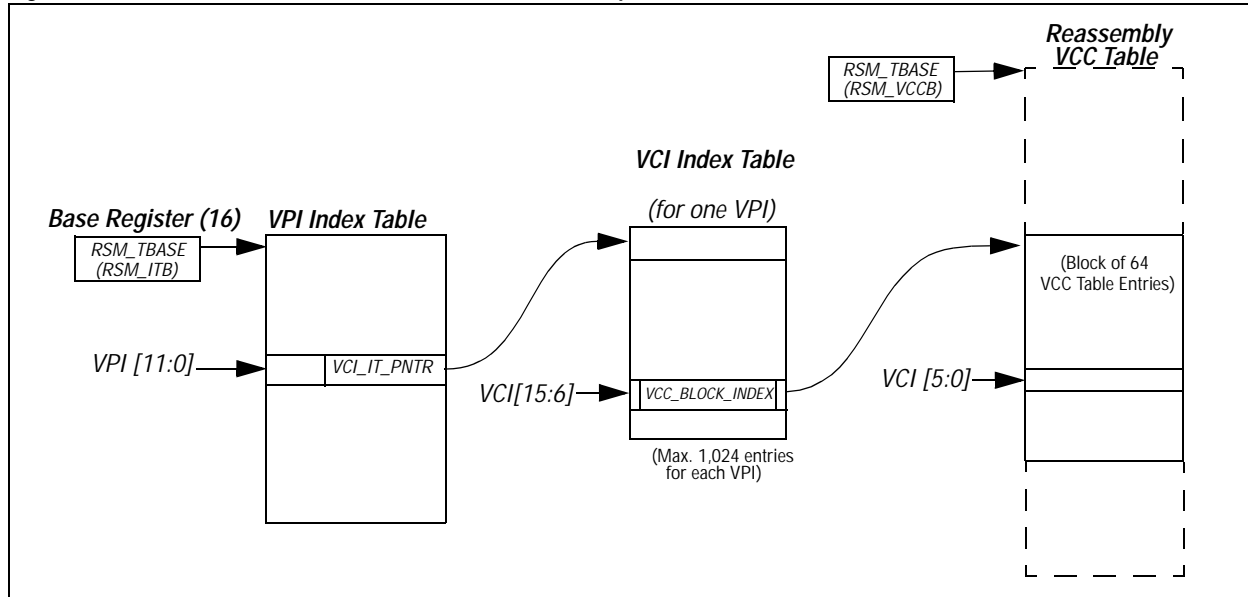
## 5.2.2 Channel Lookup

The RS8235's reassembly coprocessor implements a VPI/VCI Index Table mechanism utilizing direct index lookup 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 pre allocating large numbers of channels by requiring that only the VCI Index Table entries be pre allocated instead of the VCC Table entries.

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

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



#### 5.2.2.1 Programmable Block Size for VCC Table/VCI Index Table

Some users can have a requirement or desire to limit the amount of memory allocated to VPI/VCI channel lookup. To enable this, the RS8235 provides the user with the choice of enabling an alternative scheme for the memory allocation and table handling involved in the Direct Index Lookup mechanism. In this scheme, the user programs the size of the memory block of RSM VCC Table entries for all VCI Index table entries, to fit a range of 1 to 64 entries, instead of the default 64 entries.

Each RSM VCC entry requires 12 words of memory. By thus limiting the amount of memory space set aside for RSM VCCs, the total memory space required for VCC allocation can be substantially lessened.

To enable this scheme, set EN\_PROG\_BLK\_SZ in the RSM\_CTRL1 register to a logic high. The SAR will thus allocate block memory for VCC entries per VCI, based on the value entered in VCI\_IT\_BLK\_SZ (RSM\_CTRL1).

The data structures that facilitate this scheme are illustrated in section 5.7.1 on page 5-29. See Figure 5-4 for an illustration of the alternate lookup mechanism.

Figure 5-4. Programmable Block Size Alternate Direct Index Method

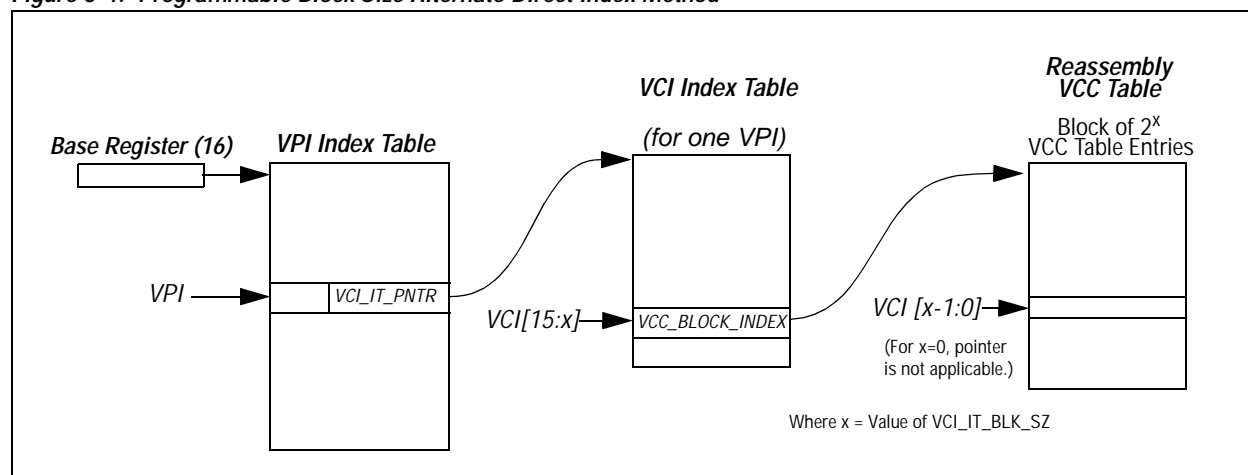


Table 5-1 shows the relationships of the values in VCI\_IT\_BLK\_SZ, and the values for the number of VCI Index Table and VCC table entries per VCI.

Table 5-1. Programmable Block Size Values for Direct Index Lookup

Value of VCI_IT_BLK_SZ	Value of x	VCI Index Table Portion of Cell's VCI	Number of possible VCI Index Table Entries per VPI	VCC Table Portion of Cell's VCI	Number of possible VCC Entries per VCI Index Table Entry
000	0	VCI[15:0]	65536	(No pointer)	1
001	1	VCI[15:1]	32768	VCI[0]	2
010	2	VCI[15:2]	16384	VCI[1:0]	4
011	3	VCI[15:3]	8192	VCI[2:0]	8
100	4	VCI[15:4]	4096	VCI[3:0]	16
101	5	VCI[15:5]	2048	VCI[4:0]	32
110	6	VCI[15:6]	1024	VCI[5:0]	64

**5.2.2.2 Setup** At system initialization, the user configures the RS8235 to comply with either the 8-bit UNI VPI field or the 12-bit NNI VPI field, by setting the RSM\_CTRL0 (VPI\_MASK) bit to a logic high for UNI operation, or a logic low for NNI operation. This configuration determines whether the RS8235 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.

The user can also enable the programmable block size for VCC Table entries as described in the section above, by setting EN\_PROG\_BLK\_SZ(RSM\_CTRL1) to a logic high.

At system initialization, the user can also limit the valid range of both VPI and VCI addresses to be processed, in order to reduce the memory size of the lookup structures being accessed. VPIs are limited by VP\_EN; and VCIs are limited by VCI\_RANGE in the VPI Index table entry, as well as BLK\_EN in the VCI Index table entry when EN\_PROG\_BLK\_SZ is enabled.

VPI/VCI address pairs can now be pre allocated 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.

### 5.2.2.3 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  $RSM\_TBASE(RSM\_ITB) \times 0x80$ . 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”. The RSM coprocessor uses the VPI value to read 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:x] values for that VPI, and thus sets the usable size of the VCI Index table for that VPI. If the value of VCI[15:x] 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 RS8235 then reads the appropriate entry in the VCI Index table. The address of the VCI Index Table entry is derived as follows:

$$(VCI\_IT\_PNTR \times 4) + (VCI[15:x] \times 4)$$

The VCC\_BLOCK\_INDEX in the VCI Index table entry selects a contiguous block of 64 reassembly VCC State Table entries (or from one to 64 VCC State Table entries if EN\_PROG\_BLK\_SZ is enabled), offset from the base address of the reassembly VCC Table. The VCC\_INDEX value is derived by concatenating the VCC\_BLOCK\_INDEX value with the VCI[x-1:0] bits from the received cell header. Thus, VCI[x-1:0] from the received header points to the reassembly VCC State Table entry for that VCC.

$$VCC\_INDEX = VCC\_BLOCK\_INDEX + VCI[x-1:0]$$

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 RS8235 increments the CELL\_RCVD\_CNT counter.

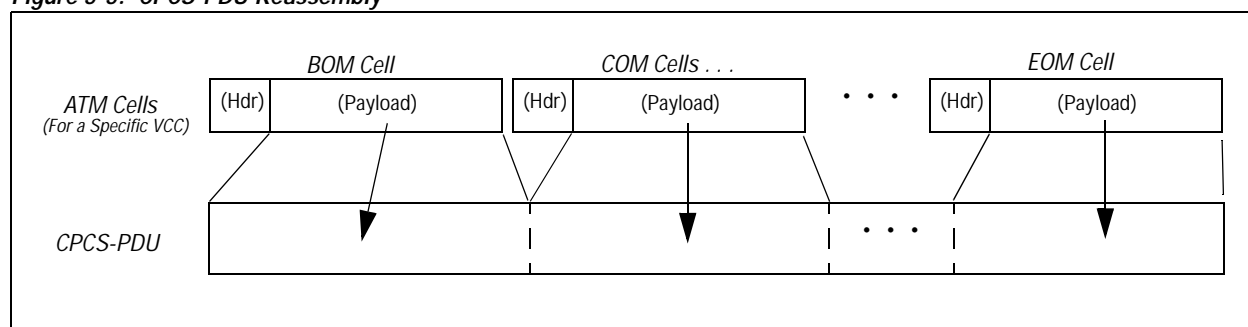
## 5.3 CPCS-PDU Processing

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 described in detail in the subsections below.

Figure 5-5 illustrates the basic process function.

Figure 5-5. CPCS-PDU Reassembly



### SETUP

Each active reassembly VCC must have a corresponding entry in the reassembly VCC table to describe its state. At channel setup time — either during system initialization for Provisioned Virtual Connections (PVCs) or dynamically, for Switched Virtual Connections (SVCs) — the host allocates a reassembly VCC table entry and configures the VCC according to its provisioned or negotiated characteristics. This includes the AAL in use, its assigned buffer pools and allocation priority, as well as the associated segmentation VCC Index (SEG\_VCC\_INDEX) for full duplex connections.

### 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 RS8235 to perform specific functions as described throughout this chapter.

### 5.3.1 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. The EOM cell contains both user data and CPCS-PDU overhead, and delineates the end of an AAL5 PDU.

#### 5.3.1.1 AAL5 COM Processing

During reassembly of the PDU, the reassembly coprocessor calculates a CRC-32 value on the received AAL5 PDU, as well as counting 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 entry for that channel.

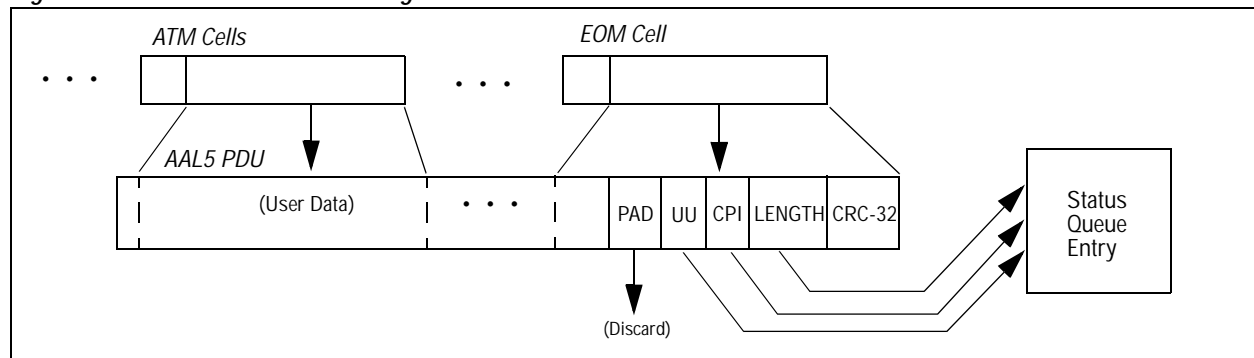
#### 5.3.1.2 AAL5 EOM Processing

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. The RSM coprocessor writes these specific fields to the status queue entry:

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

Figure 5-6 illustrates these process functions.

Figure 5-6. AAL5 EOM Cell Processing — Fields to Status Queue

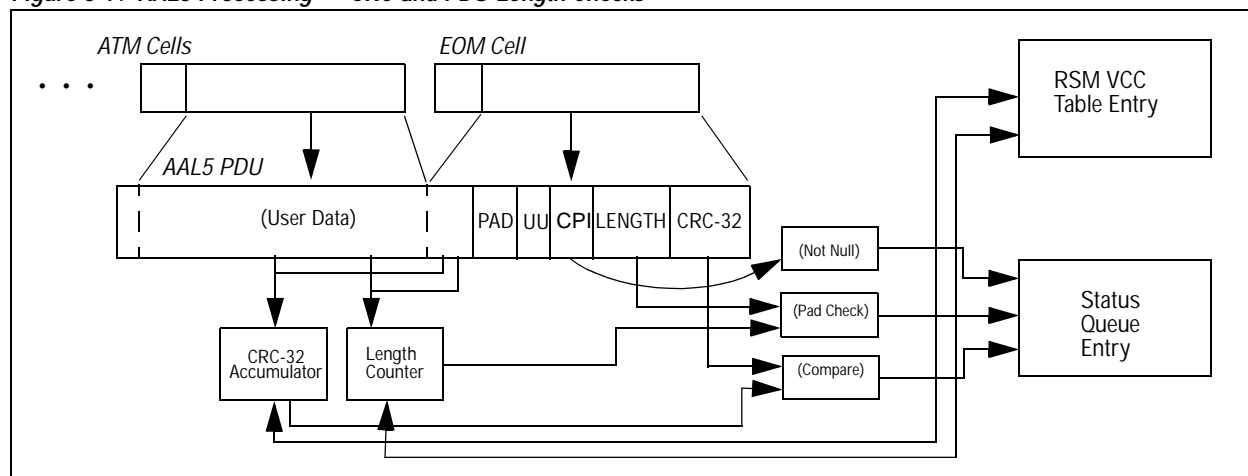


When the EOM cell is processed, the reassembly coprocessor performs the following checks:

- If the LENGTH field in the trailer of the AAL5 PDU is zero, the ABORT bit in the status queue entry is set to a logic high.
- 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.
- 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 zero 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 zero, the CPI\_ERROR bit in the status queue entry is set to a logic high.

Figure 5-7 illustrates these process functions.

Figure 5-7. AAL5 Processing — CRC and PDU Length Checks



The RS8235 reports all PDU termination events, with or without errors, in a status queue entry for that channel. See [Section 5.6](#), for full details.

### 5.3.1.3 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:

$$\min[\text{RSM\_CTRL0}(\text{MAX\_LEN}) \times 1024, 65568]$$

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 RS8235 receives a non-EOM cell, where `TOT_PDU_LEN + 48` is greater than `MAX_LEN x 1024` (or 65568), EPD is performed. The RS8235 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](#), for details on how this process is handled.

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

## 5.3.2 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.

### 5.3.2.1 Termination Methods

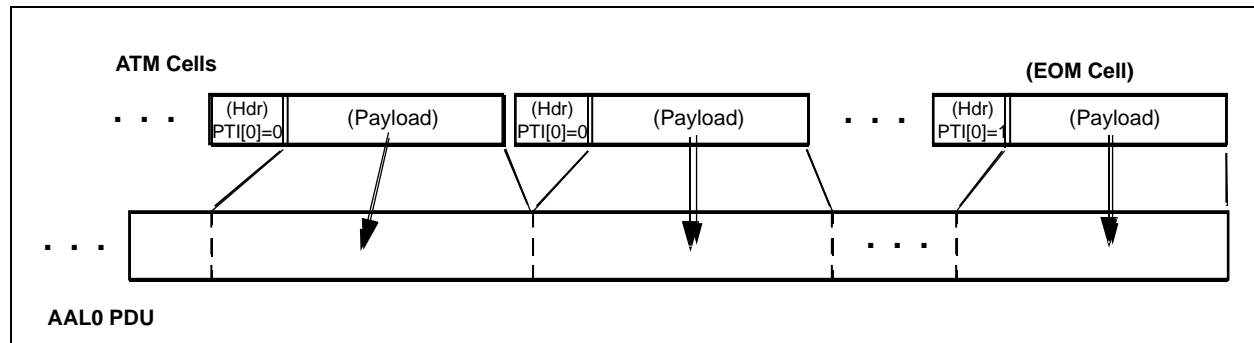
The RS8235 provides two methods of terminating an AAL0 PDU: Cell Count EOM and PTI termination.

The `TCOUNT` field in the RSM VCC state table entry determines the method for each VCC.

- If `TCOUNT = 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 one in this mode.
- If `TCOUNT = zero`, 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  $(\text{CCOUNT} \times 2)$  bytes.

See [Figure 5-8](#) for an illustration of this.

Figure 5-8. AAL0 PTI PDU Termination



The RS8235 reports all PDU termination events, with or without errors, in a status queue entry for that channel. See [Section 5.6](#).

#### 5.3.2.2 AAL0 Error Conditions

If the RS8235 receives a non-EOM cell in PTI termination mode, where  $TOT\_PDU\_LEN + 48$  is greater than  $CCOUNT \times 2$ , EPD is performed. The RS8235 reports this condition via a status queue entry, with the `LEN_ERROR` and `EPD` status bits set. Refer to [Section 5.4.8](#), for details on how this process is handled.

For each EOM cell where  $TOT\_PDU\_LEN + 48$  is greater than  $CCOUNT \times 2$ , the PDU is completed, with `BA_ERROR` status bit set.

The RS8235 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.

### 5.3.3 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.

### 5.3.4 BOM Synchronization Signal

The `STAT[1:0]` output pins can 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 given in [Table 5-2](#):

**Table 5-2. STAT Output Pin Values for BOM Synchronization**

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

The STAT output pins are valid during a SAR PCI master write address cycle. External circuitry would detect 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 four 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 RS8235 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

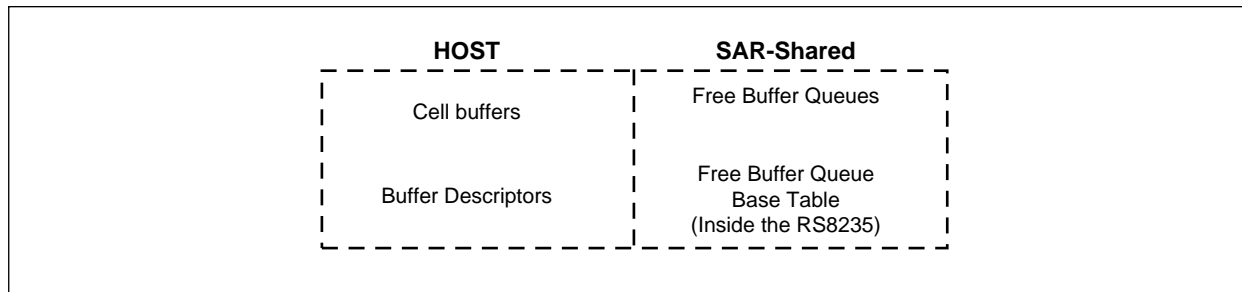
Data buffers can reside in both host and SAR shared memory. The majority of user data traffic should be reassembled in host memory. SAR shared memory reassembly is intended for low bandwidth management and control functions, such as OAM and Signaling. This allows an optional local processor to off-load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

### 5.4.2 Scatter Method

The RS8235 uses an intelligent scatter method to write cell payload data to host memory. 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.

Four data structures are maintained as illustrated in [Figure 5-9](#): two in the host memory, one in SAR 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 SAR shared memory. The free buffer queues also have an associated free buffer queue base table. This table is in internal memory. The RS8235 allows for up to four independent free buffer queues.

Figure 5-9. Host and SAR Shared Memory Data Structures for Scatter Method



8235\_004

### 5.4.3 Free Buffer Queues

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. The three most significant bits of the BFR0 and BFR1 fields must be set to zeros. BFR0 typically 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 0-1, and the BFR1 number is for free buffer queues 2-3. Thus, the free buffer queue is configured in two banks; Bank 0 containing free buffer queues 0-1, and Bank 1 containing free buffer queues 2-3.

The user can set BFR0 = BFR1, to disable this two-tier buffer structure.

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 four free buffer queues as follows:

$$\text{FBQx\_BASE} + ((\text{size of each free buffer queue}) \times \text{BFRx} \text{ MOD } 16)$$

Each of the four 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:

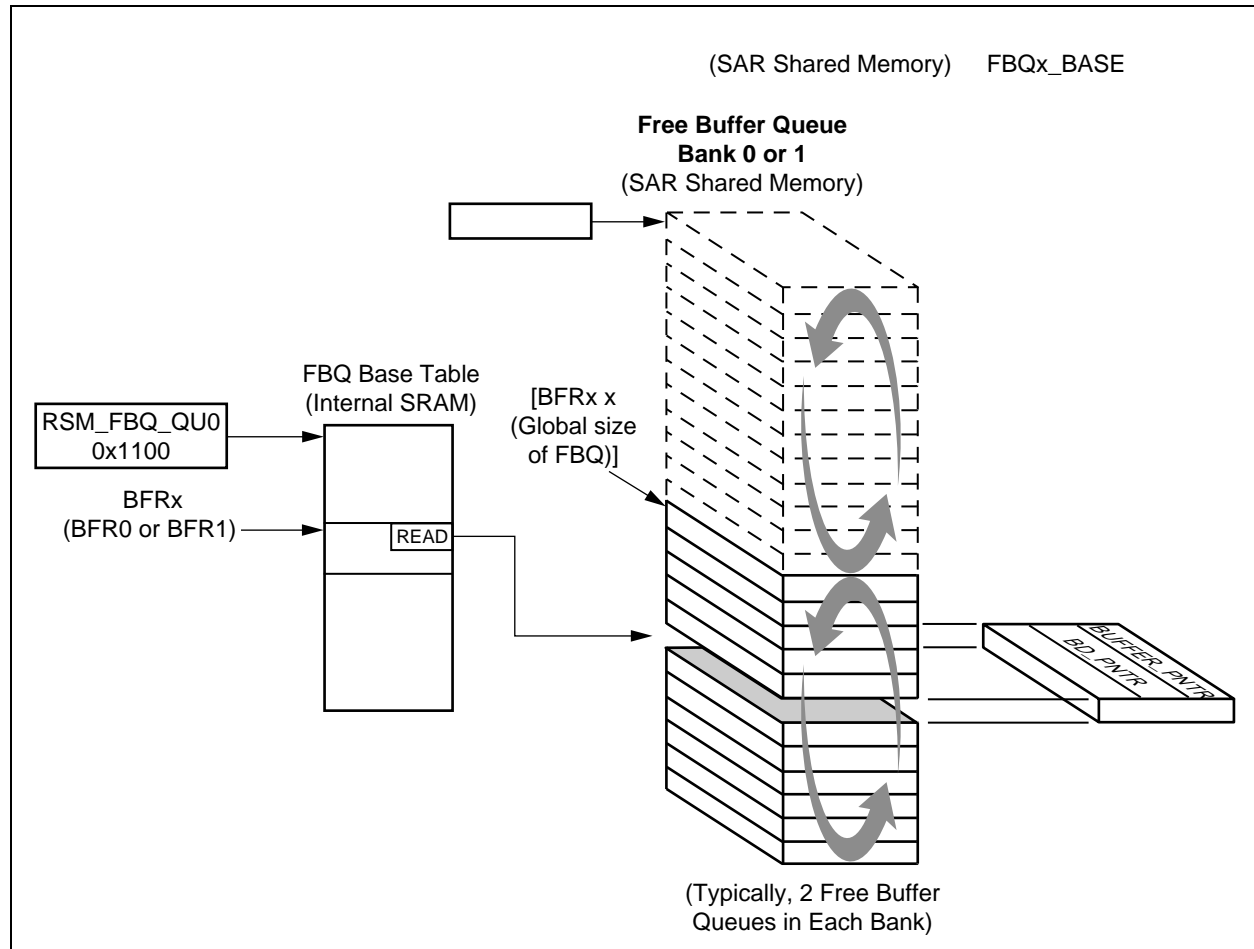
$$[\text{index of first entry for the queue}] + (\text{READ index pointer} \times \text{size of each free buffer queue entry})$$

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); and when the free buffer queue entry is read it returns those pointers.

Figure 5-10 illustrates this structure. Refer to Chapter 3.0 for more information on the operation of the free buffer queue.

Figure 5-10. Free Buffer Queue Structure



#### 5.4.4 Linked Data Buffers

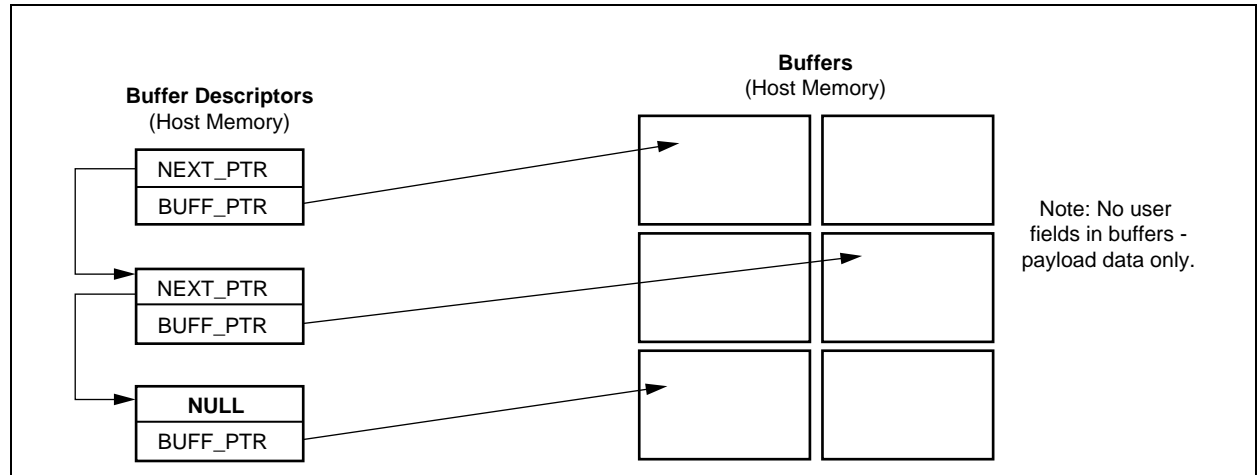
After the free buffer queue has returned pointers to a buffer descriptor and a cell buffer, the reassembly coprocessor writes payload data to the buffer.

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 5-11 illustrates this structure.

Figure 5-11. Buffer Structures



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 four free buffer queues in the `RSM_FQCTRL` register, (`FBQ_SIZE`) field, based on these values:

00 = 64  
 01 = 256  
 10 = 1,024  
 11 = 4,096

Initialize the free buffer queue update `INTERVAL` (i.e., how many buffers are taken off the free buffer queue before the RS8235 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 free buffer 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/SAR 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` fields 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 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. (The processing of this condition is described in [Section 5.4.7](#).) 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 RS8235 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, buffer descriptor pointer, and VLD bit set to a logic high. The host then increments its WRITE index pointer.

### 5.4.7 Error Conditions

A Free Buffer Queue Underflow or 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, then 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 early packet discard 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](#). 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.

**NOTE:** 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 Early Packet Discard

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 feature 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 Discard Eligibility (DE) field in the frame relay header. If this bit is a logic high, and the channel priority (the DPRI in the RSM 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.

#### 5.4.8.3 CLP Packet Discard

If the CLPD\_EN bit in the RSM VCC Table is a logic high, the Cell Loss Priority 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 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.

#### 5.4.8.4 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 LAN Emulation Clients (LECs) transmitting multicast frames (i.e., point-to-multipoint Emulated LAN 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. (The user might want to disable this function if the free buffers, buffer descriptors, and RSM status queues reside in SAR shared memory.)

Similarly, if RSM\_CTRL1(OAM\_QU\_EN) is a logic high, RSM\_CTRL1(OAM\_FF\_DSC) should be set to a logic high. (The user might want to disable this function if the global OAM buffers, buffer descriptors, and status queues reside in SAR shared memory.)

Early packet discard 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-PDUs 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 RS8235 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 RS8235 provides a status queue indication to the host. This indication allows the host to recover the buffers held by the partially completed PDU. The RS8235 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 RS8235 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 RS8235 increments CUR\_TOCNT in the RSM VCC Table entry. Then it compares CUR\_TOCNT to the time-out value selected, TERM\_TOCNT<sub>x</sub>, where  $x = TO\_INDEX$ . TERM\_TOCNT<sub>0</sub> through TERM\_TOCNT<sub>7</sub> are located at address 0x1340 through 0x134c in internal SRAM. They must be initialized to appropriate values during system initialization.

If  $CUR\_TOCNT = TERM\_TOCNT_x$ , then a time-out condition has occurred on the current VCC. The RS8235 follows the procedure described in [Section 5.4.9.4](#).

#### 5.4.9.2 Halting Time-out Processing

To halt time-out processing, the host "must" set the TO\_LAST bit to one, in the RSM VCC Table entry for the last VCC\_INDEX that the host wishes to have enabled for time-out processing. When the RS8235 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 RS8235 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 RS8235 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 will reset the counter.

#### 5.4.9.4 Reassembly Time-out Condition

The RS8235 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 RS8235 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.

$$\text{Period} = \text{SYSCLK period} \times \text{RSM\_TO\_PER} \times \text{RSM\_TO\_CNT} \times \text{TERM\_TOCNTx}$$

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 AAL0 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 one cell.

#### 5.4.10.2 Operation

During reassembly in this mode, whenever a buffer is required, 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.

*NOTE:* When firewalling is enabled in the RSM coprocessor and an FBQ empty (underflow) condition is encountered, the RX\_COUNTER field in the VCC table(s) still decrements each time the VCC receives a BOM cell. The RX\_COUNTER should not be decremented when the FBQ is empty. There is no workaround for this problem. The user “must” avoid FBQ empty conditions when firewalling is enabled.

#### 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:

- The FW\_EN bit set to a logic high enables 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 and 1, the RSM coprocessor decrements the RX\_COUNTER[15:0] in the RSM VCC Table entry for that channel. This allows COM buffers to be placed on queues 2 and 3 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 RS8235 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 PDU's 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 returned 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 RS8235 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 RS8235 “snoops” writes to free buffer queues that return firewall credits. When a write completes, the RS8235 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 RS8235 will continue to process credit return entries until FWD\_VLD is zero. Multiple free buffer queues can have credit return entries outstanding at one time. The RS8235 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 RS8235 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. They are:

- 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-PDU's 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.

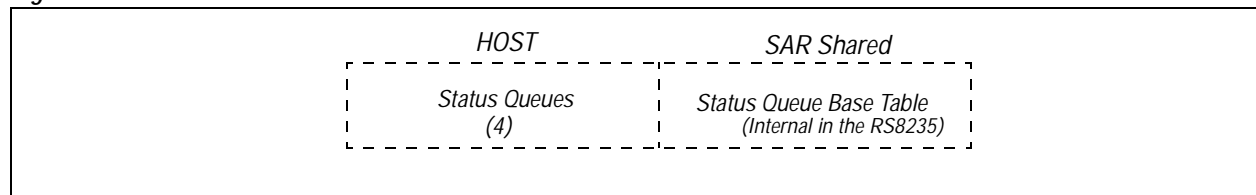
## 5.6 Status Queue Operation

The RS8235 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, and 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 as illustrated in [Figure 5-12](#): one in host memory and one in SAR shared memory. The status queues (HSTAT\_QU) reside in host memory, and the status queue base table resides in SAR shared memory, and allows for up to four independent circular status queues.

*Figure 5-12. Data Structure Locations for Status Queues*

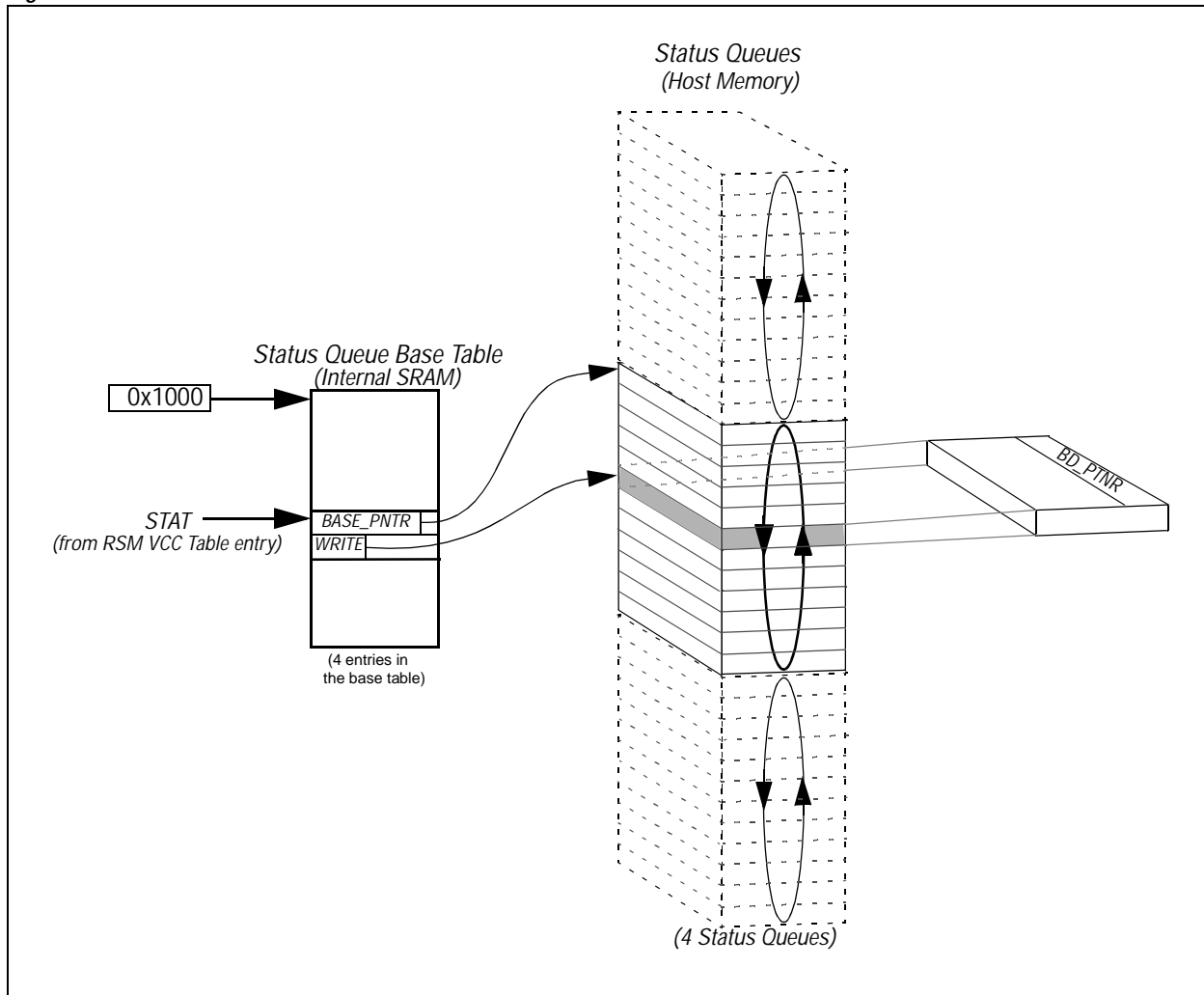


The status queue base table is located at 0x001000 in internal SRAM. The status queue base table contains status queue base information for up to four queues. The queues are accessed via a status pool number, the STAT field in the RSM VCC Table. This field is used as an index to the correct status queue 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 RS8235, incremented each time a status queue entry is written, to point to the next status queue entry for that status queue.

Figure 5-13 illustrates this structure.

Figure 5-13. Status Queue Structure Format



**5.6.1.1 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 local memory; otherwise set the bit to a logic low.

As well, initialize each status queue entry in all four status queues by setting the **VLD** bit to a logic low.

Initialize the **READ** pointers to zero for each status queue.

### 5.6.1.2 Operation

The reassembly coprocessor normally writes a status queue entry when a complete CPCS-PDU has been reassembled. It also writes a status queue entry for each received OAM cell.

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. It reads only the VLD bit at first before reading any other word, in order to maintain data coherency.

When the host finds the VLD bit set to a logic high, it processes the status queue entry, increments the host READ counter, and resets 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, to allow 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 Table entry 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, being set to a logic high, activates 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.

Refer to [Chapter 2.0](#) for more detailed information on the operation of status queues.

### 5.6.1.3 Errors

The RSM coprocessor also writes a status entry for several error conditions, including:

- 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 EPDs have occurred on channels mapped to the empty free buffer queue, then a status queue entry is written with the BOM and UNDF bits set to a logic high, and 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. It will be written a maximum of once per free buffer queue empty condition.

#### 5.6.1.4 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[3:0]) to determine which host memory status queue(s) caused the interrupt.

A typical operation for the interrupt manager would be to only read HOST\_ISTAT1 upon receiving an interrupt, and periodically read HOST\_ISTAT0 to insure 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.

### 5.6.2 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. 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 COM or EOM cell is received while the status queue is full, the channel is marked for status full packet discard. When an SSM, EOM, or OAM cell is received during a status full condition, the cell is discarded and the status queue checked. If there is now room in the status queue, then 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.

**NOTE:** Avoid having the Status Queue Overflow (or Full Condition) and DMA FIFO Buffer Full conditions at the same time.

## 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 order to assign each arriving cell to a virtual channel, based on its VPI/VCI value. Figure 5-14 illustrates this lookup mechanism.

Figure 5-14. VPI/VCI Channel Lookup Structure

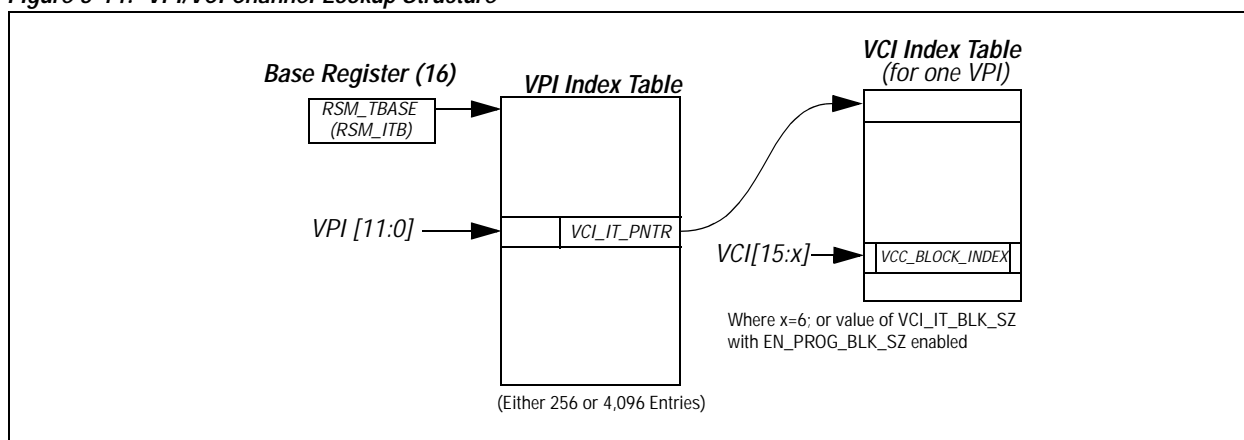


Table 5-3 describes the normal VPI Index Table format (one word per entry) without EN\_PROG\_BLK\_SZ (RSM\_CTRL1) enabled.

Table 5-4 describes the VPI Index Table format (two words per entry) with programmable VCC Table and VCI Index Table block size enabled.

Table 5-5 describes the field definitions for the VPI Index Table fields.

Table 5-3. Normal 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 5-4. VPI Index Table Entry Format with EN\_PROG\_BLK\_SX(RSM\_CTRL1) Enabled

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	Reserved										VCI_IT_PNTR																				
1	Reserved										VCI_RANGE																					

Table 5-5. 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 value allowed. If cell VCI exceeds maximum, cell is discarded, and counted as CELL_DSC_CNT. In normal operation, the 10 Most Significant Bits can be set by the user, with the 6 Least Significant Bits set at 1. When EN_PROG_BLK_SZ(RSM_CTRL1) is enabled, all 16 bits of the field can be set by the user.
VCI_IT_PNTR	VCI Index Table Base Pointer. Points to the base of the VCI Index Table for the VPI by appending two least significant zero bits to form a byte address.

Table 5-6 describes the VCI Index Table format without the programmable VCC Table and VCI Index Table block size enabled.

Table 5-7 describes the VCI Index Table format with EN\_PROG\_BLK\_SZ (RSM\_CTRL1) enabled.

Table 5-8 describes the VCI Index Table Descriptions.

