

# CN8236

## ATM ServiceSAR Plus with xBR Traffic Management

The CN8236 Service Segmentation and Reassembly Controller integrates ATM terminal functions, PCI Bus Master and Slave controllers, and a UTOPIA level 1 or 2 interface with service-specific functions in a single package for AAL0, 3/4, and 5 operations. The ServiceSAR Controller generates and terminates ATM traffic and automatically schedules cells for transmission. The CN8236 is targeted at 155 Mbps throughput systems where the number of VCCs is relatively large, or the performance of the overall system is critical. Examples of such networking equipment include Routers, Ethernet switches, ATM Edge switches, or Frame Relay switches.

### Service-Specific Performance Accelerators

The CN8236 incorporates numerous service-specific features designed to accelerate and enhance system performance. As examples, the CN8236 implements Echo Suppression of LAN traffic via LECID filtering, and supports Frame Relay DE to CLP interworking.

### Advanced xBR Traffic Management

The xBR Traffic Manager in the CN8236 supports multiple ATM service categories. This includes CBR, VBR (both single and dual leaky bucket), UBR, GFR (Guaranteed Frame Rate), and ABR. The CN8236 manages each VCC independently. It dynamically schedules segmentation traffic to comply with up to 16+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 CN8236's ABR algorithms to a specific system's or network's requirements.

–Continued–

### Distinguishing Features

#### Service-Specific Performance Accelerators

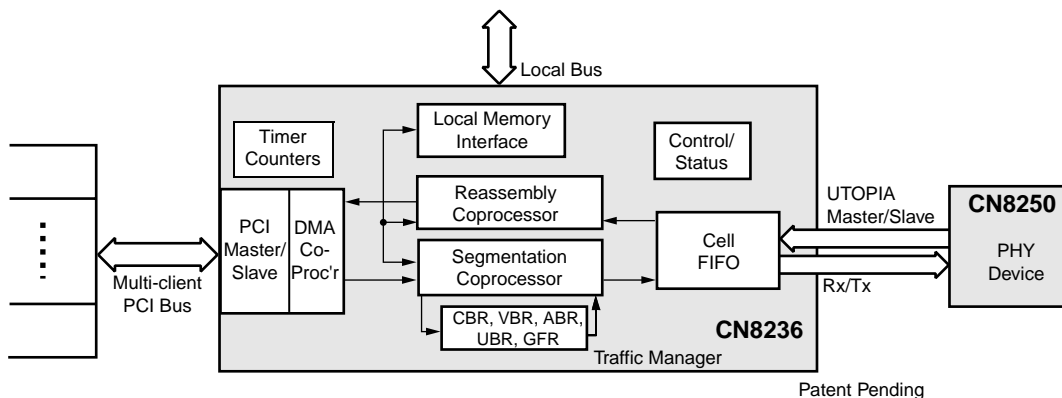
- LECID filtering and echo suppression
- Dual leaky bucket based on CLP (frame relay)
- Frame relay DE interworking
- Internal SNMP MIB counters
- IP over ATM; supports both CLP0+1 and ABR shaping

#### Flexible Architectures

- Multi-peer host
- Direct switch attachment via reverse UTOPIA
- ATM terminal
  - Host control
  - Local bus control
- Optional local processor

–Continued–

### Functional Block Diagram



## Ordering Information

| Model Number                                  | Manufacturing Part Number | Product Revision | Package     | Operating Temperature |
|---|---------------------------|------------------|-------------|-----------------------|
| CN8236  | 28236-12                  | B                | 388-pin BGA | -40 °C to 85 °C       |
| CN8236/<br>CX28250EVM<br>Evaluation<br>Module | BT00-D700-601             | —                | —           | —                     |

## Document Revision History

| Document Number | Device Revision | Comments   |
|-----------------|-----------------|--|
| N8236DSA        | CN8236 Rev. A   | This is the advanced issue of the data sheet.              |
| 100453B         | CN8236 Rev. B   | Put into new Conexant format.                              |
| 500372A         | CN8236 Rev. B   | Revisions made. Changed format from Conexant to Mindspeed. |
| 500372B         | CN8236 Rev. B   | Corrections as noted by change bars.                       |
| 28236-DSH-001-A | CN8236 Rev. B   | Corrections as noted by change bars.                       |
| 28236-DSH-001-B | CN8236 Rev. B   | Corrections as noted by change bars.                       |

© 1999-2003, Mindspeed Technologies™, 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™, the Mindspeed™ logo, and "Build It First"™. 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.

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

The CN8236's segmentation coprocessor generates ATM cells for up to 64 K VCCs. The segmentation coprocessor formats cells on each channel according to segmentation VCC tables, utilizing up to 32 independent transmit queues and reporting segmentation status on a parallel set of up to 32 segmentation status queues. The segmentation coprocessor fetches client data from the host, formats ATM cells while generating and appending protocol overhead, and forwards these to the UTOPIA port. The segmentation coprocessor operates as a slave to the xBR Traffic Manager which schedules VCCs for transmission.

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

The CN8236's reassembly coprocessor stores the payload data from the cell stream received by the UTOPIA port into host data buffers. Using a dynamic lookup method which supports NNI or UNI addressing, the reassembly coprocessor processes up to 64 K VCCs simultaneously. The host supplies free buffers on up to 32 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 32 independent reassembly status queues.

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

The CN8236 host interface architecture maximizes performance and system flexibility. The device's control and status queues enable host/SAR communication via write operations alone. This write-only architecture lowers latency and PCI bus occupancy. Flexibility is achieved by supporting a scalable peer-to-peer architecture. Multiple host clients can be addressed by the segmentation and reassembly (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 CN8236/RS8254EVM which provides a working reference design, an example of a 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.

*–Continued Distinguishing Features–*

**New Features**

- 3.3 V, 388 BGA lowers power and eases PCB assembly
- AAL3/4 CPCS generation and checking
- PCI 2.1, including support for serial EEPROM
- Enhancements to xBR Traffic Manager
  - fewer ABR templates
  - improved CBR tunneling
- Reduced memory size for VCC lookup tables
- Increased addressing flexibility
- Additional byte lane swappers for increased system flexibility
- UTOPIA level 2, 8/16 bit 50 MHz
- Programmable size routing tags up to 64 byte cells
- Selectable single/separate UTOPIA clocks
- Interworking function for AAL1 and 2 scheduling
  - Cell on demand scheduling
- Updated PM-OAM processing per i.610
- SECBC calculated per GR-1248
- Paging function in order to gluelessly control RS8228 cell delineator (SAR provides power)
- Robust EEPROM operation
- Compact PCI Hot Swap capabilities
- Master PCI write over read arbitration control
- Increase incoming DMA FIFO buffer from 2 kB to 8 kB
- Prepend VCC index on RSM BOM cells
- Optional reference clock drive scheduler
- Head of Line Flushing (HoLF) mechanism
- Internal loopback in multiPHY mode
- Programmable number of slots that the scheduler can fall behind

**xBR Traffic Management**

- TM4.1 Service Classes
  - CBR
  - VBR (single, dual and CLP-based leaky buckets)
  - Real time VBR
  - ABR
  - UBR
  - GFC (controlled & uncontrolled flows)
  - Guaranteed Frame Rate (GFR) (guaranteed MCR on UBR VCCs)
- 16 levels of priorities (16 + CBR)
- Dynamic per-VCC scheduling
- Multiple programmable ABR templates (supplied by Mindspeed or user)

- Scheduler driven by selectable clock
  - Local system clock
  - External reference clock
- Internal RM OAM cell feedback path
- Virtual FIFO buffer 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)
- 155 Mbps full duplex (two cell PDUs)

**Multi-Queue Segmentation Processing**

- 32 transmit queues with optional priority levels
- 64 K VCCs maximum
- AAL5 and AAL3/4 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 buffer - CDV - host latency matching (one to nine 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 buffer (PCI host)

**Multi-Queue Reassembly Processing**

- 32 reassembly queues
- 64 K VCCs maximum \*
- AAL5 and AAL3/4 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 buffer full
  - Frame relay DE with priority threshold
  - LECID filtering and echo suppression
  - Per-VCC firewalls

- Dynamic channel lookup (NNI or UNI addressing)
  - Supports full address space
  - Deterministic
  - Flexible VCI count per VPI
  - Optimized for signalling address assignment
- Message and streaming status modes
- Raw cell mode (52 octet)
- 200 Mbps half duplex
- 155 Mbps full duplex (w/ 2-cell PDUs)
- Distributed host or SAR-shared memory reassembly
- Eight programmable reassembly hardware time-outs (per-VCC assignable)
- Global max PDU length for AAL5
- Per-VCC buffer firewall (memory usage limit)
- Simultaneous reassembly and segmentation
- Idle cell filtering

**High Performance Host Architecture with Buffer Isolation**

- Write-only control and status
- Read multiple command for data transfer
- Up to 32 host clients control and status queues
- Physical or logical clients
  - Enables peer-to-peer architecture
- Descriptor-based buffer chaining
- Scatter/gather 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 FIFO buffers (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-RS823xHPI reference source code (C)

*–Continued–*

*–Continued Distinguishing Features–*

**Generous Implementation of OAM-PM Protocols**

- Detection of all F4/F5 OAM flows
- Internal PM monitoring and generation for up to 128 VCCs
- Optional global OAM Rx/Tx queues
- In-line OAM insertion and generation

**Standards-Based I/O**

- 33 MHz PCI 2.1 (to 40 MHz)
- Serial EEPROM to store PCI configuration information
- PHY interfaces
  - UTOPIA master (Level 1)
  - UTOPIA slave (Level 1)
  - UTOPIA master (Level 2)
  - UTOPIA slave (Level 2)
- 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 RS825x and Bt8223

**Standards Compliance**

- UNI/NNI 3.1
- TM 4.0/TM4.1
- Bellcore GR-1248
- ATM Forum B-ICI V2.0



# Table of Contents

---

- Table of Contents ..... 1-7
- List of Figures ..... 1-19
- List of Tables ..... 1-23
- 1.0 CN8236 Product Overview ..... 1-1
  - 1.1 Introduction ..... 1-1
  - 1.2 Service-Specific Performance Accelerators ..... 1-4
  - 1.3 Designer Toolkit ..... 1-7
- 2.0 Architecture Overview ..... 2-1
  - 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 CN8236 Queue Structure ..... 2-3
    - 2.2.3 Buffer Isolation Utilizing Descriptor-Based Buffer Chaining ..... 2-6
    - 2.2.4 Status Queue Relation to Buffers and Descriptors ..... 2-8
    - 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 ..... 2-18
    - 2.5.2 VBR Traffic ..... 2-18
    - 2.5.3 ABR Traffic ..... 2-19
    - 2.5.4 UBR Traffic ..... 2-19
    - 2.5.5 GFR Traffic ..... 2-19

|            |   |            |
|------------|---|------------|
| 2.5.6      | xBR Cell Scheduler                          | 2-20       |
| 2.5.7      | ABR Flow Control Manager                    | 2-21       |
| 2.6        | Burst FIFO Buffers                          | 2-22       |
| 2.7        | Implementation of OAM-PM Protocols          | 2-23       |
| 2.8        | Standards-Based I/O                         | 2-23       |
| 2.9        | Electrical/Mechanical                       | 2-25       |
| 2.10       | Logic Diagram and Pin Descriptions          | 2-25       |
| <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                      | 3-3        |
| 3.2.5      | Local Processor Clients                     | 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                            | 3-6        |
| 3.3.1.3    | Underflow Conditions                        | 3-7        |
| 3.3.2      | Write-only Status Queues                    | 3-8        |
| 3.3.2.1    | Control Variables                           | 3-8        |
| 3.3.2.2    | Queue Management                            | 3-8        |
| 3.3.2.3    | Overflow Conditions                         | 3-9        |
| 3.3.2.4    | Status Queue Interrupt Delay                | 3-10       |
| <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    | AAL3/4   | 4-9         |
| 4.2.3.3    | AAL0   | 4-11        |
| 4.2.4      | ATM PHY Layer Interface                                | 4-12        |
| 4.2.4.1    | Head-of-Line Flushing (HoLF)                           | 4-12        |
| 4.2.5      | Status Reporting                                       | 4-13        |
| 4.2.6      | Virtual FIFO Buffers                                   | 4-13        |
| <b>4.3</b> | <b>Segmentation Control and Data Structures</b>        | <b>4-14</b> |
| 4.3.1      | Segmentation VCC Table Entry                           | 4-14        |
| 4.3.2      | Data Buffers   | 4-18        |
| 4.3.3      | Segmentation Buffer Descriptors                        | 4-18        |
| 4.3.4      | Transmit Queues  | 4-23        |
| 4.3.4.1    | Entry Format   | 4-23        |
| 4.3.4.2    | Transmit Queue Management                              | 4-24        |
| 4.3.5      | Routing Tags   | 4-25        |
| 4.3.6      | Segmentation Status Queues                             | 4-28        |
| 4.3.6.1    | Entry Format   | 4-28        |
| 4.3.6.2    | Status Queue Management                                | 4-30        |
| 4.3.6.3    | Status Queue Overflow                                  | 4-30        |
| 4.3.7      | Segmentation Internal SRAM Memory Map                  | 4-31        |
| <b>5.0</b> | <b>Reassembly Coprocessor</b>                          | <b>5-1</b>  |
| <b>5.1</b> | <b>Overview</b>  | <b>5-1</b>  |
| <b>5.2</b> | <b>Reassembly Functional Description</b>               | <b>5-2</b>  |
| 5.2.1      | Reassembly VCCs  | 5-2         |
| 5.2.1.1    | Relation to Segmentation VCCs                          | 5-3         |
| 5.2.2      | Channel Lookup   | 5-4         |
| 5.2.2.1    | Programmable Block Size for VCC Table/ VCI Index Table | 5-4         |
| 5.2.2.2    | Setup  | 5-6         |
| 5.2.2.3    | Operation  | 5-6         |
| 5.2.2.4    | AAL3/4 Lookup  | 5-7         |
| 5.2.2.5    | Variable VPI/PORT_ID Lookup (Multi-phy Support)        | 5-8         |
| <b>5.3</b> | <b>CPCS-PDU Processing</b>                             | <b>5-9</b>  |
| 5.3.1      | AAL5 Processing  | 5-10        |
| 5.3.1.1    | AAL5 COM Processing                                    | 5-10        |
| 5.3.1.2    | AAL5 EOM Processing                                    | 5-10        |
| 5.3.1.3    | AAL5 Error Conditions                                  | 5-11        |
| 5.3.2      | AAL3/4 Processing                                      | 5-12        |
| 5.3.2.1    | AAL3/4 Per-Cell Processing                             | 5-13        |
| 5.3.2.2    | AAL3/4 Additional BOM/SSM Processing                   | 5-14        |
| 5.3.2.3    | AAL3/4 Additional COM Processing                       | 5-14        |

|            |   |             |
|------------|---|-------------|
| 5.3.2.4    | AAL3/4 Additional EOM/SSM Processing . . . . .                                | 5-15        |
| 5.3.2.5    | AAL3/4 MIB Counters . . . . .   | 5-15        |
| 5.3.3      | AALO Processing . . . . .   | 5-16        |
| 5.3.3.1    | Termination Methods . . . . .   | 5-16        |
| 5.3.3.2    | AALO Error Conditions . . . . .   | 5-16        |
| 5.3.4      | ATM Header Processing . . . . .   | 5-17        |
| 5.3.5      | BOM Synchronization Signal . . . . .  | 5-17        |
| 5.3.5.1    | Prepend Index . . . . .   | 5-17        |
| <b>5.4</b> | <b>Buffer Management . . . . .</b>  | <b>5-18</b> |
| 5.4.1      | Host vs. Local Reassembly . . . . .   | 5-18        |
| 5.4.2      | Scatter Method . . . . .  | 5-18        |
| 5.4.3      | Free Buffer Queues . . . . .  | 5-19        |
| 5.4.4      | Linked Data Buffers . . . . .   | 5-21        |
| 5.4.5      | Initialization of Buffer Structures . . . . .                                 | 5-22        |
| 5.4.5.1    | Buffer Descriptors . . . . .  | 5-22        |
| 5.4.5.2    | Free Buffer Queue Base Table . . . . .  | 5-22        |
| 5.4.5.3    | Free Buffer Queue Entries . . . . .   | 5-22        |
| 5.4.5.4    | Other Initialization . . . . .  | 5-22        |
| 5.4.6      | Buffer Allocation . . . . .   | 5-23        |
| 5.4.7      | Error Conditions . . . . .  | 5-23        |
| 5.4.8      | Early Packet Discard . . . . .  | 5-24        |
| 5.4.8.1    | General Description . . . . .   | 5-24        |
| 5.4.8.2    | Frame Relay Packet Discard . . . . .  | 5-24        |
| 5.4.8.3    | CLP Packet Discard . . . . .  | 5-24        |
| 5.4.8.4    | LANE-LECID Packet Discard—Echo Suppression on Multicast Data Frames . . . . . | 5-25        |
| 5.4.8.5    | DMA FIFO Buffer Full . . . . .  | 5-25        |
| 5.4.8.6    | AAL3/4 Early Packet Discard Processing . . . . .                              | 5-26        |
| 5.4.8.7    | Error Conditions . . . . .  | 5-26        |
| 5.4.9      | Hardware PDU Time-Out . . . . .   | 5-27        |
| 5.4.9.1    | Reassembly Time-Out Process . . . . .   | 5-27        |
| 5.4.9.2    | Halting Time-Out Processing . . . . .   | 5-27        |
| 5.4.9.3    | Timer Reset . . . . .   | 5-27        |
| 5.4.9.4    | Reassembly Time-Out Condition . . . . .                                       | 5-28        |
| 5.4.9.5    | Time-Out Period Calculation . . . . .   | 5-28        |
| 5.4.10     | Virtual FIFO Buffer Mode . . . . .  | 5-28        |
| 5.4.10.1   | Setup . . . . .   | 5-28        |
| 5.4.10.2   | Operation . . . . .   | 5-28        |
| 5.4.10.3   | Errors . . . . .  | 5-28        |
| 5.4.11     | Firewall Functions . . . . .  | 5-29        |
| 5.4.11.1   | Setup . . . . .   | 5-29        |
| 5.4.11.2   | Operation . . . . .   | 5-29        |
| 5.4.11.3   | Credit Return . . . . .   | 5-29        |

|            |   |      |
|------------|---|------|
| <b>5.5</b> | <b>Global Statistics</b> . . . . .  | 5-31 |
| <b>5.6</b> | <b>Status Queue Operation</b> . . . . .   | 5-32 |
| 5.6.1      | Structure. . . . .  | 5-32 |
| 5.6.1.1    | Setup . . . . .   | 5-33 |
| 5.6.1.2    | Operation. . . . .  | 5-34 |
| 5.6.1.3    | Errors . . . . .  | 5-34 |
| 5.6.1.4    | Host Detection of Status Queue Entries . . . . .  | 5-35 |
| 5.6.2      | Status Queue Overflow or Full Condition. . . . .  | 5-36 |
| <b>5.7</b> | <b>Reassembly Control and Data Structures</b> . . . . .                                 | 5-37 |
| 5.7.1      | Channel Lookup Structures . . . . .   | 5-37 |
| 5.7.2      | Reassembly VCC Table . . . . .  | 5-39 |
| 5.7.2.1    | AAL5, AAL0 and AAL3/4 VCC Table Entries . . . . .                                       | 5-40 |
| 5.7.2.2    | AAL3/4 Head VCC Table Entry . . . . .   | 5-46 |
| 5.7.3      | Reassembly Buffer Descriptor Structure. . . . .   | 5-49 |
| 5.7.4      | Free Buffer Queues . . . . .  | 5-49 |
| 5.7.5      | Reassembly Status Queues . . . . .  | 5-51 |
| 5.7.6      | LECID Table. . . . .  | 5-55 |
| 5.7.7      | Global Time-Out Table . . . . .   | 5-56 |
| 5.7.8      | Reassembly Internal SRAM Memory Map. . . . .  | 5-57 |
| <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 . . . . .  | 6-4  |
| 6.1.2      | ABR Flow Control Manager . . . . .  | 6-5  |
| <b>6.2</b> | <b>xBR Cell Scheduler Functional Description</b> . . . . .                              | 6-7  |
| 6.2.1      | Scheduling Priority . . . . .   | 6-7  |
| 6.2.1.1    | 16 Priority Levels + CBR . . . . .  | 6-7  |
| 6.2.1.2    | VCC Priority Assignment. . . . .  | 6-7  |
| 6.2.2      | Dynamic Schedule Table . . . . .  | 6-7  |
| 6.2.2.1    | Overview . . . . .  | 6-7  |
| 6.2.2.2    | Schedule Table Slots. . . . .   | 6-8  |
| 6.2.2.3    | Schedule Slot Formats without USE_SCH_CTRL Asserted . . . . .                           | 6-10 |
| 6.2.2.4    | Schedule Slot Formats with USE_SCH_CTRL Asserted . . . . .                              | 6-12 |
| 6.2.2.5    | Some Scheduling Scenarios . . . . .   | 6-13 |
| 6.2.3      | CBR Traffic . . . . .   | 6-14 |
| 6.2.3.1    | CBR Rate Selection. . . . .   | 6-14 |
| 6.2.3.2    | Available Rates . . . . .   | 6-15 |
| 6.2.3.3    | CBR Cell Delay Variation (CDV) . . . . .  | 6-16 |
| 6.2.3.4    | CBR Channel Management . . . . .  | 6-18 |
| 6.2.4      | VBR Traffic . . . . .   | 6-19 |
| 6.2.4.1    | Mapping CN8236 VBR Service Categories to <i>TM 4.1</i> VBR Service Categories . . . . . | 6-19 |
| 6.2.4.2    | Rate-Shaping vs. Policing . . . . .   | 6-19 |
| 6.2.4.3    | Single Leaky Bucket . . . . .   | 6-19 |

|            |  |             |
|------------|--|-------------|
| 6.2.4.4    | Dual Leaky Bucket  | 6-19        |
| 6.2.4.5    | CLP-Based Buckets  | 6-20        |
| 6.2.4.6    | Rate Selection   | 6-20        |
| 6.2.4.7    | Real-Time VBR and CDV                                      | 6-20        |
| 6.2.5      | UBR Traffic  | 6-20        |
| 6.2.6      | xBR Tunnels (Pipes)  | 6-20        |
| 6.2.7      | Guaranteed Frame Rate                                      | 6-22        |
| 6.2.8      | PCR Control for Priority Queues                            | 6-23        |
| <b>6.3</b> | <b>ABR Flow Control Manager</b>                            | <b>6-24</b> |
| 6.3.1      | A Brief Overview of <i>TM 4.1</i>                          | 6-24        |
| 6.3.2      | Internal ABR Feedback Control Loop                         | 6-24        |
| 6.3.2.1    | Source Flow Control Feedback                               | 6-25        |
| 6.3.2.2    | Destination Behavior                                       | 6-26        |
| 6.3.2.3    | Out-of-Rate Cells  | 6-26        |
| 6.3.3      | Source and Destination Behaviors                           | 6-27        |
| 6.3.4      | ABR VCC Parameters   | 6-27        |
| 6.3.5      | ABR Templates  | 6-27        |
| 6.3.6      | Cell Type Decisions  | 6-28        |
| 6.3.6.1    | In-rate Cell Streams                                       | 6-28        |
| 6.3.6.2    | ABR Cell Decisions   | 6-29        |
| 6.3.7      | Rate Decisions and Updates                                 | 6-32        |
| 6.3.7.1    | ABR Traffic Shaping  | 6-32        |
| 6.3.7.2    | Rate Adjustment Overview                                   | 6-32        |
| 6.3.7.3    | Backward RM Cell Flow Control                              | 6-32        |
| 6.3.7.4    | Forward RM Cell Transmission Decisions                     | 6-37        |
| 6.3.7.5    | ACR Change Notification                                    | 6-37        |
| 6.3.7.6    | Rate Adjustment in Turnaround RM Cells                     | 6-38        |
| 6.3.7.7    | Optional Rate Adjustment Due to Use-It-or-Lose-It Behavior | 6-40        |
| 6.3.8      | Boundary Conditions and Out-of-Rate RM Cells               | 6-41        |
| 6.3.8.1    | Calculated Rate Boundaries                                 | 6-41        |
| 6.3.8.2    | Out-of-Rate Forward RM Cell Generation                     | 6-41        |
| 6.3.8.3    | Out-of-Rate Backward RM Cells                              | 6-41        |
| <b>6.4</b> | <b>GFC Flow Control Manager</b>                            | <b>6-42</b> |
| 6.4.1      | A Brief Overview of GFC                                    | 6-42        |
| 6.4.2      | The CN8236's Implementation of GFC                         | 6-42        |
| 6.4.2.1    | Configuring the Link for GFC Operation                     | 6-43        |
| <b>6.5</b> | <b>Traffic Management Control and Status Structures</b>    | <b>6-44</b> |
| 6.5.1      | Schedule Table   | 6-44        |
| 6.5.2      | CBR-Specific Structures                                    | 6-44        |
| 6.5.2.1    | CBR Traffic  | 6-44        |
| 6.5.2.2    | Tunnel Traffic   | 6-44        |
| 6.5.2.3    | SCH_STATE Fields For CBR                                   | 6-45        |
| 6.5.3      | VBR-Specific Structure                                     | 6-46        |
| 6.5.3.1    | VBR SCH_STATE  | 6-46        |

|            |  |             |
|------------|--|-------------|
| 6.5.3.2    | VBR1 or VBR2 Schedule State Table                          | 6-46        |
| 6.5.3.3    | Bucket Table for VBR2 and VBRC                             | 6-48        |
| 6.5.4      | GFR-Specific Structures                                    | 6-49        |
| 6.5.4.1    | GFR Schedule State Table                                   | 6-49        |
| 6.5.4.2    | GFR MCR Limit Bucket Table                                 | 6-50        |
| 6.5.5      | ABR-Specific Structures                                    | 6-51        |
| 6.5.5.1    | ABR Schedule State Table                                   | 6-51        |
| 6.5.6      | ABR Instruction Tables                                     | 6-55        |
| 6.5.7      | RS_QUEUE   | 6-59        |
| 6.5.8      | Scheduler Internal SRAM Registers                          | 6-60        |
| <b>7.0</b> | <b>OAM Functions</b>                                       | <b>7-1</b>  |
| <b>7.1</b> | <b>OAM Overview</b>  | <b>7-1</b>  |
| 7.1.1      | OAM Functions Supported                                    | 7-1         |
| 7.1.2      | OAM Flows Supported  | 7-3         |
| 7.1.2.1    | F4 OAM Flow  | 7-3         |
| 7.1.2.2    | F5 OAM Flow  | 7-3         |
| 7.1.2.3    | Performance Monitoring (PM)                                | 7-4         |
| 7.1.3      | OAM Cell Format  | 7-5         |
| 7.1.4      | Local vs. Host Processing of OAM                           | 7-6         |
| <b>7.2</b> | <b>Segmentation of OAM Cells</b>                           | <b>7-7</b>  |
| 7.2.1      | Key OAM-Related Fields for OAM Segmentation                | 7-7         |
| 7.2.1.1    | Segmentation Buffer Descriptors                            | 7-7         |
| 7.2.1.2    | Low Latency Transmission                                   | 7-7         |
| 7.2.1.3    | Segmentation Status Queue                                  | 7-7         |
| 7.2.1.4    | F4 Flow  | 7-7         |
| 7.2.2      | Error Condition During OAM Segmentation                    | 7-8         |
| <b>7.3</b> | <b>Reassembly of OAM Cells</b>                             | <b>7-9</b>  |
| 7.3.1      | Key OAM-Related Fields for OAM Reassembly                  | 7-9         |
| 7.3.1.1    | Reassembly VCC State Table                                 | 7-9         |
| 7.3.1.2    | Reassembly Status Queue                                    | 7-9         |
| 7.3.1.3    | F4 Flow  | 7-9         |
| 7.3.2      | OAM Reassembly Operation                                   | 7-10        |
| 7.3.3      | Error Conditions During OAM Reassembly                     | 7-10        |
| <b>7.4</b> | <b>PM Processing</b>                                       | <b>7-11</b> |
| 7.4.1      | Initializing PM Operation                                  | 7-12        |
| 7.4.2      | Setting Up Channels for PM Operation                       | 7-13        |
| 7.4.3      | PM Operation   | 7-13        |
| 7.4.3.1    | Generation of Forward Monitoring PM Cells                  | 7-13        |
| 7.4.3.2    | Reassembly of Forward Monitoring PM Cells                  | 7-14        |
| 7.4.3.3    | Reassembly of Backward Reporting PM Cells                  | 7-14        |
| 7.4.3.4    | Turnaround and Segmentation of Backward Reporting PM Cells | 7-14        |
| 7.4.3.5    | Turnaround of Backward Reporting PM Cells ONLY             | 7-14        |
| 7.4.4      | Error Conditions During PM Processing                      | 7-14        |
| 7.4.5      | PASS_OAM Function  | 7-14        |

|             |   |       |
|-------------|---|-------|
| 7.5         | <b>OAM Control and Status Structures</b> .....                      | 7-15  |
| 7.5.1       | SEG_PM Structure .....  | 7-15  |
| 7.5.2       | RSM_PM Table .....  | 7-17  |
| <b>8.0</b>  | <b>DMA Coprocessor</b> .....  | 8-1   |
| 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> .....                                 | 9-1   |
| 9.1         | Overview .....  | 9-1   |
| 9.2         | Memory Bank Characteristics .....                                   | 9-3   |
| 9.3         | Memory Size Analysis .....  | 9-6   |
| <b>10.0</b> | <b>Local Processor Interface</b> .....                              | 10-1  |
| 10.1        | Overview .....  | 10-1  |
| 10.2        | Interface Pin Descriptions .....                                    | 10-3  |
| 10.3        | Bus Cycle Descriptions .....  | 10-5  |
| 10.3.1      | Single Read Cycle, Zero Wait State Example .....                    | 10-5  |
| 10.3.2      | Single Read Cycle, Wait States Inserted By Memory Arbitration ..... | 10-7  |
| 10.3.3      | Double Read Burst With Processor Wait States .....                  | 10-8  |
| 10.3.4      | Single Write With One-Wait-State Memory .....                       | 10-9  |
| 10.3.5      | Quad Write Burst, No Wait States .....                              | 10-10 |
| 10.4        | Processor Interface Signals .....                                   | 10-11 |
| 10.5        | Local Processor Operating Mode .....                                | 10-13 |
| 10.6        | Standalone Operation .....  | 10-14 |
| 10.6.1      | Microprocessor Interface for Multiple Physical Devices .....        | 10-18 |
| 10.7        | System Clocking .....   | 10-19 |
| 10.8        | Real-Time Clock Alarm .....   | 10-19 |
| 10.9        | CN8236 Reset .....  | 10-20 |
| <b>11.0</b> | <b>PCI Bus Interface</b> .....                                      | 11-1  |
| 11.1        | Overview .....  | 11-1  |
| 11.2        | Unimplemented PCI Bus Interface Functions .....                     | 11-3  |
| 11.3        | PCI Configuration Space .....                                       | 11-4  |
| 11.4        | PCI Bus Master Logic .....  | 11-4  |
| 11.5        | Burst FIFO Buffers .....  | 11-6  |

|             |  |             |
|-------------|--|-------------|
| 11.6        | PCI Bus Slave Logic                                      | 11-6        |
| 11.7        | Byte Swapping of Control Structures                      | 11-7        |
| 11.8        | Power Management   | 11-7        |
| 11.9        | Interface Module to Serial EEPROM                        | 11-8        |
| 11.9.1      | EEPROM Format  | 11-9        |
| 11.9.2      | Loading the EEPROM Data at Reset                         | 11-9        |
| 11.9.3      | Accessing the EEPROM                                     | 11-10       |
| 11.9.4      | Using the Subsystem ID Without an EEPROM                 | 11-10       |
| 11.10       | PCI Host Address Map                                     | 11-11       |
| <b>12.0</b> | <b>ATM UTOPIA Interface</b>                              | <b>12-1</b> |
| 12.1        | Overview of ATM UTOPIA Interface                         | 12-1        |
| 12.2        | ATM UTOPIA Interface Logic                               | 12-2        |
| 12.3        | ATM Physical I/O Pins                                    | 12-3        |
| 12.3.1      | UTOPIA Interface   | 12-6        |
| 12.4        | UTOPIA Level 2 Interface                                 | 12-7        |
| 12.4.1      | Cell Tagging   | 12-7        |
| 12.4.2      | UTOPIA Configuration Control                             | 12-9        |
| 12.4.3      | UTOPIA Level 2 Multi-Port Operation                      | 12-10       |
| 12.5        | UTOPIA Level 1 Mode Cell Handshake Timing                | 12-12       |
| 12.6        | UTOPIA Level 1 Mode Octet Handshake Timing               | 12-14       |
| 12.7        | Slave Level 1 UTOPIA Mode                                | 12-16       |
| 12.8        | Loopback Mode  | 12-18       |
| 12.9        | Receive Cell Synchronization Logic                       | 12-19       |
| 12.10       | Transmit Cell Synchronization Logic                      | 12-20       |
| <b>13.0</b> | <b>AALx Interworking</b>                                 | <b>13-1</b> |
| 13.1        | AALx RSM Operation                                       | 13-3        |
| 13.2        | AALx SEG Operation                                       | 13-4        |
| 13.2.1      | AALx Network Centric Operation—(EXTERNAL_SCH = 0)        | 13-4        |
| 13.2.2      | AALx Voice Centric Operation—(EXTERNAL_SCH = 1)          | 13-5        |
| <b>14.0</b> | <b>CN8236 Registers</b>                                  | <b>14-1</b> |
| 14.1        | Control and Status Registers                             | 14-1        |
| 14.2        | System Registers   | 14-4        |
| 14.3        | Segmentation Registers                                   | 14-9        |
| 14.4        | Scheduler Registers                                      | 14-12       |
| 14.4.1      | 0xc4—PCR Queue Interval 2 and 3 Register (PCR_QUE_INT23) | 14-18       |
| 14.5        | Reassembly Registers                                     | 14-19       |

|                    |   |       |
|--------------------|---|-------|
| 14.6               | <b>Counters and Status Registers</b> . . . . .                  | 14-26 |
| 14.6.1             | Host Interrupt Status Registers . . . . .                       | 14-29 |
| 14.6.2             | Local Processor Interrupt Status Registers. . . . .             | 14-34 |
| 14.7               | <b>PCI Bus Interface Registers.</b> . . . . .                   | 14-39 |
| <b>15.0</b>        | <b>SAR Initialization—Example Tables</b> . . . . .              | 15-1  |
| 15.1               | <b>Segmentation Initialization.</b> . . . . .                   | 15-1  |
| 15.1.1             | Segmentation Control Registers. . . . .                         | 15-1  |
| 15.1.2             | Segmentation Internal Memory Control Structures . . . . .       | 15-2  |
| 15.1.3             | Segmentation SAR Shared Memory Control Structures. . . . .      | 15-3  |
| 15.2               | <b>Scheduler Initialization.</b> . . . . .                      | 15-5  |
| 15.2.1             | Scheduler Control Registers. . . . .                            | 15-5  |
| 15.2.2             | Scheduler Internal Memory Control Structures . . . . .          | 15-6  |
| 15.2.3             | Scheduler SAR Shared Memory Control Structures. . . . .         | 15-6  |
| 15.3               | <b>Reassembly Initialization</b> . . . . .                      | 15-8  |
| 15.3.1             | Reassembly Control Registers . . . . .                          | 15-8  |
| 15.3.2             | Reassembly Internal Memory Control Structures . . . . .         | 15-10 |
| 15.3.3             | Reassembly SAR Shared Memory Control Structures . . . . .       | 15-11 |
| 15.4               | <b>General Initialization</b> . . . . .                         | 15-15 |
| 15.4.1             | General Control Registers . . . . .                             | 15-15 |
| <b>16.0</b>        | <b>Electrical and Mechanical Specifications</b> . . . . .       | 16-1  |
| 16.1               | <b>Timing</b> . . . . .   | 16-1  |
| 16.1.1             | PCI Bus Interface Timing . . . . .                              | 16-1  |
| 16.1.2             | ATM Physical Interface Timing—UTOPIA and Slave UTOPIA . . . . . | 16-4  |
| 16.1.3             | System Clock Timing. . . . .                                    | 16-7  |
| 16.1.4             | CN8236 Memory Interface Timing . . . . .                        | 16-9  |
| 16.1.5             | PHY Interface Timing (Standalone Mode). . . . .                 | 16-13 |
| 16.1.6             | Local Processor Interface Timing. . . . .                       | 16-14 |
| 16.2               | <b>Absolute Maximum Ratings</b> . . . . .                       | 16-19 |
| 16.3               | <b>DC Characteristics</b> . . . . .                             | 16-20 |
| 16.4               | <b>Mechanical Specifications</b> . . . . .                      | 16-22 |
| <b>Appendix A:</b> | <b>Boundary Scan</b> . . . . .                                  | A-1   |
| A.1                | <b>Instruction Register</b> . . . . .                           | A-3   |
| A.2                | <b>BYPASS Register</b> . . . . .                                | A-4   |
| A.3                | <b>Boundary Scan Register</b> . . . . .                         | A-4   |
| A.4                | <b>Boundary Scan Register Cells</b> . . . . .                   | A-5   |
| A.5                | <b>Electrical Characteristics</b> . . . . .                     | A-13  |



**A.6 Boundary Scan Description Language  
(BSDL) File A-15**

**Appendix B: List of Acronyms** ..... 1-31



## List of Figures

|              |   |      |
|--------------|---|------|
| Figure 1-1.  | CN8236 Functional Block Diagram                                       | 1-3  |
| Figure 2-1.  | Multiple Client Architecture Supports Up To 32 Clients                | 2-3  |
| Figure 2-2.  | CN8236 Queue Architecture   | 2-4  |
| Figure 2-3.  | Interaction of Queues with CN8236 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. | Data FIFO Buffers   | 2-22 |
| Figure 2-11. | CN8236 Logic Diagram (1 of 3)   | 2-26 |
| Figure 3-1.  | Client/Server Model of the CN8236                                     | 3-2  |
| Figure 3-2.  | Peer-to-Peer vs. Centralized Memory Data Transfers                    | 3-4  |
| Figure 3-3.  | Out-of-Band Control Architecture                                      | 3-4  |
| Figure 3-4.  | Write-only Control Queue  | 3-7  |
| Figure 3-5.  | 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 4-6.  | AAL3/4 CPCS-PDU Generation  | 4-11 |
| Figure 4-7.  | Route Tag Table for tag_size = 1 through 4                            | 4-25 |
| Figure 4-8.  | Route Tag Table for tag_size = 5 through 8                            | 4-26 |
| Figure 4-9.  | Route Tag Table for tag_size = 9 through 11                           | 4-26 |
| 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.  | Direct Index Lookup Method for AAL3/4                                 | 5-7  |
| Figure 5-6.  | VPI Index Table with Multiple Ports                                   | 5-8  |
| Figure 5-7.  | CPCS-PDU Reassembly   | 5-9  |
| Figure 5-8.  | AAL5 EOM Cell Processing—Fields to Status Queue                       | 5-10 |
| Figure 5-9.  | AAL5 Processing—CRC and PDU Length Checks                             | 5-11 |
| Figure 5-10. | AAL3/4 CPCS—PDU Reassembly  | 5-12 |
| Figure 5-11. | AAL0 PTI PDU Termination  | 5-16 |
| Figure 5-12. | Host and SAR-Shared Memory Data Structures for Scatter Method         | 5-19 |
| Figure 5-13. | Free Buffer Queue Structure   | 5-20 |

|              |  |       |
|--------------|--|-------|
| Figure 5-14. | Data Buffer Structures . . . . .   | 5-21  |
| Figure 5-15. | Data Structure Locations for Status Queues . . . . .                     | 5-32  |
| Figure 5-16. | Status Queue Structure Format . . . . .                                  | 5-33  |
| Figure 5-17. | VPI/VCI Channel Lookup Structure . . . . .                               | 5-37  |
| Figure 5-18. | Reassembly VCC Table Entry Lookup Mechanism. . . . .                     | 5-39  |
| Figure 5-19. | LECID Table, Illustrated . . . . .                                       | 5-55  |
| Figure 6-1.  | Non-ABR Cell Scheduling . . . . .  | 6-4   |
| Figure 6-2.  | ABR Flow Control . . . . .   | 6-5   |
| Figure 6-3.  | Schedule Table with Size = 100 . . . . .                                 | 6-8   |
| Figure 6-4.  | Schedule Slot Formats With USE_SCH_CTRL Not Asserted . . . . .           | 6-11  |
| Figure 6-5.  | Schedule Slot Formats With USE_SCH_CTRL Asserted . . . . .               | 6-12  |
| Figure 6-6.  | One Possible Scheduling Priority Scheme with the CN8236 . . . . .        | 6-13  |
| Figure 6-7.  | Assigning CBR Cell Slots. . . . .  | 6-15  |
| Figure 6-8.  | Introduction of CDV at the ATM/PHY Layer Interface . . . . .             | 6-16  |
| Figure 6-9.  | Schedule Table with Slot Conflicts at Different CBR Rates . . . . .      | 6-16  |
| Figure 6-10. | CDV Caused by Schedule Table Size at Certain CBR Rates . . . . .         | 6-17  |
| Figure 6-11. | Another Possible Scheduling Priority Scheme with the CN8236 . . . . .    | 6-22  |
| Figure 6-12. | ABR Service Category Feedback Control . . . . .                          | 6-24  |
| Figure 6-13. | CN8236 ABR-ER Feedback Loop (Source Behavior) . . . . .                  | 6-25  |
| Figure 6-14. | CN8236 ABR-ER Feedback Generation (Destination Behavior) . . . . .       | 6-26  |
| Figure 6-15. | Steady State ABR-ER Cell Stream . . . . .                                | 6-28  |
| Figure 6-16. | Cell Type Interleaving on ABR-ER Cell Stream. . . . .                    | 6-29  |
| Figure 6-17. | Cell Decision Table for Nrm = 32 . . . . .                               | 6-31  |
| Figure 6-18. | Backward_RM Flow Control, Block Diagram . . . . .                        | 6-33  |
| Figure 6-19. | RR RATE_INDEX Candidate Selection . . . . .                              | 6-34  |
| Figure 6-20. | ER RATE_INDEX Candidate Selection . . . . .                              | 6-35  |
| Figure 6-21. | Dynamic Traffic Shaping from RM Cell Feedback . . . . .                  | 6-36  |
| Figure 6-22. | ER Reduction Mapping . . . . .   | 6-39  |
| Figure 6-23. | ABR Linkage . . . . .  | 6-58  |
| Figure 6-24. | Head and Tail Pointers. . . . .  | 6-60  |
| Figure 7-1.  | OAM Cell Format . . . . .  | 7-5   |
| Figure 7-2.  | Functional Blocks for PM Segmentation and Reassembly . . . . .           | 7-11  |
| 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.  | CN8236 Memory Map . . . . .  | 9-2   |
| Figure 9-2.  | 0.5 MB SRAM Bank Utilizing by_8 Devices . . . . .                        | 9-4   |
| Figure 9-3.  | 1 MB SRAM Bank Utilizing by_16 Devices . . . . .                         | 9-4   |
| Figure 10-1. | CN8236—Local Processor Interface . . . . .                               | 10-1  |
| Figure 10-2. | Local Processor Single Read Cycle . . . . .                              | 10-6  |
| Figure 10-3. | Local Processor Single Read Cycle with Arbitration Wait States . . . . . | 10-7  |
| Figure 10-4. | Local Processor Double Read with Wait States Inserted. . . . .           | 10-8  |
| Figure 10-5. | Local Processor Single Write with One Wait State by_16 SRAM. . . . .     | 10-9  |
| Figure 10-6. | Local Processor Quad Write, No Wait States. . . . .                      | 10-10 |
| Figure 10-7. | i960CA/CF to the CN8236 Interface. . . . .                               | 10-12 |

|               |  |       |
|---------------|--|-------|
| Figure 10-8.  | CN825x and SAR (CN8236) Interface (Standalone Operation) . . . . .     | 10-15 |
| Figure 10-9.  | CN8236/PHY Functional Timing with Inserted Wait States . . . . .       | 10-16 |
| Figure 10-10. | CN8236/RS825x Read/Write Functional Timing . . . . .                   | 10-17 |
| Figure 10-11. | SAR/Peak 8 Control Connections . . . . .                               | 10-18 |
| Figure 11-1.  | EEPROM Connection . . . . .  | 11-8  |
| Figure 12-1.  | UTOPIA Level 2 Receive Timing . . . . .                                | 12-11 |
| Figure 12-2.  | Receive Timing in UTOPIA Level 1 Mode with Cell Handshake . . . . .    | 12-12 |
| Figure 12-3.  | Transmit Timing in UTOPIA Level 1 Mode with Cell Handshake . . . . .   | 12-13 |
| Figure 12-4.  | Receive Timing in UTOPIA Level 1 Mode with Octet Handshake . . . . .   | 12-14 |
| Figure 12-5.  | Transmit Timing in UTOPIA Level 1 Mode with Octet Handshake . . . . .  | 12-15 |
| Figure 12-6.  | Receive Timing in Slave UTOPIA Level 1 Mode . . . . .                  | 12-16 |
| Figure 12-7.  | Transmit Timing in Slave UTOPIA Level 1 Mode . . . . .                 | 12-17 |
| Figure 12-8.  | Source Loopback Mode Diagram . . . . .                                 | 12-18 |
| Figure 16-1.  | PCI Bus Input Timing Measurement Conditions . . . . .                  | 16-2  |
| Figure 16-2.  | PCI Bus Output Timing Measurement Conditions . . . . .                 | 16-3  |
| Figure 16-3.  | UTOPIA and Slave UTOPIA Input Timing Measurement Conditions . . . . .  | 16-5  |
| Figure 16-4.  | UTOPIA and Slave UTOPIA Output Timing Measurement Conditions . . . . . | 16-6  |
| Figure 16-5.  | Input System Clock Waveform . . . . .                                  | 16-8  |
| Figure 16-6.  | Output System Clock Waveform . . . . .                                 | 16-8  |
| Figure 16-7.  | CN8236 Memory Read Timing . . . . .                                    | 16-11 |
| Figure 16-8.  | CN8236 Memory Write Timing . . . . .                                   | 16-12 |
| Figure 16-9.  | Synchronous PHY Interface Input Timing . . . . .                       | 16-13 |
| Figure 16-10. | Synchronous PHY Interface Output Timing . . . . .                      | 16-14 |
| Figure 16-11. | Synchronous Local Processor Input Timing . . . . .                     | 16-15 |
| Figure 16-12. | Synchronous Local Processor Output Timing . . . . .                    | 16-15 |
| Figure 16-13. | Local Processor Read Timing . . . . .                                  | 16-17 |
| Figure 16-14. | Local Processor Write Timing . . . . .                                 | 16-18 |
| Figure 16-15. | 388-Pin Ball Grid Array Package (BGA) . . . . .                        | 16-23 |
| Figure 16-16. | CN8236 Pinout Configuration . . . . .                                  | 16-24 |
| Figure A-1.   | Test Circuitry Block Diagram . . . . .                                 | A-2   |
| Figure A-2.   | Timing Diagram . . . . .   | A-14  |



## List of Tables

|             |  |      |
|-------------|--|------|
| Table 2-1.  | Hardware Signal Definitions . . . . .  | 2-29 |
| Table 3-1.  | CN8236 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.  | AAL3/4 CPCS-PDU Field Generation . . . . .   | 4-9  |
| Table 4-3.  | AAL3/4 SAR-PDU Field Generation . . . . .  | 4-10 |
| Table 4-4.  | Coding of Segment Type (ST) Field . . . . .  | 4-10 |
| Table 4-5.  | Segmentation VCC Table Entry—AAL3/4-AAL5-AAL0 Format . . . . .                       | 4-14 |
| Table 4-6.  | Segmentation VCC Table Entry—AAL3/4, 5, and 0 Field Descriptions . . . . .           | 4-15 |
| Table 4-7.  | Segmentation VCC Table Entry—Virtual FIFO Buffers . . . . .                          | 4-17 |
| Table 4-8.  | Segmentation VCC Table Entry—Virtual FIFO Buffer Format Field Descriptions . . . . . | 4-17 |
| Table 4-9.  | Segmentation Buffer Descriptor Entry Format . . . . .                                | 4-18 |
| Table 4-10. | MISC_DATA Field Bit Definitions with HEADER_MOD Bit Set . . . . .                    | 4-18 |
| Table 4-11. | MISC_DATA Field Bit Definitions with RPL_VCI Bit Set . . . . .                       | 4-19 |
| Table 4-12. | MISC_DATA Field Bit Definitions with AAL_MODE Set to AAL3/4 . . . . .                | 4-19 |
| Table 4-13. | Segmentation Buffer Descriptor Field Descriptions . . . . .                          | 4-20 |
| Table 4-14. | Transmit Queue Entry Format . . . . .  | 4-23 |
| Table 4-15. | Transmit Queue Entry Field Descriptions . . . . .                                    | 4-23 |
| Table 4-16. | Transmit Queue Base Table Entry . . . . .  | 4-24 |
| Table 4-17. | Transmit Queue Base Table Entry Field Descriptions . . . . .                         | 4-24 |
| Table 4-18. | Maximum TxFIFO Size with Routing Tags . . . . .                                      | 4-25 |
| Table 4-19. | Routing Tag Cross-Reference . . . . .  | 4-27 |
| Table 4-20. | Segmentation Status Queue Entry . . . . .  | 4-28 |
| Table 4-21. | Segmentation Status Queue Entry Field Descriptions . . . . .                         | 4-28 |
| Table 4-22. | Segmentation Status Queue Format for ACR/ER . . . . .                                | 4-29 |
| Table 4-23. | Status Queue Entry Field Descriptions for ACR/ER . . . . .                           | 4-29 |
| Table 4-24. | Segmentation Status Queue Base Table Entry . . . . .                                 | 4-30 |
| Table 4-25. | Segmentation Status Queue Base Table Entry Field Descriptions . . . . .              | 4-30 |
| Table 4-26. | Segmentation Internal SRAM Memory Map . . . . .                                      | 4-31 |
| Table 5-1.  | Programmable Block Size Values for Direct Index Lookup . . . . .                     | 5-5  |
| Table 5-2.  | STAT Output Pin Values for BOM Synchronization . . . . .                             | 5-17 |
| Table 5-3.  | Prepend Index Table Format . . . . .   | 5-17 |
| Table 5-4.  | Normal VPI Index Table Entry Format . . . . .  | 5-37 |
| Table 5-5.  | VPI Index Table Entry Format with EN_PROG_BLK_SX(RSM_CTRL1) Enabled . . . . .        | 5-37 |
| Table 5-6.  | VPI Index Table Entry Descriptions . . . . .   | 5-38 |
| Table 5-7.  | Normal VCI Index Table Format . . . . .  | 5-38 |
| Table 5-8.  | VCI Index Table Format with EN_PROG_BLK_SZ (RSM_CTRL1) Enabled . . . . .             | 5-38 |
| Table 5-9.  | VCI Index Table Descriptions . . . . .   | 5-38 |

|             |  |      |
|-------------|--|------|
| Table 5-10. | Reassembly VCC Table Entry Format—AAL5                             | 5-40 |
| Table 5-11. | Reassembly VCC Table Entry Format—AAL0                             | 5-41 |
| Table 5-12. | Reassembly VCC Table Entry Format—AAL3/4                           | 5-42 |
| Table 5-13. | Reassembly VCC Table Descriptions                                  | 5-43 |
| Table 5-14. | AAL3/4 Head VCC Table Entry Format                                 | 5-46 |
| Table 5-15. | AAL3/4 Head VCC Table Descriptions                                 | 5-46 |
| Table 5-16. | Reassembly Buffer Descriptor Structure                             | 5-49 |
| Table 5-17. | Reassembly Buffer Descriptor Structure Definitions                 | 5-49 |
| Table 5-18. | Free Buffer Queue Base Table Entry Format                          | 5-49 |
| Table 5-19. | Free Buffer Queue Base Table Entry Descriptions                    | 5-50 |
| Table 5-20. | Free Buffer Queue Entry Format                                     | 5-50 |
| Table 5-21. | Free Buffer Queue Entry Descriptions                               | 5-50 |
| Table 5-22. | Reassembly Status Queue Base Table Entry Format                    | 5-51 |
| Table 5-23. | Reassembly Status Queue Base Table Entry Descriptions              | 5-51 |
| Table 5-24. | Reassembly Status Queue Entry Format with FWD_PM = 0               | 5-51 |
| Table 5-25. | Reassembly Status Queue Entry Format with FWD_PM = 1               | 5-52 |
| Table 5-26. | Reassembly Status Queue Entry Format with FWD_PM = 0 and AAL34 = 1 | 5-52 |
| Table 5-27. | PDU_CHECKS Field Bits  | 5-52 |
| Table 5-28. | PDU_CHECKS Field Bits with CNT_ROVR = 1 and AAL34 = 1              | 5-52 |
| Table 5-29. | STATUS Field Bits  | 5-52 |
| Table 5-30. | STM Field Bits   | 5-52 |
| Table 5-31. | Reassembly Status Queue Entry Descriptions                         | 5-53 |
| Table 5-32. | LECID Table Entries  | 5-55 |
| Table 5-33. | LECID Table Field Definition                                       | 5-55 |
| Table 5-34. | Global Time-Out Table Entry Format                                 | 5-56 |
| Table 5-35. | Global Time-Out Table Entry Descriptions                           | 5-56 |
| Table 5-36. | Reassembly Internal SRAM Memory Map                                | 5-57 |
| Table 6-1.  | ATM Service Category Parameters and Attributes                     | 6-2  |
| Table 6-2.  | Scheduler Clock Selection  | 6-7  |
| Table 6-3.  | Selection of Schedule Table Slot Size by System Requirements       | 6-9  |
| Table 6-4.  | CN8236 VBR to TM 4.1 VBR Mapping                                   | 6-19 |
| Table 6-5.  | ABR Cell Type Decision Vector (ACDV)                               | 6-30 |
| Table 6-6.  | Schedule Slot Entry—CBR/Tunnel Traffic                             | 6-44 |
| Table 6-7.  | CBR_TUN_ID Field, Bit Definitions—CBR Slot                         | 6-44 |
| Table 6-8.  | CBR_TUN_ID Field, Bit Definitions—Tunnel Slot                      | 6-44 |
| Table 6-9.  | Schedule Slot Field Descriptions—CBR Traffic                       | 6-45 |
| Table 6-10. | SCH_STATE for SCH_MODE = CBR                                       | 6-45 |
| Table 6-11. | CBR SCH_STATE Field Descriptions                                   | 6-45 |
| Table 6-12. | SCH_STATE for SCH_MODE = VBR1 or VBR2                              | 6-46 |
| Table 6-13. | VBR1 and VBR2 SCH_STATE Field Descriptions                         | 6-46 |
| Table 6-14. | Bucket Table Entry   | 6-48 |
| Table 6-15. | Bucket Table Entry Field Descriptions                              | 6-48 |
| Table 6-16. | SCH_STATE for SCH_MODE = GFR                                       | 6-49 |
| Table 6-17. | SCH_STATE Field Descriptions for SCH_MODE = GFR                    | 6-49 |
| Table 6-18. | GFR MCR Limit Bucket Table Entry                                   | 6-50 |
| Table 6-19. | GFR MCR Bucket Table Entry Field Descriptions                      | 6-50 |



|             |  |       |
|-------------|--|-------|
| Table 6-20. | SCH_STATE for SCH_MODE = ABR . . . . .                                       | 6-51  |
| Table 6-21. | ABR SCH_STATE Field Descriptions . . . . .                                   | 6-52  |
| Table 6-22. | ABR Cell Decision Block (ACDB) . . . . .                                     | 6-55  |
| Table 6-23. | ABR Cell Type Actions . . . . .  | 6-55  |
| Table 6-24. | ABR Cell Type Decision Vector (ACDV) . . . . .                               | 6-55  |
| Table 6-25. | ABR Rate Decision Block (ARDB) . . . . .                                     | 6-56  |
| Table 6-26. | ARDB Field Descriptions . . . . .  | 6-56  |
| Table 6-27. | ABR Rate Decision Vector (ARDV) . . . . .                                    | 6-57  |
| Table 6-28. | Exponent Table . . . . .   | 6-57  |
| Table 6-29. | Exponent Table Field Descriptions . . . . .                                  | 6-57  |
| Table 6-30. | RS_QUEUE Entry—OAM-PM Reporting Information Ready for Transmission . . . . . | 6-59  |
| Table 6-31. | RS_QUEUE Entry—Forward ER RM Cell Received . . . . .                         | 6-59  |
| Table 6-32. | RS_QUEUE Entry—Backward ER RM Cell Received . . . . .                        | 6-59  |
| Table 6-33. | RS_QUEUE Field Descriptions . . . . .  | 6-59  |
| Table 6-34. | Scheduler Internal SRAM Memory Map (Head/Tail Pointers) . . . . .            | 6-60  |
| Table 7-1.  | OAM Functions of the ATM Layer . . . . .                                     | 7-1   |
| Table 7-2.  | VCI Values for F4 OAM Flows . . . . .  | 7-3   |
| Table 7-3.  | PTI Values for F5 OAM Flows . . . . .  | 7-3   |
| Table 7-4.  | OAM Type and Function Type Identifiers . . . . .                             | 7-6   |
| Table 7-5.  | PM-OAM Field Initialization For Any PM_INDEX . . . . .                       | 7-12  |
| Table 7-6.  | SEG_PM Structure . . . . .   | 7-15  |
| Table 7-7.  | SEG_PM Field Descriptions . . . . .  | 7-16  |
| Table 7-8.  | RSM_PM Table Entry . . . . .   | 7-17  |
| Table 7-9.  | RSM_PM Table Field Descriptions . . . . .                                    | 7-17  |
| Table 9-1.  | Memory Bank Size . . . . .   | 9-3   |
| Table 9-2.  | Memory Size in Bytes . . . . .   | 9-6   |
| Table 10-1. | Processor Interface Pins . . . . .   | 10-3  |
| Table 10-2. | Standalone Interface Pins . . . . .  | 10-14 |
| Table 11-1. | EEPROM Fields . . . . .  | 11-9  |
| Table 12-1. | ATM Physical Interface Mode Select (FRCFG[1:0]) . . . . .                    | 12-3  |
| Table 12-2. | ATM Physical Interface Mode Select (UTOPIA1) . . . . .                       | 12-3  |
| Table 12-3. | UTOPIA Mode Signals . . . . .  | 12-4  |
| Table 12-4. | Slave UTOPIA Mode Interface Signals . . . . .                                | 12-5  |
| Table 12-5. | Cell Format 8 Bit Mode . . . . .   | 12-7  |
| Table 12-6. | Cell Format 16 Bit Mode . . . . .  | 12-8  |
| Table 12-7. | Cell Format, Tagging Enabled, 8 Bit Mode . . . . .                           | 12-8  |
| Table 12-8. | Cell Format, Tagging Enabled, 16 Bit Mode . . . . .                          | 12-9  |
| Table 13-1. | SEG_VCC_INDEX Format Table . . . . .   | 13-2  |
| Table 13-2. | RSM_ROUTE_TAG Format Table . . . . .   | 13-2  |
| Table 14-1. | Type Abbreviation Description . . . . .                                      | 14-1  |
| Table 14-2. | CN8236 Control and Status Registers . . . . .                                | 14-2  |
| Table 14-3. | 0x1c0—Host Processor Interrupt Status Register 0 (HOST_ISTAT0) . . . . .     | 14-30 |
| Table 14-4. | 0x1c4—Host Processor Interrupt Status Register 1 (HOST_ISTAT1) . . . . .     | 14-31 |
| Table 14-5. | 0x1d0—Host Interrupt Mask Register 0 (HOST_IMASK0) . . . . .                 | 14-32 |
| Table 14-6. | 0x1d4—Host Interrupt Mask Register 1 (HOST_IMASK1) . . . . .                 | 14-33 |
| Table 14-7. | 0x1e0—Local Processor Interrupt Status Register 0 (LP_ISTAT0) . . . . .      | 14-35 |

|              |   |       |
|--------------|---|-------|
| Table 14-8.  | 0x1e4—Local Processor Interrupt Status Register 1 (LP_ISTAT1) . . . . .     | 14-36 |
| Table 14-9.  | 0x1f0—Local Processor Interrupt Mask Register 0 (LP_IMASK0) . . . . .       | 14-37 |
| Table 14-10. | 0x1f4—Local Processor Interrupt Mask Register 1 (LP_IMASK1) . . . . .       | 14-38 |
| Table 14-11. | PCI Configuration Register . . . . .  | 14-39 |
| Table 14-12. | PCI Register Configuration Register Field Descriptions. . . . .             | 14-40 |
| Table 14-13. | PCI Command Register. . . . .   | 14-42 |
| Table 14-14. | PCI Status Register . . . . .   | 14-42 |
| Table 14-15. | PCI Special Status Register . . . . .                                       | 14-43 |
| Table 14-16. | EEPROM Register . . . . .   | 14-43 |
| Table 15-1.  | Table of Values for Segmentation Control Register Initialization . . . . .  | 15-1  |
| Table 15-2.  | Table of Values for Segmentation Internal Memory Initialization. . . . .    | 15-2  |
| Table 15-3.  | Table of Values for Segmentation SAR Shared Memory Initialization . . . . . | 15-3  |
| Table 15-4.  | Table of Values for Scheduler Control Register Initialization . . . . .     | 15-5  |
| Table 15-5.  | Table of Values for Sch SAR Shared Memory Initialization . . . . .          | 15-6  |
| Table 15-6.  | Table of Values for Reassembly Control Register Initialization . . . . .    | 15-8  |
| Table 15-7.  | Table of Values for Reassembly Internal Memory Initialization . . . . .     | 15-10 |
| Table 15-8.  | Table of Values for Reassembly SAR Shared Memory Initialization. . . . .    | 15-11 |
| Table 15-9.  | Table of Values for General Control Register Initialization . . . . .       | 15-15 |
| Table 16-1.  | PCI Bus Interface Timing Parameters . . . . .                               | 16-1  |
| Table 16-2.  | UTOPIA Interface Timing Parameters . . . . .                                | 16-4  |
| Table 16-3.  | Slave UTOPIA Interface Timing Parameters . . . . .                          | 16-5  |
| Table 16-4.  | System Clock Timing. . . . .  | 16-7  |
| Table 16-5.  | SRAM Organization Loading Dependencies . . . . .                            | 16-9  |
| Table 16-6.  | SAR Shared Memory Output Loading Conditions . . . . .                       | 16-9  |
| Table 16-7.  | CN8236 Memory Interface Timing . . . . .                                    | 16-10 |
| Table 16-8.  | PHY Interface Timing (PROCMODE = 1) . . . . .                               | 16-13 |
| Table 16-9.  | Synchronous Processor Interface Timing . . . . .                            | 16-14 |
| Table 16-10. | Local Processor Memory Interface Timing . . . . .                           | 16-16 |
| Table 16-11. | Absolute Maximum Ratings . . . . .  | 16-19 |
| Table 16-12. | DC Characteristics. . . . .   | 16-20 |
| Table 16-13. | Pin Description (Numeric List) (1 of 4) . . . . .                           | 16-25 |
| Table 16-14. | Pin Description (Alphabetic List) (1 of 4) . . . . .                        | 16-29 |
| Table 16-15. | Spare Pins Reserved for Inputs. . . . .                                     | 16-32 |
| 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 . . . . .                                      | A-5   |
| Table A-4.   | Timing Specifications . . . . .   | A-13  |

# 1.0 CN8236 Product Overview

---

## 1.1 Introduction

The CN8236 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 CN8236 service-level features provide system designers with capabilities of accelerating specific protocol interworking functions. These features include, for example, Virtual FIFO buffer 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 CN8236 complies with *ATM Forum Traffic Management Specification, TM 4.1*. The CN8236 provides traffic shaping for all service categories:

- CBR, Variable Bit Rate (VBR)—both single and dual leaky bucket
- Unspecified Bit Rate (UBR)
- Available Bit Rate (ABR)
- GFC—both controlled and uncontrolled flows
- Guaranteed Frame Rate (GFR), that is, 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 CN8236's architecture is designed to minimize and control host traffic congestion. The host manages the CN8236 terminal with an efficient architecture that uses write-only control and status queues. For example, the host submits data for transmit by writing buffer descriptor pointers to one of 32 transmit queues. These entries can be thought of as task lists for the *ServiceSAR* to perform. The CN8236 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 CN8236 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. EPD is supported for VCCs that exceed their resource allotments. On transmit, peers are assigned fixed or round-robin priority to ensure predictable servicing. The host can implement a congestion notification algorithm for ABR with a simple one-word write to a SAR control register. The SAR reduces the Explicit Rate (ER) field or sets the Congestion Indication (CI) bit in Turnaround Resource Management (RM) cells, based on user configuration.

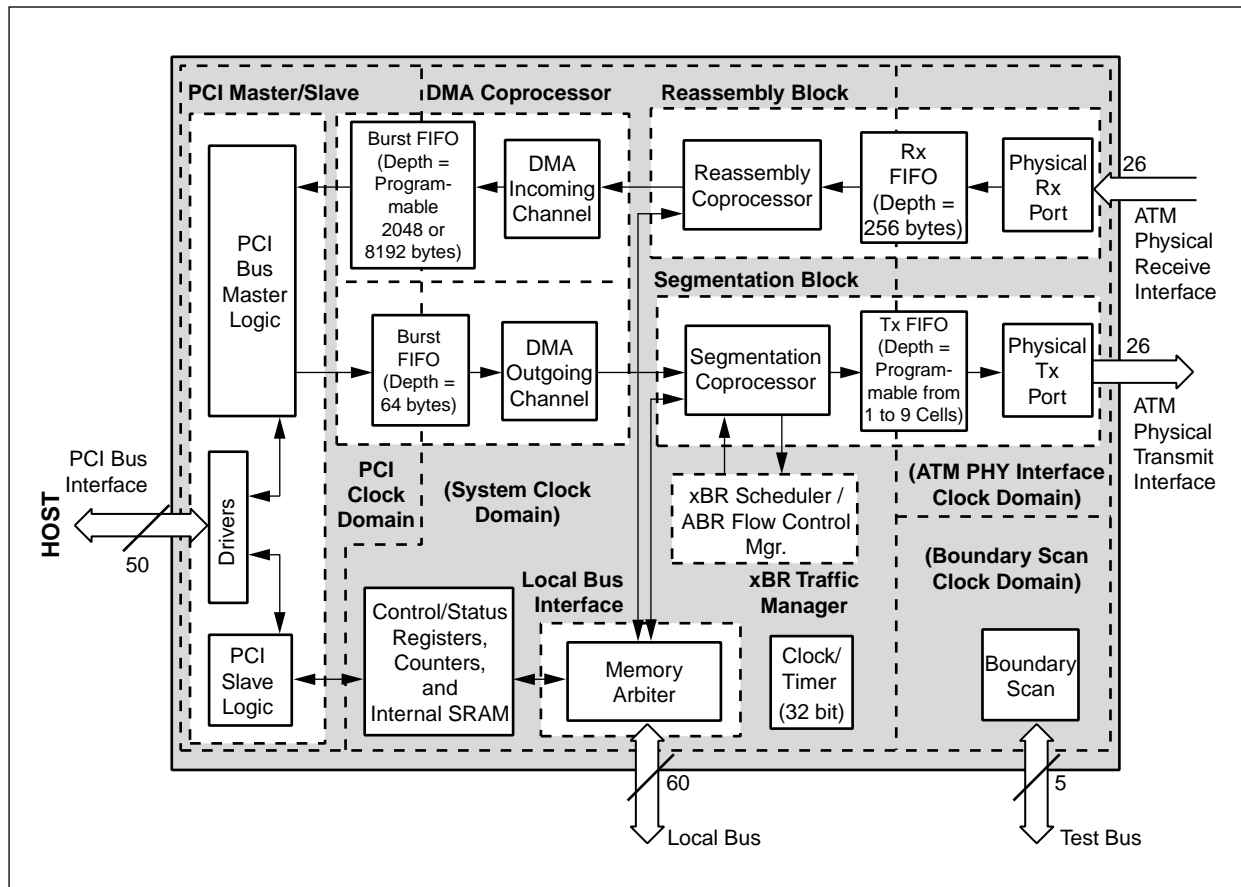
The CN8236 consists of five separate coprocessors:

- Incoming DMA,
- Outgoing DMA,
- Reassembly,
- Segmentation, and
- xBR Traffic Manager

Each coprocessor maintains state information in shared, off-chip memory. This memory is controlled by the SAR through the local bus interface, which arbitrates access to the bus between the various coprocessors. Although these coprocessors run off the same system clock, they operate asynchronously from each other. Communication between the coprocessors takes place through on-chip FIFO buffers or through queues in SAR-shared memory (that is, memory local to the SAR and accessible both to the SAR and the host).

The CN8236'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. CN8236 Functional Block Diagram



8236\_001

## 1.2 Service-Specific Performance Accelerators

The CN8236 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 CN8236 handles both User-Network Interface (UNI) addresses, which use an 8-bit Virtual Path Identifier (VPI) field, and Network-to-Network Interface (NNI) addresses, which use 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 CN8236 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 CN8236 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 CN8236 allows for are as follows:

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

### *Guaranteed Frame Rate*

The CN8236 can rate-shape ATM Adaptation Layer Type 5 (AAL5) Common Part Convergence Sublayer Protocol Data Units (CPCS-PDUs, that is, frames) in the UBR service category, by providing a guaranteed MCR for UBR VCCs.

### *Early Packet Discard*

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

- Frame Relay packet discard based on the DE field in the received frame and the channel exceeding a user-defined priority threshold.
- Packet discard based on the CLP bit.
- 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 buffer full condition/threshold.
- Various AAL3/4 Management Information Base (MIB) errors.

### ***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 CN8236 to segment traffic from a fixed PCI address (that is, a Virtual FIFO buffer) for circuit-based CBR traffic.

The user can delineate up to sixteen 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, 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 ER algorithms are used 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 MCR and ICR fields reduce the number of ABR templates needed in local memory.
- The CN8236 allows rate adjustments on Turnaround RM cells, based on congestion in the host.
- The CN8236 allows rate adjustments due to use-it-or-lose-it behavior.
- The CN8236 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 CN8236 optionally posts the current Allowed Cell Rate (ACR) on the segmentation status queue for the host monitoring functions.

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

The CN8236 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 CN8236 can interleave segmentation of numerous VCCs (that is, 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 CN8236 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.

***Optional Local Processing of ATM Management Traffic***

The CN8236's Local Processor Interface allows for an optional local processor to direct segmentation and reassembly of ATM management level traffic, such as Operations and Maintenance (OAM) cells, Performance Monitoring (PM) cells, signalling, and Interim Local Management Interface (ILMI) traffic. This off-loads network control traffic from the host, thereby focusing host processing power on the user application.

***Internal SNMP MIB Counters***

CN8236 has three internal counters that measure cells received, cells discarded, and AAL5 PDUs discarded (to meet ILMI and RFC1695 requirements).

***CompactPCI Hot Swap***

Circuitry and I/O have been added in to functionally comply with the *CompactPCI Hot Swap Specification (PICMG 2.1 R1.0)*. Refer to sections 7.2 and 3.1.8 of the specification for details of the Hot Swap operation. The HSWITCH\* input indicates the state of the handle switch. A logic low indicates that the handle is locked, whereas a logic high indicates that the handle is unlocked. The HLED\* output is a 12 mA open drain capable of driving an LED directly. A logic low illuminates the LED. The HENUM\* output is an 8 mA open drain in compliance with the ENUM# signal defined in the CompactPCI specification.



## 1.3 Designer Toolkit

The CN8236 ATM evaluation environment provides evaluation capability for the CN8236 *ServiceSAR*. 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:

- rapid customer product development
- hardware reference design
- software reference design (based on VxWorks)
- traffic generation and checking capability

Comprising part of this development environment is the CN8236/8250EVM, a PCI card specifically designed to be a full-featured ATM controller implementing the full functionality of the CN8236 *ServiceSAR*. The CN8236 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 (CN823xHPI), a software driver to the CN8236, 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 CN8236. 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, and artwork for the CN8236EVM PCI card.



## 2.0 Architecture Overview

---

### 2.1 Introduction

The CN8236 *ServiceSAR* architecture efficiently handles 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. The second section describes the segmentation and reassembly functions 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 and Performance Monitoring (OAM/PM), the various device inputs and outputs, and the logic diagram with pin descriptions.

This architectural overview serves as a solid foundation for understanding the complete functionality of the CN8236.

## 2.2 High Performance Host Architecture with Buffer Isolation

Once initialized and given a segmentation or reassembly task, the CN8236 operates autonomously. Because the CN8236 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

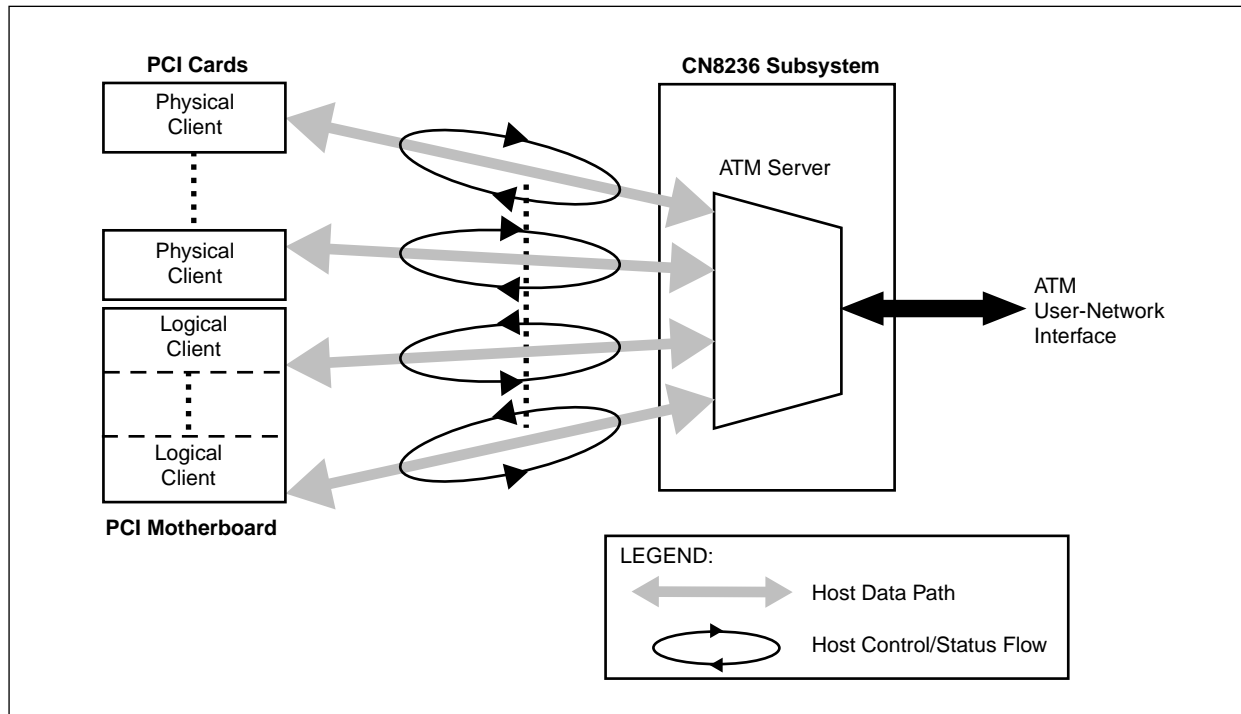
The CN8236, functioning as an ATM UNI, provides a high throughput uplink to a broadband network. Most individual ATM service users (or clients) do not have the bandwidth requirements to equal the throughput capability of the CN8236. As an application example, ATM clients can be Ethernet or Frame Relay ports. Therefore, many 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 CN8236 functions as an ATM server for up to 32 clients. In this way, the bandwidth requirements of the service user (the client) can be balanced with the service throughput capability of the CN8236 and the rest of the specific system.

The CN8236 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 PHY 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 CN8236 independently. Due to its server architecture, the CN8236 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 PHY entities. All communicate directly with the CN8236.

Figure 2-1. Multiple Client Architecture Supports Up To 32 Clients



8236\_002

## 2.2.2 CN8236 Queue Structure

The flow of the reassembly, scheduling, and segmentation processes in the CN8236 is monitored, coordinated, and controlled through the use of a full array of circular queues, serviced by the CN8236 or the host.

The following queues exist in local memory:

- Transmit queues (up to 32 queues)
- Reassembly/segmentation queue
- Free buffer queues (up to 32 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 CN8236 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-PM cells to be transmitted and as data on ABR-class received Backward\_RM and Forward\_RM cells. The RSM/SEG queue is a private queue for the SAR that 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.

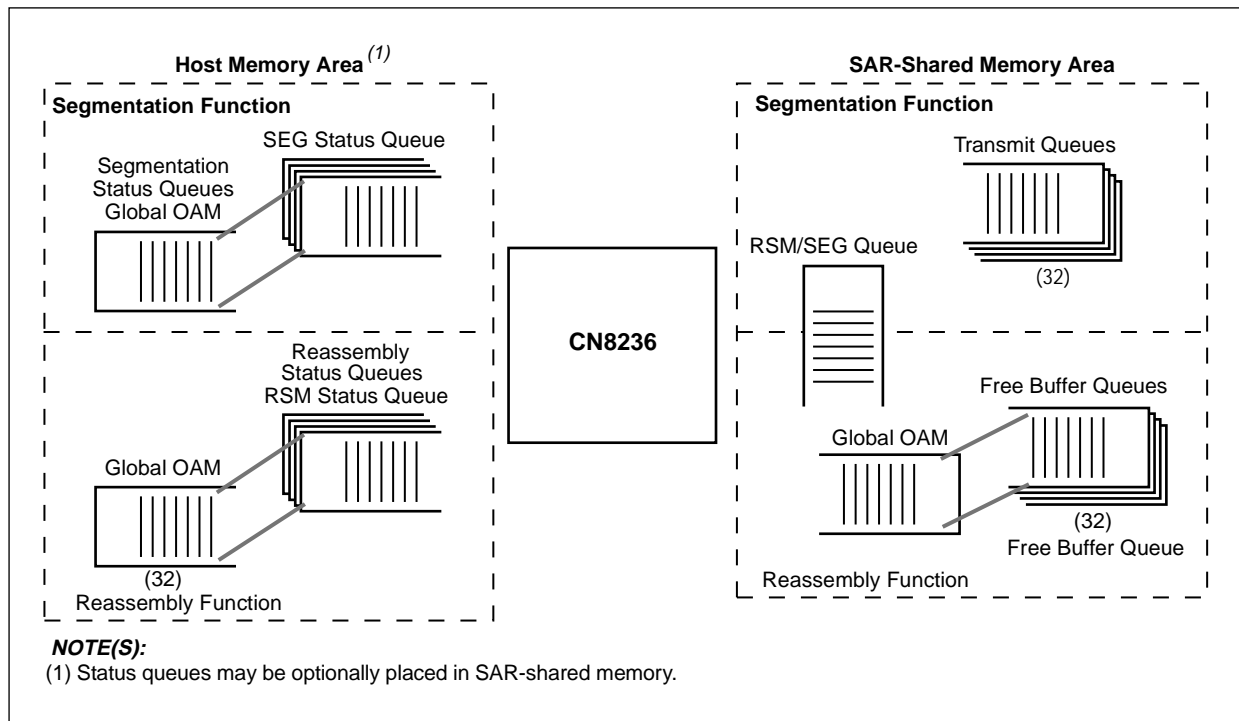
The following queues exist in host (or optionally, SAR-shared) memory:

- segmentation status queues (up to 32 queues)—includes the Global OAM segmentation status queue
- reassembly status queues (up to 32 queues)—includes the Global OAM reassembly status queue

The CN8236 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 CN8236 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. CN8236 Queue Architecture

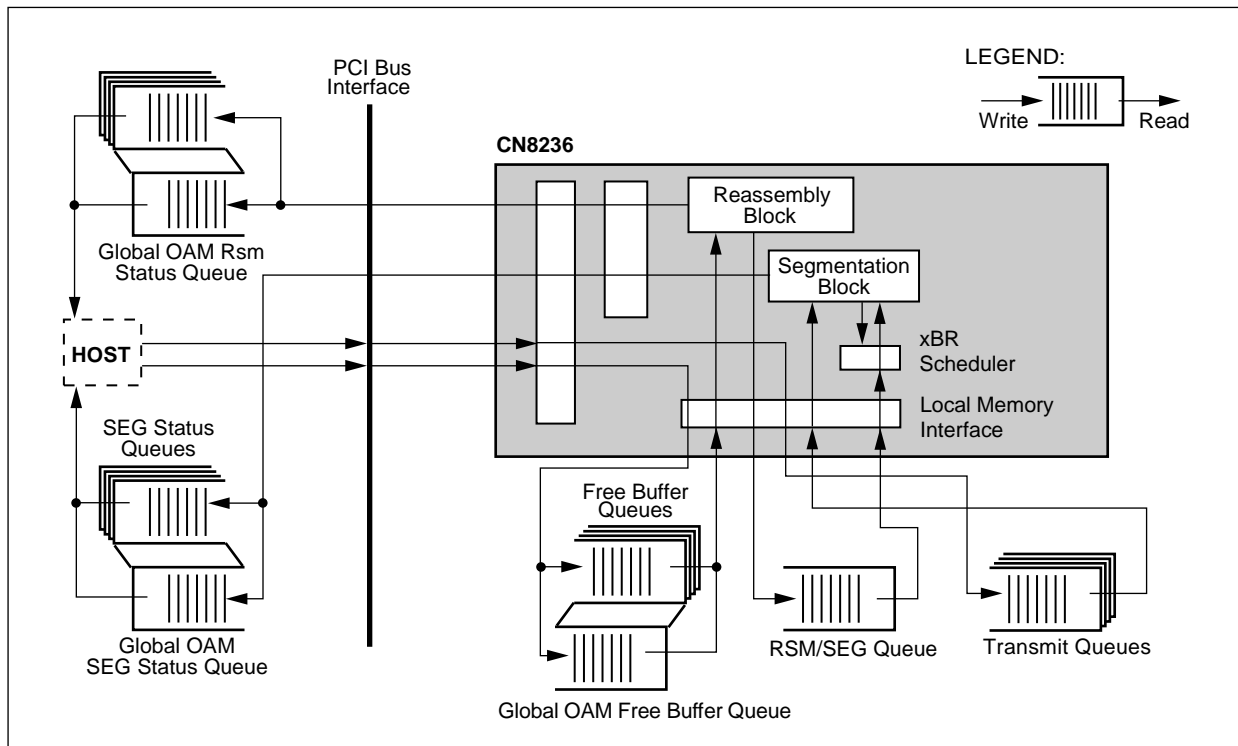


8236\_096

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 CN8236 during processing and associate each of the major functional blocks of the CN8236 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 CN8236 Functional Blocks



8236\_003

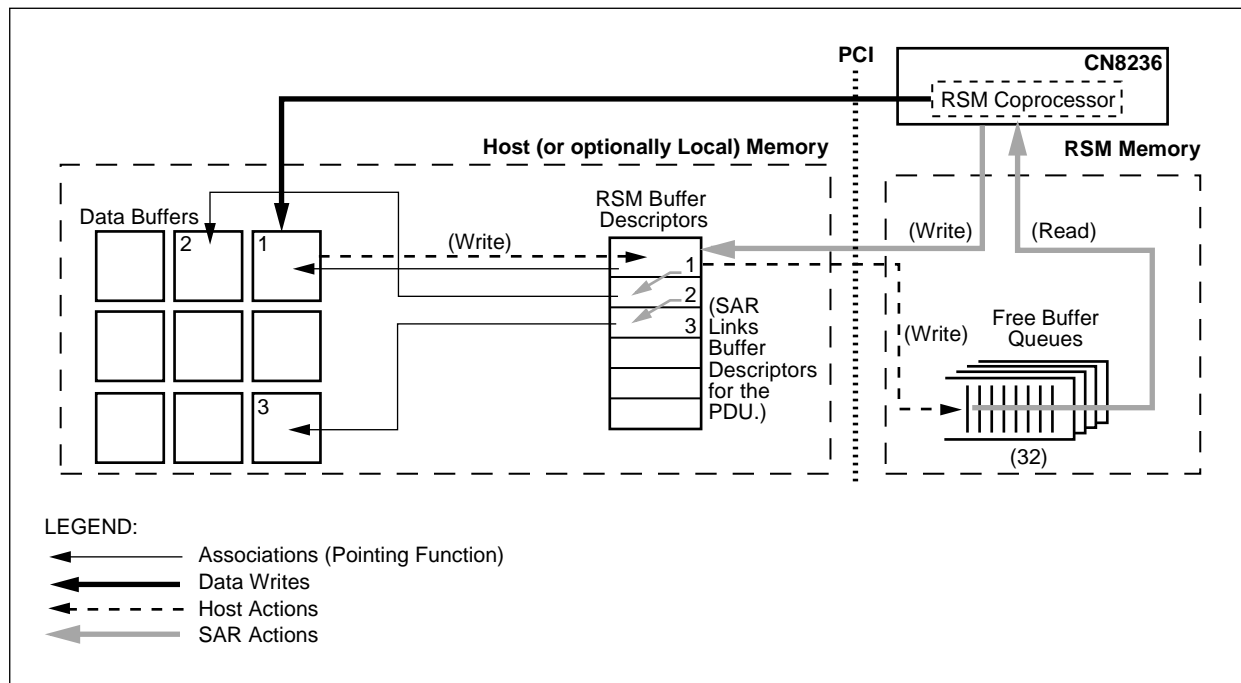
### 2.2.3 Buffer Isolation Utilizing Descriptor-Based Buffer Chaining

The CN8236 uses buffer structures for reassembly and segmentation. The buffer structures 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 separately 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



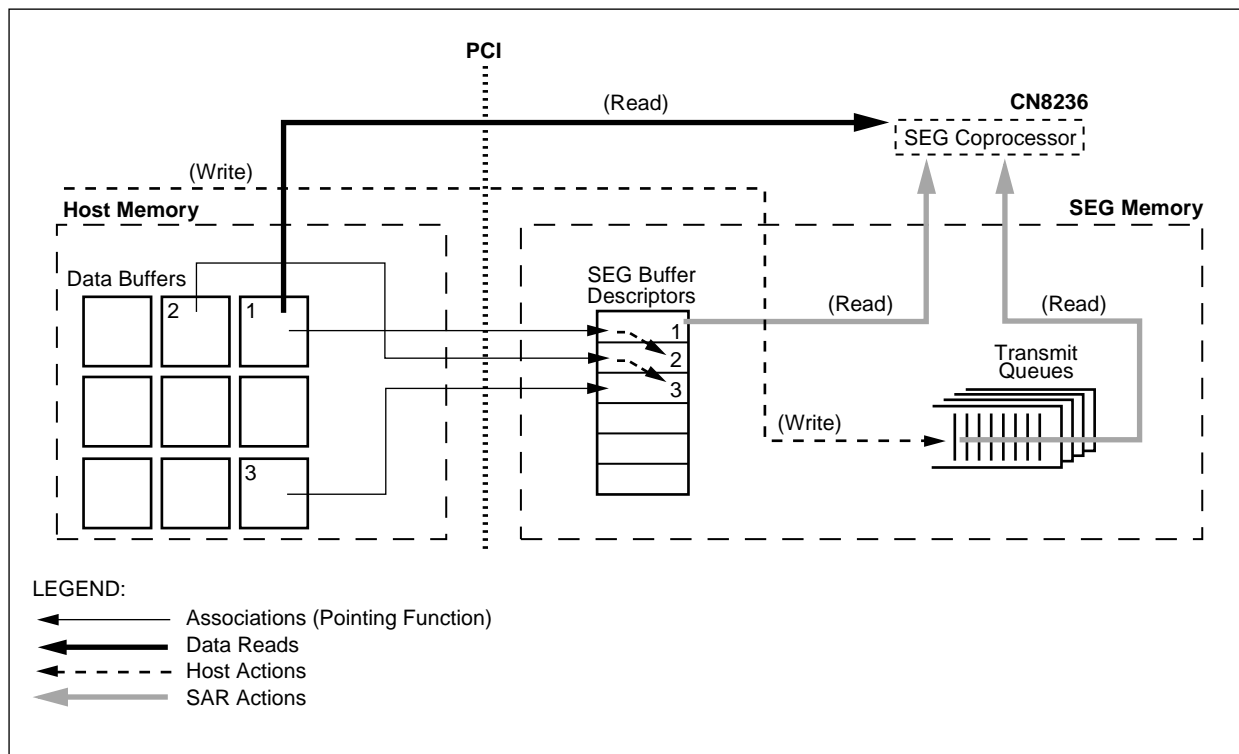
8236\_004



Figure 2-5 illustrates how the host submits linked data buffers (that is, 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 2-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 buffer for segmentation task pointers.

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



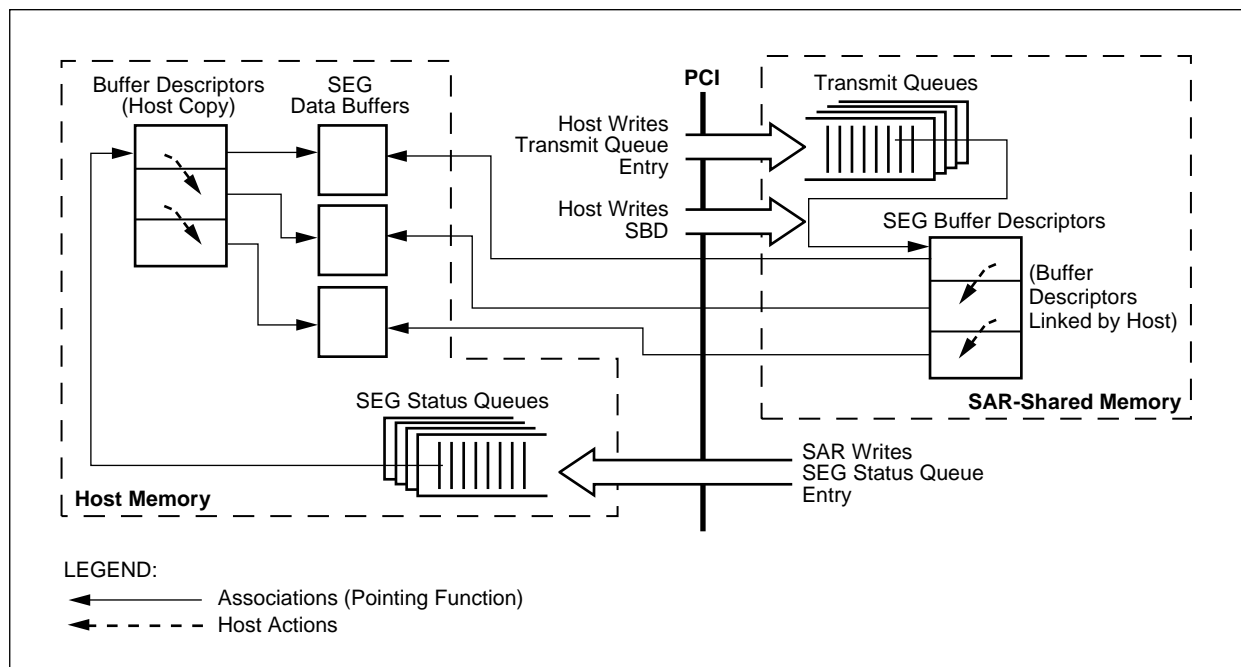
8236\_005

### 2.2.4 Status Queue Relation to Buffers and Descriptors

The status queues employed by the CN8236 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.

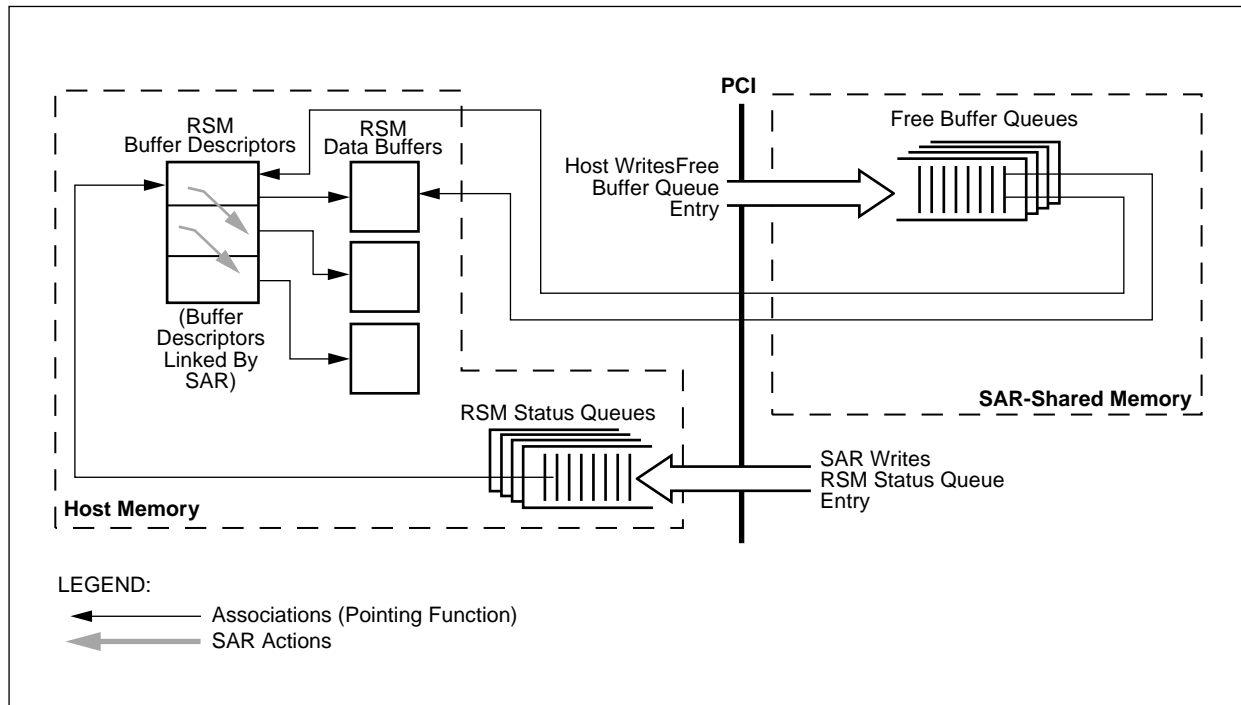
Figure 2-6. Segmentation Status Queues Related to Data Buffers and Descriptors



8236\_006

Figure 2-7 illustrates the association between the reassembly status queues and reassembly data buffers and descriptors. The host submits free buffers to the SAR by writing pointers to them in the free buffer queue entries. The SAR links the buffer descriptors pointing to the three data buffers containing the reassembled PDU, and writes the RSM status queue entry containing the pointer to the first buffer descriptor for that PDU. The host further processes the PDU using that data.

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

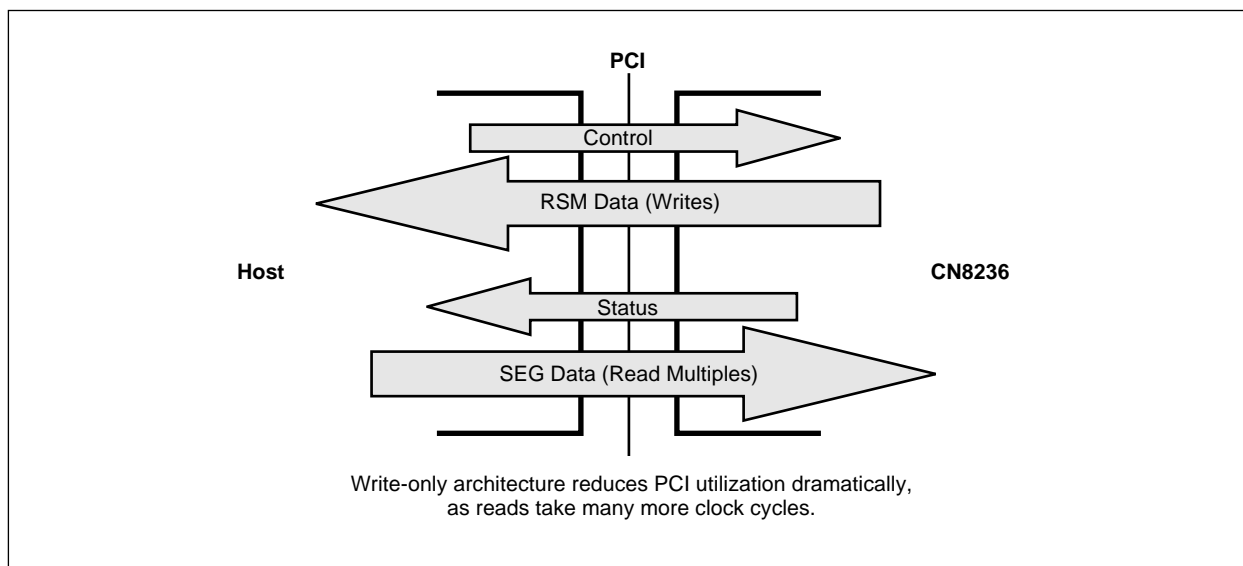


8236\_007

### 2.2.5 Write-only Control/Status

Figure 2-8 illustrates the CN8236's write-only PCI control architecture. The host manages the CN8236 ATM terminal using write-only control and status queues. This architecture minimizes PCI bus utilization by eliminating reads from control activities. PCI writes use the bus much more efficiently than PCI reads. During a PCI write, the Bus Master can post the write data to an internal FIFO buffer 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 CN8236 eliminates read operations except for burst reads to gather segmentation data.

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



8236\_008

### 2.2.6 Scatter/Gather DMA

The CN8236'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 14-word burst accesses for data, or 1- to 4-word accesses for control and status messages.

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

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

## 2.2.7 Interrupts

The CN8236 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 CN8236 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 data communications processing.

## 2.3 Automated Segmentation Engine

The CN8236 can segment up to 64 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 CN8236 provides full support of the AAL5 and AAL3/4 protocols and 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 VC or 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 that point to the buffers containing the PDU to be transmitted, and then submitting that chained message to the SAR by writing to one of 32 independent circular transmit queues.

The segmentation coprocessor then operates autonomously, formatting the cells on each channel according to the host-defined segmentation VCC table entries for each channel. The 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.
- Each AAL3/4 cell carries 44 octets of payload and four octets (in five fields) of header and trailer information. The SAR performs the formatting steps necessary to create AAL3/4 cells.
- 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, where 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 buffer to the PHY interface for transmission.

The system designer can set the depth of the transmit FIFO buffer from one to nine cells deep to optimize the balance between Cell Delay Variation (CDV). This increases with longer transmit FIFO buffer depth and PCI latency protection, which decreases with shorter transmit FIFO buffer depth.

The CN8236 provides a method to segment traffic from a fixed PCI address (or Virtual FIFO buffer). This is intended for circuit-based CBR traffic such as voice channels.

The CN8236 reports segmentation status to the host on one of a set of 32 independent parallel segmentation status queues. The CN8236 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 PHY 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 CN8236 can reassemble data.

Using a dynamic Channel Directory lookup method, the CN8236 reassembles up to 64 K VCCs simultaneously at a maximum rate of 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 CN8236 supports AAL5, AAL3/4, and AAL0 reassembly and 52-octet Raw Cell mode.

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

For AAL3/4, the reassembly coprocessor performs all error detection and checking procedures incorporated in AAL3/4, and reassembles SDUs based on Message ID (MID).

The CN8236 provides two methods of terminating an AAL0 PDU:

1. Payload Type Identifier (PTI) termination, where the PTI bit in the cell header is monitored for the End of Message (EOM) cell indication.
2. Cell Count termination, where the CN8236 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 CN8236 provides the user with generous per-channel control of the reassembly process, including the following:

- 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 CN8236 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 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 buffer full condition occurs.
- Early packet discard on channels encountering a reassembly status queue full (overflow) condition.
- Early packet discard on AAL3/4 channels with these MIB errors: ST\_ERR (Segment Type error), SN\_ERR (Sequence Number error), and LI\_ERR (SAR-PDU Length error).

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 CN8236 implements ATM's inherent robust traffic management capabilities for CBR, VBR, ABR, UBR, GFR, and GFC. The CN8236 manages each VCC independently and dynamically.

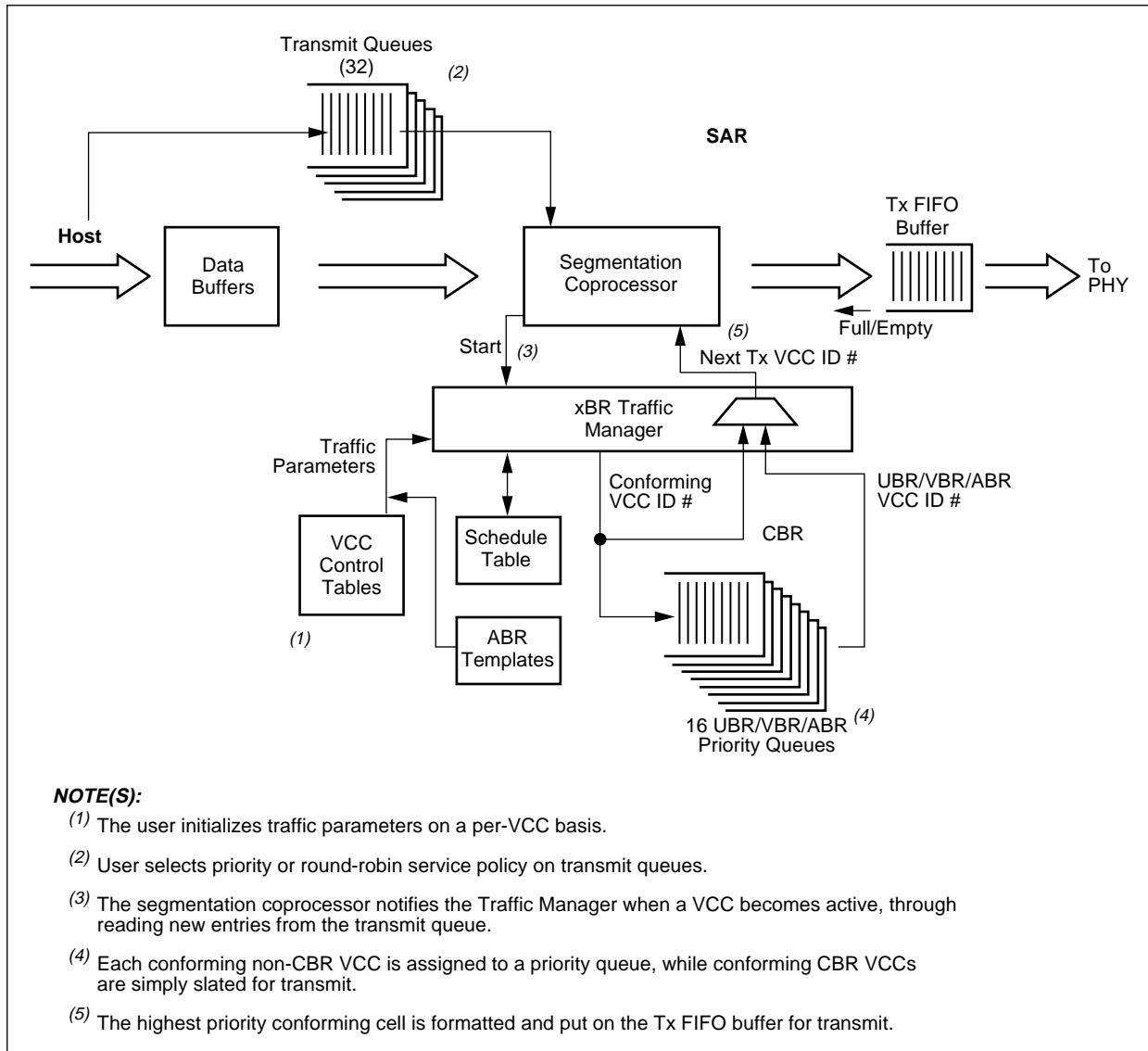
- The user assigns each connection a service class, a priority level, and a rate if applicable. Then, the on-chip traffic controller, the xBR Traffic Manager, optimizes use of the line bandwidth according to the VCC's traffic parameters and control information stored in SAR-shared memory. The xBR Traffic 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 CN8236 provides 16 segmentation priorities. The user configures these priorities for the remaining service categories, including the *TM 4.1*-defined ABR class. The CN8236's xBR Traffic Manager implements multiple functional levels of traffic prioritizing. This is illustrated in [Figure 2-9](#).
- The host submits data to be sent by writing entries to the transmit queues for segmentation. The SAR processes these transmit queues either in round-robin order (transmit queue 0 through 31, looped back to 0), or in priority order (with transmit queue 31 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 CN8236 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 0 being the lowest priority and priority 15 being highest).

Figure 2-9 shows this prioritization as a global set of queues, but it is actually maintained on a per-transmit opportunity basis. In this way, high priority traffic is transmitted up to its GCRA limits but does not block lower priority traffic when idle. The CN8236 asynchronously multiplexes traffic based on the above schemes as the Tx FIFO buffer empties.

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



8236\_009

### 2.5.1 CBR Traffic

The CBR service category requires guaranteed transmission rates, and constrained CDV. The CN8236 facilitates these needs when generating CBR traffic by pre-assigning specific schedule slots to CBR VCCs. For each CBR-assigned cell slot, the CN8236 generates a cell for that specific VCC unless data is not available. The CN8236 also minimizes CDV by basing all traffic management on a local reference clock.

The CN8236 provides a mechanism to exactly match the scheduled rate of a CBR channel to the rate of its data source. To accomplish this rate-matching, the host can occasionally instruct the xBR Scheduler to skip one transmit opportunity on a channel.

The CN8236 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 CN8236 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). VBR3 (also called VBRC) controls PCR and CDVT on all cells, but controls SCR on only CLP = 0 (high priority) cells.

### 2.5.3 ABR Traffic

The CN8236 implements the ATM Forum ABR flow control algorithms. The CN8236 acts as 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 CN8236 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 CN8236 injects an in-rate stream of Forward RM cells for each ABR VCC. When these cells return to the CN8236's receive port as Backward RM cells after a round trip through the network, the CN8236 processes these cells and uses the data returned as feedback to dynamically adjust the rates on each ABR channel.

The CN8236 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 nonreal-time applications that 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 covered previously 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 CN8236 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 to which 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 CN8236, the Bt8230 SAR, proved the core algorithms. The CN8236 extends algorithms use to other service classes. The CN8236 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 CN8236 dynamically manages the entire table.

Some key features of the xBR Scheduler are as follows:

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 CN8236 fairly distributes the link bandwidth to channels based upon their QoS parameters and assigned transmission priority. As covered previously, the CN8236 supports 16 priorities in addition to preallocated CBR time slots.

The xBR Scheduler facilitates advanced network traffic management topologies. The CN8236 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 CN8236 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 define conformance ABR behavioral responses to network and connection states. The CN8236 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 *TM 4.1* specification.
3. Mindspeed provides the initial templates, which can be customized by the user later, shortening development time.

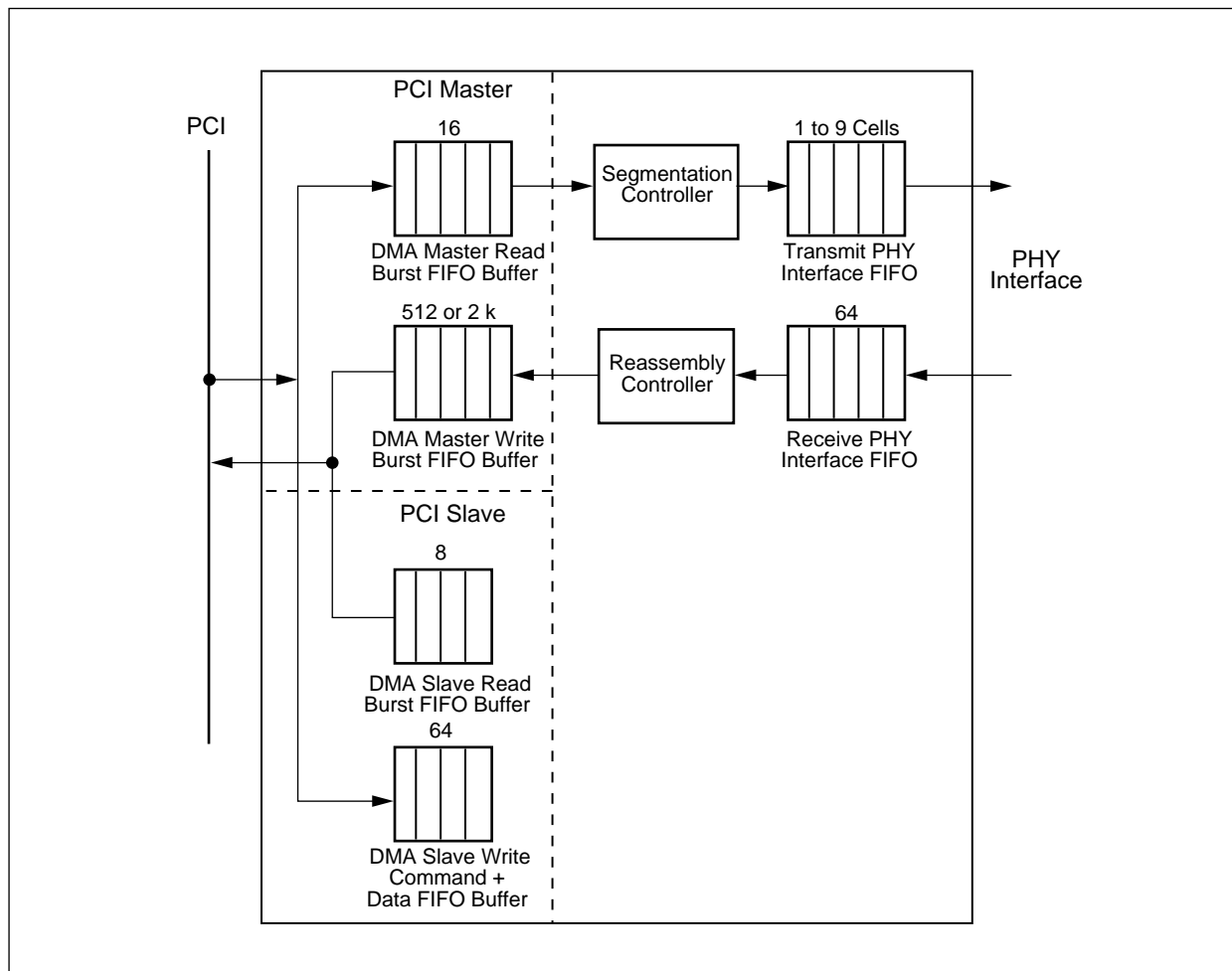
## 2.6 Burst FIFO Buffers

To conserve local memory bandwidth, the CN8236 does not use its local SRAM as a buffer for incoming or outgoing data. Instead, the CN8236 uses six dedicated internal FIFO buffers data as follows:

- Two DMA master burst FIFO buffers (read = 16 words; write, 512 or 2 K words, programmable via CONFIG1 bit 1).
- Two DMA slave burst FIFO buffers, (read = 8 words and write = 64 words).
- One FIFO buffer between the PHY interface and the segmentation coprocessor (1 to 9 cells). See [Section 4.2.4](#).
- One FIFO buffer between the PHY interface and the reassembly coprocessor (64 words).

Figure 2-10 illustrates the data FIFO buffer.

Figure 2-10. Data FIFO Buffers



8236\_097



## 2.7 Implementation of OAM-PM Protocols

The CN8236 provides internal support for the detection and generation of OAM traffic, including PM OAM.

The CN8236 supports the F4 and F5 OAM flows according to I.610. It monitors up to 128 channels and generates in-rate PM-OAM cells.

The CN8236 includes a Local Processor Interface, providing the capability of SAR-shared memory segmentation and reassembly. The user can thus route OAM traffic, including PM traffic, to and from this optional local processor, thereby off-loading ATM network management from the host. To facilitate this, the CN8236 provides the option of user-defined global status queues for both segmentation and reassembly and a global buffer queue for reassembly, to which the user can assign SAR-shared memory addresses. The CN8236 then processes OAM traffic via the local processor, thereby isolating the host from these management functions and focusing host processing power on ATM user data traffic.

## 2.8 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 for 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 CN8236's ATM PHY interface communicates with and controls the ATM link interface device, which carries out all the transmission convergence and PHY media-dependent functions defined by the ATM protocol. Two modes of operation are provided: standard UTOPIA and slave UTOPIA. Standard UTOPIA mode conforms to both UTOPIA Level 1 and Level 2 standards for ATM Layer devices. 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 CN8236 integrates a complete memory controller designed for direct interface to common Static RAMs (SRAMs). The CN8236'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.

### ***Local Processor I/O***

The Local Processor Interface in the CN8236 allows an optional external CPU to be directly connected to the device to serve as a local controlling intelligence that can handle initialization, connection management, overall data management, error recovery, and OAM functions. The use of a local processor for these functions allows ATM message data to flow to and from host system memory in a substantially larger bandwidth, because the local processor is handling the out-of-band functions described above.

The processor interface is loosely coupled, meaning that the processor connects to the CN8236 through bidirectional transceivers and buffers for the address and data buses. This allows the processor fast access to CN8236 memory and registers, but insulates the CN8236 from processor instruction and data cache fills. It also allows the processor to control multiple CN8236s or PHY devices if desired.

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

The CN8236 includes five pins for Joint Test Action Group (JTAG) Boundary Scan, for board-level testing. The CN8236 incorporates an internal loopback from the segmentation coprocessor to the reassembly coprocessor to facilitate system diagnostics.