Table 5-6. Normal 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 5-7. VCI Index Table Format with EN\_PROG\_BLK\_SZ(RSM\_CTRL1) Enabled

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	BLK_EN	Reserved															VCC_BLOCK_INDEX															

Table 5-8. VCI Index Table Descriptions

Field Name	Description/Function
BLK_EN	Block Enable. If logic high, enables entry. If a logic low, the cell is discarded.
VCC_BLOCK_INDEX	VCC block index. When EN_PROG_BLK_SZ is not enabled, this is the index to the block of 64 RSM VCC Table entries allocated to VCI[15:6]. In this case, VCC_BLOCK_INDEX is concatenated with the six least significant bits of the VCI to form the index into the RSM VCC Table to access the VCC Table entry for that channel. When EN_PROG_BLK_SZ is enabled, the (16 - (value of VCI_IT_BLK_SZ)) most significant bits of VCC_BLK_INDEX are concatenated with the ((value of VCI_IT_BLK_SZ) - 1) least significant bits of the VCI to form the index into the VCC Table to access the VCC Table entry. For example, if the value of VCI_IT_BLK_SZ = 4, then the resultant RSM VCC_INDEX pointer = {VCC_BLK_INDEX[15:4], CELL_VCI[3:0]}.

## 5.7.2 Reassembly VCC Table

Each reassembly VCC Table entry occupies one 12-word descriptor of the reassembly VCC Table.

There are two basic formats for the reassembly VCC Table entry—AAL5 and AAL0. Each completely describe the state of the reassembly process for individual VCCs.

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

Figure 5-15. Reassembly VCC Table Entry Lookup Mechanism

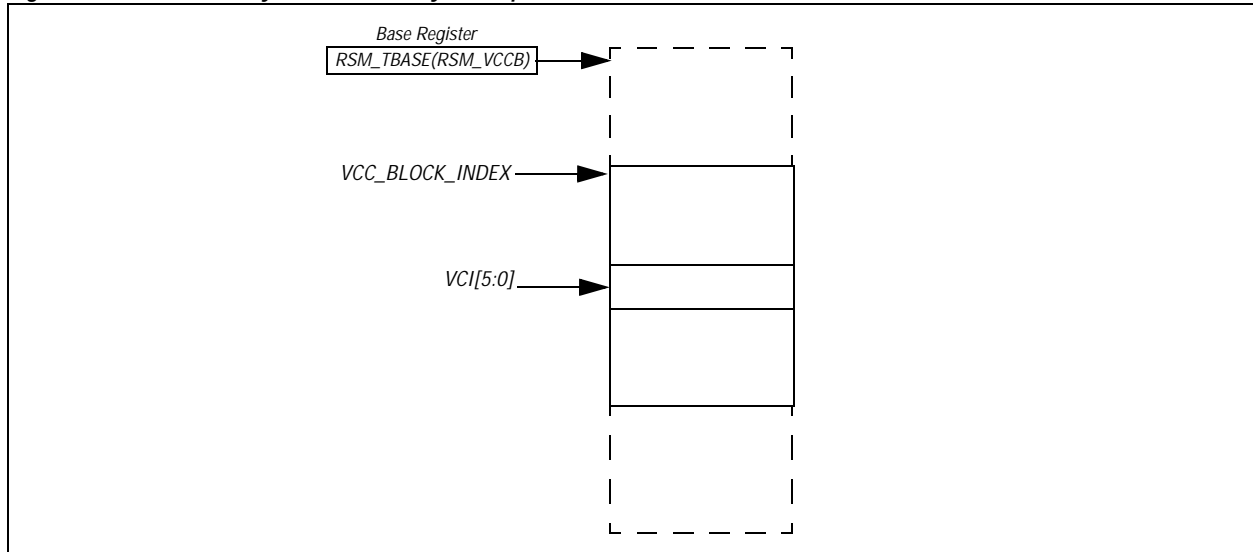


Table 5-9 describes the format of AAL5 reassembly VCC Table entries.

Table 5-10 describes the format of AAL0 RSM VCC Table entries.

Table 5-11 through Table 5-13 describe various bit definitions for certain fields in the VCC Table.

**KEY:**

	= Written by host at VCC setup.
	= Can be dynamically modified during active reassembly.

Table 5-9. 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	Reserved				AAL_EN															
1	TO_LAST	TO_EN	CUR_TOCNT										Rsvd	Reserved				ABR_CTRL														
2	PDU_FLAGS						Reserved				Reserved																					
3	CRCREM																															
4	Reserved												Rsvd	Rsvd	STAT	Rsvd	BFR1	Rsvd	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																															

Table 5-10. 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	Reserved				AAL_EN																
1	TO_LAST	TO_EN	CUR_TOCNT										Rsvd	Reserved				ABR_CTRL														
2	PDU_FLAGS						Reserved				TOT_PDU_LEN																					
3	CCOUNT										TCOUNT																					
4	Reserved										Rsvd	Rsvd	STAT	Rsvd	BFR1	Rsvd	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																															

Table 5-11. PDU\_FLAGS Field Bit Definitions

Bit	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 5-12. ABR\_CTRL Field Bit Definitions

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

Table 5-13. AAL\_EN Field Bit Definitions

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

Table 5-14 details the field descriptions for the reassembly VCC Table.

Table 5-14. 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; otherwise, 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 LECD table that holds channel LECD values.																				
FF_DSC	FIFO Full Discard. When a logic high, cells will be discarded when 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.																				
AAL_EN	<p>Enable various cell processes. The AAL_EN field contains the following control bits:</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>Rsvd</td><td>FIFO_EN</td><td>LNK_EN</td><td>FW_EN</td><td>M52_EN</td><td>BINTR</td><td>STM_MODE</td><td>LECID_EN</td><td>FRD_EN</td><td>CLPD_EN</td> </tr> </table> <p>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_CTRL0.  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 Discard Eligibility (DE) mode.  CLPD_EN = Enable CLP discard mode.</p>	9	8	7	6	5	4	3	2	1	0	Rsvd	FIFO_EN	LNK_EN	FW_EN	M52_EN	BINTR	STM_MODE	LECID_EN	FRD_EN	CLPD_EN
9	8	7	6	5	4	3	2	1	0												
Rsvd	FIFO_EN	LNK_EN	FW_EN	M52_EN	BINTR	STM_MODE	LECID_EN	FRD_EN	CLPD_EN												
TO_LAST	Indicates the last VCC table entry to process for time-out.																				
TO_EN	Enable time-out processing on the channel.																				
CUR_TOCNT	Current time-out counter for the channel.																				
ABR_CTRL	<p>Set various control bits related to QFC. The QFC_CTRL field contains the following control bits:</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>ER_EN</td><td>Reserved</td><td>ABR_VPC</td><td>Rsvd</td><td>Rsvd</td><td>Rsvd</td><td>Rsvd</td> </tr> </table> <p>ER_EN = Enable ER operation.  ABR_VPC = Indicates ER connection is VPC (zero 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.</p>	6	5	4	3	2	1	0	ER_EN	Reserved	ABR_VPC	Rsvd	Rsvd	Rsvd	Rsvd						
6	5	4	3	2	1	0															
ER_EN	Reserved	ABR_VPC	Rsvd	Rsvd	Rsvd	Rsvd															

Table 5-14. Reassembly VCC Table Descriptions (2 of 2)

Field Name	Description/Function										
PDU_FLAGS	<p>Set various flags related to PDUs. The PDU_FLAGS field contains the following control bits:</p> <p style="text-align: center;">9      8      7      6      5      4      3      2      1      0</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">CNT_ROVR</td> <td style="width: 10%;">SFPD_PND</td> <td style="width: 10%;">EPD</td> <td style="width: 10%;">CI</td> <td style="width: 10%;">CLP</td> <td style="width: 10%;">BFR_LOCAL</td> <td style="width: 10%;">BD_LOCAL</td> <td style="width: 10%;">ACT_PDU</td> <td style="width: 10%;">BOM_BUF</td> <td style="width: 10%;">FFPD_PND</td> </tr> </table> <p>CNT_ROVR = Indication that SERVICE_CNT counters have rolled over. The next status entry will indicate this condition.</p> <p>SFPD_PND = Status Full Packet Discard Pending. Set when status queue is full, and CPCS needs to be discarded when status entries available.</p> <p>EPD = EPD Flag. Set when early packet discard occurs on a channel. Cleared when new packet starts and error condition cleared.</p> <p>CI = Congestion Indication. PTI[1] header bit ORed across the CPCS-PDU.</p> <p>CLP = Cell Loss Priority. CLP header bit ORed across the CPCS-PDU.</p> <p>BFR_LOCAL = Buffer Local. If high, the current cell buffer is located in local memory; otherwise, host memory. SAR maintains this bit.</p> <p>BD_LOCAL = Buffer Descriptor Local. If high, the buffer descriptors reside in local memory; otherwise, host memory. SAR maintains this bit.</p> <p>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.</p> <p>BOM_BUF = BOM buffer flag. Set high when filling the first buffer of a PDU.</p> <p>FFPD_PND = DMA FIFO Full Packet Discard Pending.</p>	CNT_ROVR	SFPD_PND	EPD	CI	CLP	BFR_LOCAL	BD_LOCAL	ACT_PDU	BOM_BUF	FFPD_PND
CNT_ROVR	SFPD_PND	EPD	CI	CLP	BFR_LOCAL	BD_LOCAL	ACT_PDU	BOM_BUF	FFPD_PND		
TOT_PDU_LEN	Total PDU length in bytes.										
CRCREM	Cycle Redundancy Check 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 one.										
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.										
SEG_VCC_INDEX	Channel index of corresponding segmentation channel. Used by ABR and 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 reside on word-aligned addresses in host memory. The host controls the allocation and management of reassembly buffer descriptors.

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

[Table 5-15](#) and [Table 5-16](#) describe the format of the reassembly buffer descriptors.

**Table 5-15. 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 5-16. 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. <i>NOTE:</i> The SAR does not access this word, so the user can place it anywhere in the buffer descriptor.

### 5.7.4 Free Buffer Queues

The host initializes the free buffer queues in SAR 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 RS8235 internal memory. [Table 5-17](#) and [Table 5-18](#) describe the format of the free buffer queue base table entries.

[Table 5-19](#) and [Table 5-20](#) describe the format of the free buffer queue entries.

Table 5-17. 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 5-18. 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 SAR shared memory, otherwise in host memory.
BD_LOCAL	If logic high, buffer descriptor and READ_UD word are located in SAR shared memory, otherwise 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 (Firewalling).
READ_UD_PNTR	Word address in user memory to write READ pointer every UPDATE interval.

Table 5-19. 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 <sup>(1)</sup>	Reserved																Rsvd	FWD_VLD	VCC_INDEX													
3 <sup>(1)</sup>	(not used)																															

**NOTE(S):**

(1) These words are used only in Bank 0 if RSM\_FBQCTL(FBQ0\_RTN) = 1.

Table 5-20. 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 zero.
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 RS8235 reports reassembly status to the host on any one of four reassembly status queues.

At initialization the host assigns the location and size of up to four RSM status queues by initializing the reassembly status queue base table entries in RS8235 internal memory. The location and size of each RSM status queue is independently programmable via these base table entries.

[Table 5-21](#) and [Table 5-22](#) describe the format of the reassembly status queue base table entries.

[Table 5-23](#) through [Table 5-27](#) describe the formats of the reassembly status queue entries.

Table 5-21. 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 5-22. 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 local memory, otherwise it is located in host memory.
SIZE	Size of the status queue. 00 = 64 01 = 256 10 = 1,024 11 = 4,096
WRITE	Current Write index pointer maintained by the SAR.
READ_UD	Periodic Read update index pointer maintained by the user.

Table 5-23. Reassembly Status 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	BD_PNTR																														"00"	
1	UU						CPI						CPCS_LENGTH																			
2	Rsvd	PDU_CHECKS										VCC_INDEX																				
3	VLD	Reserved						Reserved						STATUS						Rsvd	OAM			STM								

Table 5-24. PDU\_CHECKS Field Bits

Bit	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved	BA_ERROR	Reserved	LEN_ERROR	PAD_ERROR	CPI_ERROR	Reserved	CRC_ERROR	Reserved	Reserved	Reserved	LP	CI_LAST	CI

Table 5-25. STATUS Field Bits

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

Table 5-26. STM Field Bits

Bit	3	2	1	0
	EOM	BOM	STM_MODE	BFR1

Table 5-27. 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.																		
PDU_CHECKS	Set various error flags related to PDUs.  BA_ERROR = AAL5: Indicates that the total PDU length exceeds MAX_LENGTH 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 zero. 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: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">8</td> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">FFPD</td> <td style="text-align: center;">EPD</td> <td style="text-align: center;">FW</td> <td style="text-align: center;">UNDF</td> <td style="text-align: center;">OVFL</td> <td style="text-align: center;">SFPD</td> <td style="text-align: center;">TO</td> <td style="text-align: center;">ABORT</td> <td style="text-align: center;">CNT_ROVR</td> </tr> </table> FFPD = DMA FIFO Full Packet Discard. EPD = Early Packet Discard occurred. A partially reassembled CPCS-PDU has been discarded due to firewall, buffer underflow, LI_EPD, SN_EPD, ST_EPD, CLP discard or Max PDU length exceeded. FW = Firewall error condition occurred. If early packet discard occurs, EPD will be set. UNDF = Free Buffer Queue Underflow occurred. If early packet discard occurs, EPD will be set. OVFL = Last available Status Queue entry. SFPD = Status Full Packet Discard occurred. EPD will not be set. 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.	8	7	6	5	4	3	2	1	0	FFPD	EPD	FW	UNDF	OVFL	SFPD	TO	ABORT	CNT_ROVR
8	7	6	5	4	3	2	1	0											
FFPD	EPD	FW	UNDF	OVFL	SFPD	TO	ABORT	CNT_ROVR											
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																		

Table 5-27. Reassembly Status Queue Entry Descriptions (2 of 2)

Field Name	Description/Function								
STM	<p>Sets various bits related to Streaming Mode. The STM field contains the following control bits:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">EOM</td> <td style="text-align: center;">BOM</td> <td style="text-align: center;">STM_MODE</td> <td style="text-align: center;">BFR1</td> </tr> </table> <p>EOM = In Streaming Mode or BOM interrupt mode, this bit identifies that the buffer contains an EOM.</p> <p>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.</p> <p>STM_MODE = Indication that Streaming Mode is enabled on the channel.</p> <p>BFR1 = In Streaming Mode, indicates which free buffer queue the buffer came from. Logic high indicates COM(BFR1), logic low indicates BOM(BFR0).</p>	3	2	1	0	EOM	BOM	STM_MODE	BFR1
3	2	1	0						
EOM	BOM	STM_MODE	BFR1						

### 5.7.6 LECID Table

The LECID Table illustrated in Figure 5-16, includes up to 32 unique identifiers for LAN Emulation Clients (LECIDs). The DPRI field in the reassembly VCC Table entry is used as the index into this table.

Table 5-28 and Table 5-29 describe the LECID Table entries and field definitions.

Figure 5-16. LECID Table

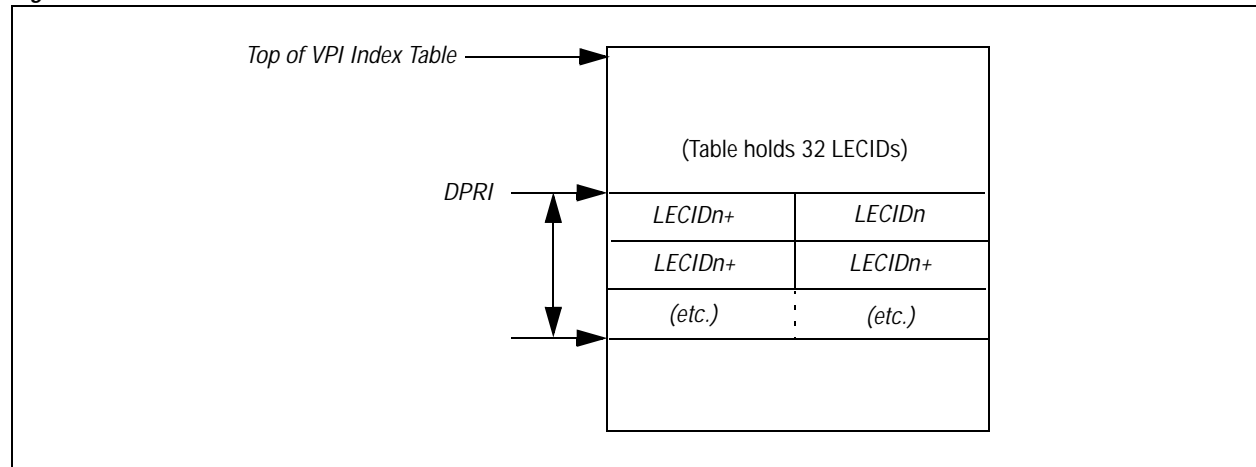


Table 5-28. LECID Table Entries

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 5-29. LECID Table Field Definition**

Field Name	Function/Description
LECIDn	LAN Emulation Client Identifier: a unique identifier for a LAN Emulation Client (LEC). 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 timeout values as described in [Section 5.4.9](#). [Table 5-30](#) and [Table 5-31](#) display the Global Time-out Table entry format and field descriptions.

**Table 5-30. 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 5-31. 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.



# 6.0 Traffic Management

## 6.1 Overview

The traffic management capabilities of ATM differentiate it from other communication technologies. The RS8235 xBR Traffic Manager implements the complete set of ATM Service Categories as defined in the ATM Forum's Traffic Management TM 4.1 Specification. These categories include Constant Bit Rate (CBR), Real-Time and Non-Real-Time Variable Bit Rate (rt-VBR and nrt-VBR), Unspecified Bit Rate (UBR), GFR, and Available Bit Rate (ABR).

Table 6-1 provides a list of ATM attributes detailed in the TM 4.1 Specification (i.e., traffic parameters, QoS parameters, and feedback characteristics), and identifies whether the RS8235 supports these for each service category. The shaded areas are not indicative that the service category and attribute are undefined for the SAR—they simply indicate that the TM 4.1 Specification does not detail them.

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

Attribute	ATM Layer Service Category					
	CBR	rt-VBR <sup>(1)</sup>	nrt-VBR <sup>(2)</sup>	UBR <sup>(3)</sup>	ABR	GFR
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:						

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

Attribute	ATM Layer Service Category					
	CBR	rt-VBR <sup>(1)</sup>	nrt-VBR <sup>(2)</sup>	UBR <sup>(3)</sup>	ABR	GFR
peak-to-peak Cell Delay Variation (CDV)	Supported					
max Cell Transfer Delay (CTD)	Supported <sup>(4)</sup>		Not Supported <sup>(4)</sup>			
Cell Loss Ratio (CLR)	Supported <sup>(4)</sup>			Not Supported <sup>(4)</sup>	Supported <sup>(4)</sup>	Not Supported <sup>(4)</sup>
<b>OTHER ATTRIBUTES:</b>						
Feedback <sup>(5)</sup>					Supported	
<b>NOTE(S):</b> (1) The rt-VBR service category is intended for real-time applications; i.e., those requiring tightly constrained delay and delay variation, as would be appropriate for voice and video applications. (2) The nrt-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 delay variation. Examples of such applications are traditional computer communications applications, such as file transfer and e-mail. (4) This is a network parameter. The RS8235 provides counters which monitor this parameter. (5) Feedback refers to the several types of control cells called Resource Management Cells (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 RS8235 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, UBR, and GFR traffic. Second, for ABR service classes, is the ABR Flow Control Manager, an additional state machine 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 TM 4.1 ABR specification. The RS8235 supports the rate based flow control and service models specified for ABR in TM 4.1.

The RS8235 also provides Generic Flow Control. This XON/XOFF protocol complements the GFC algorithm by allowing switches to significantly overallocate port bandwidth.

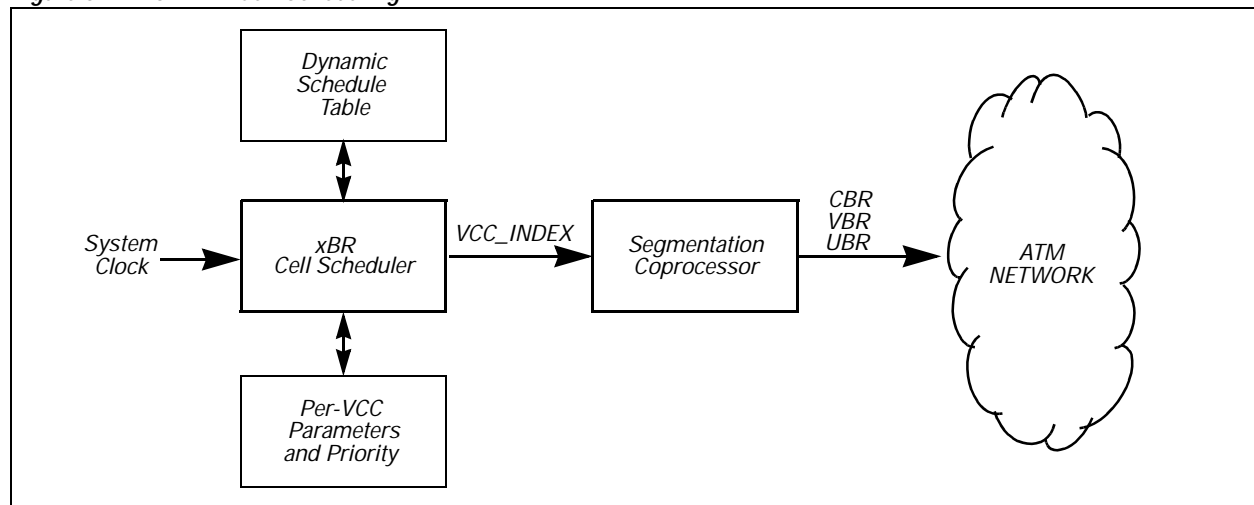
### 6.1.1 xBR Cell Scheduler

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. Schedule Slots in this table can be pre-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 RS8235 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 RS8235 can manage VCC traffic on either a VC or a VP level. As well, 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 based on their traffic parameters). Traffic into this CBR tunnel can be of types UBR, VBR and ABR. To enhance flexibility, the RS8235 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



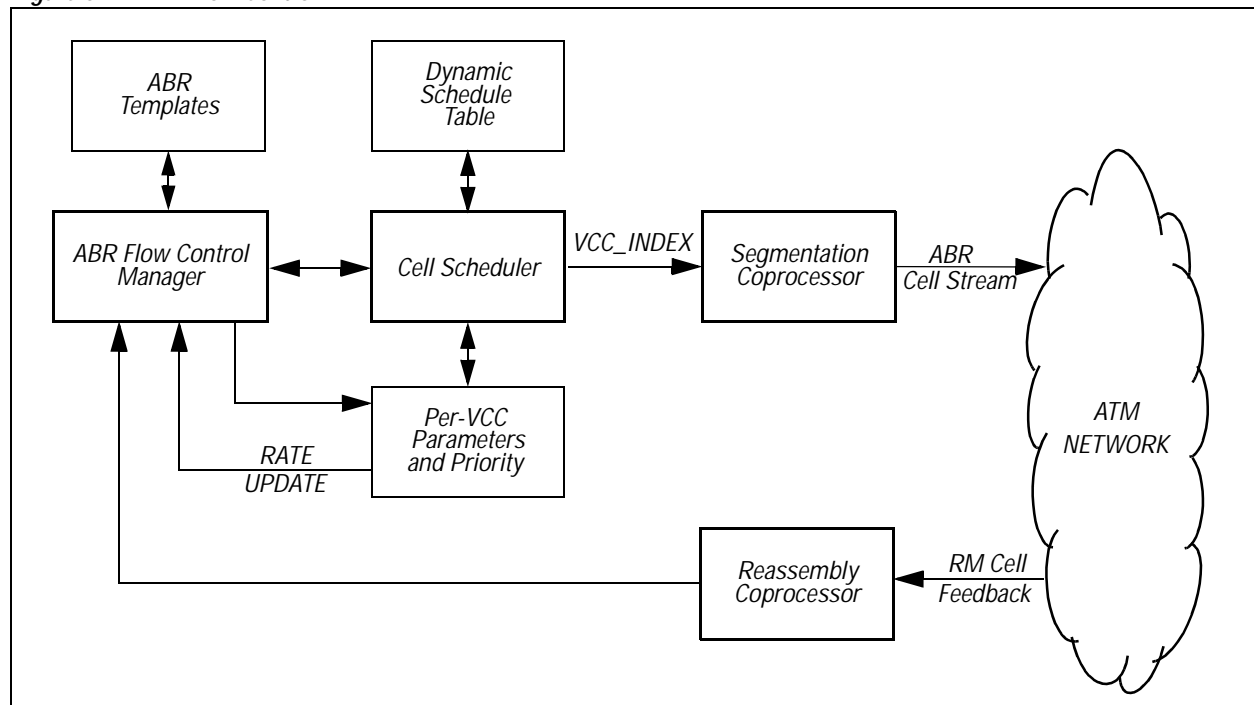
### 6.1.2 ABR Flow Control Manager

The RS8235 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 RS8235'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 SAR 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 TM 4.1 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 TM 4.1 specification, the state machine is programmable through Mindspeed-supplied ABR templates. These templates, resident in SAR 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.

**APPLICATION EXAMPLE: Application-Specific Templates**

Each template can have different flow control parameters. For instance, a system designer might wish to configure a VCC traversing a network with all Explicit Rate (ER) switches with a Rate Increase Factor (RIF) and Rate Decrease Factor (RDF) = 1. However, a VCC traversing a network with switches doing CI/NI (binary) marking will desire an RIF and RDF  $\neq$  1. Thus, separate templates would be desired for these VCCs.

## 6.2 xBR Cell Scheduler Functional Description

### 6.2.1 Scheduling Priority

**6.2.1.1 8 Priority Levels + CBR** The RS8235 supports eight scheduling priorities (the “PRI” field in the SEG VCC Table entry), in addition to the optional CBR service category, with “7” being the highest priority down to “0” being the lowest priority. CBR channels are given pre-assigned schedule slots, which is in effect a priority higher than the eight scheduling priorities discussed here. From one to four of these scheduling 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/ABR priorities in the VBR\_OFFSET field in SEG\_CTRL or SCH\_CTRL. How the value of VBR\_OFFSET is determined is illustrated in [Figure 6-4](#) and [Figure .](#)

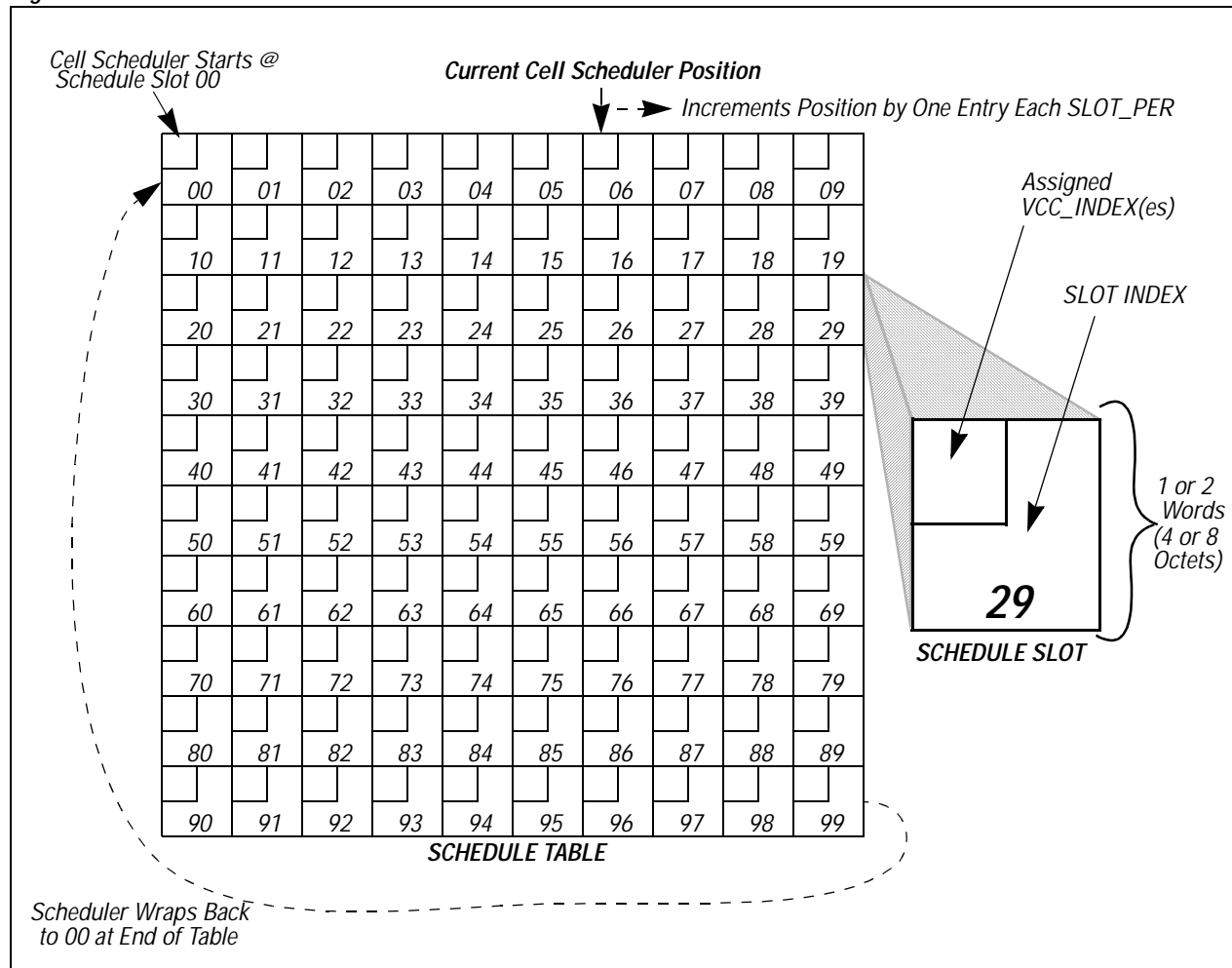
**6.2.1.2 VCC Priority Assignment** The host assigns the priority of all VCCs, except CBR VCCs, by setting the PRI field in the Segmentation VCC Table Entry. This priority should be set at connection setup and should not be changed dynamically.

### 6.2.2 Dynamic Schedule Table

**6.2.2.1 Overview** 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.

Figure 6-3 represents an example 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 can choose to schedule cells close to the full payload data rate of STS-3C at 353.2 K cells/second. 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 9,500 clocks, the scheduler position returns to 00.

Figure 6-3. Schedule Table with Size = 100



#### 6.2.2.2 Schedule Table Slots

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	Number of 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 thru 3
Yes	3	2 Words (64 bits)	1	1	VBR_OFFSET + 1 thru 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

### 6.2.2.3 Schedule Slot Formats

The VCC Index contents of each Schedule Table slot, based on the settings in the last four rows of the table above, are illustrated in [Figure 6-4](#) below. The SAR can 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, any of these VBR VCC\_Indexes assigned by the SAR can be the first of a linked list of VBR VCCs.

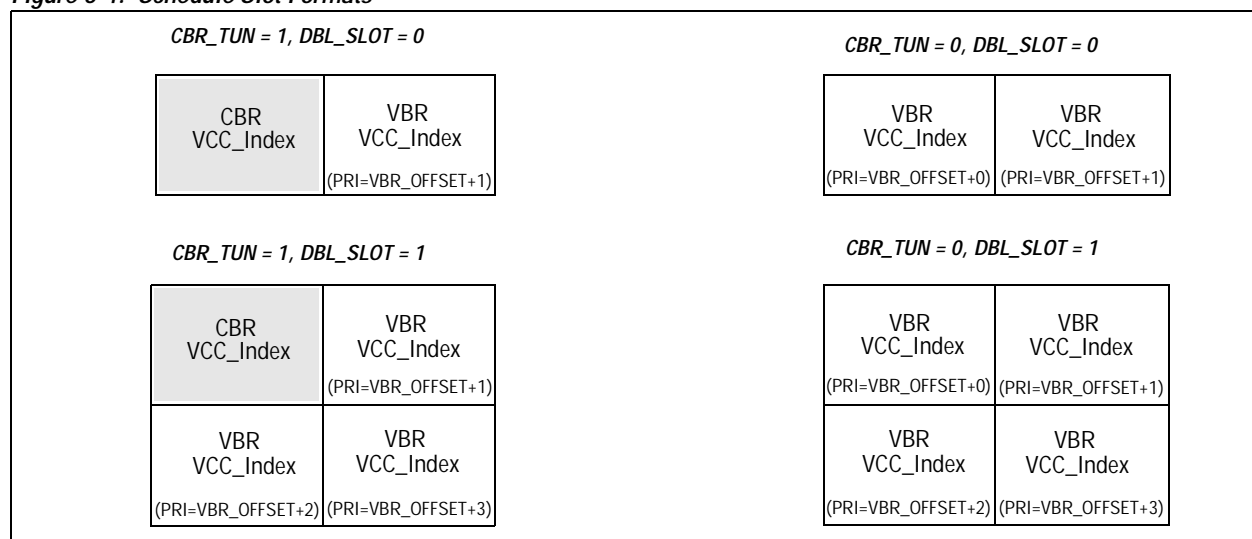
#### VBR\_OFFSET Described

VBR\_OFFSET gives the offset difference in priority level between what the user or system designer wishes to assign VBR channels and what the SAR assigns via the VBR Schedule Slot format. Thus, the value of VBR\_OFFSET is the difference between the highest VBR/ABR priority being actively scheduled and the largest priority of the VBR VCC\_Index field in the VBR scheduling slot (either one or three). 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:

$$\text{VBR\_OFFSET} = 5 - 3 = 2.$$

[Figure 6-4](#) illustrates how these priorities are assigned to the VBR fields.

Figure 6-4. Schedule Slot Formats

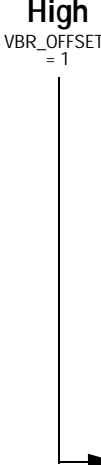


### 6.2.2.4 Some Scheduling Scenarios

Described and illustrated here are just a couple of examples of scheduling priority schemes that system designers can decide to implement, which demonstrate the range of flexibility that the RS8235 affords system designers and network administrators.

Figure 6-5 illustrates how scheduling priorities are assigned in scheduling slots. In this illustration, the number of VBR/ABR priorities is 3. In addition, the tunnel at PRI=5 has shared VBR traffic. CBR\_TUN = 1, DBL\_SLOT = 1. The highest VBR/ABR scheduling priority (i.e., PRI) is 5. Thus, VBR\_OFFSET = 1.

Figure 6-5. One Possible Scheduling Priority Scheme with the RS8235

<u>Scheduling Priority</u>		<u>Service/Application</u>	
<b>High</b> VBR_OFFSET = 1 		CBR	
	7	UBR1	Voice - AAL1 on AAL0 VCCs
	6	Reserved for Tunnel 1 (UBR only)	Signalling, ILMI, PNNI Traffic
	5	Reserved for Tunnel 2 (UBR & VBR shared)	Tunnel through public ATM Network
	4	VBR1 (VBR/ABR Priority 3)	Tunnel through private ATM Network
	3	VBR2 (VBR/ABR Priority 2)	rt-VBR — Video
	2	ABR (VBR/ABR Priority 1)	nrt-VBR — Frame Relay/ABR ≤ MCR
	1	UBR2	ABR > MCR — LAN Data Traffic
	0	UBR3	High Priority UBR User Traffic
<b>Low</b>		Low Priority UBR User Traffic	

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 Cell Delay Variation (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 RS8235 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 systems 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 RS8235 generates one cell on the specified VCC. The maximum or base rate of CBR channels is determined by the duration of a cell slot according to the equation below, where  $R_{max}$  is the maximum rate in cells per second.

$$SLOT\_PER = \frac{\text{frequency (SYSCLK)}}{R_{max}}$$

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.

#### **APPLICATION EXAMPLE: Determining Maximum Rate**

frequency(SYSCLK) = 33 MHz  
SLOT\_PER = 93 clocks/slot

$$R_{max} = 354.8 \text{ K cells/second}$$

To achieve the maximum rate, the user would assign one VCC to every cell slot in the Schedule Table. This would prevent any other VCC from being scheduled since this channel uses all of the available slots.

**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:

$$R, R \times (TBL\_SIZE - 1)/TBL\_SIZE, R \times (TBL\_SIZE - 2)/TBL\_SIZE, \dots, R/TBL\_SIZE$$

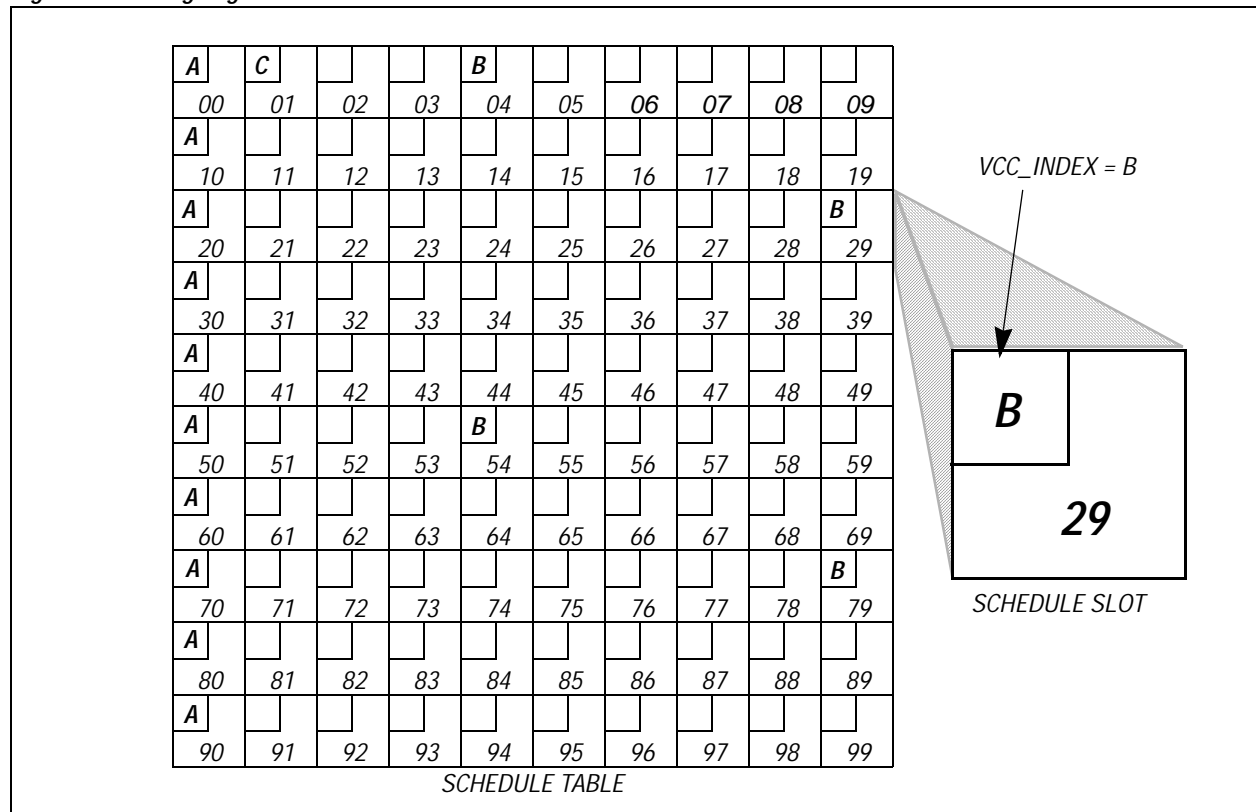
#### **Minimum Rate**

The minimum rate available is therefore  $R/TBL\_SIZE$ .

**Assigning CBR Cell Slots**

Figure 6-6 displays 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 and therefore it will transmit at  $R/10$ , or 35.4 K cells/second. VCC\_INDEX B occupies a slot every 25 cell slots and will transmit at a rate of  $R/25$ , or 14.2 K cells/second. VCC\_INDEX C occupies only one schedule slot and therefore will transmit at the minimum rate,  $R/TBL\_SIZE$ , or 3.54 K cells/second. Note that not all cell slots have been assigned to CBR channels. During these slots, the RS8235 will dynamically schedule traffic from the other service classes. However, the total bandwidth of channels A, B, and C will be reserved.