## 2.9 Electrical/Mechanical

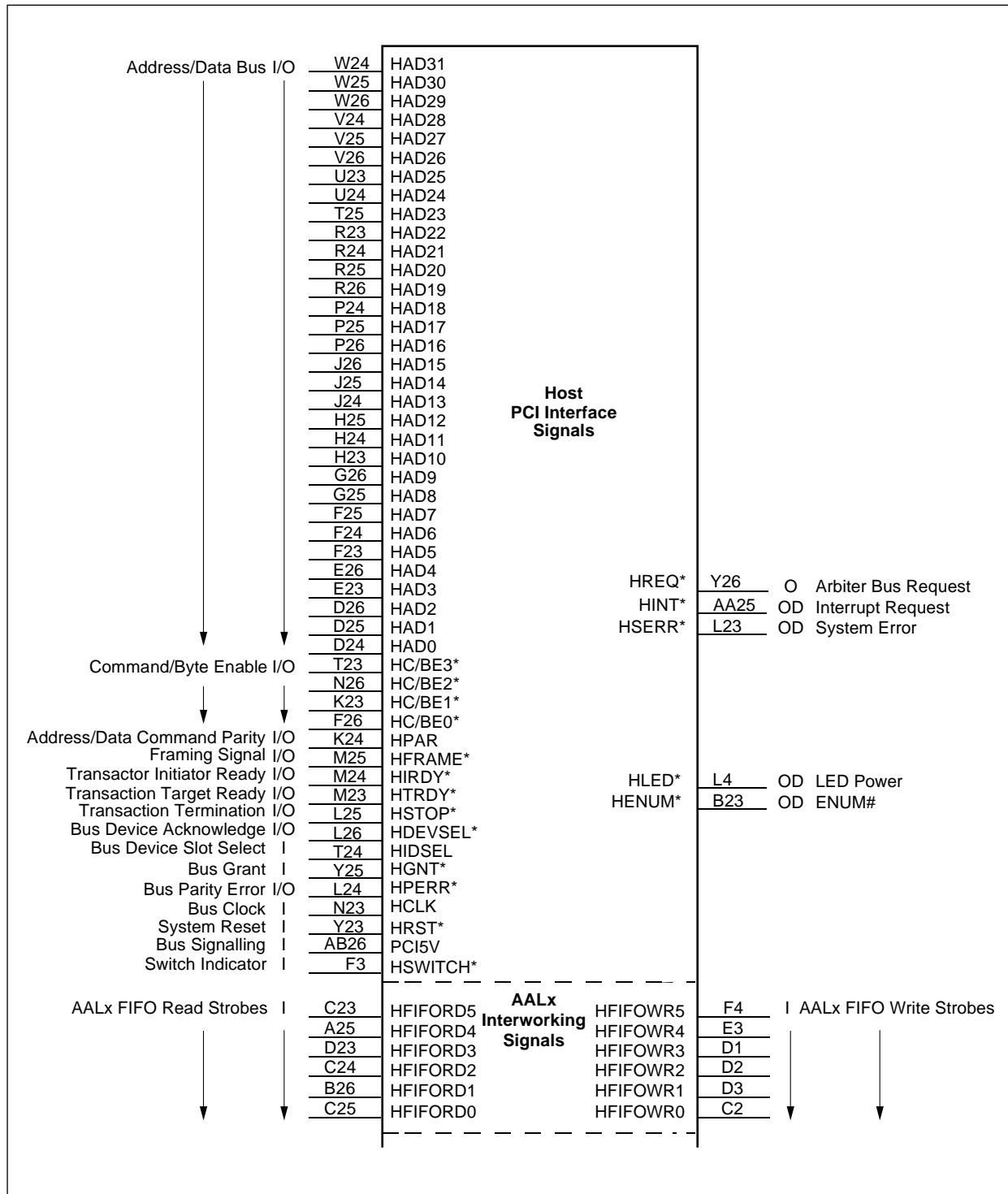
The CN8236 is a CMOS device packaged in a 388-Pin Ball Grid Array (BGA) format. It operates from a 3.3 V power supply and within the standard industrial temperature range.

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

## 2.10 Logic Diagram and Pin Descriptions

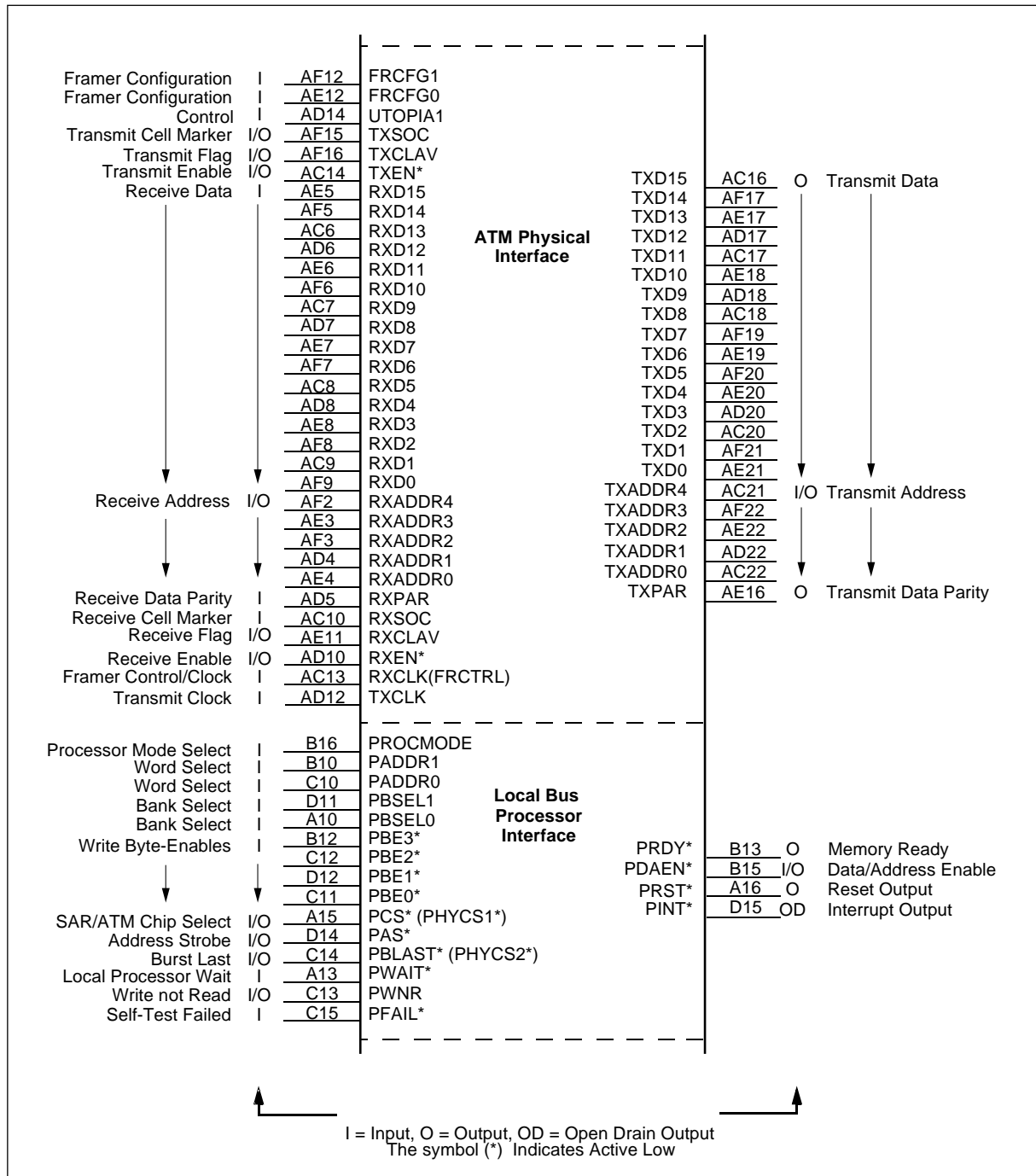
A functionally partitioned logic diagram of the CN8236 is illustrated in [Figure 2-11](#). Pin descriptions, names, and input/output assignments are detailed in [Table 2-1](#).

Figure 2-11. CN8236 Logic Diagram (1 of 3)



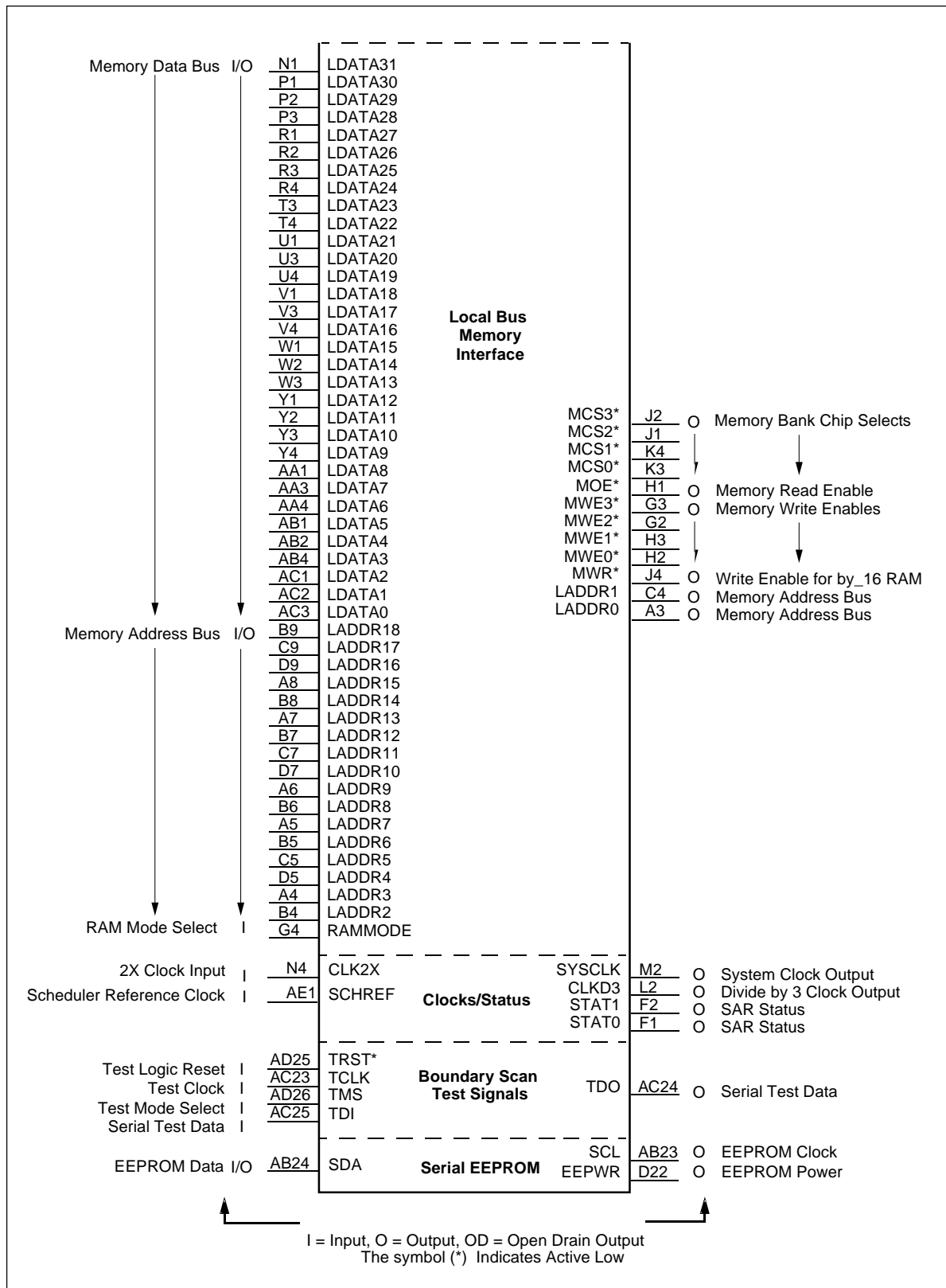
8236\_098

Figure 2-11. CN8236 Logic Diagram (2 of 3)



8236\_099

Figure 2-11. CN8236 Logic Diagram (3 of 3)



8236\_100

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

|                            | Pin Label   | Signal Name                  | I/O | Definition  |
|----------------------------|-------------|------------------------------|-----|---|
| Host PCI Interface Signals | HAD[31:0]   | Multiplexed Address/Data Bus | I/O | Used by the PCI host or CN8236 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 CN8236 is acting as a target) or driven (when the CN8236 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 ends the current transaction.   |
|                            | HIRDY*      | Transaction Initiator Ready  | I/O | Used by the transaction initiator or bus master (either the CN8236 or the PCI host) to indicate ready for data transfer. A valid data transfer occurs when both HIRDY* and HTRDY* are active on the same clock edge.  |
|                            | HTRDY*      | Transaction Target Ready     | I/O | Used by the transaction target or bus slave (either the CN8236 or the PCI bus memory) to indicate that it is ready for a data transfer. A valid data transfer occurs when both HIRDY* and HTRDY* are active on the same clock edge.   |
|                            | HSTOP*      | Transaction Termination      | I/O | Driven by the current target or slave (either the CN8236 or the PCI bus memory) to abort, disconnect, or retry the current transfer. The HSTOP* line is used by the PCI master in conjunction with the HTRDY* and HDEVSEL* lines to determine the type of transaction termination.  |
|                            | HDEVSEL*    | Bus Device Acknowledge       | I/O | Driven by a target to indicate to the initiator that the address placed on the HAD[31:0] lines (together with the command on the HC/BE[3:0]* lines) has been decoded and accepted as a valid reference to the target's address space. Once asserted, it is held by the CN8236 (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 CN8236 that it is being selected for a configuration space access.  |
|                            | HREQ*       | Arbiter Bus Request          | O   | Asserted by the CN8236 to request control of the PCI bus.   |
|                            | HGNT*       | Bus Grant                    | I   | Asserted to indicate to the CN8236 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).   |
|                            | HINT*       | Interrupt Request            | OD  | Signals an interrupt request to the PCI host, and is tied to the INTA_ line on the PCI bus.   |

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

|                            | Pin Label    | Signal Name             | I/O | Definition   |
|----------------------------|--------------|-------------------------|-----|--|
| Host PCI Interface Signals | HPERR*       | Bus Parity Error        | I/O | Driven asserted by the CN8236 (as a bus slave) or by a target addressed by the CN8236 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 CN8236 is a bus slave or sampled when the CN8236 is a bus master on the second clock edge after a valid data phase. The CN8236 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 CN8236 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 CN8236 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.  |
|                            | HSWITCH*     | Switch Indicator        | I   | Logic low means switch locked, logic high means switch unlocked. Signal pulled up internally. Compact PCI Hot Swap Signal.   |
|                            | HLED*        | LED Power               | OD  | 12 mA open drain. Logic low turns on LED. Compact PCI Hot Swap Signal.   |
|                            | HENUM*       | ENUM#                   | OD  | 8 mA open drain. Compact PCI Hot Swap Signal.  |
| AALx                       | HFIFORD[5:0] | AALx FIFO Read Strokes  | I   | AALx ingress FIFO buffer read strobe. This signal is edge-detected inside the SAR so no setup/hold time required. Signals pulled up internally.  |
|                            | HFIFOWR[5:0] | AALx FIFO Write Strokes | I   | AALx egress FIFO buffer write strobe. This signal is edge-detected inside the SAR so no setup/hold time required. Signals pulled up internally.  |
| ATM PHY Interface          | FRCFG[1:0]   | Framer Configuration    | I   | Configuration pins FRCFG[1,0] determine what framer interface the CN8236 supports.<br>00 = Reserved; do not use<br>01 = UTOPIA interface<br>10 = Slave UTOPIA interface<br>11 = Reserved; do not use   |
|                            | TxData[15:0] | Transmit Data           | O   | Carries outgoing data bytes to the framer chip in all framer modes (8 mA drive).   |
|                            | TxAddr[4:0]  | Transmit Address        | I/O | UTOPIA Transmit address (8 mA drive).  |
|                            | TxPar        | Transmit Data Parity    | O   | Outputs the 8-bit odd parity computed over the TxData[15:0] lines in all framer modes (8 mA drive).  |
|                            | TxSOC        | Transmit Cell Marker    | I/O | In both UTOPIA and slave UTOPIA modes, the TxSOC line is asserted by the CN8236 when the starting byte of a 53-byte cell is being output. (aka TxMark)   |



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

|                   | Pin Label            | Signal Name         | I/O  | Definition   |
|-------------------|----------------------|---------------------|--|--|
| ATM PHY Interface | 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 CN8236 transmit buffer is empty. (Has pulldown resistor to pull inactive in master mode when not driven externally.) (aka TxFlag*) (8 mA drive)   |
|                   | TxEn*                | Transmit Enable     | I/O  | Indicates that valid data has been placed on the TxData[15:0], TxPar, and TxSOC lines in the current clock cycle when the CN8236 is in UTOPIA or slave UTOPIA mode. This pin is an output in UTOPIA mode and an input in slave UTOPIA mode. (8 mA drive. Has pullup resistor to pull inactive in slave mode when not driven externally.)   |
|                   | TxCik                | Transmit Clock      | I  | UTOPIA Transmit Clock. Has an internal pullup to V <sub>DD</sub> (CMOS level).   |
|                   | RxAddr[4:0]          | Receive Address     | I/O  | UTOPIA receive address (8 mA drive).   |
|                   | RxData[15:0]         | Receive Data        | I  | Transfers incoming data bytes from the link interface or framer chip to the CN8236 in all framer modes.  |
|                   | RxPar                | Receive Data Parity | I  | Should be driven with the 8-bit odd parity computed over the RxData[7:0] lines by the link interface or framer chip in all framer modes.   |
|                   | RxSOC                | Receive Cell Marker | I  | Indicates that the current byte being transferred on the RxData[7:0] lines is the starting byte of a 53-byte cell. Has internal pulldown resistor. (aka RxMark)  |
|                   | 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 RxData[7:0], RxPar, and RxSOC lines are invalid. In slave UTOPIA mode, this pin indicates to the framer chip that the receive FIFO buffer in the CN8236 is full. 8 mA drive. Has pulldown resistor to pull inactive in master mode when not driven externally. (aka RxFlag*) |
|                   | RxEn*                | Receive Enable      | I/O  | In UTOPIA mode, RXEN* indicates that the CN8236 is ready to receive data on the RxData[15:0], RxPar, and RxSOC lines in the next clock cycle. This pin is an output in UTOPIA mode and an input in slave UTOPIA mode. Has pullup resistor to pull inactive in slave mode when not driven externally. 8 mA drive.   |
| RxCik             | Framer Control/Clock | I                   | In UTOPIA and slave UTOPIA mode, the RxCLK line should be driven with a clock that is synchronous to that used by the framer device for interfacing to the CN8236. The TxData[15:0], TxPar, TxSOC, TxClav, TxEn*, RxData[15:0], 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. When MULTI_CLK is set to a 1 in the CONFIG1 register, the Tx side of the UTOPIA interface is synchronized to the TxClk. |  |

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

|                               | Pin Label            | Signal Name  | I/O | Definition  |
|-------------------------------|----------------------|--|-----|---|
|                               | UTOPIA1              | UTOPIA Mode Select   | I   | Selects level 1/level 2 operation. In level 1 mode, address pins are forced to be outputs independent of master/slave mode. This input has a pullup resistor so the default is UTOPIA level 1 mode. When UTOPIA1 input is a logic high, the UTOPIA address signals, TxAddr[4:0] and RxAddr[4:0], are forced as outputs.                                       |
| Local Bus Processor Interface | PROCMODE             | Processor Mode Select  | I   | When grounded, this input selects the local processor mode. When pulled to a logic high, the standalone mode is selected.   |
|                               | PADDR[1:0]           | Word Select Inputs   | I   | The PADDR[1,0] inputs are connected to the word select field of the CPU address bus (address bits [3, 2] for the Intel i80960CA processor, which can perform 4-word burst transactions). These inputs are used by the CN8236 to allow single-cycle bursts to be performed without requiring very short memory access times.                                   |
|                               | PBSEL[1:0]           | Bank Select Inputs   | I   | Select one of four banks of memory to be accessed. They are decoded by the memory controller to generate the appropriate chip/bank selects to the external memory.  |
|                               | PBE[3:0]*            | Write Byte-Enables   | I   | Supplies byte enables for each local processor memory access. These pins are only relevant during writes by the local processor to SAR-shared memory. Each byte enable line corresponds to a specific byte lane in the LDATA[31:0] data bus: PBE[0]* corresponds to LDATA[7:0], PBE[1]* to LDATA[15:8], PBE[2]* to LDATA[23:16], and PBE[3]* to LDATA[31:24]. |
|                               | PCS*<br>(PHYCS1*)    | SAR Chip Select<br>ATM PHY Chip Select<br>(in standalone mode) | I/O | In local processor mode with PROCMODE tied low, PCS* is the SAR chip select input. In standalone mode, this pin is PHYCS1*, which can be connected to the chip select input of the Mindspeed PHY device.  |
|                               | PAS*                 | Address Strobe   | I/O | Indicates a local processor address cycle. In standalone mode, PAS* is used to drive the AS* pin of the Mindspeed PHY device.   |
|                               | PWNR                 | Write/not Read   | I/O | The PWNR input indicates the direction of a local processor transfer. A logic 1 indicates a write; a logic 0 indicates a read. During standalone mode, this output provides the same function for the Mindspeed PHY device.   |
|                               | PWAIT*               | Processor Wait   | I   | Used by the local processor or external logic to insert wait states for read or write transactions.   |
|                               | PBLAST*<br>(PHYCS2*) | Burst Last<br>ATM PHY Chip Select<br>(in standalone mode)      | I/O | In local processor mode, this input is used by the processor to indicate the end of a transaction. During standalone mode, this output is a second chip select, PHYCS2*.  |
|                               | PRDY*                | Memory Ready   | O   | Signals that the memory or control register has accepted the data on a write, or that data is available to latch by the local processor on a read cycle.  |

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

|                               | Pin Label   | Signal Name               | I/O | Definition  |
|-------------------------------|-------------|---------------------------|-----|---|
| Local Bus Processor Interface | PDAEN*      | Data/Address Enable       | I/O | Connected to the output enable input of the bidirectional transceivers and buffers used to isolate the CN8236 data and address bus from the local processor. In standalone mode, this input is connected to the PHY device's interrupt output(s). |
|                               | PFAIL*      | Self-Test Failed          | I   | The local processor can indicate a failure of its internal self-test or initialization processes by asserting the PFAIL* input to the CN8236.   |
|                               | PINT*       | Interrupt Output          | OD  | Asserted by the CN8236 to the local processor to signal an interrupt request in local processor mode.   |
|                               | PRST*       | Reset Output              | O   | Asserted by the CN8236 to the local processor 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, and control and status register access by the local processor.  |
|                               | LADDR[18:2] | Memory Address Bus        | I/O | Address I/O bus. Used for memory reads and writes, and control and status register access by the local processor.   |
|                               | LADDR[1:0]  | Memory Address Bus        | O   | The two least significant bits of address I/O bus. Used for memory reads and writes, and control and status register access by the local processor.   |
|                               | MCS[3:0]*   | Memory Bank Chip Selects  | O   | Selects one of four addressable banks of SRAM memory.   |
|                               | MOE*        | Memory Read Enable        | O   | Indicates that a read cycle is proceeding and the memory device output buffers should be enabled, driving data onto the LDATA[31:0] lines.  |
|                               | MWE[3:0]*   | Memory 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.<br>1 = by_16 memory devices<br>0 = by_4 or by_8 memory devices   |
| Clocks/Status                 | CLK2X       | 2x Clock Input            | I   | Double frequency (from SYSCLK) CMOS level input (66 MHz maximum).   |
|                               | SYSCLK      | System Clock Output       | O   | This divide by 2 of CLK2X is the internal system clock and 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 maximum).   |
|                               | STAT[1,0]   | SAR Status                | O   | CN8236 internal status outputs. Internal status controlled by the STAT_MODE[4:0] field in the CONFIG0 Register.   |
|                               | SCHREF      | Scheduler Reference Clock | I   | External Scheduler Reference Clock.   |

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

|                            | Pin Label | Signal Name                    | I/O | Definition  |
|----------------------------|-----------|--------------------------------|-----|---|
| Boundary Scan Test Signals | TRST*     | Test Logic Reset               | I   | When a logic low, this signal asynchronously resets the boundary scan test circuitry and puts the test controller into the reset state. This state allows normal system operation. Tie to ground when boundary scan is not implemented. 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 I <sup>2</sup> C data.   |
|                            | SCL       | EEPROM Clock                   | O   | Serial I <sup>2</sup> C clock.  |
|                            | EEPWR     | EEPROM Power                   | O   | 12 mA drive. Direct Buffer of HRST* input.  |
| Supply Voltage             | VDD       | Power                          | —   | Forty-nine balls are provided for supply voltage.   |
|                            | VSS       | Ground                         | —   | Eighty-five balls are provided for ground. (The 36 balls in the center help in heat dissipation.)   |
|                            | VGG       | ESD Protection – Voltage Clamp | I   | Provides Electrostatic Discharge (ESD) protection and over voltage protection. When the device is used with 5 V devices on the board, tie this pin to 5 V for 5 V signal tolerance. Otherwise, tie to 3.3 V.<br><br><b>NOTE:</b> The 5 V supply must be applied concurrent to the 3.3 V supply. |

## 3.0 Host Interface

---

### 3.1 Overview

The CN8236 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 CN8236's high throughput, the host must process control and status information at a comparable rate.

#### *APPLICATION EXAMPLE*

**An Ethernet Switch uses a CN8236-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 CN8236 converts 176.6 K packets/second (Kpps) to cells in each direction. Therefore, the host must process control and status information for a total of 353.2 Kpps. This packet rate equates to a packet service time of 2.83  $\mu$ s/packet.**

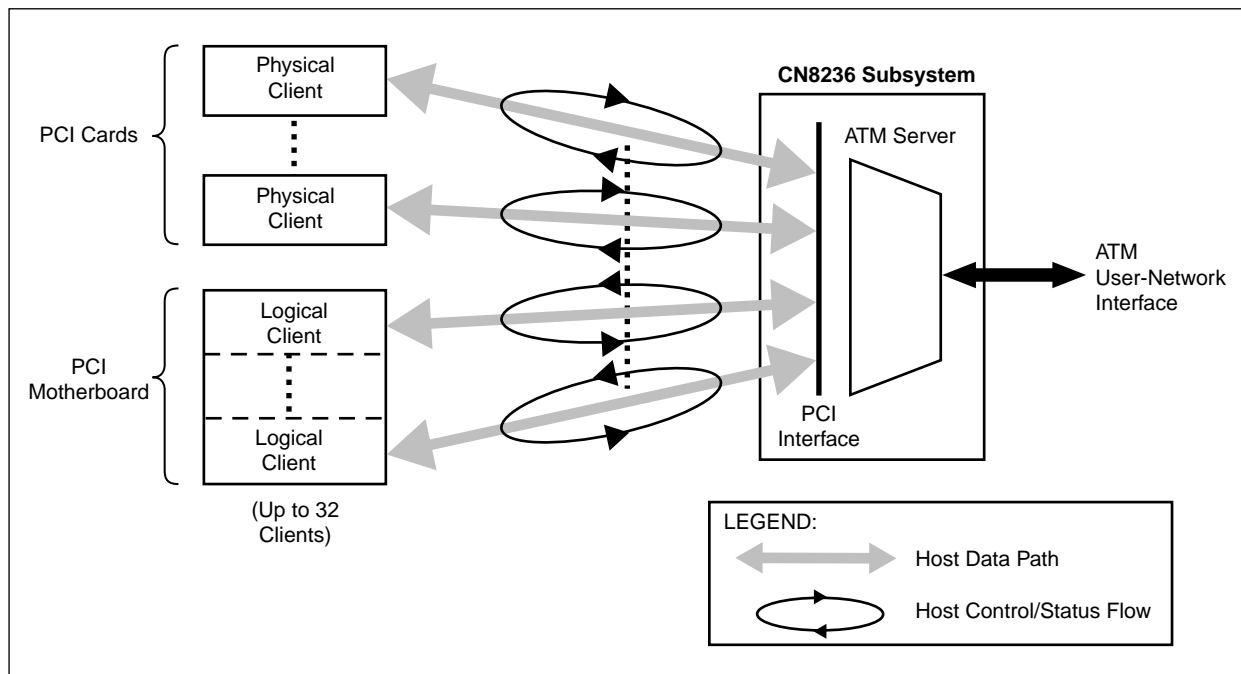
In cases such as the example above, the service rate places extraordinary performance requirements on the host system. High throughput systems require large numbers of processing cycles and efficient use of system buses.

The CN8236 provides a flexible, high-performance, host interface architecture. With this interface, the CN8236 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 CN8236 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 CN8236 provides an ATM server for up to 32 clients. Figure 3-1 illustrates this client/server relationship.

Figure 3-1. Client/Server Model of the CN8236



8236\_010

### 3.2.1 Logical Clients

As shown in Figure 3-1, the clients do not need to 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 CN8236 independently. Due to its server architecture, the CN8236 supplies the synchronization between asynchronous tasks requiring ATM services.

### 3.2.2 Resource Allocation

With either PHY PCI peers, logical peers, or some combination of the two types, the CN8236 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

Because each peer is assigned to an independent control and status path, the CN8236 isolates the resources of each peer. This simplifies resource management. In addition, queue error conditions caused by a single peer do not affect any other peers.

#### ***APPLICATION EXAMPLE***

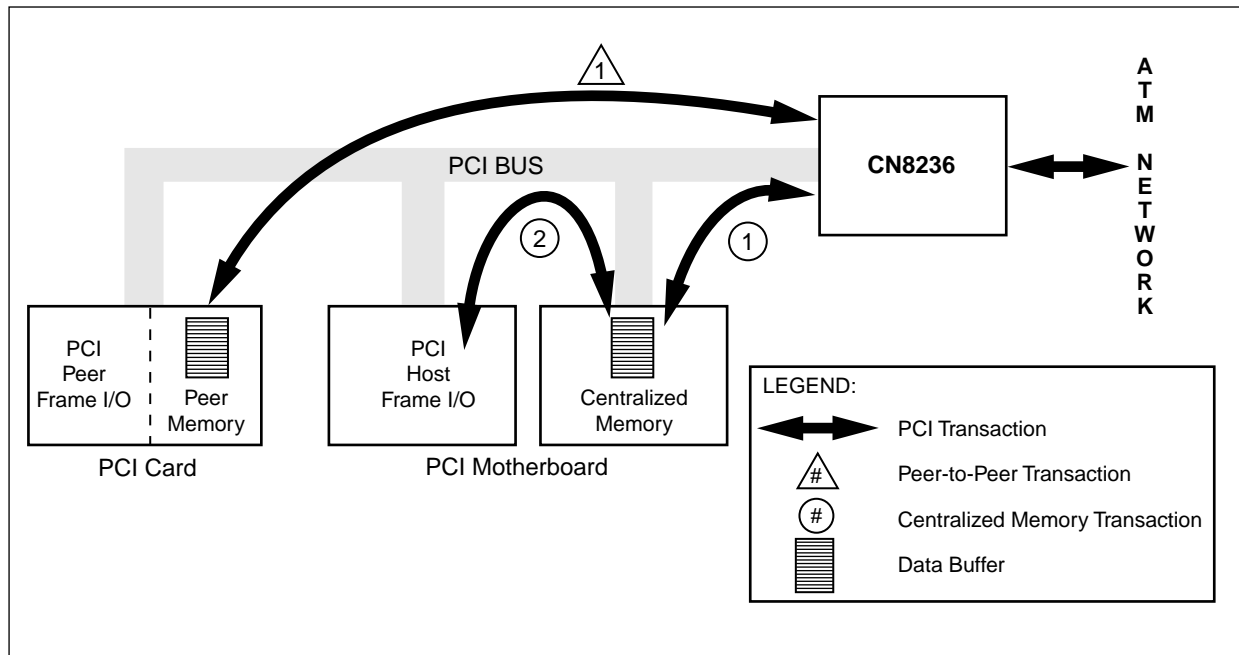
**A system designer implements a CN8236 terminal as an ATM uplink for a Service Access Multiplexer (SAM). The SAM is comprised of the ATM card and several Frame Relay adapter cards. The host assigns each Frame Relay adapter card to a set of CN8236 control and status queues at initialization. During operation, one of the Frame Relay adapter cards experiences a hardware failure. The failure prevents the card's processor from servicing the CN8236's reassembly status queue. Eventually, the CN8236 fills the queue and is unable to proceed—this situation is referred to a queue overflow. The CN8236 shuts down reassembly on VCCs that 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 CN8236 also enables peer-to-peer PCI transfers. The CN8236 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 CN8236 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. Centralized memory buffers require store-forward operations, while the peer buffers enable peer-to-peer transfers. Thus, peer-to-peer transfers reduce the use of the PCI bus.

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

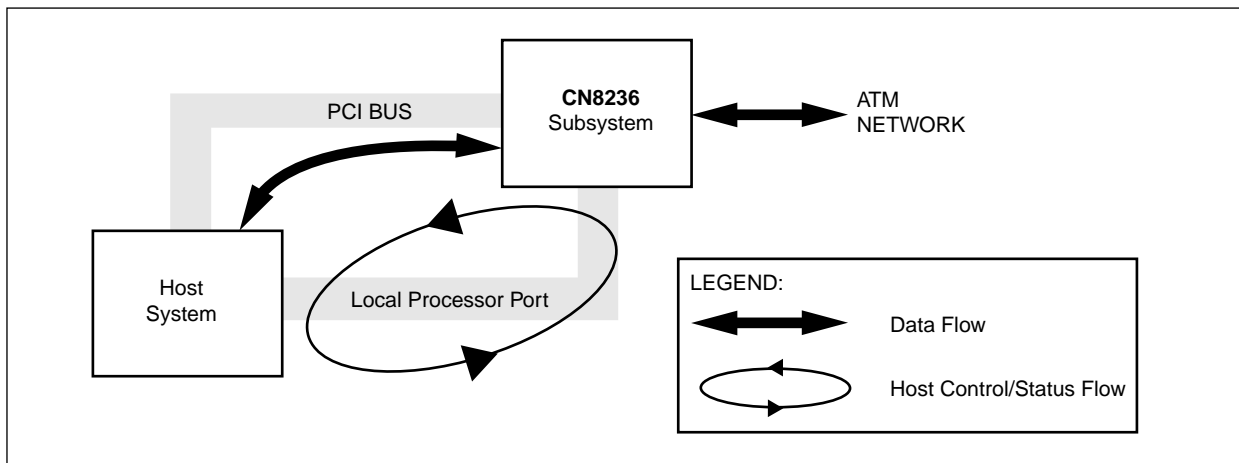


8236\_011

### 3.2.5 Local Processor Clients

The CN8236 supports limited bandwidth SAR-shared memory segmentation and reassembly. Any peer can use the local processor port instead of the PCI bus for data traffic, control, and status. Hosts can use SAR-shared memory for control and status, but transfer data across the PCI bus. This out-of-band control configuration diverts control overhead from the PCI bus, lessening the burden of ATM's high throughput and robust management requirements on the host system. [Figure 3-3](#) shows an out-of-band control architecture.

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



8236\_012



### 3.3 Write-only Control and Status

For host-based applications, the host manages the CN8236 SAR using write-only control and status queues. This architecture minimizes PCI bus use by eliminating reads from the control path. PCI writes use the bus much more efficiently than PCI reads. During a PCI write, the target can post the write data in an internal FIFO buffer, 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 CN8236's write-only architecture uses reads only for segmentation data (PDU) fetches. All control and status transactions are writes. This section describes the management of write-only queues. The purpose and entries of each class of queue are described in later chapters.

[Table 3-1](#) defines the CN8236 control and status queues.

*Table 3-1. CN8236 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 CN8236 supports 32 queues of each type. The SAR and host manage each queue independently, through queue management variables. The SAR stores its variables in internal registers called base tables. The host maintains its own variables within its driver. Each queue contains a programmable number of queue entries.

### 3.3.1.1 Control Variables

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

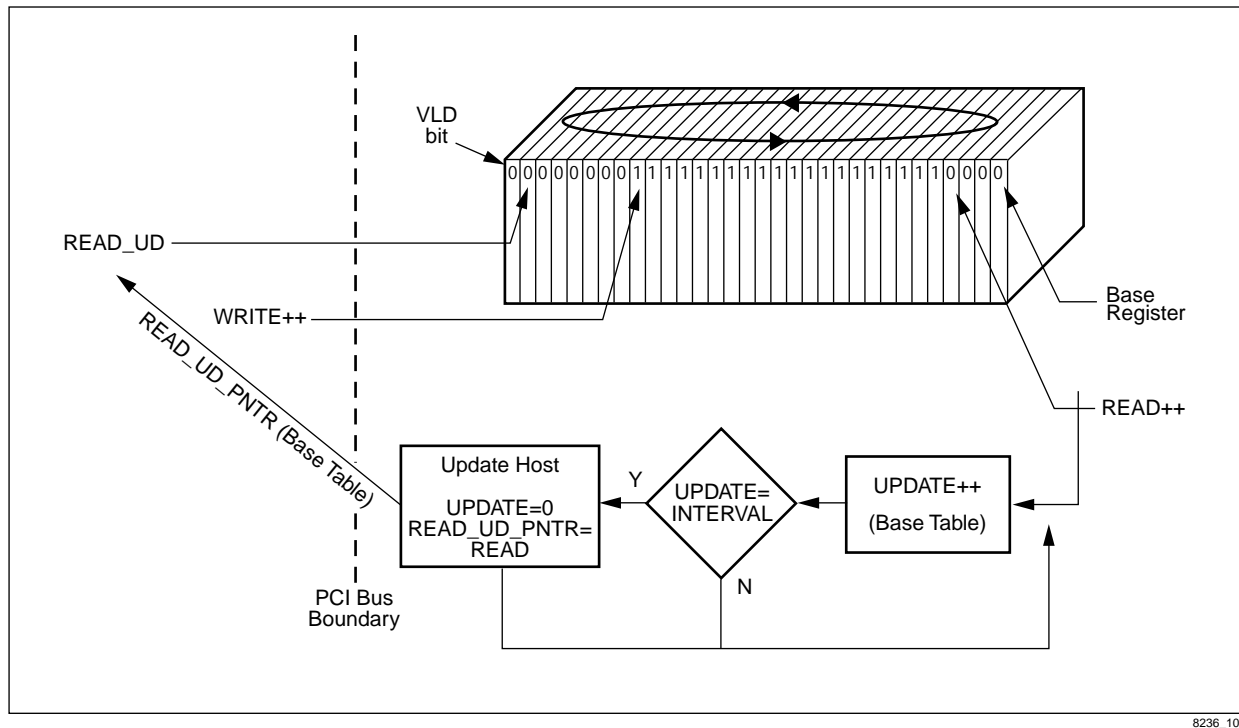
**Table 3-2. Write-only Control Queue Variables**

| Variable     | Definition  | Location                   | Initialization |
|--------------|---|----------------------------|----------------|
| WRITE        | Current host position in queue                                  | Host variable              | 0              |
| READ_UD      | Last known SAR position in queue as seen by host                | Host word aligned variable | 0              |
| READ         | Current SAR position in queue                                   | SAR Base table             | 0              |
| INTERVAL     | Number of queue entries processed by 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 3-4** illustrates the control queue management algorithm. The host initializes all of the variables described in **Table 3-2**. Once the SAR is enabled, it maintains the READ pointer. When the SAR processes a queue entry, it advances the READ pointer (READ++). Since the queues are circular, the pointer eventually wraps back to 0. 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 also introduces larger latency between host updates, which reduces the effective size of the queue.

Figure 3-4. Write-only Control Queue



8236\_101

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  $(WRITE+1) \text{ modulo } \text{size\_of\_queue} = \text{READ\_UD}$ , the host halts 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 1.

The CN8236 snoops the 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 CN8236 has processed the entry, it resets the VLD bit to 0.

### 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 CN8236 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 CN8236 provides 32 queues of each type. The SAR and host manage each queue independently through queue management variables. The SAR stores its variables in internal base table registers. The host maintains its variables in its driver. Each queue contains a programmable number of queue entries.

**3.3.2.1 Control Variables** [Table 3-3](#) describes the variables for write-only status queue management.

*Table 3-3. Write-only Status Queue Variables*

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

**3.3.2.2 Queue Management** [Figure 3-5](#) illustrates the status queue management algorithm. The host initializes all of the variables described in [Table 3-3](#).

The SAR maintains its own write pointer, WRITE. The CN8236 reports status to the host by writing a status queue entry. After it writes the entry, the CN8236 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 1 when it writes a status queue entry. The host resets it to 0 after processing an entry.



#### 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 hold-off mechanism and an event counter mechanism are implemented and work in parallel. The timer hold-off 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. Interrupts due to conditions other than status queue writes are not delayed.

##### *Timer Hold-off Mechanism*

The timer hold-off mechanism is enabled by setting INT\_DELAY (EN\_TIMER) to a logic high. The ALARM1 register is set to a value that holds off the interrupt for a specified period of time. The user initializes the CLOCK register to 0. When the value in the CLOCK register is greater than the value in the ALARM1 register, status queue interrupts are allowed to propagate to the appropriate interrupt pins, HINT\* or PINT\*. The CLOCK register is set to 0 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 is closed for STAT\_CNT number of events. When the internal counter has reached the value of STAT\_CNT, the interrupt window is opened, which allows the interrupt to propagate to the output pin. The counter is reset when the status registers are read and the interrupt output goes inactive.

## 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 PHY interface. The segmentation process converts user data (typically Ethernet, Token Ring, or Frame Relay packets) into ATM cells.

The CN8236 can segment 64 K VCCs simultaneously. The segmentation coprocessor block independently segments each channel and multiplexes the VCCs onto the line with cell level interleaving. The CN8236 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 CN8236 generates all AAL overhead for AAL3/4 and AAL5, or optionally uses a null adaptation layer, AAL0. The CN8236 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 CN8236 supports up to 64 K segmentation VCCs, referenced by a unique index, VCC\_INDEX. The VCC\_INDEX identifies a location in the Segmentation VCC table, an array of 10-word segmentation VCC channel descriptors.

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

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

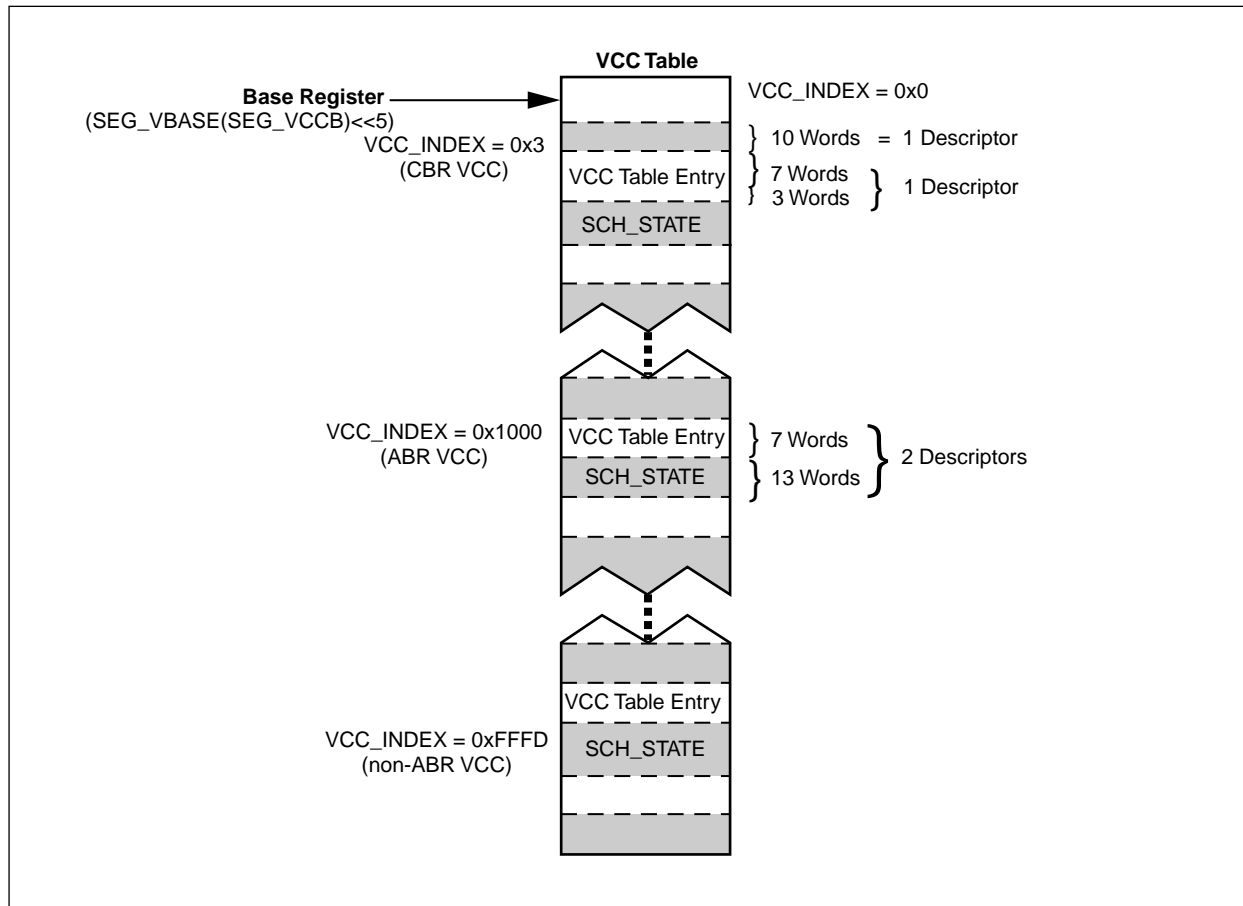
#### 4.2.1.1 Segmentation VCC Table

[Figure 4-1](#) shows a VCC table with a CBR VCC at VCC\_INDEX 3, an ABR VCC located at VCC\_INDEX 0x1000, and a non-ABR VCC at VCC\_INDEX 0xFFFF. 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 (that is, 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 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 indicates a NULL VCC.

While the CN8236 accepts 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 SAR-shared memory available in which to allocate and create segmentation VCC tables, transmit queues, etc.



Figure 4-1. Segmentation VCC Table



8236\_095

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 segmentation 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 CN8236 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](#) describes how to use these bits.

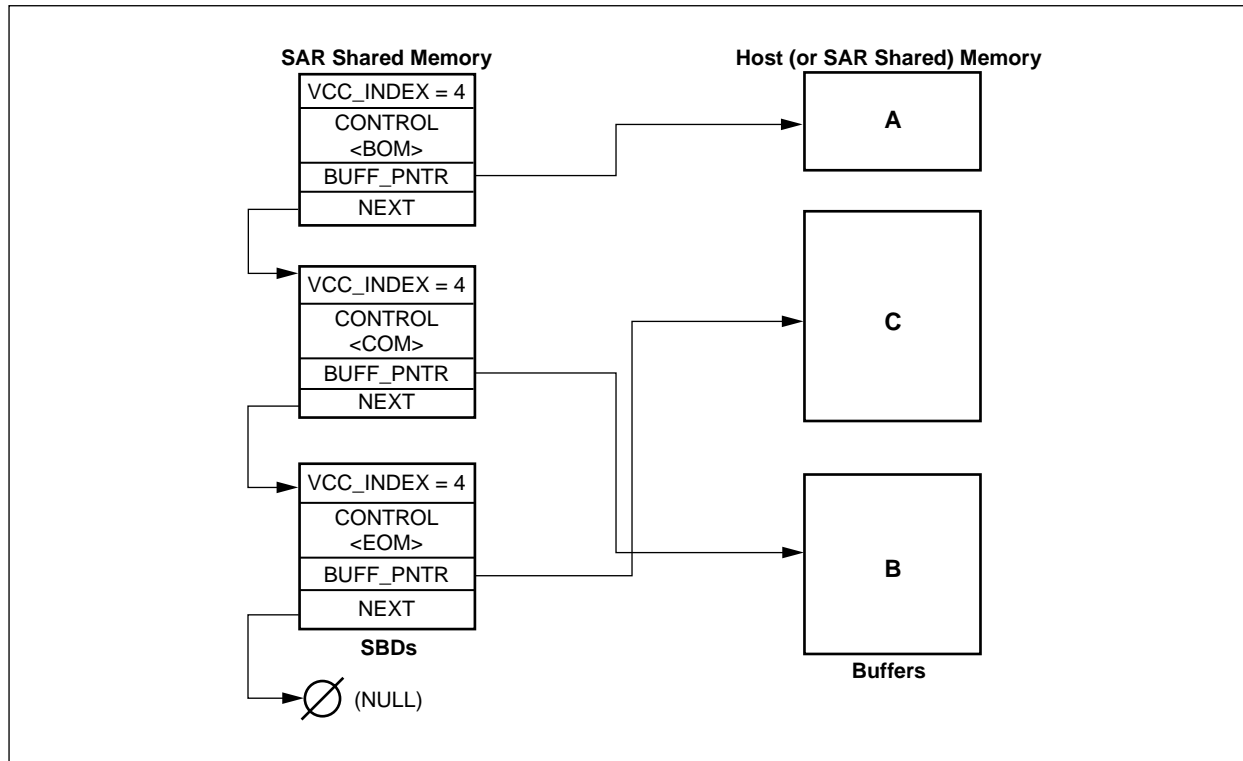
*Table 4-1. Segmentation PDU Delineation*

| 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 illustrates an example of SBD chaining. The host has formed a three-buffer message by linking three SBDs. In this example, the SAR transmits 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



8236\_013

#### 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 32 transmit queues for this purpose. Each transmit queue is a circular queue of one-word entries. The host identifies the next available transmit queue entry according to the write-only host interface specification described in Section 3.3.1. 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 CN8236 detects this write by snooping SAR-shared memory accesses. When a write occurs to any of the transmit queues, the CN8236 marks that queue as pending. Once every cell slot, the CN8236 services one queue entry from one Transmit Queue. The system designer selects one of two service order priority schemes. With the SEG\_CTRL(TX\_RND) bit set to 1, the CN8236 services queues in round-robin order (that is, one entry per queue in transmit queue sequence for all active queues). With the SEG\_CTRL(TX\_RND) bit set to 0, the CN8236 services the queues in fixed priority order (that is, entries from higher priority active queues are serviced before lower priority queue entries, with transmit queue 31 having highest priority).

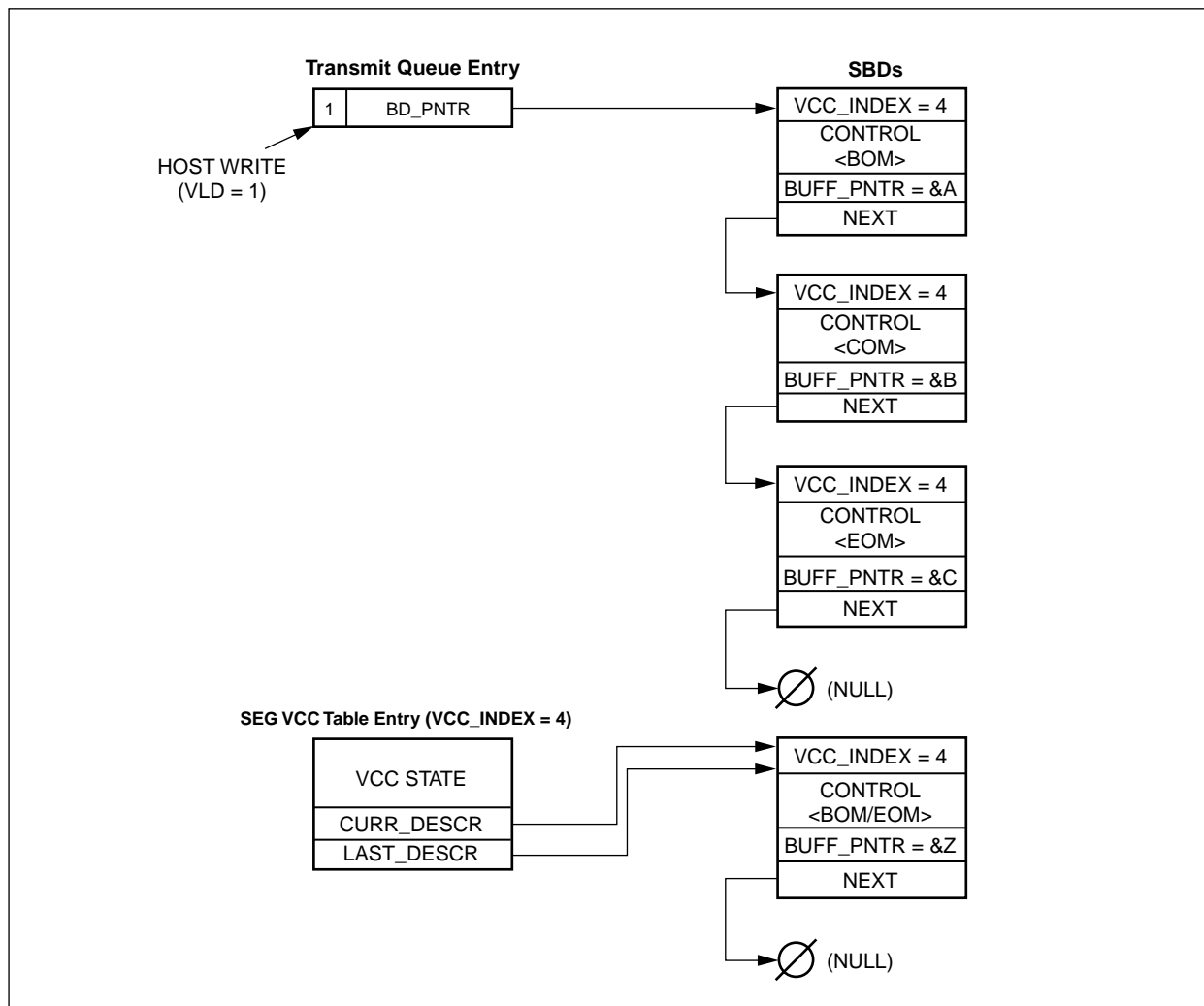
In either case, the SAR services one transmit queue entry by linking the new buffer descriptor chain to the VCC table entry identified by the VCC\_INDEX in the first SBD. The VCC table entry includes pointers to buffer descriptors for segmentation. The SAR links the new SBDs to the current chain of SBDs 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 CN8236 resets the transmit queue VLD bit to 0.

**SAR Transmit Queue Processing**

Figures 4-3 and 4-4 illustrate the process of linking SBDs to a VCC table entry.

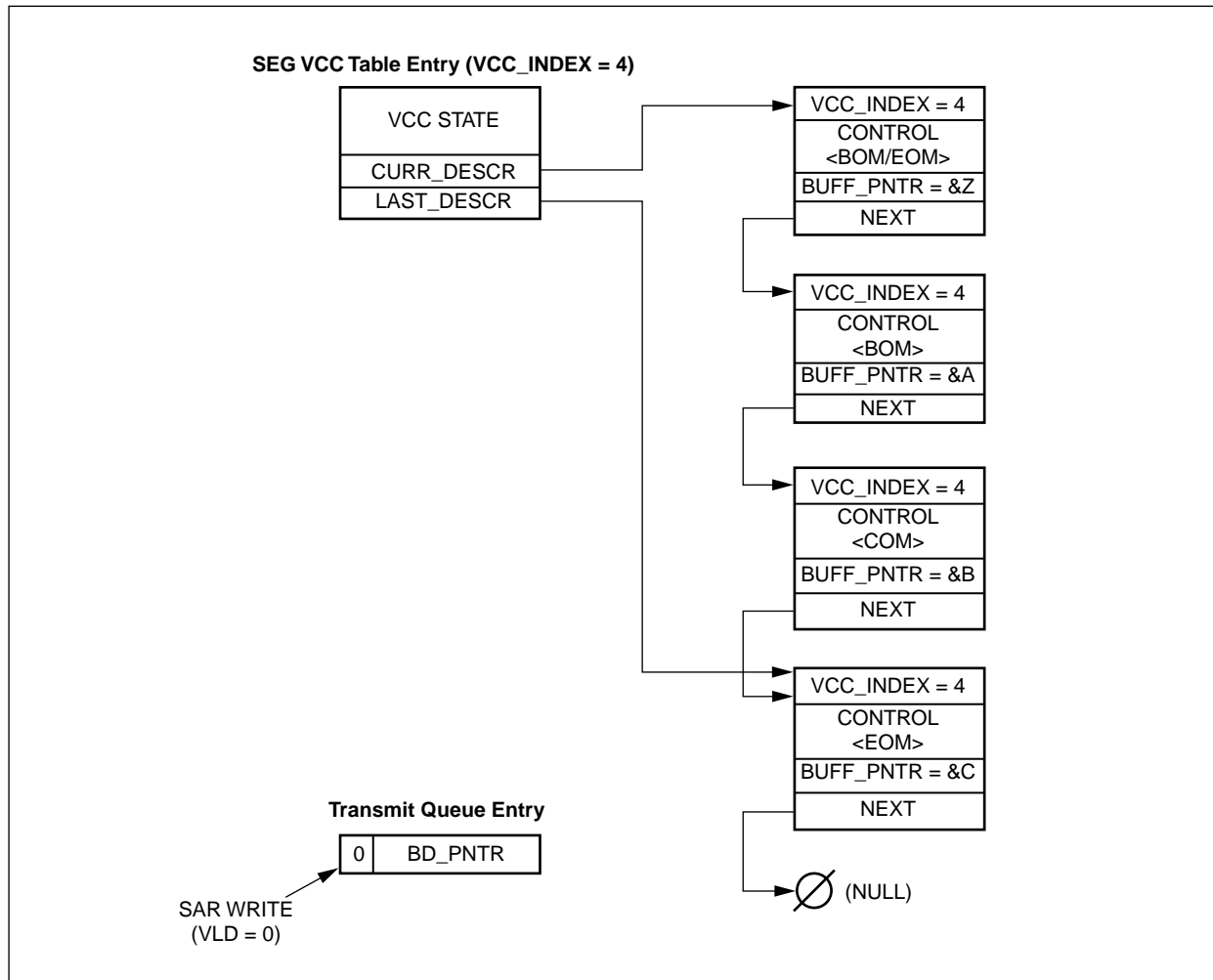
**NOTE:** As the new buffers are submitted, the VCC is processing a single buffer PDU (BOM/EOM). The CN8236 accepts new PDUs while it is processing outstanding buffers.

Figure 4-3. Before SAR Transmit Queue Entry Processing



8236\_014

Figure 4-4. After SAR Transmit Queue Entry Processing



8236\_015

#### 4.2.2.5 Partial PDUs

The host can submit partial PDUs to the CN8236. 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 CN8236 tracks the remaining segmentation data. If a partial cell remains, the CN8236 holds the buffer until it can complete a cell. Once the host submits an additional buffer, the CN8236 resumes segmentation.

#### 4.2.2.6 Virtual Paths

For network management simplicity, the host can create Virtual Path VCCs (that is, it can segment many VCIs on one VP). The host supplies the VCI for the ATM header within each Segmentation Buffer Descriptor (SBD). When using this method, the CN8236 must be provided with contiguous linked SBDs that are of the same VCC\_INDEX (that is, the same VCI) for the length of the PDU. This allows the CN8236 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 CN8236 adds the CPCS-PDU protocol fields to the CPCS-SDU. The SAR supports three AAL levels: AAL5, AAL3/4, and a transparent adaptation layer (AAL0).

Specific features also allow the generation of OAM cells. [Chapter 7.0](#), covers OAM generation in detail.

#### 4.2.3.1 AAL5

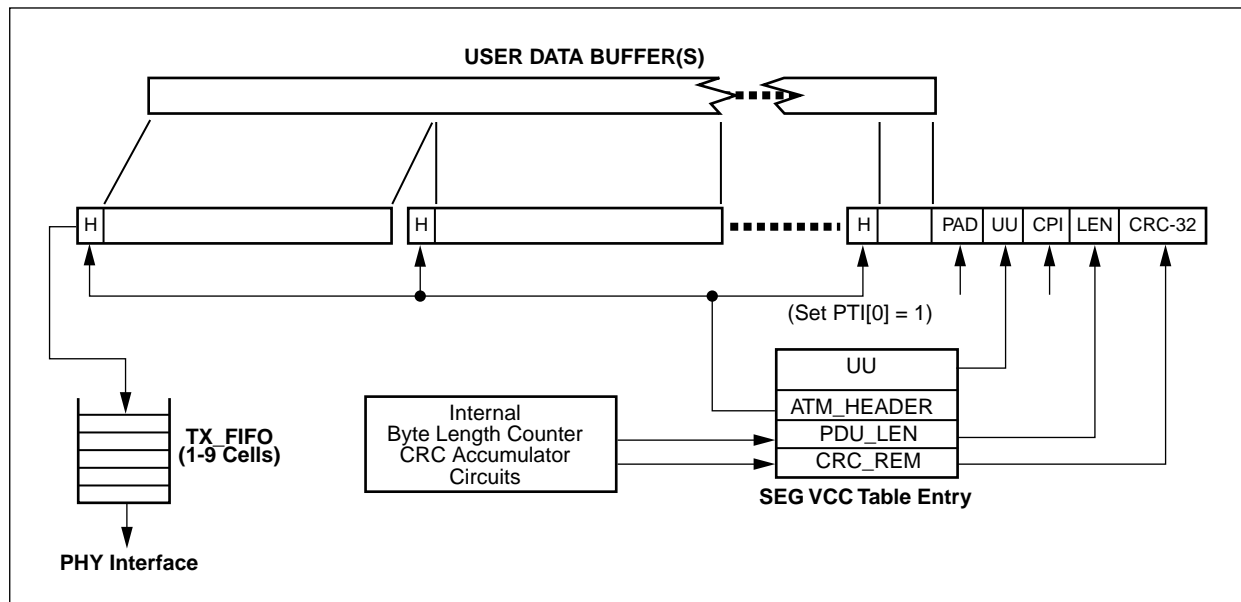
For AAL5, the SAR generates the CPCS-PDU trailer and pads the CPCS-SDU to align the PDU to a cell boundary. The CN8236 generates the PAD, Length (LEN), Common Part Indicator (CPI), and Cyclic Redundancy Check (CRC) fields according to I.363. The host supplies the CPCS User-to-User Indication (UU) field in the first buffer descriptor in a message, and the CN8236 transmits it following I.363.

The SAR generates the ATM header according to host-initialized settings in the VCC table entry. The CN8236 terminates AAL5 PDUs by setting bit 0 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](#) illustrates the CN8236'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 CN8236 transmits these fields within the EOM cell. The PAD and CPI fields are generated internally.

Figure 4-5. AAL5 CPCS-PDU Generation



8236\_016

**4.2.3.2 AAL3/4** When a segmentation buffer descriptor's AAL\_MODE field is set to AAL3/4 (value = b10), the CN8236 generates the CPCS-PDU's CPI, Btag, Etag, BASIZE, Alignment (AL), and Length fields in the header and trailer of the CPCS-PDU and pads the PDU to align to a cell boundary.

On the first cell of a buffer with BOM set, the segmentation coprocessor generates the CPCS-PDU header fields as the first four bytes of the first SAR-PDU's payload.

On the last cell of a buffer with EOM set, the segmentation coprocessor writes an all-0s PAD field after the end of the segmentation buffer data to complement the CPCS-PDU payload to an integral number of four-byte words. The segmentation coprocessor then adds the 4-byte CPCS-PDU trailer and fills octets with 0s for the remainder of the SAR-PDU payload.

Each CPCS-PDU field is generated as described in [Table 4-2](#).

**Table 4-2. AAL3/4 CPCS-PDU Field Generation**

| Field  | # Bits | Function   |
|--------|--------|--|
| CPI    | 8      | Common Part Identifier; set to 0.                              |
| BTAG   | 8      | Beginning Tag; read from SEG VCC table entry.                  |
| BASIZE | 16     | Buffer Allocation Size; read from SEG Buffer Descriptor entry. |
| AL     | 8      | Alignment Filler, aligns the trailer to fit a 32-bit word.     |
| ETAG   | 8      | Ending Tag; read from SEG VCC table entry.                     |
| LENGTH | 16     | CPCS-PDU Length; the calculated size of the PDU's payload.     |

The CN8236 also generates the SAR-PDU's Segment Type (ST), Sequence Number (SN), Message Identification (MID), Length Indication (LI), and CRC fields in each segmented cell for that CPCS-PDU.

Each AAL3/4 cell carries 44 octets of payload and four octets (five fields) of header and trailer information. On the last generated cell for a CPCS-PDU with the EOM set in the buffer descriptor, the segmentation coprocessor pads the SAR-PDU payload with 0 to 44 bytes.

Each SAR-PDU field is generated as described in [Table 4-3](#).

**Table 4-3. AAL3/4 SAR-PDU Field Generation**

| Field | # Bits | Function   |
|-------|--------|--|
| ST    | 2      | BOM value for the first cell generated from the buffer with the BOM option in the buffer descriptor set.<br><br>EOM value for the last cell generated from the buffer with the EOM option in the buffer descriptor set.<br><br>SSM value for the cell generated from the buffer with both BOM and EOM options in the buffer descriptor set, and the buffer descriptor LENGTH field $\leq 44$ .<br><br>COM value for all other generated cells. (See the next table for Binary values.) |
| SN    | 4      | Read from the SN field in the SEG VCC structure. The SN field is incremented modulo 16 after each use.   |
| MID   | 10     | Read from the MID field in the SEG VCC structure.  |
| LI    | 6      | Generated by the segmentation coprocessor in an internal byte length counter.  |
| CRC   | 10     | Generated as all zeros. The CRC field can be overwritten with the CRC-10 generator before transmission by setting the CRC_10 option in the SEG buffer descriptor.  |

[Table 4-4](#) shows the settings for the ST (Segment Type) field.

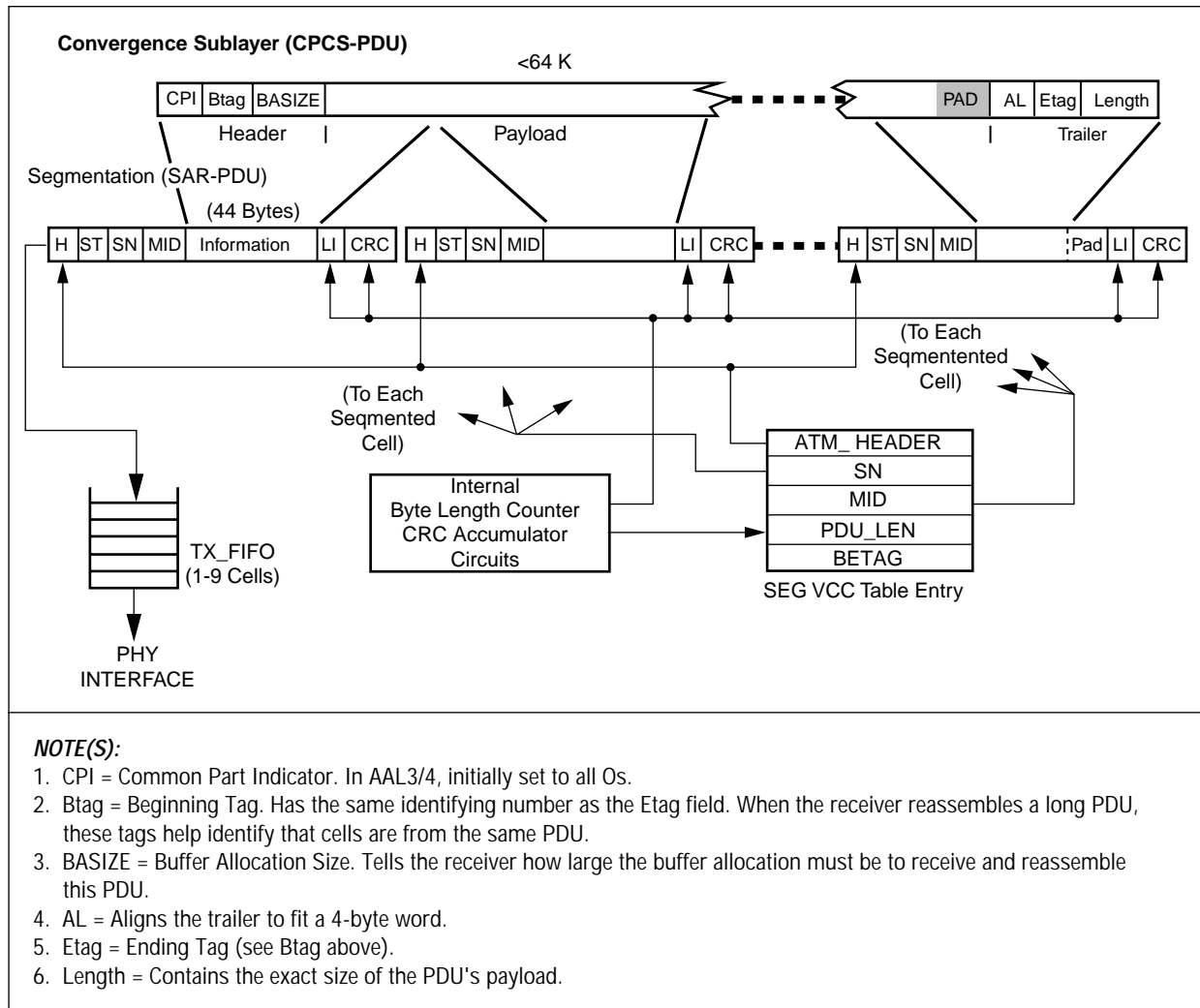
**Table 4-4. Coding of Segment Type (ST) Field**

| Segment Type | Encoding | Usage                   |
|--------------|----------|-------------------------|
| BOM          | 10       | Beginning of Message    |
| COM          | 00       | Continuation of Message |
| EOM          | 01       | End of Message          |
| SSM          | 11       | Single Segment Message  |



Figure 4-6 shows the CN8236's AAL3/4 PDU generation scheme.

Figure 4-6. AAL3/4 CPCS-PDU Generation



8236\_094

**4.2.3.3 AAL0** The CN8236 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 PHY Layer Interface

Once the segmentation coprocessor has formed an ATM cell, the CN8236 transfers the cell to the transmit FIFO buffer. The user chooses the length of this FIFO buffer, with possible sizes from one to nine cells. The FIFO buffer depth is programmable, since there is a trade-off between absorbing PCI latency with a longer FIFO buffer, and introducing greater jitter. [Section 6.2.3.3](#), in the [Chapter 6.0](#), discusses this trade-off in greater depth. Once sent to the transmit FIFO buffer, the cell passes through the FIFO buffer to the PHY layer interface circuits.

### 4.2.4.1 Head-of-Line Flushing (HoLF)

If the SAR is set up as a UTOPIA Master in multi-phy operations, cells are transmitted to a PHY device after the PHY indicated that it can receive another cell by asserting its UTOPIA CLAV signal. The UTOPIA Master waits until it receives the CLAV signal from the PHY device, and therefore blocks the transmission of all other cells in the transmit FIFO. If this PHY device stops working, all other PHY devices are blocked. This is called head-of-line blocking.

In order to avoid head-of-line blocking in the transmit FIFO, a mechanism to flush the blocking cell out of the transmit FIFO is incorporated (Head-of-Line Flushing). This mechanism is enabled by setting the TX\_FIFO\_FLUSH\_EN bit in the CONFIG1 register.

When the UTOPIA Master puts out the address of a PHY, a counter is reset to 0 and increased based on the UTOPIA tx\_clk. Once the counter reaches the values TX\_CNTR set by the user in the CSR register, the cell is discarded, and the bit corresponding to the blocking PHY is set in the TX\_PORT\_STATUS register. The counter is reset automatically. If any of the eight bits in the TX\_STATUS register is set, TX\_DISCARD in HOST\_ISTAT1 and LP\_ISTAT1 are set. If the corresponding mask bit EN\_TX\_DISCARD in HOST\_IMASK1 or LP\_IMASK1 is set, an interrupt is generated. The TX\_DISCARD is cleared once the TX\_STATUS register is read by the host.

**NOTE:** The cell discard does not disable the port.

The user might decide to disable a specific port on the UTOPIA interface. Then all cells belonging to this port are discarded or flushed. Ports are disabled by setting the corresponding bit in the TX\_PORT\_CTRL register. If a port x is disabled and a cell pertaining to port x is discarded, bit x in TX\_STATUS is not set, and therefore no interrupt is generated.

For fault-tolerant multi-phy operations, a maximum TX\_CNTR value of 65,535 is recommended. For specific DSL applications with variable rate PHY devices, a value between 50 and 100 is suggested.

### 4.2.5 Status Reporting

The CN8236 informs the host of segmentation completion using segmentation status queues. The host assigns each VCC to one of 32 status queues, enabling a multiple peer architecture as described in [Section 3.2](#). The CN8236 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 CN8236 subsystem's segmentation block. The segmentation status queues operate according to the write-only host interface, defined in [Section 3.3.2](#).

The CN8236 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` can contain the address of the buffer or of a host data structure describing the buffer. To simplify host management, the CN8236 also returns the `VCC_INDEX` of the VCC on which the buffer was transmitted.

### 4.2.6 Virtual FIFO Buffers

In addition to gathering PDU data from buffers, the CN8236 provides an optional method to segment from a fixed PCI address, or Virtual FIFO buffer. The CN8236 supports AAL0, CBR Virtual FIFO buffer segmentation.

The host configures the channel for Virtual FIFO buffer operation by setting the `CURR_PNTR` and `RUN` fields to 0 in the SEG VCC table entry. The host writes the FIFO buffer 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 buffer. Once the FIFO buffer is almost full, the host sets the `RUN` bit to a logic high. The SAR then starts reading from the FIFO buffer. When the FIFO buffer gets below almost empty, the host sets the `SCH_OPT` bit to a logic high. The SAR then skips a cell transmit opportunity in order to allow the FIFO buffer to refill. After the SAR skips a cell, it resets the `SCH_OPT` bit to a logic low.

In this mode, the segmentation coprocessor reads 48 bytes of payload from the host FIFO buffer, and prepends (that is, attaches to the beginning) the `ATM_HEADER` value in the VCC table entry. The host does not use the transmit queue for Virtual FIFO buffers. The CN8236 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 7 words are generic, independent of traffic class. The last 3 or 13 words provide additional parameters specific to service classes.

There are two basic formats for the generic part of the VCC table entry: AAL3/4-AAL5-AAL0 and Virtual FIFO buffer. Each format completely describes the state of the segmentation process for individual VCCs. [Table 4-5](#) and [Table 4-6](#) describe the format of AAL3/4, AAL5, and AAL0 VCC table entries.

**KEY:**



= Written by host at VCC setup

= May be dynamically modified during active segmentation

**Table 4-5. Segmentation VCC Table Entry—AAL3/4-AAL5-AAL0 Format**

| Word   | 31         | 30                | 29  | 28                | 27      | 26      | 25       | 24     | 23        | 22     | 21     | 20        | 19        | 18 | 17 | 16         | 15   | 14 | 13 | 12 | 11 | 10  | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--|------------|-------------------|-----|-------------------|---------|---------|----------|--------|-----------|--------|--------|-----------|-----------|----|----|------------|------|----|----|----|----|-----|---|---|---|---|---|---|---|---|---|---|
| 0  | PM_INDEX   |                   |     |                   |         |         |          | Rsvd   | ACK_PM    | FWD_PM | Rsvd   | LAST_PNTR |           |    |    |            |      |    |    |    |    |     |   |   |   |   |   |   |   |   |   |   |
| 1  | UU         |                   |     |                   |         |         |          |        | PORT_ID   |        |        | BOM_PNTR  |           |    |    |            |      |    |    |    |    |     |   |   |   |   |   |   |   |   |   |   |
| 2  | ATM_HEADER |                   |     |                   |         |         |          |        |           |        |        |           |           |    |    |            |      |    |    |    |    |     |   |   |   |   |   |   |   |   |   |   |
| 3  | PDU_LEN    |                   |     |                   |         |         |          |        |           |        |        |           |           |    |    | BUFFER_LEN |      |    |    |    |    |     |   |   |   |   |   |   |   |   |   |   |
| 4 <sup>(1)</sup>                             | CRC_REM    |                   |     |                   |         |         |          |        |           |        |        |           |           |    |    |            |      |    |    |    |    |     |   |   |   |   |   |   |   |   |   |   |
| 4 <sup>(2)</sup>                             | Reserved   |                   |     |                   |         |         |          |        | BETAG     |        |        |           |           |    |    |            | Rsvd | SN |    |    |    | MID |   |   |   |   |   |   |   |   |   |   |
| 5  | STM_MODE   | STAT/<br>HPORT_ID |     |                   |         |         |          | PM_LEN | LAST_CLIP | RUN    | IX_EOM | Rsvd      | CURR_PNTR |    |    |            |      |    |    |    |    |     |   |   |   |   |   |   |   |   |   |   |
| 6  | SCH        | BCK_PM            | VPC | SCH_GFR/CBR_W_TUN | FLOW_ST | GFR_PRI | SCH_MODE |        |           |        | PRI    | SCH_OPT   | NEXT_VCC  |    |    |            |      |    |    |    |    |     |   |   |   |   |   |   |   |   |   |   |
| 7-9/19                                       | SCH_STATE  |                   |     |                   |         |         |          |        |           |        |        |           |           |    |    |            |      |    |    |    |    |     |   |   |   |   |   |   |   |   |   |   |
| <b>NOTE(S):</b>                              |            |                   |     |                   |         |         |          |        |           |        |        |           |           |    |    |            |      |    |    |    |    |     |   |   |   |   |   |   |   |   |   |   |
| (1) This word is used in AAL5 and AAL0 VCCs. |            |                   |     |                   |         |         |          |        |           |        |        |           |           |    |    |            |      |    |    |    |    |     |   |   |   |   |   |   |   |   |   |   |
| (2) This word is used in AAL3/4 VCCs.        |            |                   |     |                   |         |         |          |        |           |        |        |           |           |    |    |            |      |    |    |    |    |     |   |   |   |   |   |   |   |   |   |   |

Table 4-6. Segmentation VCC Table Entry—AAL3/4, 5, and 0 Field Descriptions (1 of 2)

| Field Name    | Description   |
|---------------|---|
| PM_INDEX      | Performance Monitoring index. 128 VCCs can be selected for automatic OAM PM generation. Each monitored VCC has a unique performance monitoring index.<br>If this field is changed while the VCC is active, only the byte containing the field should be written. (See <a href="#">Chapter 6.0</a> .)  |
| ACK_PM        | Toggled after each backward reporting PM cell is sent. Used to prevent PM information in PM table from being overwritten before cell is sent.   |
| FWD_PM        | Indicates that next cell should be forward monitoring PM cell.  |
| LAST_PNTR     | Points to last buffer descriptor currently linked to the VCC. The two least significant bits of the pointer are assumed to be 0 (word-aligned). The address of the buffer descriptor is (LAST_PNTR << 2) or (LAST_PNTR x 4).  |
| UU            | AAL5 User-to-User indication. This field is copied from the buffer descriptor UU field. The CN8236 inserts the UU field in the CPCS-PDU trailer contained in the EOM cell.  |
| BOM_PNTR      | In Message mode (STM_MODE=0), points to the first buffer descriptor of a message composed of more than one buffer descriptor. The CN8236 returns the corresponding USER_PNTR from this buffer descriptor in the status queue.<br>In Streaming mode (STM_MODE=1), it is not used.<br>The two least significant bits of the pointer are assumed to be zero (word-aligned). The address of the buffer descriptor is (BOM_PNTR<<2) or (BOM_PNTR * 4). |
| ATM_HEADER    | Used for each ATM cell for the VCC. The transmitted header may be modified by option bits in the current buffer descriptor.   |
| PDU_LEN       | CPCS-PDU Trailer Length field. This field is generated by the SAR and inserted in the Length field of the EOM cell.   |
| BUFFER_LEN    | Buffer Length. The number of bytes of data read from the current segmentation buffer.   |
| PORT_ID       | Identifies the Physical Device (Port) ID for the VCC.   |
| CRC_REM       | Accumulated value of AAL5 32-bit CRC, calculated on-the-fly by the SAR and appended to the PDU at EOM.  |
| BETAG         | AAL3/4 Btag and Etag fields.  |
| SN            | AAL3/4 Sequence Number field.   |
| MID           | AAL3/4 Message Id field.  |
| STM_MODE      | Streaming Status Mode.<br>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.<br>1 = Streaming Mode—a status entry is written (with a corresponding maskable interrupt) for each buffer as it completes segmentation.   |
| STAT/HPORT_ID | Identifies the Segmentation Status Queue used for all status entries, or if the AALx_EN bit is set, identifies the AALx port.   |
| PM_EN         | Enables performance monitoring for the VCC. If this bit is changed while the VCC is active, only the byte containing the bit should be written. (See <a href="#">Chapter 7.0</a> .)   |
| LAST_CLP      | Last transmitted CLP bit for VCC. This bit is updated by the SRC after each transmitted cell. This bit is in the same word as the PM_EN bit, and is copied by the SAR from the current buffer descriptor. The bit is used to select the correct VBR bucket for certain VBR VCCs.  |

Table 4-6. Segmentation VCC Table Entry—AAL3/4, 5, and 0 Field Descriptions (2 of 2)

| Field Name            | Description   |
|-----------------------|---|
| RUN                   | Flag that indicates segmentation is proceeding. This flag is set to one by the SAR when the host supplies data for the VCC; it is set to zero by the SAR when the data available for the VCC is not sufficient to send an entire ATM cell.  |
| NX_EOM                | Indicates that the next cell is the end of a CPCS-PDU.  |
| 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 0 (word-aligned). The address of the descriptor is (CURR_PNTR << 2) or (CURR_PNTR x 4).   |
| SCH                   | Indicates VCC is currently scheduled for segmentation.  |
| BCK_PM                | Toggled after each backward reporting PM cell is scheduled. Used to prevent PM information in PM table from being overwritten before cell sent.   |
| VPC                   | On a VCC with SCH_MODE = ABR this indicates a VP (instead of a VCC) connection, and RM cells are generated on VCI = 6.  |
| SCH_GFR/<br>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. This host does not have to set this bit—it is set by the SAR. 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. |
| FLOW_ST               | Flow control state. This bit is active only in ER mode. Indicates that the priority of the connection is increased to insure MCR.   |
| GFR_PRI               | Specifies the UBR priority level for a GFR service connection. GFR_PRI must be < PRI.   |
| SCH_MODE              | Traffic class of VCC. (See <a href="#">Chapter 6.0</a> , for details.)<br>000 UBR = Unspecified Bit Rate<br>001 CBR = Constant Bit Rate<br>010 = Reserved<br>011 GFR = Guaranteed Frame Rate<br>100 VBR1 = Single Leaky Bucket VBR<br>101 VBR2 = Dual Leaky Bucket VBR with both buckets always active<br>110 VBRC = Dual Leaky Bucket VBR with bucket 1 applied only to CLP = 0 cells<br>111 ABR = Available Bit Rate (as specified by TM 4.1)   |
| PRI                   | Segmentation priority. The lowest priority is 0. The highest priority is 15. 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.<br>VBR1, VBR2, VBRC, ER: Initializes bucket state to send maximum burst. The SAR writes this bit to 0 after bucket state initialized.<br>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.<br>If this bit is changed while the VCC is active, only the byte containing the bit should be written.                 |
| NEXT_VCC              | Used by SAR to link VCCs in schedule chains.  |
| 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 <a href="#">Chapter 6.0</a> .)   |

Table 4-7 shows the format for Virtual FIFO buffer VCC table entries, including setup values for restricted fields. Table 4-8 describes the field that differs from the AAL3/4-AAL5-AAL0 format.

Table 4-7. Segmentation VCC Table Entry—Virtual FIFO Buffers

| Word   | 31         | 30       | 29 | 28 | 27       | 26    | 25       | 24    | 23        | 22                    | 21 | 20 | 19 | 18  | 17      | 16       | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------------|----------|----|----|----------|-------|----------|-------|-----------|-----------------------|----|----|----|-----|---------|----------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0      | PM_INDEX   |          |    |    | Reserved |       |          |       | LAST_PNTR |                       |    |    |    |     |         |          |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 1      | Reserved   |          |    |    | Reserved |       |          |       | BOM_PNTR  |                       |    |    |    |     |         |          |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 2      | ATM_HEADER |          |    |    |          |       |          |       |           |                       |    |    |    |     |         |          |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 3      | FIFO_PNTR  |          |    |    |          |       |          |       |           |                       |    |    |    |     |         |          |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 4      | CRC_REM    |          |    |    |          |       |          |       |           |                       |    |    |    |     |         |          |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 5      | STM_MODE   | STAT     |    |    |          | PM_EN | Reserved | RUN=1 | Reserved  | CURR_PNTR= 0x00000    |    |    |    |     |         |          |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 6      |            | Reserved |    |    |          |       |          |       |           | SCH_MODE (=001 - CBR) |    |    |    | PRI | SCH_OPT | Reserved |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 7-9/19 | SCH_STATE  |          |    |    |          |       |          |       |           |                       |    |    |    |     |         |          |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

Table 4-8. Segmentation VCC Table Entry—Virtual FIFO Buffer Format Field Descriptions

| Field Name | Description   |
|------------|---|
| FIFO_PNTR  | Pointer to the PCI space data FIFO buffer 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 CN8236 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 64 KB.

### 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.

Tables 4-9 through 4-13 describe the entry formats and field definitions for the SBDs.

**Table 4-9. 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 <sup>(1)</sup>  | UU          |       |        |         |            |         |          |         | Rsvd  | NEXT_PNTR |         |      |     |     |               |    |    |    |    |    |    |    |   |   |   | Rsvd |   |   |   |   |   |   |
| 0 <sup>(2)</sup>  | BASIZE_H    |       |        |         |            |         |          |         | Rsvd  | NEXT_PNTR |         |      |     |     |               |    |    |    |    |    |    |    |   |   |   | Rsvd |   |   |   |   |   |   |
| 1   | USER_PNTR   |       |        |         |            |         |          |         |       |           |         |      |     |     |               |    |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| 2   | BUFFER_PNTR |       |        |         |            |         |          |         |       |           |         |      |     |     |               |    |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| 3   | Reserved    | 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 |    |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| <b>NOTE(S):</b>   |             |       |        |         |            |         |          |         |       |           |         |      |     |     |               |    |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| <sup>(1)</sup> This version of word 0 is used for AAL5 and AAL0 VCCs. |             |       |        |         |            |         |          |         |       |           |         |      |     |     |               |    |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| <sup>(2)</sup> This version of word 0 is used for AAL3/4 VCCs.        |             |       |        |         |            |         |          |         |       |           |         |      |     |     |               |    |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |

**Table 4-10. MISC\_DATA Field Bit Definitions with HEADER\_MOD Bit Set**

| Bit  | 31       | 30 | 29 | 28 | 27       | 26 | 25 | 24     | 23     | 22     | 21       | 20 | 19 | 18       | 17 | 16 |  |
|--|----------|----|----|----|----------|----|----|--------|--------|--------|----------|----|----|----------|----|----|--|
| Def.   | GFC_DATA |    |    |    | Reserved |    |    | WR_GFC | WR_PTI | WR_VCI | PTI_DATA |    |    | VCI_DATA |    |    |  |
| <b>NOTE(S):</b> This definition of bits 31:16, MISC_DATA field, applies when the HEADER_MOD bit is set, RPL_VCI=0, and AAL_MODE is not = AAL3/4; used when generating OAM cells. |          |    |    |    |          |    |    |        |        |        |          |    |    |          |    |    |  |



**Table 4-11. MISC\_DATA Field Bit Definitions with RPL\_VCI Bit Set**

| Bit   | 31      | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Def.  | NEW_VCI |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>NOTE(S):</b> 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 4-12. MISC\_DATA Field Bit Definitions with AAL\_MODE Set to AAL3/4**

| Bit  | 31       | 30 | 29 | 28 | 27       | 26 | 25     | 24       | 23       | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|--|----------|----|----|----|----------|----|--------|----------|----------|----|----|----|----|----|----|----|
| Def.   | GFC_DATA |    |    |    | Reserved |    | WR_GFC | Reserved | BASIZE_L |    |    |    |    |    |    |    |
| <b>NOTE(S):</b> This definition of bits 31:16, MISC_DATA field, applies when the AAL_MODE field = AAL3/4 and RPL_VCI = 0. In order to activate GFC override, the HEADER_MOD bit must be set to a logic high. |          |    |    |    |          |    |        |          |          |    |    |    |    |    |    |    |

Table 4-13. Segmentation Buffer Descriptor Field Descriptions (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.   |
| BASIZE_H    | The high order bits used for the BASIZE field in the AAL3/4 header when the GEN_PDU option is selected.  |
| NEXT_PNTR   | Pointer to next buffer descriptor for the VCC. The two least significant bits of the pointer are assumed to be 0 (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 may equal BUFFER_PNTR. The SAR circulates this field back to the host in the status entry without using it internally.  |
| BUFFER_PNTR | Pointer to segmentation buffer in host- or 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.<br>1 = BUFFER_PNTR is a word-aligned address in SAR-shared memory instead of host memory.   |
| SET_CI      | 0 = The CN8236 generates bit 1 of PTI[2:0] from the VCC table entry ATM_HEADER field.<br>1 = Sets bit 1 of the ATM header PTI[2:0] field for all cells in buffer to 1.   |
| SET_CLP     | 0 = The CN8236 generates the CLP bit from the VCC table entry ATM_HEADER field.<br>1 = Sets the ATM header CLP bit for all cells in buffer to 1.<br>Also used to control VBR CLP Dual Leaky Bucket mode.   |
| HEADER_MOD  | 0 = The CN8236 ignores the WR_GFC, WR_PTI, and WR_VCI bits in this buffer descriptor.<br>1 = Activates the WR_GFC, WR_PTI, and WR_VCI bits for this buffer descriptor.   |
| RPL_VCI     | 0 = The CN8236 generates the VCI field from the VCC table entry ATM_HEADER field.<br>1 = The CN8236 generates the VCI field from the NEW_VCI for all cells in the buffer. NEW_VCI is also copied in to the VCI portion of the ATM_HEADER field in the VCC entry.   |
| OAM_STAT    | Buffer reports status to the global OAM Status Queue SEG_CTRL(OAM_STAT_ID) instead of the STAT specified in the VCC table entry.<br>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).<br>See <a href="#">Chapter 7.0</a> —OAM for more details. |
| GEN_PDU     | 1 = Generate AAL3/4 header and trailer, or AAL5 trailer. In AAL3/4 mode, the SAR-PDU is always generated internally.   |
| CRC10       | Overwrite last ten bits of each cell with CRC10 calculation. Used for OAM and AAL3/4 cells.  |
| AAL_MODE    | Controls AAL segmentation mode.<br>00 = AAL5<br>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.<br>10 = AAL3/4<br>11 = Reserved   |

Table 4-13. Segmentation Buffer Descriptor Field Descriptions (2 of 3)

| Field Name | Description   |
|------------|---|
| AAL_OPT    | Options for AAL_MODE:<br>For AAL_MODE = AAL0<br>00 = Normal operation<br>01 = SINGLE<br>10 = Reserved<br>11 = Reserved<br>For AAL_MODE = AAL3/4 or AAL5<br>00 = Normal operation<br>01 = ABORT<br>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 the start of buffer descriptor chain for VCC. This means that there are no concerns with mid-cell insertion and that the cell has low latency. This is intended for OAM cells.<br>NOTE: This option MUST be set in the buffer descriptor for any Tx Queue entry with LINK_HEAD set. To do otherwise may result in corrupted SEG data.<br>ABORT: Send AAL3/4 or 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 = CN8236 reads the 48-octet payload of ATM cells from memory and generates the ATM header internally.<br>1 = The CN8236 reads the entire 52-octet ATM cell from segmentation buffer. The ATM_HEADER stored in the VCC table entry is not used in this mode, and AAL_MODE is ignored.  |
| BOM        | 1 = Buffer contains the beginning of a message. (See Table 4-1.)  |
| EOM        | 1 = Buffer contains the end of a message. (See Table 4-1.)  |
| LENGTH     | Number of bytes of data contained in the segmentation buffer. Local memory, non-EOM buffer lengths must be multiples of 4 bytes (mod 4). Maximum allowable size is 64 KB.   |
| GFC_DATA   | Data for WR_GFC option.   |
| WR_GFC     | 0 = The CN8236 generates the GFC field from the VCC table entry ATM_HEADER field.<br>1 = The CN8236 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_PT1     | 0 = The CN8236 generates the PTI field from the VCC table entry ATM_HEADER field.<br>1 = The CN8236 overwrites the ATM header PTI field for all cells in the buffer with PTI_DATA. The host can use this feature to generate F5 and PM OAM cells. (See Chapter 7.0.) This bit is active only when HEADER_MOD is set. This bit disables PM TUC and BIP16 calculations.   |
| WR_VCI     | 0 = The CN8236 generates the VCI field from the VCC table entry ATM_HEADER field.<br>1 = The CN8236 overwrites the ATM header VCI field for all cells in the buffer with (0x0000 VCI_DATA). (MSBs of VCI are set to 0.) Used to generate F4 OAM cells. (See Chapter 7.0.) This bit is active only when HEADER_MOD is set. This bit disables PM TUC and BIP16 calculations.  |
| PTI_DATA   | Data for WR_PT1 option. Normally used to generate OAM cells.  |
| VCI_DATA   | Data for WR_VCI option. Normally used to generate OAM cells.  |
| NEW_VCI    | Data for the RPL_VCI. The CN8236 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.   |

Table 4-13. Segmentation Buffer Descriptor Field Descriptions (3 of 3)

| Field Name    | Description   |
|---------------|---|
| BASIZE_L      | The low order bits used for the BASIZE field in the AAL3/4 header when the GEN_PDU option is selected.    |
| SEG_VCC_INDEX | Identifies the VCC entry in the VCC table. The CN8236 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 CN8236 by writing a single word transmit queue entry. [Table 4-14](#) and [Table 4-15](#) describe the format of these entries.

**Table 4-14. 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 | Reserved |    |    |    |    |    |    | SEG_BD_PNTR |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | Rsvd |   |   |   |   |   |

**Table 4-15. Transmit Queue Entry Field Descriptions**

| Field Name  | Description   |
|-------------|---|
| VLD         | 0 = Entry invalid. Waiting for the host to submit new data for segmentation.<br>1 = Entry valid. The SAR processes the entry when its read pointer into the queue advances to this entry.<br>Written to 1 by the host when submitting a new entry. The SAR clears this bit to 0 when it has successfully linked the buffer descriptor chain to the VCC table.   |
| LINK_HEAD   | 0 = The CN8236 links the new descriptor chain at the end of the existing chain on the VCC.<br>1 = The CN8236 links the new descriptor chain at the head of the existing chain.<br>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.<br>This bit is intended for use with the buffer descriptor SINGLE option to send in-line management cells with reduced latency.<br><b>NOTE(S):</b> 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 may result in corrupted segmentation data. |
| FIND_CHAIN  | Indicates the SAR is searching for the end of the buffer descriptor chain.<br>The host always writes this bit to 0.   |
| 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 0 (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 4,096 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, tracks the current state of the queues. [Table 4-16](#) and [Table 4-17](#) below describe the fields of these queues.

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

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

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 4-16. 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    | Reserved     |    |    |    |    |    |    |    | UPDATE |    |    |    | Reserved |    |    |    | READ |       |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

**Table 4-17. 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 writes its read pointer into the queue to this address periodically. (See <a href="#">Chapter 2.0</a> , for details.) |
| LOCAL        | 0 = READ_UD located in PCI address space.<br>1 = READ_UD located in SAR-shared memory.<br><b>NOTE(S):</b> 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 Routing Tags

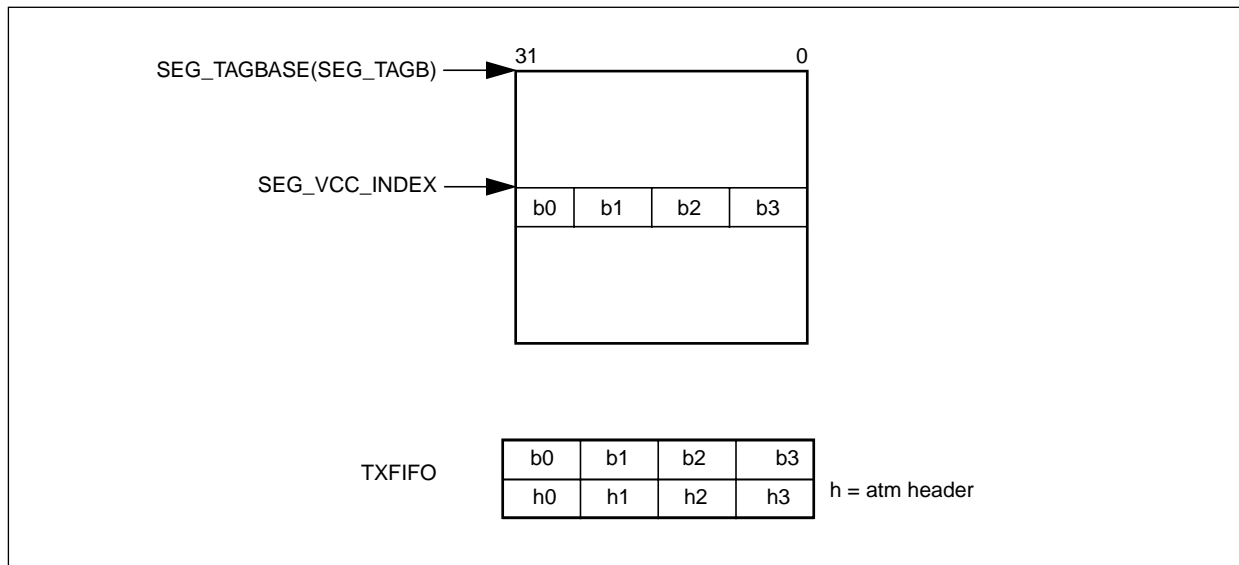
Based upon the setting of CONFIG1(TAG\_SIZE) field, routing tags need to be prepended to the cell written into the TXFIFO. Figures 4-7 through 4-9 show the routing tag tables and position in TXFIFO. The segmentation block writes either 4, 8, or 12 bytes into the TXFIFO, depending upon the size of tag required. The UTOPIA block strips off the appropriate number of bytes to form the desired size cell. Table 4-19 is a cross-reference between the routing tag table and the UTOPIA cell.

*NOTE:* The maximum Tx FIFO size becomes reduced when using routing tags.

Table 4-18. Maximum Tx FIFO Size with Routing Tags

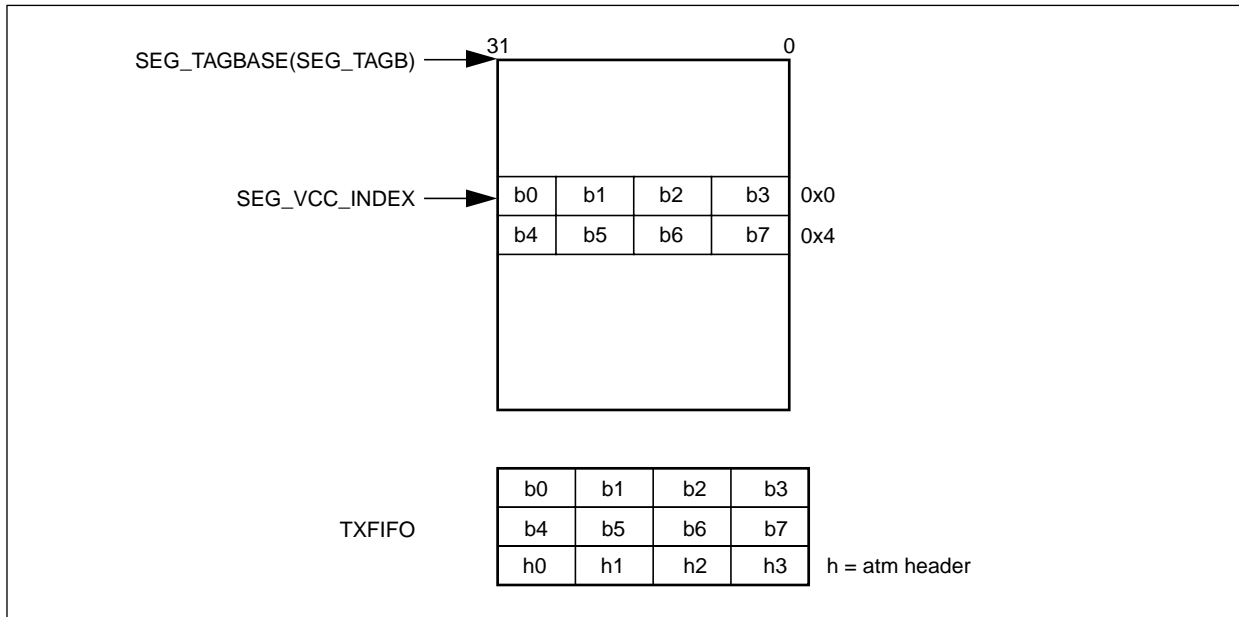
| Tag Size | TxFIFO<br>(maximum number of cells) |
|----------|-------------------------------------|
| 5        | 8                                   |
| 6        |                                     |
| 7        |                                     |
| 8        |                                     |
| 9        | 7                                   |
| 10       |                                     |
| 11       |                                     |

Figure 4-7. Route Tag Table for tag\_size = 1 through 4



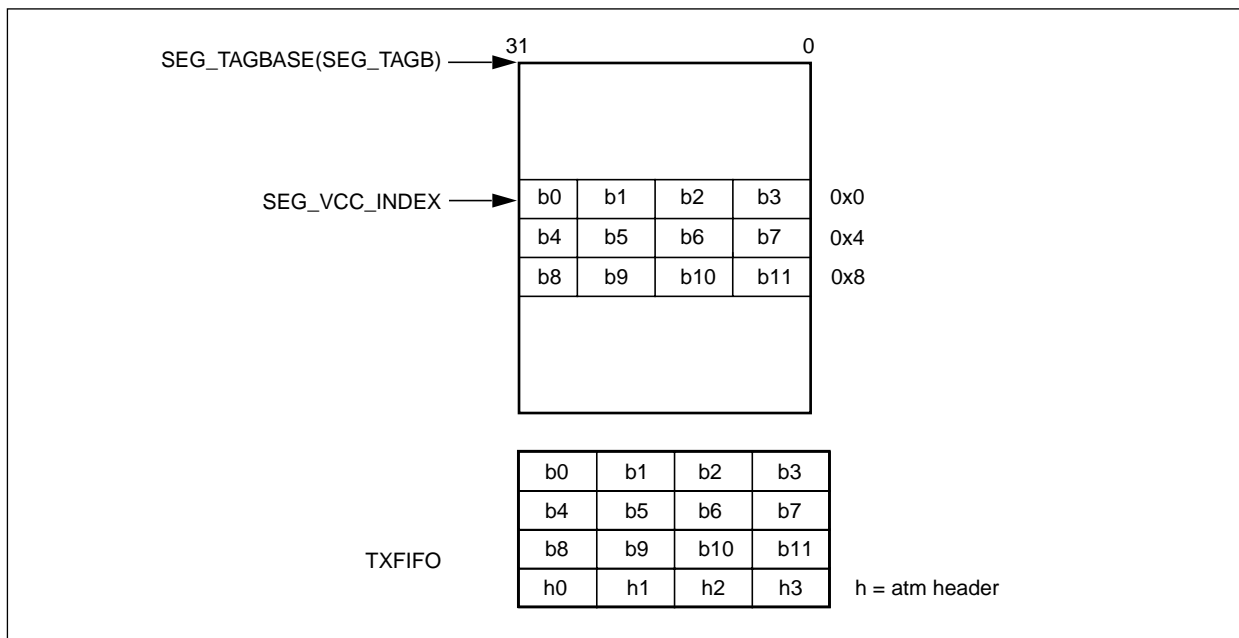
8236\_017

Figure 4-8. Route Tag Table for tag\_size = 5 through 8



8236\_018

Figure 4-9. Route Tag Table for tag\_size = 9 through 11



8236\_019



Table 4-19. Routing Tag Cross-Reference

| Tag Size | b0 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 | b11 |
|----------|----|----|----|----|----|----|----|----|----|----|-----|-----|
| 1        |    |    |    | t1 |    |    |    |    |    |    |     |     |
| 2        |    |    | t1 | t2 |    |    |    |    |    |    |     |     |
| 3        |    | t1 | t2 | t3 |    |    |    |    |    |    |     |     |
| 4        | t1 | t2 | t3 | t4 |    |    |    |    |    |    |     |     |
| 5        |    |    |    | t1 | t2 | t3 | t4 | t5 |    |    |     |     |
| 6        |    |    | t1 | t2 | t3 | t4 | t5 | t6 |    |    |     |     |
| 7        |    | t1 | t2 | t3 | t4 | t5 | t6 | t7 |    |    |     |     |
| 8        | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 |    |    |     |     |
| 9        |    |    |    | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8  | t9  |
| 10       |    |    | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9  | t10 |
| 11       |    | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | t10 | t11 |

### 4.3.6 Segmentation Status Queues

**4.3.6.1 Entry Format** The CN8236 reports segmentation status to the host on one of 32 status queues. Each entry on the queue is two words. [Tables 4-20](#) and [4-21](#) describe the format of a standard SEG status queue entry. A second special status queue format for ACR/ER change notification is described in [Tables 4-22](#) and [4-23](#). Refer to [Section 6.3.7.5](#) for a description of the trigger mechanism for posting special status.

**Table 4-20. 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       | NCR=0 | STOP | DONE | SINGLE | OVFL | I_EXP |    |    |    |    | I_MAN | Reserved |    |    |    | SEG_VCC_INDEX |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

**Table 4-21. 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.<br>1 = Entry valid. Indicates that the host may process the entry.<br>Written to 1 by the SAR. Written to 0 by the host.                                  |
| NCR           | When set to a 0, indicates a normal status queue entry.   |
| 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 <a href="#">Chapter 6.0</a> .)   |
| 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 <a href="#">Chapter 6.0</a> .)  |
| SEG_VCC_INDEX | Segmentation VCC index on which the SAR transmitted the buffer or PDU.  |

Table 4-22. Segmentation Status Queue Format for ACR/ER

|      |     |       |         |              |          |      |          |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------|-----|-------|---------|--------------|----------|------|----------|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Word | 31  | 30    | 29      | 28           | 27       | 26   | 25       | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15            | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0    | ER  |       |         |              |          |      |          |    |    |    |    |    |    |    |    |    | ACR           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 1    | VLD | NCR=1 | NCR_DIR | SRC_NOT_DEST | Reserved | OVFL | Reserved |    |    |    |    |    |    |    |    |    | SEG_VCC_INDEX |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

Table 4-23. Status Queue Entry Field Descriptions for ACR/ER

| Field Name    | Description   |
|---------------|---|
| ER            | For SRC_NOT_DEST = 0, reflects TA_ER, current ER field in BCK RM cell. Not valid for SRC_NOT_DEST = 1.                                      |
| ACR           | For SRC_NOT_DEST = 0, reflects TA_CCR current CCR field in BCK RM cell.<br>For SRC_NOT_DEST = 1, reflects current CCR field in FWD RM cell. |
| VLD           | Entry valid. Written to 1 by the SRC. Written to 0 by the host.   |
| NCR           | When set to a 1, indicates that status entry is an ACR Notification status entry.   |
| SRC_NOT_DEST  | A value of 1 indicates a source ACR change notification.<br>A value of 0 indicates a destination ACR change notification.                   |
| NCR_DIR       | A logic high indicates latest NCR threshold crossed was NCR_HI, a logic low indicates NCR_LO threshold was crossed last.                    |
| OVFL          | Overflow: status entry is last entry available.   |
| SEG_VCC_INDEX | Segmentation VCC index for buffer.  |

#### 4.3.6.2 Status Queue Management

At initialization, the host assigns the location and size of up to 32 queues by initializing internal registers, the segmentation status queue base table entries. 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.

Tables 4-24 and 4-25 describe the format of these entries.

Table 4-24. 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 |      |       |    |    |    |    |    |          |    |    |    |    |    |         |    | Reserved |    | LOCAL |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 1    | SIZE      | Rsvd | WRITE |    |    |    |    |    | Reserved |    |    |    |    |    | READ_UD |    |          |    |       |    |    |    |   |   |   |   |   |   |   |   |   |   |

Table 4-25. Segmentation Status Queue Base Table Entry Field Descriptions

| Field Name | Description   |
|------------|---|
| BASE_PNTR  | Points (Bits 31:2) to base of status queue. Bits 1:0 are always 0 (word-aligned).   |
| LOCAL      | 0 = Status queue located in PCI address space.<br>1 = Status queue located in SAR-shared memory address space.<br>This bit should be set to 0 for a write-only PCI host architecture. |
| SIZE       | Number of entries in this status queue:<br>00 = 64<br>01 = 256<br>10 = 1,024<br>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.  |

#### 4.3.6.3 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 CN8236 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 CN8236 inhibits segmentation on all channels that report on the overflowed status queue. All other channels are unaffected.

### 4.3.7 Segmentation Internal SRAM Memory Map

As indicated in [Table 4-26](#), the segmentation internal SRAM is in the address range 0x1400–0x1617.

The segmentation status queue base table registers (SEG\_ST\_QUn) are in the address range 0x1400–0x14FF.

The internal transmit queue base table registers (SEG\_TQ\_QUn) are in the address range 0x1500–0x15FF.

AALx registers are in the address range of 0x1600–0x1617.

Other internal segmentation and scheduler registers are in the address range 0x1618–0x17FF.

**Table 4-26. 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                    |
| ⋮   | ⋮           | ⋮  |
| 0x14F8–0x14FF   | SEG_SQ_QU31 | Status Queue 31 Base Table                   |
| <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                  |
| ⋮   | ⋮           | ⋮  |
| 0x15F8–0x15FF   | SEG_TQ_QU31 | Transmit Queue 31 Base Table                 |
| 0x1600–0x1603   | AALx_AD0    | PCI address for AALx 0 connected to HFIFOWR0 |
| 0x1604–0x1607   | AALx_AD1    | PCI address for AALx 1 connected to HFIFOWR1 |
| 0x1608–0x160B   | AALx_AD2    | PCI address for AALx 2 connected to HFIFOWR2 |
| 0x160C–0x160F   | AALx_AD3    | PCI address for AALx 3 connected to HFIFOWR3 |
| 0x1610–0x1613   | AALx_AD4    | PCI address for AALx 4 connected to HFIFOWR4 |
| 0x1614–0x1617   | AALx_AD5    | PCI address for AALx 5 connected to HFIFOWR5 |



# 5.0 Reassembly Coprocessor

---

## 5.1 Overview

The reassembly (RSM) coprocessor processes cells received from the ATM PHY 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 CN8236 supports AAL5, AAL3/4, and AAL0 reassembly, and cell mode (1-cell PDUs through a Virtual FIFO buffer, for CBR voice traffic).

The CN8236 reassembles up to 64 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 32 independent buffer pools, from which the RSM coprocessor draws buffers as needed.

The CN8236 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 and AAL information are passed to the host on one of 32 independent status queues.

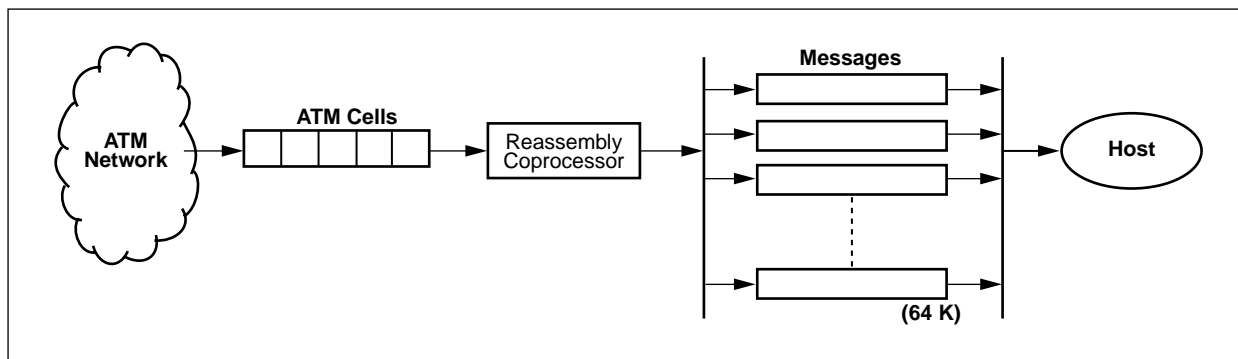
This chapter provides information on the functions and data structures of the reassembly coprocessor. For detailed information on how the CN8236 handles PM cells, deals with OAM functions, and interacts with the segmentation coprocessor in handling traffic management and scheduling, refer to [Chapter 6.0](#), and [Chapter 7.0](#).

## 5.2 Reassembly Functional Description

Each cell received from the ATM PHY Interface block belongs to any one of a possible 64 K virtual channels, or simultaneous messages. Due to the asynchronous nature of ATM, the cell contained in any incoming cell slot can belong to any VCC. Thus, the reassembly coprocessor must assign each arriving cell to the proper VCC, thereby de-multiplexing the incoming messages.

Figure 5-1 illustrates the basic reassembly process flow.

Figure 5-1. Reassembly—Basic Process Flow



8236\_020

### 5.2.1 Reassembly VCCs

As with segmentation VCCs, the CN8236 supports up to 64 K reassembly VCCs, referenced by VCC\_INDEX, which identifies a location in the reassembly VCC table.

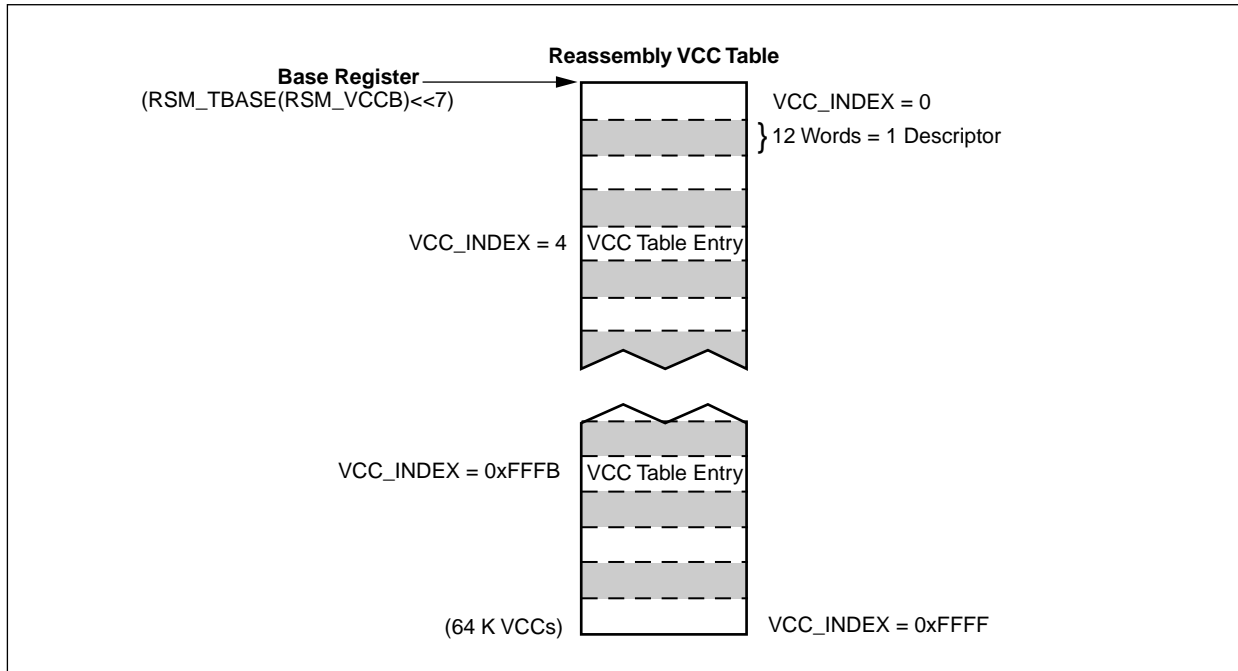
Each entry in the reassembly VCC table consists of 12 words and describes a single VC. Each VC can be processed as either AAL5, AAL3/4, or AAL0. AAL0 VCs can optionally be treated as Virtual FIFO buffers.