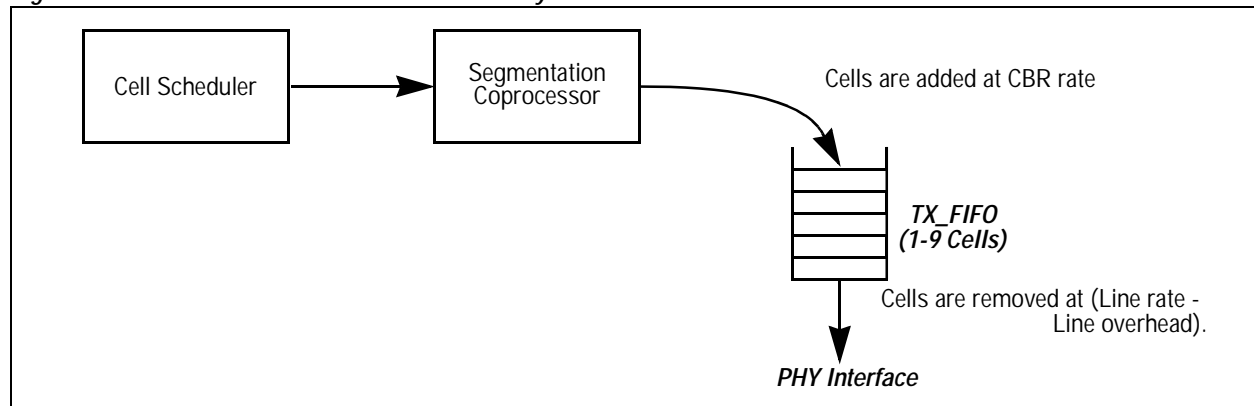
Figure 6-6. Assigning CBR Cell Slots



### 6.2.3.3 CBR Cell Delay Variation (CDV)

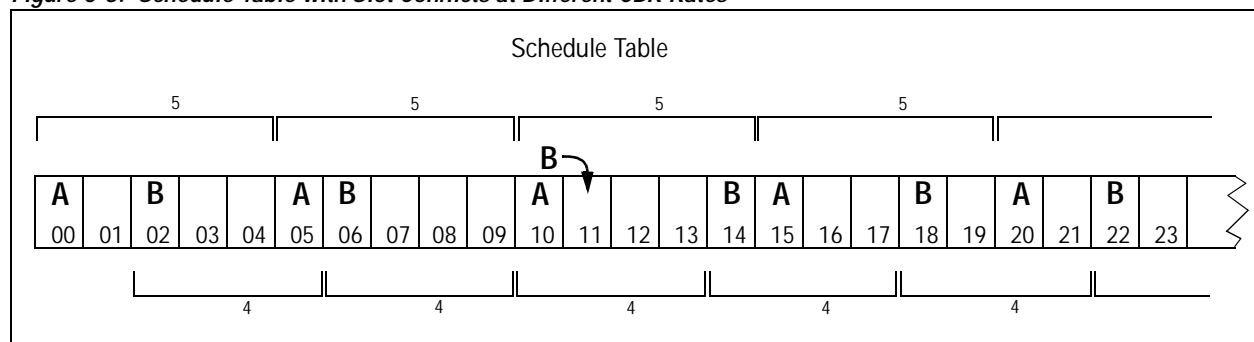
By definition, CBR connections are sensitive to CDV<sup>1</sup>. The RS8235's Traffic Manager minimizes CDV by basing all traffic management on the SYSCLK frequency, and providing the user the ability to explicitly decide the transmit time for CBR traffic. However, no system is without some CDV. In the case of terminals using the RS8235, 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. Illustrated is a linear representation of a Schedule Table with 100 schedule slots. CBR channel A has reserved bandwidth for a rate of 70.8 K cells/second, or every fifth cell slot. CBR channel B has reserved bandwidth for a rate of 88.7 K cells/second, or every fourth 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.

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

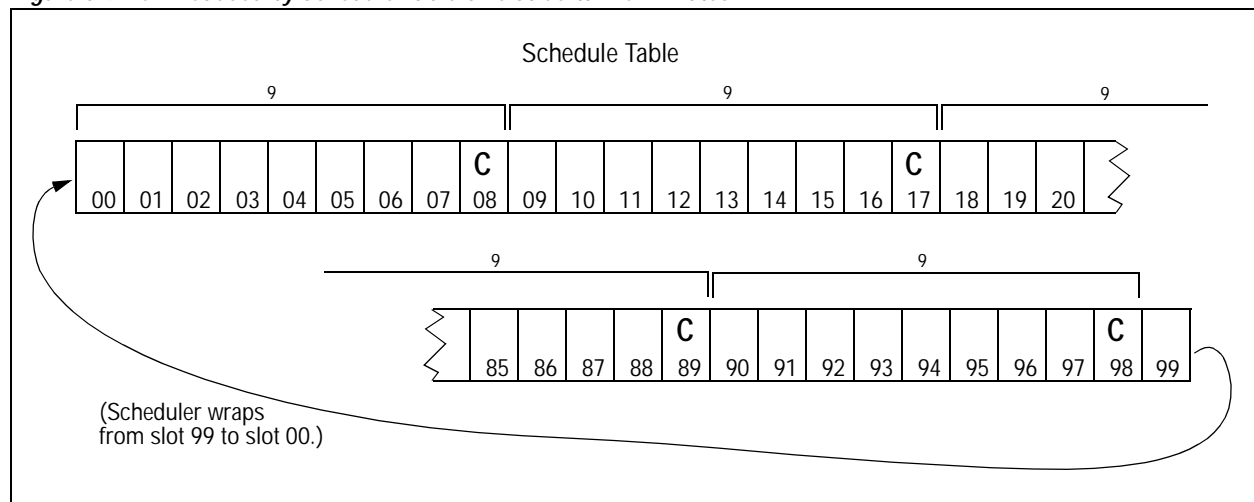


Another possible source of Schedule-Table-dependent CDV comes about for certain CBR rates whose schedule slot spacings do not evenly divide the table slot size. An example of this is illustrated in Figure 6-9 below. In this example, the beginning and end of a 100 slot Schedule Table is shown. The system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

<sup>1</sup> See ATM Forum UNI 3.1 or TM 4.1 for a complete definition of CDV.

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 ten slots, not nine.

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



The worst case CDV is determined by the following formula:

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

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:

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

#### 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). Thereafter, the RS8235 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 any CBR provisioned virtual connections (PVCs). This process takes place before the segmentation process is initiated. Once segmentation has begun, the RS8235 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 RS8235 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 RS8235 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 RS8235 encounters a CBR slot for this VCC, it will not transmit data on that VCC. Then the RS8235 indicates to the host that a slot has been skipped by clearing the SCH\_OPT bit.

With this method, the host effectively synchronizes the RS8235'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.

## 6.2.4 VBR Traffic

The RS8235 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 Mapping RS8235 VBR Service Categories to TM 4.1 VBR Service Categories

The ATM Forum TM 4.1 Specification describes the different categories of VBR service in a different manner than is employed in the RS8235 device. These relationships are described here:

<u>RS8235 VBR</u>	<u>TM 4.1 VBR</u>	<u>Comments</u>
VBR1	(none)	TM 4.1 does not employ single leaky bucket.
VBR2	VBR.1	Double leaky bucket.
VBR3 (or VBRC)	VBR.2 /VBR.3	TM 4.1 defines two conformance stds for CLP(0+1).

### 6.2.4.2 Rate Shaping vs. Policing

The Cell Scheduler rate shapes the segmentation traffic for up to 64 K 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 can 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 SEG VCC Table entry.

- 6.2.4.3 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 Cell Delay Variation Tolerance (CDVT) of the outgoing cell stream. The user enables this scheme by setting the SCH\_MODE bits to “100” (VBR1).
- 6.2.4.4 Dual Leaky Bucket** The user can 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 *I2* and *L2* 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 Sustained Cell Rate (SCR) or PCR. In VBR2, *I1* and *L1* can specify either PCR and CVDT, or SCR and Burst Tolerance (BT), with *I2* and *L2* used to specify the parameters not assigned to *I1* and *L1*.  
For example, you can configure:  
I1 = PCR, L1 = CDVT, I2 = SCR, L2 = BT  
or  
I1 = SCR, L1 = BT, I2 = PCR, L2 = CDVT
- 6.2.4.5 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 controls only the SCR of CLP=0 cells. The user enables this scheme by setting the SCH\_MODE bits to “110” (VBRC — also called VBR3).
- 6.2.4.6 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 some even integer of table slots (i.e., they can be some fractional value), they have a much finer rate granularity. The minimum rate for VBR channels is  $R_{\max}/(TBL\_SIZE-1)$ .
- 6.2.4.7 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:
- $$\frac{[(\# \text{ VBR VCCs at same-or-higher priority}) + TX\_FIFO\_LEN]}{\text{line rate in cells/sec}}$$

## 6.2.5 UBR Traffic

The remaining priority levels not already assigned to ABR, VBR, or CBR tunnel 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 can use a CBR tunnel in that priority level. Another method of limiting the bandwidth that a UBR priority consumes is described in [Section 6.2.8](#).

### 6.2.6 CBR Tunnels (Pipes)

A CBR tunnel occupies schedule table slots in the same manner as a CBR VCC. However, instead of serving one VCC, from one to four priority levels are served. The VCCs within a CBR tunnel can be UBR, VBR (VBR1 or VBR2), and/or ABR traffic. In the case of a UBR priority in a tunnel, the VCCs of the served priority level are serviced in round robin order. For any VBR/ABR priorities in a tunnel, each VCC is shaped to its GCRA parameters within the CBR tunnel.

**NOTE:** The format of a CBR tunnel schedule table slot is not backward-compatible with the CBR tunnel format used in the Bt8233 SAR.

When the user establishes a CBR tunnel, the user-defined transmit bandwidth for that tunnel is reserved in the schedule table as schedule slots. The one to four priority levels assigned to that tunnel are named at this time, and are written to the PRI3, PRI2, PRI1 and PRI0 fields in the CBR\_TUN\_ID field for the schedule slots reserved. PRI3 should specify the highest priority level for that tunnel, down to PRI0 as the lowest assigned priority level for that tunnel. The scheduler services the highest priority level that has data available on that priority queue at each point a schedule slot for that tunnel becomes active. This serves as a secondary shaping of the traffic assigned to these tunnel priorities. Please note that any one priority level can be assigned to only one CBR tunnel, and must not be assigned as a priority to more than one tunnel. Additionally, note that all priority levels utilized within a CBR tunnel need to be activated as a tunnel by setting the TUN\_ENA(x) bit(s) in the SCH\_PRI register.

Eight tunnels can be active at once, if each of these tunnels has only one scheduling priority assigned. All can be VBR/ABR tunnels. Or the system designer can establish two tunnels, each of which has four scheduling priorities assigned to it, or any combination of tunnels which include up to a total of eight scheduling priorities. 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.

If both non-tunnel and tunnel scheduling priorities exist, the host must assign the highest priority level(s) to CBR tunnel(s).

#### **APPLICATION EXAMPLES: Tunnels**

Figure 6-6 shows priorities five and six used as CBR tunnels for UBR and VBR traffic. 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 = 6 might currently be segmenting data. Each will get 1/100th of the CBR bandwidth assigned to the tunnel. Tunneling enables system level end users to purchase CBR services from a WAN service provider. The purchaser can then dynamically manage the traffic within this leased CBR tunnel as CBR and/or a combination of other service categories.

In this example, the user has configured the RS8235 to manage two independent tunnels. One, priority five, is through a private ATM network, perhaps a corporate ATM campus backbone. The other, priority six, 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.

## 6.2.7 Guaranteed Frame Rate

GFR is a new service category defined by the ATM Forum to provide a Minimum Cell Rate (MCR) QoS guarantee for AAL5 CPCS-PDUs (or frames) not exceeding a specified frame length. This class of service is designed specifically to utilize the UBR service category. A GFR service connection is thus treated as UBR with a guaranteed MCR.

The RS8235 implements GFR by scheduling/shaping the connections using both the VBR1 scheduling procedure (for the MCR rate value) and a UBR priority queue, thereby providing fair sharing for all GFR connections to excess bandwidth. The VBR1 queue priority (the PRI field in the SEG VCC Table entry) should be set to a higher priority than the UBR queue (the GFR\_PRI field in the SEG VCC table entry). The priority level entered in the GFR\_PRI field can only be one of the lower eight scheduling priorities. This establishes the condition where those GFR connections sharing the same GFR\_PRI would fair-share any excess bandwidth above the MCR limits. Call control must be sure not to oversubscribe MCR on GFR channels and other rate-guaranteed services.

The RS8235 helps guarantee MCR on a GFR channel by providing a mechanism to trigger an increase in the scheduling priority by one priority level, if the transmit rate on that channel falls below its specified MCR. This is done using the MCRLIM\_IDX field in that VCC's SCH\_STATE entry in the SEG VCC Table entry. MCRLIM\_IDX is an 8-bit index into a table containing 256 entries, each formatted as described in [Table 6-15](#), each containing MCR limit values, and each indicating a trigger point for an increase in the VCC's priority for that scheduled slot.

To disable this priority bumping, set MCR\_LIMIT to its maximum value. The user can accomplish this by setting each of the values in the GFR MCR Limit bucket table entries described in [Table 6-15](#) as follows:

1. Set NONZERO bits to 1.
2. Set MCR\_EXP fields to decimal value of 31 (i.e., all-ones).
3. Set MCR\_MAN fields to decimal value of 511 (i.e., all-ones).

An additional level of traffic shaping can be established on GFR-UBR priority queues by shaping to a specified PCR. (See [Section 6.2.8](#) below.)

## 6.2.8 PCR Control for Priority Queues

The RS8235 provides an optional method of creating tunnels (i.e., of limiting the bandwidth of a group of channels), by allowing the user to assign a PCR to a scheduling priority queue, so that the aggregate of the rates of the channels assigned to that priority queue will not be greater than the PCR assigned. This can be done for UBR, VBR, ABR and UBR-GFR traffic classes.

A maximum of four of the eight priority queues can be shaped to a PCR less than line rate. The queues to be shaped are identified using the QPCR\_ENAx bits in the SCH\_PRI register. The associated PCR for each queue thus enabled is specified in one of the two PCR\_QUE\_INTxx registers. These PCR values are stored as schedule table intervals. QPCR\_INT3 maps to the highest priority queue that is enabled for PCR shaping, QPCR\_INT2 to the next highest priority PCR shaped queue, and so on. If only one priority queue is enabled for PCR shaping, it will be shaped using the QPCR\_INT3 value; if two priority queues are enabled for PCR shaping, QPCR\_INT3 and QPCR\_INT2 will be used; and so on.

## 6.3 ABR Flow Control Manager

### 6.3.1 A Brief Overview of TM 4.1

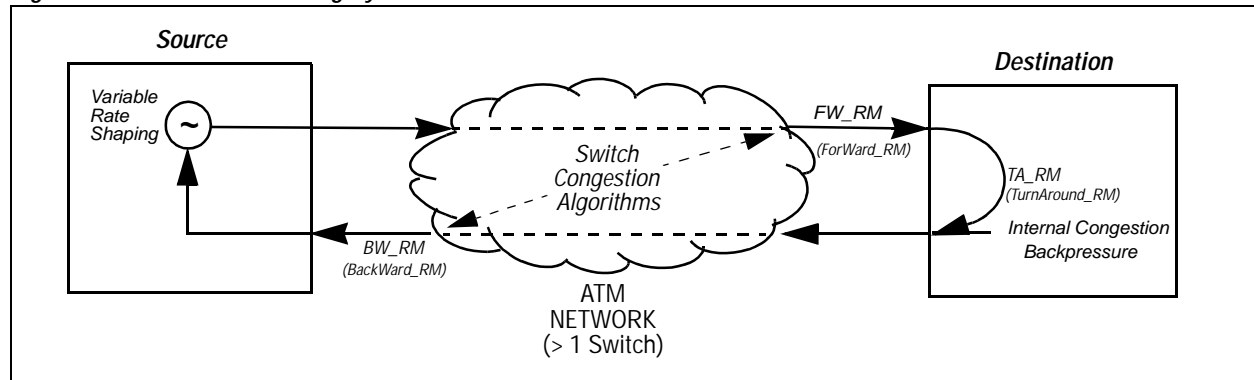
This section briefly describes the TM 4.1 ABR Flow Control algorithms. However, it is strongly recommended that the reader be familiar with the ATM Forum's TM 4.1 specification before attempting to understand the RS8235's ABR implementation.

### 6.3.2 Internal ABR Feedback Control Loop

As a complete implementation of the UNI ATM layer, the RS8235 acts as both an ABR Source and Destination, complying with all required TM 4.1 ABR behaviors. The RS8235 uses 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.

Figure 6-10 illustrates this basic concept.

Figure 6-10. ABR Service Category Feedback Control

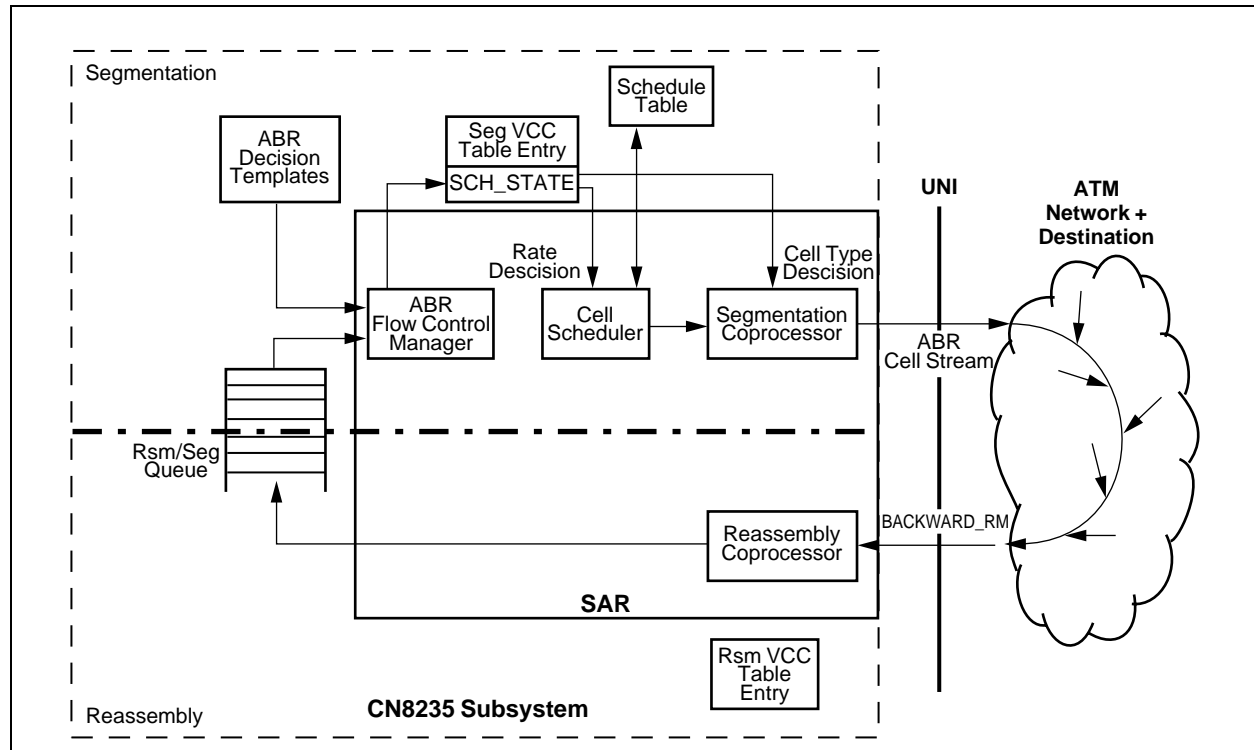


#### 6.3.2.1 Source Flow Control Feedback

Figure 6-11 illustrates the feedback loop which controls the Source cell stream. The RS8235 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 RS8235 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 SAR 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 TM 4.1.

Figure 6-11. RS8235 ABR-ER Feedback Loop (Source Behavior)



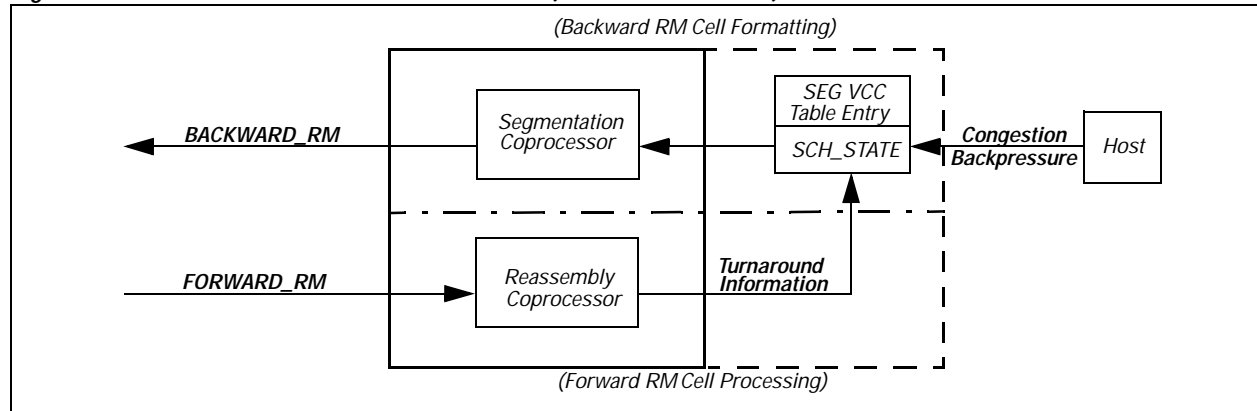
### 6.3.2.2 Destination Behavior

The RS8235 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 via the SCH\_STATE part of the segmentation VCC Table entry. The segmentation coprocessor then formats Backward RM cells containing this information and inserts these turnaround RM cells into the transmit cell stream.

The RS8235 also provides a mechanism for the Destination host to apply backpressure to the Source. The host notifies the RS8235 of internal system congestion. The RS8235 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 TM 4.1.

This process is illustrated in Figure 6-12.

Figure 6-12. RS8235 ABR-ER Feedback Generation (Destination Behavior)



### 6.3.2.3 Out of Rate Cells

In addition to in-rate cell streams, the RS8235 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

ABRs Source and Destination Behavior Definitions are each listed in the ATM Forum's TM 4.1 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 portion of the segmentation VCC Table entry. Due to the large number of ABR parameters, the SCH\_STATE of ABR VCCs requires an additional location in the SEG VCC Table.

### 6.3.5 ABR Templates

The ABR Templates reside in SAR 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. The key parameters contained in and controlled by the ABR Templates are: Peak Cell Rate (PCR), Initial Cell Rate (ICR), Minimum Cell Rate (MCR), Rate Increase Factor (RIF), Rate Decrease Factor (RDF), and Nrm.

The user has the choice of defining MCR and ICR from the ABR Templates or from fields in the SCH\_STATE portion of the SEG VCC Table entry for each connection, thus giving either per-template or per-connection control of MCR and ICR. This choice is globally set using the ADV\_ABR\_TEMPL bit in the SEG\_CTRL register.

These ABR Templates are furnished by Mindspeed, and are downloaded to the RS8235 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 allows the network-side costs of connections to be significantly reduced. 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 that 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 RS8235's Implementation of GFC

The RS8235 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 sub-section 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 be sent only 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.

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.

### 6.4.3 Configuring the Link for GFC Operation

The following describes an example 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 the GFC initialization timer expires, 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 displayed 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/ABR pointer)															
1 <sup>a</sup>	(Reserved for 2 16-bit VBR/ABR pointers)																															

<sup>a</sup>. Word one is present only when the DBL\_SLOT field in SEG\_CTRL is set, meaning two words per schedule slot.

### 6.5.2 CBR-Specific Structures

**6.5.2.1 CBR Traffic** Dedicate a schedule slot to a CBR connection by formatting the CBR bit to a logic high and the CBR\_TUN\_ID field in the slot entry as displayed in [Table 6-4](#).

**Table 6-4. CBR\_TUN\_ID Field, Bit Definitions — CBR Slot**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Def.	1	CBR_VCC_INDEX (15 bits)														

**6.5.2.2 Tunnel Traffic** Dedicate a schedule slot to a CBR tunnel by formatting the CBR bit to a logic low and the CBR\_TUN\_ID field of the slot entry as displayed in [Table 6-5](#).

**Table 6-5. CBR\_TUN\_ID Field, Bit Definitions — Tunnel Slot**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Def.	0	PRI3			PRI2			PRI1			PRI0					

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

Field Name	Description
CBR	Set high to indicate a CBR slot; set low to indicate a tunnel slot.
PRI3	Specifies the highest priority of four possible scheduling priority queues to service when a scheduling slot for this CBR tunnel becomes active.
PRI2	Specifies the second highest priority of four possible scheduling priority queues to service when a scheduling slot for this CBR tunnel becomes active.
PRI1	Specifies the third highest priority of four possible scheduling priority queues to service when a scheduling slot for this CBR tunnel becomes active.
PRI0	Specifies the lowest priority of four possible scheduling priority queues to service when a scheduling slot for this CBR tunnel becomes active.
CBR_VCC_INDEX	Segmentation VCC Index for dedicated CBR schedule slot. CBR VCC indexes range from 0x0000 to 0x7FFE.

**NOTE:** If the user wants only one priority of traffic scheduled in the CBR tunnel, then assign the priority level to PRI3 and make PRI2 the same priority. PRI1 and PRI0 values are “don’t cares” in this case. If two priorities are to be assigned to the tunnel, then assign the higher priority to PRI3 and the lower priority to PRI2, making PRI1 the same as PRI2. PRI0 is a “don’t care” in this case. If three priorities are to be assigned to the tunnel, then assign the priorities by level to PRI3 (highest), PRI2 and PRI1 (lowest), making PRI0 the same as PRI1. Note that TUN\_PRI0\_OFFSET can be used to set PRI0 = PRI1, as needed.

**6.5.2.3 SCH\_STATE Fields For CBR** Table 6-7 and Table 6-8 specify the SCH\_STATE portion (words 7 through 9) of the Segmentation VCC Table entry, when SCH\_MODE = CBR.

Table 6-7. SCH\_STATE for SCH\_MODE = CBR

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	Reserved														TUN_PRI_3		TUN_PRI_2		TUN_PRI_1		TUN_PRI_0											
8	Reserved																															
9	Reserved																															

Table 6-8. CBR SCH\_STATE Field Descriptions (1 of 2)

Field Name	Description
TUN_PRI_3	When SCH_MODE is CBR and CBR_W_TUN bit is set to 1, this field specifies the highest priority of four possible scheduling priority queues to service in place of the unused CBR slot. (This field is not active when CBR_W_TUN = 0.)

Table 6-8. CBR SCH\_STATE Field Descriptions (2 of 2)

Field Name	Description
TUN_PRI_2	When SCH_MODE is CBR and CBR_W_TUN bit is set to 1, this field specifies the second highest priority of four possible scheduling priority queues to service in place of the unused CBR slot. (This field is not active when CBR_W_TUN = 0.)
TUN_PRI_1	When SCH_MODE is CBR and CBR_W_TUN bit is set to 1, this field specifies the third highest priority of four possible scheduling priority queues to service in place of the unused CBR slot. (This field is not active when CBR_W_TUN = 0.)
TUN_PRI_0	When SCH_MODE is CBR and CBR_W_TUN bit is set to 1, this field specifies the lowest priority of four possible scheduling priority queues to service in place of the unused CBR slot. (This field is not active when CBR_W_TUN = 0.)

### 6.5.3 VBR-Specific Structures

#### 6.5.3.1 VBR SCH\_STATE

Table 6-9 and Table 6-10 specify the SCH\_STATE part of the Segmentation VCC Table entry, which consists of words seven through nine for VBR.

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

#### 6.5.3.2 VBR1 or VBR2 Schedule State Table

Table 6-9. 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																															

Table 6-10. VBR1 and VBR2 SCH\_STATE Field Descriptions (1 of 2)

Field Name	Description
BUCKET2	Index into Bucket table to give I2 and L2 for SCHED_MODE = VBR2 & VBRC. The bucket table base is given by the SEG_BCKB register field. Entries in the bucket table have the 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 $L1\_EXP \leq \min(I1\_EXP + 9, 29)$ . L1_EXP that does not satisfy $L1\_EXP \geq I1\_EXP - 9$ will have an effective L1 value of zero.

Table 6-10. VBR1 and VBR2 SCH\_STATE Field Descriptions (2 of 2)

Field Name	Description
L1_MAN	GCRA L parameter mantissa for bucket 1. Bucket 1 L value is: $2^{(L1\_EXP - 10)}(1+L1\_MAN/512)$ .
I1_EXP	GCRA I parameter exponent for bucket 1. I1_EXP must satisfy $10 \leq I1\_EXP \leq 25$ .
I1_MAN	GCRA I parameter mantissa for bucket 1. Bucket 1 I value is: $2^{(I1\_EXP - 10)}(1+I1\_MAN/512)$ .

### 6.5.3.3 Bucket Table for VBR2 and VBRC

The Bucket Table has only 256 entries. Table 6-11 and Table 6-12 display the format and description of the Bucket Table.

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

## 6.5.4 GFR-Specific Structures

### 6.5.4.1 GFR Schedule State Table

Table 6-13. SCH\_STATE for SCH\_MODE = GFR

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	MCRLIM_IDX (MSB)				L1_EXP				L1_MAN								I1_EXP				I1_MAN															
8	MCRLIM_IDX (LSB)				Reserved																															
9	Rsvd				Reserved								Reserved								Reserved															

Table 6-14. SCH\_STATE Field Descriptions for SCH\_MODE = GFR

Field Name	Description
MCRLIM_IDX	Index into table of 256 MCR LIMIT values each specifying an increase in the VCC's priority by one if the VCC's rate falls below MCR. The table base is located on top of the BUCKET2 table whose base is given by SEG_BCKB register field. The BUCKET2 table is 1KB in length. Table values are in the form of a non-zero bit, exponent, and mantissa such that MCR_LIMIT is: $\text{MCR\_LIM\_NZ} * 2^{(\text{MCR\_LIM\_EXP}-10)}(1+\text{MCR\_LIM\_MAN}/512).$
L1_EXP	GCRA L parameter exponent for MCR bucket. L1_EXP must satisfy: $L1\_EXP \leq \min(I1\_EXP + 9, 29)$ . L1_EXP that does not satisfy $L1\_EXP \geq I1\_EXP - 9$ will have effective L1 value of zero.
L1_MAN	GCRA L parameter mantissa for MCR bucket Bucket 1 L value is: $2^{(L1\_EXP - 10)}(1+L1\_MAN/512)$ .
I1_EXP	GCRA I parameter exponent for MCR bucket. I1_EXP must satisfy $10 \leq I1\_EXP \leq 25$ .
I1_MAN	GCRA I parameter mantissa for MCR bucket. Bucket 1 I value is: $2^{(I1\_EXP - 10)}(1+I1\_MAN/512)$ .

**6.5.4.2 GFR MCR Limit Bucket Table**

The 128 words of this table contain 256 MCR Limit values, two bucket table entries per word. The table base is on top of the Bucket2 Table. Table values are in the form of a non-zero bit, exponent and mantissa such that MCR\_LIMIT is:

$$MCR\_LIM\_NZ \cdot 2^{(MCR\_LIM\_EXP \ominus 10)} (1 + MCR\_LIM\_MAN/512) .$$

Table 6-15 and Table 6-16 display the format of the MCR Limit Bucket Table.

Table 6-15. GFR MCR Limit 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	NONZERO	MCR_EXP					MCR_MAN					Rsvd	NONZERO	MCR_EXP					MCR_MAN												

Table 6-16. GFR MCR Bucket Table Entry Field Descriptions

Field Name	Description
MCR_EXP	GFR MCR limit exponent. MCR_EXP must satisfy MCR_EXP <= 29.
MCR_MAN	GFR MCR limit mantissa. MCR limit is: NONZERO * 2 <sup>(MCR_EXP-10)</sup> (1+MCR_MAN/512). VCC priority will be increased by one when rate falls below 1/MCR limit.
NONZERO	GFR MCR limit is non-zero.

### 6.5.5 ABR-Specific Structures

6.5.5.1 ABR Schedule State Table Table 6-17 and Table 6-18 display the format and descriptions of the SCH\_STATE portion of the VCC Table for the ABR service class.

**KEY:**

 = Values are furnished by the ABR Templates.

Table 6-17. SCH\_STATE for SCH\_MODE = ABR

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																															
9	Rsvd		MCR_LIM_NZ	DELTA_SIGN	DELTA_NZ	MCR_LIM_EXP			MCR_LIM_MAN						DELTA_EXP			DELTA_MAN														
10	CONG_ID			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	Rsvd		CRM												UNACK																	
14	Reserved												NEXT_OOR																			
15	MCR_INDEX												ICR_INDEX																			
16 <sup>(1)</sup>	TA_ID				TA_DIR	TA_BN	TA_CI	TA_NI	TA_RA	Rsvd						TA_ER																
17 <sup>(1)</sup>	TA_CCR												TA_MCR																			
18	FWD_ID				FWD_DIR	FWD_BN	FWD_CI	FWD_NI	FWD_RA	Rsvd						FWD_ER																
19	Reserved												FWD_MCR																			

**NOTE(S):**  
<sup>(1)</sup> Words 16 and 17 are written directly by the RSM coprocessor (turnaround information).

Table 6-18. 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.
PRESENT	Current schedule table position.
MCR_LIM_NZ	MCR Limit is non-zero. To disable priority bumping, MCR_LIMIT must be set to its maximum value. Thus, MCR_LIM_NZ must be set to 1.
DELTA_SIGN	Delta is negative for ER rate.
DELTA_NZ	Delta is non-zero for ER rate.
MCR_LIM_EXP	MCR Limit exponent. MCR_LIM_EXP must satisfy $MCR\_LIM\_EXP \leq 29$ . To disable priority bumping, MCR_LIMIT must be set to its maximum value. Thus, MCR_LIM_EXP must be set to a decimal value of 31 (i.e., binary all-ones).
MCR_LIM_MAN	MCR Limit mantissa. MCR_LIMIT is: $MCR\_LIM\_NZ \times 2^{(MCR\_LIM\_EXP-10)} (1+MCR\_LIM\_MAN/512)$ . This format for calculating MCR_LIMIT is the same as used with the GCRA I parameter to calculate rates. VCC priority will be increased by one when the rate falls below $1/MCR\_LIM$ . To disable priority bumping, MCR_LIMIT must be set to its maximum value. Thus, MCR_LIM_MAN must be set to a decimal value of 511 (i.e., binary all-ones).
DELTA_EXP	Delta exponent for ER rate.
DELTA_MAN	Delta mantissa for ER rate. Desired position for ER rate is: $DELTA\_NZ \times 2^{(DELTA\_EXP-10)} (1+DELTA\_MAN/512)+PRESENT$ .
CONG_ID	Congestion Index for changing ER on turnaround RM cells. The ER in the turnaround cell is changed only if the FBO_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.
OOD_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.

Table 6-18. ER SCH\_STATE Field Descriptions (2 of 2)

Field Name	Description
CELL_INDEX	ER cell type decision block index for cell type decisions. The cell type decision block is located at byte address $[(SCH\_ABRB \times 128) + (8 \times CELL\_INDEX)]$ .
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 $[(SCH\_ABRB \times 128) + (32 \times RATE\_INDEX)]$ .
EB_INDEX	ER exponent table index for rate index block. The exponent table is located at byte address $[(SCH\_ABRB \times 128) + (EB\_INDEX \times 128)]$ .
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.
MCR_INDEX	Minimum Cell Rate decision block index. The rate decision block is located at byte address $[(SCH\_ERB \times 128) + (32 \times MCR\_INDEX)]$ .
ICR_INDEX	Initial Cell Rate decision block index. The rate decision block is located at byte address $[(SCH\_ERB \times 128) + (32 \times ICR\_INDEX)]$ .
TA_ID	ID field from most recent received Forward RM cell. This field is written by the SAR.
TA_DIR	DIR field for turnaround (Backward) RM cell. This field is written by the SAR.
TA_BN	BN field for turnaround (Backward) RM cell. This field is written by the SAR.
TA_CI	CI field for turnaround (Backward) RM cell. This field is written by the SAR.
TA_NI	NI field for turnaround (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 turnaround (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.
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.

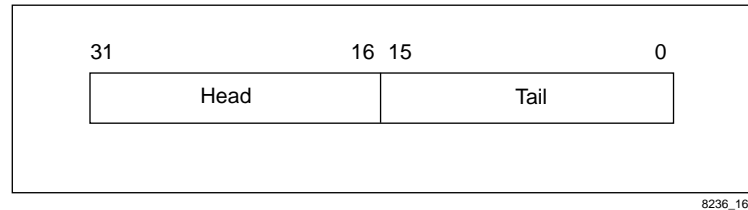
### 6.5.6 Scheduler Internal SRAM Registers

**NOTE:** FOR SAR INTERNAL USE ONLY!

Application program should ignore these registers.

The following graphic is a layout for the head and tail pointers .

**Figure 6-13. Head and Tail Pointers**



Scheduler SRAM registers are located in the address range 0x1640–00x17FF.

**Table 6-19. Scheduler Internal SRAM Memory Map**

Address	Name	Description
0x1640–0x1643	PRI_PNTR0	Global priority 0.
0x1644–0x1647	PRI_PNTR1	Global priority 1.
0x1648–0x164B	PRI_PNTR2	Global priority 2.
0x164C–0x164F	PRI_PNTR3	Global priority 3.
0x1650–0x1653	PRI_PNTR4	Global priority 4.
0x1654–0x1657	PRI_PNTR5	Global priority 5.
0x1658–0x165B	PRI_PNTR6	Global priority 6.
0x165C–0x165F	PRI_PNTR7	Global priority 7.
0x1660–0x17FF	Reserved	Not implemented.



# 7.0 OAM Functions

---

## 7.1 OAM Overview

Operation, Administration, and Maintenance (OAM) cells are ATM Layer management messages. These are generated by the host on the segmentation side of the RS8235, and are detected and either monitored or processed on the reassembly side of the RS8235.

ATM's OAM capabilities differentiate it from other less robustly managed communication technologies. The RS8235 provides internal support for the detection and generation of OAM traffic.

The RS8235 supports the F4 and F5 OAM flows according to ITU-T Recommendation I.610.

### 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 RS8235 internally supports the following functions as described in I.610:

11.
  - Detection 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 Virtual Path level, provided by OAM cells dedicated to ATM layer OAM functions for Virtual Path Connections (VPCs). The F4 flow is bidirectional.

OAM cells for the F4 flow have the same Virtual Path Identifier (VPI) value as the user cells of the VPC.

F4 flow OAM cells are identified as F4 flow cells by a pre-assigned Virtual Channel Identifier (VCI) value of three (segment flow cell) or four (end-to-end flow cell). The same pre-assigned VCI value is used for both directions of the F4 flow.

**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 Virtual Channel Connections (VCCs). The F5 flow is bidirectional.

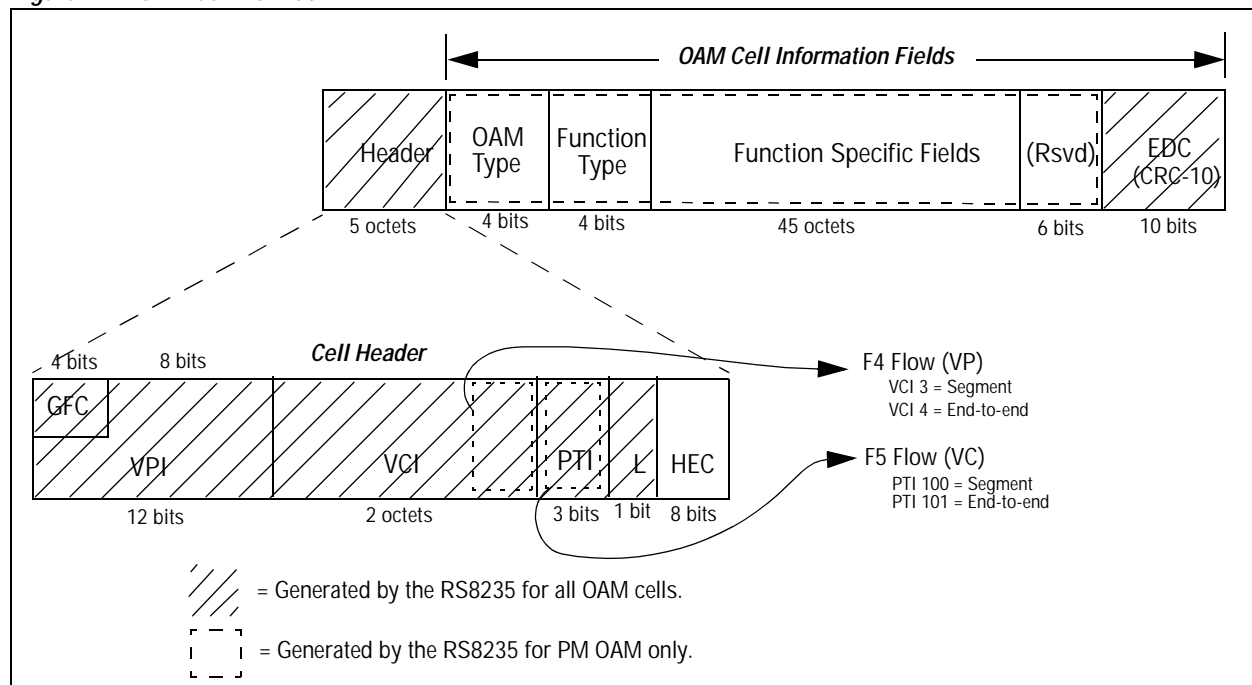
OAM cells for the F5 flow have the same VPI value and VCI value as the user cells of the VCC.

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.

### 7.1.3 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.

## 7.2 Segmentation of OAM Cells

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

The host processor then writes a pointer to that SBD in the next available transmit queue entry, and sets the VLD bit to one.