While the CN8236 accepts any reassembly 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 SAR-shared memory available in which to allocate and create RSM VCC tables, free buffer queues, RSM status queues, etc.



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

Figure 5-2. Reassembly VCC Table



8236\_021

**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 CN8236 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.

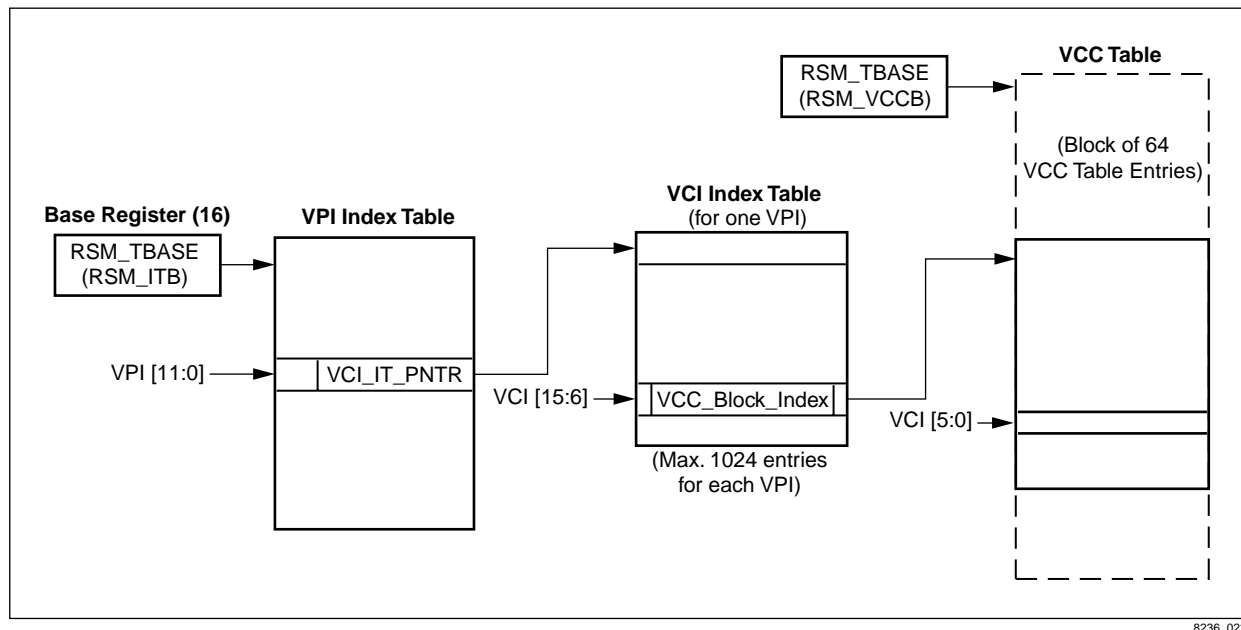
## 5.2.2 Channel Lookup

The CN8236's reassembly coprocessor implements a VPI/VCI index table mechanism using 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/VCI 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



8236\_022

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

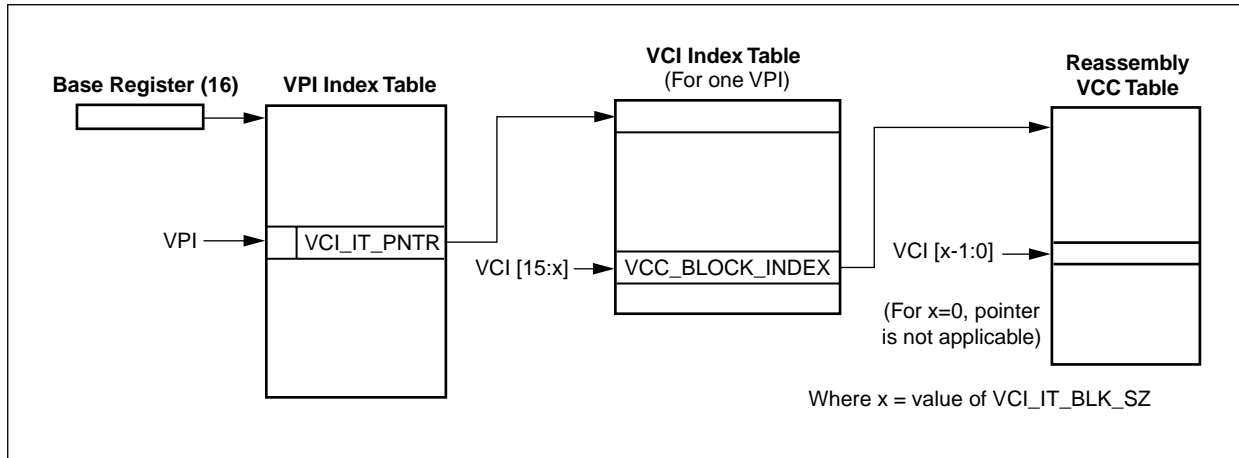
Some users might have a requirement or desire to limit the amount of memory allocated to VPI/VCI channel lookup. To enable this, the CN8236 provides the user with the choice of enabling an alternative scheme for the memory allocation and table handling involved in the Direct Index Lookup mechanism. In 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 thus allocates 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. Figure 5-4 illustrates the alternate lookup mechanism.

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



8236\_103

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 CN8236 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 CN8236 treats 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), which 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 5.2.2.2](#) 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. VCIs are limited by VCI\_RANGE in the VPI Index table entry and 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) x 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 CN8236 then reads the appropriate entry in the VCI Index table. The address of the VCI Index table entry is derived as follows:

$$\text{VCI\_IT\_PNTR} \times 4 + \text{VCI}[15:x] \times 4$$

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.

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

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 CN8236 increments the CELL\_RCVD\_CNT counter.

**5.2.2.4 AAL3/4 Lookup**

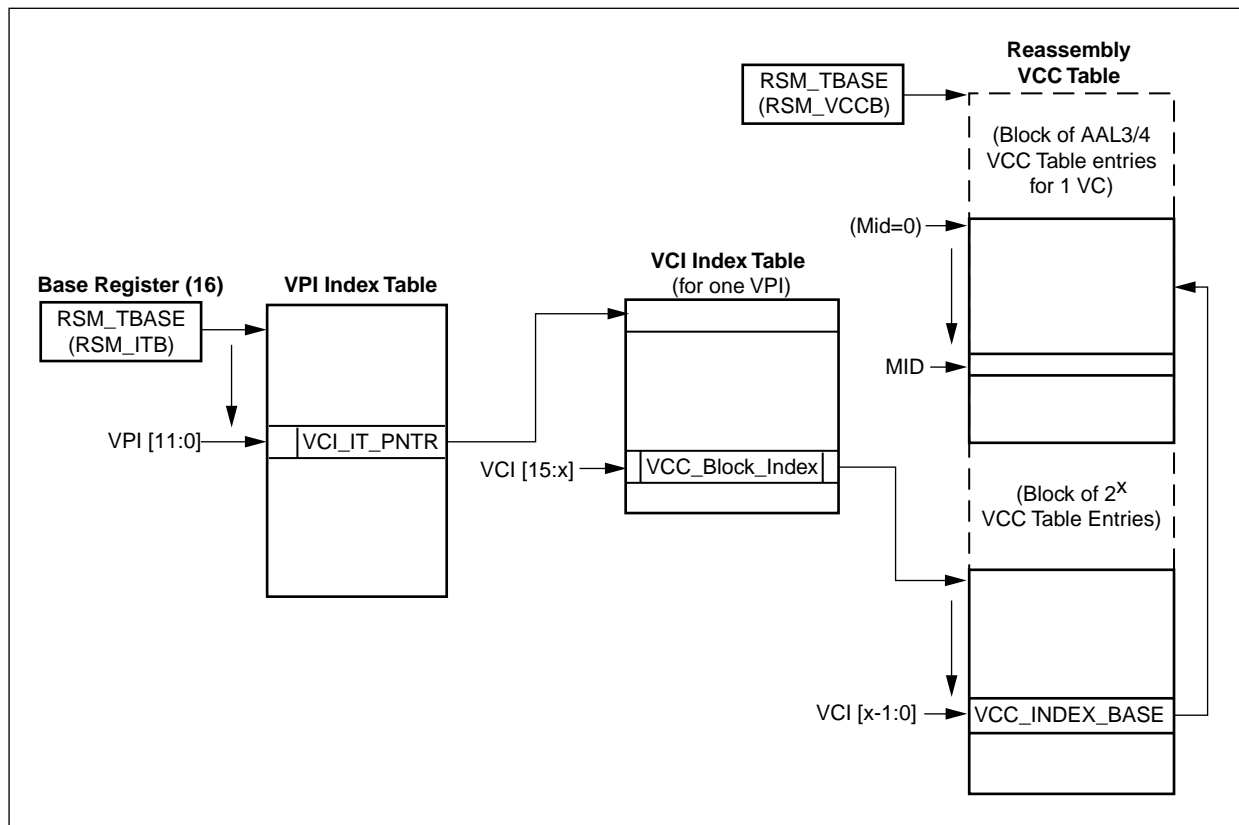
AAL3/4 MID multiplexing requires an additional level of indirection in channel lookup for received AAL3/4 traffic. This is because one Virtual Connection can have a multiple number of connectionless messages (like SMDS datagrams) multiplexed onto that one received connection. Each of these long SDUs is identified by its MID value. Thus, the VCC lookup also includes the MID value in the lookup mechanism.

The VPI Index table and VCI Index table lookup is performed exactly as described in Section 5.2.2.4. However, for AAL3/4 connections, VCC\_BLOCK\_INDEX + VCI[5:0] points to an AAL3/4 Head RSM VCC table entry (see Table 5-14).

The AAL3/4 Head entry contains the VCC\_INDEX\_BASE field which points to the first entry of a block of AAL3/4 entries for one Virtual Connection, with the MID value = 0.

VCC\_INDEX\_BASE + MID points to the RSM VCC table entry for the received cell's Virtual Connection and MID. The AAL3/4 lookup mechanism is illustrated in Figure 5-5.

Figure 5-5. Direct Index Lookup Method for AAL3/4

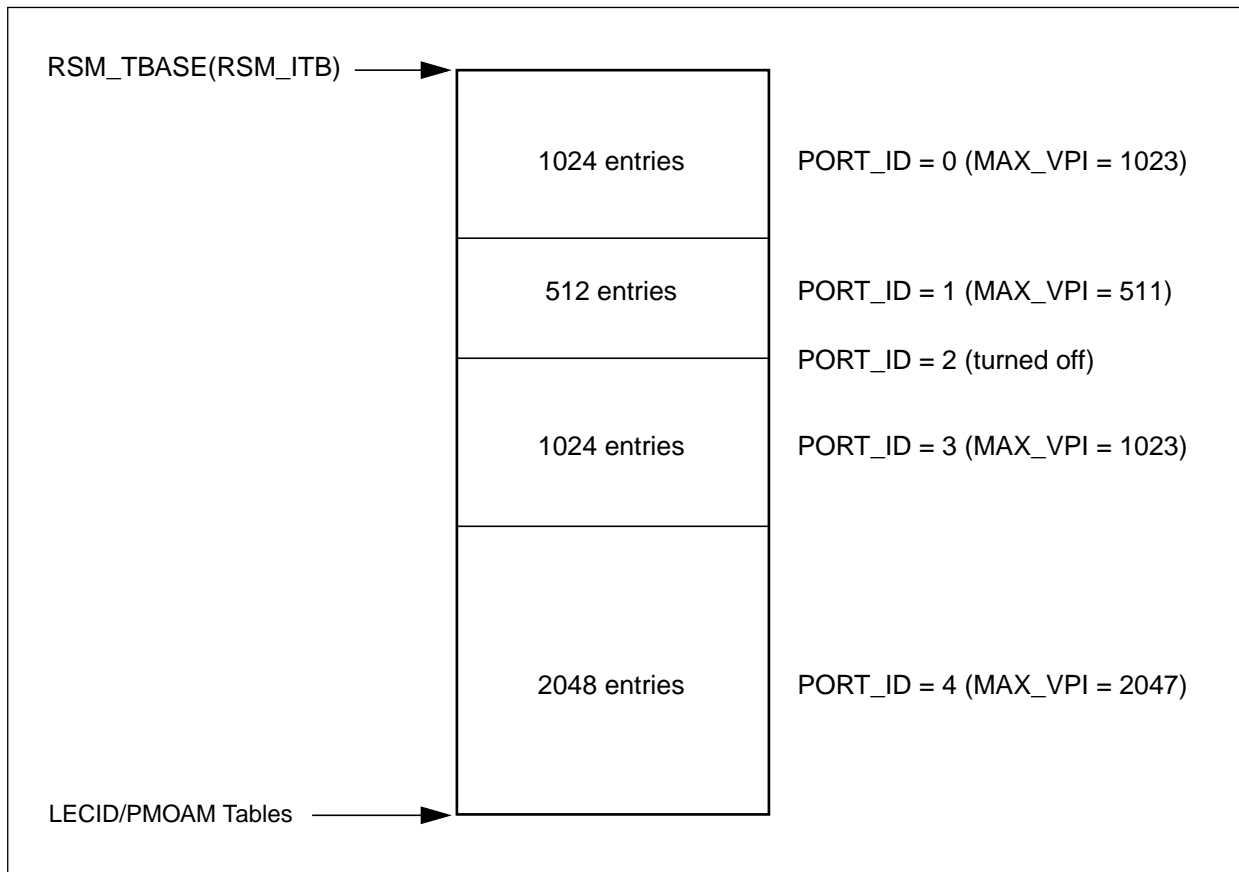


8236\_023

### 5.2.2.5 Variable VPI/PORT\_ID Lookup (Multi-phy Support)

In order to support multi-phy operation and/or reduce memory requirements of the VPI Index table, a new Variable VPI/PORT\_ID lookup mechanism has been implemented. This mechanism is enabled by setting  $RSM\_CTRL1(EN\_VPI\_SIZE) = 1$ . When enabled, a new register called  $VPI\_SIZE$  determines the maximum allowable VPI on each port in binary increments up to 4095. In addition, the VPI can be turned off. Figure 5-6 shows the VPI INDEX table with  $CONFIG1(NUM\_PORTS) = 4$  and  $VPI\_SIZE = 0x000b\_a09a$ . For all ports  $\geq CONFIG1(NUM\_PORTS)$ , their  $VPI\_SIZE$  entry should be set to  $0x0$  in order to turn it off.

Figure 5-6. VPI Index Table with Multiple Ports



8236\_024

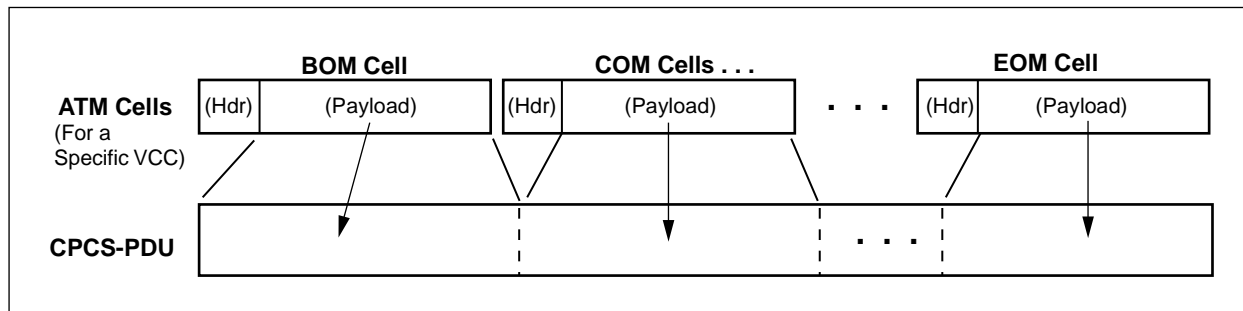
## 5.3 CPCS-PDU Processing

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, reassembly into an AAL3/4 PDU, or reassembly into a transparent AAL0 PDU. The exact process is governed by the AAL type described in detail in the subsections below.

Figure 5-7 illustrates the basic process function.

Figure 5-7. CPCS-PDU Reassembly



8236\_025

### 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 Permanent 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, and the associated segmentation VCC index (SEG\_VCC\_INDEX) for full duplex connections.

### OPERATION

Once the reassembly process is activated, this RSM VCC table entry is used to track the current state of the connection and direct the CN8236 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, and counts the length of the PDU. The CRC-32 value is collected in an accumulator, and the LENGTH value is collected in a Length Counter. After each received cell is processed, the reassembly coprocessor writes the CRC-32 and LENGTH values to the reassembly VCC state table entry for that channel.

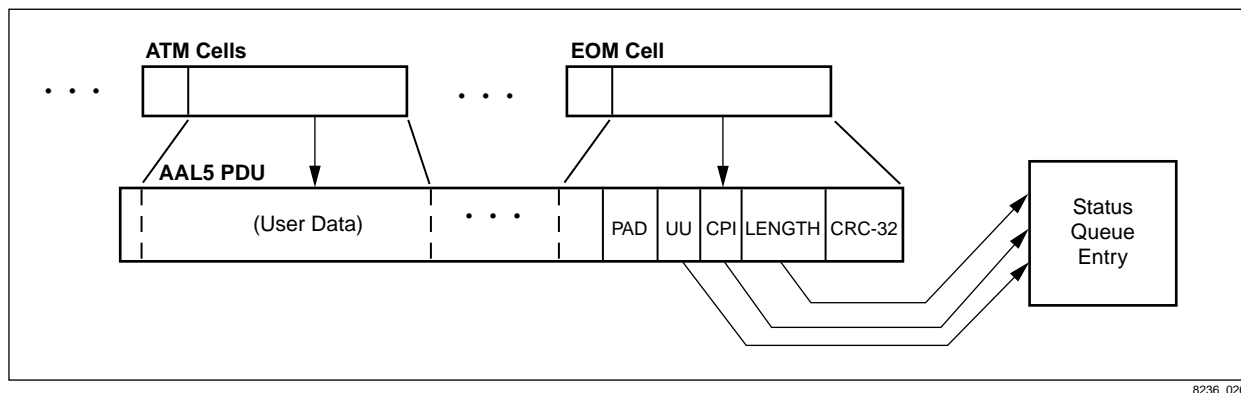
#### 5.3.1.2 AAL5 EOM Processing

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:

- UU information
- CPI field
- LENGTH field

[Figure 5-8](#) illustrates these process functions.

*Figure 5-8. AAL5 EOM Cell Processing—Fields to Status Queue*



8236\_026

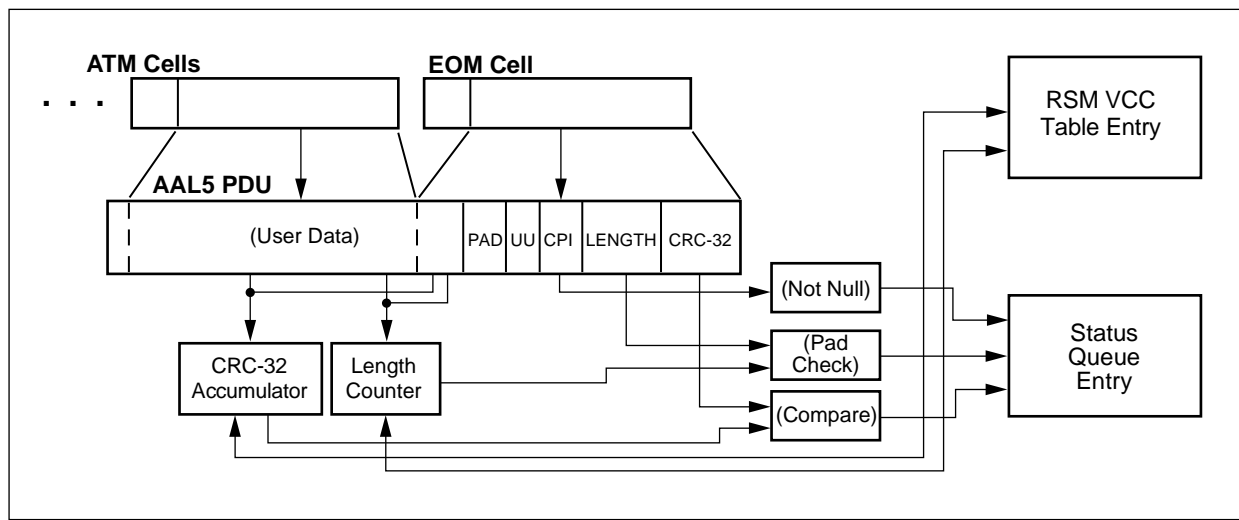
When the EOM cell is processed, the reassembly coprocessor performs the following checks:

- If the LENGTH field in the trailer of the AAL5 PDU is 0, the reassembly coprocessor sets the ABORT bit in the status queue entry to a logic high.
- 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 0 or greater than the 47 the reassembly coprocessor sets, the PAD\_ERROR bit in the status queue entry to a logic high.
- If the CPI field in the AAL5 trailer is not 0, the CPI\_ERROR bit in the status queue entry is set to a logic high.
- All of the AAL5 trailer information is also written into the end of the PDU buffer(s) in memory.



Figure 5-9 illustrates these process functions.

Figure 5-9. AAL5 Processing—CRC and PDU Length Checks



8236\_027

The CN8236 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 CN8236 receives a non-EOM cell, where

$$\text{TOT\_PDU\_LEN} + 48 > \text{MAX\_LEN} \times 1024 \text{ (or } 65568)$$

Early Packet Discard is performed.

The CN8236 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

$$\text{TOT\_PDU\_LEN} + 48 > \text{MAX\_LEN} \times 1024 \text{ (or } 65568)$$

the PDU is completed, with BA\_ERROR status bit set.

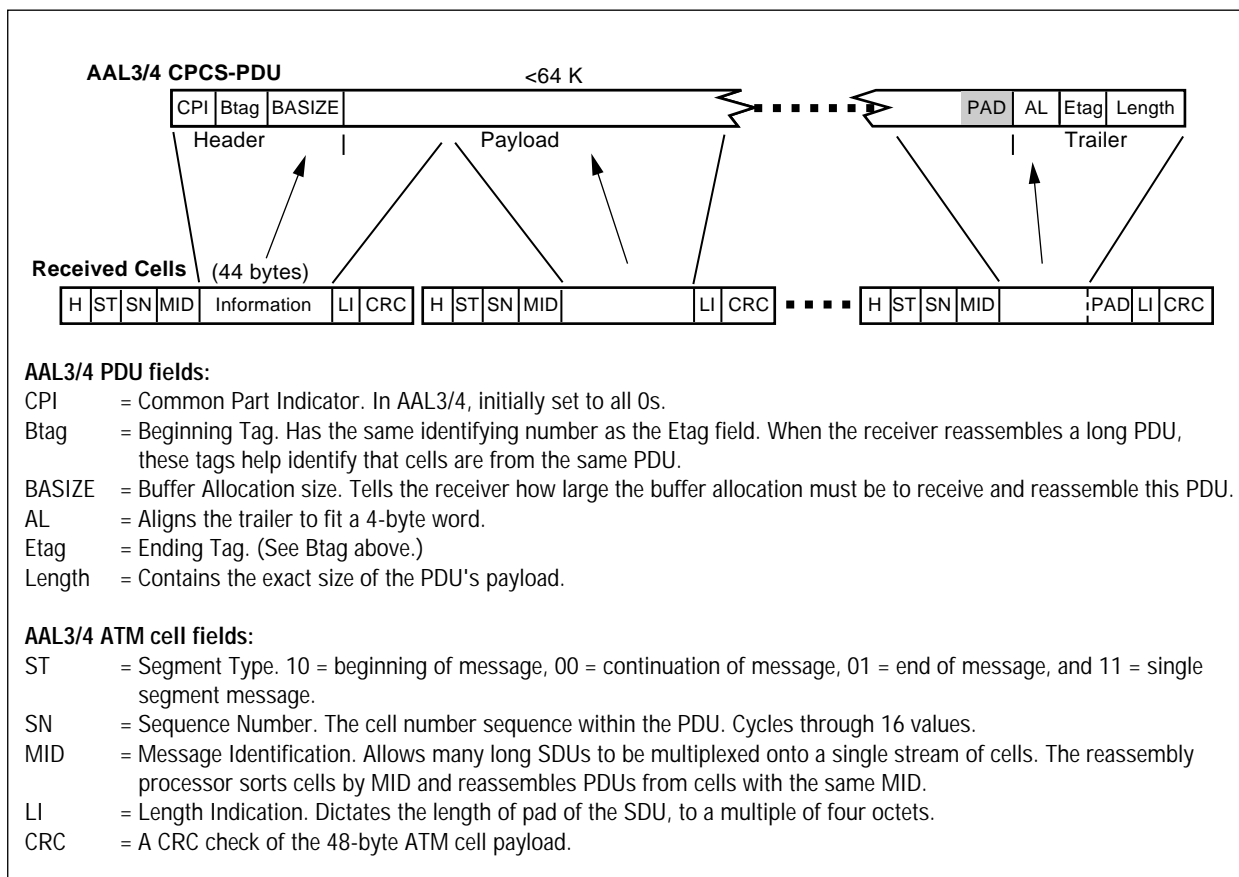
### 5.3.2 AAL3/4 Processing

The BOM cell contains the header information for the AAL3/4 PDU, and the EOM cell contains the trailer information for the PDU. All of the other cell payloads for that PDU contain user data. The reassembly coprocessor writes all header, trailer, and 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 AAL3/4 PDU.

The BOM cell for a PDU sets the Segment Type (ST) field in the cell to b10, and the EOM cell sets the ST field set to b01. All COM cells set the ST field set to b00. An ST field value of b11 indicates a Single Segment Message (SSM); that is, the cell holds all of a short PDU.

[Figure 5-10](#) illustrates an AAL3/4 CPCS-PDU reassembled from the received cell stream.

Figure 5-10. AAL3/4 CPCS—PDU Reassembly



8236\_104

### 5.3.2.1 AAL3/4 Per-Cell Processing

The following processing steps and checks occur on a per-cell basis:

- If the CRC10\_EN bit in the AAL3/4 Head VCC table entry is a logic high, the CRC10 field in the cell is checked. If in error, the SAR increments the CRC10\_ERR counter in the VCC Head table entry, and discards the cell.
- The MID field value is checked against active MID values specified by the MID\_BITS and MID0 fields in the AAL3/4 Head VCC table entry. If invalid, the SAR discards the cell, and increments the MID\_ERR counter in the VCC Head table entry.
- If the cell is an EOM cell and LI = 63, the SAR writes a status queue entry with ABORT bit set, CPCS\_LENGTH=0, and BD\_PNTR pointing to the partially reassembled PDU.
- If the LI\_EN bit in the AAL3/4 Head VCC table entry is a logic high, the LI field is checked, and the following error detection and error processing is done:
  - If the cell received is a BOM, and LI  $\neq$  44; the SAR discards the cell, and increments the LI\_ERR counter in the VCC Head table entry.
  - If the cell received is a COM, and LI  $\neq$  44; the SAR discards the cell, and terminates the current CPCS-PDU. The SAR also writes a status queue entry with the LI\_ERROR bit set, the CPCS\_LENGTH = 0, and BD\_PNTR pointing to the partially reassembled PDU. The SAR also increments the LI\_ERR counter in the VCC Head table entry.
  - If the cell received is an EOM, and (LI < 4 or LI > 44); the SAR discards the cell, and terminates the current CPCS-PDU. The SAR also writes a status queue entry with the LI\_ERROR bit set, the CPCS\_LENGTH = 0, and BD\_PNTR pointing to the partially reassembled PDU. The SAR also increments the LI\_ERR counter in the VCC Head table entry.
  - If the cell received is an SSM, and (LI < 8 or LI > 44); the SAR discards the cell, and increments the LI\_ERR counter in the VCC Head table entry.
- If the ST\_EN bit in the AAL3/4 Head VCC table entry is a logic high, the ST field in the cell is checked. The NEXT\_ST field in the VCC entry is used for this check. A value of 01 in the NEXT\_ST field indicates that the SAR was expecting a BOM/SSM cell. An 00 value indicates that the SAR was expecting a COM/EOM cell. The following error detection and error processing is done:
  - If the cell received is a BOM, and the SAR was expecting a COM or EOM, the SAR terminates the current CPCS-PDU and writes a status queue entry with the ST\_ERROR bit set, the CPCS\_LENGTH = 0, and the BD\_PNTR pointing to the partially reassembled PDU. The SAR also increments the BOM\_SSM\_ERR counter in the VCC Head table entry, and starts a new CPCS-PDU with the current BOM cell.
  - If the cell received is an SSM, and the SAR was expecting a COM or EOM, the SAR terminates the current CPCS-PDU, and writes a status queue entry with the ST\_ERROR bit set, the CPCS\_LENGTH =0, and the BD\_PNTR pointing to the partially reassembled PDU. The SAR also increments the BOM\_SSM\_ERR counter in the VCC Head table entry, and processes the current cell as a valid CPCS-PDU.
  - If the cell received is a COM, and the SAR was expecting a BOM or SSM, the SAR discards the cell.

- If the cell received is an EOM, and the SAR was expecting a BOM or SSM, the SAR discards the cell, and increments the EOM\_ERR counter in the VCC Head table entry.
- If the SN\_EN bit in the AAL3/4 Head VCC table entry is a logic high, the SN field in the cell is checked. If the cell received is an COM or EOM, and the SN field does not equal the NEXT\_SN field in the VCC table entry, the SAR discards the cell, terminates the current CPCS-PDU, and writes a status queue entry with the SN\_ERROR bit set, CPCS\_LENGTH = 0, and the BD\_PNTR pointing to the partially reassembled PDU. The SAR also increments the SN\_ERR counter in the VCC Head table entry.

### 5.3.2.2 AAL3/4 Additional BOM/SSM Processing

The CN8236 performs the following additional checks and functions on each BOM or SSM cell received:

- When a BOM cell is received for an AAL3/4 CPCS-PDU (that is, with ST field set to b10), the CN8236 checks if the CPI\_EN bit in the RSM AAL3/4 Head VCC table entry is set to a logic high. If so, the CPI field in the received cell is checked for a 0 value. If not a 0 value, the CN8236 treats this as an error condition and discards the cell, terminates the current CPCS-PDU, writes a status queue entry with the CPI\_ERROR bit set, and the CPCS\_LENGTH and BD\_PNTR fields set to 0. Cells up to and including the next EOM are discarded.
- The CN8236 also checks if the BAH\_EN bit in the RSM AAL3/4 Head entry is set to a logic high. If so, it checks if the BASIZE field in the CPCS-PDU header is less than 37 octets, and if so, the CN8236 discards the current cell, terminates the current CPCS-PDU, and writes a status queue entry with the BA\_ERROR bit set, and the CPCS\_LENGTH and BD\_PNTR fields set to 0. Cells up to and including the next EOM are discarded.
- If the CPI and BASIZE fields are correct in the BOM cell, the CN8236 copies the BASIZE and BTAG fields into the VCC table entry for that MID, and sets the NEXT\_ST and NEXT\_SN values in the VCC table entry. It also writes the CPCS-PDU header into the data buffer.
- If the LI field in the SAR-PDU > (BASIZE+7), the SAR discards the cell, terminates the CPCS-PDU, and writes a status queue entry with the LEN\_ERROR bit set and CPCS\_LENGTH = 0.

### 5.3.2.3 AAL3/4 Additional COM Processing

The CN8236 performs the following additional checks and functions on each COM cell received:

- The CN8236 checks if the sum of the LI fields for the CPCS-PDU are greater than (BASIZE + 7). If so, the CN8236 discards the cell, terminates the CPCS-PDU, and writes a status queue entry with the LEN\_ERROR bit set high, CPCS\_LENGTH set to 0, and BD\_PNTR pointing to the partially reassembled PDU.

#### 5.3.2.4 AAL3/4 Additional EOM/SSM Processing

Upon termination of a CPCS-PDU, the CN8236 performs the following additional checks and functions on each EOM or SSM cell received:

- The Length field in the CPCS-PDU trailer is written to the CPCS\_LENGTH field of a RSM status queue entry written for the VCC.
- The CN8236 performs a PAD length check to see if the sum of all LIs for the CPCS-PDU – LENGTH – 8 = [0 to 3] octets. If in error, it sets the PAD\_ERROR bit in the status queue entry.
- The CN8236 performs a Modulo 32 bit check. If the sum of all LIs for the CPCS-PDU is not modulo 32 bit, the SAR sets the MOD\_ERROR bit in the status queue entry.
- If the Alignment (AL) field in the CPCS-PDU trailer is not all 0s, it sets the AL\_ERROR bit in the status queue entry.
- If the BTAG field in the RSM VCC table entry does not match the ETAG field in the CPCS-PDU trailer, it sets the TAG\_ERROR bit in the status queue entry.
- The CN8236 checks the BAT\_EN bit in the AAL3/4 Head VCC table entry. If BAT\_EN is high, it compares the BASIZE field to the Length field in the CPCS-PDU trailer. If not a match, it sets the BA\_ERROR bit in the status queue entry. If BAT\_EN is low, it checks if the Length field is > BASIZE; and if so, sets the BA\_ERROR bit in the status queue entry.
- The CN8236 writes the AAL3/4 CPCS-PDU trailer to the data buffer.

#### 5.3.2.5 AAL3/4 MIB Counters

Whenever an AAL3/4 MIB counter rolls over to all 0s, the CN8236 writes a RSM status queue entry with the CNT\_ROVR bit set to a logic high, and the HEAD\_VCC\_INDEX field pointing to the AAL3/4 Head VCC entry which contains the rolled-over counter. This processing step takes place for the CRC10\_ERR, MID\_ERR, LI\_ERR, SN\_ERR, BOM\_SSM\_ERR, and EOM\_ERR counters.

### 5.3.3 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.

#### 5.3.3.1 Termination Methods

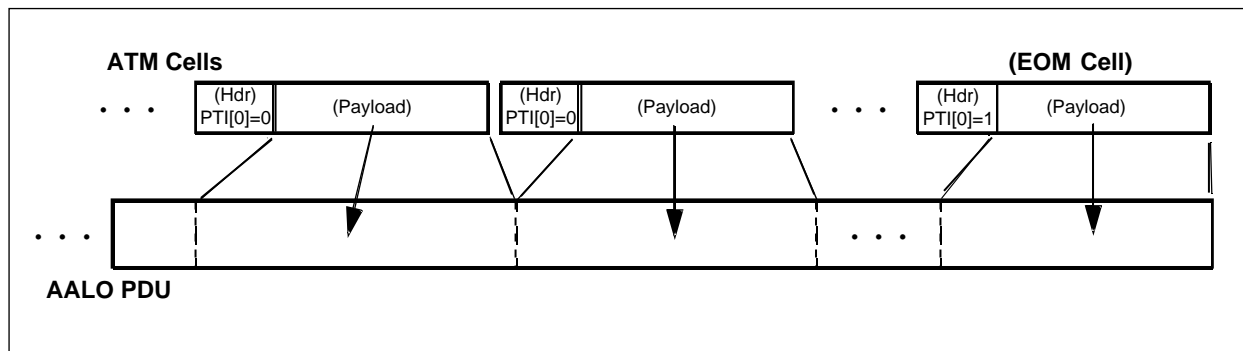
The CN8236 provides two methods of terminating an AAL0 PDU: Cell Count EOM and PTI termination.

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

- If TCOUNT = non-0, Cell Count EOM PDU termination is enabled. PDUs 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 = 0, then PTI termination is enabled. In this case, a received cell with PTI[0] set to 1 indicates the end of the AAL0 message. The total maximum allowable length of an AAL0 PDU in this mode is (CCOUNT × 2) bytes.

Figure 5-11 provides an illustration of this.

Figure 5-11. AAL0 PTI PDU Termination



8236\_028

The CN8236 reports all PDU termination events, with or without errors, in a Status Queue entry for that channel. See Section 5.6.

#### 5.3.3.2 AAL0 Error Conditions

If the CN8236 receives a non-EOM cell in PTI termination mode, where

$$\text{TOT\_PDU\_LEN} + 48 > \text{CCOUNT} \times 2$$

EPD is performed. The CN8236 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

$$\text{TOT\_PDU\_LEN} + 48 > \text{CCOUNT} \times 2$$

the PDU is completed, with BA\_ERROR status bit set.

The CN8236 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.

### 5.3.4 ATM Header Processing

ATM level CI and CLP 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.

### 5.3.5 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 0x00. The STAT output truth table illustrated 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       |
| AAL0 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.

#### 5.3.5.1 Prepend Index

In order to allow protocol processing to start upon reception of the BOM cell, the VCC\_INDEX can be optionally prepended to the beginning of the BOM cell. This, in conjunction with the BOM SYNC signals via the STAT pins, can be used to eliminate the need for a host lookup process before starting protocol processing. When RSM\_CTRL0(PREPEND\_INDEX) is a logic high, the VCC\_INDEX is appended to the BOM cell as follows:

**Table 5-3. Prepend 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_INDEX |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

## 5.4 Buffer Management

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

*NOTE:* The process cycle time of a read transaction across the PCI bus is much longer than a write transaction, due to the PCI bus being held in a busy state while the remote processor accesses and processes the read request. Therefore, to speed up processing flow during reassembly, the CN8236 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.

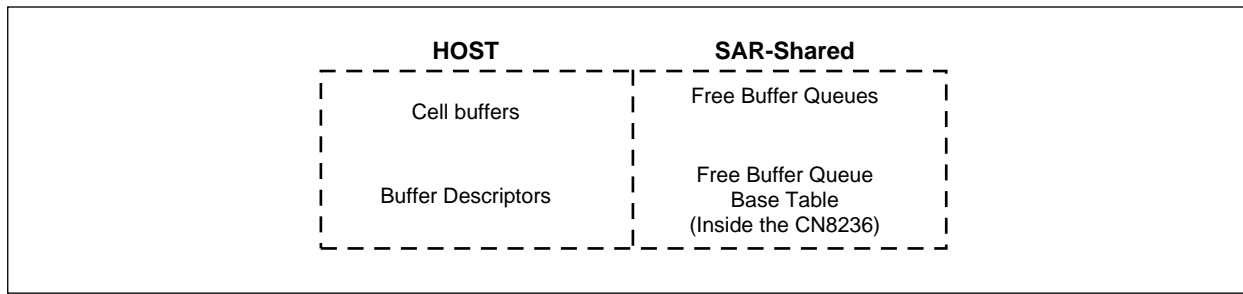
### 5.4.2 Scatter Method

The CN8236 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 5-12](#): 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 CN8236 allows for up to 32 independent free buffer queues.



Figure 5-12. Host and SAR-Shared Memory Data Structures for Scatter Method



8236\_029

### 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. BFR0 contains the BOM free buffer queue number, and BFR1 contains the COM free buffer queue number. Typically, the BFR0 number is for free buffer queues 0–15, and the BFR1 number is for free buffer queues 16–31.

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

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 32 free buffer queues as follows:

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

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

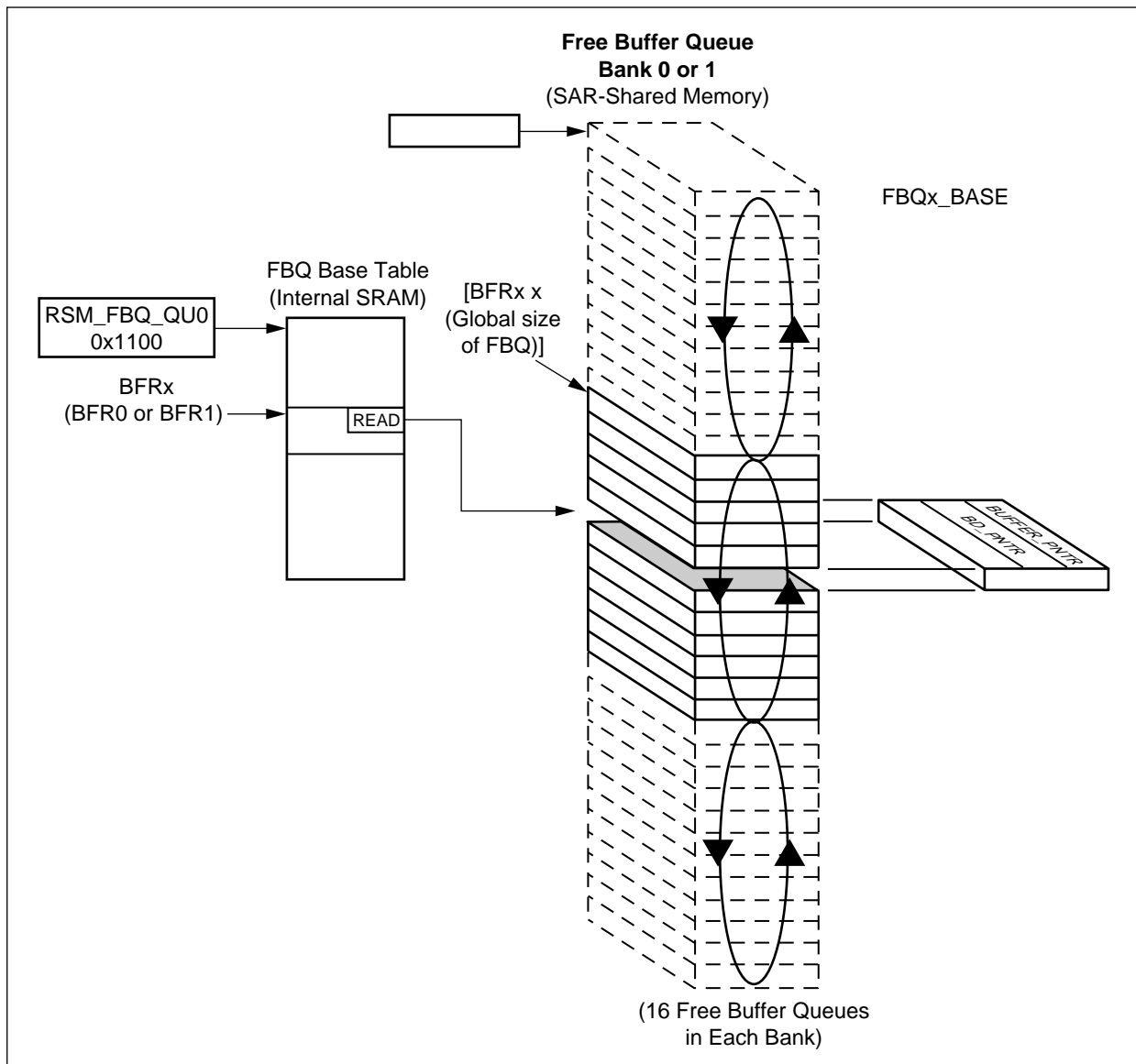
$$(\text{index of first entry for the queue}) + [(\text{READ index pointer}) \times (\text{size of each free buffer queue entry})]$$

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

Figure 5-13 illustrates this structure. Refer to Chapter 3.0, for more information on the operation of the free buffer queue.

Figure 5-13. Free Buffer Queue Structure



8236\_030

### 5.4.4 Linked Data Buffers

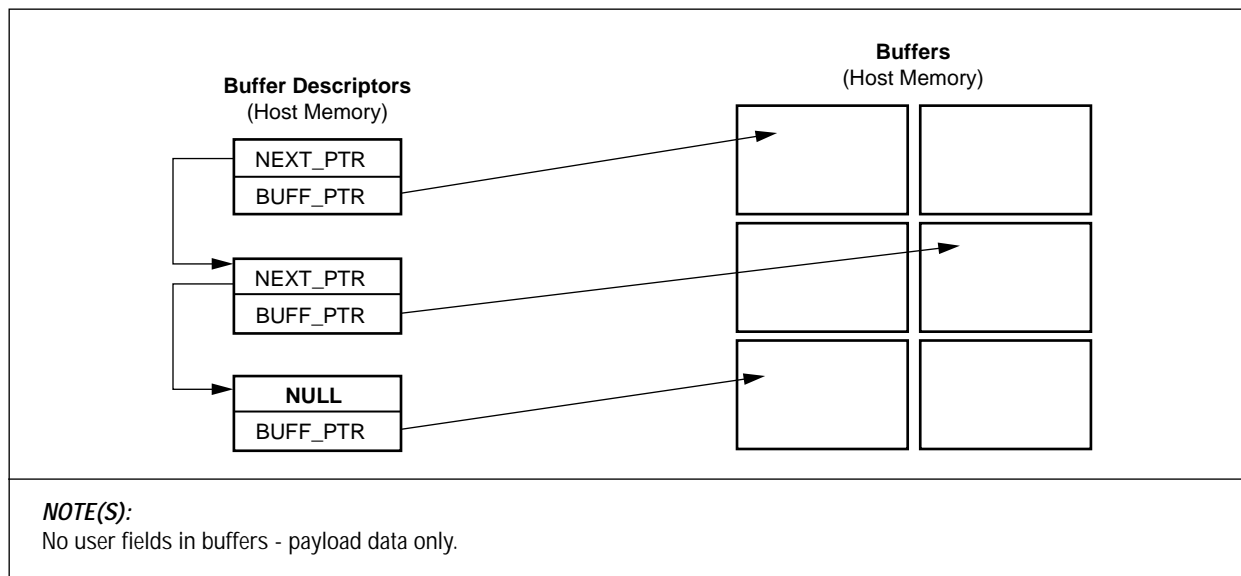
After the free buffer queue has returned pointers to a buffer descriptor and a cell buffer, the reassembly coprocessor writes payload data to the buffer.

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 5-14 illustrates this structure.

Figure 5-14. Data Buffer Structures



8236\_105

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 (that is, 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 (that is,  $RSM\_FQCTRL(FBQ0\_RTN) = 0$ ), and therefore, all free buffer queue entries are two words.

- 5.4.5.1 Buffer Descriptors** In every buffer descriptor entry, write the pointer to an available data buffer in the `BUFF_PTR` field. This assigns every data buffer to its own buffer descriptor.
- 5.4.5.2 Free Buffer Queue Base Table** Allocate the size (in number of entries) of each of the 32 free buffer queues in the `RSM_FQCTRL` register, (`FBQ_SIZE`) field, based on these values:
- 00 = 64
  - 01 = 256
  - 10 = 1,024
  - 11 = 4,096
- Initialize the free buffer queue update `INTERVAL`, that is, how many buffers are taken off the free buffer queue before the CN8236 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 0s.
- 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), the READ index pointer is written to the location pointed to by READ\_UD\_PNTR, and the UPDATE counter is reset to 0.

When the host wants to return a buffer to a free buffer queue, the host WRITE index pointer is compared to the CN8236 READ index pointer located at READ\_UD\_PNTR. If the WRITE index pointer + 1 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

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

If an empty condition occurs after the first buffer of a CPCS-PDU is written, the reassembly coprocessor performs early packet discard on the channel and writes 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.

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 automatically recovers 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 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), 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 is 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 following 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 is 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 use this feature to discard superfluous traffic on the ATM network caused by LAN Emulation Clients (LECs) transmitting multicast frames, that is, 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 is 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 Buffer Full

The purpose of this function is to allow a graceful recovery from an incoming DMA FIFO buffer full condition. Without this function, the reassembly coprocessor is stalled when the FIFO buffer 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 marks all channels that receive a cell during the full condition for subsequent early packet discard. Upon recovery from the full condition, the reassembly block performs early packet discard on the appropriate channels as cells are received on those channels. In addition, cells continue to be dropped on each channel until after an EOM cell is received for that channel. Early packet discard processing is delayed until recovery from the full condition, since the status entry also requires the use of the incoming DMA FIFO buffer.

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

The user can 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 can 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 buffer full condition is indicated by the FFPD bit in the RSM status queue entry being a logic high.

**5.4.8.6 AAL3/4 Early Packet Discard Processing**

The CN8236 performs EPD for AAL3/4 CPCS-PDUs with these steps:

- Sets the appropriate error bit in the PDU\_CHECKS field for a new RSM Status Queue entry.
- Sets the CPCS\_LENGTH field to 0.
- Sets the BD\_PNTR field to point to the partially reassembled PDU.
- Writes the Status Queue entry.
- Discards all cells of that PDU up to and including the EOM cell for that PDU.

**5.4.8.7 Error Conditions**

Partially reassembled CPCS-PDUs is 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 CN8236 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 CN8236 provides a status queue indication to the host. This indication allows the host to recover the buffers held by the partially completed PDU. The CN8236 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. Once the RSM\_TO register is enabled, it controls the process activity. The process is activated every RSM\_TO\_PER rising edges of SYSCLK on cell boundaries.

*NOTE:* GTO\_EN set to 0 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 0 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 CN8236 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 0, TO\_VCC\_INDEX is incremented by 1 and compared to RSM\_TO\_CNT in the RSM\_TO register. If TO\_VCC\_INDEX = RSM\_TO\_CNT, TO\_VCC\_INDEX is reset to 0, and the time-out search is restarted at the beginning of the VCC table.

If both bits are set, the CN8236 increments CUR\_TOCNT in the RSM VCC table entry. It then 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<sub>x</sub>, a time-out condition has occurred on the current VCC. The CN8236 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 1 in the RSM VCC table entry for the last VCC\_INDEX that the host needs enabled for time-out processing. When the CN8236 detects this bit set to 1, it halts time-out processing.

When time-out processing is halted, the time-out process is still activated, but the VCC is not checked for a time-out condition. The CN8236 simply increments TO\_VCC\_INDEX and compares it to RSM\_TO\_CNT. If they are equal, TO\_VCC\_INDEX is reset to 0, and the full time-out processing is re-enabled.

### 5.4.9.3 Timer Reset

The CN8236 reassembly time-out process increments the CUR\_TOCNT value. If it reaches a threshold value, a time-out condition has occurred. In AAL5 and AAL0, PTI termination modes, the reception of a non-EOM cell resets the counter.

#### 5.4.9.4 Reassembly Time-Out Condition

The CN8236 reports reassembly time-out conditions via the VCC's reassembly status queue. The TO bit in the STATUS field of the status queue entry is set to 1. In Message Mode, the BD\_PNTR points to the beginning of the partial buffer descriptor chain. In Streaming Mode, the BD\_PNTR points to the last buffer descriptor in the chain. The only other valid fields in the status queue entry are VCC\_INDEX and VLD.

Once status has been reported, the CN8236 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 that require time-out processing.

### 5.4.10 Virtual FIFO Buffer Mode

This mode provides a logical FIFO buffer 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 buffer 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

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

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

#### 5.4.10.3 Errors

When the FIFO buffer 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 accesses the FIFO buffer. External circuitry must also ensure that only complete cells are written into the host FIFO buffer.

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 4-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 is initially returned. Typically, this is 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

Whenever a buffer is taken off free buffer queues 0 through 15 during reassembly on a channel enabled for firewall processing, 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 16 through 31 without being firewalled.

If the RX\_COUNTER[15:0] for a channel is 0 when a buffer is required, the RSM coprocessor declares a firewall condition. If the firewall condition occurs on a BOM or SSM, the CN8236 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 recovered to the free buffer queue, by writing the third word of the free buffer queue. The VCC\_INDEX is written to the channel to which credit is returned. The FWD\_VLD bit is set to a logic high, and the QFC bit is set to a logic low. The RSM coprocessor increments

the RX\_COUNTER[15:0] of the applicable channel. For proper operation of the update interval function, buffers must be returned at the same time as credits are returned.

Credits are returned to VCCs through Bank 0 free buffer queues. In order to return buffer credits independently from buffer usage, the CN8236 maintains a separate read pointer into free buffer queues that return credits. This pointer name is FORWARD, located in the free buffer queue base table entry. The host determines the number of Bank 0 free buffer queues that return credits by setting FWD\_EN in the RSM\_FQCTRL register.

The CN8236 snoops writes to free buffer queues that return firewall credits. When a write completes, the CN8236 begins processing firewall return credits on that queue. The third word of each entry is read, and if FWD\_VLD is set, a credit is added to the VCC\_INDEX indicated. The CN8236 continues to process credit return entries until FWD\_VLD is zero. Multiple free buffer queues might have credit return entries outstanding at one time. The CN8236 processes the entries according to the priority set in FWD\_RND in the RSM\_FQCTRL register. If FWD\_RND is a logic low, the CN8236 exhausts the credit returns on the highest number active queue before proceeding to other queues. Otherwise, it services 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 is returned. Typically, this is the first entry after the initial buffers placed on the queue.

## 5.5 Global Statistics

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

- CELL\_RCVD\_CNT—Number of cells received that map to active channels.
- CELL\_DSC\_CNT—Number of cells received that map to inactive channels. This includes idle cells, since those channels is turned off.
- AAL5\_DSC\_CNT—Number of AAL5 CPCS-PDUs discarded due to per channel firewall, buffer queue underflow, FIFO buffer 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 0 upon a reset and are not reset to 0 upon a read of the counter by the host. The counters roll over and optionally cause an interrupt upon rollover.

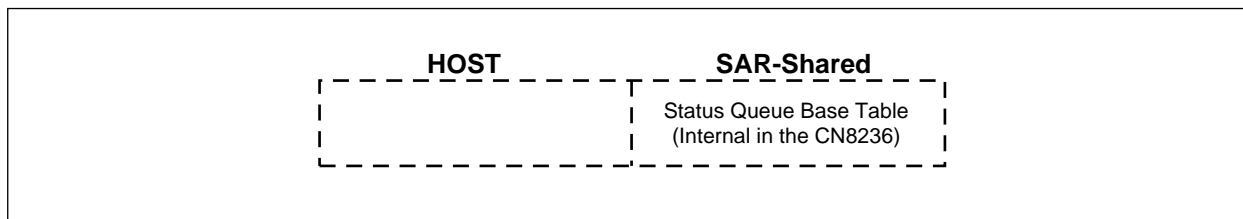
## 5.6 Status Queue Operation

The CN8236 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 provides data on the status of the reassembled PDU. These fields are then used by the host in directing further processing.

### 5.6.1 Structure

Two data structures are maintained as illustrated in [Figure 5-15](#), 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 32 independent circular status queues.

Figure 5-15. Data Structure Locations for Status Queues



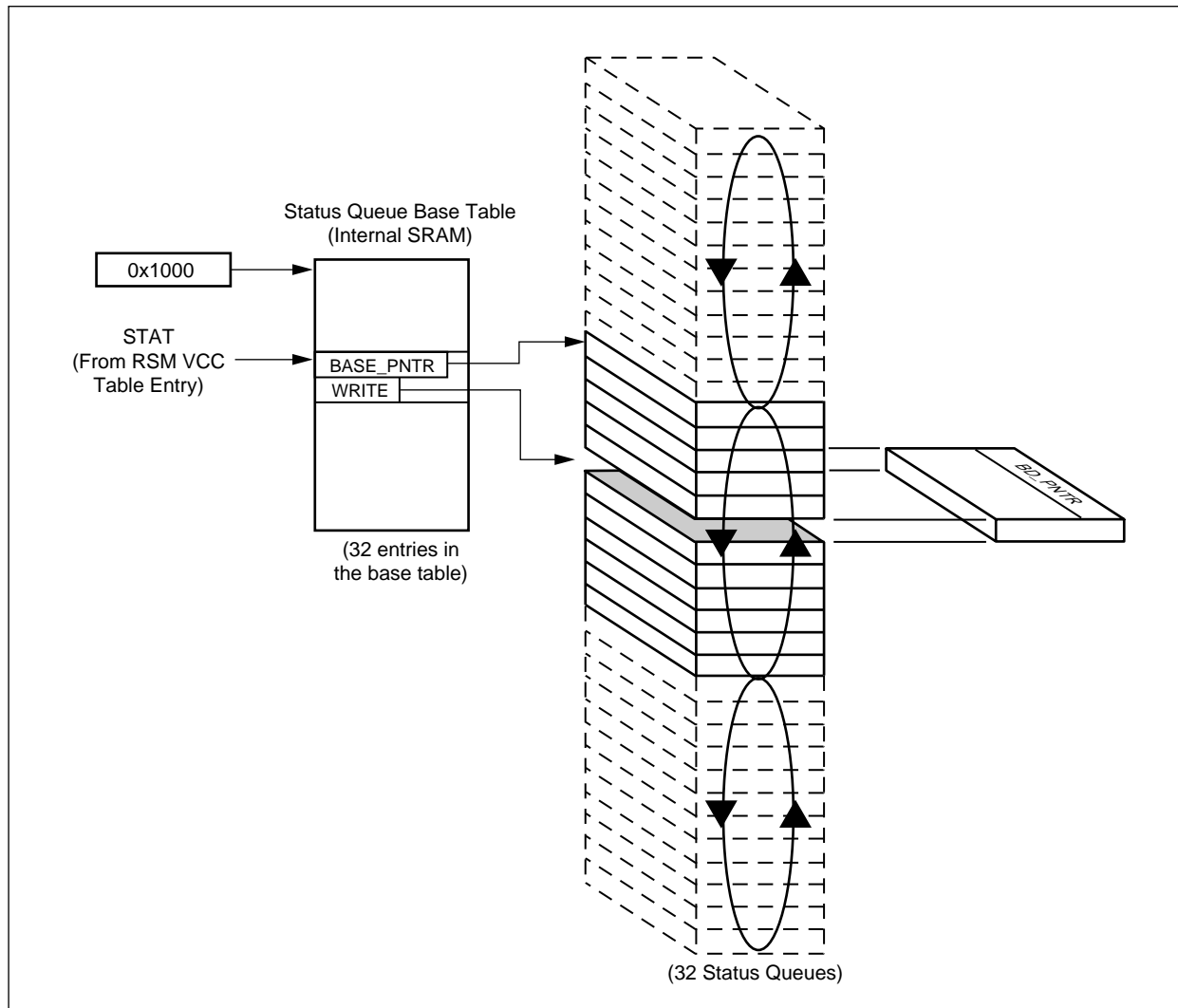
8236\_031

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

The BASE\_PNTR field in the status queue base table entry points to the base address of the status queue associated with that status queue base table entry. The WRITE field is the index pointer maintained by the CN8236, incremented each time a status queue entry is written, to point to the next status queue entry for that status queue.