When the segmentation coprocessor processes that transmit queue entry, it submits the OAM cell data buffer to the xBR Traffic Manager and the cell thus is 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 SBD 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 RS8235 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.
  - Set the single-bit HEADER\_MOD field to a logic one. This activates the WR\_PTI and WR\_VCI bits in the buffer descriptor, which signal the RS8235 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 RS8235.
  - The VCI\_DATA field set to a value of three (segment cell) or four (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 RS8235 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 can 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 Segmentation

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 RSM\_CTRL1 to a logic high.

The RS8235 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 RS8235 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 RS8235 then routes OAM traffic to these global buffer and status queues.

**NOTE:** The cell buffer size must be large enough to hold a complete cell, when OAM detection is enabled.

### 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 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 RS8235 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.

**NOTE:** OAM cells that get buffers from the global OAM free buffer queue do not effect the per-VCC firewall calculation.

### 7.3.3 Error Conditions During OAM Reassembly

The RSM coprocessor computes the CRC-10 error detection code over each 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 OAM Control and Status Structures

Refer to the “Control and Data Structures” sections of [Chapter 4.0](#) (Segmentation Coprocessor) and [Chapter 5.0](#) (Reassembly Coprocessor) for information on VCC Tables, buffer descriptors, the transmit queue, and status queues, as they apply to generating and processing OAM cells.

# 8.0 DMA Coprocessor

---

## 8.1 Overview

The Direct Memory Access (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 does this by using burst transfers when possible to maximize utilization of the host bus bandwidth, performing byte switching to accommodate misaligned transfers, and carrying 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 correspond to one cell. 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.

## 8.3 DMA Write

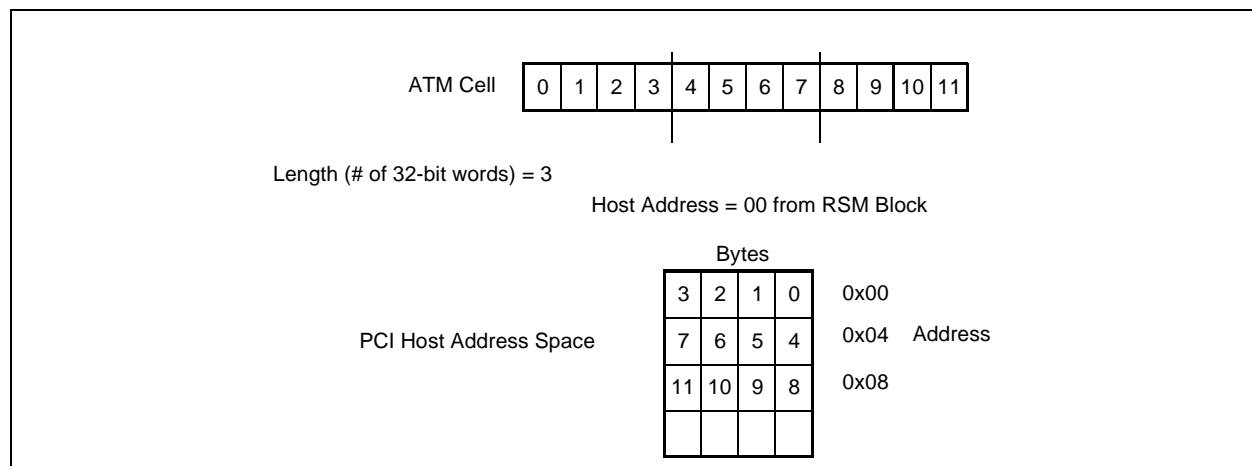
For incoming messages, DMA write cycles move data from the reassembly coprocessor to host memory using a scatter DMA method. The maximum burst size is fourteen 32-bit words which correspond to one ATM cell. The burst size can be reduced by setting the MAX\_BURST\_LEN field in the PCI Configuration Register at a value less than 13 words.

## 8.4 Misaligned Transfers

The reassembly and segmentation coprocessors 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 RS8235. When the RS8235 specifies a host address with the Least Significant Bits (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 RS8235 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

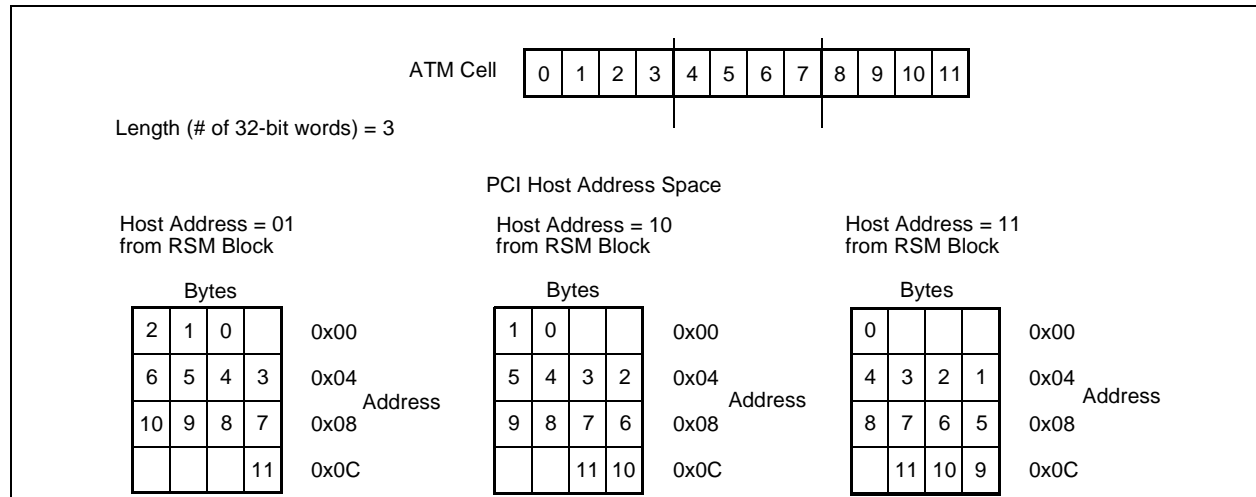
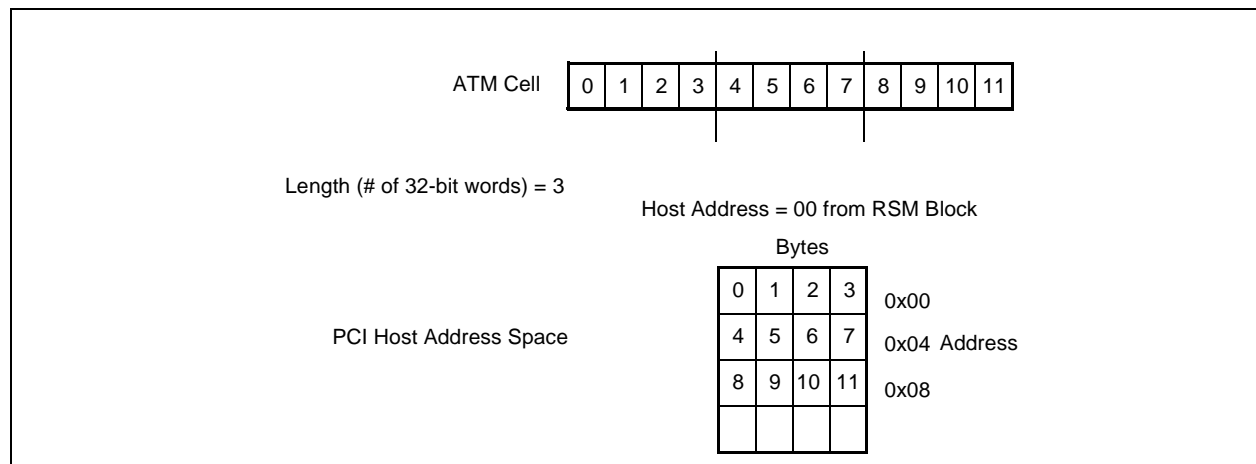


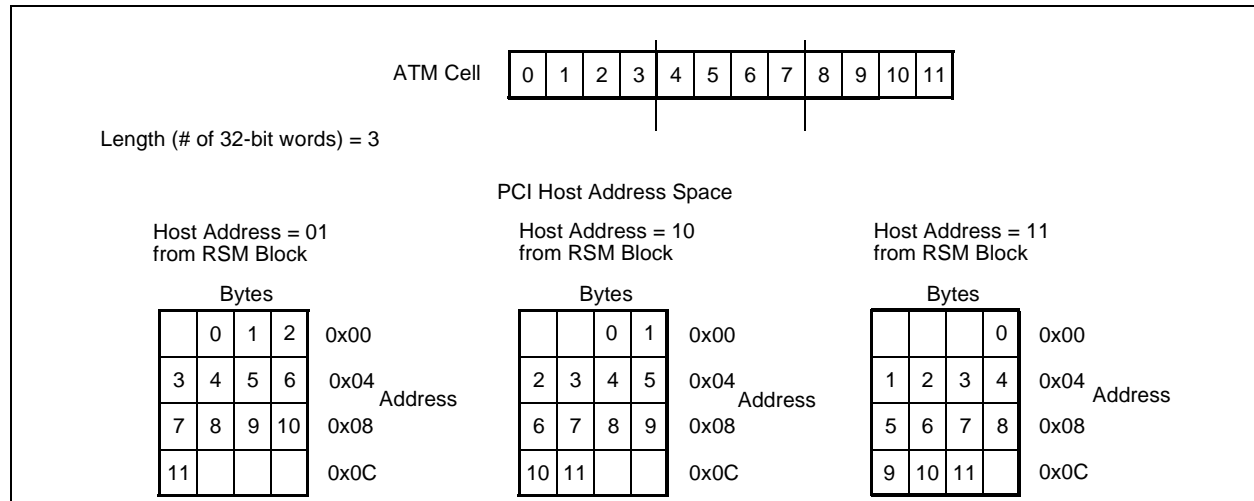
Figure 8-3 shows how a byte-aligned address would map into the PCI host address space for a big endian system.

Figure 8-3. Big Endian Aligned Transfer



When the RS8235 specifies a host address with the LSBs not equal to 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



## 8.5 Control Word Transfers

If a host system sends control words to the SAR in little endian format, the reassembly and segmentation blocks must have the capability of byte swapping, to format these control words to or from big endian.

Control word byte swapping is controlled by bits 30 (Master Control Byte Swap, `MSTR_CTRL_SWAP`) and 29 (Slave Control Byte Swap, `SLAVE_SWAP`), located in the Special Status Register of the PCI Configuration Space (address 0x40).

When `SLAVE_SWAP` is a logic high, the slave interface swaps the bytes of a slave write or read access. When `MSTR_CTRL_SWAP` is a logic high, the control structures that the SAR writes are written with bytes swapped.

An active `HRST*` will cause these bits to be a logic low.

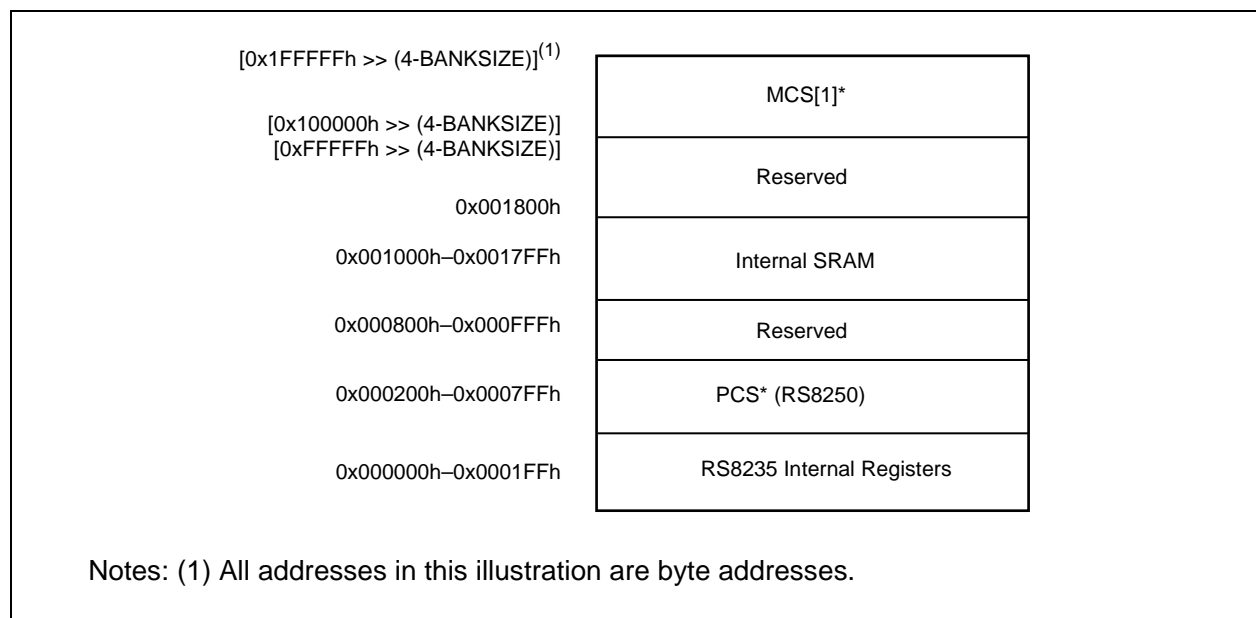
# 9.0 Local Memory Interface

## 9.1 Overview

To simplify system implementations, the RS8235 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 standalone mode of operation, are mapped into the bottom of the memory map. Consequently, accesses to these resources are also controlled by the memory controller.

Up to 1 MByte of external memory using SRAM devices can be accessed by the RS8235. 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](#). [Figure 9-1](#) shows the RS8235 address map.

Figure 9-1. RS8235 Memory Map



## 9.2 Memory Bank Characteristics

The external memory is organized into one bank and selected by MCS[1]\*. The starting address of the external memory is a function of the quantity of memory selected. BANKSIZE[2:0] (bits 23–21) in the CONFIG0 register denote the size of the memory banks. If BANKSIZE = 000, then the starting address of the external memory is 0x10000h (byte address). [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)	PCI Address Bits	Typical Implementation
111	Reserved	—	—	—
110	Reserved	—	—	—
101	Reserved	—	—	—
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 14.0, for detailed timing information.)

Figure 9-2. 0.5 MB SRAM Bank Using by\_8 Devices

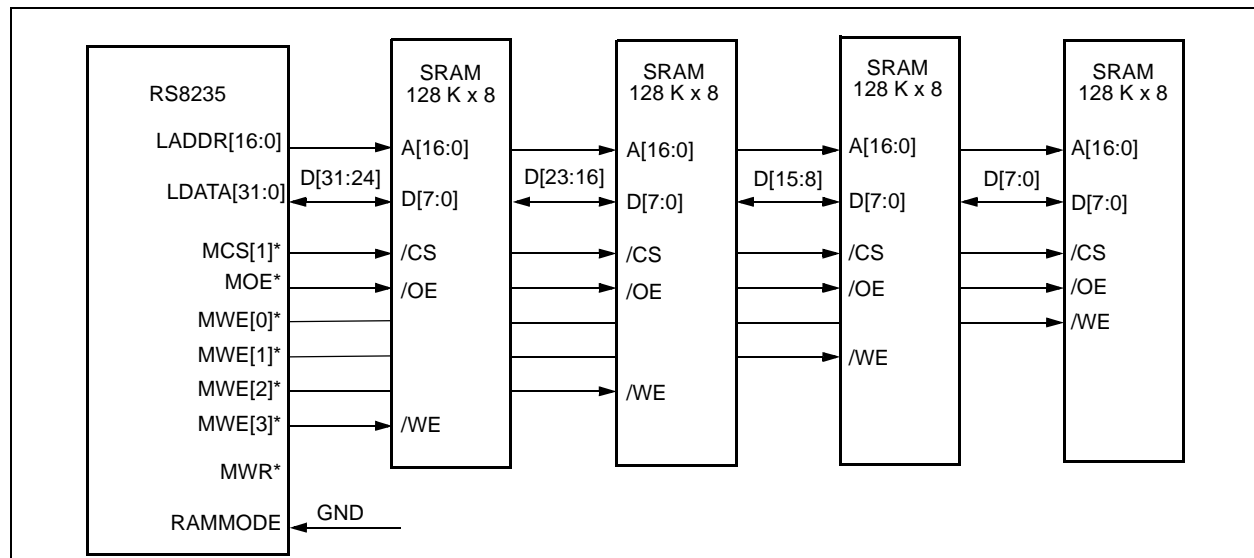
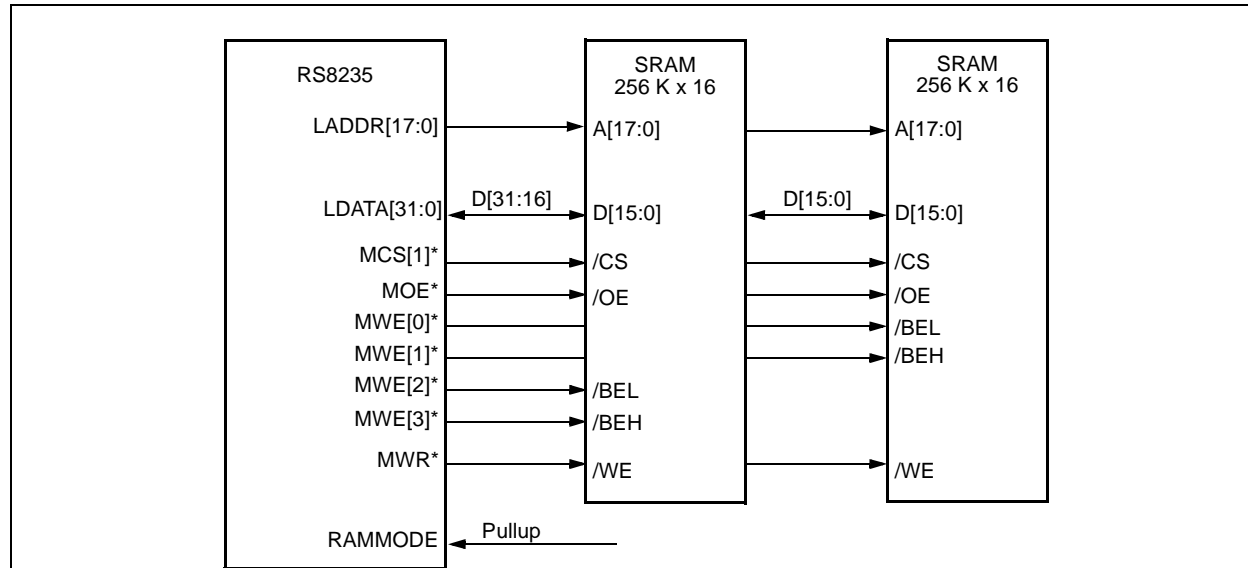


Figure 9-3. 1 MB SRAM Bank Using by\_16 Devices



The memory map contains space allocated to the RS8250 physical interface IC, and to a future second Mindspeed PHY device. This mapping is valid for standalone operation, when the PROCMODE input pin is pulled high. In the RS8235 the PROCMODE pin must be pulled high. Standalone operation is detailed in [Section 11.2](#).

The MEMCTRL bit in the CONFIG0 Register selects the number of wait states that the memory controller uses to access the SRAM. A logic zero indicates zero wait state or single-cycle memory, while a logic one 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 one, two, and four 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 can choose the appropriate SRAM characteristics to suit the amount of memory and organization required for the application. See [Chapter 14.0](#) for timing information.

## 9.3 Memory Size Analysis

Table 9-2 gives the segmentation memory size requirements for 1,024 configured VCCs under the following assumptions:

### SEGMENTATION

1. Schedule Table is 2,112 schedule slots and each slot is a double word. This allows CBR and three-priority VBR schedule in 64 Kbps increments for an OC-3 connection.
2. Eight ABR Templates.
3. Transmit queues configured for 256 entries.
4. No status queues in SAR local memory.
5. Each active channel has an average of one active segmentation buffer.
6. RS\_Queue size is 1,024.

### REASSEMBLY

Table 9-2. Memory Size in Bytes

Data Structure	One Peer	4 Peers
SEG Schedule Table	16,896	16,896
RSM Queue	8,192	8,192
SEG Transmit Queues	1,024	4,096
SEG ER Tables	67,200	67,200
RSM Free Buffer Queues	4,096	16,384
RSM VPI Index Table	1,024	1,024
Total Fixed	98,432	113,792
SEG VCC Table	40,960	40,960
SEG Buffer Descriptors	20,480	20,480
RSM VCC Table	49,152	49,152
RSM VCI Index Table	92	92
Total Incremental	110,684	110,684
Grand Total	209,116	224,476

1. Eight VPIs per 1,024 channels.
2. 1 K preallocation on VPI=0 only.
3. UNI VPI space.
4. Free buffer queues configured for 256 entries.
5. No status queues in SAR local memory.
6. Free buffer queue pools 0–1 have 4-word entries, and pools 2–3 have 2-word entries.



# 10.0 PCI Bus Interface

---

## 10.1 Overview

The PCI bus interface is compliant with PCI Local Bus Specification, Revision 2.1. 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 RS8235 to be valid before the next rising edge of HCLK.

The maximum PCI bus clock rate supported by the RS8235 is 33 MHz. The PCI bus interface logic is clocked directly from the PCI bus clock, while the remainder of the RS8235 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 RS8235 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 RS8235 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 RS8235 registers via the PCI slave port.
- Logic to enable read/write access to the SAR shared memory space from the host CPU, again via the PCI slave port.
- An Interface module that allows the PCI core to connect to a serial EEPROM.

## 10.2 Unimplemented PCI Bus Interface Functions

The PCI bus interface on the RS8235 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 Dual Address Cycle 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 RS8235, 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.

## 10.3 PCI Configuration Space

In accordance with the PCI bus specification Revision 2.1, the RS8235 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 is given in the PCI bus specification, and will not be repeated here. The descriptions and definitions of these register fields as implemented in the RS8235 is shown in Chapter 12, RS8235 Registers.

## 10.4 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:

- A command decoder which interprets commands issued from the DMA coprocessor.
- A burst controller that counts off read and write cycles in each burst on the PCI bus (and also latches and drives the address and command during the address phase of each transfer).
- Arbitration logic that acquires control of the PCI bus.
- Supported arbitration parking.
- A bus state machine that sequences and controls transfers.

It is possible for the addressed slave to 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, then 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, then 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 RS8235 PCI bus master logic (using the M\_EN bit in the Command field of the PCI bus configuration registers), then 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.

**NOTE(S):** The above errors permanently affect 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 one 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 can also determine if the transaction was a read or write by inspecting the MRD bit, and then retrieve the read or write address at which an error occurred by reading the MASTER\_READ\_ADDR or MASTER\_WRITE\_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.

## 10.5 Burst FIFO Buffers

Two small FIFO buffers are implemented to support PCI burst-mode operation, to allow synchronization between the RS8235 internal logic and the PCI bus interface, and to 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.

## 10.6 PCI Bus Slave Logic

The PCI slave logic permits the host CPU on the PCI bus to access and modify RS8235 resources (the external SAR shared memory, internal memory, and internal registers). Because 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. These registers control the PCI slave and master interfaces, and might 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 RS8235 system clock. Also, small FIFOs are used to speed up burst reads (8x32) and writes (64x32) performed by the host processor to local resources, by buffering prefetched read data and absorbing latency during consecutive writes.

In general, the PCI slave interface functions as a normal memory-mapped PCI target, responding to Memory Read, Memory Write, Configuration Read, and Configuration Write commands from any initiator on the PCI bus. Note that the slave interface will respond only 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.

## 10.7 Byte Swapping of Control Structures

Two control bits in the PCI configuration space are used to configure byte swapping, in order to align with various big and little endian host system requirements.

The SLAVE\_SWAP control bit is bit 29 of address offset 0x40 in the PCI Configuration register. When SLAVE\_SWAP is set to a logic high, the slave interface swaps the bytes of a slave write or read access. The default setting for this bit is logic low.

The MSTR\_CTRL\_SWAP control bit is bit 30 of address offset 0x40 in the PCI Configuration register. When MSTR\_CTRL\_SWAP is set to a logic high, the control structures that the SAR writes are written with bytes swapped. The default setting for this bit is logic low.

The HRST\* pin made active will cause both of these bits to be logic low.

## **10.8 Power Management**

Power Management, as a defined class of functions, consists of mechanisms in software and hardware to minimize system power consumption, manage system thermal limits, and maximize system battery life. The RS8235 supports Power Management on the PCI bus according to the PCI Bus Power Management Interface Specification, Revision 1.0.

Power management states are defined as varying, distinct levels of power savings. The RS8235 device supports the two mandatory power states, D0 and D3, of the PCI Bus Power Management Interface Specification. D0 is the maximum powered state (on) and D3 is the minimum powered state (off). When in the D3 state, SYSCLK is turned off.

Refer to PCI Bus Interface Registers on page 12-25, for detailed information on the PCI Configuration space and PCI registers concerned with implementing the Power Management functions.

Power Management is enabled by default. This capability can be disabled via bit 2 of the EEPROM (Disable Capability Register), which sets bit 20 of the PCI Status Register to a logic low. See the PCI Bus Power Management Interface Specification, Revision 1.0, for specific information on the functions involved in Power Management.

## 10.9 Interface Module to Serial EEPROM

The Interface Module implements the protocol to allow the PCI core to connect to a serial EEPROM.

### 10.9.1 EEPROM Format

The first 32 bytes of the 128-byte EEPROM are used to store PCI configuration information, loaded into the PCI Configuration space at reset. Unless otherwise specified, all unused bytes are reserved and should be programmed to 0x00. Bytes above address offset 0x20 can be used by application software or device drivers as needed. The EEPROM fields are described in Table 10-1.

Table 10-1. EEPROM Fields

Address Offset	Name	Description
0x00	FIELD_ENABLES	Bit 5: Load Memory Size Mask from EEPROM. Bit 4: Load Latency Timer from EEPROM. Bit 3: Load General Enables from EEPROM. Bit 2: Disable Capability Registers (for Power Management). Bit 1: Load Subsystem ID (SID) from EEPROM. Bit 0: Load Subsystem Vendor ID (SVID) from EEPROM.
0x01–0x03	Reserved	Set to zeros.
0x04–0x05	SVID	Subsystem Vendor ID.
0x06–0x07	SID	Subsystem ID.
0x08	General Enables	Bit 4: Special Status Register Bit 29 (SLAVE_SWAP, Slave Control Byte Swap). Bit 3: Special Status Register Bit 30 (MSTR_CTRL_SWAP, Master Control Byte Swap). Bit 2: PCI Command Register Bit 6 (PE_EN, Enable Detection of Parity Errors). Bit 1: PCI Command Register Bit 2 (M_EN, Master Enable). <sup>(1)</sup> Bit 0: PCI Command Register Bit 1 (MS_EN, Memory Space Enable). <sup>(1)</sup>
0x09	Latency Timer	Master Latency Timer
0x0a	Memory Size Mask	Valid Mask Values: Bit 7 6 5 4 3 2 1 0 = Size x x x 0 0 0 0 0 = 2 M x x x 1 0 0 0 0 = 1 M x x x 1 1 0 0 0 = 512 K x x x 1 1 1 0 0 = 256 K x x x 1 1 1 1 0 = 128 K x x x 1 1 1 1 1 = 64 K
Note: (1) System BIOS is typically responsible for setting these bits after programming the PCI Base address register.		

## 10.9.2 Loading the EEPROM Data at Reset

At reset, the PCI Configuration block first reads byte 0x00 of the EEPROM to determine which fields of the EEPROM should be read. It first looks at the FIELD\_ENABLES bits. If bit 2 is set, it sets bit 20 in the PCI Status Register to zero, to disable Power Management capabilities. It then looks at bits 1 and 0 to see if the corresponding SVID and/or SID fields are to be loaded into the corresponding PCI Configuration register fields. If either of these is set to zero, the SUBSYSTEM\_ID and/or SUBSYSTEM\_VENDOR\_ID fields will default to all zeros.

## 10.9.3 Accessing the EEPROM

The EEPROM is accessed through the PCI Configuration space at offset 0x4C, the EEPROM Register. See [Table 12.7.1.5 on page 12-30](#) for a description of the EEPROM Register's contents.

Before starting an EEPROM operation, the BUSY bit must be a logic zero (not busy). When in this state, the EEPROM can be either written to or read from. After initiating a read or write operation, the BUSY bit will be a logic one (busy) until the transfer completes. During this time, the application software must poll the BUSY bit to determine when the transfer has completed. Once completed, the NO\_ACK bit will indicate the status of the operation. A logic one (no acknowledge) indicates that no device responded to the request.

To initiate a write operation, the application software must write the BYTE\_ADDR and DATA fields and set the READ\_WRITE bit to zero. The module will then transfer the data in bits 7:0 (the DATA field) to the device on the bus at the hardware address at the BYTE\_ADDR specified. Application software should then poll the register until the BUSY bit is read as zero (not busy), which indicates that the transfer has completed. Software must then check the NO\_ACK bit to ensure that the transaction completed normally. If not, the software should retry the transaction or signal the error to the user. Since the EEPROM can not respond until after a few milliseconds after a write transaction, it is recommended that all operations resulting in NO\_ACK=1 be retried several times before issuing the failure.

For read operations the application software must also write to the EEPROM Register, specifying the BYTE\_ADDR to be read and setting the READ\_WRITE bit to one. The software must then poll the BUSY bit until the operation completes. At this point the data is returned in the DATA field of the EEPROM register. The software should check the NO\_ACK bit to ensure proper completion of the transfer with no error.

### 10.9.4 Using the Subsystem ID Without an EEPROM

For applications that can use BIOS or boot code to initialize devices before loading high-level operating system software, the RS8235 allows for the programming of the PCI Configuration space fields, SUBSYSTEM\_ID and SUBSYSTEM\_VENDOR\_ID. This feature allows a user to employ the RS8235 without an EEPROM, but still allowing for unique Subsystem IDs.

To program the SUBSYSTEM\_ID and SUBSYSTEM\_VENDOR\_ID fields, BIOS must first write a logic one to bit 31 of the PCI Special Status Register. This enables the writing to these two fields in the PCI Configuration space. BIOS can then update the IDs by writing the desired values to the PCI Configuration space at offset 0x2C. Once the values are written, BIOS should then disable the writing to these fields by setting bit 31 of the PCI Special Status Register to logic zero. When bit 31 is set to zero, writes to these two fields are ignored.

Tie the SCL and SDA pins low when not using an EEPROM.

## 10.10 PCI Host Address Map

The address map of the RS8235 resources seen by the PCI bus is the same as that seen by the host processor. The base address of the RS8235 resource mapping is defined in the BASE\_ADDRESS\_REGISTER\_0 field, located in the PCI configuration space.

Burst reads of the Control and Status (CSR) registers will only return valid data for the first address. Subsequent data words read during a burst read will be indeterminate.

# 11.0 ATM UTOPIA Interface

---

## 11.1 Overview of ATM UTOPIA Interface

The ATM UTOPIA interface contains receive and transmit framer interface logic and receive error generation logic. The ATM UTOPIA interface block also interfaces with the segmentation and reassembly coprocessors.

The ATM UTOPIA interface for the RS8235 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 UTOPIA 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 UTOPIA interface accommodates the Mindspeed RS825x framer device, a UTOPIA-compatible framer or a Mindspeed-conceived slave UTOPIA interface, and is responsible for converting between these devices and the internal data interfaces. The Slave UTOPIA interface connects the RS8235 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 prior to transferring it to the framer interface.
- Generates and checks odd parity on the octet transmit and receive data buses.

## 11.2 Typical Interface Connection to the RS8250 PHY

The interface pins and descriptions are given in Table 11-1. Figure 11-1 shows the signal interface between the RS8235 and the RS8250 ATM receiver/transmitter device. The PCS\*, PAS\*, and PWNR pins are outputs providing chip select, address strobe, and write/read control to the RS8250. PDAEN\* is an input connected to the interrupt sources of the RS8250. 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 11-2. Physical interface devices other than the RS8250 can be connected by using PWAIT\* to extend the read or write cycle, and by using external logic to translate the RS8235 control signals.

Table 11-1. Standalone Interface Pins

Signal	Dir <sup>(1)</sup>	Description
PROCMODE	I	Processor interface mode select input. This bit MUST be set to a logic high.
PCS*	O	Chip select output for PHY device number one. Synchronous to SYSCLK.
PAS*	O	PHY address strobe. Synchronous to SYSCLK.
PWNR	O	PHY write/read select. A logic one on this output indicates a write cycle, a logic zero 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 <sup>(2)</sup> .
PADDR[1:0]	I	Not used, pull to logic zero.
PBSEL[1:0]	I	Not used, pull to logic zero.
PBE[3:0]*	I	Not used, pull to logic zero.
<b>NOTE(S):</b> (1) Direction given with respect to the RS8235. (2) See the HOST_ISTAT0 Register for details.		

Figure 11-1. RS8250 and SAR (RS8235) Interface

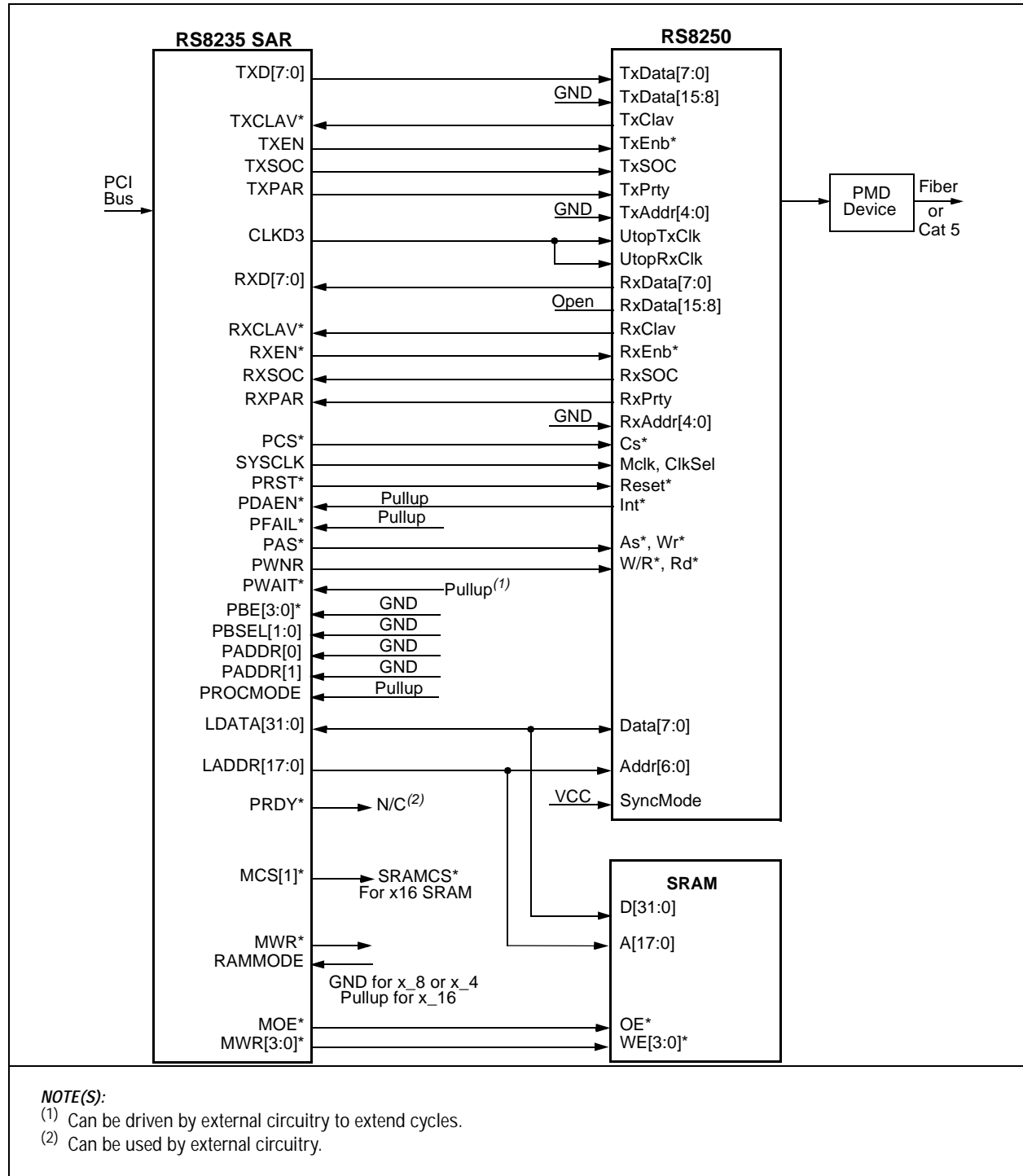


Figure 11-2. RS8235/PHY Functional Timing with Inserted Wait States

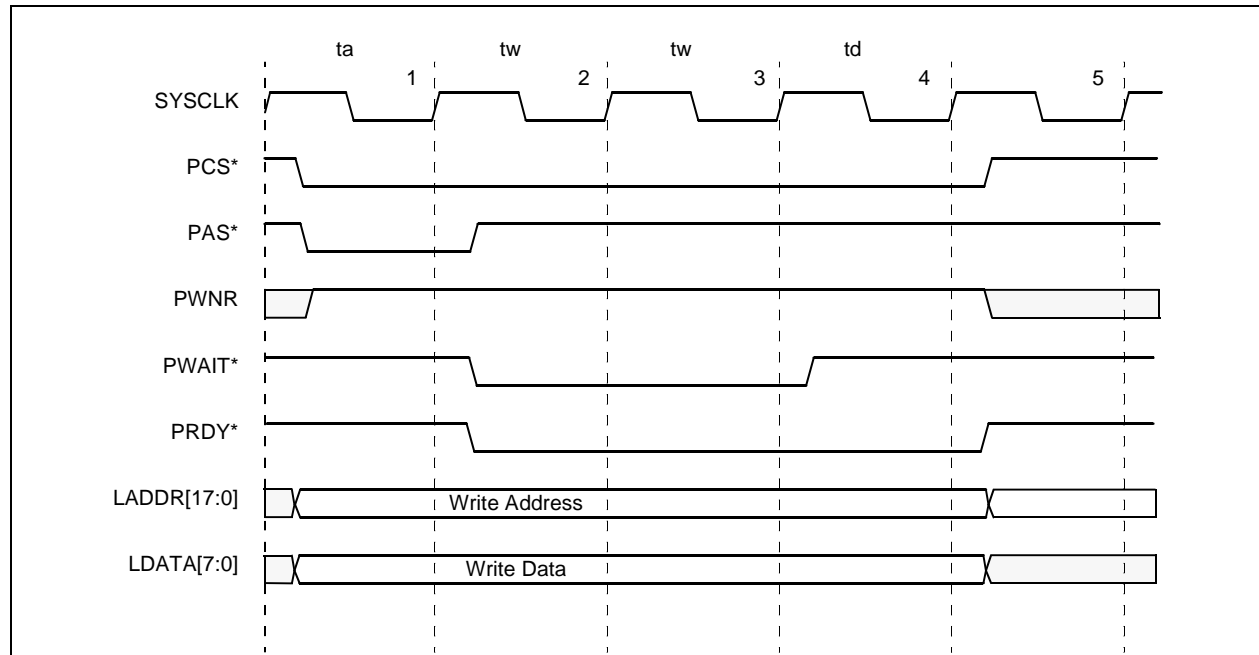
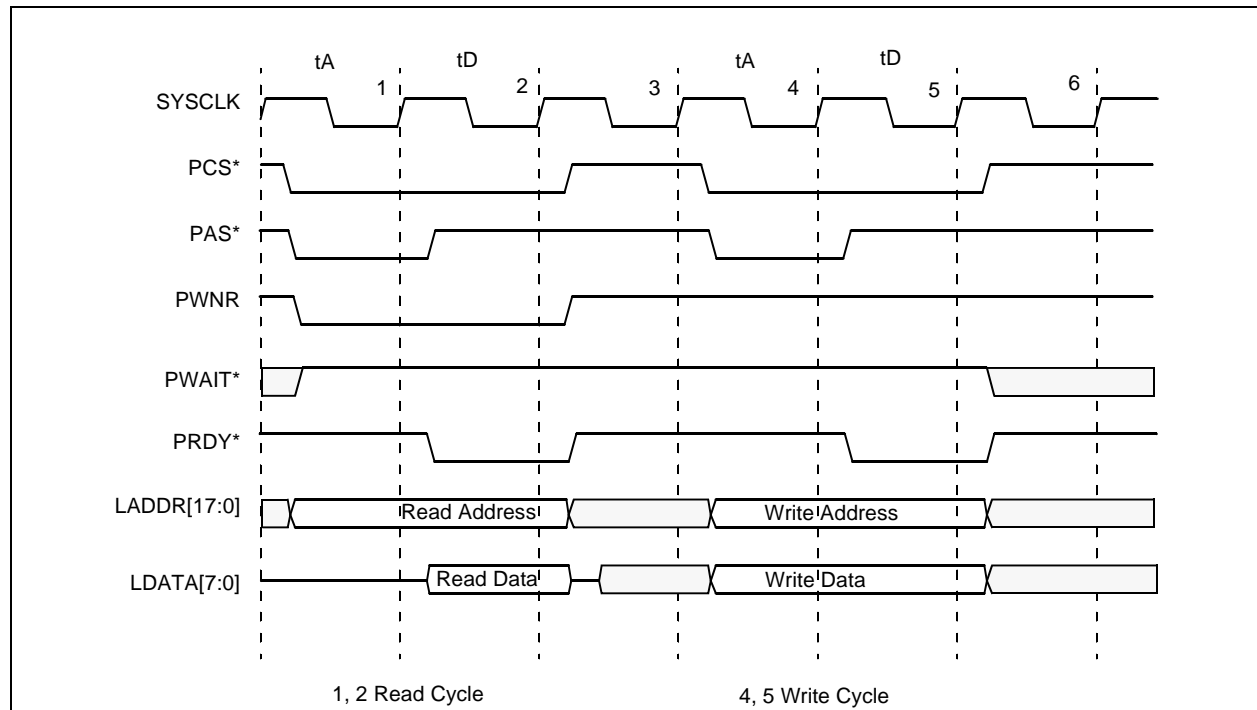


Figure 11-3 shows a read and write RS8250 access. At cycle one (the rising edge of SYSCLK) the RS8250 samples PCS\*, PAS\*, and PWNR low, indicating a read cycle. By the next rising edge of SYSCLK at cycle two, the data is output by the RS8250 to be latched by the RS8235. The same procedure occurs for a write except that at cycle four, PWNR is sampled high. The RS8250 then latches the data to the appropriate internal register on the next SYSCLK rising edge at cycle five.