Figure 5-16 illustrates this structure.

Figure 5-16. Status Queue Structure Format



8236\_032

#### 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 0s.
- 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.

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

Initialize the READ pointers to 0 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 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 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:

- 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 causes a status queue entry to be written. If a BOM cell is received and no early packet discards have occurred on channels mapped to the empty free buffer queue, status queue entry is written with the BOM and UNDF bits set to a logic high. In addition, 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 is 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[15:0]) to determine which host memory status queue(s) caused the interrupt.

*NOTE:* Only status queues 0 through 15 are reported in this register.

A typical operation for the interrupt manager is to only read HOST\_ISTAT1 upon receiving an interrupt, and periodically read HOST\_ISTAT0 to ensure that no error conditions have occurred. Once the interrupt manager has determined which status queue(s) caused the interrupt, the host starts 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+1 and READ\_UD index pointers. If the pointers are equal, a status overflow condition is detected and 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\_FULLL bit in the HOST\_ISTAT1 register, or the RSM\_LS\_FULLL 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. Once 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, the status-full condition is exited.

For multiple peer configurations, the user can configure an interrupt manager 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. Because 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 performs 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 are discarded. Status queue overflow protection can be globally disabled by setting RSM\_CTRL0(RSM\_STAT\_DIS) to a logic high.

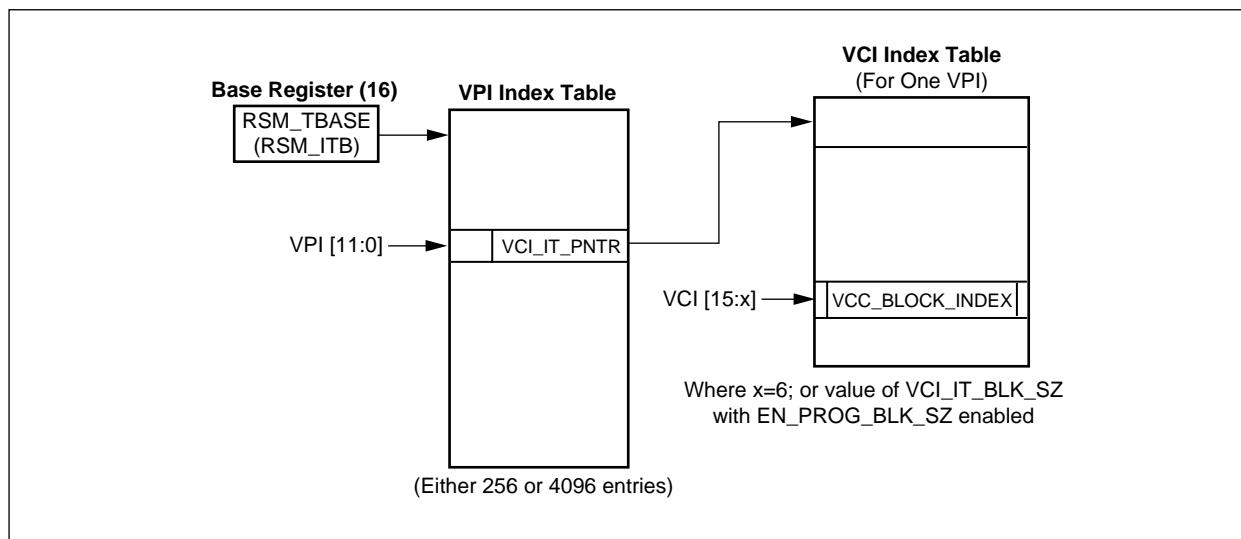
**NOTE:** Avoid having the Status Queue Overflow (or Full Condition) and DMA FIFO Buffer Full conditions at the same time.

## 5.7 Reassembly Control and Data Structures

### 5.7.1 Channel Lookup Structures

The reassembly coprocessor utilizes 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 5-17 illustrates this lookup mechanism.

Figure 5-17. VPI/VCI Channel Lookup Structure



8236\_106

Table 5-4 describes the normal VPI Index table format (one word per entry) without EN\_PROG\_BLK\_SZ (RSM\_CTRL1) enabled.

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

Table 5-6 describes the field definitions for the VPI Index table fields.

Table 5-4. 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 5-5. VPI Index Table Entry 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    | VP_EN    | Reserved |    |    |    |    |    |    |    |    |    | VCI_IT_PNTR |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 1    | Reserved |          |    |    |    |    |    |    |    |    |    |             |    |    |    |    | VCI_RANGE |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

Table 5-6. 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.<br>In normal operation, the 10 most significant bits can be set by the user, with the six least significant bit 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 0 bits to form a byte address.   |

Table 5-7 describes the VCI Index table format without the programmable VCC table and VCI Index table block size enabled.

Table 5-8 describes the VCI Index table format with EN\_PROG\_BLK\_SZ (RSM\_CTRL1) enabled.

Table 5-9 describes the VCI Index table Descriptions.

Table 5-7. 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 5-8. 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 5-9. 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.<br>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.<br>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, the resultant RSM VCC_INDEX pointer = {VCC_BLK_INDEX[15:4], CELL_VCI[3:0]}. |

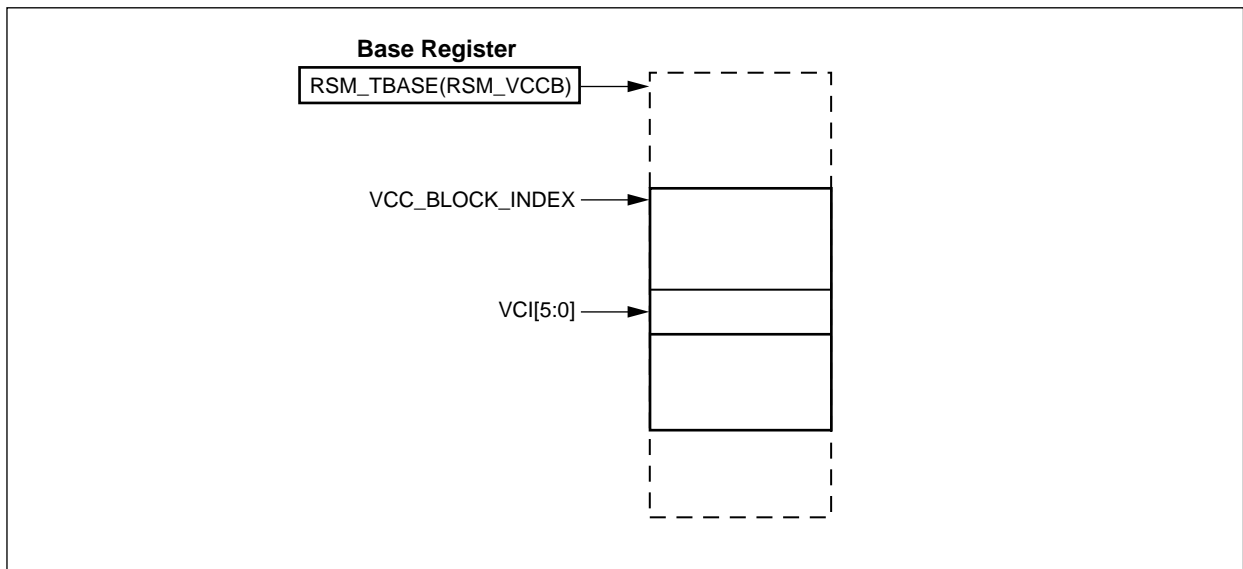
## 5.7.2 Reassembly VCC Table

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

There are three basic formats for the reassembly VCC table entry—AAL5, AAL0, and AAL3/4. Each completely describe the state of the reassembly process for individual VCCs. In addition, the AAL3/4 Head VCC table entry describes the AAL3/4-specific reassembly process state for an AAL3/4 VCC.

Figure 5-18 illustrates the VCC table entry lookup mechanism as a continuation from Figure 5-17.

Figure 5-18. Reassembly VCC Table Entry Lookup Mechanism



8236\_033

## 5.7.2.1 AAL5, AAL0 and AAL3/4 VCC Table Entries

Table 5-10 describes the format of AAL5 reassembly VCC table entries. Table 5-11 describes the format of AAL0 RSM VCC table entries. Table 5-12 describes the format of AAL3/4 RSM VCC table entries.

**KEY:**

= Written by host at VCC setup.

= May be dynamically modified during active reassembly.

Table 5-10. 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  | PASS_OAM | Reserved | FF_DSC | TO_INDEX             |          |            | PM_INDEX   |               |           |           | AAL_EN |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 1    | TO_LAST       | TO_EN        | CUR_TOCNT  |           |          |               |           |          |          |        | ER_EFCI              | Reserved |            |            |               | ABR_CTRL  |           |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 2    | PDU_FLAGS     |              |            |           |          |               | Reserved  |          |          |        | TOT_PDU_LEN          |          |            |            |               |           |           |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 3    | CRCREM        |              |            |           |          |               |           |          |          |        |                      |          |            |            |               |           |           |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 4    | CBUFF_LEN     |              |            |           |          |               |           |          | FW       | STAT   |                      |          |            | BFR1       |               |           |           | BFR0   |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 5    | CBUFF_PNTR    |              |            |           |          |               |           |          |          |        |                      |          |            |            |               |           |           |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 6    | BOM_BD_PNTR   |              |            |           |          |               |           |          |          |        |                      |          |            |            |               |           | Rsvd      |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 7    | CURR_BD_PNTR  |              |            |           |          |               |           |          |          |        |                      |          |            |            |               |           | Rsvd      | CBFR1  |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 8    | SEG_VCC_INDEX |              |            |           |          |               |           |          | SERV_DIS |        |                      |          |            |            |               |           |           |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 9    | Reserved      |              |            |           |          |               | ERS_INDEX |          |          |        | RX_COUNTER/VPC_INDEX |          |            |            |               |           |           |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 10   | D_EN_NCR      | ACR_NOT_ER   | D_NCR_TRIG | D_NCR_DIR | Reserved |               |           |          | CONG_ID  |        |                      |          | EN_EXP_CNG | EN_IMP_CNG | EXP_TA_CI     | EXP_TA_NI | EXP_TA_ER |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 11   | Rsvd          | D_NCR_HI_EXP |            |           |          | D_NCR_HI_MANT |           |          |          | Rsvd   | D_NCR_LO_EXP         |          |            |            | D_NCR_LO_MANT |           |           |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

Table 5-11. 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 | PASS_OAM | AALx_EN | FF_DSC | TO_INDEX             |            |              | PM_INDEX  |           |               |    |          | AAL_EN |    |    |    |   |   |   |   |   |      |   |   |       |   |
| 1    | TO_LAST       | TO_EN      | CUR_TOCNT    |           |          |               |          |          |         |        |                      |            | ER_EFCI      | Reserved  |           |               |    | ABR_CTRL |        |    |    |    |   |   |   |   |   |      |   |   |       |   |
| 2    | PDU_FLAGS     |            |              |           |          | Reserved      |          |          |         |        | TOT_PDU_LEN          |            |              |           |           |               |    |          |        |    |    |    |   |   |   |   |   |      |   |   |       |   |
| 3    | CCOUNT        |            |              |           |          |               |          |          |         |        | TCOUNT               |            |              |           |           |               |    |          |        |    |    |    |   |   |   |   |   |      |   |   |       |   |
| 4    | CBUFF_LEN     |            |              |           |          |               |          |          |         |        | FW                   | STAT       |              |           |           | BFR1          |    |          | BFR0   |    |    |    |   |   |   |   |   |      |   |   |       |   |
| 5    | CBUFF_PNTR    |            |              |           |          |               |          |          |         |        |                      |            |              |           |           |               |    |          |        |    |    |    |   |   |   |   |   |      |   |   |       |   |
| 6    | BOM_BD_PNTR   |            |              |           |          |               |          |          |         |        |                      |            |              |           |           |               |    |          |        |    |    |    |   |   |   |   |   | Rsvd |   |   |       |   |
| 7    | CURR_BD_PNTR  |            |              |           |          |               |          |          |         |        |                      |            |              |           |           |               |    |          |        |    |    |    |   |   |   |   |   | Rsvd |   |   | CBFR1 |   |
| 8    | SEG_VCC_INDEX |            |              |           |          |               |          |          |         |        | SERV_DIS             |            |              |           |           |               |    |          |        |    |    |    |   |   |   |   |   |      |   |   |       |   |
| 9    | Reserved      |            |              |           |          | ERS_INDEX     |          |          |         |        | RX_COUNTER/VPC_INDEX |            |              |           |           |               |    |          |        |    |    |    |   |   |   |   |   |      |   |   |       |   |
| 10   | D_EN_NCR      | ACR_NOT_ER | D_NCR_TRIG   | D_NCR_DIR | Reserved |               |          | CONG_ID  |         |        | EN_EXP_CNG           | EN_IMP_CNG | EXP_TA_CI    | EXP_TA_NI | EXP_TA_ER |               |    |          |        |    |    |    |   |   |   |   |   |      |   |   |       |   |
| 11   | Rsvd          |            | D_NCR_HI_EXP |           |          | D_NCR_HI_MANT |          |          |         |        | Rsvd                 |            | D_NCR_LO_EXP |           |           | D_NCR_LO_MANT |    |          |        |    |    |    |   |   |   |   |   |      |   |   |       |   |

Table 5-12. Reassembly VCC Table Entry Format—AAL3/4

| 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 | PASS_OAM | Reserved | FF_DSC  | TO_INDEX | PM_INDEX |             |      |      | AAL_EN   |                      |      |    |    |    |    |   |   |   |   |   |   |   |   |      |       |
| 1    | TO_LAST       | TO_EN    | CUR_TOCNT |    |    |    |          |          |          |         | ER_EFCI  | Reserved |             |      |      | ABR_CTRL |                      |      |    |    |    |    |   |   |   |   |   |   |   |   |      |       |
| 2    | PDU_FLAGS     |          |           |    |    |    |          |          | Reserved |         |          |          | TOT_PDU_LEN |      |      |          |                      |      |    |    |    |    |   |   |   |   |   |   |   |   |      |       |
| 3    | BASIZE        |          |           |    |    |    |          |          | Rsvd     | NEXT_ST | NEXT_SN  |          |             |      | BTAG |          |                      |      |    |    |    |    |   |   |   |   |   |   |   |   |      |       |
| 4    | Reserved      |          |           |    |    |    |          |          | Rsvd     | STAT    |          |          |             | BFR1 |      |          |                      | BFR0 |    |    |    |    |   |   |   |   |   |   |   |   |      |       |
| 5    | CBUFF_PNTR    |          |           |    |    |    |          |          |          |         |          |          |             |      |      |          |                      |      |    |    |    |    |   |   |   |   |   |   |   |   |      |       |
| 6    | BOM_BD_PNTR   |          |           |    |    |    |          |          |          |         |          |          |             |      |      |          |                      |      |    |    |    |    |   |   |   |   |   |   |   |   | Rsvd |       |
| 7    | CURR_BD_PNTR  |          |           |    |    |    |          |          |          |         |          |          |             |      |      |          |                      |      |    |    |    |    |   |   |   |   |   |   |   |   | Rsvd | CBFR1 |
| 8    | SEG_VCC_INDEX |          |           |    |    |    |          |          |          |         |          |          |             |      |      |          | SERV_DIS             |      |    |    |    |    |   |   |   |   |   |   |   |   |      |       |
| 9    | Reserved      |          |           |    |    |    |          |          |          |         |          |          |             |      |      |          | RX_COUNTER/VPC_INDEX |      |    |    |    |    |   |   |   |   |   |   |   |   |      |       |
| 10   | Reserved      |          |           |    |    |    |          |          |          |         |          |          |             |      |      |          | Reserved             |      |    |    |    |    |   |   |   |   |   |   |   |   |      |       |
| 11   | Reserved      |          |           |    |    |    |          |          |          |         |          |          |             |      |      |          | Reserved             |      |    |    |    |    |   |   |   |   |   |   |   |   |      |       |



Table 5-13 details the descriptions of the reassembly VCC table fields.

Table 5-13. Reassembly VCC Table Descriptions (1 of 3)

| 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 as listed below; otherwise, when VC_EN is a logic low, a value of 11 causes the CELL_DSC_CNT counter not to be incremented.<br>00 = AAL5<br>01 = AAL0<br>10 = AAL3/4<br>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 LECDID table that holds channel LECDID values.   |        |       |        |       |          |          |        |         |   |   |       |         |        |       |        |       |          |          |        |         |
| PASS_OAM   | When bit is set OAM cells are processed and data cells are discarded without incrementing any discard counters.  |        |       |        |       |          |          |        |         |   |   |       |         |        |       |        |       |          |          |        |         |
| AALx_EN    | When set, enables the AALx function.   |        |       |        |       |          |          |        |         |   |   |       |         |        |       |        |       |          |          |        |         |
| FF_DSC     | FIFO buffer Full Discard. When a logic high, cells are discarded when the incoming DMA FIFO buffer is almost full. This includes OAM cells when RSM_CTRL1(OAM_QU_EN) is a logic low.   |        |       |        |       |          |          |        |         |   |   |       |         |        |       |        |       |          |          |        |         |
| TO_INDEX   | Selects one of eight INIT_TOCNT values in internal SRAM.   |        |       |        |       |          |          |        |         |   |   |       |         |        |       |        |       |          |          |        |         |
| PM_INDEX   | Pointer to a PM OAM processing word. Index with reference to top of VPI Index table.   |        |       |        |       |          |          |        |         |   |   |       |         |        |       |        |       |          |          |        |         |
| AAL_EN     | <p>Enable various cell processes. The AAL_EN field contains the following control bits:</p> <table border="1" style="margin-left: 40px;"> <tr> <td style="text-align: center;">9</td> <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;">PM_EN</td> <td style="text-align: center;">FIFO_EN</td> <td style="text-align: center;">LNK_EN</td> <td style="text-align: center;">FW_EN</td> <td style="text-align: center;">M52_EN</td> <td style="text-align: center;">BINTR</td> <td style="text-align: center;">STM_MODE</td> <td style="text-align: center;">LECID_EN</td> <td style="text-align: center;">FRD_EN</td> <td style="text-align: center;">CLPD_EN</td> </tr> </table> <p>PM_EN = Enable PM OAM processing on this channel.<br/>           FIFO_EN = Enable Logical FIFO Buffer mode.<br/>           LNK_EN = Enable writing of buffer descriptor NEXT field.<br/>           FW_EN = Enable firewall processing. Used in conjunction with FWALL_EN bit in RSM_CTRL0.<br/>           M52_EN = If set high, all 52 octets of the cell are written to a cell buffer.<br/>           BINTR = Enable interrupt after BOM buffer filled in Message Mode.<br/>           STM_MODE = Enable Streaming Mode.<br/>           LECDID_EN = Enable LANE-LECID echo suppression. Invalid in AAL3/4.<br/>           FRD_EN = Enable frame relay DE (Discard Eligibility) mode. Invalid in AAL3/4.<br/>           CLPD_EN = Enable CLP discard mode. Invalid in AAL3/4.</p> | 9      | 8     | 7      | 6     | 5        | 4        | 3      | 2       | 1 | 0 | PM_EN | FIFO_EN | LNK_EN | FW_EN | M52_EN | BINTR | STM_MODE | LECID_EN | FRD_EN | CLPD_EN |
| 9          | 8  | 7      | 6     | 5      | 4     | 3        | 2        | 1      | 0       |   |   |       |         |        |       |        |       |          |          |        |         |
| PM_EN      | FIFO_EN  | LNK_EN | FW_EN | M52_EN | BINTR | STM_MODE | LECID_EN | FRD_EN | CLPD_EN |   |   |       |         |        |       |        |       |          |          |        |         |
| 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.  |        |       |        |       |          |          |        |         |   |   |       |         |        |       |        |       |          |          |        |         |
| ER_EFCI    | ER EFCI bit of the previous data cell.   |        |       |        |       |          |          |        |         |   |   |       |         |        |       |        |       |          |          |        |         |

Table 5-13. Reassembly VCC Table Descriptions (2 of 3)

| Field Name  | Description/Function  |         |          |          |           |          |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
|-------------|---|---------|----------|----------|-----------|----------|---------|---------|----------|----------|---------|----------|----------|----------|----------|-----|-----------|----------|---------|---------|----------|
| ABR_CTRL    | <p>Set various control bits related to QFC. The QFC_CTRL field contains the following control bits:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <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;">ER_EN</td> <td style="text-align: center;">Reserved</td> <td style="text-align: center;">ABR_VPC</td> <td style="text-align: center;">Reserved</td> <td style="text-align: center;">Reserved</td> <td style="text-align: center;">Reserved</td> <td style="text-align: center;">Reserved</td> </tr> </table> <p>ER_EN = Enable ER operation.<br/>           ABR_VPC = Indicates ER connection is VPC (0 indicates VCC). For VPC operation, all VCC entries in the VPC group except VCI=6, must be initialized with VPC_INDEX pointing to the VCC entry corresponding with VCI=6. The VCI=6 entry contains the integrated EFCI bit over the VPC group. Also, all entries in the VPC group including VCI=6 must set the ABR_VPC bit to a logic high.</p>   | 6       | 5        | 4        | 3         | 2        | 1       | 0       | ER_EN    | Reserved | ABR_VPC | Reserved | Reserved | Reserved | Reserved |     |           |          |         |         |          |
| 6           | 5   | 4       | 3        | 2        | 1         | 0        |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
| ER_EN       | Reserved  | ABR_VPC | Reserved | Reserved | Reserved  | Reserved |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
| PDU_FLAGS   | <p>Set various flags related to PDUs. The PDU_FLAGS field contains the following control bits:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">31</td> <td style="text-align: center;">30</td> <td style="text-align: center;">29</td> <td style="text-align: center;">28</td> <td style="text-align: center;">27</td> <td style="text-align: center;">26</td> <td style="text-align: center;">25</td> <td style="text-align: center;">24</td> <td style="text-align: center;">23</td> <td style="text-align: center;">22</td> </tr> <tr> <td style="text-align: center;">CNT_ROVR</td> <td style="text-align: center;">SFPD_PND</td> <td style="text-align: center;">EPD</td> <td style="text-align: center;">CI</td> <td style="text-align: center;">CLP</td> <td style="text-align: center;">BFR_LOCAL</td> <td style="text-align: center;">BD_LOCAL</td> <td style="text-align: center;">ACT_PDU</td> <td style="text-align: center;">BOM_BUF</td> <td style="text-align: center;">FFPD_PND</td> </tr> </table> <p>CNT_ROVR = Indication that SERVICE_CNT counters have rolled over. The next status entry indicates this condition.<br/>           SFPD_PND = Status Full Packet Discard Pending. Set when status queue is full, and CPCS must be discarded when status entries available.<br/>           EPD = Early Packet Discard Flag. Set when EPD occurs on a channel. Cleared when new packet starts and error condition cleared.<br/>           CI = Congestion Indication. PTI[1] header bit ORed across the CPCS-PDU.<br/>           CLP = Cell Loss Priority. CLP header bit ORed across the CPCS-PDU.<br/>           BFR_LOCAL = Buffer Local. If high, the current cell buffer is located in local memory; otherwise, host memory. SAR maintains this bit.<br/>           BD_LOCAL = Buffer Descriptor Local. If high, the buffer descriptors reside in local memory; otherwise, host memory. SAR maintains this bit.<br/>           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.<br/>           BOM_BUF = BOM buffer flag. Set high when filling the first buffer of a PDU.<br/>           FFPD_PND = DMA FIFO buffer Full Packet Discard Pending.</p> | 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 |
| 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 |          |         |          |          |          |          |     |           |          |         |         |          |
| TOT_PDU_LEN | Total PDU length in bytes.  |         |          |          |           |          |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
| CRCREM      | Cycle Redundancy Check Remainder. CRC32 remainder used in AAL5 only.  |         |          |          |           |          |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
| CBUFF_LEN   | Current buffer length. Unused space of the current buffer in bytes.   |         |          |          |           |          |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
| FW          | Firewall condition. Indicates that a firewall condition has occurred.   |         |          |          |           |          |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
| CCOUNT      | Termination Cell count. Used in AAL0 to terminate packet. When PTI termination mode is enabled, the maximum total length of a CPCS-PDU is CCOUNT × 2 bytes. In fixed length mode, initialize to 1.  |         |          |          |           |          |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
| TCOUNT      | Termination Count. Used in AAL0 to determine the number of cells in a packet. If this field is 0 in AAL0 mode, PTI termination mode is enabled.   |         |          |          |           |          |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
| BASIZE      | AAL3/4 BASIZE field. Used to record the BASIZE field from the AAL3/4 PDU, in order to check against the LENGTH field in the AAL3/4 PDU trailer.   |         |          |          |           |          |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
| NEXT_ST     | Next Segment Type expected.   |         |          |          |           |          |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
| NEXT_SN     | Next Sequence Number expected.  |         |          |          |           |          |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
| BTAG        | Records the AAL3/4 CPCS-PDU's Btag field, in order to check against the Etag field in the CPCS-PDU trailer.   |         |          |          |           |          |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |
| STAT        | Status queue pool number.   |         |          |          |           |          |         |         |          |          |         |          |          |          |          |     |           |          |         |         |          |

Table 5-13. Reassembly VCC Table Descriptions (3 of 3)

| Field Name               | Description/Function  |
|--------------------------|---|
| BFR1                     | COM free buffer queue pool number.  |
| BFR0                     | BOM free buffer queue pool number.  |
| CBUFF_PNTR               | Current buffer pointer. Pointer to the current unused position of the cell buffer where cell payload data may be written. In Logical FIFO Mode, the address of the FIFO.                        |
| BOM_BD_PNTR              | BOM buffer descriptor pointer. Pointer to the BOM buffer descriptor.  |
| CURR_BD_PNTR             | Current buffer descriptor pointer. Pointer to the buffer descriptor corresponding to the current buffer in use.   |
| CBFR1                    | Current BFR1 indication. A logic high indicates that current buffer is from BFR1 pool and logic low indicates from BFR0 pool.   |
| SEG_VCC_INDEX            | Channel index of corresponding segmentation channel. Used by ER and PM-OAM processing.  |
| SERV_DIS                 | Service Discard counter. Counts the number of CPCS-PDUs discarded due to either LANE_LECID echo suppression, CLP discard, or frame relay DE discard.  |
| ERS_INDEX                | Index into ER_SHIFT table for implicit congestion ER reduction. Base table address is given by ER_SHIFT_B register field.   |
| RX_COUNTER/<br>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. |
| D_EN_NCR                 | Enable destination ACR/ER change notification processing.   |
| ACR_NOT_ER               | Logic high results in CR change notification; logic low results in ER change notification.  |
| D_NCR_TRIG               | Destination ACR/ER change has been triggered. Initialize to logic low.  |
| D_NCR_DIR                | Indicates direction of destination ACR/ER trigger, logic high for HI, logic low for LO.   |
| CONG_ID                  | Congestion Identification number.   |
| EN_EXP_CNG               | Enables explicit congestion ER reduction mechanism.   |
| EN_IMP_CNG               | Enables implicit congestion ER reduction mechanism; not valid when EN_EXP_CNG is logic high.  |
| EXP_TA_CI                | Set CI to logic high in turned-around BCK RM cells.   |
| EXP_TA_NI                | Set NI to logic high in turned-around BCK RM cells.   |
| EXP_TA_ER                | Set ER to minimum of this value and in coming FWD RM ER value in turned-around BCK RM cells when EN_EXP_CNG is logic high.  |
| D_NCR_HI_EXP             | Destination ACR/ER upper threshold, exponent portion.   |
| D_NCR_HI_MANT            | Destination ACR/ER upper threshold, mantissa portion.   |
| D_NCR_LO_EXP             | Destination ACR/ER lower threshold, exponent portion.   |
| D_NCR_LO_MANT            | Destination ACR/ER lower threshold, mantissa portion.   |

5.7.2.2 AAL3/4 Head VCC Table Entry [Table 5-14](#) shows the AAL3/4 Head VCC table structure. [Table 5-15](#) details the AAL3/4 Head VCC table field definitions.

**Table 5-14. AAL3/4 Head VCC 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    | VC_EN         | AAL_TYPE     | Reserved   |           |               |    |    |    |          |           |                | FF_DSC       | Reserved   |            |               | PM_INDEX             |           |          |    | AAL_EN |    |    |          |   |   |   |   |   |   |   |   |   |
| 1    | TO_LAST       | TO_EN        | Reserved   | MIDO      | MID_BITS      |    |    |    | CRC10_EN | LI_EN     | ST_EN          | SN_EN        | CPI_EN     | BAT_EN     | BAH_EN        | Reserved             | ER_EFCI   | Reserved |    |        |    |    | ABR_CTRL |   |   |   |   |   |   |   |   |   |
| 2    | Reserved      |              |            |           |               |    |    |    |          |           | VCC_INDEX_BASE |              |            |            |               |                      |           |          |    |        |    |    |          |   |   |   |   |   |   |   |   |   |
| 3    | Reserved      |              |            |           |               |    |    |    |          |           |                |              |            |            |               |                      |           |          |    |        |    |    |          |   |   |   |   |   |   |   |   |   |
| 4    | Reserved      |              |            |           |               |    |    |    |          |           |                |              |            |            |               | Rsvd                 | STAT      |          |    | BFR1   |    |    | BFR0     |   |   |   |   |   |   |   |   |   |
| 5    | CRC10_ERR     |              |            |           |               |    |    |    |          |           |                |              |            |            |               | MID_ERR              |           |          |    |        |    |    |          |   |   |   |   |   |   |   |   |   |
| 6    | LI_ERR        |              |            |           |               |    |    |    |          |           |                |              |            |            |               | SN_ERR               |           |          |    |        |    |    |          |   |   |   |   |   |   |   |   |   |
| 7    | BOM_SSM_ERR   |              |            |           |               |    |    |    |          |           |                |              |            |            |               | EOM_ERR              |           |          |    |        |    |    |          |   |   |   |   |   |   |   |   |   |
| 8    | SEG_VCC_INDEX |              |            |           |               |    |    |    |          |           |                |              |            |            |               | Reserved             |           |          |    |        |    |    |          |   |   |   |   |   |   |   |   |   |
| 9    | Reserved      |              |            |           |               |    |    |    |          | ERS_INDEX |                |              |            |            |               | RX_COUNTER/VPC_INDEX |           |          |    |        |    |    |          |   |   |   |   |   |   |   |   |   |
| 10   | D_EN_NCR      | ACR_NOT_ER   | D_NCR_TRIG | D_NCR_DIR | Reserved      |    |    |    | CONG_ID  |           |                |              | EN_EXP_CNG | EN_IMP_CNG | EXP_TA_CI     | EXP_TA_NI            | EXP_TA_ER |          |    |        |    |    |          |   |   |   |   |   |   |   |   |   |
| 11   | Rsvd          | D_NCR_HI_EXP |            |           | D_NCR_HI_MANT |    |    |    |          |           | Rsvd           | D_NCR_LO_EXP |            |            | D_NCR_LO_MANT |                      |           |          |    |        |    |    |          |   |   |   |   |   |   |   |   |   |

**Table 5-15. AAL3/4 Head VCC Table Descriptions (1 of 3)**

| 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 AAL as listed below; otherwise, when VC_EN is a logic low, a value of 11 causes the CELL_DSC_CNT counter not to be incremented.<br>00 = AAL5<br>01 = AAL0<br>10 = AAL3/4<br>11 = Reserved |
| FF_DSC     | FIFO buffer Full Discard. When a logic high and RSM_CTRL1(OAM_QU_EN) is a logic low, OAM cells are discarded when the incoming DMA FIFO buffer is almost full.  |
| PM_INDEX   | Pointer to a PM-OAM processing word. Index with reference to top of VPI Index table. Used by the OAM cells detected on AAL3/4 channels.   |
| AAL_EN     | Same as AAL_EN field in standard RSM VCC entry. Used by OAM cells detected on AAL3/4 channels.  |
| TO_LAST    | Indicates the last entry in the VCC table in which time-out processing occurs.  |
| TO_EN      | Enable time-out process of the VCC entry. Must be set to a logic 0.   |
| MIDO       | When logic high, allow a MID value of 0; otherwise, 0 is invalid.   |

Table 5-15. AAL3/4 Head VCC Table Descriptions (2 of 3)

| Field Name     | Description/Function  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
|----------------|---|----------|---------------------|------|-------------------------|------|---------|------|---------|------|---------|------|----------|------|----------|------|----------|------|-----------|------|-----------|------|-----------|------|------------|------|----------|------|----------|------|----------|------|----------|------|----------|
| MID_BITS       | <p>Number of significant bits for MID field. Defines the limit of MID values allowed for the AAL3/4 Virtual Connection. If the MID_BITS subfield is non-0 and the AAL_TYPE is AAL3/4, CPCS_MID multiplexing processing is enabled. A MID of 0 is valid only if MIDO is a logic high. MID_BITS is decoded as follows:</p> <table border="1"> <thead> <tr> <th>MID_BITS</th> <th>Range of MID values</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>MID processing disabled</td> </tr> <tr> <td>0001</td> <td>0 / 1-1</td> </tr> <tr> <td>0010</td> <td>0 / 1-3</td> </tr> <tr> <td>0011</td> <td>0 / 1-7</td> </tr> <tr> <td>0100</td> <td>0 / 1-15</td> </tr> <tr> <td>0101</td> <td>0 / 1-31</td> </tr> <tr> <td>0110</td> <td>0 / 1-63</td> </tr> <tr> <td>0111</td> <td>0 / 1-127</td> </tr> <tr> <td>1000</td> <td>0 / 1-255</td> </tr> <tr> <td>1001</td> <td>0 / 1-511</td> </tr> <tr> <td>1010</td> <td>0 / 1-1023</td> </tr> <tr> <td>1011</td> <td>Reserved</td> </tr> <tr> <td>1100</td> <td>Reserved</td> </tr> <tr> <td>1101</td> <td>Reserved</td> </tr> <tr> <td>1110</td> <td>Reserved</td> </tr> <tr> <td>1111</td> <td>Reserved</td> </tr> </tbody> </table> | MID_BITS | Range of MID values | 0000 | MID processing disabled | 0001 | 0 / 1-1 | 0010 | 0 / 1-3 | 0011 | 0 / 1-7 | 0100 | 0 / 1-15 | 0101 | 0 / 1-31 | 0110 | 0 / 1-63 | 0111 | 0 / 1-127 | 1000 | 0 / 1-255 | 1001 | 0 / 1-511 | 1010 | 0 / 1-1023 | 1011 | Reserved | 1100 | Reserved | 1101 | Reserved | 1110 | Reserved | 1111 | Reserved |
| MID_BITS       | Range of MID values   |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 0000           | MID processing disabled   |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 0001           | 0 / 1-1   |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 0010           | 0 / 1-3   |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 0011           | 0 / 1-7   |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 0100           | 0 / 1-15  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 0101           | 0 / 1-31  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 0110           | 0 / 1-63  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 0111           | 0 / 1-127   |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 1000           | 0 / 1-255   |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 1001           | 0 / 1-511   |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 1010           | 0 / 1-1023  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 1011           | Reserved  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 1100           | Reserved  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 1101           | Reserved  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 1110           | Reserved  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| 1111           | Reserved  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| CRC10_EN       | If set high, enable AAL3/4 crc10 field checking and error counting.   |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| LI_EN          | If set high, enable AAL3/4 LI field checking and error counting.  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| ST_EN          | If set high, enable AAL3/4 ST field checking and error counting.  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| SN_EN          | If set high, enable AAL3/4 SN field checking and error counting.  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| CPI_EN         | If set high, enable AAL3/4 CPI field checking.  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| BAT_EN         | If set high, enable AAL3/4 BASIZE $\neq$ Length checking. If low, LENGTH > BASIZE check is performed.   |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| BAH_EN         | If set high, enable AAL3/4 BASIZE < 37 checking.  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| ER_EFCI        | ER EFCI bit of the previous data cell.  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| VCC_INDEX_BASE | Pointer to the first VCC table entry for this VCC with MID value = 0. All multiplexed MIDs on this channel are contained in a contiguous block of VCC table entries.  |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |
| STAT           | Status queue pool number.   |          |                     |      |                         |      |         |      |         |      |         |      |          |      |          |      |          |      |           |      |           |      |           |      |            |      |          |      |          |      |          |      |          |      |          |

Table 5-15. AAL3/4 Head VCC Table Descriptions (3 of 3)

| Field Name               | Description/Function  |
|--------------------------|---|
| BFR1                     | COM free buffer queue pool number.  |
| BFR0                     | BOM free buffer queue pool number.  |
| CRC10_ERR                | Number of AAL3/4 cells received with CRC10 error.   |
| MID_ERR                  | Number of AAL3/4 cells received that did not have an active MID as determined by the MID_BITS and MID0 fields.  |
| LI_ERR                   | Number of AAL3/4 cells received that had an unexpected LI value.  |
| SN_ERR                   | Number of AAL3/4 cells received that had an unexpected SN value.  |
| BOM_SSM_ERR              | Number of unexpected AAL3/4 BOM or SSM cells received.  |
| EOM_ERR                  | Number of unexpected AAL3/4 EOM cells received.   |
| SEG_VCC_INDEX            | Channel index of corresponding segmentation channel. Used by ER and PM-OAM processing.  |
| ERS_INDEX                | Index into ER_SHIFT table for implicit congestion ER reduction. Base table address is given by ER_SHIFT_B register field.   |
| RX_COUNTER/<br>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. |
| D_EN_NCR                 | Enable destination ACR/ER change notification processing.   |
| ACR_NOT_ER               | Logic high results in CR change notification; logic low results in ER change notification.  |
| D_NCR_TRIG               | Destination ACR/ER change has been triggered. Initialize to logic low.  |
| D_NCR_DIR                | Indicates direction of destination ACR/ER trigger, logic high for HI, logic low for LO.   |
| CONG_ID                  | Congestion Identification number.   |
| EN_EXP_CNG               | Enables explicit congestion ER reduction mechanism.   |
| EN_IMP_CNG               | Enables implicit congestion ER reduction mechanism; not valid when EN_EXP_CNG is logic high.  |
| EXP_TA_CI                | Set CI to logic high in turned-around BCK RM cells.   |
| EXP_TA_NI                | Set NI to logic high in turned-around BCK RM cells.   |
| EXP_TA_ER                | Set ER to minimum of this value and in coming FWD RM ER value in turned-around BCK RM cells when EN_EXP_CNG is logic high.  |
| D_NCR_HI_EXP             | Destination ACR/ER upper threshold, exponent portion.   |
| D_NCR_HI_MANT            | Destination ACR/ER upper threshold, mantissa portion.   |
| D_NCR_LO_EXP             | Destination ACR/ER lower threshold, exponent portion.   |
| D_NCR_LO_MANT            | Destination ACR/ER lower threshold, mantissa portion.   |

### 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 5-16](#) and [Table 5-17](#) describe the format of the reassembly buffer descriptors.

**Table 5-16. 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 5-17. 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.<br><i>NOTE(S):</i> The SAR does not access this word; 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 CN8236 internal memory.

[Table 5-18](#) and [Table 5-19](#) describe the format of the free buffer queue base table entries.

**Table 5-18. 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       |    |    |    |    |    |    |    | Reserved | BD_LOCAL | Rsvd | FORWARD |    |    |    |    |      |           |      |      |    |    |   |   |   |   |   |   |   |   |   |   |
| 2    | READ_UD_PNTR |    |    |    |    |    |    |    |          |          |      |         |    |    |    |    | Rsvd |           |      |      |    |    |   |   |   |   |   |   |   |   |   |   |
| 3    | Reserved     |    |    |    |    |    |    |    |          |          |      |         |    |    |    |    |      |           |      |      |    |    |   |   |   |   |   |   |   |   |   |   |

Table 5-19. 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.   |
| 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.       |
| BD_LOCAL     | If logic high, buffer descriptor and READ_UD word are located in SAR-shared memory, otherwise in host memory.                     |
| FORWARD      | Current Forward processing Read index pointer (firewalling).  |
| READ_UD_PNTR | Word address in user memory to write READ pointer every UPDATE interval.  |

Table 5-20 and Table 5-21 describe the format of the free buffer queue entries.

Table 5-20. 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     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |         |           |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 2 <sup>(1)</sup>  | Reserved    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved | FWD_VLD | VCC_INDEX |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 3 <sup>(1)</sup>  | (not used)  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |         |           |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>NOTE(S):</b>   |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |         |           |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <sup>(1)</sup> These words are used only in Bank 0 if RSM_FBOCTL(FBQ0_RTN) = 1. |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |         |           |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

Table 5-21. 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.            |
| Reserved    | Always set to 0.   |
| FWD_VLD     | Forward Valid. If logic high, word contains valid buffer return information. |
| VCC_INDEX   | Channel of corresponding buffer return.                                      |



### 5.7.5 Reassembly Status Queues

The CN8236 reports reassembly status to the host on any one of 32 reassembly status queues.

At initialization, the host assigns the location and size of up to 32 RSM status queues by initializing the reassembly status queue base table entries in CN8236 internal memory. The location and size of each RSM status queue is independently programmable via these base table entries.

Table 5-22 and Table 5-23 describe the format of the reassembly status queue base table entries.

Table 5-24 through Table 5-31 describe the formats of the reassembly status queue entries.

Table 5-22. 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 |    |    |    |    |    | Reserved |    |    |    |    |    | READ_UD |    |      |       |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

Table 5-23. 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.<br>00 = 64<br>01 = 256<br>10 = 1,024<br>11 = 4,096           |
| WRITE      | Current Write index pointer maintained by the SAR.                                     |
| READ_UD    | Periodic Read update index pointer maintained by the user.                             |

Table 5-24. Reassembly Status Queue Entry Format with FWD\_PM = 0

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

Table 5-25. Reassembly Status Queue Entry Format with FWD\_PM = 1

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

NOTE(S): OVFL and CNT\_ROVR are defined in Table 5-31 under "Status."

Table 5-26. Reassembly Status Queue Entry Format with FWD\_PM = 0 and AAL34 = 1

| Word | 31             | 30         | 29 | 28 | 27 | 26 | 25 | 24    | 23       | 22 | 21          | 20        | 19 | 18 | 17     | 16 | 15 | 14 | 13     | 12 | 11  | 10 | 9 | 8   | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|----------------|------------|----|----|----|----|----|-------|----------|----|-------------|-----------|----|----|--------|----|----|----|--------|----|-----|----|---|-----|---|---|---|---|---|---|---|---|
| 0    | BD_PNTR        |            |    |    |    |    |    |       |          |    |             |           |    |    |        |    | 00 |    |        |    |     |    |   |     |   |   |   |   |   |   |   |   |
| 1    | HEAD_VCC_INDEX |            |    |    |    |    |    |       |          |    | CPCS_LENGTH |           |    |    |        |    |    |    |        |    |     |    |   |     |   |   |   |   |   |   |   |   |
| 2    | Rsvd           | PDU_CHECKS |    |    |    |    |    |       |          |    |             | VCC_INDEX |    |    |        |    |    |    |        |    |     |    |   |     |   |   |   |   |   |   |   |   |
| 3    | VLD            | Reserved   |    |    |    |    |    | AAL34 | Reserved |    |             |           |    |    | STATUS |    |    |    | FWD_PM |    | OAM |    |   | STM |   |   |   |   |   |   |   |   |

Table 5-27. PDU\_CHECKS Field Bits

| Bit | 29        | 28       | 27       | 26        | 25        | 24        | 23        | 22        | 21       | 20       | 19       | 18 | 17      | 16 |
|-----|-----------|----------|----------|-----------|-----------|-----------|-----------|-----------|----------|----------|----------|----|---------|----|
|     | MOD_ERROR | BA_ERROR | AL_ERROR | LEN_ERROR | PAD_ERROR | CPI_ERROR | TAG_ERROR | CRC_ERROR | ST_ERROR | SN_ERROR | LI_ERROR | LP | CI_LAST | CI |

Table 5-28. PDU\_CHECKS Field Bits with CNT\_ROVR = 1 and AAL34 = 1

| Bit | 29        | 28       | 27       | 26        | 25        | 24        | 23        | 22        | 21     | 20     | 19     | 18       | 17 | 16 |
|-----|-----------|----------|----------|-----------|-----------|-----------|-----------|-----------|--------|--------|--------|----------|----|----|
|     | MOD_ERROR | BA_ERROR | AL_ERROR | LEN_ERROR | PAD_ERROR | CPI_ERROR | TAG_ERROR | CRC_ERROR | ST_EPD | SN_EPD | LI_EPD | A3L2_ERR |    |    |

Table 5-29. STATUS Field Bits

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

Table 5-30. STM Field Bits

| Bit | 3   | 2   | 1        | 0    |
|-----|-----|-----|----------|------|
|     | EOM | BOM | STM_MODE | BFR1 |

Table 5-31. 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.   |
| TRCC0          | PM total received cell count for CLP = 0.  |
| TRCC0+1        | PM total received cell count for CLP = 0+1.  |
| HEAD_VCC_INDEX | AAL3/4 Head VCC table entry index. When CNT_ROVR = 1, AAL34 = 1, and FWD_PM = 0, this field along with A3L2_ERR in the PDU_CHECKS field points to the MID that rolled over. This field is also active when CNT_ROVR=0, AAL34=1, and FWD_PM=0.  |
| PDU_CHECKS     | Set various error flags related to PDUs.<br><br>MOD_ERROR= AAL3/4 CPCS-PDU is not 32-bit aligned.<br>BA_ERROR = AAL5: Indicates that the total PDU length exceeds maximum length when AUU = 1.<br>AAL3/4: BASIZE field error occurred.<br>AL_ERROR = AAL3/4 AL field is not all 0s.<br>LEN_ERROR = Total CPCS-PDU length exceeds maximum length and not at end of message.<br>PAD_ERROR= PAD field length is not correct.<br>CPI_ERROR = CPI field is not all 0.<br>TAG_ERROR = AAL3/4 Btag and Etag fields in the CPCS-PDU do not match.<br>CRC_ERROR= AAL5 CPCS or OAM CRC error.<br>ST_EPD = AAL3/4 Segment Type error; causes EPD.<br>SN_EPD = AAL3/4 Sequence Number error; causes EPD.<br>LI_EPD = AAL3/4 SAR-PDU length error; causes EPD.<br>LP = Value of CLP header bit ORed across the CPCS-PDU.<br>CI_LAST = Value of the PTI[1] header bit in the last cell of the CPCS-PDU.<br>CI = Value of the PTI[1] header bit ORed across the CPCS-PDU.<br>A3L2_ERR = When CNT_ROVR = 1, AAL34 = 1, and FWD_PM = 0, indicates which MIB counter rolled over, based on these values:<br>0 = MID_ERR<br>1 = CRC10_ERR<br>2 = SN_ERR<br>3 = LI_ERR<br>4 = EOM_ERR<br>5 = BOM_SSM_ERR |
| VCC_INDEX      | VCC index of channel. If the connection is AAL3/4 and a CRC10 or MID error has occurred, this index points to a valid MID entry even though MID cannot be correctly resolved due to errors. In both cases, CNT_ROVR is set and the index is only used to indicate an AAL3/4 connection.  |
| VLD            | Valid. Indication that status entry is valid. The host must set to 0 after processing status.  |
| AAL34          | Indication that the status entry is relative to an AAL3/4 PDU. Also indicates that HEAD_VCC_INDEX and A3L2_ERR fields are active.<br><b>NOTE(S):</b> This field is only active when FWD_PM = 0.  |

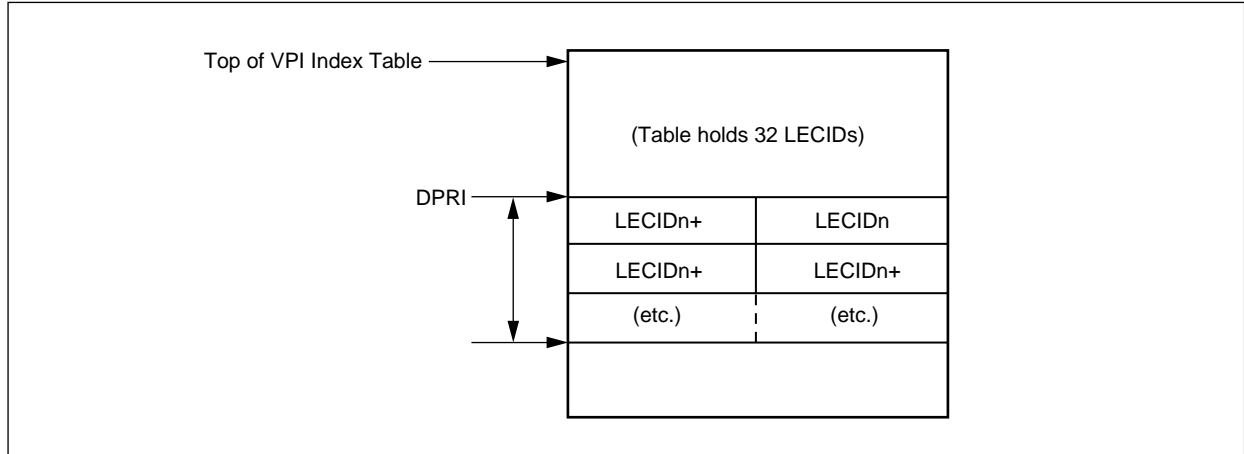
Table 5-31. Reassembly Status Queue Entry Descriptions (2 of 2)

| Field Name | Description/Function  |          |      |      |      |     |       |          |      |   |      |     |    |      |      |      |    |       |          |
|------------|---|----------|------|------|------|-----|-------|----------|------|---|------|-----|----|------|------|------|----|-------|----------|
| STATUS     | <p>Sets various status bits. The STATUS field contains the following control bits:</p> <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 0 10px;">16</td> <td style="text-align: center; padding: 0 10px;">15</td> <td style="text-align: center; padding: 0 10px;">14</td> <td style="text-align: center; padding: 0 10px;">13</td> <td style="text-align: center; padding: 0 10px;">12</td> <td style="text-align: center; padding: 0 10px;">11</td> <td style="text-align: center; padding: 0 10px;">10</td> <td style="text-align: center; padding: 0 10px;">9</td> <td style="text-align: center; padding: 0 10px;">8</td> </tr> <tr> <td style="text-align: center; border: 1px solid black;">FFPD</td> <td style="text-align: center; border: 1px solid black;">EPD</td> <td style="text-align: center; border: 1px solid black;">FW</td> <td style="text-align: center; border: 1px solid black;">UNDF</td> <td style="text-align: center; border: 1px solid black;">OVFL</td> <td style="text-align: center; border: 1px solid black;">SFPD</td> <td style="text-align: center; border: 1px solid black;">TO</td> <td style="text-align: center; border: 1px solid black;">ABORT</td> <td style="text-align: center; border: 1px solid black;">CNT_ROVR</td> </tr> </table> <p>FFPD = DMA FIFO buffer Full Packet Discard.<br/> 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.<br/> FW = Firewall error condition occurred. If early packet discard occurs, EPD is set.<br/> UNDF = Free Buffer Queue Underflow occurred. If early packet discard occurs, EPD is set.<br/> OVFL = Last available Status Queue entry.<br/> SFPD = Status Full Packet Discard occurred. EPD is not set.<br/> TO = Reassembly time-out condition occurred on this channel. EPD is not set.<br/> ABORT = Abort function detected. EPD not set.<br/> CNT_ROVR = Indication that the SERVICE_CNT counter has rolled over on the channel or an AAL3/4 MIB counter has rolled over in an AAL3/4 Head VCC entry. If AAL3/4 MIB, then the A3L2_ERR field is active in the PDU_CHECKS field and indicates which AAL3/4 MIB counter has rolled over.</p> | 16       | 15   | 14   | 13   | 12  | 11    | 10       | 9    | 8 | FFPD | EPD | FW | UNDF | OVFL | SFPD | TO | ABORT | CNT_ROVR |
| 16         | 15  | 14       | 13   | 12   | 11   | 10  | 9     | 8        |      |   |      |     |    |      |      |      |    |       |          |
| FFPD       | EPD   | FW       | UNDF | OVFL | SFPD | TO  | ABORT | CNT_ROVR |      |   |      |     |    |      |      |      |    |       |          |
| FWD_PM     | PM-OAM Forward Monitoring cell detected.  |          |      |      |      |     |       |          |      |   |      |     |    |      |      |      |    |       |          |
| OAM        | <p>A non-0 field indicates that an OAM or management cell has been received as follows:</p> <p>001 = F4 OAM<br/> 010 = F4 OAM End to End<br/> 100 = F5 OAM<br/> 101 = F5 OAM End to End<br/> 110 = PTI = 6<br/> 111 = PTI = 7</p>   |          |      |      |      |     |       |          |      |   |      |     |    |      |      |      |    |       |          |
| STM        | <p>Sets various bits related to Streaming Mode. The STM field contains the following control bits:</p> <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 0 10px;">3</td> <td style="text-align: center; padding: 0 10px;">2</td> <td style="text-align: center; padding: 0 10px;">1</td> <td style="text-align: center; padding: 0 10px;">0</td> </tr> <tr> <td style="text-align: center; border: 1px solid black;">EOM</td> <td style="text-align: center; border: 1px solid black;">BOM</td> <td style="text-align: center; border: 1px solid black;">STM_MODE</td> <td style="text-align: center; border: 1px solid black;">BFR1</td> </tr> </table> <p>EOM = In Streaming Mode or BOM Interrupt mode, this bit identifies that the buffer contains an EOM.<br/> 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.<br/> STM_MODE = Indication that Streaming Mode is enabled on the channel.<br/> 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 |      |      |     |       |          |      |   |      |     |    |      |      |      |    |       |          |
| BIPV       | BIP 16 violations. Same as BLER0+1 field in Backward Reporting PM-OAM cell.   |          |      |      |      |     |       |          |      |   |      |     |    |      |      |      |    |       |          |

### 5.7.6 LECID Table

The LECID table illustrated in [Figure 5-19](#) 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 5-32](#) and [Table 5-33](#) display the LECID table entries and field definitions.

Figure 5-19. LECID Table, Illustrated



8236\_034

Table 5-32. 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 5-33. LECID Table Field Definition

| Field Name | Function/Description  |
|------------|---|
| LECIDn     | <p>LAN Emulation Client Identifier: a unique identifier for a LAN Emulation Client (LEC). The LECID table is capable of storing 32 LECIDs.</p> <p>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.</p> |

### 5.7.7 Global Time-Out Table

This table exists in internal SRAM starting at address 0x1340. The values in this table should be initialized during system initialization, before reassembly processing is started. These values set the selectable hardware PDU time-out values as described in [Section 5.4.9](#). [Tables 5-34](#) and [5-35](#) display the entries and field definitions for the Global Time-Out table.

**Table 5-34. 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 5-35. Global Time-Out Table Entry Descriptions**

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

### 5.7.8 Reassembly Internal SRAM Memory Map

As indicated in [Table 5-36](#), the Reassembly Internal SRAM is in the address range 0x1000-0x13FF.

**Table 5-36. Reassembly Internal SRAM Memory Map**

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





## 6.0 Traffic Management

---

### 6.1 Overview

The traffic management capabilities of ATM differentiate it from other communication technologies. The CN8236 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 CBR, Real-Time and Non-Real-Time Variable Bit Rate (rt-VBR and nrt-VBR), UBR, GBR, and ABR.

[Table 6-1](#) provides a list of ATM attributes detailed in the *TM 4.1 Specification* (that is, traffic parameters, QoS parameters, and feedback characteristics), and identifies whether the CN8236 supports these for each service category. The shaded areas do not indicate 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

| Attribute  | ATM Layer Service Category |                       |                              |                              |                          |                              |
|--|----------------------------|-----------------------|------------------------------|------------------------------|--------------------------|------------------------------|
|  | CBR                        | rt-VBR <sup>(1)</sup> | nrt-VBR <sup>(2)</sup>       | UBR <sup>(3)</sup>           | ABR                      | GFR                          |
| <b>TRAFFIC PARAMETERS:</b>   |                            |                       |                              |                              |                          |                              |
| 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                |                              |
| <b>QOS PARAMETERS:</b>   |                            |                       |                              |                              |                          |                              |
| 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><br>(1) The rt-VBR service category is intended for real-time applications; that is, those requiring tightly constrained delay and delay variation, as would be appropriate for voice and video applications.<br>(2) The nrt-VBR service category is intended for non-real-time applications which have bursty traffic characteristics.<br>(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.<br>(4) This is a network parameter. The CN8236 provides counters that monitor this parameter.<br>(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 CN8236 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 CN8236 supports the rate based flow control and service models specified for ABR in *TM 4.1*.