Figure 11-3. RS8235/RS8250 Read/Write Functional Timing



## 11.3 ATM Physical Interface Logic

The RS8235 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 or slave UTOPIA interface protocol, and the internal byte streams. Configuration pins (FRCFG[1:0]) determine whether the UTOPIA or slave UTOPIA protocol will be used.

The operational mode desired is indicated to the RS8235 by appropriately driving the FRCFG[1:0] input according to [Table 11-2](#).

*Table 11-2. ATM Physical Interface Mode Select*

FRCFG[1:0]	ATM Physical Interface Mode
0 0	Reserved - Do Not Use
0 1	UTOPIA
1 0	Slave UTOPIA
1 1	Reserved - Do Not Use

## 11.4 ATM Physical I/O Pins

The interpretation of the ATM physical interface pins on the RS8235, and the actual signals generated or received by the framer in UTOPIA mode are shown in [Table 11-3](#). Note that both the TxClk and RxClk signals of the UTOPIA interface can be derived from the CLKD3 output of the RS8235 (which must also be connected to the RXCLK/TXCLK input).

*Table 11-3. UTOPIA Mode Signals*

RS8235 Signal	PHY Signal	Active Polarity	RS8235 Direction
TXD[7:0]	TxData[7:0]	—	Out
TXPAR	TxPrty	—	Out
TXSOC	TxSOC	High	Out
TXEN*	TxEnb*	Low	Out
TXCLAV*	TxFull*	Low	In
RXD[7:0]	RxData[7:0]	—	In
RXPAR	RxPrty	—	In
RXSOC	RxSOC	High	In
RXEN*	RxEnb*	Low	Out
RXCLAV*	RxEmpty*	Low	In
RXCLK/TXCLK	RxCik/TxCik	Rising Edge	In

The interpretation of the ATM physical interface pins on the RS8235 and the actual signals generated or received by the framer in slave UTOPIA mode is shown in [Table 11-4](#).

**Table 11-4. Slave UTOPIA Mode Interface Signals**

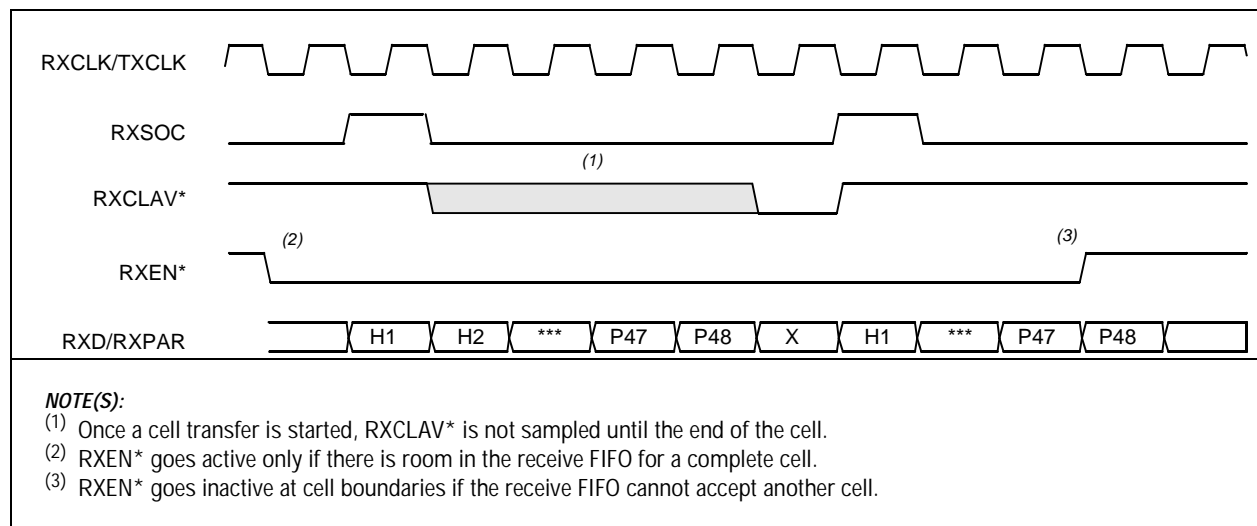
RS8235 Signal	PHY Signal	Active Polarity	RS8235 Direction
TXD[7:0]	TxData[7:0]	—	Out
TXPAR	TxPrty	—	Out
TXSOC	TxSOC	High	Out
TXEN*	TxEnb*	Low	In
TXCLAV*	TxEmp*	Low	Out
RXD[7:0]	RxData[7:0]	—	In
RXPAR	RxPrty	—	In
RXSOC	RxSOC	High	In
RXEN*	RxEnb*	Low	In
RXCLAV*	RxFull*	Low	Out
RXCLK/TXCLK	RxCIk/TxCIk	Rising Edge	In

## 11.5 UTOPIA Mode Cell Handshake Timing

If the UTOPIA\_MODE bit in the CONFIG0 Register is set to a logic high, the device implements cell-level handshaking. Received data is latched from the RXD[7:0] and RXPAR lines on the rising edge of RXCLK/TXCLK, after RXEN\* is sampled active (see Figure 11-4). 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 RXSOC signals to the RS8235 the start of cell. The RXCLAV\* input is the physical layer FIFO empty signal. When it is active, a complete cell is not present in the physical receive FIFO. The PHY device sets RXCLAV\* inactive when it has a complete cell to transfer. The RS8235 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 RS8235 samples the RXCLAV\* input. If RXCLAV\* is low, the PHY device does not have a cell to transfer; but if high, the PHY device does have another cell to transfer and the RS8235 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.

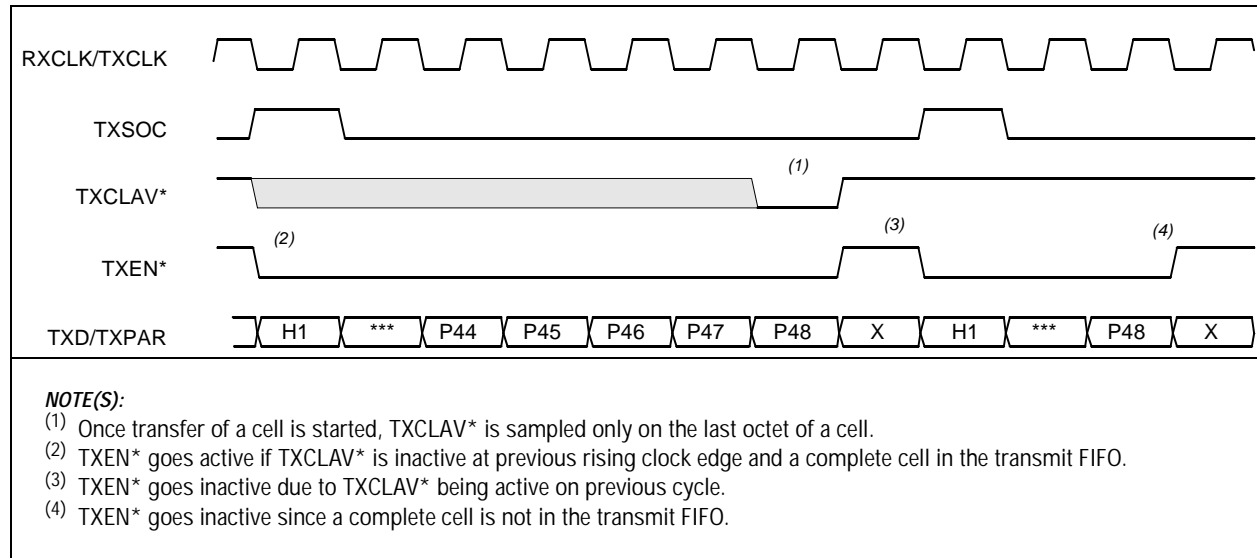
Figure 11-4. Receive Timing in UTOPIA Mode with Cell Handshake



Transmit data is driven on TXD[7:0] on the rising edge of RXCLK/TXCLK when TXEN\* is asserted. TXEN\* is only asserted when there is data in the RS8235 transmit FIFO. Simultaneously, the 8-bit odd parity computed over the TXD[7:0] lines is driven on to the TXPAR output. The TXSOC line is driven by the framer device to indicate start of cell. If the TXCLAV\* input is asserted by the framer device, the framer device is full and another cell is not transmitted to the physical framer. (See Figure 11-5.)

In UTOPIA mode, the RXCLK/TXCLK input can be connected to the RS8235 CLKD3 output; a 50% duty cycle clock derived by dividing CLK2X by three.

Figure 11-5. Transmit Timing in UTOPIA Mode with Cell Handshake

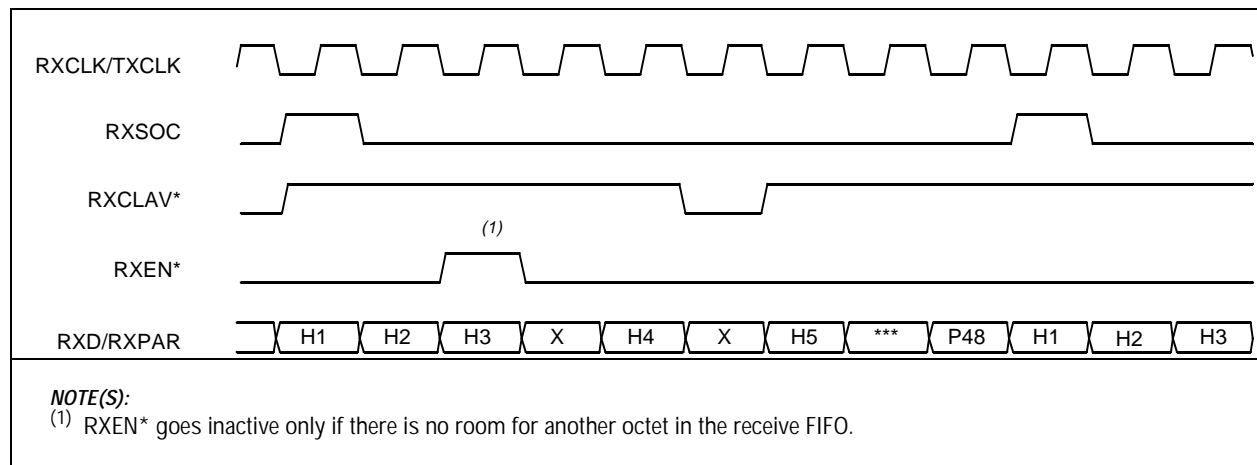


## 11.6 UTOPIA Mode Octet Handshake Timing

If the UTOPIA\_MODE bit in the CONFIG0 Register is set to a logic low, the device implements octet-level handshaking. Received data is latched from the RXD[7:0] and RXPAR lines on the rising edge of RXCLK/TXCLK after RXEN\* is sampled active (see Figure 11-4). 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 RXSOC signals the start of cell to the RS8235. The RXCLAV\* 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 RXCLAV\* inactive when it has an octet to transfer. The RS8235 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.

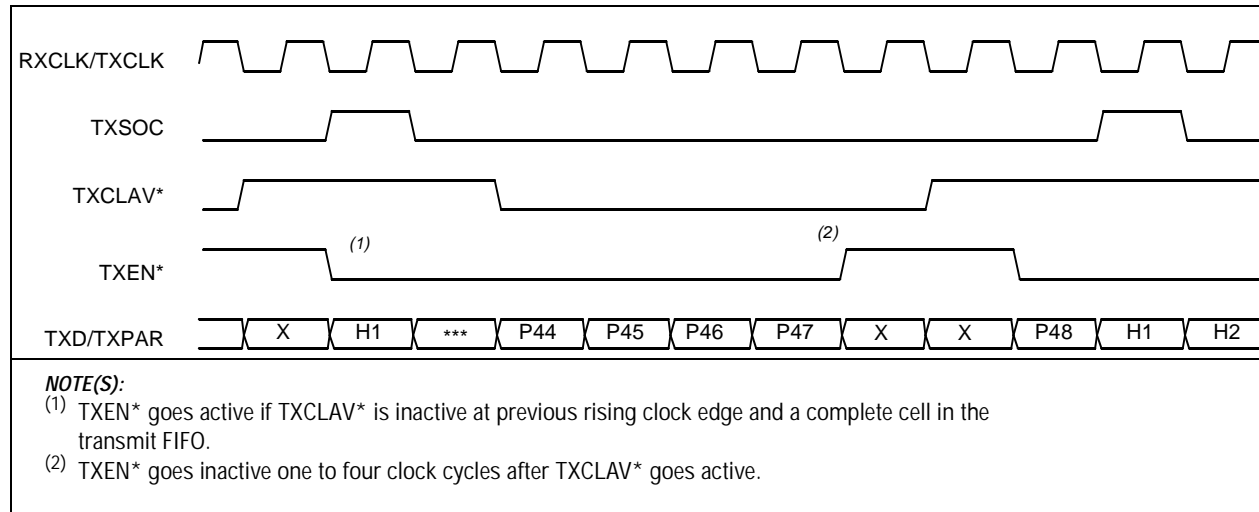
Figure 11-6. Receive Timing in UTOPIA Mode with Octet Handshake



Transmit data is driven on TXD[7:0] on the rising edge of RXCLK/TXCLK when TXEN\* is asserted. TXEN\* is asserted only when there is data in the RS8235 transmit FIFO. Simultaneously, the 8-bit odd parity computed over the TXD[7:0] lines is driven on to the TXPAR output. The TXSOC line is driven by the framer device to indicate start of cell. If the TXCLAV\* input is asserted by the framer device, the framer device is full and can accept only one to four more octets. (See Figure 11-5.)

In UTOPIA mode, the RXCLK/TXCLK input can be connected to the RS8235 CLKD3 output, which is a 50% duty cycle clock derived by dividing the CLK2X input by three.

Figure 11-7. Transmit Timing in UTOPIA Mode with Octet Handshake

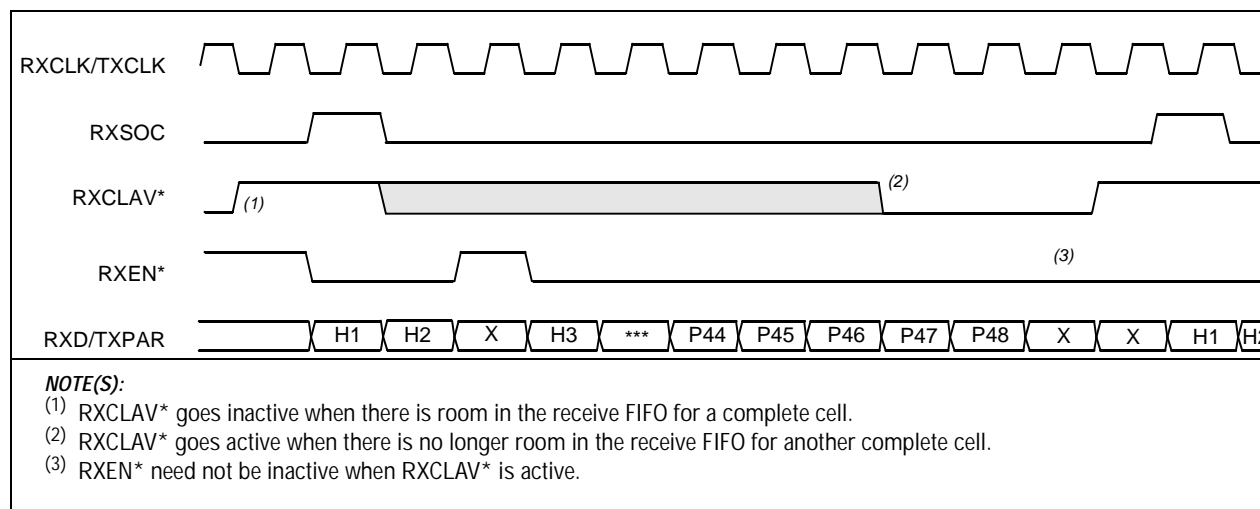


## 11.7 Slave UTOPIA Mode

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 RS8235. The TxFull\* signal is changed to the TxEmpty\* signal and is an output of the RS8235. The RxEmpty\* signal is changed to the RxFull\* signal, and is also an output of the RS8235. This mode supports only a cell-level handshake protocol.

Received data is latched from the RXD[7:0] and RXPAR lines on the rising edge of RXCLK/TXCLK when RXEN\* is active (see Figure 11-4). The 8-bit odd parity computed over the RXD[7:0] lines is compared to the RXPAR input. If there is a parity 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 RXSOC signals to the RS8235 the start of cell. The RXCLAV\* output is the receive FIFO full signal. When it is active, the RS8235 cannot accept another cell. The RS8235 sets RXCLAV\* 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 11-8. Receive Timing in Slave UTOPIA Mode





## 11.8 Loopback Mode

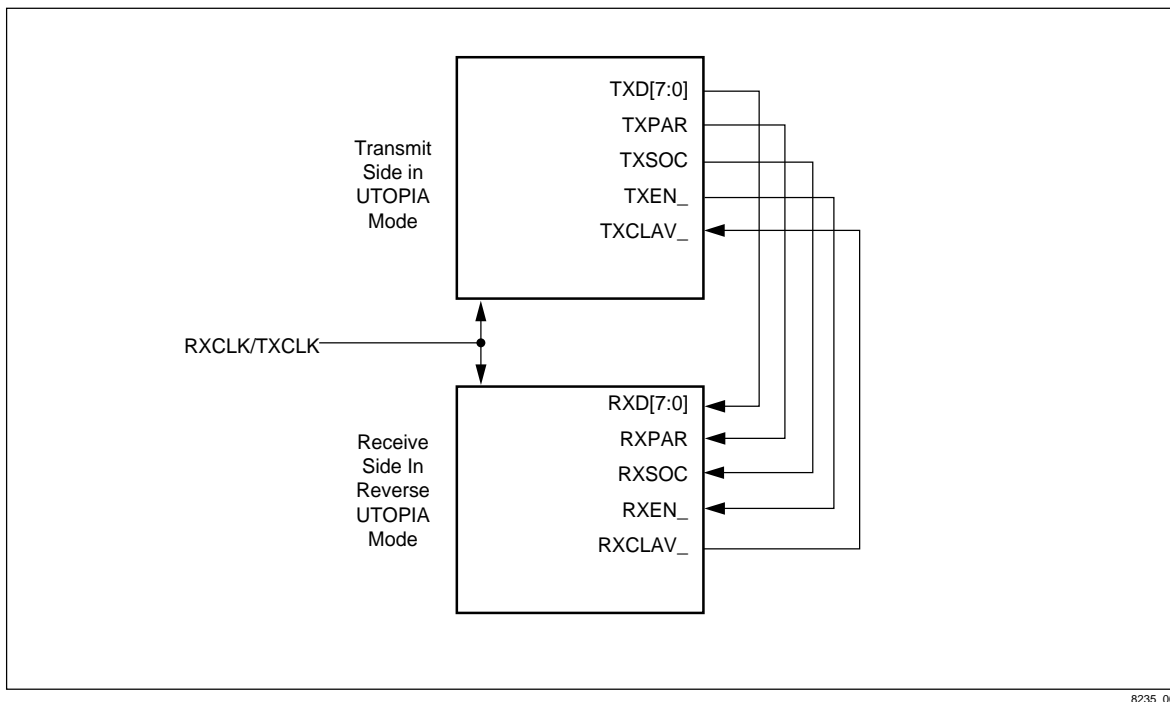
The physical interface can be internally looped by setting the FR\_LOOP bit in the CONFIG0 Register to a logic high. This mode uses the internal system clock for operation; therefore, a framer clock is not needed during loopback operation.

When the signal “fr\_src\_loop” equals one, the interface will loop back the data and control signals so that 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 11-10](#) illustrates. 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, txsoc, txen\_, and rxclav\_ are three-stated, so there are no output conflicts with other devices connected to these signals.

**NOTE:** A reset of the reassembly coprocessor is required after the changing of the FR\_LOOP bit, because this changes the source of the clock to the physical interface circuitry. To accomplish this reset, assert RSM\_CTRL0 (RSM\_RESET).

Figure 11-10. Source Loopback Mode Diagram



## 11.9 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.
- Formats consecutive 4-byte segments into 32-bit words; thus, the header forms the first word, the first four bytes of the payload form the next word, and so on. A total of thirteen 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.

## 11.10 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 four bytes (the ATM header) have been transferred.
- Generates appropriate cell delineation pulses to the transmit ATM physical interface logic, for use in generating the TXSOC\* output, and also in verifying synchronization with the framer device.
- If the ATM physical transmit interface runs out of cells to transmit, the device sets the SEG\_UNFL bit 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.



# 12.0 RS8235 Registers

---

## 12.1 CSR Registers

Control and status registers of the RS8235 are listed in [Table 12-1](#). Detailed register descriptions are provided in this section. Note that byte enables are ignored by the RS8235 when writing to control and status registers.

*Table 12-1. RS8235 Control and Status Registers (1 of 2)*

Address	Name	Type	Description
0x00	CLOCK	R/W	Real Time Clock Register
0x04	ALARM1	R/W	Alarm Register 1
0x08	Reserved	—	Not Implemented
0x0c	SYS_STAT	R/O	System Status Register
0x10	Reserved	—	Not Implemented
0x14	CONFIG0	R/W	Configuration Register 0
0x18	Reserved	—	Not Implemented
0x1c	INT_DELAY	R/W	Interrupt Delay Register
0x20–0x7c	Reserved	—	Not implemented
0x80	SEG_CTRL	R/W	Segmentation Control Register
0x84	SEG_VBASE	R/W	Seg VCC Table & Schedule Table Base Address Register
0x88	SEG_PMBASE	R/W	Segmentation PM Base Register
0x8c	SEG_TXBASE	R/W	Segmentation Transmit Queue Base Register
0x90–0x9c	Reserved	—	Not Implemented
0xa0	SCH_PRI	R/W	Schedule Priority Register
0xa4	Reserved	—	Not Implemented
0xa8	SCH_SIZE	R/W	Schedule Size Register
0xac	SCH_CTRL	R/W	Scheduler Control Register
0xb0	SCH_ABR_MAX	R/W	Maximum ABR VCC_INDEX Register

Table 12-1. RS8235 Control and Status Registers (2 of 2)

Address	Name	Type	Description
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 Register
0xc0	PCR_QUE_INT01	R/W	PCR Queue Interval 0 and 1 Register
0xc4	PCR_QUE_INT23	R/W	PCR Queue Interval 2 and 3 Register
0xc8–0xec	Reserved	—	Not Implemented
0xf0	RSM_CTRL0	R/W	Reassembly Control Register 0
0xf4	RSM_CTRL1	R/W	Reassembly Control Register 1
0xf8	RSM_FOBASE	R/W	Reassembly Free Buffer Queue Base Register
0xfc	RSM_FOCTRL	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 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
0x170–0x1a0	Reserved	—	Not Implemented
0x1a4	HOST_ST_WR	R/O	Host Status Write Register
0x1a8–0x1bc	Reserved	—	Not Implemented
0x1c0	HOST_ISTAT0	R/O	Host Interrupt Status Register 0
0x1c4	HOST_ISTAT1	R/O	Host Interrupt Status Register 1
0x1c8–0x1cc	Reserved	—	Not Implemented
0x1d0	HOST_IMASK0	R/W	Host Interrupt Mask Register 0
0x1d4	HOST_IMASK1	R/W	Host Interrupt Mask Register 1
0x1d8–0x1fc	Reserved	—	Not Implemented

### 12.1.1 Register Terminology

The access type terminology given below applies to all registers in this section. Each register description is prefaced with one of these appropriate abbreviated access types:

- R/W — Read and write access
- R/O — Read only access

## 12.2 System Registers

### 12.2.1 0x00 - Real-Time Clock Register (CLOCK)

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 can be written to any value by the host processor, and can allow the SAR to propagate an interrupt when the value in the CLOCK register is greater than the user-defined value in the ALARM1 register. By this means, status queue interrupt signals to the host can be delayed.

Bit	Field Size	Name	Description
31-0	32	CLOCK[31:0]	Real-time clock value.

### 12.2.2 0x04 - Alarm Register 1 (ALARM1)

This register contains a 32-bit value which is compared against the CLOCK register. The host can populate the ALARM1 register with any value. When the CLOCK register value is greater than the ALARM1 register value, the SAR is allowed to propagate a status queue interrupt to the host. By this means, status queue interrupt signals to the host can be delayed or held off by the value set in the ALARM1 register.

Bit	Field Size	Name	Description
31-0	32	ALARM1[31:0]	ALARM1 comparison value.

### 12.2.3 0x0c - System Status Register (SYS\_STAT)

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 might not be visible to the processors. It also contains expanded information for the status located in the HOST\_ISTAT0 register.

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. Must be tied high; therefore, the state of this pin must be 1.
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 RS8235; Rev A = 0; Rev B = 1.
3-0	4	DEVICE[3:0]	Device ID for the RS8235; Rev A = 0; Rev B = 1.

### 12.2.4 0x14 - Configuration Register 0 (CONFIG0)

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 can 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	Enables 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 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 (BT8222/3)	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.
14	1	FR_LOOP	When set, this bit enables loopback of cells at the ATM physical interface. Loopback uses SYSCLK.
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

Bit	Field Size	Name	Description
11	1	LP_BWAIT	MUST be set to logic low.
10	1	MEMCTRL	Selects zero or one wait states SAR shared memory (1 or 2 cycle).
9-7	3	BANKSIZE[2:0]	Selects size of memory banks for contiguous memory support. See <a href="#">Section 9.2</a> 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.

### 12.2.5 0x1c - Interrupt Delay Register (INT\_DELAY)

Bit	Field Size	Name	Description
31-11	21	Reserved	Set to zero.
10	1	Reserved	Set to zero.
9	1	EN_TIMER	Enable status queue interrupt timer delay.
8	1	EN_STAT_CNT	Enable status queue interrupt counter delay.
7-0	8	STAT_CNT[7:0]	Number of status queue entries written before allowing interrupt to propagate to output pin.

## 12.3 Segmentation Registers

### 12.3.1 0x80 - Segmentation Control Register (SEG\_CTRL)

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. ( $VBR\_OFFSET + (\# VBR/ABR \text{ priorities}) \leq 7.$ )
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	1	ADV_ABR_TMPLT	Advanced ABR template mode. When logic high, per-connection MCR and ICR enabled. When logic low, per-template MCR and ICR enabled.
22–16	7	Reserved	Program and read as zero.
15–12	4	TX_FIFO_LEN	Depth of transmit FIFO in cells. Valid range is 3–9. To ensure optimum performance, the depth of the FIFO should be at least three.
11	1	CLPO_EOM	Set CLP in ATM header to zero on last cell of CPCS-PDU.
10–8	3	Reserved	Program and read as zero.
7–6	2	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 packet.
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.

Bit	Field Size	Name	Description
2	1	TX_RND	Set for transmit queue servicing in round robin order. 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 = 1,024 11 = 4,096

### 12.3.2 0x84 - Seg VCC Table & Schedule Table Base Address Register (SEG\_VBASE)

The SEG\_VBASE register sets the base address in SAR 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.

### 12.3.3 0x88 - Segmentation PM Base Register (SEG\_PMBASE)

The SEG\_PMBASE register sets the base address in SAR-shared memory for the VBR bucket table. The base address is 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. (See Section 6.2.4.4, Dual Leaky Bucket, for details on loading bucket table entries.)
15-0	16	Reserved	Set to zero.

### 12.3.4 0x8c - Segmentation Transmit Queue Base Register (SEG\_TXBASE)

The SEG\_TXBASE register sets the base address in SAR 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 zero.
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.

## 12.4 Scheduler Registers

### 12.4.1 0xa0 - Schedule Priority Register (SCH\_PRI)

This register specifies the use of each global priority pointer, for priority queues 0 through 7.

Bit	Field Size	Name	Description
31	1	QPCR_ENA7	Enable PCR limits on global priority pointer 7.
30	1	QPCR_ENA6	Enable PCR limits on global priority pointer 6.
29	1	QPCR_ENA5	Enable PCR limits on global priority pointer 5.
28	1	QPCR_ENA4	Enable PCR limits on global priority pointer 4.
27	1	QPCR_ENA3	Enable PCR limits on global priority pointer 3.
26	1	QPCR_ENA2	Enable PCR limits on global priority pointer 2.
25	1	QPCR_ENA1	Enable PCR limits on global priority pointer 1.
24	1	QPCR_ENA0	Enable PCR limits on global priority pointer 0.
23	1	Reserved	Program and read as zero.
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 zero.
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 zero.
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 zero.
13	1	TUN_ENA4	Enable tunnel on global priority pointer 4.

Bit	Field Size	Name	Description
12	1	GFC4	Enable GFC on priority pointer 4.
11	1	Reserved	Program and read as zero.
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 zero.
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 zero.
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 zero.
1	1	TUN_ENA0	Enable tunnel on global priority pointer 0.
0	1	GFC0	Enable GFC on priority pointer 0.

### 12.4.2 0xa8 - Schedule Size Register (SCH\_SIZE)

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 zero.
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.

### 12.4.3 0xac - Scheduler Control Register (SCH\_CTRL)

The SCH\_CTRL register defines the configured schedule slot and priority and VBR offsets, when 16 priority queues are used.

Bit	Field Size	Name	Description
31-28	4	Reserved	Program and read as 0.
27	1	Reserved	Program and read as 0.

Bit	Field Size	Name	Description
26	1	Reserved	Program and read as 0.  Note: Must be set to zero during SAR initialization and remain zero during SAR operation.
25	1	Reserved	Program and read as 0.
24	1	NCR_EN_DEST	Global enable for destination ACR notification.
23	1	NCR_EN_SRC	Global enable for source ACR notification.
22#18	5	NCR_STAT_ID	Identifies the status queue to be used for both source and destination ACR/ER notification when EN_NCR_STAT is asserted.
17	1	EN_NCR_STAT	Enable global status queue for both source and destination ACR/ER notification.
16#15	2	Reserved	Program and read as 0.
14#12	3	Reserved	Program and read as 0.
11#10	2	Reserved	Program and read as 0.
9#6	4	Reserved	Program and read as 0.
5#4	2	Reserved	Program and read as 0.
3#0	4	Reserved	Program and read as 0.

#### 12.4.4 0xb0 - Maximum ABR VCC\_INDEX Register (SCH\_ABR\_MAX)

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 zero.
15–0	16	VCC_MAX	Maximum ABR VCC index.

#### 12.4.5 Schedule ABR Constant Register (SCH\_ABR\_CON)

access types: R/W

### 12.4.6 0xb4 - Schedule ABR Constant Register (SCH\_ABR\_CON)

The SCH\_ABR\_CON register sets the ABR TRM and ADTF timeout. Timeout values are in units of 64 cell slots.

Bit	Field Size	Name	Description
31–16	16	ABR_TRM	ABR TRM parameter. Parameter value is SYSCLK period x SLOT_PER x ABR_TRM x 64.
15–0	16	ABR_ADTF	ABR ADTF parameter. Parameter value is SYSCLK period x SOT_PER x ABR_ADTF x 64.

### 12.4.7 Schedule ABR Constant Register (SCH\_ABR\_CON)

access types: R/W

### 12.4.8 0xb8 - ABR Decision Table Lookup Base Register (SCH\_ABRBASE)

The SCH\_ABRBASE register sets the base address in SAR 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\_ABRBASE register.

Bit	Field Size	Name	Description
31–29	3	Reserved	Program and read as zero.
28	1	OOR_EN	Enable ABR out-of-rate Forward RM cell generation.
27–16	12	OOR_INT	ABR out-of-rate Forward RM cell interval. A VCC is examined for an out-of-rate cell every OOR_INT schedule slots.
15–0	16	SCH_ABRB[15:0]	Base address for the ABR decision table.

### 12.4.9 Schedule ABR Constant Register (SCH\_ABR\_CON)

access types: R/W

### 12.4.10 0xbc - ABR Congestion Register (SCH\_CNG)

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	FBO_CNG[31:0]	Congestion state for each free buffer queue.

### 12.4.11 Schedule ABR Constant Register (SCH\_ABR\_CON)

access types: R/W

### 12.4.12 0xc0 - PCR Queue Interval 0 and 1 Register (PCR\_QUE\_INT01)

The PCR\_QUE\_INT01 register fields are used to store the two lower Peak Cell Rate (PCR) values used in PCR shaping on those priority queues that have been enabled for PCR shaping by setting the QPCR\_ENAx bits in the SCH\_PRI register. QPCR\_INTx values are used in order: 3, 2, 1, and 0; highest to lowest. PCR is determined by:  $1/(QPCR\_INTx \times SYSCLK\_period \times SLOT\_PER)$ .

Bit	Field Size	Name	Description
31–16	16	QPCR_INT1	The assigned PCR interval, entered as number of schedule table slots, used to map to the 3rd-highest priority queue that is enabled for PCR shaping (using the QPCR_ENAx bits in the SCH_PRI register).
15–0	16	QPCR_INT0	The assigned PCR interval, entered as number of schedule table slots, used to map to the 4th-highest priority queue that is enabled for PCR shaping (using the QPCR_ENAx bits in the SCH_PRI register).

### 12.4.13 0xc4 - PCR Queue Interval 2 and 3 Register (PCR\_QUE\_INT23)

The PCR\_QUE\_INT23 register fields are used to store the two highest Peak Cell Rate (PCR) values used in PCR shaping on those priority queues that have been enabled for PCR shaping by setting the QPCR\_ENAx bits in the SCH\_PRI register. QPCR\_INTx values are used in order: 3, 2, 1, and 0; highest to lowest. PCR is determined by:  $1/(QPCR\_INTx \times SYSCLK\_period \times SLOT\_PER)$ .

Bit	Field Size	Name	Description
31–16	16	QPCR_INT3	The assigned PCR interval, entered as number of schedule table slots, used to map to the highest priority queue that is enabled for PCR shaping (using the QPCR_ENAx bits in the SCH_PRI register).
15–0	16	QPCR_INT2	The assigned PCR interval, entered as number of schedule table slots, used to map to the 2nd-highest priority queue that is enabled for PCR shaping (using the QPCR_ENAx bits in the SCH_PRI register).

## 12.5 Reassembly Registers

### 12.5.1 0xf0 - Reassembly Control Register 0 (RSM\_CTRL0)

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. This includes 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 four 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 4,096.
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.
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_FBO_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/SAR 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.

Bit	Field Size	Name	Description
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 x 1,024 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 can be discarded.

### 12.5.2 0xf4 - Reassembly Control Register 1 (RSM\_CTRL1)

The Reassembly Control Register 1 contains additional general control bits for the reassembly coprocessor.

Bit	Field Size	Name	Description														
31	1	EN_PROG_BLK_SZ	Enable Programmable Block Size. When a logic high, the programmable block size VPI/VCI lookup method is enabled. This method consists of programmable block sizes, and an additional BLK_EN bit in the VCI Index Table.														
30–28	3	VCI_IT_BLK_SZ	VCC Table/VCI Index Table Block Size. This field determines the number of Rsm VCC entries accessed per VCI Index Table entry. <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>VALUE</th> <th>VCC BLOCK_SIZE</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>1</td> </tr> <tr> <td>001</td> <td>2</td> </tr> <tr> <td>010</td> <td>4</td> </tr> <tr> <td>011</td> <td>8</td> </tr> <tr> <td>100</td> <td>16</td> </tr> <tr> <td>101</td> <td>32</td> </tr> </tbody> </table>	VALUE	VCC BLOCK_SIZE	000	1	001	2	010	4	011	8	100	16	101	32
VALUE	VCC BLOCK_SIZE																
000	1																
001	2																
010	4																
011	8																
100	16																
101	32																
27–24	4	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	OAM 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.														
9–7	3	Reserved	Not implemented at this time.														
6–5	2	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.														
4–2	3	Reserved	Not implemented at this time.														
1–0	2	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.														

### 12.5.3 0xf8 - Reassembly Free Buffer Queue Base Register (RSM\_FQBASE)

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 SAR 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.

### 12.5.4 0xfc - Reassembly Free Buffer Queue Control Register (RSM\_FQCTRL)

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 = 1,024 3 = 4,096
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 Queue 0 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 0-2 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 zero in FWD_EN selects only one queue, and a value of 1 selects 2 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 SAR shared memory.

### 12.5.5 0x100 - Reassembly Table Base Register (RSM\_TBASE)

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 SAR 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.

### 12.5.6 0x104 - Reassembly Time-out Register (RSM\_TO)

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. If a value of zero, the interrupt period is 65,538 SYSCLK periods.
15–0	16	RSM_TO_CNT	Reassembly Time-out Counter. The value in this register plus one determines the number of time-out interrupts that occur in each pass through the RSM VCC Table.

### 12.5.7 0x108 - Reassembly/Segmentation Queue Base Register (RS\_QBASE)

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 eight octets in length. 00 = 256 01 = 1,024 10 = 4,096 11 = 16,384
15–0	16	RS_QBASE[15:0]	Base address of the reassembly/segmentation queue.

## 12.6 Counters and Status Registers

### 12.6.1 0x160 - ATM Cells Transmitted Counter (CELL\_XMIT\_CNT)

This register counts the number of user data cells transmitted by the segmentation coprocessor. The counter is reset to zero by an assertion of either the HRST\* system reset pin or the GLOBAL\_RESET bit in the CONFIG0 register. Optionally, an interrupt can be programmed when the counter rolls over.

Bit	Field Size	Name	Description
31-0	32	CELL_XMIT_CNT	Count of user data cells transmitted.

### 12.6.2 0x164 - ATM Cells Received Counter (CELL\_RCVD\_CNT)

This register counts the number of assigned cells received by the reassembly coprocessor. The counter is reset to zero by an assertion of either the HRST\* system reset pin or the GLOBAL\_RESET bit in the CONFIG0 register. Optionally, an interrupt can be programmed when the counter rolls over.

Bit	Field Size	Name	Description
31-0	32	CELL_RCVD_CNT	Count of assigned received cells.

### 12.6.3 0x168 - ATM Cells Discarded Counter (CELL\_DSC\_CNT)

This register counts the number of unassigned cells received by the reassembly coprocessor. The counter is reset to zero by an assertion of either the HRST\* system reset pin or the GLOBAL\_RESET bit in the CONFIG0 register. Optionally, an interrupt can be programmed when the counter rolls over.

Bit	Field Size	Name	Description
31-0	32	CELL_DSC_CNT	Count of unassigned cells dropped.

### 12.6.4 0x16c - AAL5 PDUs Discarded Counter (AAL5\_DSC\_CNT)

This register counts the number of AAL5 CPCS-PDUs discarded due to buffer firewall, buffer underflow, or status overflow. The counter is reset to zero by an assertion of either the HRST\* system reset pin or the GLOBAL\_RESET bit in the CONFIG0 register. Optionally, an interrupt can be programmed when the counter rolls over.

Bit	Field Size	Name	Description
31–16	16	Reserved	Not implemented at this time.
15–0	16	AAL5_DSC_CNT	AAL5 PDUs discarded by the reassembly coprocessor.

### 12.6.5 0x1a4 - Host Status Write Register (HOST\_ST\_WR)

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.

### 12.6.6 Host Interrupt Status Registers

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(S):* Only host reads will reset the status bits in the HOST\_ISTAT0 register.

### 12.6.6.1 0x1c0 - Host Interrupt Status Register 0 (HOST\_ISTAT0)

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 can be connected to a PHY interrupt source.
29	1	-	Reserved	Read as zero.
28	1	-	Reserved	Read as zero.
27	1	-	Reserved	Read as zero.
26	1	-	Reserved	Read as zero.
25-24	2	-	Reserved	Read as zero.
23	1	-	Reserved	Read as zero. 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 zero.
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 RS8235 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 RS8235.
14-12	3	-	Reserved	Read as zero.
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 RS8235 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 RS8235.
8-4	5	-	Reserved	Read as zero.
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.

### 12.6.6.2 0x1c4 - 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 1 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 zero.
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 zero.
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 zero.
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.

**12.6.6.3 0x1d0 - Host Interrupt Mask Register 0 (HOST\_IMASK0)** 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.

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 zero.
28	1	Reserved	Set to zero.
27	1	Reserved	Set to zero.
26	1	Reserved	Set to zero.
25–24	2	Reserved	Set to zero.
23	1	Reserved	Set to zero. 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 zero.
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 zero.
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 zero.
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.

**12.6.6.4 0x1d4 - Host Interrupt Mask Register 1 (HOST\_IMASK1)** 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 zero.
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 zero.
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 zero.
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.

## 12.7 PCI Bus Interface Registers

In accordance with the PCI Bus Specification Revision 2.1, 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 are given in the PCI bus specification, and are not repeated here. The implementation of the configuration space registers in the RS8235 is shown in the table below. Descriptions of fields, and other registers within the PCI configuration space are shown in the tables following.

### 12.7.1 PCI Configuration Register

Byte Addr	PCI Configuration Register Layout																														
	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
0x00	DEVICE_ID (0x8235)														VENDOR_ID (0x14F1)																
0x04	STATUS														COMMAND																
0x08	CLASS_CODE (0x020300)																						REV_ID (0x00)								
0x0c	Reserved				HEADER_TYPE (0x00)				LAT_TIMER				CACHE_LINE_SIZE (0x00)																		
0x10	BASE_ADDRESS_REGISTER_0																														
0x14–0x28	Reserved																														
0x2c	SUBSYSTEM_ID (SID)														SUBSYSTEM_VENDOR_ID (SVID)																
0x30	Reserved																														
0x34	Reserved																						CAPABILITY_PTR (0x50)								
0x38	Reserved																														
0x3c	MAX_LATENCY (0x05)				MIN_GRANT (0x02)				INTERRUPT_PIN (0x01)				INTERRUPT_LINE																		
0x40	SPECIAL_STATUS_REGISTER																														
0x44	MASTER_READ_ADDR																														
0x48	MASTER_WRITE_ADDR																														
0x4c	EEPROM_REGISTER																														
0x50	PM_CAPABILITY														NEXT_CAP_PTR				CAPABILITY_ID (0x01)												
0x54	PM_DATA				Reserved				PMCSR																						

Byte Addr	PCI Configuration Register Layout
0x58–0x7C	Reserved (0x00000000)

### 12.7.1.1 PCI Configuration Register Field Descriptions

Field Name	Description/Function
DEVICE_ID	16-bit device identifier. Serves to uniquely identify the SAR to the host operating system. Set to 0x8235.
VENDOR_ID	16-bit vendor identifier code, allocated on a global basis by the PCI Special Interest Group. Set to 0x14F1.
STATUS	PCI bus interface status register. The PCI host can monitor its operation using the STATUS field. This field is further divided into subfields as shown below. These bits can be reset by writing a logic high to the appropriate bit. See the PCI Status Register Description below for a description of the bits in the register.
COMMAND	PCI bus interface control/command register. The PCI host can 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. See the PCI Command Register Description table below for a description of the bits in the register.
CLASS_CODE	The CLASS_CODE register is read-only and is used to identify the generic function of the device. See PCI Bus Specification Revision 2.1 for the specific allowed settings for this field. Set to 0x020300 (indicates a network controller, specifically an ATM controller).
REV_ID	Revision ID code for the RS8235 chip. Rev A = 0, Rev B = 1.
HEADER_TYPE	This field identifies the layout of the second part of the predefined header of the PCI Configuration space (beginning at 0x10). See PCI Bus Specification Revision 2.1 for the specific possible settings for this field. Set to 0x00.
LAT_TIMER	Latency timer. Value after HRST* active is 0x00. All bits are writable. The suggested value is 0x10 in order to allow the complete transfer of a cell.
CACHE_LINE_SIZE	This read/write register specifies the system cache line size in units of 32-bit words. Must be initialized to 0x00 at initialization and reset.
BASE_ADDRESS_REGISTER_0	Base address of PCI address space occupied by the RS8235 (as seen and assigned by the host processor). Value after HRST* active is 0x00. The upper 9 bits are writable, resulting in an 8 MB address space.
SUBSYSTEM_ID (SID)	This register value is used to uniquely identify the add-in board or subsystem where the PCI device resides. Thus, it provides a mechanism for add-in card vendors to distinguish their cards from one another even though the cards might have the same PCI controller on them (and therefore the same DEVICE_ID).
SUBSYSTEM_VENDOR_ID (SVID)	This register value is used to uniquely identify the vendor of an add-in board or subsystem where the PCI device resides. Thus, it provides a mechanism for add-in card vendors to distinguish their cards from one another even though the cards might have the same PCI controller on them (and therefore the same VENDOR_ID).
CAPABILITY_PTR	This field provides an offset into the PCI Configuration space for the location of the first item in the Capabilities Linked List. Set to 0x50.

Field Name	Description/Function
MAX_LATENCY	This read-only register specifies how often the RS8235 needs to gain access to the PCI bus, assuming a clock rate of 33 MHz. Set to 0x02 (a period of time in units of 1/4 microseconds).
MIN_GRANT	This read-only register specifies how long a burst period the RS8235 needs to gain access to the PCI bus. Set to 0x50 (a period of time in units of 1/4 microseconds).
INTERRUPT_PIN	This read-only register tells which interrupt pin the device (or device function) uses. Set to 0x01 (corresponds to interrupt pin INTA_ on the PCI bus).
INTERRUPT_LINE	Interrupt line identifier. The value in this register tells which input of the system interrupt controller(s) the device's interrupt pin is connected to. The device itself does not use this value; rather, it is used by device drivers and operating systems.
SPECIAL_STATUS_REGISTER	Device status not defined by the PCI specification. The field is further subdivided into subfields as shown in the PCI Special Status Register Description table, below. Detailed descriptions of these 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.
MASTER_READ_ADDR	Current read target address used by PCI bus master (read only).
MASTER_WRITE_ADDR	Current write target address used by PCI bus master (read only).
EEPROM_REGISTER	A 32-bit register controlling access to the Serial EEPROM. See the EEPROM Register Description table below, for a description of the specific fields in the EEPROM_REGISTER.
PM_CAPABILITY	Power Management Capabilities register. A 16-bit read-only register which provides information on the capabilities of the function related to Power Management. See the PCI Bus Power Management Interface Specification Revision 1.0 for specific information related to this register.
NEXT_CAP_PTR	Next Item Pointer register. This field provides an offset into the PCI Configuration space pointing to the location of the next item of the linked capability list. If there are no additional items in the Capabilities List, this register is set to 0x00.
CAPABILITY_ID	Capability Identifier. When set to 0x01, indicates that the linked list item being pointed to is the PCI Power Management registers. Default value is 0x01.
PM_DATA	Power Management Data register. This 8-bit read-only register provides a mechanism for the Power Management function to report state-dependent operating data, such as power consumed or heat dissipation. See the PCI Bus Power Management Interface Specification Revision 1.0 for specific information related to this register.
PMCSR	Power Management Control/Status register. This 16-bit register is used to manage the PCI function's power management state, as well as to enable and monitor power management events. See the PCI Bus Power Management Interface Specification Revision 1.0 for specific information related to this register.

### 12.7.1.2 PCI Command Register Description

Bit	Field Size	Name	Description
15–10	6	Reserved	Set to 0x000.
9	1	FB_EN	Master Fast Back-to-back enable across target.
8	1	SE_EN	SERR* pin output enable.
7	1	Reserved	Set to zero.
6	1	PE_EN <sup>(1)</sup>	Parity Error detection and report enable.
5–3	3	Reserved	Set to zero.
2	1	M_EN <sup>(1)</sup>	Master enable. M_EN must be asserted (i.e., set to one) before the RS8235 can act as master on the PCI bus.
1	1	MS_EN <sup>(1)</sup>	Memory Space enable. MS_EN must be asserted (i.e., set to one) before the RS8235 address space (registers and memory) can be accessed across the PCI interface.
0	1		I/O Space enable. (Not used.) Set to zero.

**NOTE(S):**  
<sup>(1)</sup> Can be loaded from EEPROM.

### 12.7.1.3 PCI Status Register Description

Bit	Field Size	Name	Description
31	1	DPE	Parity Error Detected.
30	1	SSE	Signalled System Error. (Device asserted SERR*.)
29	1	RMA	Received Master Abort. (Device Master aborted transfer.)
28	1	RTA	Received Target Abort. (Detected target abort as master.)
27	1		Target aborted as slave.
26–25	2		Devsel Speed (00 Fast).
24	1	DPR	Reported Data Parity Error. (Parity error detected as master).

Bit	Field Size	Name	Description
23	1		Fast Back-to-back supported as Slave. Set to one.
22	1		UDF Support. Set to zero.
21	1		66 MHz Support. Set to zero.
20	1	NEW_CAP	Capability List Support. This bit indicates whether this function implements a list of extended capabilities defined in PCI Bus Spec Revision 2.1, such as PCI Power Management. When set to one, indicates the presence of New Capabilities. A value of 0 indicates that this function does not implement New Capabilities. Set to one. <sup>(1)</sup>
19–16	4	Reserved	Set to 0x00.
<b>NOTE(S):</b> <sup>(1)</sup> Can be set to zero from EEPROM.			

#### 12.7.1.4 PCI Special Status Register Description

Bit	Field Size	Name	Description
31	1	EN_SID_WR	Enable SVID/SID Write. Default=0.
30	1	MSTR_CTRL_SWAP <sup>(1)</sup>	Enable byte swapping on control words that the SAR writes. Default = low.
29	1	SLAVE_SWAP <sup>(1)</sup>	Enable byte swapping on control words of a slave write or read access. Default = low.
28–23	6	Reserved	Set to 0b000000.
22–16	7	Memory Size Mask <sup>(1)</sup>	A 7-bit mask which sets one of a range of values for the size of the PCI address space. Default = 0000000. 0000000 = 8 M
15–12	4	Reserved	Set to 0x00.

Bit	Field Size	Name	Description
11	1	INTF_DIS	Master Interface Disabled. If the M_EN bit in the COMMAND field is a logic low, any attempt by the RS8235 to perform a DMA transaction to the PCI bus will result in an error. INTF_DIS and MERROR bits will both be set to a logic high. This bit can be reset by writing a logic high to itself.
10	1	INT_FAIL	Master Interface Failed. Set to a logic high when an internal PCI/DMA synchronization error has occurred. The MERROR bit will also be set to a logic high. This bit can be reset by writing a logic high to itself.
9	1	MERROR	Memory Error. 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 can be reset by writing a logic high to itself.
8	1	MRD	Error on Master Read/Write. If a logic high, indicates that the errored transaction was a read and the address of the read is located in the MASTER_READ_ADDR field. If a logic low, indicates the errored transaction was a write and the corresponding address is located in the MASTER_WRITE_ADDR field.
7-0	8	Reserved	Set to 0x00.
<b>NOTE(S):</b> (1) Can be loaded from EEPROM.			

#### 12.7.1.5 EEPROM Register Description

Bit	Field Size	Name	Description
31	1	BUSY	Indicates that an EEPROM operation is currently in progress. This bit must be read as a zero before initiating an EEPROM transfer.
30	1	NO_ACK	When set to a one, indicates that the previous transfer failed (i.e., no hardware address response).
29-24	6	Reserved	Reserved for future use.
23-17	7	HARDWARE_ADDR	The 7-bit hardware address for a transfer that indicates the target of the transfer. The EEPROM has the hardware address of b1010000.
16	1	READ_WRITE	Indicates the desired operation. 0 = Write; 1 = Read.
15-8	8	BYTE_ADDR	The desired byte address of the EEPROM.
7-0	8	DATA	For writes, the data to be written to the EEPROM. For reads, the data read from the EEPROM.

## 13.0 SAR Initialization - Example Tables

---

The following tables provide an example RS8235 SAR initialization of control registers, internal memory control structures, and external memory control structures.

## 13.1 Segmentation Initialization

### 13.1.1 Segmentation Control Registers

Before segmentation is enabled, the host must allocate and initialize all of the segmentation control registers. [Table 13-1](#) lists the initial value(s) for each field.

*Table 13-1. Table of Values for Segmentation Control Register Initialization*

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.
	ADV_ABR_TMPLT	1	Enable per-connection MCR & ICR ABR parameters.
	TX_FIFO_LEN	0x4	Set Transmit FIFO depth to four cells.
	CLP0_EOM	0	Disable CLP on EOM processing.
	OAM_STAT_ID	0x3	OAM global status queue set to 3.
	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.	
SEG_VBASE (SEG VCC Table & Schedule Table Base Address Register)	SEG_SCHB	0x1A5	Schedule table starts at 0xD280 in SAR shared memory.
	SEG_VCCB	0x17D	SEG VCC table starts at 0xBE80 in SAR shared memory.
SEG_TXBASE (Segmentation Transmit Queue Base Register)	SEG_TXB	0x13D	Transmit queues start at 0x9E80 in SAR shared memory.
	XMIT_INTERVAL	0x20	Transmit queue update interval set to 32.
	TX_EN	0x8	Transmit queues 0 through 8 are enabled.

### 13.1.2 Segmentation Internal Memory Control Structures

Before segmentation is enabled, the Host must allocate and initialize all of the segmentation internal memory control structures. [Table 13-2](#) lists the initial value(s) for each field.

**Table 13-2. Table of Values for Segmentation 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 zero.
	READ_UD	0x0	Must be initialized to zero.
	Rsvd	0x0	Must be initialized to zero.
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 zero.
	READ	0x0	Must be initialized to zero.
	Rsvd	0x0	Must be initialized to zero.

### 13.1.3 Segmentation SAR Shared Memory Control Structures

Before segmentation is enabled, the Host must allocate and initialize all of the segmentation SAR shared memory control structures. [Table 13-3](#) lists the initial value(s) for each field.

*Table 13-3. Table of Values for SEG SAR Shared Memory Initialization (1 of 2)*

Table	Field	Initialized Value	Notes
SEG VCC Table Entry 0 - (Words 0–6)	LAST_PNTR	0x0	Must be initialized to zero.
	ATM_HEADER	0x00100100	ATM Header, VPI = 0x1, VCI = 0x10.
	CRC_REM	0xFFFF_FFFF	Must be initialized to 0xFFFF_FFFF for AAL5 channels.
	BETAG	0	First AAL3/4 BTag/ETag pair will be zero.
	SN	0	First AAL3/4 SN = 0.
	MID	5	AAL3/4 MID = 5.
	STM_MODE	0	Status Message Mode enabled.
	STAT	0x2	Channel will use status queue 2.
	CURR_PNTR	0x0	Must be initialized to zero.
	VPC	0	VCC connection.
	GFR_PRI	0x2	GFR UBR Priority is 2.
	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 zero.	
SEG Buffer Descriptors			(No initialization required.)
SEG Transmit Queue Entries	VLD	0	Must be initialized to zero.
	LINK_HEAD	0	Must be initialized to zero.
	FND_CHAIN	0	Must be initialized to zero.
	SEG_BD_PNTR	0x0	Must be initialized to zero.
	Rsvd	0x0	Must be initialized to zero.

Table 13-3. Table of Values for SEG SAR Shared Memory Initialization (2 of 2)

Table	Field	Initialized Value	Notes
SEG Status Queue Entries	USER_PNTR	0x0	Must be initialized to zero.
	VLD	0	Must be initialized to zero.
	STOP	0	Must be initialized to zero.
	DONE	0	Must be initialized to zero.
	SINGLE	0	Must be initialized to zero.
	OVFL	0	Must be initialized to zero.
	I_EXP	0x0	Must be initialized to zero.
	I_MAN	0x0	Must be initialized to zero.
	SEG_VCC_INDEX	0x0	Must be initialized to zero.
	Rsvd	0x0	Must be initialized to zero.

## 13.2 Scheduler Initialization

### 13.2.1 Scheduler Control Registers

Before segmentation is enabled, the Host must allocate and initialize all of the Scheduler control registers. [Table 13-4](#) lists the initial value(s) for each field.

**Table 13-4. 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.
	QPCR_ENA7-0	0x04	Enable PCR limits on queue 2.
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.
PCR_QUE_INT01 (PCR Queue Interval 0 & 1 Register)	QPCR_INT0	0x0	Not used since only one global PCR queue enabled.
	QPCR_INT1	0x0	Not used since only one global PCR queue enabled.
PCR_QUE_INT_23 (PCR Queue Interval 2 & 3 Register)	QPCR_INT2	0x0	Not used since only one global PCR queue enabled.
	QPCR_INT3	0x16C	PCR interval = 364 schedule slots.
SCH_ABR_CON (Schedule ABR Constant Register)	ABR_TRM	0x23C	Set TRM to TM 4.1 default of 100 ms.
	ABR_ADTF	0xB2E	Set ADTF to TM 4.1 default of 0.5 second.
SCH_ABRBASE (ABR Decision Table Lookup Base Reg)	OOD_ENA	1	Enable out-of-rate ABR RM cells.
	OOD_INT	0x1C9D	Produces an out-of-rate interval of one cell per second.
	SCH_ABRB	0x1D1	ABR Table starts at 0xE880 in SAR shared memory.
SCH_CNG (ABR Congestion Register)	FBO_CNG	0x0	No congestion experienced.

### 13.2.2 Scheduler Internal Memory Control Structures

There are no internal memory scheduler structures that need to be initialized.

### 13.2.3 Scheduler SAR Shared Memory Control Structures

Before segmentation is enabled, the Host must allocate and initialize all of the scheduler SAR shared memory control structures. Table 13-5 lists the initial value(s) for each field.

Table 13-5. Table of Values for Scheduler SAR 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 one.
Schedule Table (Tunnel slot)	CBR	0	Slot will schedule a tunnel.
	PRI	0x3	Tunnel Priority Queue = 3.
	Rsvd	0xF	Set all reserved bits to one.
Schedule Table (DEFAULT)	Rsvd	0xF	Set all reserved bits to one.
SEG VCC Table Entry for VBR (Words 7–8)	BUCKET2	0x4	Offset of four 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 zero.
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 zero.
SEG VCC Table Entry for GFR (Words 7–9) <b>NOTE(S):</b> Set PRI_GFR lower than PRI in word 6.	MCRLIM_IDX	0x3	Offset of 3 into GFR MCR_Limit table.
	L1_EXP	0xF	
	L1_MAN	0x0	GCRA L = 32.
	I1_EXP	0xF	
	I1_MAN	0x120	GCRA I = 50. (Provides MCR ~ 3 Mbps.)
	Rsvd	0x0	Must be initialized to zero.

Table 13-5. Table of Values for Scheduler SAR Shared Memory Initialization (2 of 2)

Table	Field	Initialized Value	Notes
SEG VCC Table Entry for ER (Words 7–19)	CONG_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	TM 4.1 CRM parameter set to 20.
	MCR_INDEX	—	Output of ABR template.
	ICR_INDEX	—	Output of ABR template.
	FWD_ID	0x01	Forward RM cell ID field set to one.
	FWD_DIR	0	Forward RM cell DIR field set to zero.
	FWD_BN	0	Forward RM cell BN field set to zero.
	FWD_CI	0	Forward RM cell CI field set to zero.
	FWD_NI	0	Forward RM cell NI field set to zero.
	FWD_RA	0	Forward RM cell RA field set to zero.
	FWD_ER	0x64BF	Forward RM cell ER field set to approximately 360,000 cells per second.
	Rsvd	0x0	Must be initialized to zero.
	(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.

## 13.3 Reassembly Initialization

### 13.3.1 Reassembly Control Registers

Before reassembly is enabled, the Host must allocate and initialize all of the reassembly control registers. [Table 13-6](#) lists the initial value(s) for each field.

*Table 13-6. Table of Values for Reassembly 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 = 16,384 bytes.
	GDPRI	0x4	Global Service Discard Priority = 4.
Rsvd	0x0	Must be initialized to zero.	
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	0x3	Global OAM free buffer queue = 3.
	OAM_STAT_QU	0x3	Global OAM status queue = 3.
	Rsvd	0x0	Must be initialized to zero.
RSM_FQBASE (RSM Free Buffer Queue Base Register)	FBO1_BASE	0xB0	Free buffer queue bank 1 starts at 0x5800 in SAR shared memory.
	FBO0_BASE	0x30	Free buffer queue bank 0 starts at 0x1800 in SAR shared memory.

Table 13-6. Table of Values for Reassembly Control Register Initialization (2 of 2)

Register	Field	Initialized Value	Notes
RSM_FQCTRL (RSM Free Buffer Queue Control Register)	FBO_SIZE	0x0	Free buffer queue size is 64 entries.
	FWD_RND	1	Free buffer queues are processed in round robin fashion for credit return.
	FBO0_RTN	0	Free buffer queue bank 0 selected for buffer return processing.
	FWD_EN	0x4	Free buffers queues 0–4 enabled for buffer return processing.
	FBO_UD_INT	0x20	Free buffer queue update interval set to 32.
	Rsvd	0x0	Must be initialized to zero.
RSM_TBASE (RSM Table Base Register)	RSM_VCCB	0x105	RSM VCC Table starts at 0x8280 in SAR shared memory.
	RSM_ITB	0xF0	VPI Index Table starts at 0x7800 in SAR 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 0–16 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 SAR shared memory.
	Rsvd	0x0	Must be initialized to zero.

### 13.3.2 Reassembly Internal Memory Control Structures

Before reassembly is enabled, the Host must allocate and initialize all of the reassembly internal memory control structures. [Table 13-7](#) lists the initial value(s) for each field.

**Table 13-7. Table of Values for Reassembly 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 zero.
	READ_UD	0x0	Must be initialized to zero.
	Rsvd	0x0	Must be initialized to zero.
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 zero.
	UPDATE	0x0	Must be initialized to zero.
	READ	0x0	Must be initialized to zero.
	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 zero.
Global Time-out Table	TERM_TOCNT0	0x400	Provides a 133 ms time-out period.
	TERM_TOCNT1	0xFFFF	Provides a 8.5 sec time-out period.
	TERM_TOCNT2	0x400	Provides a 133 ms time-out period.
	TERM_TOCNT3	0x400	Provides a 133 ms time-out period.
	TERM_TOCNT4	0x400	Provides a 133 ms time-out period.
	TERM_TOCNT5	0x400	Provides a 133 ms time-out period.
	TERM_TOCNT6	0x400	Provides a 133 ms time-out period.
	TERM_TOCNT7	0x400	Provides a 133 ms time-out period.
	TO_VCC_INDEX	0x0	Must be initialized to zero.
	Rsvd	0x0	Must be initialized to zero.

### 13.3.3 Reassembly SAR Shared Memory Control Structures

Before reassembly is enabled, the Host must allocate and initialize all of the reassembly SAR shared memory control structures. Table 13-8 lists the initial value(s) for each field.

Table 13-8. Table of Values for RSM SAR Shared Memory Initialization (1 of 2)

Table	Field	Initialized Value	Notes
VPI Index Table Entry 0	VP_EN	1	VPI = 0 enabled. <b>NOTE(S):</b> 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 SAR 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 zero.
RSM VCC Table Entry 0	FF_DSC	1	Enable FIFO Full EPD.
	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 two.
	TO_INDEX	0x1	Point to TERM_TOCNT1 global time-out value.
	AAL_EN	0x082	Message Status mode and Frame Relay Discard enabled.
	TO_LAST	0	Not last VCC entry processed for time-out.
RSM VCC Table Entry 0	TO_EN	1	Time-out processing enabled on this channel.
	CUR_TOCNT	0x0	Must be initialized to zero.
	ABR_CTRL	0x0	No ABR service enabled on this channel.
	PDU_FLAGS	0x002	Must be initialized to two.
	TOT_PDU_LEN	0x0	Must be initialized to zero.
	CRC_REM	0xFFFF_FFFF	Must be initialized to 0xFFFF_FFFF for AAL5 channels.
	STAT	0x1	Channel uses status queue 1.
	BFR1	0x2	Channel uses free buffer queue 2 for COM buffers.

Table 13-8. Table of Values for RSM SAR Shared Memory Initialization (2 of 2)

Table	Field	Initialized Value	Notes
RSM VCC Table Entry 0	BFR0	0x1	Channel uses free buffer queue 1 for BOM buffers.
	SEG_VCC_INDEX	0x4	Corresponding seg channel = 4 for ABR service.
	SERV_DIS	0x0	Must be initialized to zero.
	RX_COUNTER	0x100	Initial firewall credit = 256.
	Rsvd	0x0	Must be initialized to zero.
RSM Buffer Descriptors	NEXT_PTR	0x0	Initialization optional.
	BUFF_PNTR	0x2000	Buffer address of 0x2000.
RSM Free Buffer Queue Entries	VLD	0-1	Free buffer queue can 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 zero.
	VCC_INDEX	0	Must be initialized to zero.
	Rsvd	0x0	Must be initialized to zero.
RSM Status Queue Entries	VLD	0	Must be initialized to zero.
	(All Other Entries)	0	Must be initialized to zero.
LECID Table	LECID0-31	0x20	LANE LECID = 32.

## 13.4 General Initialization

Before the SAR is enabled, the Host must allocate and initialize all of the general SAR control registers. [Table 13-9](#) lists the initial value(s) for each field.

*Table 13-9. Table of Values for General Control Register Initialization (1 of 2)*

Register	Field	Initialized Value	Notes
CONFIG0 (Configuration Register 0)	LP_ENABLE	0	Local processor not 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 zero.
	INT_LBANK	0-1	Should be set to zero 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.
	LP_BWAIT	0	Selects zero wait states between consecutive data cycles during local processor.
	MEMCTRL	0	Selects zero wait states SAR shared memory.
	BANKSIZE	0x3	512 KB banks selected.
	DIVIDER	0x0	Divide by 128 selected for CLOCK prescaler.
Rsvd	0x0	MUST be initialized to zero.	
INT_DELAY (Interrupt Delay Register)	TIMER_LOC	0	Interrupt hold-off timer used with HINT*.
	EN_TIMER	0	Disable status queue interrupt timer delay.
	EN_STAT_CNT	1	Enable status queue interrupt counter delay.
	STAT_CNT[7:0]	0x35	Interrupt delay counter set to 53.

Table 13-9. Table of Values for General Control Register Initialization (2 of 2)

Register	Field	Initialized Value	Notes
HOST_ST_WR (Host Status Write Register)	RSM_HS_WRITE	—	Read twice after SAR reset.
	RSM_LS_WRITE	—	Read twice after SAR reset.
HOST_I_STAT1 (Host Interrupt Status Register 1)	(ALL)	—	Read twice HOST_ST_WR reset.
HOST_I_STAT0 (Host Interrupt Status Register 0)	(ALL)	—	Read twice HOST_I_STAT1 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 relieves and framer interrupt.
PCI Configuration Space	COMMAND	0x0346	Enable all functions of PCI interface.
	LAT_TIMER	0x10	Latency timer = 16 clock periods.
	BASE_ADDRESS_REGISTER_0	0x01	Base address of SAR device in PCI memory space is 0x0100_0000.
	INTERRUPT_LINE	0x00	Interrupt vector = 0.
	MAX_BUR_LEN	0x10	Max burst length = 16.
	(ALL OTHER FIELDS)	—	Hard-coded reads only.
PCI COMMAND Register	FB_EN	1	Enable master fast back-to-back across target.
	SE_EN	1	Enable SERR* output pin.
	PE_EN	1	Enable parity error detection and report.
	M_EN	1	Enable RS8235 device master on the PCI bus.
	MS_EN	1	Enable RS8235 memory space access across the PCI bus.
PCI STATUS Register			(No initialization required.)
PCI SPECIAL_STATUS Register			(No initialization required.)
PCI EEPROM Register			(No initialization required.)



## 14.0 Electrical and Mechanical Specifications

### 14.1 Timing

#### 14.1.1 PCI Bus Interface Timing

All PCI bus interface signals are synchronous to the PCI bus clock, HCLK, except for HRST\* and HINT\*. [Table 14-1](#) provides the PCI bus interface timing parameters. [Figure 14-1](#) and [Figure 14-2](#) illustrate this timing.

*Table 14-1. PCI Bus Interface Timing Parameters (1 of 2)*

Symbol	Parameter	Min	Max	Units
$t_{cyc}$	HCLK Cycle Time <sup>(1)</sup>	30	—	ns
$t_{high}$	HCLK High Time <sup>(1)</sup>	11	19	ns
$t_{low}$	HCLK Low Time <sup>(1)</sup>	11	19	ns
$t_{su}$	HAD Input Setup Time to HCLK <sup>(1)</sup>	7	—	ns
	HC/BE Input Setup Time to HCLK <sup>(1)</sup>	7	—	ns
	HPAR Input Setup Time to HCLK <sup>(1)</sup>	7	—	ns
	HFRAME* Input Setup Time to HCLK <sup>(1)</sup>	7	—	ns
	HIRDY* Input Setup Time to HCLK <sup>(1)</sup>	7	—	ns
	HTRDY* Input Setup Time to HCLK <sup>(1)</sup>	7	—	ns
	HSTOP* Input Setup Time to HCLK <sup>(1)</sup>	7	—	ns
	HDEVSEL* Input Setup Time to HCLK <sup>(1)</sup>	7	—	ns
	HIDSEL Input Setup Time to HCLK <sup>(1)</sup>	7	—	ns
	HGNT* Input Setup Time to HCLK <sup>(1)</sup>	10	—	ns
HPERR* Input Setup Time to HCLK <sup>(1)</sup>	7	—	ns	
$t_h$	Input Hold Time from HCLK—All Inputs <sup>(1)</sup>	0	—	ns

Table 14-1. PCI Bus Interface Timing Parameters (2 of 2)

Symbol	Parameter	Min	Max	Units
$t_{val}$	HCLK to HAD Valid Delay <sup>(2)</sup>	2	11	ns
	HCLK to HC/BE Valid Delay <sup>(2)</sup>	2	11	ns
	HCLK to HPAR Valid Delay <sup>(2)</sup>	2	11	ns
	HCLK to HFRAME* Valid Delay <sup>(2)</sup>	2	11	ns
	HCLK to HIRDY* Valid Delay <sup>(2)</sup>	2	11	ns
	HCLK to HSTOP* Valid Delay <sup>(2)</sup>	2	11	ns
	HCLK to HDEVSEL Valid Delay <sup>(2)</sup>	2	11	ns
	HCLK to HPERR* Valid Delay <sup>(2)</sup>	2	11	ns
	HCLK to HREQ Valid Delay <sup>(2)</sup>	2	12	ns
	HCLK to HSERR* Valid Delay <sup>(2)</sup>	2	11	ns
	HCLK to STAT Valid Delay <sup>(3)</sup>	2	20	ns
$t_{on}$	Float to Active Delay—All Three-state Outputs <sup>(2)</sup>	2	—	ns
$t_{off}$	Active to Float Delay—All Three-state Outputs <sup>(2)</sup>	—	28	ns
$t_{rst-off}$	Reset Active to Output Float Delay	—	40	ns

**NOTE(S):**

- (1) See Figure 14-1 for waveforms and definitions.
- (2) See Figure 14-2 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 14-1. PCI Bus Input Timing Measurement Conditions

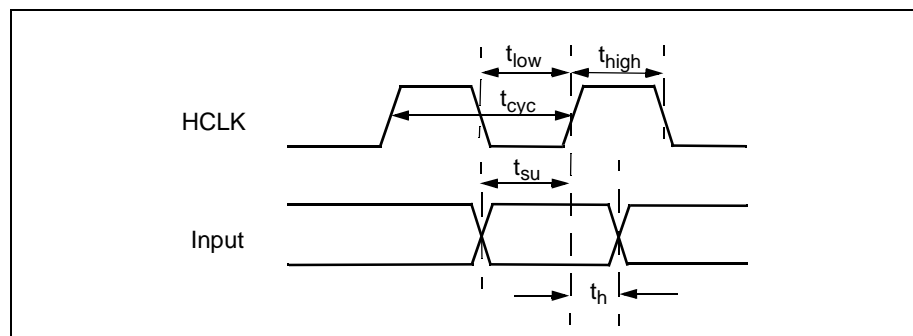
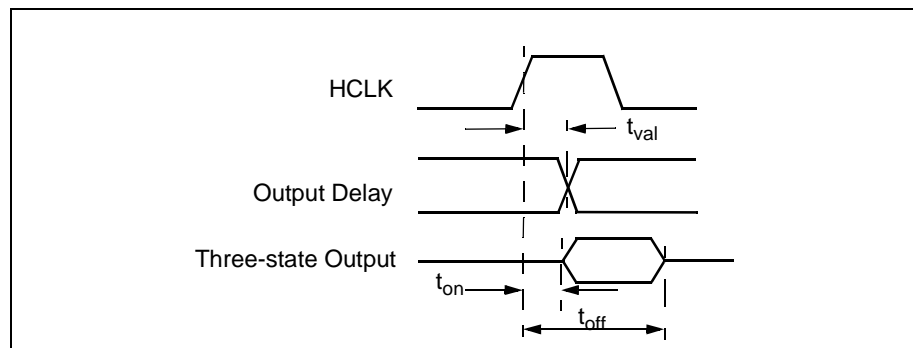


Figure 14-2. PCI Bus Output Timing Measurement Conditions



### 14.1.2 ATM Physical Interface Timing—UTOPIA and Slave UTOPIA

All ATM physical interface signals are synchronous to the interface clock, RXCLK/TXCLK, except for TXCLAV\* and RXCLAV\* in the slave UTOPIA mode. Timing parameters for the UTOPIA interface are provided in Table 14-2. Table 14-3 provides the timing parameters for the slave UTOPIA interface. Timing diagrams for both interfaces are provided in Figure 14-3 and Figure 14-4.

Table 14-2. UTOPIA Interface Timing Parameters

Symbol	Parameter	Min	Max	Units
$t_{cyc}$	RXCLK/TXCLK Cycle Time <sup>(1)</sup>	30	—	ns
$t_{high}$	RXCLK/TXCLK High Time <sup>(1)</sup> (% of $t_{cyc}$ )	40	60	%
$t_{low}$	RXCLK/TXCLK Low Time <sup>(1)</sup> (% of $t_{cyc}$ )	40	60	%
$t_{su}$	RXD Input Setup Time to RXCLK <sup>(1)</sup>	10	—	ns
	RXPAR Input Setup Time to RXCLK <sup>(1)</sup>	10	—	ns
	RXSOC Input Setup Time to RXCLK <sup>(1)</sup>	10	—	ns
	RXCLAV* Input Setup Time to RXCLK <sup>(1)</sup>	10	—	ns
	TXCLAV* Input Setup Time to TXCLK <sup>(1)</sup>	10	—	ns
$t_h$	RXD Input Hold Time from RXCLK <sup>(1)</sup>	1	—	ns
	RXPAR Input Hold Time from RXCLK <sup>(1)</sup>	1	—	ns
	RXSOC Input Hold Time from RXCLK <sup>(1)</sup>	1	—	ns
	RXCLAV* Input Hold Time from RXCLK <sup>(1)</sup>	1	—	ns
	TXCLAV* Input Hold Time from TXCLK <sup>(1)</sup>	1	—	ns
$t_{val}$	TXCLK to TXD Valid Delay <sup>(2)</sup>	2	18	ns
	TXCLK to TXPAR Valid Delay <sup>(2)</sup>	2	18	ns
	TXCLK to TXSOC Valid Delay <sup>(2)</sup>	2	18	ns
	TXCLK to TXEN* Valid Delay <sup>(2)</sup>	2	20	ns
	RXCLK to RXEN* Valid Delay <sup>(2)(loca)</sup>	2	20	ns

**NOTE(S):**

(1) See Figure 14-3 for waveforms and definitions.

(2) See Figure 14-4 for waveforms and definitions. The output delays are measured with a 25 pF load.

Table 14-3. Slave UTOPIA Interface Timing Parameters

Symbol	Parameter	Min	Max	Units
$t_{cyc}$	RXCLK/TXCLK Cycle Time <sup>(1)</sup>	30	—	ns
$t_{high}$	RXCLK/TXCLK High Time <sup>(1)</sup> (% of $t_{cyc}$ )	40	60	%
$t_{low}$	RXCLK/TXCLK Low Time <sup>(1)</sup> (% of $t_{cyc}$ )	40	60	%
$t_{su}$	RXD Input Setup Time to RXCLK <sup>(1)</sup>	6	—	ns
	RXPAR Input Setup Time to RXCLK <sup>(1)</sup>	3	—	ns
	RXSOC Input Setup Time to RXCLK <sup>(1)</sup>	8	—	ns
	RXEN* Input Setup Time to RXCLK <sup>(1)</sup>	10	—	ns
	TXEN* Input Setup Time to TXCLK <sup>(1)</sup>	3	—	ns
$t_h$	RXD Input Hold Time from RXCLK <sup>(1)</sup>	1	—	ns
	RXPAR Input Hold Time from RXCLK <sup>(1)</sup>	1	—	ns
	RXSOC Input Hold Time from RXCLK <sup>(1)</sup>	1	—	ns
	RXEN* Input Hold Time from RXCLK <sup>(1)</sup>	1	—	ns
	TXEN* Input Hold Time from TXCLK <sup>(1)</sup>	1	—	ns
$t_{val}$	TXCLK to TXD Valid Delay <sup>(2)</sup>	2	16	ns
	TXCLK to TXPAR Valid Delay <sup>(2)</sup>	2	16	ns
	TXCLK to TXCLAV* Valid Delay <sup>(2)</sup>	2	20	ns
	RXCLK to RXCLAV* Valid Delay <sup>(2)</sup>	2	20	ns
	TXCLK to XSOC Valid Delay <sup>(2)</sup>	2	17	ns

**NOTE(S):**  
 (1) See Figure 14-3 for waveforms and definitions.  
 (2) See Figure 14-4 for waveforms and definitions. The output delays are measured with a 25 pF load.

Figure 14-3. UTOPIA and Slave UTOPIA Input Timing Measurement Conditions

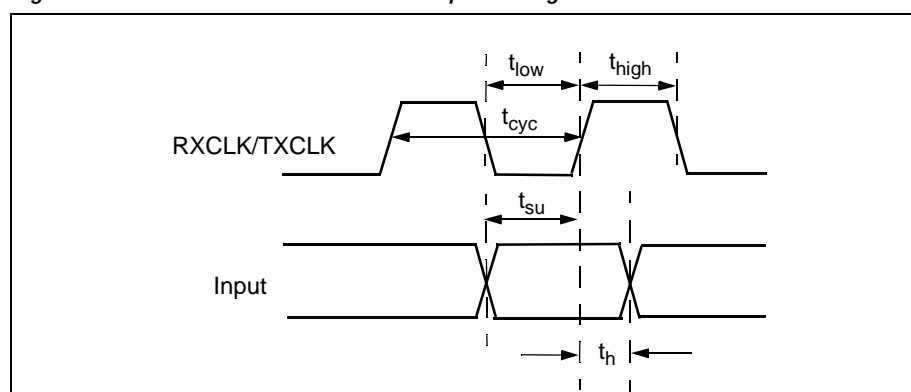
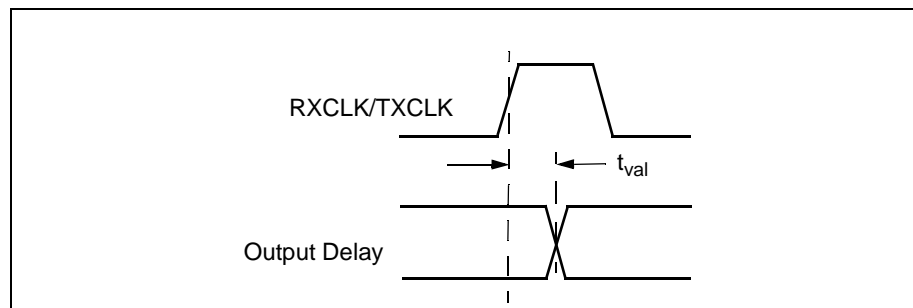


Figure 14-4. UTOPIA and Slave UTOPIA Output Timing Measurement Conditions



### 14.1.3 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 SAR 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 14-4 specifies the timing parameters of the three clocks. Figure 14-5 and Figure 14-6 illustrate this timing.

Table 14-4. System Clock Timing (1 of 2)

Symbol	Parameter	Min	Max	Units
<b>Input Clocks<sup>(1)</sup></b>				
$t_{cf}$	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
<b>Output Clocks<sup>(2)</sup></b>				
$t_{sf}$	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

Table 14-4. System Clock Timing (2 of 2)

Symbol	Parameter	Min	Max	Units
$t_d$	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

**NOTE(S):**  
 (1) See Figure 14-5 for waveforms and definitions.  
 (2) See Figure 14-6 for waveforms and definitions. The outputs are measured with a load of 35 pf.