The CN8236 also provides Generic Flow Control. This XON/XOFF protocol complements the GFC algorithm by allowing switches to significantly overallocate port bandwidth.

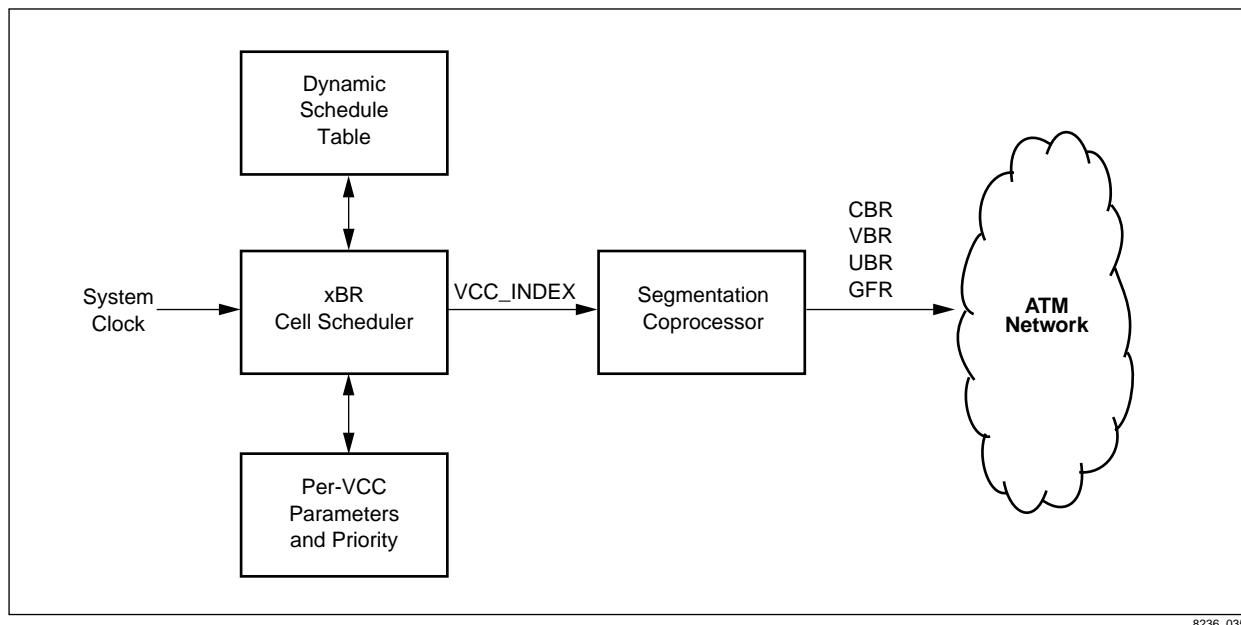
### 6.1.1 xBR Cell Scheduler

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 CN8236 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 CN8236 can manage VCC traffic on either a VC or a VP level. In addition, it can schedule traffic as a CBR tunnel (or pipe), that is, 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 CN8236 supports 16 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



8236\_035

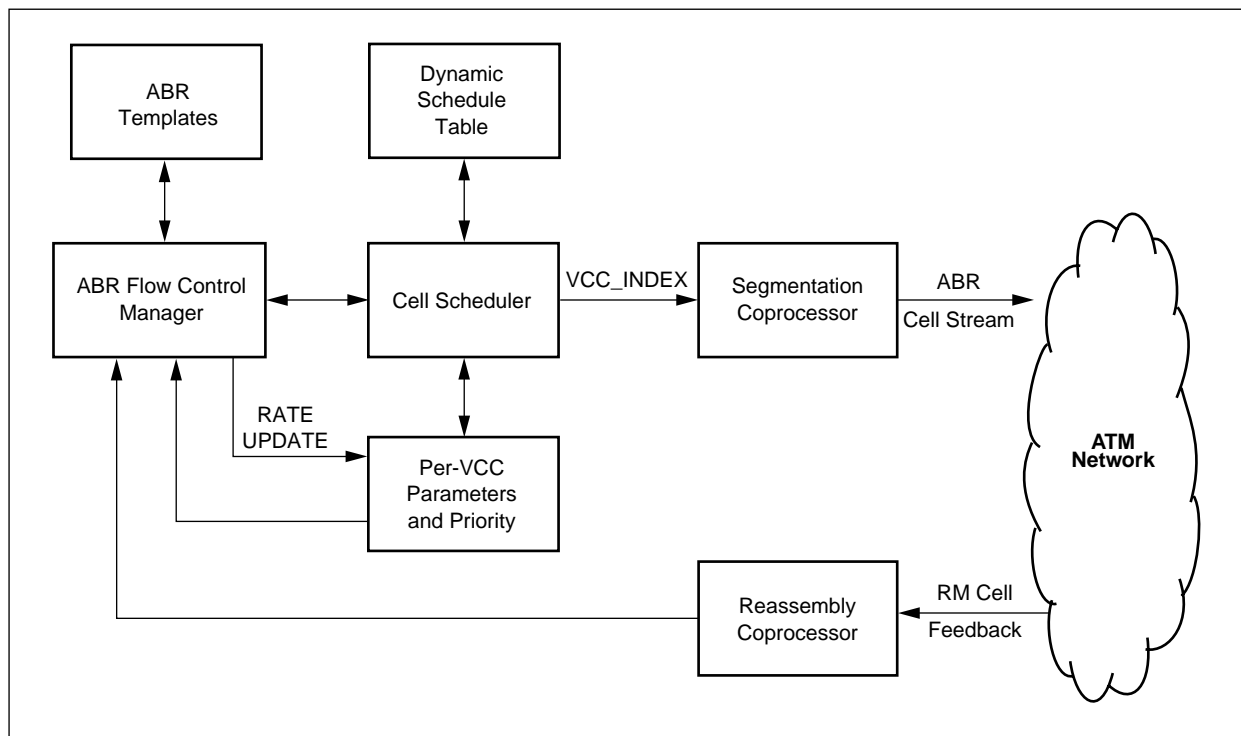
### 6.1.2 ABR Flow Control Manager

The CN8236 implements the ATM Forum ABR flow control algorithms, referred to in the aggregate as ABR by this document. The ABR service category effectively allows 0 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 CN8236'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



8236\_036

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 may have different flow control parameters. For instance, a system designer may wish to configure a VCC traversing a network with all 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 16 Priority Levels + CBR** The CN8236 supports 16 scheduling priorities (the PRI field in the SEG VCC table entry), in addition to the optional CBR service category, with 15 being the highest priority, down to 0 being the lowest priority. CBR channels are assigned a priority which is in effect higher than the 16 scheduling priorities discussed here. From one to 16 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 16 scheduling priorities, and thus accessible by an offset pointer. The host sets the offset of the VBR/ABR priorities in the VBR\_OFFSET field of SEG\_CTRL or SCH\_CTRL.
- 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 clocks cycles, set as the number of clock cycles 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](#). 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.

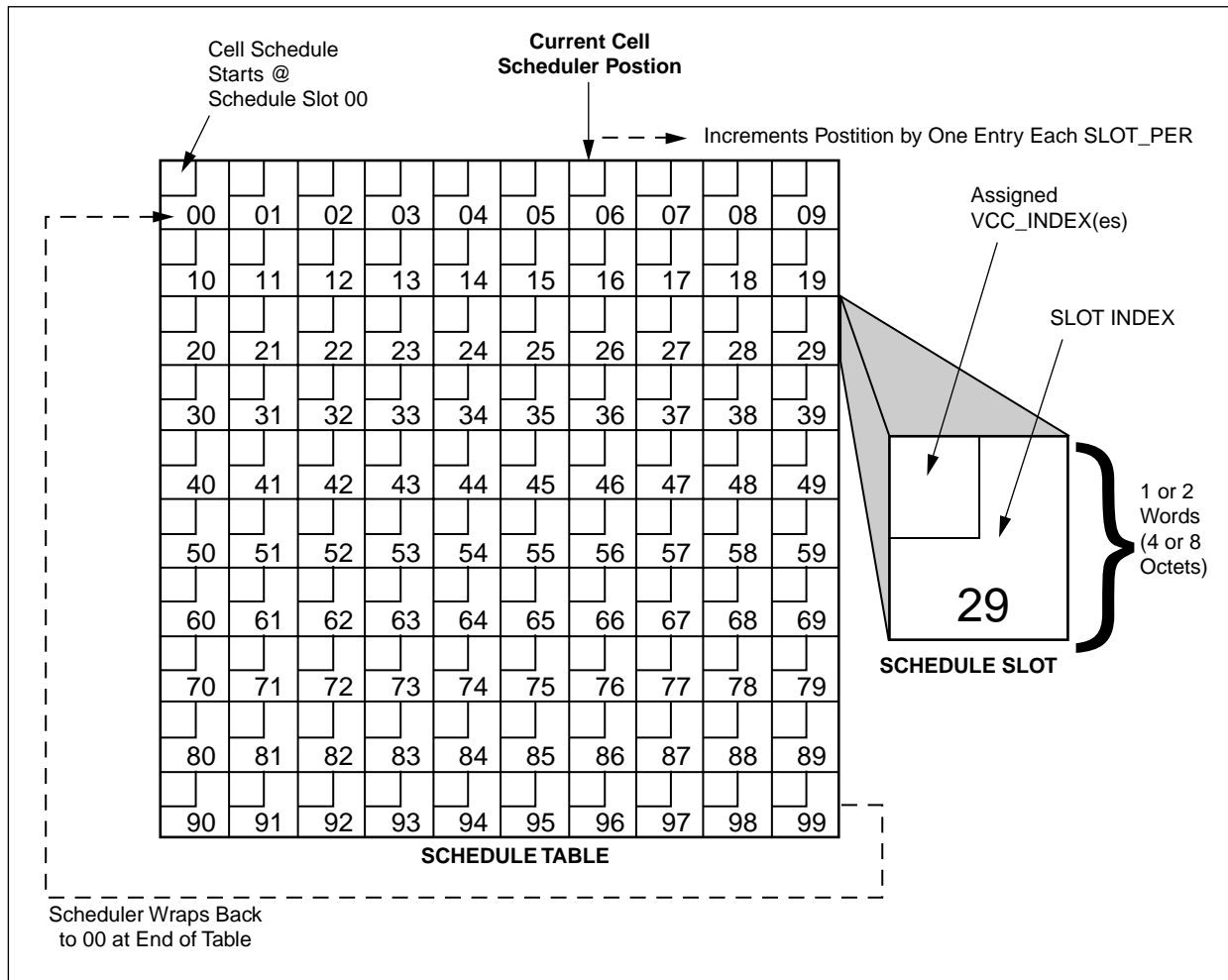
The scheduler clock is selected by bit 26 (USE\_SCHREF) of the SCH\_CTRL register, as shown in [Table 6-2](#).

*Table 6-2. Scheduler Clock Selection*

| USE_SCHREF | Scheduler Clock |
|------------|-----------------|
| 0          | SYSClk          |
| 1          | SCHREF          |

Figure 6-3 represents an example of a schedule table. In this example, the system designer has chosen 100 schedule slots. The duration of each slot is a programmable number of clock cycles. 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 takes 95 clocks with a clock frequency of 33 MHz. The scheduler begins at slot 00 and increments its position in the table every 95 clocks. After 9500 clocks, the scheduler position returns to 00.

Figure 6-3. Schedule Table with Size = 100



8236\_037

### 6.2.2.2 Schedule Table Slots

The user can select from a wide range of possible schedule slot formats. The user selects a format based on system requirements. If the system needs to generate CBR traffic, 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 USE\_SCH\_CTRL bit in the SEG\_CTRL register is used to turn on and turn off the mechanisms that create the 16 scheduling priorities. This maintains backward compatibility to earlier versions of the SAR, where only eight scheduling priorities were used. If the USE\_SCH\_CTRL bit is asserted, the user controls the size and format of all schedule slots through the SLOT\_DEPTH, 4-bit VBR\_OFFSET, and TUN\_PRI0-OFFSET fields in the SCH\_CTRL register,



and the CBR\_TUN bit in the SEG\_CTRL register. If the USE\_SCH\_CTRL bit in the SEG\_CTRL register is not asserted, the user controls the size and format of all schedule slots through the DBL\_SLOT, 3-bit VBR\_OFFSET, and CBR\_TUN fields in the SEG\_CTRL register. These factors, coupled with the size of the Schedule table, determine the memory requirements for the Schedule table.

Table 6-3. Selection of Schedule Table Slot Size by System Requirements

| CBR Service | Number of VBR/ABR Priorities Required | Schedule Slot Size | CBR_TUN | SLOT_DEPTH                           | Available VBR/ABR Priority Levels |
|-------------|---------------------------------------|--------------------|---------|--------------------------------------|-----------------------------------|
| No          | 16                                    | 8 Words (256 bits) | 0       | 111                                  | 0–15 <sup>(1)</sup>               |
| Yes         | 15                                    | 8 Words (256 bits) | 1       | 111                                  | 1–15 <sup>(1)</sup>               |
| No          | 14                                    | 7 Words (224 bits) | 0       | 110                                  | VBR_OFFSET + 0–13                 |
| Yes         | 13                                    | 7 Words (224 bits) | 1       | 110                                  | VBR_OFFSET + 1–13                 |
| No          | 12                                    | 6 Words (192 bits) | 0       | 101                                  | VBR_OFFSET + 0–11                 |
| Yes         | 11                                    | 6 Words (192 bits) | 1       | 101                                  | VBR_OFFSET + 1–11                 |
| No          | 10                                    | 5 Words (160 bits) | 0       | 100                                  | VBR_OFFSET + 0–9                  |
| Yes         | 9                                     | 5 Words (160 bits) | 1       | 100                                  | VBR_OFFSET + 1–9                  |
| No          | 8                                     | 4 Words (128 bits) | 0       | 011                                  | VBR_OFFSET + 0–7                  |
| Yes         | 7                                     | 4 Words (128 bits) | 1       | 011                                  | VBR_OFFSET + 1–7                  |
| No          | 6                                     | 3 Words (96 bits)  | 0       | 010                                  | VBR_OFFSET + 0–5                  |
| Yes         | 5                                     | 3 Words (96 bits)  | 1       | 010                                  | VBR_OFFSET + 1–5                  |
| No          | 4                                     | 2 Words (64 bits)  | 0       | 001<br>(DBL_SLOT = 1) <sup>(2)</sup> | VBR_OFFSET + 0–3                  |
| Yes         | 3                                     | 2 Words (64 bits)  | 1       | 001<br>(DBL_SLOT = 1) <sup>(2)</sup> | VBR_OFFSET + 1–3                  |
| No          | 2                                     | 1 Word (32 bits)   | 0       | 000<br>(DBL_SLOT = 0) <sup>(2)</sup> | VBR_OFFSET + 0 or 1               |
| Yes         | 1                                     | 1 Word (32 bits)   | 1       | 000<br>(DBL_SLOT = 0) <sup>(2)</sup> | VBR_OFFSET + 1                    |

**NOTE(S):**  
<sup>(1)</sup> VBR\_OFFSET would be set to 0 for this mode of operation.  
<sup>(2)</sup> The bottom four rows of this table describe the slot size formats when USE\_SCH\_CTRL is not asserted.

### 6.2.2.3 Schedule Slot Formats without USE\_SCH\_CTRL Asserted

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](#). 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 is assigned a VBR VCC\_Index has 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.

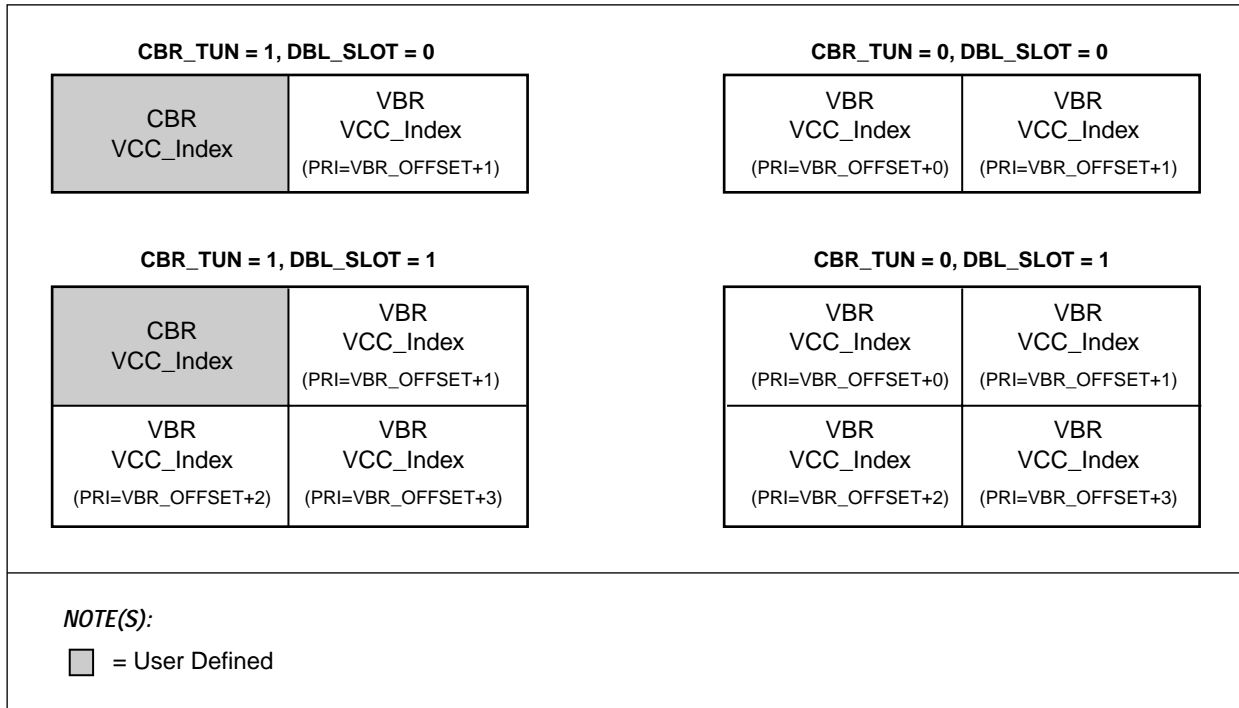
#### *VBR\_OFFSET Described*

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 (that is, PRI = 5), then

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

Figure 6-4 illustrates how these priorities are assigned to the VBR fields.

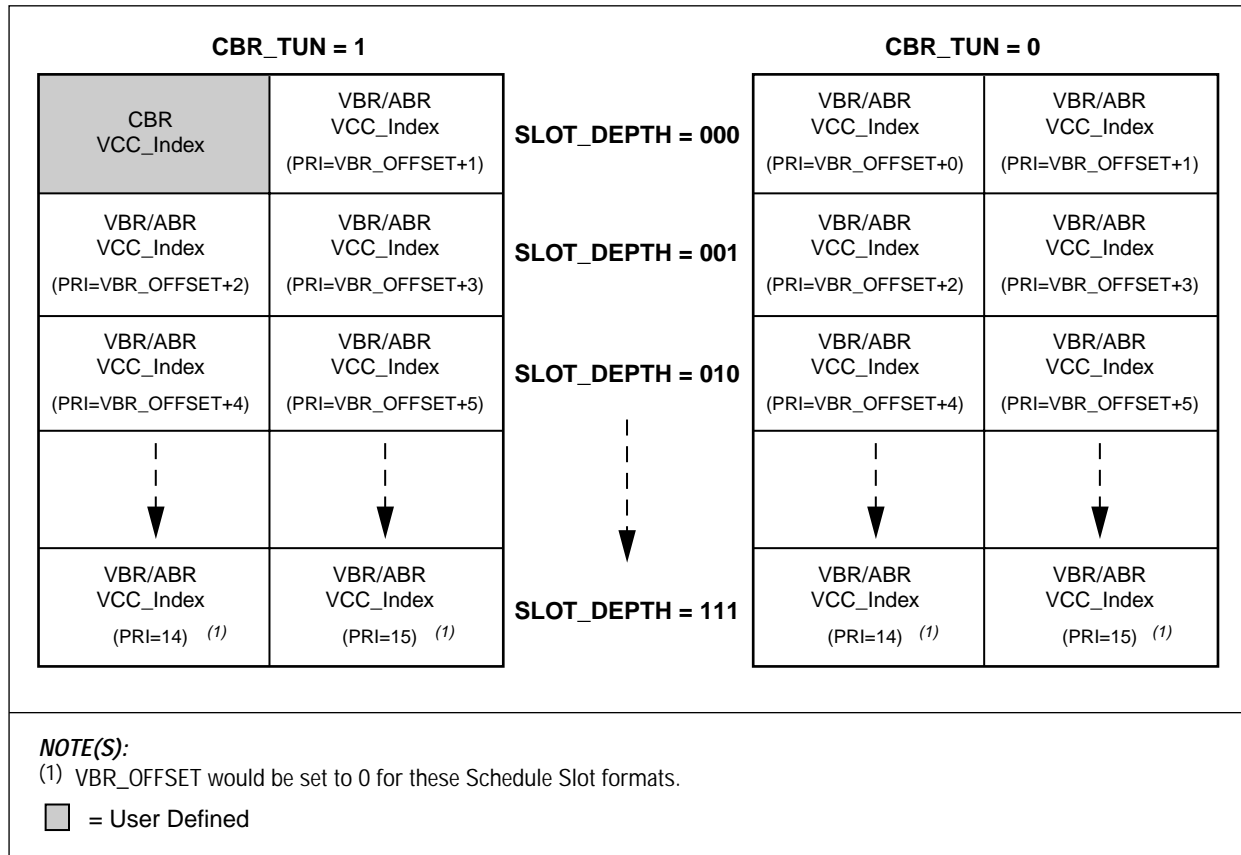
Figure 6-4. Schedule Slot Formats With USE\_SCH\_CTRL Not Asserted



8236\_107

**6.2.2.4 Schedule Slot Formats with USE\_SCH\_CTRL Asserted** When USE\_SCH\_CTRL is asserted, the SLOT\_DEPTH and VBR\_OFFSET fields in the SCH\_CTRL register and the CBR\_TUN field in the SEG\_CTRL register dictate the format of each schedule slot. This format is illustrated in [Figure 6-5](#).

Figure 6-5. Schedule Slot Formats With USE\_SCH\_CTRL Asserted



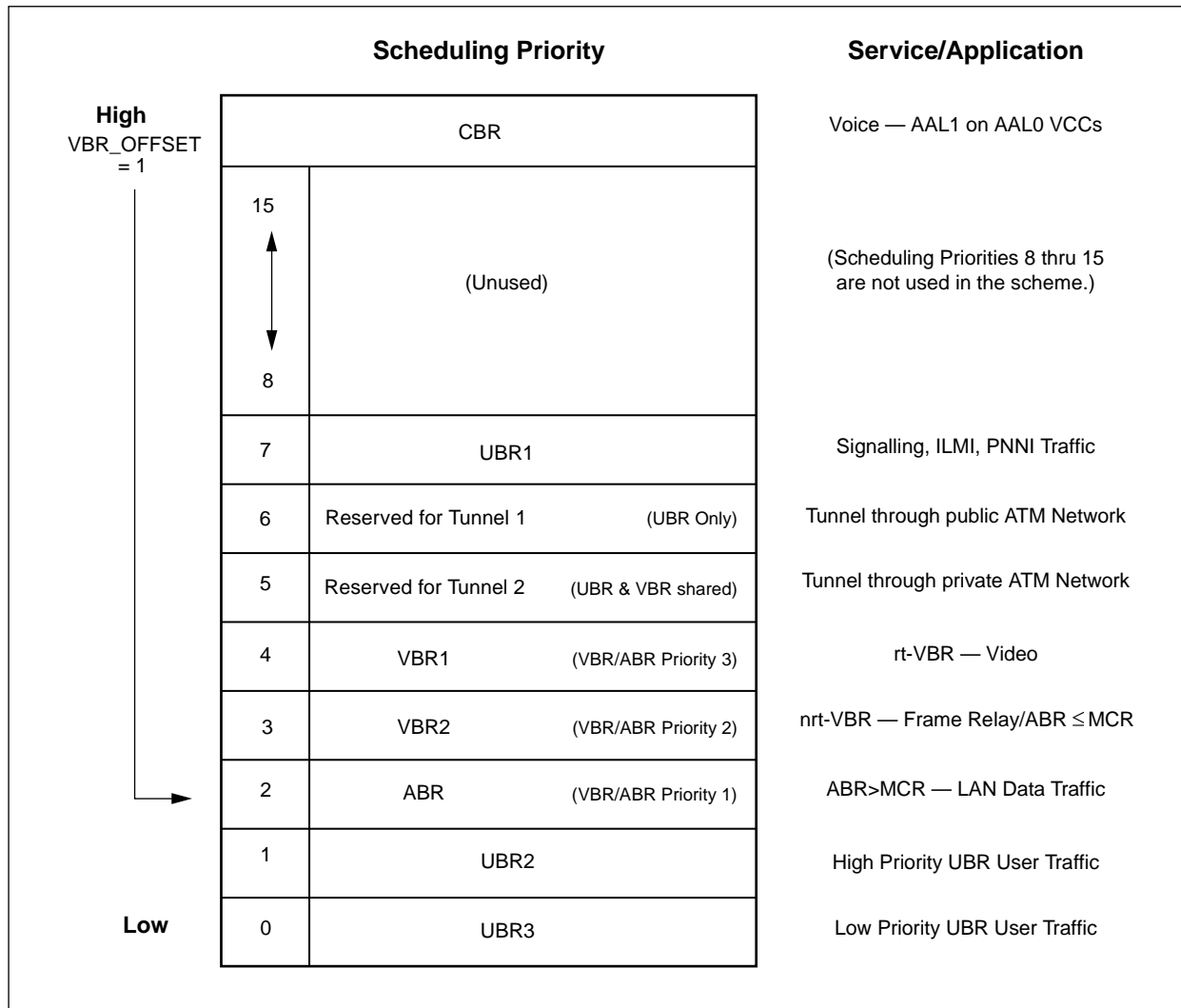
8236\_108

**6.2.2.5 Some Scheduling Scenarios**

This section discusses a few examples of scheduling priority schemes that system designers may decide to implement. These examples demonstrate the range of flexibility that the CN8236 affords system designers and network administrators.

Figure 6-6 illustrates how scheduling priorities are assigned in scheduling slots. In this illustration, the number of VBR/ABR priorities = 3. In addition, the tunnel at PRI = 5 has shared VBR traffic. CBR\_TUN = 1, SLOT\_DEPTH = 001. The highest VBR/ABR scheduling priority (PRI) is 5. Thus, VBR\_OFFSET = 1.

Figure 6-6. One Possible Scheduling Priority Scheme with the CN8236



8236\_038

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:

$$VBR\_OFFSET + (\# \text{ of VBR/ABR priorities}) \leq 15$$

### 6.2.3 CBR Traffic

The CBR service category guarantees end-to-end bandwidth through the network. Certain data sources, such as voice circuits and constrained Cell Delay Variation (CDV), require this guarantee. 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 CN8236 generates CBR traffic by assigning specific slots in the Schedule table to a CBR VCC. This connection always sends 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 clock cycles 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 CN8236 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.

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

SLOT\_PER must 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 ( $R_{max}$ ) 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 as follows:

$$R_{max}, R_{max} \times (TBL\_SIZE - 1) / TBL\_SIZE, R_{max} \times (TBL\_SIZE - 2) / TBL\_SIZE, \dots, R_{max} / TBL\_SIZE$$

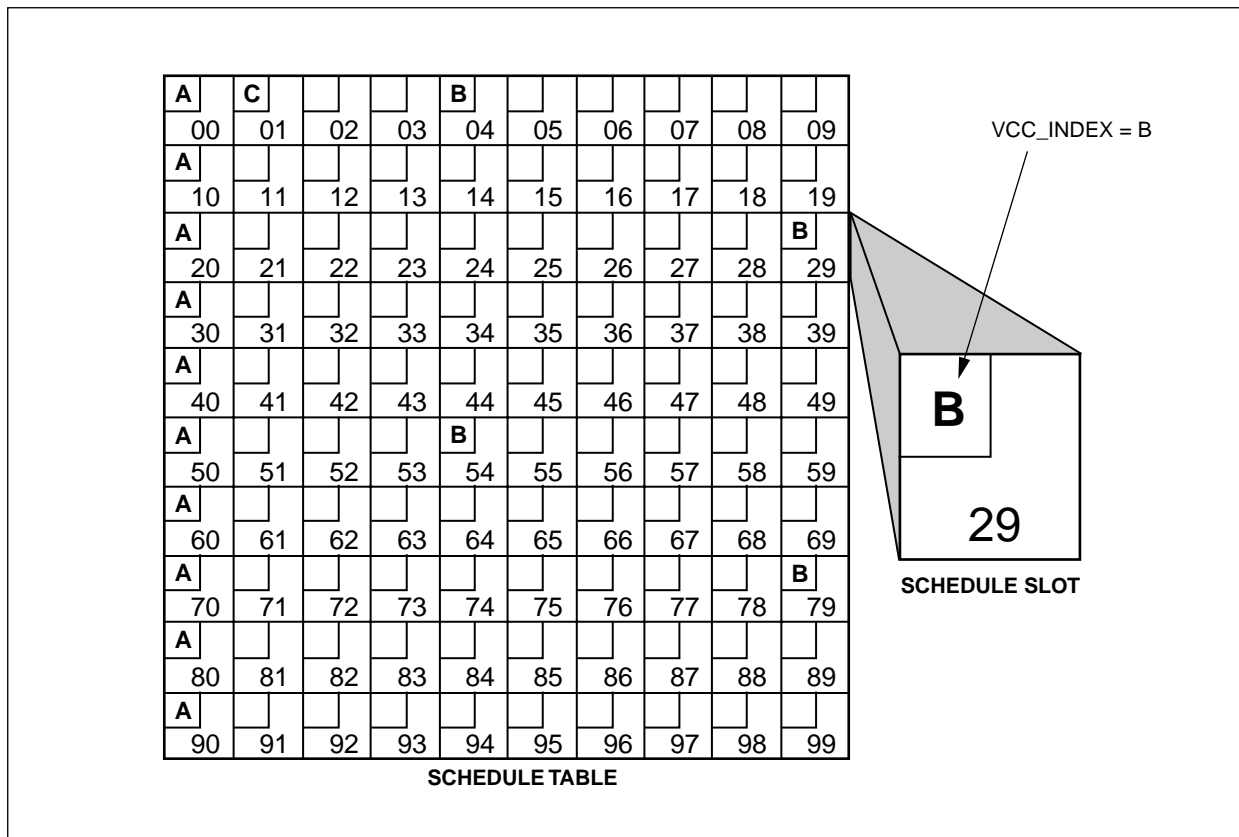
**Minimum Rate**

The minimum rate available is therefore  $R_{max} / TBL\_SIZE$ .

**Assigning CBR Cell Slots**

Figure 6-7 displays an example of a Schedule table with slots assigned to various CBR channels, each with a different rate. In this example,  $TBL\_SIZE = 100$  and  $SLOT\_PER = 93$ . VCC\_INDEX A occupies every tenth schedule slot; therefore, it transmits at  $R/10$ , or 35.4 K cells/second. VCC\_INDEX B occupies a slot every 25 cell slots and transmits at a rate of  $R/25$ , or 14.2 K cells/second. VCC\_INDEX C occupies only one schedule slot and therefore transmits at the minimum rate,  $R/TBL\_SIZE$ , or 3.54 K cells/second. Not all cell slots have been assigned to CBR channels. During these slots, the CN8236 dynamically schedules traffic from the other service classes. However, the total bandwidth of channels A, B, and C is reserved.

Figure 6-7. Assigning CBR Cell Slots

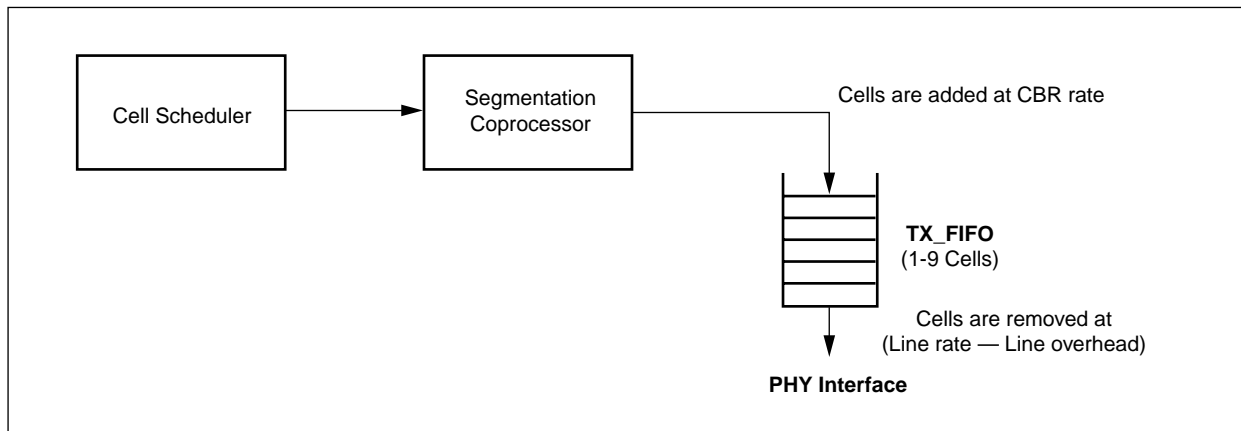


8236\_039

6.2.3.3 CBR Cell Delay Variation (CDV)

CBR connections are sensitive to CDV. (See *ATM Forum UNI 3.1* or *TM 4.1* for a complete definition of CDV.) The CN8236's Traffic Manager minimizes CDV by basing all traffic management on the Scheduler clock 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 CN8236, the dominant factor in CDV is the variation introduced between the segmentation coprocessor and the PHY layer device at the Transmit FIFO buffer (TX\_FIFO). Line overhead created by the framer in the PHY layer device causes this variation. [Figure 6-8](#) illustrates this interface.

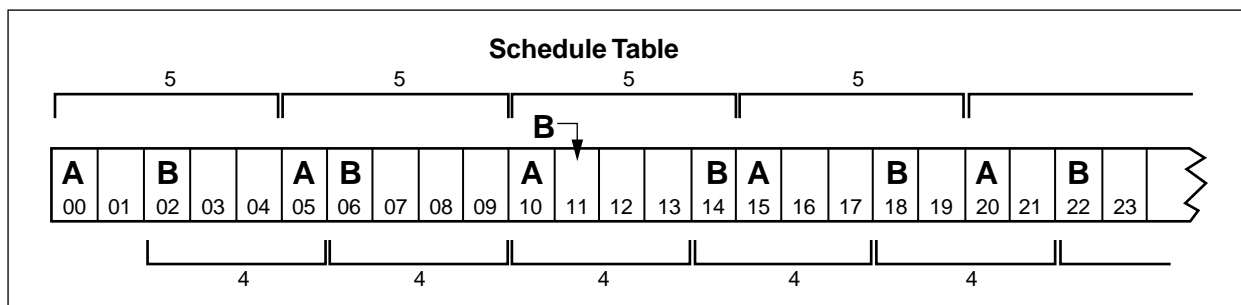
Figure 6-8. Introduction of CDV at the ATM/PHY Layer Interface



8236\_040

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 6-9](#) illustrates an example of a linear representation of a Schedule table with 100 schedule slots. CBR channel A has reserved bandwidth for a rate of 70 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 is a schedule slot reservation conflict between channels A and B every 20th cell slot, and one of the channel's slots has to be reserved one slot later.

Figure 6-9. Schedule Table with Slot Conflicts at Different CBR Rates

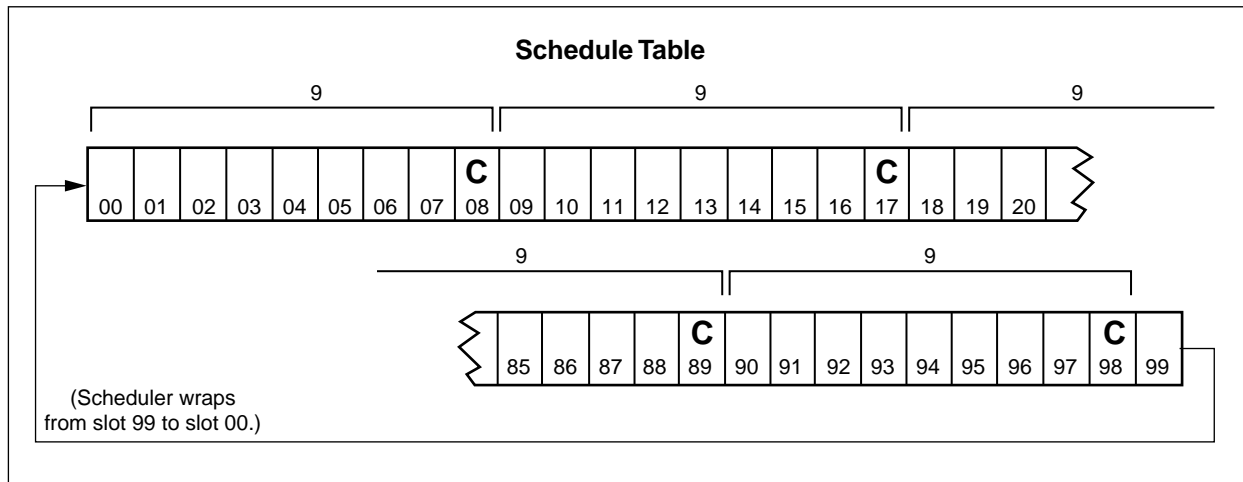


8236\_041



Another possible source of Schedule-table-dependent CDV occurs for certain CBR rates whose schedule slot spacings do not evenly divide the table slot size. An example of this is illustrated in Figure 6-10, where 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. When the Scheduler wraps from the end of the table to the beginning of the table, the number of schedule slots between the last C channel slot at the end of the table and the first C channel slot at the beginning of the table is 10 slots, not 9.

Figure 6-10. CDV Caused by Schedule Table Size at Certain CBR Rates



8236\_042

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 Section 6.2.2.4).

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

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 must be set to 001 (CBR). Thereafter, the CN8236 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 CN8236 dynamically allocates 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 does not exactly match the rate of its data source. To compensate for this asynchronous data source behavior, the CN8236 provides a rate-matching mechanism for use when CBR traffic is mapped to a Virtual FIFO buffer. (This method is needed only for Virtual FIFO buffers. For VCCs that are not Virtual FIFO buffers, the cell transmission is skipped automatically if there is no data available.)

For an individual CBR channel mapped to a Virtual FIFO buffer, the host directs the CN8236 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 CN8236 encounters a CBR slot for this VCC, it does not transmit data on that VCC. Then, the CN8236 indicates to the host that a slot has been skipped by clearing the SCH\_OPT bit.

With this method, the host effectively synchronizes the CN8236'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 CN8236 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 use of the shared bandwidth resources.

### 6.2.4.1 Mapping CN8236 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 CN8236 device. These relationships are described in [Table 6-4](#).

*Table 6-4. CN8236 VBR to TM 4.1 VBR Mapping*

| CN8236 VBR     | TM 4.1 VBR   | Comments   |
|----------------|--------------|--|
| 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 standards 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 segmentation 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 PCR, and *L* is used to control the CDVT of the outgoing cell stream. The user enables this scheme by setting the SCH\_MODE bits to 100 (VBR1).

### 6.2.4.4 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, the limitation is 256 values for the *I2* and *L2* parameters. These parameters are stored as bucket table entries. (See [Table 6-14](#), for the definition of a bucket table entry.) There is complete flexibility with regard to using these 256 values to specify SCR or PCR. In VBR2, *I1* and *L1* can specify either PCR and CDVT, or SCR and Burst Tolerance (BT), with *I2* and *L2* used to specify the parameters not assigned to *I1* and *L1*.

For example, to 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 correspond to PCR. This controls the PCR of the total cell stream, but only controls 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. Since VBR channels are internally scheduled by the Cell Scheduler based on that channel's *I* parameter, a floating point value, they are not constrained to repeat at some *n* integer value of  $R_{\max} / (TBL\_SIZE - n)$ . Thus, VBR channels 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

$$((\# \text{ VBR VCCs at same-or-higher priority}) + TX\_FIFO\_LEN) / (\text{line rate in cells/sec})$$

### 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, 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 xBR 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. 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, all priority levels used within a

CBR tunnel need to be activated as a tunnel by setting the TUN\_ENA( × ) bit(s) in the SCH\_PRI and SCH\_PRI2 registers.

Sixteen tunnels can be active at once, if each of these tunnels has only one scheduling priority assigned. All can be VBR/ABR tunnels. On the other hand, the system designer can establish four tunnels, each of which has four scheduling priorities assigned to it, or any combination of tunnels, which includes up to a total of 16 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.

The 3-bit PRI0 field allows the entry of only priority levels 0 through 7. If the user wishes to enter a priority level higher than priority 7 in PRI0, assign the difference in values as an offset, in the TUN\_PRI0\_OFFSET field in the SCH\_CTRL register.

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 gets 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.

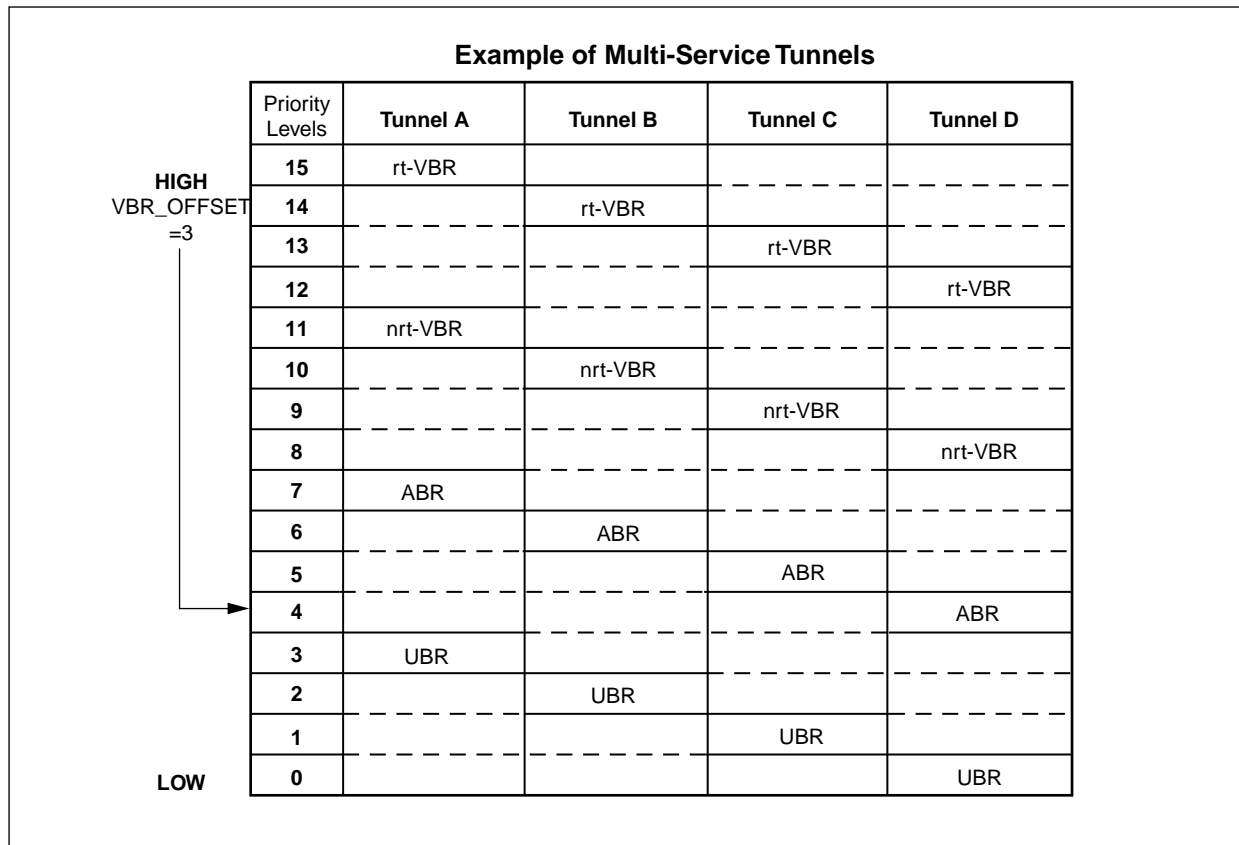
In this example, the user has configured the CN8236 to manage two independent tunnels. The first tunnel priority five, is through a private ATM network, perhaps a corporate ATM campus backbone. The other tunnel, priority six, carries traffic through a public network. This topology allows the end user to lease reserved CBR bandwidth from an administrative domain, but manage the usage of the tunnel in an arbitrary fashion.

**Figure 6-11** illustrates a different use of CBR tunnels. In this example, the user has established 4 separate tunnels or pipes. Each can have an equal 1/4 share of the bandwidth available to the port by provisioning every fourth schedule table slot to one tunnel for each of the four tunnels.

In each of the tunnels, 4 priority levels are assigned. The highest priority in each tunnel is assigned to real-time VBR, the next highest priority to non-real-time VBR, the third highest priority to ABR, and the lowest priority to UBR. This in effect establishes four multi-service pipes, each on equal priority to the others (since the scheduler services each of the four pipes equally). Thus, the 4 rt-VBR priorities have effectively the same priority, and so on through the four levels of priority serviced in each pipe. Number of VBR/ABR priorities is 12. CBR\_TUN = 1, SLOT\_DEPTH = 110.

Highest VBR/ABR Scheduling priority is 15 (that is, PRI = 15). Thus, VBR\_OFFSET = 3.

Figure 6-11. Another Possible Scheduling Priority Scheme with the CN8236



8236\_043

### 6.2.7 Guaranteed Frame Rate

GFR is a new service category defined by the ATM Forum to provide an 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 CN8236 implements GFR by scheduling or 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) must 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 not oversubscribe MCR on GFR channels and other rate-guaranteed services.

The CN8236 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 6-18](#)), 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 as follows:

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

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](#).)

### 6.2.8 PCR Control for Priority Queues

The CN8236 provides an optional method of creating tunnels (that is, 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 are 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 sixteen 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 and SCH\_PRI\_2 registers. 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 is shaped using the QPCR\_INT3 value.

If two priority queues are enabled for PCR shaping, QPCR\_INT3 and QPCR\_INT2 are used, and so on.

## 6.3 ABR Flow Control Manager

### 6.3.1 A Brief Overview of *TM 4.1*

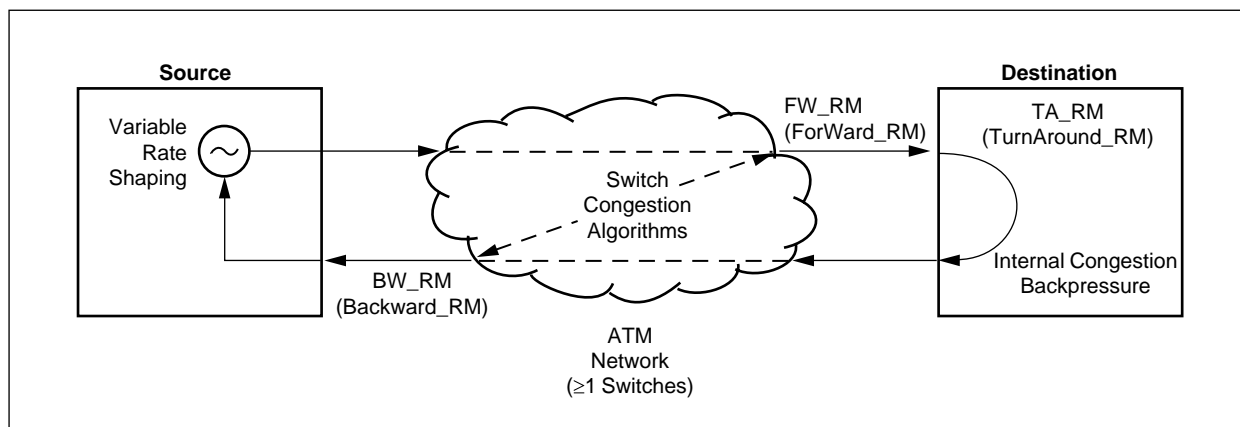
This section briefly describes the *TM 4.1* ABR Flow Control algorithms. However, it is strongly recommended that the reader be familiar with the *ATM Forum's TM 4.1 Specification* before attempting to understand the CN8236's ABR implementation.

### 6.3.2 Internal ABR Feedback Control Loop

As a complete implementation of the UNI ATM layer, the CN8236 acts as an ABR Source and Destination, complying with all required *TM 4.1* ABR behaviors. The CN8236 utilizes the dynamic rate adjustment capability of the xBR Cell Scheduler as the Source's variable rate-shaper. An internal feedback mechanism supplies feedback from the received cell stream to the ABR Flow Control Manager, a special purpose state machine. This state machine translates the feedback extracted from received Backward RM cells, to instructions for the xBR Cell Scheduler and segmentation coprocessor. It supports both binary and ER flow control methods.

Figure 6-12 illustrates this basic concept.

Figure 6-12. ABR Service Category Feedback Control



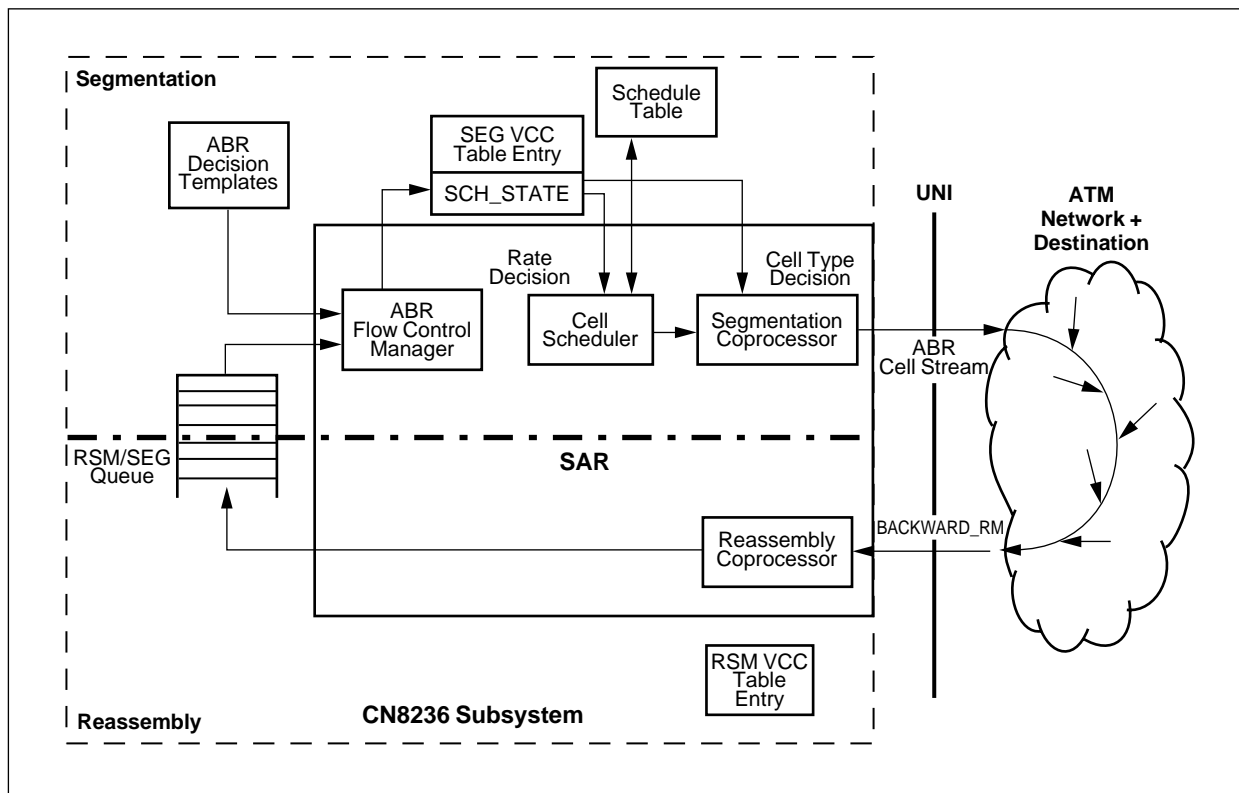


6.3.2.1 Source Flow Control Feedback

Figure 6-13 illustrates the feedback loop which controls the Source cell stream. The CN8236 injects an in-rate cell stream (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 CN8236 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 6-13. CN8236 ABR-ER Feedback Loop (Source Behavior)



8236\_045

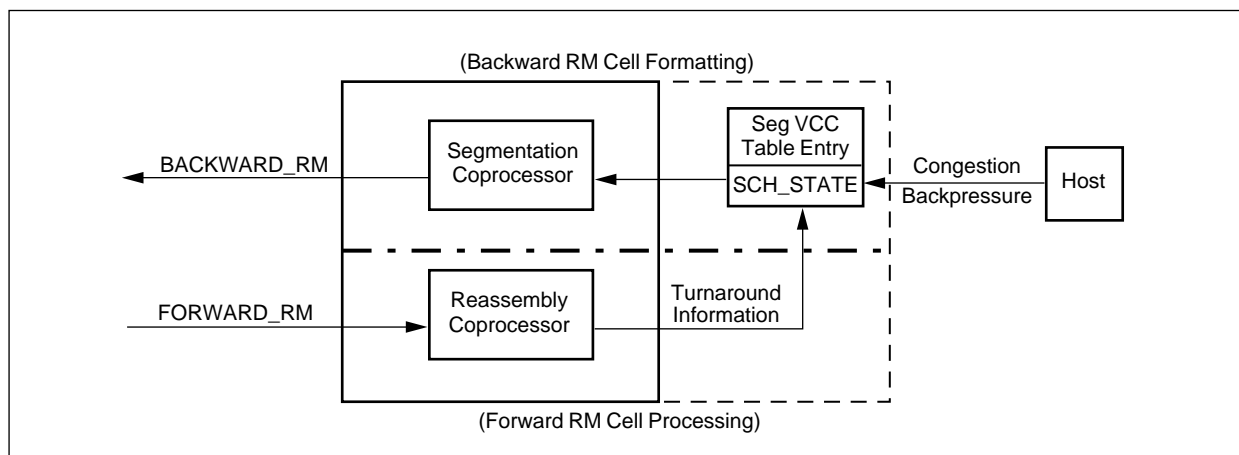
### 6.3.2.2 Destination Behavior

The CN8236 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 CN8236 also provides a mechanism for the Destination host to apply backpressure to the Source. The host notifies the CN8236 of internal system congestion. The CN8236 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 6-14](#).

Figure 6-14. CN8236 ABR-ER Feedback Generation (Destination Behavior)



8236\_046

### 6.3.2.3 Out-of-Rate Cells

In addition to in-rate cell streams, the CN8236 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 is 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 0 or below the schedule table minimum rate.

### 6.3.3 Source and Destination Behaviors

ABR's Source and Destination Behavior Definitions are each listed in the *ATM Forum's 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 PCR, Initial Cell Rate (ICR), MCR, RIF, RDF, and Nrm.

The user can define 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 CN8236 as a complete microcoded state machine.

Three components constitute a complete template: a Cell Decision table, a Rate Decision table, and an Exponent table.

The CN8236 uses the Cell Decision table to select a cell type for insertion into the transmitted in-rate cell stream.

The Rate Decision table maps ABR cell stream rates to logical states. Each state includes several vectors to other states, corresponding to different rates. When an event forces a rate decision, the CN8236 decides which vector to follow based on network stimulus (CI/NI), ER decisions, or internal timer/counter expirations (ADTF and CRM).

The Exponent table contains parameters for a piece-wise linear mapping function. This function maps a *TM 4.1* floating point rate representation (in particular, the RM cell Explicit Rate field) to a Rate Decision table index. The Exponent table is indexed by the Explicit Rate exponent. This mapping function normalizes the ER field rate to the CI/NI state-based rate decision. Once normalized, a rate can be chosen based on the minimum of the two rates.

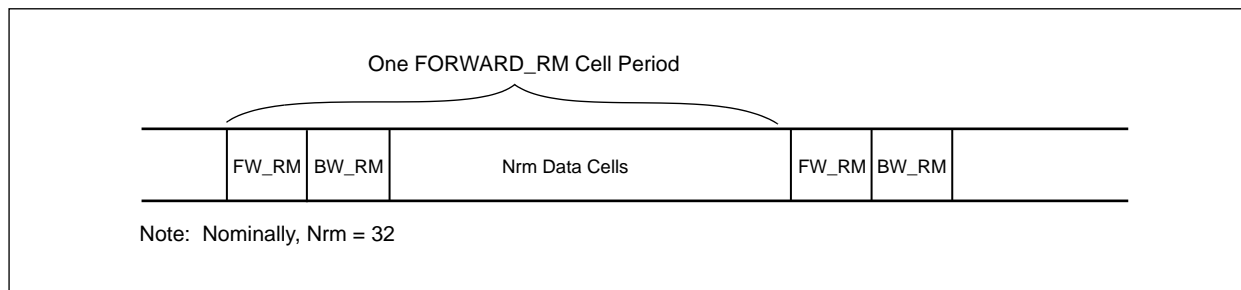
### 6.3.6 Cell Type Decisions

#### 6.3.6.1 In-rate Cell Streams

Since every ABR connection in a network is full duplex, the CN8236 is a Source and a Destination for each VCC. The CN8236 multiplexes user data cells, Forward RM cells, and Backward RM cells into the in-rate ABR cell stream. *TM 4.1 Source Behaviors* specify rules for the in-rate cell stream.