Figure 14-5. Input System Clock Waveform

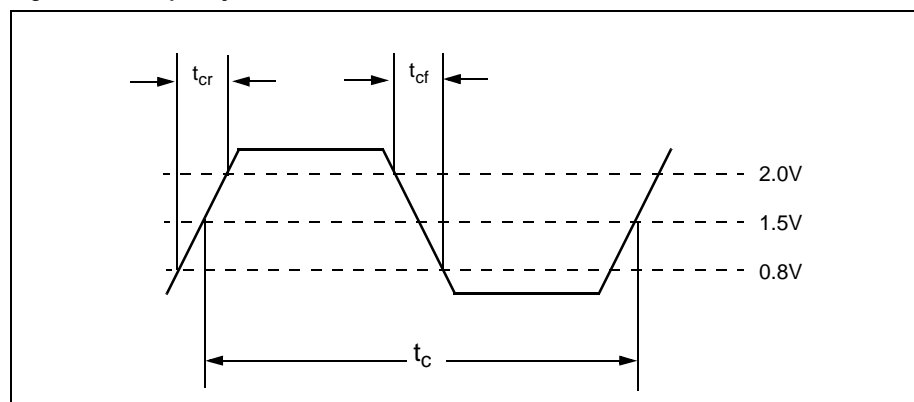
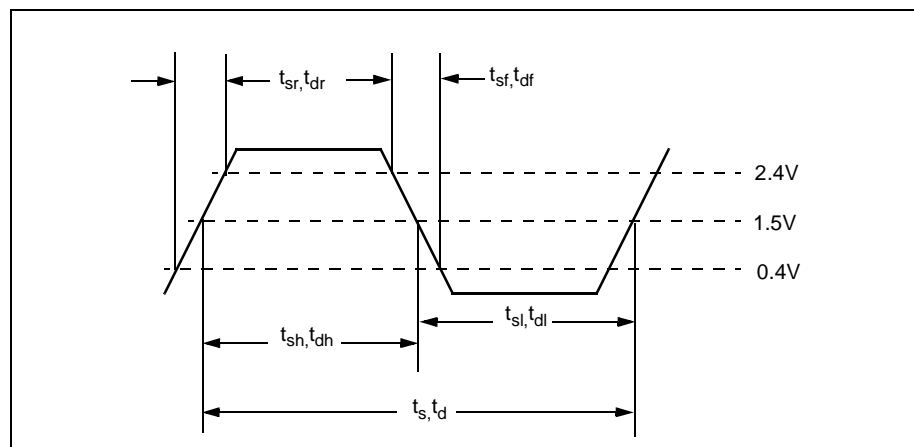


Figure 14-6. Output System Clock Waveform



### 14.1.4 RS8235 Memory Interface Timing

Memory access times and other timing requirements are specified at three typical implementations of one, two, and four banks of by\_8 SRAM. Table 14-5 gives the number of loads per bank for the different SRAM organizations while Table 14-6 gives the capacitive loading used in the timing specifications given for the three typical implementations. RS8235 memory interface timing is given in Table 14-7. Figure 14-7 and Figure 14-8 illustrate the RS8235's memory read and write timing. See Table 9-2, *Memory Size in Bytes* for details of memory bank organization.

Table 14-5. SRAM Organization Loading Dependencies

Signal	Loads/Bank <sup>(1)</sup>		
	by_16 SRAM	by_8 SRAM	by_4 SRAM
LADDR[18:0] <sup>(1)</sup>	2	4	8
LDATA[31:0] <sup>(1)</sup>	1	1	1
MWR* <sup>(1)</sup>	2	N/A	N/A
MOE* <sup>(1)</sup>	2	4	8
MCSx* <sup>(1, 2)</sup>	2	4	8
MWEx* <sup>(1)</sup>	1	1	2

**NOTE(S):**  
<sup>(1)</sup> Typical input loading for SRAM is 7 pF. For exact values, consult the SRAM databook.  
<sup>(2)</sup> Only connected to one bank by definition.

Table 14-6. SAR 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
<b>Memory Interface Loading<sup>(1)</sup></b>				
LADDR[18:0] <sup>(1)</sup>	150	100	50	pF
MOE*	150	100	50	pF
MWR* <sup>(2)</sup>	—	50	35	pF
LDATA[31:0]	50	35	25	pF
MWE[3:0]*	50	35	25	pF
MCS[3:0]*	50	35	25	pF

**NOTE(S):**  
<sup>(1)</sup> In general, the LADDR loading is the most critical parameter, one bank of by\_8 SRAM has the same address loading as two banks of by\_16 and 1/2 bank of by\_4 SRAM. For example, use the timing from the two banks of by\_8 column for four banks of by\_16 SRAM, or one bank of by\_4 SRAM.  
<sup>(2)</sup> For by\_16 SRAM, the WE\* input has the same loading as the address bus and OE\*; therefore, 16 loads specified by the four banks of by\_8 is not applicable, since the maximum number of by\_16 SRAMs supported is eight.

Table 14-7. RS8234 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	
<b>Memory Read Timing</b>								
$t_{rc}$	READ Cycle Time <sup>(2)</sup>	T		T		T		ns
$t_{aov}$	LADDR[18:0] Output Valid <sup>(1)</sup>	1	10	1	8	1	7	ns
$t_{mcssov}$	MCS[3:0]* Output Valid <sup>(1)</sup>	1	10	1	8	1	7	ns
$t_{mweov}$	MWE* Byte Enables Output Valid <sup>(1)</sup> (RAMMODE = 1)	1	10	1	8	1	7	ns
$t_{moel}$	MOE* Low <sup>(1)</sup>		17		16		15	ns
$t_{moeh}$	MOE* High <sup>(1)</sup>		10		8		7	ns
$t_{dod}$	LDATA Output Disable <sup>(1)</sup>		7		6		6	ns
$t_{dis}$	LDATA Input Setup <sup>(1)</sup>	5		5		5		ns
$t_{dh}$	LDATA Input Hold <sup>(1)</sup>	0		0		0		ns
$t_{doe}$	LDATA Driven by SAR <sup>(1)</sup>	12		11		11		ns
$t_{dodmoe}$	LDATA Disable to MOE* Low <sup>(1)</sup>	0		0		0		ns
$t_{moedoe}$	MOE* High to LDATA Driven <sup>(1)</sup>	6		7		8		ns
<b>Memory Write Timing</b>								
$t_{wc}$	WRITE Cycle Time <sup>(2)</sup>	T		T		T		ns
$t_w$	MWR*, MWE[3:0]* Width <sup>(2, 4)</sup>	T/2-2		T/2-2		T/2-2		ns
$t_{aov}$	LADDR[18:0] Output Valid <sup>(4)</sup>	1	10	1	8	1	7	ns
$t_{mcssov}$	MCS[3:0]* Output Valid <sup>(4)</sup>	1	10	1	8	1	7	ns
$t_{mweov}$	MWE* Byte Enables Output Valid <sup>(1)</sup> (RAMMODE = 1)	1	10	1	8	1	7	ns
$t_{dov}$	LDATA Output Valid <sup>(4)</sup>	1	10	1	10	1	10	ns
$t_{mwel}$	MWE[3:0]* Low <sup>(4)</sup>	T/2-2	16		16		16	ns
$t_{mweh}$	MWE[3:0]* High <sup>(4)</sup>		1		1		1	ns
$t_{mwrl}$	MWR[3:0]* Low <sup>(4)</sup>	T/2-2	16		16		16	ns
$t_{mwrh}$	MWR[3:0]* High <sup>(4)</sup>		1		1		1	ns
<b>NOTE(S):</b>								
(1) See Figure 14-7 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 14-8 for waveforms.								
(5) SYSCLK shown for reference only.								

Figure 14-7. RS8235 Memory Read Timing

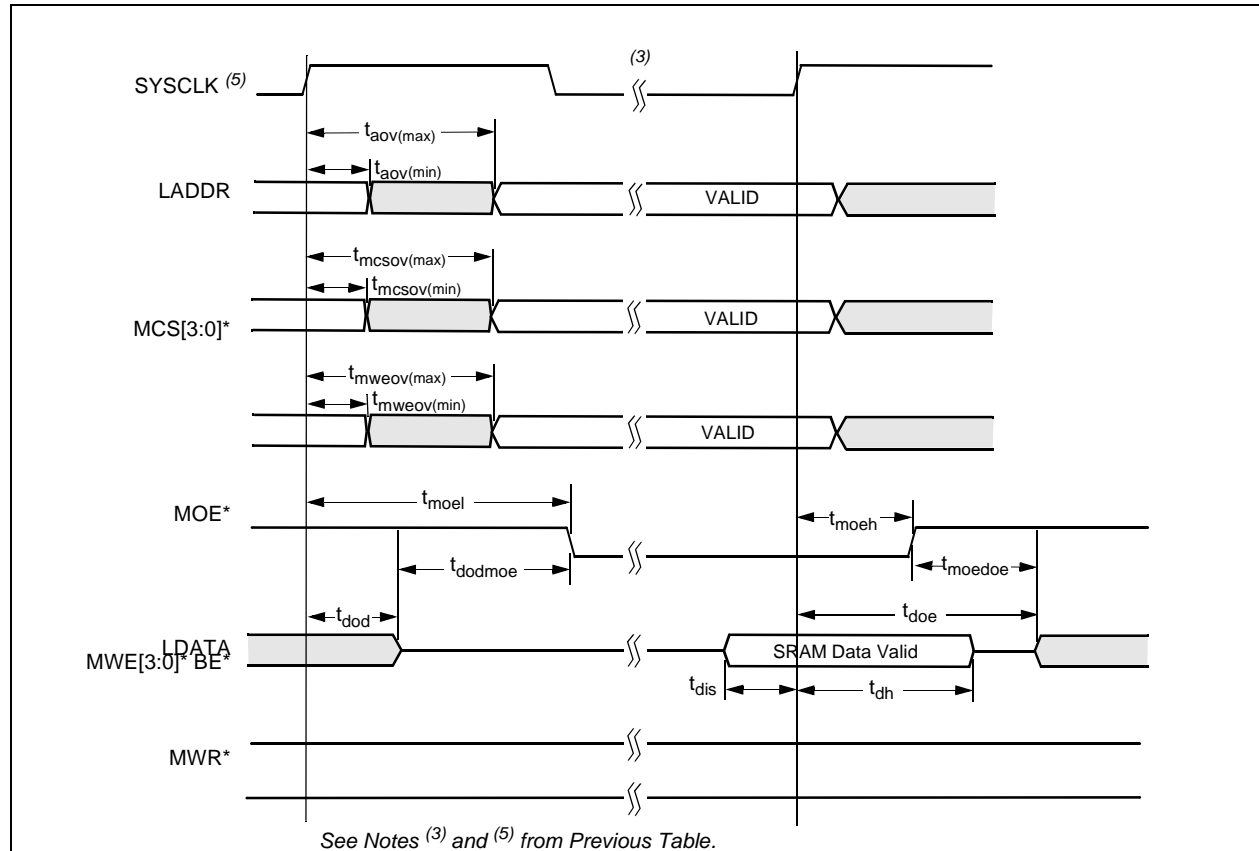
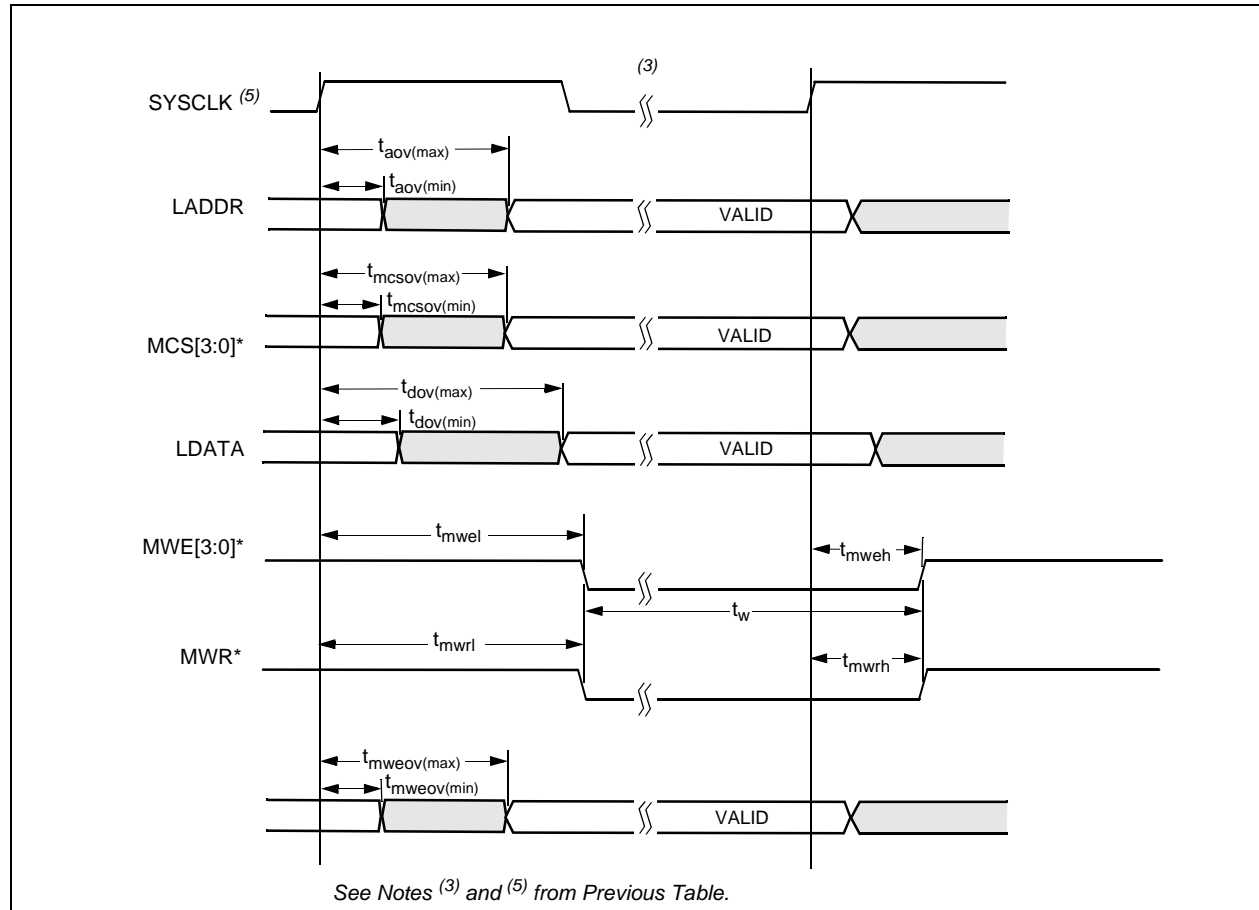


Figure 14-8. RS8235 Memory Write Timing



### 14.1.5 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 RS8235 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 RS825x ATM Receiver/Transmitter. Timing is given in [Table 14-8](#), [Figure 14-9](#), and [Figure 14-10](#).

**NOTE:** PROCMODE must be set to a logic high on the RS8235.

**Table 14-8. PHY Interface Timing (PROCMODE = 1)**

Symbol	Parameter	Min	Max	Units
<b>Synchronous Inputs</b>				
$t_{is}$	PWAIT* Input Setup <sup>(1)</sup>	14		ns
	LDATA Input Setup <sup>(1)</sup>	5		ns
$t_{ih}$	PWAIT* Input Hold <sup>(1)</sup>	0		ns
	LDATA Input Hold <sup>(1)</sup>	0		ns
<b>Synchronous Outputs</b>				
$t_{ov}$	PRDY* Output Valid Delay <sup>(2)</sup>	5	15	ns
	PAS Output Valid Delay <sup>(2)</sup>	5	15	ns
	PCS* Output Valid Delay <sup>(2)</sup>	5	15	ns
	PWNR Output Valid Delay <sup>(2)</sup>	5	15	ns
	LDATA* Output Valid Delay <sup>(2)</sup>	1	10	ns
	LADDR Output Valid Delay <sup>(2)</sup>	1	10	ns
<b>NOTE(S):</b>				
(1) See <a href="#">Figure 14-9</a> for waveforms and definitions.				
(2) See <a href="#">Figure 14-10</a> for waveforms and definitions. The outputs are measured with a load of 35 pF.				

**Figure 14-9. Synchronous PHY Interface Input Timing**

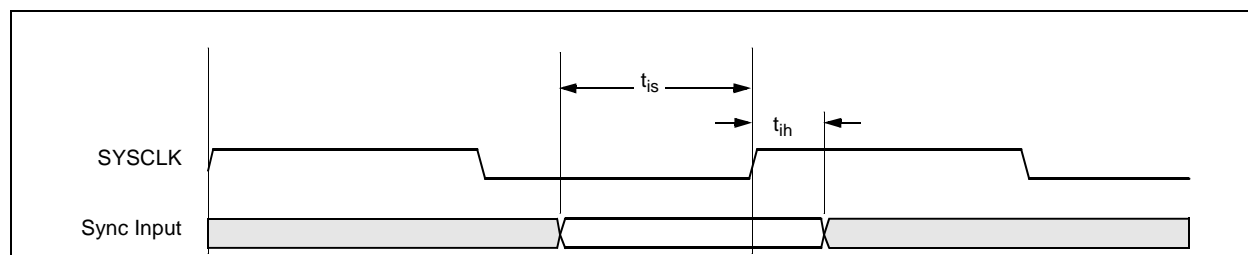
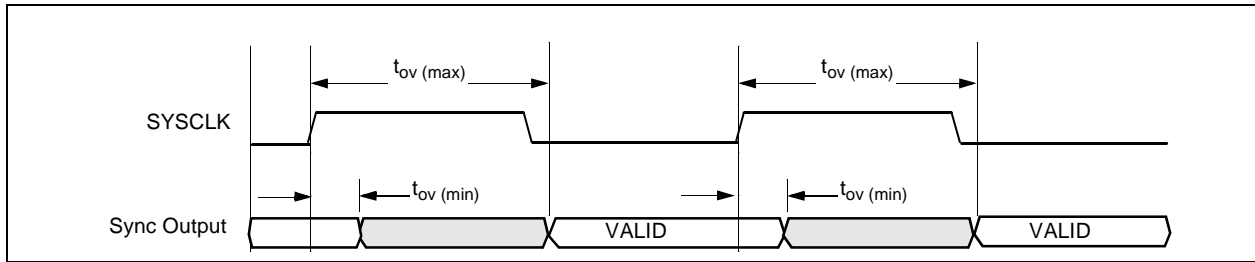


Figure 14-10. Synchronous PHY Interface Output Timing



## 14.2 Absolute Maximum Ratings

Stresses above those listed as absolute maximum ratings (Table 14-9) may cause permanent damage to the device. This is a stress rating only, and 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 5.5 V can induce destructive latchup.

Table 14-9. Absolute Maximum Ratings

Parameter	Value	Unit
Supply Voltage (Vdd)	-0.5 to +4	Volts
Input Voltage	-0.5 to +5.5	Volts
Output Voltage	-0.5 to (Vdd + 0.3)	Volts
Storage Temperature	-40 to 125	C
Maximum Current @ Max Clock Frequencies and 3.6 V	400	mA
Moisture Sensitivity Level (MSL)	4	—
FIT @ 55 jC	55	—
Theta Ja ( $T_a = 85$ jC, $T_j = 125$ jC)	17	C/W

## 14.3 DC Characteristics

Table 14-10. DC Characteristics

Parameter	Conditions	Min	Max	Unit
Operating Supply Voltage (Vdd)		3.0	3.6	Volts
Output Voltage High	$I_{OH} = -500 \mu\text{A}$ (for all PCI signals)	$0.9 \cdot V_{dd}$		Volts
	$I_{OH} = -4.0 \text{ mA}$ (for all other signals)	2.4		Volts
Output Voltage Low	$I_{OL} = 1500 \mu\text{A}$ (for all PCI signals)		$0.1 \cdot V_{dd}$	Volts
	$I_{OL} = 4.0 \text{ mA}$ (for all other signals)		0.4	Volts
Input Voltage High (PCI)		$0.5 \cdot V_{dd}$	5.25	Volts
Input Voltage High (CLK2X, RXCLK/TXCLK)		$0.7 \cdot V_{dd}$	5.25	Volts
Input Voltage High (all others)		2.0	5.25	Volts
Input Voltage Low (PCI)		-0.5	$0.3 \cdot V_{dd}$	Volts
Input Voltage Low (CLK2X, RXCLK/TXCLK)		0	$0.3 \cdot V_{dd}$	Volts
Input Voltage Low (all others)		0	0.8	Volts
Input Leakage Current	$V_{in} = V_{dd} \text{ or } GND$	-10	10	$\mu\text{A}$
Three-state Output Leakage Current	$V_{out} = V_{dd} \text{ or } GND$	-10	10	$\mu\text{A}$
Pullup Current		30	100	$\mu\text{A}$
Input Capacitance			7	pF
Output Capacitance			7	pF
<b>NOTE(S):</b> All outputs are CMOS drive levels and can be used with CMOS or TTL logic.				

## 14.4 Mechanical Specifications

The RS8235's 208-pin Quad Flat Pack package is illustrated in Figure 14-11. Figure 14-12 is a pinout configuration of the RS8235, and a pin listing is given in Table 14-11.

Figure 14-11. 208-Pin Plastic Quad Flat Pack

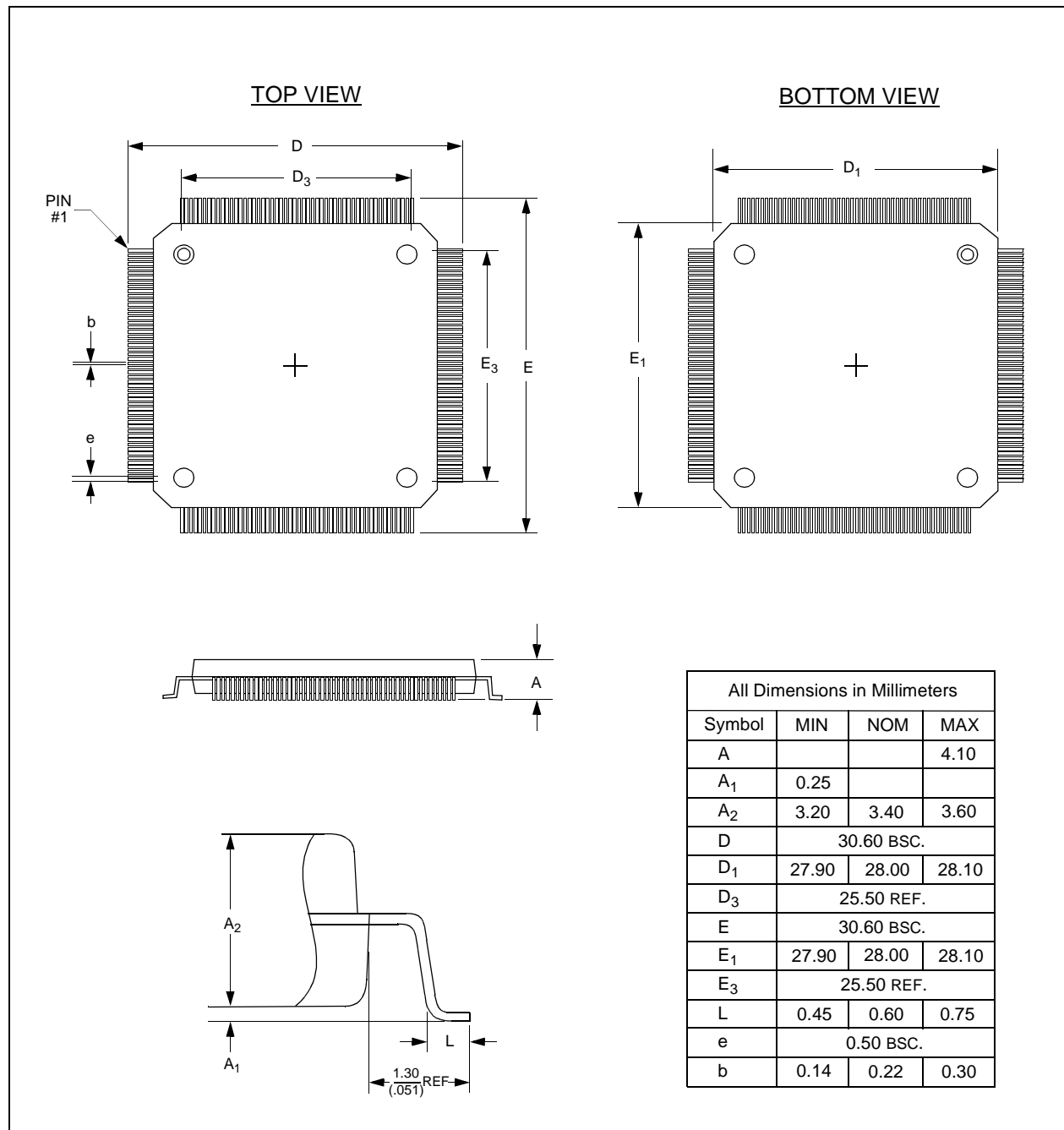


Figure 14-12. RS8235 Pinout Configuration

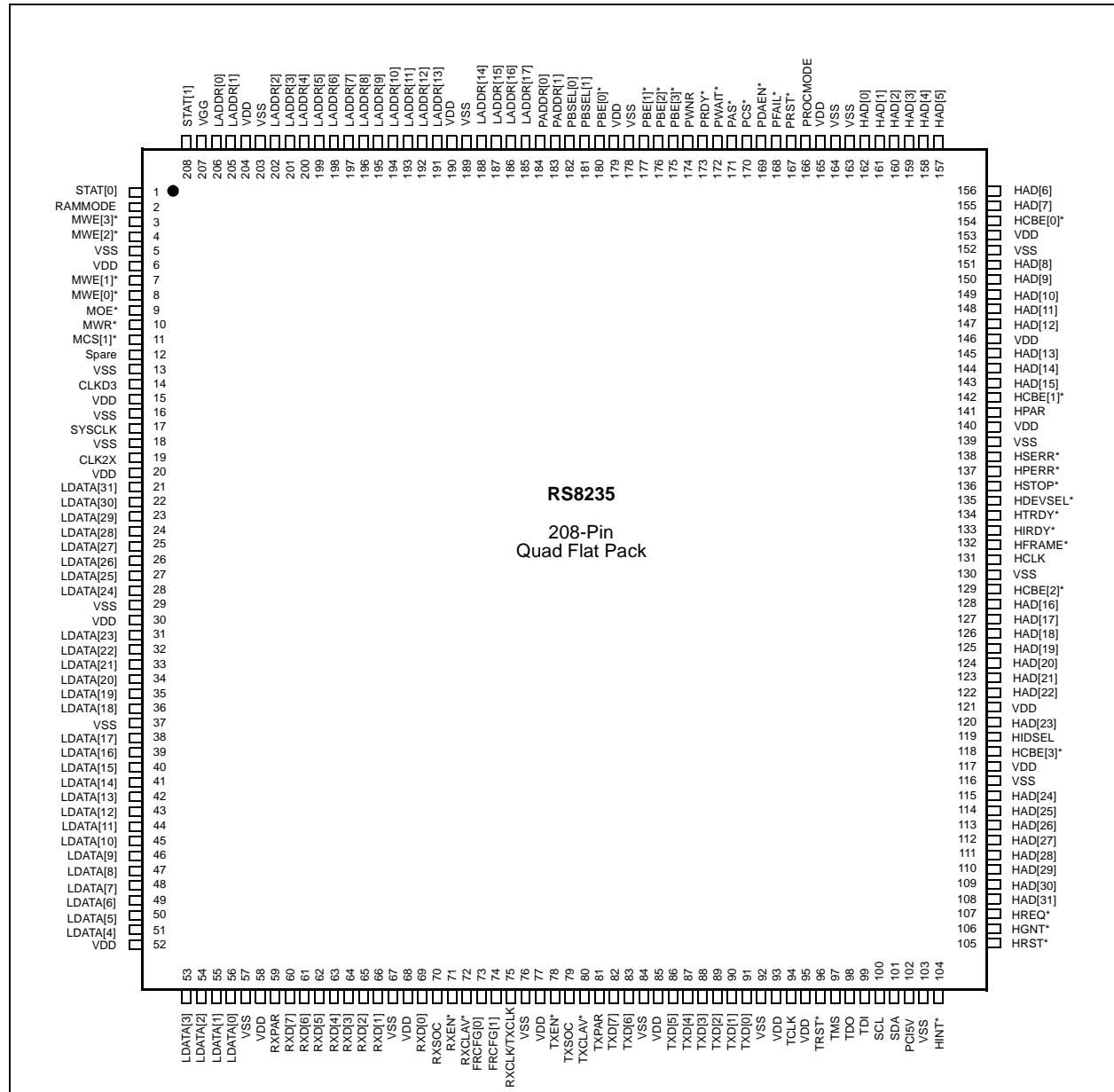


Table 14-11. RS8235 Pinout (1 of 6)

Pin	Pin Label	I/O
1	STAT[0]	0
2	RAMMODE	I
3	MWE[3]*	0
4	MWE[2]*	0
5	VSS(GND)	-
6	VDD(PWR)	-
7	MWE[1]*	0
8	MWE[0]*	0
9	MOE*	0
10	MWR*	0
11	MCS[1]*	0
12	spare	
13	VSS[GND]	-
14	CLKD3	0
15	VDD[PWR]	-
16	VSS[GND]	-
17	SYSCLK	0
18	VSS[GND]	-
19	CLK2X	I
20	VDD[PWR]	-
21	LDATA[31]	I/O
22	LDATA[30]	I/O
23	LDATA[29]	I/O
24	LDATA[28]	I/O
25	LDATA[27]	I/O
26	LDATA[26]	I/O
27	LDATA[25]	I/O
28	LDATA[24]	I/O
29	VSS[GND]	-
30	VDD[PWR]	-
31	LDATA[23]	I/O
32	LDATA[22]	I/O
33	LDATA[21]	I/O
34	LDATA[20]	I/O
35	LDATA[19]	I/O
36	LDATA[18]	I/O

## 14.4 Mechanical Specifications

## Endstation ATM ServiceSAR with xBR Traffic Management

Table 14-11. RS8235 Pinout (2 of 6)

Pin	Pin Label	I/O
37	VSS[GND]	–
38	LDATA[17]	I/O
39	LDATA[16]	I/O
40	LDATA[15]	I/O
41	LDATA[14]	I/O
42	LDATA[13]	I/O
43	LDATA[12]	I/O
44	LDATA[11]	I/O
45	LDATA[10]	I/O
46	LDATA[9]	I/O
47	LDATA[8]	I/O
48	LDATA[7]	I/O
49	LDATA[6]	I/O
50	LDATA[5]	I/O
51	LDATA[4]	I/O
52	VDD[PWR]	–
53	LDATA[3]	I/O
54	LDATA[2]	I/O
55	LDATA[1]	I/O
56	LDATA[0]	I/O
57	VSS[GND]	–
58	VDD[PWR]	–
59	RXPAR	I
60	RXD[7]	I
61	RXD[6]	I
62	RXD[5]	I
63	RXD[4]	I
64	RXD[3]	I
65	RXD[2]	I
66	RXD[1]	I
67	VSS[GND]	–
68	VDD[PWR]	–
69	RXD[0]	I/O
70	RXSOC	I
71	RXEN*	I/O
72	RXCLAV*	I/O

Table 14-11. RS8235 Pinout (3 of 6)

Pin	Pin Label	I/O
73	FRCFG[0]	I
74	FRCFG[1]	I
75	RXCLK/TXCLK	I
76	VSS[GND]	–
77	VDD[PWR]	–
78	TXEN*	I/O
79	TXSOC	I/O
80	TXCLAV*	I/O
81	TXPAR	O
82	TXD[7]	O
83	TXD[6]	O
84	VSS[GND]	–
85	VDD[PWR]	–
86	TXD[5]	O
87	TXD[4]	O
88	TXD[3]	O
89	TXD[2]	O
90	TXD[1]	O
91	TXD[0]	O
92	VSS[GND]	–
93	VDD[PWR]	–
94	TCLK	I
95	VDD[PWR]	–
96	TRST*	I
97	TMS	I
98	TDO	O
99	TDI	I
100	SCL	
101	SDA	
102	PCI5V	I
103	VSS[GND]	–
104	HINT*	OD
105	HRST*	I
106	HGNT*	I
107	HREQ*	O
108	HAD[31]	I/O

Table 14-11. RS8235 Pinout (4 of 6)

Pin	Pin Label	I/O
109	HAD[30]	I/O
110	HAD[29]	I/O
111	HAD[28]	I/O
112	HAD[27]	I/O
113	HAD[26]	I/O
114	HAD[25]	I/O
115	HAD[24]	I/O
116	VSS[GND]	–
117	VDD[PWR]	–
118	HCBE[3]*	I/O
119	HIDSEL	I
120	HAD[23]	I/O
121	VDD[PWR]	–
122	HAD[22]	I/O
123	HAD[21]	I/O
124	HAD[20]	I/O
125	HAD[19]	I/O
126	HAD[18]	I/O
127	HAD[17]	I/O
128	HAD[16]	I/O
129	HCBE[2]*	I/O
130	VSS[GND]	–
131	HCLK	I
132	HFRAME*	I/O
133	HIRDY*	I/O
134	HTRDY*	I/O
135	HDEVSEL*	I/O
136	HSTOP*	I/O
137	HPERR*	I/O
138	HSERR*	OD
139	VSS[GND]	–
140	VDD[PWR]	–
141	HPAR	I/O
142	HCBE[1]*	I/O
143	HAD[15]	I/O
144	HAD[14]	I/O

Table 14-11. RS8235 Pinout (5 of 6)

Pin	Pin Label	I/O
145	HAD[13]	I/O
146	VDD[PWR]	–
147	HAD[12]	I/O
148	HAD[11]	I/O
149	HAD[10]	I/O
150	HAD[9]	I/O
151	HAD[8]	I/O
152	VSS[GND]	–
153	VDD[PWR]	–
154	HCBE[0]*	I/O
155	HAD[7]	I/O
156	HAD[6]	I/O
157	HAD[5]	I/O
158	HAD[4]	I/O
159	HAD[3]	I/O
160	HAD[2]	I/O
161	HAD[1]	I/O
162	HAD[0]	I/O
163	VSS[GND]	–
164	VSS[GND]	–
165	VDD[PWR]	–
166	PROCmode	I
167	PRST*	O
168	PFAIL*	I
169	PDAEN*	I/O
170	PCS*	I/O
171	PAS*	I/O
172	PWAIT*	I
173	PRDY*	O
174	PWNR	I/O
175	PBE[3]*	I
176	PBE[2]*	I
177	PBE[1]*	I
178	VSS[GND]	–
179	VDD[PWR]	–
180	PBE[0]*	I

Table 14-11. RS8235 Pinout (6 of 6)

Pin	Pin Label	I/O
181	PBSEL[1]	I
182	PBSEL[0]	I
183	PADDR[1]	I
184	PADDR[0]	I
185	LADDR[17]	I/O
186	LADDR[16]	I/O
187	LADDR[15]	I/O
188	LADDR[14]	I/O
189	VSS[GND]	–
190	VDD[PWR]	–
191	LADDR[13]	I/O
192	LADDR[12]	I/O
193	LADDR[11]	I/O
194	LADDR[10]	I/O
195	LADDR[9]	I/O
196	LADDR[8]	I/O
197	LADDR[7]	I/O
198	LADDR[6]	I/O
199	LADDR[5]	I/O
200	LADDR[4]	I/O
201	LADDR[3]	I/O
202	LADDR[2]	I/O
203	VSS[GND]	–
204	VDD[PWR]	–
205	LADDR[1]	I/O
206	LADDR[0]	I/O
207	VGG	I
208	STAT[1]	O

# Appendix A : Boundary Scan

The RS8235 supports boundary scan testing conforming to IEEE standard 1149.1-1990 and Supplement B, 1994. This appendix is intended to assist the customer in developing boundary scan tests for printed circuit boards and systems that use the RS8235. It is assumed that the reader is familiar with boundary scan terminology.

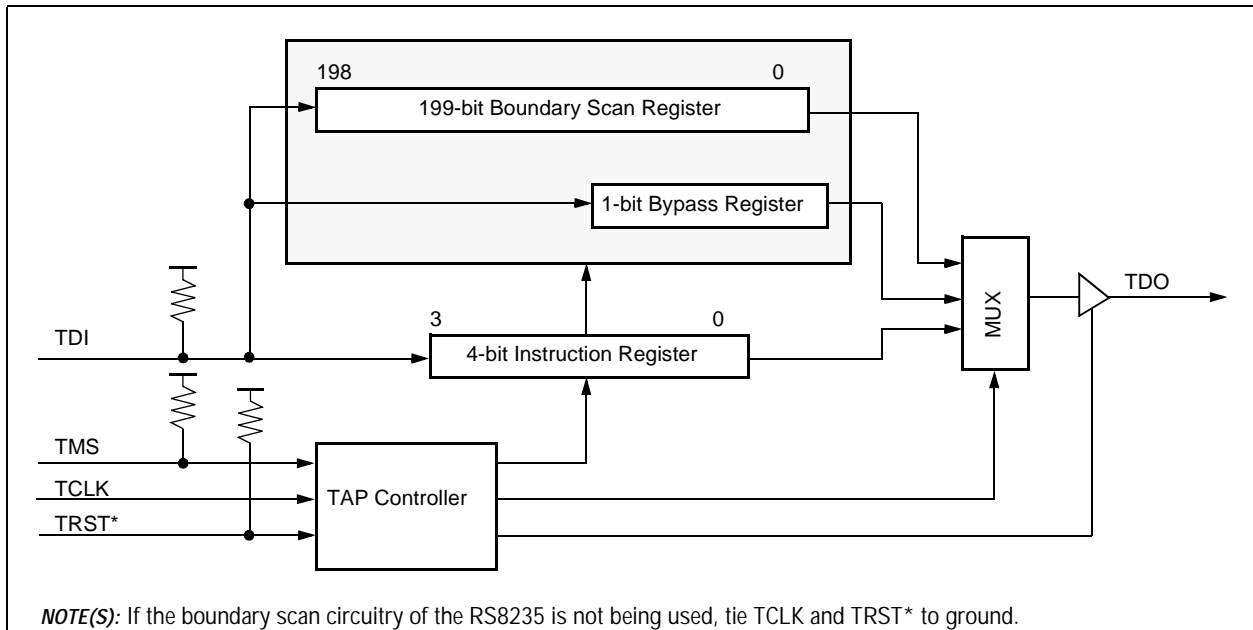
The Boundary Scan section of the RS8235 provides access to all external I/O signals of the device for board and system-level testing. This circuitry also conforms to IEEE Std 1149.1-1990. The boundary scan test logic is accessed through five dedicated pins on the RS8235 (see [Table A-1](#)).

*Table A-1. Boundary Scan Signals*

Pin Name	Signal Name	I/O	Definition
TRST*	Test Logic Reset	In	When at 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.
TCLK	Test Clock	In	Test clocking is generated externally by the system board or by the tester. TCLK can be stopped in either the high state or the low state.
TMS	Test Mode Select	In	Decoded to control test operations.
TDO	Serial Test Data Output	Out	Outputs serial test pattern data.
TDI	Serial Test Data Input	In	Input for serial test pattern data.

The test circuitry includes the Boundary Scan Register, a BYPASS Register, an Instruction Register, and the Tap Access Port (TAP) controller (Figure A-1).

Figure A-1. Test Circuitry Block Diagram



## A.1 Instruction Register

The Instruction Register (IR) is a 4-bit register with no parity. When the boundary scan circuitry is reset, the IR is loaded with the binary value b1111, which is equivalent to the BYPASS Instruction value. The Capture-IR binary value is b0001 and is shifted out TDO as the IR is loaded.

The 16 instructions displayed in [Table A-2](#) include three IEEE 1149.1 mandatory public instructions (BYPASS, EXTEST, and SAMPLE/PRELOAD) and 13 private instructions for manufacturing use only. Bit-0 (LSB) is shifted into the instruction register first.

**Table A-2. IEEE Std. 1149.1 Instructions**

Bit 3	Bit 2	Bit 1	Bit 0	Instruction	Register Accessed
0	0	0	0	EXTEST	Boundary Scan
0	0	0	1	RSTHIGH - Private	—
0	0	1	0	SAMPLE/PRELOAD	Boundary Scan
0	0	1	1	TMWAFIFO - Private	—
0	1	0	0	TMWBFIFO - Private	—
0	1	0	1	TMRFIFO - Private	—
0	1	1	0	TSEGFIFO - Private	—
0	1	1	1	TRSMFIFO - Private	—
1	0	0	0	T38AFIFO - Private	—
1	0	0	1	T38BFIFO - Private	—
1	0	1	0	RSVD0 - Private	—
1	0	1	1	RSVD1 - Private	—
1	1	0	0	RSVD2 - Private	Boundary Scan
1	1	0	1	RSVD3 - Private	Boundary Scan
1	1	1	0	PM_EN - Private	—
1	1	1	1	BYPASS	—

## A.2 BYPASS Register

The BYPASS Register is a 1-bit shift register used to pass TDI data to TDO to facilitate the testing of other devices in the scan path without having to shift the data patterns through the complete Boundary Scan Register of the RS8235.

## A.3 Boundary Scan Register

The Boundary Scan Register consists of two different types of registers corresponding to IEEE Std. 1149.1a-1993 attributes and definitions. These register names are STD\_1149\_1\_1993 Standard Boundary Cell names BC-1, and BC-7.

## A.4 Boundary Scan Register Cells

Table A-3 defines the Boundary Scan Register cells.

Cell 0 is closest to TDO in the chain.

These are the cell type definitions:

- Output3 = Output-three-state
- Input = Input-Observe
- Bidir = Reversible cell for bidirectional pin
- Control = Output-Control
- Controlr = Output-Control that is forced to its disable state in the Test-Logic-Reset controller state.
- Internal = Control-and-Observe internal cell, not associated with an I/O pin.

All controlling cells put their respective output cell into the inactive state with a value of one.

Table A-3. Boundary Scan Register Cells (1 of 6)

Cell	Related Pin Name	Cell Type	Controlling Cell
0	TXD[0]	output3	7
1	TXD[1]	output3	7
2	TXD[2]	output3	7
3	TXD[3]	output3	7
4	TXD[4]	output3	7
5	TXD[5]	output3	7
6	TXD[6]	output3	7
7	—	controlr	
8	TXD[7]	output3	7
9	TXPAR	output3	7
10	—	controlr	
11	TXFLAG_NEG	bidir	10
12	—	controlr	
13	TXMARK	bidir	12
14	—	controlr	
15	TXEN_NEG	bidir	14
16	FRCTRL	input	
17	FRCFG[1]	input	
18	FRCFG[0]	input	
19	—	controlr	
20	RXFLAG_NEG	bidir	19
21	—	controlr	
22	RXEN_NEG	bidir	21
23	RXMARK	input	
24	RXD[0]	input	
25	RXD[1]	input	
26	RXD[2]	input	
27	RXD[3]	input	
28	RXD[4]	input	
29	RXD[5]	input	
30	RXD[6]	input	
31	RXD[7]	input	
32	RXPAR	input	
33	—	internal	

Table A-3. Boundary Scan Register Cells (2 of 6)

Cell	Related Pin Name	Cell Type	Controlling Cell
34	LDATA[0]	bidir	41
35	LDATA[1]	bidir	41
36	LDATA[2]	bidir	41
37	LDATA[3]	bidir	41
38	LDATA[4]	bidir	41
39	LDATA[5]	bidir	41
40	LDATA[6]	bidir	41
41	—	controlr	
42	LDATA[7]	bidir	41
43	LDATA[8]	bidir	50
44	LDATA[9]	bidir	50
45	LDATA[10]	bidir	50
46	LDATA[11]	bidir	50
47	LDATA[12]	bidir	50
48	LDATA[13]	bidir	50
49	LDATA[14]	bidir	50
50	—	controlr	
51	LDATA[15]	bidir	50
52	LDATA[16]	bidir	59
53	LDATA[17]	bidir	59
54	LDATA[18]	bidir	59
55	LDATA[19]	bidir	59
56	LDATA[20]	bidir	59
57	LDATA[21]	bidir	59
58	LDATA[22]	bidir	59
59	—	controlr	
60	LDATA[23]	bidir	59
61	LDATA[24]	bidir	68
62	LDATA[25]	bidir	68
63	LDATA[26]	bidir	68
64	LDATA[27]	bidir	68
65	LDATA[28]	bidir	68
66	LDATA[29]	bidir	68
67	LDATA[30]	bidir	68
68	—	controlr	

Table A-3. Boundary Scan Register Cells (3 of 6)

Cell	Related Pin Name	Cell Type	Controlling Cell
69	LDATA[31]	bidir	68
70	CLK2X	input	
71	SYSCLK	output3	115
72	CLKD3	output3	115
73	—	internal	
74	MCS[1]_NEG	output3	115
75	—	internal	
76	—	internal	
77	MWR_NEG	output3	115
78	MOE_NEG	output3	115
79	MWE[0]_NEG	output3	115
80	MWE[1]_NEG	output3	115
81	MWE[2]_NEG	output3	115
82	MWE[3]_NEG	output3	115
83	RAMMODE	input	
84	STAT[0]	output3	115
85	STAT[1]	output3	115
86	LADDR[0]	output3	115
87	LADDR[1]	output3	115
88	LADDR[2]	bidir	104
89	LADDR[3]	bidir	104
90	LADDR[4]	bidir	104
91	LADDR[5]	bidir	104
92	LADDR[6]	bidir	104
93	LADDR[7]	bidir	104
94	LADDR[8]	bidir	104
95	LADDR[9]	bidir	104
96	LADDR[10]	bidir	104
97	LADDR[11]	bidir	104
98	LADDR[12]	bidir	104
99	LADDR[13]	bidir	104
100	LADDR[14]	bidir	104
101	LADDR[15]	bidir	104
102	LADDR[16]	bidir	104
103	LADDR[17]	bidir	104

Table A-3. Boundary Scan Register Cells (4 of 6)

Cell	Related Pin Name	Cell Type	Controlling Cell
104	—	controlr	
105	—	internal	
106	PADDR[0]	input	
107	PADDR[1]	input	
108	PBSEL[0]	input	
109	PBSEL[1]	input	
110	PBE[0]_NEG	input	
111	PBE[1]_NEG	input	
112	PBE[2]_NEG	input	
113	PBE[3]_NEG	input	
114	PWNR	bidir	120
115	—	controlr	
116	PRDY_NEG	output3	115
117	PWAIT_NEG	input	
118	—	internal	
119	PAS_NEG	bidir	120
120	—	controlr	
121	PCS_NEG	bidir	120
122	—	controlr	
123	PDAEN_NEG	bidir	122
124	PFAIL_NEG	input	
125	—	internal	
126	—	internal	
127	PRST_NEG	output3	115
128	PROCMODE	input	
129	HAD[0]	bidir	136
130	HAD[1]	bidir	136
131	HAD[2]	bidir	136
132	HAD[3]	bidir	136
133	HAD[4]	bidir	136
134	HAD[5]	bidir	136
135	HAD[6]	bidir	136
136	—	controlr	
137	HAD[7]	bidir	136
138	HCBE[0]_NEG	bidir	177

Table A-3. Boundary Scan Register Cells (5 of 6)

Cell	Related Pin Name	Cell Type	Controlling Cell
139	HAD[8]	bidir	146
140	HAD[9]	bidir	146
141	HAD[10]	bidir	146
142	HAD[11]	bidir	146
143	HAD[12]	bidir	146
144	HAD[13]	bidir	146
145	HAD[14]	bidir	146
146	—	controlr	
147	HAD[15]	bidir	146
148	HCBE[1]_NEG	bidir	177
149	—	controlr	
150	HPAR	bidir	149
151	—	controlr	
152	HSERR_NEG	output3	151
153	—	controlr	
154	HPERR_NEG	bidir	153
155	—	controlr	
156	HSTOP_NEG	bidir	155
157	—	controlr	
158	HDEVSEL_NEG	bidir	157
159	—	controlr	
160	HTRDY_NEG	bidir	159
161	—	controlr	
162	HIRDY_NEG	bidir	161
163	—	controlr	
164	HFRAME_NEG	bidir	163
165	HCLK	input	
166	HCBE[2]_NEG	bidir	177
167	HAD[16]	bidir	174
168	HAD[17]	bidir	174
169	HAD[18]	bidir	174
170	HAD[19]	bidir	174
171	HAD[20]	bidir	174
172	HAD[21]	bidir	174
173	HAD[22]	bidir	174

Table A-3. Boundary Scan Register Cells (6 of 6)

Cell	Related Pin Name	Cell Type	Controlling Cell
174	—	controlr	
175	HAD[23]	bidir	174
176	HIDSEL	input	
177	—	controlr	
178	HCBE[3]_NEG	bidir	177
17	HAD[24]	bidir	186
180	HAD[25]	bidir	186
181	HAD[26]	bidir	186
182	HAD[27]	bidir	186
183	HAD[28]	bidir	186
184	HAD[29]	bidir	186
185	HAD[30]	bidir	186
186	—	controlr	
187	HAD[31]	bidir	186
188	—	controlr	
189	HREQ_NEG	output3	188
190	HGNT_NEG	input	
191	HRST_NEG	input	
192	—	controlr	
193	HINT_NEG	output3	192
194	PCI5V	input	
195	—	controlr	
196	SDA	bidir	195
197	—	controlr	
198	SCL	bidir	197

## A.5 Electrical Characteristics

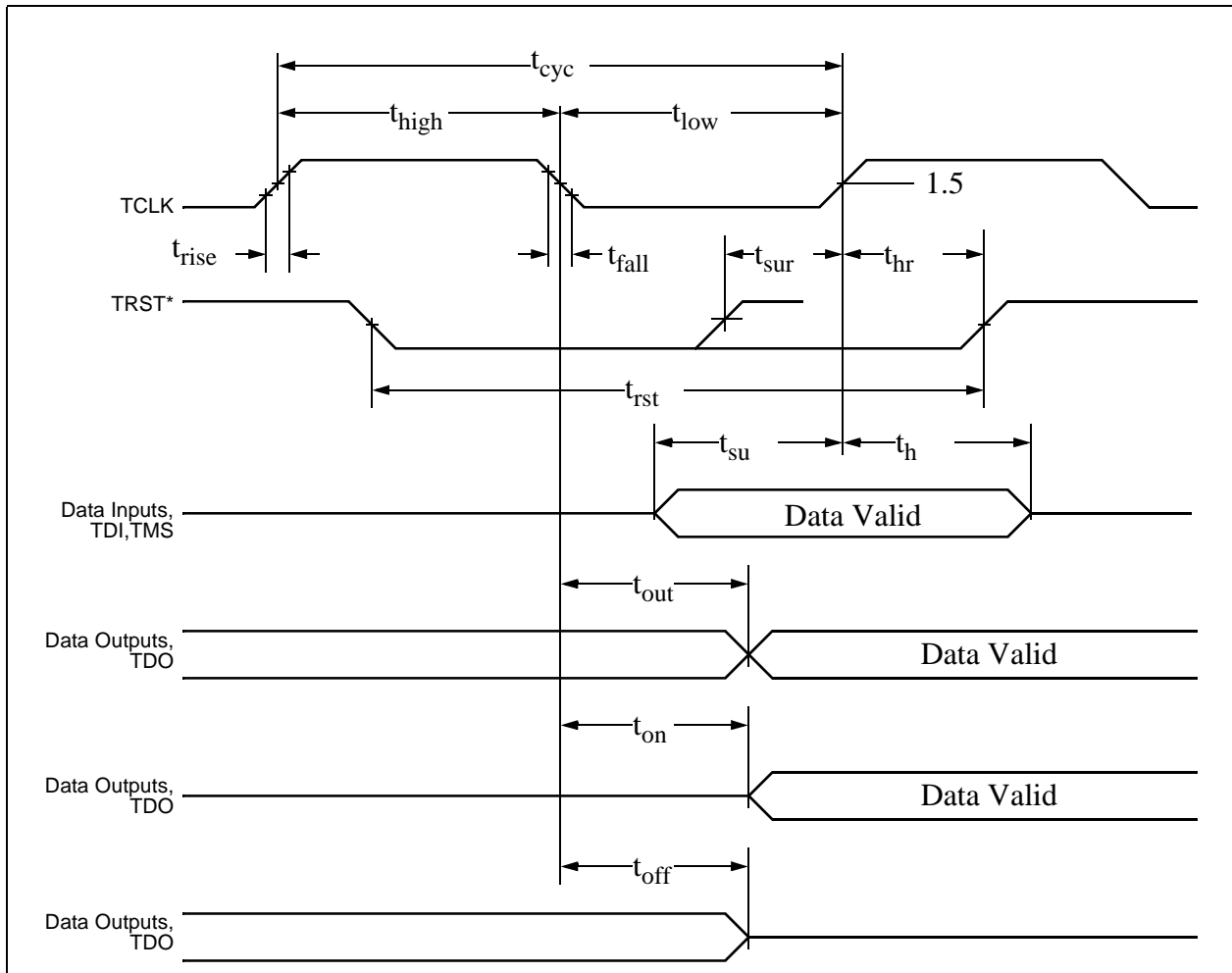
This section describes the electrical characteristics for boundary scan. [Table A-4](#) provides timing specifications and [Figure A-2](#) shows a timing diagram.

**Table A-4. Timing Specifications**

Symbol	Description	Min	Max	Unit
	TCLK Frequency	TBD	TBD	MHz
$t_{cyc}$	TCLK Cycle	TBD	TBD	ns
$t_{high}$	TCLK High Pulse Width	TBD	TBD	ns
$t_{low}$	TCLK Low Pulse Width	TBD	TBD	ns
$t_{rise}$	TCLK Rise	TBD	TBD	ns
$t_{fall}$	TCLK Fall	TBD	TBD	ns
$t_{sur}$	TRST* Setup to TCLK Rising Edge <sup>(1)</sup>	TBD	TBD	ns
$t_{hr}$	TRST* Hold after TCLK Rising Edge <sup>(1)</sup>	TBD	TBD	ns
$t_{rst}$	TRST* Active	1 $t_{cyc}$	-	
$t_{su}$	TDI, TMS, and Data Inputs Setup	TBD	TBD	ns
$t_h$	TDI, TMS, and Data Inputs Hold	TBD	TBD	ns
$t_{out}$	TDO and Data Outputs Valid	TBD	TBD	ns
$t_{on}$	TDO and Data Outputs Float to Valid	TBD	TBD	ns
$t_{off}$	TDO and Data Outputs Valid to Float	TBD	TBD	ns

<sup>(1)</sup> TRST\* going "inactive" timing only required if TMS = 0 at the rising edge of TCLK.

Figure A-2. Timing Diagram For Boundary Scan



## A.6 Boundary Scan Description Language (BSDL) File

To obtain an electronic copy of this file, go to the Mindspeed web site at [www.mindspeed.com](http://www.mindspeed.com).

--

-- BSDL File created/edited by AT&T BSD Editor Version 3.0

--

--BSDE:Revision: 8235 version 1.0

--BSDE:Description: ATM Segmentation and Reassembly Controller - SAR

entity RS8235 is

generic (PHYSICAL\_PIN\_MAP: string: = "PQFP\_208" );

port (

VSS: linkage bit;

VDD: linkage bit;

VGG: linkage bit;

STAT1: out bit;

STAT0: out bit;

RAMMODE: in bit;

MWE3\_NEG: out bit;

MWE2\_NEG: out bit;

MWE1\_NEG: out bit;

MWE0\_NEG: out bit;

MOE\_NEG: out bit;

MWR\_NEG: out bit;

MCS1\_NEG: out bit;

SPARE: linkage bit;

CLKD3: out bit;

SYSCLK: out bit;

CLK2X: in bit;

LDATA31: inout bit;

LDATA30: inout bit;

LDATA29: inout bit;

LDATA28: inout bit;

LDATA27: inout bit;

LDATA26: inout bit;

LDATA25: inout bit;

LDATA24: inout bit;

LDATA23: inout bit;

LDATA22: inout bit;

LDATA21: inout bit;

LDATA20: inout bit;  
LDATA19: inout bit;  
LDATA18: inout bit;  
LDATA17: inout bit;  
LDATA16: inout bit;  
LDATA15: inout bit;  
LDATA14: inout bit;  
LDATA13: inout bit;  
LDATA12: inout bit;  
LDATA11: inout bit;  
LDATA10: inout bit;  
LDATA9: inout bit;  
LDATA8: inout bit;  
LDATA7: inout bit;  
LDATA6: inout bit;  
LDATA5: inout bit;  
LDATA4: inout bit;  
LDATA3: inout bit;  
LDATA2: inout bit;  
LDATA1: inout bit;  
LDATA0: inout bit;  
RXPAR: in bit;  
RXD7: in bit;  
RXD6: in bit;  
RXD5: in bit;  
RXD4: in bit;  
RXD3: in bit;  
RXD2: in bit;  
RXD1: in bit;  
RXD0: in bit;  
RXMARK: in bit;  
RXEN\_NEG: inout bit;  
RXFLAG\_NEG: inout bit;  
FRCFG0: in bit;  
FRCFG1: in bit;  
FRCTRL: in bit;  
TXEN\_NEG: inout bit;  
TXMARK: inout bit;  
TXFLAG\_NEG: inout bit;  
TXPAR: out bit;  
TXD7: out bit;  
TXD6: out bit;  
TXD5: out bit;  
TXD4: out bit;  
TXD3: out bit;  
TXD2: out bit;

TXD1: out bit;  
TXD0: out bit;  
TCLK: in bit;  
TRST\_NEG: in bit;  
TMS: in bit;  
TDO: out bit;  
TDI: in bit;  
SCL: inout bit;  
SDA: inout bit;  
PCI5V: in bit;  
HINT\_NEG: out bit;  
HRST\_NEG: in bit;  
HGNT\_NEG: in bit;  
HREQ\_NEG: out bit;  
HAD31: inout bit;  
HAD30: inout bit;  
HAD29: inout bit;  
HAD28: inout bit;  
HAD27: inout bit;  
HAD26: inout bit;  
HAD25: inout bit;  
HAD24: inout bit;  
HCBE3\_NEG: inout bit;  
HIDSEL: in bit;  
HAD23: inout bit;  
HAD22: inout bit;  
HAD21: inout bit;  
HAD20: inout bit;  
HAD19: inout bit;  
HAD18: inout bit;  
HAD17: inout bit;  
HAD16: inout bit;  
HCBE2\_NEG: inout bit;  
HCLK: in bit;  
HFRAME\_NEG: inout bit;  
HIRDY\_NEG: inout bit;  
HTRDY\_NEG: inout bit;  
HDEVSEL\_NEG: inout bit;  
HSTOP\_NEG: inout bit;  
HPERR\_NEG: inout bit;  
HSERR\_NEG: out bit;  
HPAR: inout bit;  
HCBE1\_NEG: inout bit;  
HAD15: inout bit;  
HAD14: inout bit;  
HAD13: inout bit;

HAD12: inout bit;  
HAD11: inout bit;  
HAD10: inout bit;  
HAD9: inout bit;  
HAD8: inout bit;  
HCBE0\_NEG: inout bit;  
HAD7: inout bit;  
HAD6: inout bit;  
HAD5: inout bit;  
HAD4: inout bit;  
HAD3: inout bit;  
HAD2: inout bit;  
HAD1: inout bit;  
HAD0: inout bit;  
PROCMODE: in bit;  
PRST\_NEG: out bit;  
PFAIL\_NEG: in bit;  
PDAEN\_NEG: inout bit;  
PCS\_NEG: inout bit;  
PAS\_NEG: inout bit;  
PWAIT\_NEG: in bit;  
PRDY\_NEG: out bit;  
PWNR: inout bit;  
PBE3\_NEG: in bit;  
PBE2\_NEG: in bit;  
PBE1\_NEG: in bit;  
PBE0\_NEG: in bit;  
PBSEL1: in bit;  
PBSEL0: in bit;  
PADDR1: in bit;  
PADDR0: in bit;  
LADDR17: inout bit;  
LADDR16: inout bit;  
LADDR15: inout bit;  
LADDR14: inout bit;  
LADDR13: inout bit;  
LADDR12: inout bit;  
LADDR11: inout bit;  
LADDR10: inout bit;  
LADDR9: inout bit;  
LADDR8: inout bit;  
LADDR7: inout bit;  
LADDR6: inout bit;  
LADDR5: inout bit;  
LADDR4: inout bit;  
LADDR3: inout bit;

```
LADDR2: inout bit;  
LADDR1: out bit;  
LADDR0: out bit  
);
```

```
use STD_1149_1_1994.all;
```

```
attribute COMPONENT_CONFORMANCE of RS8235 :  
entity is "STD_1149_1_1993";
```

```
attribute PIN_MAP of RS8235 : entity is PHYSICAL_PIN_MAP;
```

```
constant PQFP_208: PIN_MAP_STRING:=
```

```
"VSS:5," &  
"VDD:6," &  
"VGG:207," &  
"STAT1:208," &  
"STAT0:1," &  
"RAMMODE:2," &  
"MWE3_NEG:3," &  
"MWE2_NEG:4," &  
"MWE1_NEG:7," &  
"MWE0_NEG:8," &  
"MOE_NEG:9," &  
"MWR_NEG:10," &  
"MCS1_NEG:11," &  
"SPARE:12," &  
"CLKD3:14," &  
"SYSCLK:17," &  
"CLK2X:19," &  
"LDATA31:21," &  
"LDATA30:22," &  
"LDATA29:23," &  
"LDATA28:24," &  
"LDATA27:25," &  
"LDATA26:26," &  
"LDATA25:27," &  
"LDATA24:28," &  
"LDATA23:31," &  
"LDATA22:32," &  
"LDATA21:33," &  
"LDATA20:34," &  
"LDATA19:35," &  
"LDATA18:36," &  
"LDATA17:38," &  
"LDATA16:39," &
```

"LDATA15:40," &  
"LDATA14:41," &  
"LDATA13:42," &  
"LDATA12:43," &  
"LDATA11:44," &  
"LDATA10:45," &  
"LDATA9:46," &  
"LDATA8:47," &  
"LDATA7:48," &  
"LDATA6:49," &  
"LDATA5:50," &  
"LDATA4:51," &  
"LDATA3:53," &  
"LDATA2:54," &  
"LDATA1:55," &  
"LDATA0:56," &  
"RXPAR:59," &  
"RXD7:60," &  
"RXD6:61," &  
"RXD5:62," &  
"RXD4:63," &  
"RXD3:64," &  
"RXD2:65," &  
"RXD1:66," &  
"RXD0:69," &  
"RXMARK:70," &  
"RXEN\_NEG:71," &  
"RXFLAG\_NEG:72," &  
"FRCFG0:73," &  
"FRCFG1:74," &  
"FRCTRL:75," &  
"TXEN\_NEG:78," &  
"TXMARK:79," &  
"TXFLAG\_NEG:80," &  
"TXPAR:81," &  
"TXD7:82," &  
"TXD6:83," &  
"TXD5:86," &  
"TXD4:87," &  
"TXD3:88," &  
"TXD2:89," &  
"TXD1:90," &  
"TXD0:91," &  
"TCLK:94," &  
"TRST\_NEG:96," &  
"TMS:97," &

"TDO:98," &  
"TDI:99," &  
"SCL:100," &  
"SDA:101," &  
"PCI5V:102," &  
"HINT\_NEG:104," &  
"HRST\_NEG:105," &  
"HGNT\_NEG:106," &  
"HREQ\_NEG:107," &  
"HAD31:108," &  
"HAD30:109," &  
"HAD29:110," &  
"HAD28:111," &  
"HAD27:112," &  
"HAD26:113," &  
"HAD25:114," &  
"HAD24:115," &  
"HCBE3\_NEG:118," &  
"HIDSEL:119," &  
"HAD23:120," &  
"HAD22:122," &  
"HAD21:123," &  
"HAD20:124," &  
"HAD19:125," &  
"HAD18:126," &  
"HAD17:127," &  
"HAD16:128," &  
"HCBE2\_NEG:129," &  
"HCLK:131," &  
"HFRAME\_NEG:132," &  
"HIRDY\_NEG:133," &  
"HTRDY\_NEG:134," &  
"HDEVSEL\_NEG:135," &  
"HSTOP\_NEG:136," &  
"HPERR\_NEG:137," &  
"HSERR\_NEG:138," &  
"HPAR:141," &  
"HCBE1\_NEG:142," &  
"HAD15:143," &  
"HAD14:144," &  
"HAD13:145," &  
"HAD12:147," &  
"HAD11:148," &  
"HAD10:149," &  
"HAD9:150," &  
"HAD8:151," &

"HCBE0\_NEG:154," &  
"HAD7:155," &  
"HAD6:156," &  
"HAD5:157," &  
"HAD4:158," &  
"HAD3:159," &  
"HAD2:160," &  
"HAD1:161," &  
"HAD0:162," &  
"PROCMODE:166," &  
"PRST\_NEG:167," &  
"PFAIL\_NEG:168," &  
"PDAEN\_NEG:169," &  
"PCS\_NEG:170," &  
"PAS\_NEG:171," &  
"PWAIT\_NEG:172," &  
"PRDY\_NEG:173," &  
"PWNR:174," &  
"PBE3\_NEG:175," &  
"PBE2\_NEG:176," &  
"PBE1\_NEG:177," &  
"PBE0\_NEG:180," &  
"PBSEL1:181," &  
"PBSEL0:182," &  
"PADDR1:183," &  
"PADDR0:184," &  
"LADDR17:185," &  
"LADDR16:186," &  
"LADDR15:187," &  
"LADDR14:188," &  
"LADDR13:191," &  
"LADDR12:192," &  
"LADDR11:193," &  
"LADDR10:194," &  
"LADDR9:195," &  
"LADDR8:196," &  
"LADDR7:197," &  
"LADDR6:198," &  
"LADDR5:199," &  
"LADDR4:200," &  
"LADDR3:201," &  
"LADDR2:202," &  
"LADDR1:205," &  
"LADDR0:206";

attribute TAP\_SCAN\_IN of TDI : signal is true;  
 attribute TAP\_SCAN\_OUT of TDO : signal is true;  
 attribute TAP\_SCAN\_MODE of TMS : signal is true;  
 attribute TAP\_SCAN\_CLOCK of TCLK : signal is (2.00e+07, BOTH);  
 attribute TAP\_SCAN\_RESET of TRST\_NEG : signal is true;

attribute INSTRUCTION\_LENGTH of RS8235 : entity is 4;

attribute INSTRUCTION\_OPCODE of RS8235 : entity is

"EXTEST (0000)," &  
 "RSTHIGH (0001)," &  
 "SAMPLE (0010)," &  
 "TMWAFIFO (0011)," &  
 "TMWBFIFO (0100)," &  
 "TMRFIFO (0101)," &  
 "TSEGFIFO (0110)," &  
 "TRSMFIFO (0111)," &  
 "T38AFIFO (1000)," &  
 "T38BFIFO (1001)," &  
 "RSVD0 (1010)," &  
 "RSVD1 (1011)," &  
 "RSVD2 (1100)," &  
 "RSVD3 (1101)," &  
 "PM\_EN (1110)," &  
 "BYPASS (1111)";

attribute INSTRUCTION\_CAPTURE of RS8235 : entity is "0001";

attribute INSTRUCTION\_PRIVATE of RS8235 : entity is

" RSTHIGH, TMWAFIFO, TMWBFIFO, TMRFIFO, TSEGFIFO, TRSMFIFO, T38AFIFO," &  
 " T38BFIFO, RSVD0, RSVD1, RSVD2, RSVD3, PM\_EN";

attribute REGISTER\_ACCESS of RS8235 : entity is

"BYPASS ( BYPASS, RSTHIGH, TMWAFIFO, TMWBFIFO, TMRFIFO, TSEGFIFO,  
 TRSMFIFO, T38AFIFO, T38BFIFO, RSVD0, RSVD1, PM\_EN)," &  
 "BOUNDARY ( EXTEST, SAMPLE, RSVD2, RSVD3)";

attribute BOUNDARY\_LENGTH of RS8235 : entity is 199;

attribute BOUNDARY\_REGISTER of RS8235 : entity is

" 0 (BC\_1, TXD0, output3, X, 7, 1, Z)," &  
 " 1 (BC\_1, TXD1, output3, X, 7, 1, Z)," &  
 " 2 (BC\_1, TXD2, output3, X, 7, 1, Z)," &  
 " 3 (BC\_1, TXD3, output3, X, 7, 1, Z)," &  
 " 4 (BC\_1, TXD4, output3, X, 7, 1, Z)," &

```

" 5 (BC_1, TXD5, output3, X, 7, 1, Z)," &
" 6 (BC_1, TXD6, output3, X, 7, 1, Z)," &
" 7 (BC_1, *, controlr, 1)," &
" 8 (BC_1, TXD7, output3, X, 7, 1, Z)," &
" 9 (BC_1, TXPAR, output3, X, 7, 1, Z)," &
" 10 (BC_1, *, controlr, 1)," &
" 11 (BC_7, TXFLAG_NEG, bidir, X, 10, 1, Z)," &
" 12 (BC_1, *, controlr, 1)," &
" 13 (BC_7, TXMARK, bidir, X, 12, 1, Z)," &
" 14 (BC_1, *, controlr, 1)," &
" 15 (BC_7, TXEN_NEG, bidir, X, 14, 1, Z)," &
" 16 (BC_1, FRCTRL, input, X)," &
" 17 (BC_1, FRCFG1, input, X)," &
" 18 (BC_1, FRCFG0, input, X)," &
" 19 (BC_1, *, controlr, 1)," &
" 20 (BC_7, RXFLAG_NEG, bidir, X, 19, 1, Z)," &
" 21 (BC_1, *, controlr, 1)," &
" 22 (BC_7, RXEN_NEG, bidir, X, 21, 1, Z)," &
" 23 (BC_1, RXMARK, input, X)," &
" 24 (BC_1, RXD0, input, X)," &
" 25 (BC_1, RXD1, input, X)," &
" 26 (BC_1, RXD2, input, X)," &
" 27 (BC_1, RXD3, input, X)," &
" 28 (BC_1, RXD4, input, X)," &
" 29 (BC_1, RXD5, input, X)," &
" 30 (BC_1, RXD6, input, X)," &
" 31 (BC_1, RXD7, input, X)," &
" 32 (BC_1, RXPARG, input, X)," &
" 33 (BC_1, *, internal, 1)," &
" 34 (BC_7, LDATA0, bidir, X, 41, 1, Z)," &
" 35 (BC_7, LDATA1, bidir, X, 41, 1, Z)," &
" 36 (BC_7, LDATA2, bidir, X, 41, 1, Z)," &
" 37 (BC_7, LDATA3, bidir, X, 41, 1, Z)," &
" 38 (BC_7, LDATA4, bidir, X, 41, 1, Z)," &
" 39 (BC_7, LDATA5, bidir, X, 41, 1, Z)," &
" 40 (BC_7, LDATA6, bidir, X, 41, 1, Z)," &
" 41 (BC_1, *, controlr, 1)," &
" 42 (BC_7, LDATA7, bidir, X, 41, 1, Z)," &
" 43 (BC_7, LDATA8, bidir, X, 50, 1, Z)," &
" 44 (BC_7, LDATA9, bidir, X, 50, 1, Z)," &
" 45 (BC_7, LDATA10, bidir, X, 50, 1, Z)," &
" 46 (BC_7, LDATA11, bidir, X, 50, 1, Z)," &
" 47 (BC_7, LDATA12, bidir, X, 50, 1, Z)," &
" 48 (BC_7, LDATA13, bidir, X, 50, 1, Z)," &
" 49 (BC_7, LDATA14, bidir, X, 50, 1, Z)," &
" 50 (BC_1, *, controlr, 1)," &

```

```

" 51 (BC_7, LDATA15, bidir, X, 50, 1, Z)," &
" 52 (BC_7, LDATA16, bidir, X, 59, 1, Z)," &
" 53 (BC_7, LDATA17, bidir, X, 59, 1, Z)," &
" 54 (BC_7, LDATA18, bidir, X, 59, 1, Z)," &
" 55 (BC_7, LDATA19, bidir, X, 59, 1, Z)," &
" 56 (BC_7, LDATA20, bidir, X, 59, 1, Z)," &
" 57 (BC_7, LDATA21, bidir, X, 59, 1, Z)," &
" 58 (BC_7, LDATA22, bidir, X, 59, 1, Z)," &
" 59 (BC_1, *, controlr, 1)," &
" 60 (BC_7, LDATA23, bidir, X, 59, 1, Z)," &
" 61 (BC_7, LDATA24, bidir, X, 68, 1, Z)," &
" 62 (BC_7, LDATA25, bidir, X, 68, 1, Z)," &
" 63 (BC_7, LDATA26, bidir, X, 68, 1, Z)," &
" 64 (BC_7, LDATA27, bidir, X, 68, 1, Z)," &
" 65 (BC_7, LDATA28, bidir, X, 68, 1, Z)," &
" 66 (BC_7, LDATA29, bidir, X, 68, 1, Z)," &
" 67 (BC_7, LDATA30, bidir, X, 68, 1, Z)," &
" 68 (BC_1, *, controlr, 1)," &
" 69 (BC_7, LDATA31, bidir, X, 68, 1, Z)," &
" 70 (BC_1, CLK2X, input, X)," &
" 71 (BC_1, SYSCLK, output3, X, 115, 1, Z)," &
" 72 (BC_1, CLKD3, output3, X, 115, 1, Z)," &
" 73 (BC_1, *, internal, 1)," &
" 74 (BC_1, MCS1_NEG, output3, X, 115, 1, Z)," &
" 75 (BC_1, *, internal, 1)," &
" 76 (BC_1, *, internal, 1)," &
" 77 (BC_1, MWR_NEG, output3, X, 115, 1, Z)," &
" 78 (BC_1, MOE_NEG, output3, X, 115, 1, Z)," &
" 79 (BC_1, MWE0_NEG, output3, X, 115, 1, Z)," &
" 80 (BC_1, MWE1_NEG, output3, X, 115, 1, Z)," &
" 81 (BC_1, MWE2_NEG, output3, X, 115, 1, Z)," &
" 82 (BC_1, MWE3_NEG, output3, X, 115, 1, Z)," &
" 83 (BC_1, RAMMODE, input, X)," &
" 84 (BC_1, STAT0, output3, X, 115, 1, Z)," &
" 85 (BC_1, STAT1, output3, X, 115, 1, Z)," &
" 86 (BC_1, LADDR0, output3, X, 115, 1, Z)," &
" 87 (BC_1, LADDR1, output3, X, 115, 1, Z)," &
" 88 (BC_7, LADDR2, bidir, X, 104, 1, Z)," &
" 89 (BC_7, LADDR3, bidir, X, 104, 1, Z)," &
" 90 (BC_7, LADDR4, bidir, X, 104, 1, Z)," &
" 91 (BC_7, LADDR5, bidir, X, 104, 1, Z)," &
" 92 (BC_7, LADDR6, bidir, X, 104, 1, Z)," &
" 93 (BC_7, LADDR7, bidir, X, 104, 1, Z)," &
" 94 (BC_7, LADDR8, bidir, X, 104, 1, Z)," &
" 95 (BC_7, LADDR9, bidir, X, 104, 1, Z)," &
" 96 (BC_7, LADDR10, bidir, X, 104, 1, Z)," &

```

```

" 97 (BC_7, LADDR11, bidir, X, 104, 1, Z)," &
" 98 (BC_7, LADDR12, bidir, X, 104, 1, Z)," &
" 99 (BC_7, LADDR13, bidir, X, 104, 1, Z)," &
" 100 (BC_7, LADDR14, bidir, X, 104, 1, Z)," &
" 101 (BC_7, LADDR15, bidir, X, 104, 1, Z)," &
" 102 (BC_7, LADDR16, bidir, X, 104, 1, Z)," &
" 103 (BC_7, LADDR17, bidir, X, 104, 1, Z)," &
" 104 (BC_1, *, controlr, 1)," &
" 105 (BC_1, *, internal, 1)," &
" 106 (BC_1, PADDR0, input, X)," &
" 107 (BC_1, PADDR1, input, X)," &
" 108 (BC_1, PBSEL0, input, X)," &
" 109 (BC_1, PBSEL1, input, X)," &
" 110 (BC_1, PBE0_NEG, input, X)," &
" 111 (BC_1, PBE1_NEG, input, X)," &
" 112 (BC_1, PBE2_NEG, input, X)," &
" 113 (BC_1, PBE3_NEG, input, X)," &
" 114 (BC_7, PWNR, bidir, X, 120, 1, Z)," &
" 115 (BC_1, *, controlr, 1)," &
" 116 (BC_1, PRDY_NEG, output3, X, 115, 1, Z)," &
" 117 (BC_1, PWAIT_NEG, input, X)," &
" 118 (BC_1, *, internal, 1)," &
" 119 (BC_7, PAS_NEG, bidir, X, 120, 1, Z)," &
" 120 (BC_1, *, controlr, 1)," &
" 121 (BC_7, PCS_NEG, bidir, X, 120, 1, Z)," &
" 122 (BC_1, *, controlr, 1)," &
" 123 (BC_7, PDAEN_NEG, bidir, X, 122, 1, Z)," &
" 124 (BC_1, PFAIL_NEG, input, X)," &
" 125 (BC_1, *, internal, 1)," &
" 126 (BC_1, *, internal, 1)," &
" 127 (BC_1, PRST_NEG, output3, X, 115, 1, Z)," &
" 128 (BC_1, PROCMODE, input, X)," &
" 129 (BC_7, HAD0, bidir, X, 136, 1, Z)," &
" 130 (BC_7, HAD1, bidir, X, 136, 1, Z)," &
" 131 (BC_7, HAD2, bidir, X, 136, 1, Z)," &
" 132 (BC_7, HAD3, bidir, X, 136, 1, Z)," &
" 133 (BC_7, HAD4, bidir, X, 136, 1, Z)," &
" 134 (BC_7, HAD5, bidir, X, 136, 1, Z)," &
" 135 (BC_7, HAD6, bidir, X, 136, 1, Z)," &
" 136 (BC_1, *, controlr, 1)," &
" 137 (BC_7, HAD7, bidir, X, 136, 1, Z)," &
" 138 (BC_7, HCBE0_NEG, bidir, X, 177, 1, Z)," &
" 139 (BC_7, HAD8, bidir, X, 146, 1, Z)," &
" 140 (BC_7, HAD9, bidir, X, 146, 1, Z)," &
" 141 (BC_7, HAD10, bidir, X, 146, 1, Z)," &
" 142 (BC_7, HAD11, bidir, X, 146, 1, Z)," &

```

" 143 (BC\_7, HAD12, bidir, X, 146, 1, Z)," &  
" 144 (BC\_7, HAD13, bidir, X, 146, 1, Z)," &  
" 145 (BC\_7, HAD14, bidir, X, 146, 1, Z)," &  
" 146 (BC\_1, \*, controlr, 1)," &  
" 147 (BC\_7, HAD15, bidir, X, 146, 1, Z)," &  
" 148 (BC\_7, HCBE1\_NEG, bidir, X, 177, 1, Z)," &  
" 149 (BC\_1, \*, controlr, 1)," &  
" 150 (BC\_7, HPAR, bidir, X, 149, 1, Z)," &  
" 151 (BC\_1, \*, controlr, 1)," &  
" 152 (BC\_1, HSERR\_NEG, output3, X, 151, 1, Z)," &  
" 153 (BC\_1, \*, controlr, 1)," &  
" 154 (BC\_7, HPERR\_NEG, bidir, X, 153, 1, Z)," &  
" 155 (BC\_1, \*, controlr, 1)," &  
" 156 (BC\_7, HSTOP\_NEG, bidir, X, 155, 1, Z)," &  
" 157 (BC\_1, \*, controlr, 1)," &  
" 158 (BC\_7, HDEVSEL\_NEG, bidir, X, 157, 1, Z)," &  
" 159 (BC\_1, \*, controlr, 1)," &  
" 160 (BC\_7, HTRDY\_NEG, bidir, X, 159, 1, Z)," &  
" 161 (BC\_1, \*, controlr, 1)," &  
" 162 (BC\_7, HIRDY\_NEG, bidir, X, 161, 1, Z)," &  
" 163 (BC\_1, \*, controlr, 1)," &  
" 164 (BC\_7, HFRAME\_NEG, bidir, X, 163, 1, Z)," &  
" 165 (BC\_1, HCLK, input, X)," &  
" 166 (BC\_7, HCBE2\_NEG, bidir, X, 177, 1, Z)," &  
" 167 (BC\_7, HAD16, bidir, X, 174, 1, Z)," &  
" 168 (BC\_7, HAD17, bidir, X, 174, 1, Z)," &  
" 169 (BC\_7, HAD18, bidir, X, 174, 1, Z)," &  
" 170 (BC\_7, HAD19, bidir, X, 174, 1, Z)," &  
" 171 (BC\_7, HAD20, bidir, X, 174, 1, Z)," &  
" 172 (BC\_7, HAD21, bidir, X, 174, 1, Z)," &  
" 173 (BC\_7, HAD22, bidir, X, 174, 1, Z)," &  
" 174 (BC\_1, \*, controlr, 1)," &  
" 175 (BC\_7, HAD23, bidir, X, 174, 1, Z)," &  
" 176 (BC\_1, HIDSEL, input, X)," &  
" 177 (BC\_1, \*, controlr, 1)," &  
" 178 (BC\_7, HCBE3\_NEG, bidir, X, 177, 1, Z)," &  
" 179 (BC\_7, HAD24, bidir, X, 186, 1, Z)," &  
" 180 (BC\_7, HAD25, bidir, X, 186, 1, Z)," &  
" 181 (BC\_7, HAD26, bidir, X, 186, 1, Z)," &  
" 182 (BC\_7, HAD27, bidir, X, 186, 1, Z)," &  
" 183 (BC\_7, HAD28, bidir, X, 186, 1, Z)," &  
" 184 (BC\_7, HAD29, bidir, X, 186, 1, Z)," &  
" 185 (BC\_7, HAD30, bidir, X, 186, 1, Z)," &  
" 186 (BC\_1, \*, controlr, 1)," &  
" 187 (BC\_7, HAD31, bidir, X, 186, 1, Z)," &  
" 188 (BC\_1, \*, controlr, 1)," &

```
" 189 (BC_1, HREQ_NEG, output3, X, 188, 1, Z)," &  
" 190 (BC_1, HGNT_NEG, input, X)," &  
" 191 (BC_1, HRST_NEG, input, X)," &  
" 192 (BC_1, *, controlr, 1)," &  
" 193 (BC_1, HINT_NEG, output3, X, 192, 1, Z)," &  
" 194 (BC_1, PCI5V, input, X)," &  
" 195 (BC_1, *, controlr, 1)," &  
" 196 (BC_7, SDA, bidir, X, 195, 1, Z)," &  
" 197 (BC_1, *, controlr, 1)," &  
" 198 (BC_7, SCL, bidir, X, 197, 1, Z)";
```

```
end RS8235;
```



**MINDSPEED™**

**[www.mindspeed.com](http://www.mindspeed.com)**

General Information:  
U.S. and Canada: (800) 854-8099  
International: (949) 483-6996  
Headquarters - Newport Beach  
4311 Jamboree Rd. P.O. Box C  
Newport Beach, CA. 92658-8902