Figure 6-15 shows the desired result of the specification. First, the Source transmits one Forward RM cell. A Backward RM cell follows. Finally, the Source sends *Nrm* user data cells. At steady state, this sequence would be repeated with a Forward RM cell marking the beginning of each sequence.

Figure 6-15. Steady State ABR-ER Cell Stream



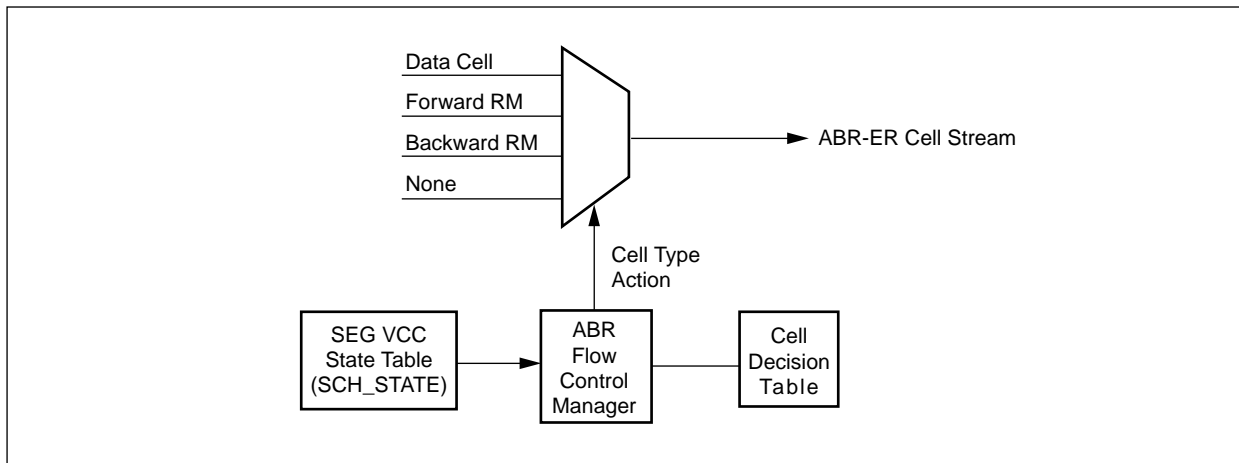
8236\_047

Unfortunately, due to the asynchronous, asymmetrical, and fluctuating characteristics of ABR connections, all connections do not maintain a steady state sequence. Furthermore, since it is an immature specification, the rules for cell interleaving of in-rate cell streams may be modified slightly.

The CN8236 provides a programmable mechanism to comply with *TM 4.1's* specification on cell stream interleaving. This Cell Type Decision algorithm makes on-the-fly decisions concerning which type of cell to send, based on the current state of the connection. Figure 6-16 shows a block diagram of the algorithm. The ABR Flow Control Manager chooses a cell type based on a Cell Decision table and the VCC state. The segmentation coprocessor then formats the appropriate cell type and sends it on the ABR in-rate cell stream.

**NOTE:** The ABR Flow Control Manager may choose to send no cell. This occurs when the VCC has no user data to segment.

Figure 6-16. Cell Type Interleaving on ABR-ER Cell Stream



8236\_048

Since the Cell Decision table template resides in SAR-shared memory, the user can modify it to optimize performance or react to changes in the ABR specification. Furthermore, multiple tables allow groups of VCCs to be tuned for different network policies.

### 6.3.6.2 ABR Cell Decisions

The Cell Decision Tables reside in SAR-shared memory as a segment of the Flow Control Manager's ABR Instruction table. Each entry of a table is an ABR Cell Decision Block (ACDB). Each ACDB contains sixteen Cell Type Actions. Cell Type Actions result in one of four decisions: send in-rate Forward RM cell, send in-rate Backward RM cell, send user data cell, or no action. A Cell Type Action is chosen by a 4-bit ABR Cell Type Decision Vector (ACDV). The four bits of the ACDV correspond to four current ABR VCC conditions.

#### Initialization Instructions

At system initialization, the host loads one or more Cell Decision Tables into SAR-shared memory. Mindspeed provides standard Cell Decision Tables. The system designer uses either these or customized templates.

To assign an ABR VCC to a table, the host initializes two SCH\_STATE fields at connection setup time. The first field, CELL\_INDEX, tracks the current ACDB position of the VCC. Source Behavior #2 specifies that the first in-rate cell on a VCC is a Forward RM cell. Therefore, CELL\_INDEX should be initialized to an ACDB position which always decides to send a Forward RM cell. The second field, FWD\_INDEX, provides a reset point for the in-rate cell stream sequence. Whenever the CN8236 transmits an in-rate Forward RM cell, it copies FWD\_INDEX to CELL\_INDEX.

#### Run-time Operation

Each time an ABR cell-transmit opportunity arises, the CN8236 makes a cell decision. The SAR retrieves CELL\_INDEX from SCH\_STATE and chooses a Cell Type Action location within the current ACDB. Table 6-5 shows the four conditions that are used to form this vector. The presence of a condition sets the vector bit to a 1.

Table 6-5. ABR Cell Type Decision Vector (ACDV)

| Bit | Name    | Description                                   |
|-----|---------|---|
| 3   | TRM_EXP | Time since last forward RM transmitted >= TRM |
| 2   | TA_PND  | TA_PND bit set in VCC table entry             |
| 1   | TA_XMIT | TA_XMIT bit set in VCC table entry            |
| 0   | RUN     | RUN bit set in VCC table entry                |

The SAR updates the CELL\_INDEX field after each cell decision is made. If an in-rate Forward RM cell is transmitted, FWD\_INDEX is copied to CELL\_INDEX; otherwise, CELL\_INDEX is incremented by one (CELL\_INDEX++). Therefore, the number of blocks in a Decision table is bounded by the maximum number of cell decisions made between transmitted Forward RM cells, typically Nrm.

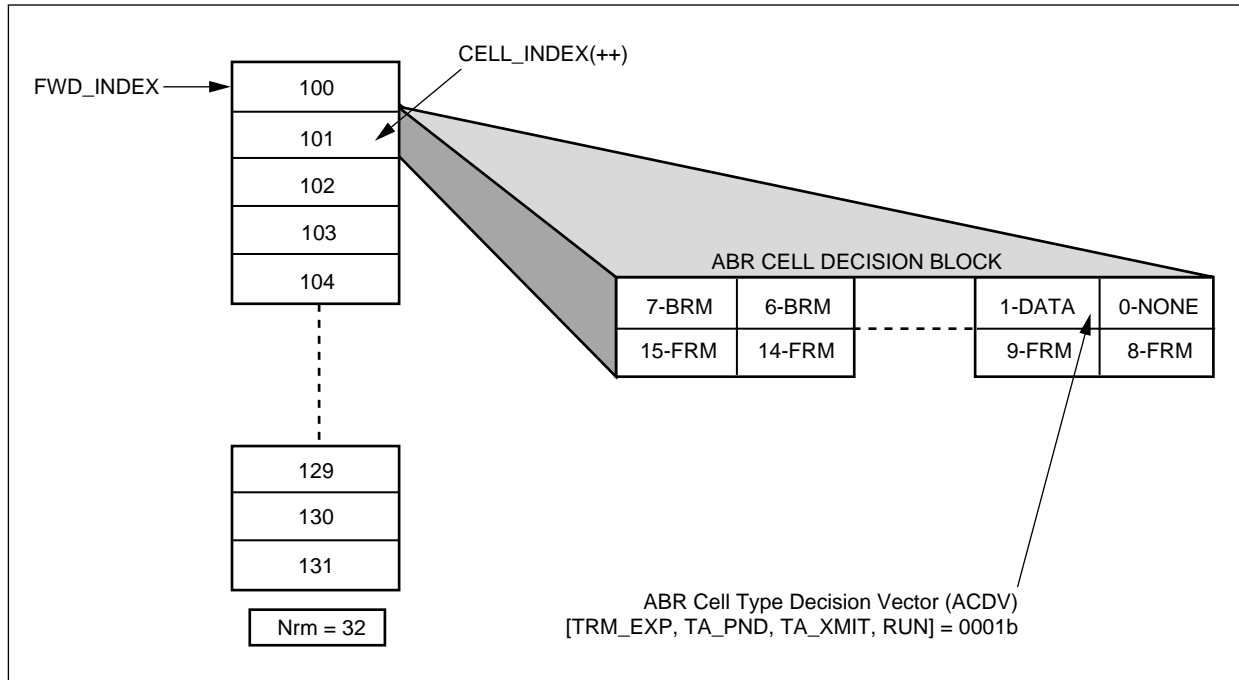
**APPLICATION EXAMPLE: Cell Decision Table for Nrm = 32**

**Figure 6-17** shows a typical Cell Decision table. The anchor of the table, the FWD\_INDEX ACDB, is located at index 100 of the ABR Decision table. Whenever a Forward RM cell is sent, this ARCB serves as the reset point for the decision state machine. When CELL\_INDEX = FWD\_INDEX, a steady state ABR channel transmits a Backward RM cell and advances the CELL\_INDEX. The diagram shows the next ACDB (101) in more detail. It contains the 16 Cell Type Actions (For example, BRM = Send Backward RM). At the next cell transmission opportunity, the ABR Flow Control Manager state machine selects a cell type by indexing into the ACDB according to the ACDV. In this case, the only condition present is RUN = 1, so ACDV = 0001b. Cell Type Action Index 1 is selected. For this ACDB, Cell Type Action 1 is DATA or Send Data Cell. Therefore, the CN8236 transmits a user data cell.

In steady state, CELL\_INDEX is incremented until it reaches ACDB 131. This ACDB transmits a Forward RM Cell under all circumstances. Therefore, the CN8236 resets the CELL\_INDEX for this VCC to FWD\_INDEX = 100, when it reaches this ACDB.

In this example, the host should initialize CELL\_INDEX to 131. The CN8236 would transmit a Forward RM cell, and the sequence would continue from the FWD\_INDEX.

Figure 6-17. Cell Decision Table for Nrm = 32



## 6.3.7 Rate Decisions and Updates

### 6.3.7.1 ABR Traffic Shaping

The CN8236 schedules ABR VCCs as a single leaky bucket GCRA channel, but the rate is subject to continual adjustment. At initialization, the host assigns flow-controlled VCCs *I* and *L* parameters corresponding to the negotiated ICR. The Cell Scheduler rate-shapes the initial traffic according to these GCRA parameters. This shaped cell stream includes in-rate Forward and Backward RM cells.

When feedback from the network results in flow control rate adjustments, the ABR Flow Control Manager overwrites the *I* and *L* parameters in the SCH\_STATE of the VCC. The updated rates fall within the range of MCR and PCR, which are specified on a per-connection basis.

### 6.3.7.2 Rate Adjustment Overview

The CN8236 dynamically adjusts the rate of transmission for ABR VCCs based upon network feedback, individual VCC parameters, and the current transmission rate. The ABR Flow Control Manager uses all of these inputs to calculate the ACR of the connection. This ACR corresponds to *I* and *L* GCRA parameters. The CN8236 updates the *I* and *L* parameters of the VCC with the new ACR *I* and *L* values.

The Flow Control Manager updates the transmission rate in response to two types of events—the reception of a Backward RM cell or the transmission of a Forward RM cell. The decision process is unique for each event type.

**NOTE:** Both of these event types may occur in one cell slot. In this case, the CN8236 makes both decisions before updating ACR.

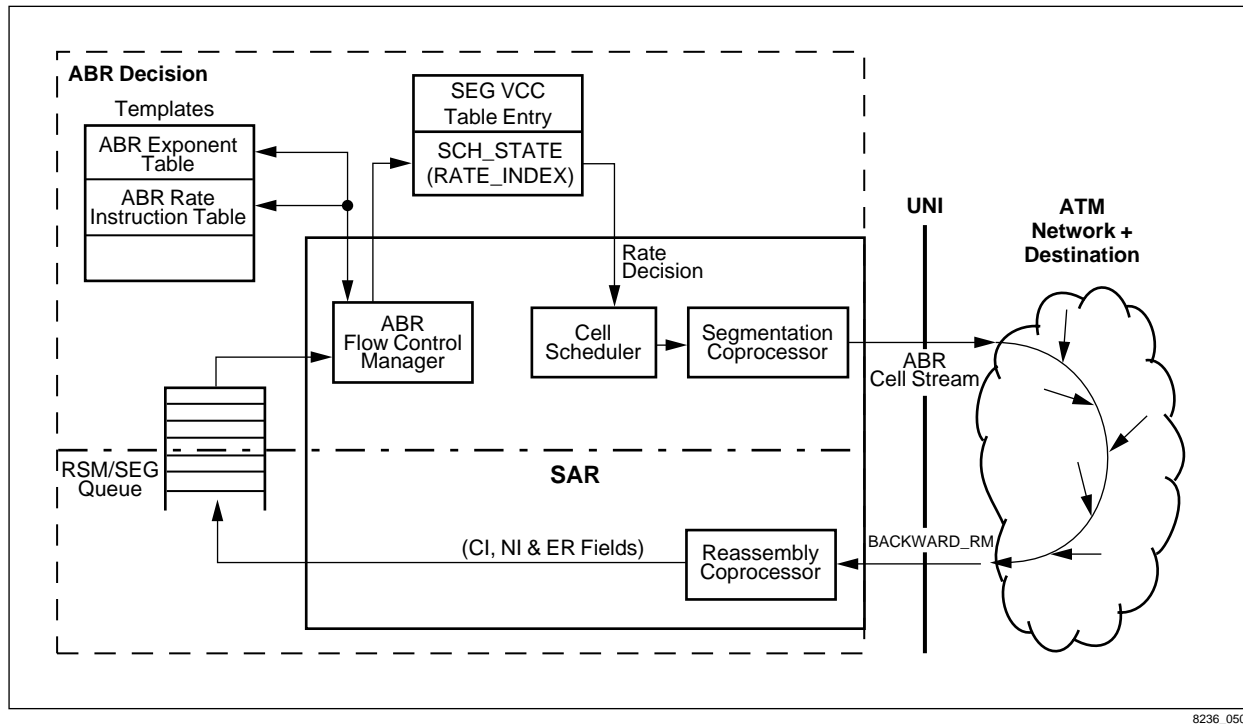
### 6.3.7.3 Backward RM Cell Flow Control

The ABR Rate Decision table consists of many indexed entries. Each of these entries is called an ABR Rate Decision Block or ARDB. Each ARDB represents a possible transmission rate for an ABR VCC. The rate of an ARDB is a monotonically increasing function of the index of the block.

The CN8236 tracks the state of each ABR connection's transmission rate by storing its current ARDB index in the RATE\_INDEX field of the SCH\_STATE VCC structure. This RATE\_INDEX uniquely identifies the current transmission rate of the VCC. [Figure 6-18](#) illustrates a block diagram of Backward RM flow control.



Figure 6-18. Backward\_RM Flow Control, Block Diagram



8236\_050

Backward RM cells deliver stimulus for two flow control algorithms. The first algorithm adjusts the state of a connection by a relative rate (RR) algorithm, in response to the CI and NI bits. The second algorithm allows the network to explicitly request a transmission rate in the ER RM cell field. Each of these algorithms produces a candidate rate. The Source must adjust its rate to be less than or equal to the lesser of these two candidates.

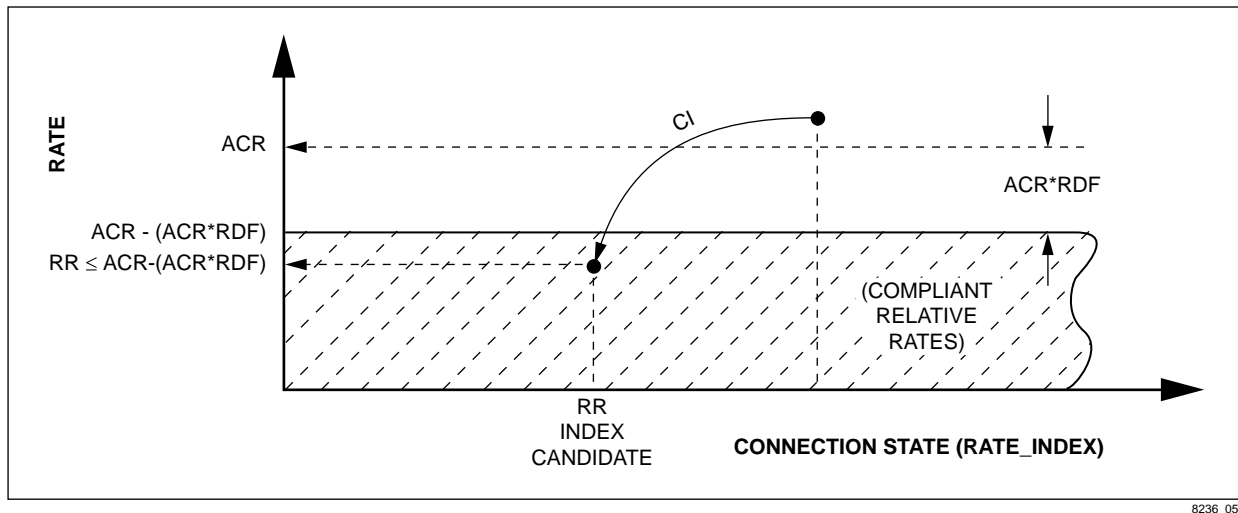
The CN8236 computes both rate candidates by first mapping all rate parameters into rate index space. Since rate indexes are a monotonically increasing function of rate, the CN8236 selects the candidate with the lowest rate index.

For the RR algorithm, the conversion is embedded in the ARDBs themselves. This algorithm can adjust the rate relative to the current rate. The adjusted rate relates to the current rate by a constant multiplicative decrease or additive increase. Specifically, the adjusted rate increases by  $(RIF \times PCR)$  or decreases by  $(RDF \times ACR)$ . These values are pre-calculated and used to formulate a CN8236 ABR Rate Instruction table, avoiding real time math.

Each ARDB contains vectors to eight other ARDBs. Four of these vectors are responses to the relative rate elements in backward RM cells. The CN8236 uses the CI/NI bits to choose an ARDB candidate from one of these four vectors. For instance, if CI is set, the VCC must reduce its ACR by  $(ACR \times RDF)$ . The vectors in the ARDB that is chosen when CI is set point to ARDBs whose corresponding rates are less than  $[ACR - (ACR \times RDF)]$ . The CN8236 recognizes the ARDB indicated by the chosen vector, as the RR rate index candidate.

Figure 6-19 illustrates the RR Rate\_INDEX candidate selection.

Figure 6-19. RR RATE\_INDEX Candidate Selection

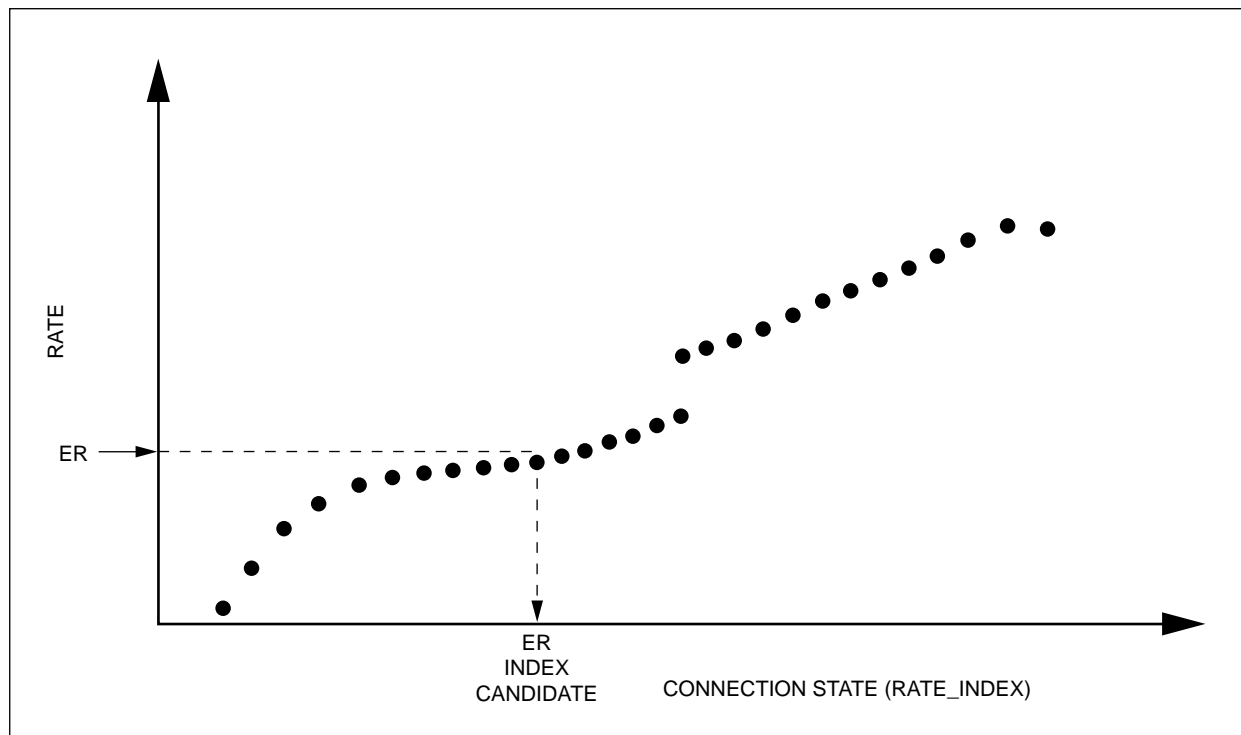


8236\_051

The CN8236 calculates the ER candidate by mapping the ER field from the Backward RM cell into rate index space. This mapping converts the floating point ER field rate representation to an absolute rate index. This mapping does not depend on the current rate of the connection. The CN8236 maps ER field to rate indexes with a programmable piece-wise linear function. Each piece-wise linear segment is described in the ABR Exponent table. Again, the system designer pre-calculates this mapping function to eliminate real-time floating point math.

Figure 6-20 illustrates the ER RATE\_INDEX candidate selection.

Figure 6-20. ER RATE\_INDEX Candidate Selection

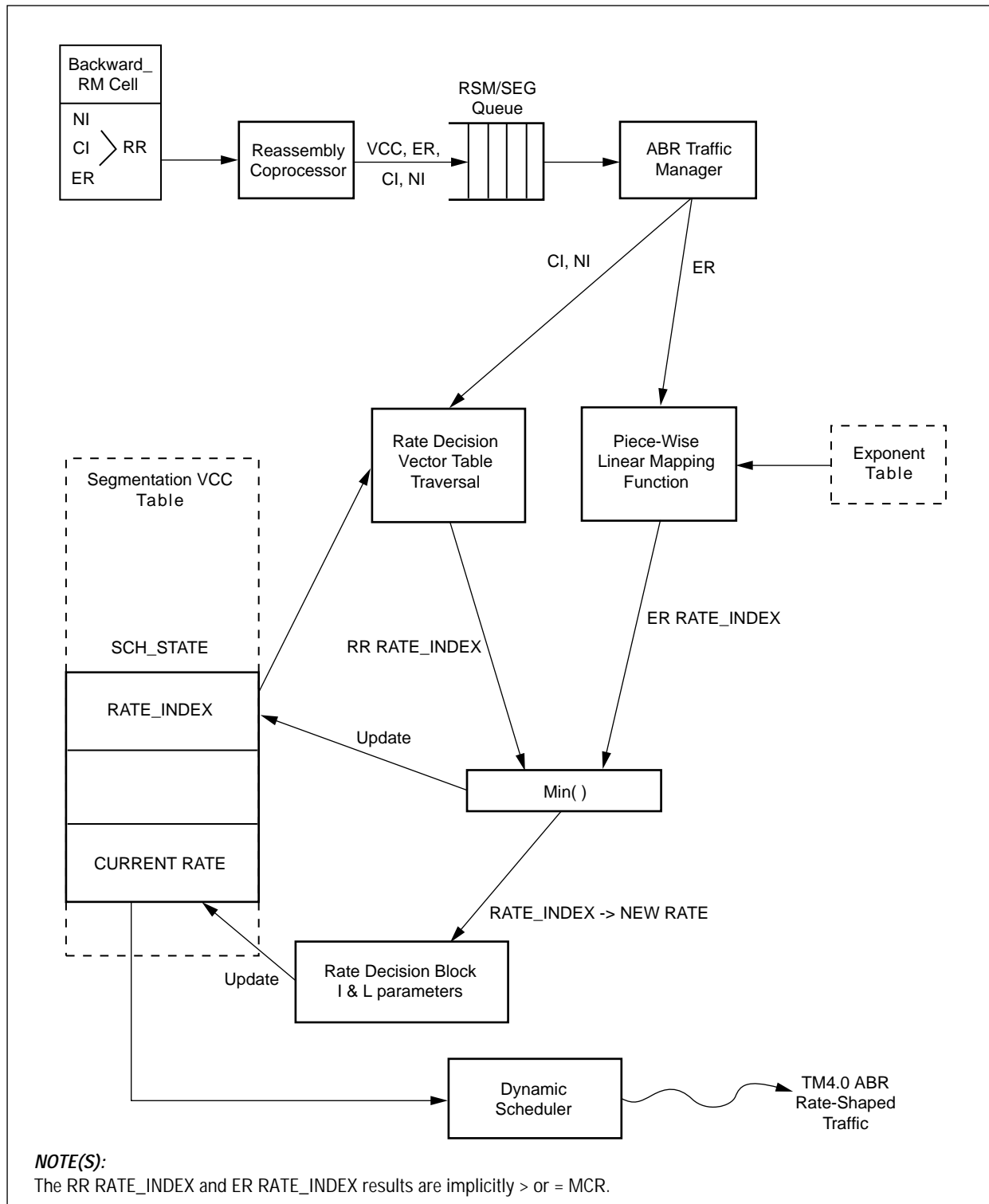


8236\_052

Once both candidates are identified, the CN8236 selects the smaller of the two, each of which are  $\geq$  MCR. The winning candidate then replaces the current RATE\_INDEX in the VCC's SCH\_STATE entry. Then, the CN8236 uses this RATE\_INDEX to point to the corresponding appropriate Rate Decision Block (RDB) and, as specified in *TM 4.1*, updates the VCC's rate parameter with the RDB's *I* and *L* parameters.

Figure 6-21 provides a block diagram of this process. The reassembly coprocessor passes the CI, NI, and ER fields from the RM cell to the ABR Flow Control Manager via the RSM/SEG Queue. The Flow Control Manager uses the ABR Rate Decision table and the Exponent table, both programmable components of an ER template, to calculate the two RATE\_INDEX candidates, each of which are  $\geq$  MCR. The min() function is applied to select a winning candidate. Then the CN8236 updates the VCC's rate parameters with the newly selected rate index. This rate index points to a Rate Decision Block whose *I* and *L* parameters establish the new rate. The Cell Scheduler uses the updated rate to schedule all future traffic until the next rate update.

Figure 6-21. Dynamic Traffic Shaping from RM Cell Feedback



8236\_109

#### 6.3.7.4 Forward RM Cell Transmission Decisions

The CN8236 also adjusts the transmission rates of an ABR in-rate cell stream before sending a Forward RM cell. This adjustment utilizes a Rate Decision Vector selection process. As stated above, four of the eight ARDB vectors are used for CI/NI rate adjustment. The other four are used for Forward RM cell rate adjustments.

Instead of the CI/NI bits, the Forward RM cell vector selection is based upon the ADTF timer and the CRM counter. These internal measures may force the connection to decrease its rate when sending an RM cell.

The ER for Forward RM cells on any channel is set in word 18 of the VCC table entry (FWD\_ER). The normal value for FWD\_ER is PCR. When this field is initialized in the VCC table entry, it must be set to the rate specified by the exponent table entries so that the resulting selected rate is  $\geq$  PCR. This is because when a Forward RM cell is eventually received as a Backward RM cell, the CN8236 maps the available cell rate (ACR) to a rate specified by the exponent table. If PCR falls between two exponent table rates and FWD\_ER is set to PCR, the ACR of the connection is limited to the lower of the two exponent table rates, thereby lowering the rate below PCR.

#### 6.3.7.5 ACR Change Notification

An ACR change notification mechanism per ATM Forum AF-SAA-0108\* Appendix D is implemented for both source and destination. Five fields in the SEG VCC table entry (S\_EN\_NCR, S\_NCR\_LO, S\_NCR\_HI, S\_NCR\_TRIG, and S\_NCR\_DIR) are used for source ACR change notification. Six fields in the RSM VCC table entry (D\_EN\_NCR, D\_NCR\_LO, D\_NCR\_HI, ND\_CR\_TRIG, D\_NCR\_DIR, and ACR\_NOT\_ER) are used for destination ACR/ER change notification. x\_EN\_NCR enables the ACR change notification mechanism. x\_NCR\_LO is the low threshold value, and x\_NCR\_HI is the high threshold value. x\_NCR\_TRIG indicates that a notification has been triggered. x\_NCR\_DIR indicates which threshold was crossed last, logic high for HI, logic low for LO. Finally, ACR\_NOT\_ER allows the user to choose between destination ACR or ER change notification. A special status queue entry is written when the following conditions occur:

- x\_NCR\_LO triggers a notification when
  - The new value of ACR<sup>(1)</sup> is less than or equal to NCR\_LO, AND
  - Either this is the first notification OR the last notification was triggered from NCR\_HI.
- x\_NCR\_HI triggers a notification when
  - The new value of ACR<sup>(1)</sup> is greater than or equal to NCR\_HI, AND
  - Either this is the first notification or the last notification was triggered from NCR\_LO.

**NOTE:** <sup>(1)</sup>For the destination change notification, this can be either ACR or ER depending on ACR\_NOT\_ER value.

The special segmentation status queue (see [Section 4.3.6](#)) containing the current ACR value and is indicated by the NCR bit set to a logic 1. The SRC\_NOT\_DEST bit indicates whether the notification is source- or destination-generated. In the case of a destination change notification the current ER value (TA\_ER) is also provided.

### 6.3.7.6 Rate Adjustment in Turnaround RM Cells

#### *Implicit ER Modification*

The system designer has various options for effecting a lowered explicit rate in Turnaround RM cells.

Added to word 10 of the RSM VCC table entry and AAL3/4 head VCC entry are the following: EN\_IMP\_CHG, CONG\_ID. Added to word 9 is ERS\_INDEX.

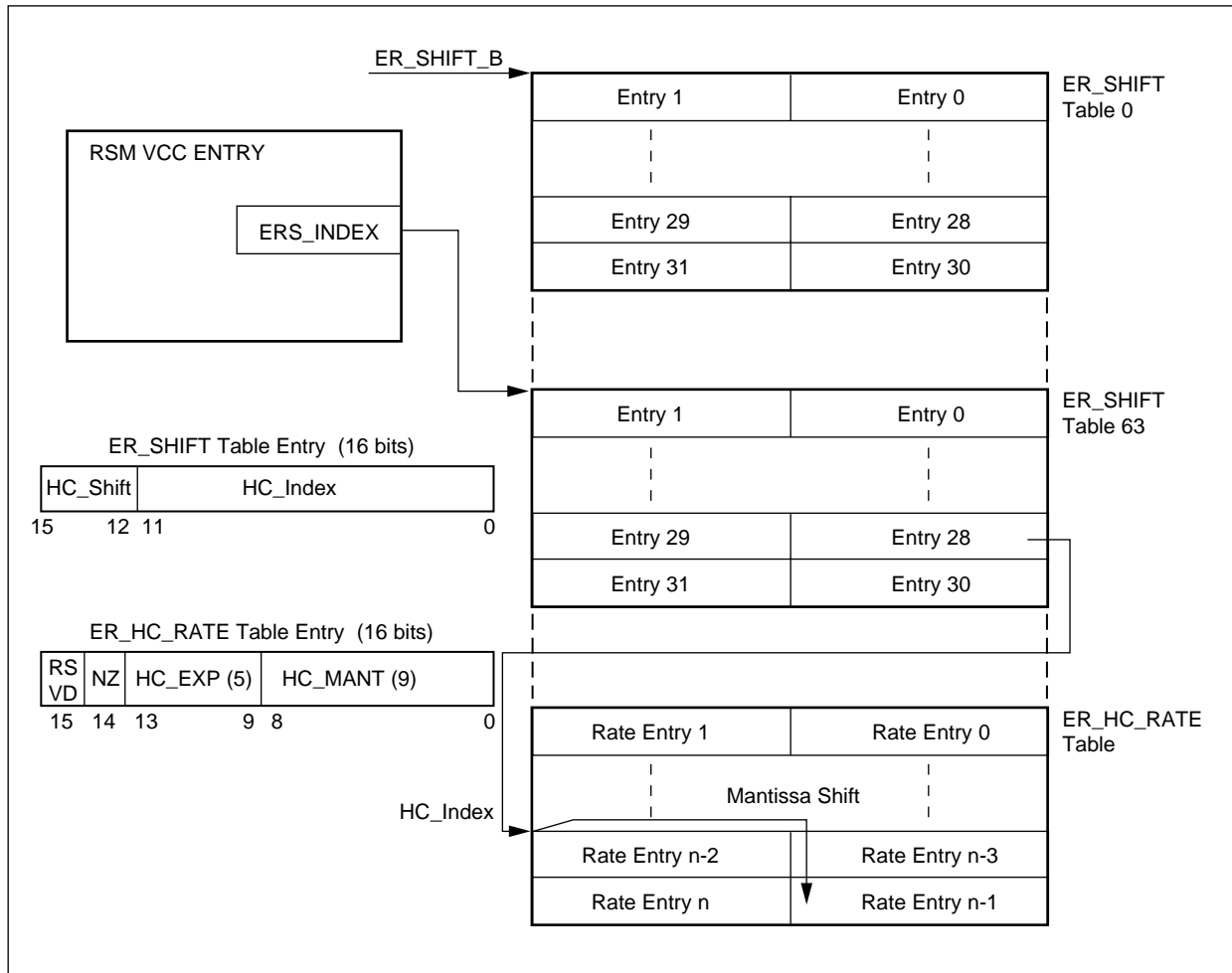
When EN\_IMP\_CNG is a logic high and a forward RM cell is received, SCH\_CNG(CONG\_ID) is checked. If a logic high, the ER field is modified per the mapping mechanism described below and written into TA\_ER field of the SEG VCC table entry. This mechanism provides an efficient per-connection hardware reduction of ER values when turning around RM cells.

The implicit ER reduction mechanism uses a ER\_SHIFT table and the ER\_HC\_RATE table to provide programmable scaled reduction of the ER value based on the value's range. In addition, multiple ER\_SHIFT tables and ER\_HC\_RATE tables may be used to allow different VCCs to utilize different mappings to ER\_HC\_RATE table entries. For example, different mappings may be associated with various ABR templates. A 6-bit index, ERS\_INDEX, located the RS VCC table entry, indicates which ER\_SHIFT table to use. The exponent portion of the ER value is used as an index into the ER\_SHIFT table. Thus, there are 32 entries in each ER\_SHIFT table. Each entry contains two values, the first, HC\_SHIFT, specifies the mantissa shift mask, and the second HC\_INDEX, specifies the address of an entry into the ER\_HC\_RATE table. For HC\_SHIFT values larger than 9, no ER\_HC\_RATE table lookup is performed, and the ER value from the incoming RM cell is written into the TA\_ER field in the SEG VCC table. The number of rates in the ER\_HC\_RATE table for each exponent value is determined by:  $2^{(9 \gg \text{HC\_SHIFT})}$ . The address of an ER\_HC\_RATE table entry is calculated as  $(\text{ER\_SHIFT\_B} \ll 5 + \text{HC\_INDEX}) + \text{RM\_CELL mantissa [8:1]} \gg \text{HC\_SHIFT}$ . The right entry for this address is chosen, if the bit value of the RM\_CELLS mantissa at its bit position HC\_SHIFT is zero, and the left otherwise. The new reduced rate is then

$$2^{\text{HC\_EXP}} \times (1 + \text{HC\_MANT}/512)$$

The maximum index into the rate table has an offset of 4096 words to the ER\_SHIFT\_B base address. Multiple ER\_HC\_RATE tables might be used.

Figure 6-22. ER Reduction Mapping



8236\_053

**Explicit ER Modification**

Added to word 10 of the RSM VCC table entry and AAL3/4 head VCC entry are EN\_EXP\_CNG and EXP\_TA\_ER.

When EX\_EXP\_CNG is a logic high and a forward RM cell is received, the ABR block compares the ER value in the RM cell with EXP\_TA\_ER and writes the lower value to TA\_ER field of the SEG VCC table entry.

**Explicit CI and NI Modification**

Added to word 10 of the RSM VCC table entry and AAL3/4 head VCC table entry are EXP\_TA\_CI and EXP\_TA\_NI.

When EXP\_TA\_CI is a logic high, the value of CI in the turned-around RM cell are asserted. When EXP\_TA\_CI is a logic low the value of CI in the turnaround RM cell is as it has been; that is, it reflects the EFCI field of the last data cell OR'd with the CI value from the FWD RM cell.

When EXP\_TA\_NI is a logic high, the value of NI in the turned around RM cell are logic high; otherwise, the FWD RM cell value of NI is reflected in the turned around BCK RM cell.

#### 6.3.7.7 Optional Rate Adjustment Due to Use-It-or-Lose-It Behavior

The use-it-or-lose-it behavior defined in *TM 4.1* describes a rate adjustment for a VCC that has a high ACR but is not actually using that bandwidth in transmission. In this case, the ACR is lowered to the level of ICR for that channel.

The CN8236's implementation of the *TM 4.1* use-it-or-lose-it behavior allows the system designer to exactly tailor its operation. It uses a simple time-out mechanism, dictatable for each explicit rate.

The designer selectively enables this function for the chosen rate(s) by setting the ACR\_ENA bit to a logic high in each of the corresponding ABR Rate Decision Blocks. As a general guideline, this bit would only be set in those ARDBs dictating rates above ICR.

In those ARDBs which have this function enabled, the designer also specifies two other values: ACR\_TO (the trigger time for the timeout value when the rate adjustment function activates), and ACR\_INDEX (the new lower explicit rate index to be established when the use-it-or-lose-it behavior is triggered).

As a general guideline, ACR\_TO should be patterned on the ACR Decrease Time Factor (ADTF), and ACR\_INDEX should be patterned on the rate index for ICR.

During run time on those ARDBs with the ACR\_ENA bit set, the CN8236 compares RM\_TIME (the schedule slot count at the time the last Forward RM cell was generated) with ACR\_TO. If the trigger point for the time-out has been reached, the CN8236 generates a new Forward RM cell with the ACR\_INDEX value as the assigned new explicit rate.



## 6.3.8 Boundary Conditions and Out-of-Rate RM Cells

**6.3.8.1 Calculated Rate Boundaries** When the system designer pre-calculates the rates for the ARDB tables, those rates must fall within the two obvious upper and lower rate boundaries—the lower rate boundary should not fall below MCR, and the upper rate boundary should not be calculated above PCR.

**6.3.8.2 Out-of-Rate Forward RM Cell Generation** A mechanism must exist to restart scheduling of a VCC once the rate on that channel has dropped to 0, or is below the schedule table minimum rate (that is, the rate is less than one schedule slot on the schedule table). Out-of-rate Forward RM cell generation provides this mechanism.

To globally enable out-of-rate Forward RM cell generation, set SCH\_ABRBASE(OOR\_ENA) to a logic high.

The system designer can set the SET\_OOR bit to a logic high in any ABR Rate Decision Block (ARDB). This enables out-of-rate Forward RM cell generation for that rate.

When the actual cell rate on a channel has lowered to the point where scheduling of the VCC has halted, and SET\_OOR at that rate = 1, the CN8236 sets the SCH\_OOR bit in the VCC's SCH\_STATE to a logic high. The CN8236 then generates an out-of-rate Forward RM cell on that channel.

The CN8236 calculates the rate for generation of out-of-rate Forward RM cells as follows:

$$\begin{aligned} R &= (\text{maximum scheduled rate}) = \text{sysclk}/\text{SLOT\_PER} \text{ generated rate} \\ &= R/(\text{OOR\_INT} + 1)/(\text{VCC\_MAX}/2 + 1) \end{aligned}$$

**6.3.8.3 Out-of-Rate Backward RM Cells** The system designer can set the TA\_OOR bit to a logic high in any ABR Rate Decision Block. This enables the CN8236 sending a Backward RM cell out-of-rate when there is a pending Turnaround RM cell scheduled at that rate but not yet sent, and another Forward RM cell is received.

## 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.

GFC is useful because 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 used 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 CN8236's Implementation of GFC

The CN8236 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 [Section 6.4.2.1](#)), a received HALT indication causes the segmentation coprocessor to halt processing of all channels, both controlled and uncontrolled. This halt condition continues until a cell is received without the HALT indication.

A received SET\_A indication increments the GFC credit counter by one. A GFC-controlled cell can be sent only when the GFC credit counter is equal to 1. Transmission of a GFC-controlled cell decrements the credit counter by 1. Each of the eight transmit priority queues can be configured for GFC control by setting the appropriate GFC<sub>n</sub> 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 are active only when the GFC credit counter is equal to 1.

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

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

#### 6.4.2.1 Configuring the Link for GFC Operation

The following example describes a 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 6-6](#).

**Table 6-6. 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) |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 1-7 <sup>(1)(2)</sup>  | (Reserved for 2 16-bit VBR/ABR pointers) |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>NOTE(S):</b>  |  |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <sup>(1)</sup> Word 1 is present only when the DBL_SLOT field in SEG_CTRL is set and USE_SCH_CTRL is not asserted, meaning two words per schedule slot.                  |  |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <sup>(2)</sup> When USE_SCH_CTRL is asserted, the value of SLOT_DEPTH determines how many additional words are used in each schedule slot; from 0 to 7 additional words. |  |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

### 6.5.2 CBR-Specific Structures

**6.5.2.1 CBR Traffic** A schedule slot is dedicated to a CBR connection by formatting the CBR bit to a logic high and the CBR\_TUN\_ID field in the slot entry as illustrated in [Table 6-7](#).

**Table 6-7. 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** A schedule slot is dedicated to a CBR tunnel by formatting the CBR bit to a logic low and the CBR\_TUN\_ID field of the slot entry as illustrated in [Table 6-8](#). The Schedule Slot field descriptions are detailed in [Table 6-9](#).

**Table 6-8. 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 6-9. 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.  |
| <p><b>NOTE(S):</b> If the user wants only one priority of traffic scheduled in the CBR tunnel, 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, 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, assigns the priorities by level to PRI3 (highest), PRI2 and PRI1 (lowest), making PRI0 the same as PRI1. TUN_PRI0_OFFSET can be used to set PRI0 = PRI1, as needed.</p> |   |

### 6.5.2.3 SCH\_STATE Fields For CBR

This section specifies the SCH\_STATE portion (words 7 through 9) of the Segmentation VCC table entry, when SCH\_MODE = CBR.

The CBR SCH\_STATE is shown in Table 6-10, and the field descriptions are detailed in Table 6-11.

Table 6-10. 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 |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved  |    |          |           |    |    |           |    |   |           |   |   |   |   |   |   |   |   |
| 9    | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    | Reserved |           |    |    |           |    |   |           |   |   |   |   |   |   |   |   |

Table 6-11. CBR SCH\_STATE Field Descriptions

| 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.)        |
| 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 Structure

**6.5.3.1 VBR SCH\_STATE** Tables 6-12 and 6-13 define the SCH\_STATE part of the Segmentation VCC table entry, which consists of words seven through nine for VBR.

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

#### 6.5.3.2 VBR1 or VBR2 Schedule State Table

Table 6-12. 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) |           |             |           | PRESENT    |    |    |    |            |    |    |    |            |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 9    | DELTA2_SIGN   | DELTA2_NZ | DELTA1_SIGN | DELTA1_NZ | DELTA2_EXP |    |    |    | DELTA2_MAN |    |    |    | DELTA1_EXP |    |    |    | DELTA1_MAN |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

Table 6-13. VBR1 and VBR2 SCH\_STATE Field Descriptions

| Field Name  | Description  |
|-------------|--|
| BUCKET2     | Index into Bucket table to give I2 and L2 for SCHED_MODE = VBR2 & VBRC. Bucket table base is given by SEG_BCKB register field. Entries in Bucket table have same format as L1_EXP, L1_MAN, I1_EXP, and I1_MAN. |
| L1_EXP      | GCRA L parameter exponent for bucket 1.<br>L1_EXP must satisfy $L1\_EXP \leq \min(I1\_EXP + 9, 29)$ .<br>L1_EXP that does not satisfy $L1\_EXP \geq I1\_EXP - 9$ will have an effective L1 value of zero.      |
| L1_MAN      | GCRA L parameter mantissa for bucket 1.<br>Bucket 1 L value is: $2^{(L1\_EXP - 10)}(1 + L1\_MAN/512)$ .  |
| I1_EXP      | GCRA I parameter exponent for bucket 1.<br>I1_EXP must satisfy $10 \leq I1\_EXP \leq 25$ .   |
| I1_MAN      | GCRA I parameter mantissa for bucket 1.<br>Bucket 1 I value is: $2^{(I1\_EXP - 10)}(1 + I1\_MAN/512)$ .  |
| PRESENT     | Current Schedule Table position.   |
| DELTA2_SIGN | When set, indicates delta is negative for bucket 2.  |
| DELTA2_NZ   | When set, indicates delta is non-zero for bucket 2.  |
| DELTA1_SIGN | When set, indicates delta is negative for bucket 1.  |
| DELTA1_NZ   | When set, indicates delta is non-zero for bucket 1.  |
| DELTA2_EXP  | Delta exponent for bucket 2.   |

Table 6-13. VBR1 and VBR2 SCH\_STATE Field Descriptions

| Field Name  | Description  |
|---|--|
| DELTA2_MAN  | Delta mantissa for bucket 2.<br>Desired position for bucket 2 is:<br>$\text{DELTA2\_NZ} * 2^{(\text{DELTA2\_EXP} - 10)}(1 + \text{DELTA2\_MAN}/512) + \text{PRESENT.}$ |
| DELTA1_EXP  | Delta exponent for bucket 1.   |
| DELTA1_MAN  | Delta mantissa for bucket 1.<br>Desired position for bucket 1 is:<br>$\text{DELTA1\_NZ} * 2^{(\text{DELTA1\_EXP} - 10)}(1 + \text{DELTA1\_MAN}/512) + \text{PRESENT.}$ |
| <p><b>NOTE(S):</b> The GCRA I value is expressed in cell slot intervals. Example <math>I = \frac{1}{PCR} \times R_{MAX}</math> (I is not expressed in cells/sec.)</p> |  |

- 6.5.3.3 Bucket Table for VBR2 and VBRC** The Bucket table has only 256 entries. [Tables 6-14](#) and [6-15](#) display the entry format and field descriptions for the Bucket table.

Table 6-14. 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    | Reserved |    |    |    | L2_EXP |    |    |    | L2_MAN |    |    |    |    |    | I2_EXP |    |    |    | I2_MAN |    |    |    |   |   |   |   |   |   |   |   |   |   |

Table 6-15. Bucket Table Entry Field Descriptions

| Field Name | Description  |
|------------|--|
| L2_EXP     | GCRA L parameter exponent for bucket 2.<br>L2_EXP must satisfy $L2\_EXP \leq \min(I2\_EXP + 9 \text{ or } 29)$ .<br>L2_EXP that does not satisfy $L2\_EXP \geq I2\_EX - 9$ has an effective L2 value of 0. |
| L2_MAN     | GCRA L parameter mantissa for bucket 2.<br>Bucket 2 L value is $2^{(L2\_EXP - 10)}(1 + L2\_MAN / 512)$ .   |
| I2_EXP     | GCRA I parameter exponent for bucket 2.<br>I2_EXP must satisfy $10 \leq I2\_EXP \leq 25$ .   |
| I2_MAN     | GCRA I parameter mantissa for bucket 2.<br>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 6-16](#) and [Table 6-17](#) detail the SCH\_STATE structure for GFR traffic.

Table 6-16. 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    | Reserved         |    | Reserved |    |    |    |        |    |    |    |    |    |        |    | Reserved |    |        |    | Reserved |    |    |    |   |   |   |   |   |   |   |   |   |   |

Table 6-17. 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 1 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 1 kB in length. Table values are in the form of a non-0 bit, exponent, and mantissa such that MCR_LIMIT is $MCR\_LIM\_NZ \times 2^{(MCR\_LIM\_EXP-10)}(1 + MCR\_LIM\_MAN / 512)$ |
| L1_EXP     | GCRA L parameter exponent for MCR bucket.<br>L1_EXP must satisfy $L1\_EXP \leq \min(I1\_EXP + 9 \text{ or } 29)$ .<br>L1_EXP that does not satisfy $L1\_EXP \geq I1\_EXP - 9$ will have effective L1 value of 0.   |
| L1_MAN     | GCRA L parameter mantissa for MCR bucket<br>Bucket 1 L value is $2^{(L1\_EXP - 10)}(1 + L1\_MAN / 512)$  |
| I1_EXP     | GCRA I parameter exponent for MCR bucket.<br>I1_EXP must satisfy $10 \leq I2\_EXP \leq 25$   |
| I1_MAN     | GCRA I parameter mantissa for MCR bucket.<br>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-0 bit, exponent and mantissa such that MCR\_LIMIT is

$$\text{MCR\_LIM\_NZ} \times 2^{(\text{MCR\_LIM\_EXP} - 10)} (1 + \text{MCR\_LIM\_MAN} / 512)$$

Tables 6-18 and 6-19 describe the MCR Limit Bucket table.

Table 6-18. 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    | Reserved | NON-0 | MCR_EXP |    |    |    | MCR_MAN |    |    |    |    |    |    |    | Reserved | NONZERO | MCR_EXP |    |    |    | MCR_MAN |    |   |   |   |   |   |   |   |   |   |   |

Table 6-19. GFR MCR Bucket Table Entry Field Descriptions

| Field Name | Description  |
|------------|--|
| MCR_EXP    | GFR MCR limit exponent.<br>MCR_EXP must satisfy $\text{MCR\_EXP} \leq 29$ .  |
| MCR_MAN    | GFR MCR limit mantissa.<br>MCR limit is: $\text{NONZERO} \times 2^{(\text{MCR\_EXP}-10)} (1 + \text{MCR\_MAN} / 512)$ .<br>VCC priority is increased by 1 when rate falls below 1/MCR limit. |
| NONZERO    | GFR MCR limit is non-0.  |

### 6.5.5 ABR-Specific Structures

**6.5.5.1 ABR Schedule State Table** Table 6-20 describes the SCH\_STATE entries in the segmentation VCC table for the ABR service class. Table 6-21 details the field descriptions for the ABR SCH\_STATE fields.

**KEY:**

= Values are furnished by the ABR Templates.

Table 6-20. 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  | Reserved   |              | L_EXP      |          |               | L_MAN    |        |        |        |           |             | I_EXP   |         |        | I_MAN      |    |           |    |          |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 8  | Reserved   |              | PRESENT    |          |               |          |        |        |        |           |             |         |         |        |            |    |           |    |          |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 9  | Reserved   | MCR_LIM_NZ   | DELTA_SIGN | DELTA_NZ | MCR_LIM_EXP   |          |        |        |        |           | MCR_LIM_MAN |         |         |        |            |    | DELTA_EXP |    |          | DELTA_MAN |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 10   | EN_SRC_NCR | S_NCR_TRIG   | S_NCR_DIR  | Rsvd     | OOB_PRI       | DCR_STAT |        |        | TM_EXP | BCK_OOR   | FWD_OOR     | SCH_OOR | TA_XMIT | TA_PND | CELL_INDEX |    |           |    |          |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 11   | FWD_INDEX  |              |            |          |               |          |        |        |        |           |             |         |         |        | RM_TIME    |    |           |    |          |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 12   | RATE_INDEX |              |            |          |               |          |        |        |        |           |             |         |         |        | EB_INDEX   |    |           |    |          |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 13   | CRM        |              |            |          |               |          |        |        |        |           |             |         |         |        | UNACK      |    |           |    |          |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 14   | Rsvd       | S_NRC_LO_EXP |            |          | S_NRC_LO_MANT |          |        |        |        |           |             |         |         |        |            |    |           |    | NEXT_OOR |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 15   | MCR_INDEX  |              |            |          |               |          |        |        |        |           |             |         |         |        | ICR_INDEX  |    |           |    |          |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 16 <sup>1</sup>  | TA_ID      |              |            |          | TA_DIR        | TA_BN    | TA_CI  | TA_NI  | TA_RA  | D_NCR_DIR | Rsvd        | TA_ER   |         |        |            |    |           |    |          |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 17 <sup>1</sup>  | TA_CCR     |              |            |          |               |          |        |        |        |           |             |         |         |        | TA_MCR     |    |           |    |          |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 18   | FWD_ID     |              |            |          | FWD_DIR       | FWD_BN   | FWD_CI | FWD_NI | FWD_RA | Reserved  |             |         |         |        |            |    |           |    |          |           |    |    |   | FWD_ER |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 19   | Rsvd       | S_NCR_HI_EXP |            |          | S_NCR_HI_MANT |          |        |        |        |           |             |         |         |        |            |    |           |    | FWD_MCR  |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| <b>NOTE(S):</b>  |            |              |            |          |               |          |        |        |        |           |             |         |         |        |            |    |           |    |          |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |
| <sup>(1)</sup> Words 16 and 17 are written directly by the RSM coprocessor (turnaround information). |            |              |            |          |               |          |        |        |        |           |             |         |         |        |            |    |           |    |          |           |    |    |   |        |   |   |   |   |   |   |   |   |  |  |  |  |  |

Table 6-21. ABR SCH\_STATE Field Descriptions (1 of 3)

| 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-0.<br>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-0 for ER rate.   |
| MCR_LIM_EXP | MCR Limit exponent.<br>MCR_LIM_EXP must satisfy $MCR\_LIM\_EXP \leq 29$ .<br>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 (binary all-1s).   |
| MCR_LIM_MAN | MCR Limit mantissa.<br>MCR_LIMIT is<br>$MCR\_LIM\_NZ \times 2^{(MCR\_LIM\_EXP-10)} (1 + MCR\_LIM\_MAN / 512)$<br>This format for calculating MCR_LIMIT is the same as used with the GCRA / parameter to calculate rates. VCC priority is increased by 1 when rate falls below $1 / MCR\_LIM$ .<br>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 (binary all-1s). |
| DELTA_EXP   | Delta exponent for ER rate.   |
| DELTA_MAN   | Delta mantissa for ER rate.<br>Desired position for ER rate is<br>$DELTA\_NZ \times 2^{(DELTA\_EXP-10)} (1+DELTA\_MAN / 512) + PRESENT$   |
| EN_SRC_NCR  | Enable source ACR change notification processing.   |
| S_NCR_TRIG  | Source ACR change has been triggered. Initialize to logic flow.   |
| S_NCR_DIR   | Indicate direction of source ACR trigger. 1 for HI, 1 for low.  |
| OOR_PRI     | Segmentation priority for out-of-rate RM cells.   |
| DCR_STAT    | Destination ACR change status to be sent. Initialize to logic low.  |
| TM_EXP      | RM_TIME value has expired and is no longer valid.   |
| BCK_OOR     | Backward RM cell has been scheduled out-of-rate.  |
| FWD_OOR     | Forward RM cell has been scheduled out-of-rate.   |
| SCH_OOR     | VCC has halted due to low ACR and has out-of-rate Forward RM cells enabled.   |
| TA_XMIT     | A Backward RM cell has been transmitted after the last Forward RM cell transmitted.   |
| TA_PND      | A Backward RM cell is waiting for transmission.   |
| CELL_INDEX  | ER cell type decision block index for cell type decisions. The cell type decision block is located at byte address $SCH\_ABRB \times 128 + 8 \times CELL\_INDEX$ .  |

Table 6-21. ABR SCH\_STATE Field Descriptions (2 of 3)

| Field Name    | Description   |
|---------------|---|
| 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 0.   |
| S_NCR_LO_EXP  | Source ACR lower threshold, exponent portion.   |
| S_NCR_LO_MANT | Source ACR lower threshold, mantissa portion.   |
| 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 locate 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.   |
| D_NCR_DIR     | Indicates direction of destination ACR trigger. 1 for HI, 0 for low.  |
| 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.  |
| S_NCR_HI_EXP  | Source ACR upper threshold, exponent portion.   |
| S_NCR_HI_MANT | Source ACR upper threshold, mantissa portion.   |

Table 6-21. ABR SCH\_STATE Field Descriptions (3 of 3)

| Field Name | Description   |
|------------|---|
| FWD_MCR    | MCR field for transmitted Forward RM cells. This field is supplied by the user. |

### 6.5.6 ABR Instruction Tables

An ABR cell decision block consists of 16 Cell Type Actions. The correct cell type action is selected with a 4-bit cell type decision vector.

**Table 6-22. ABR Cell Decision Block (ACDB)**

| 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    | CT_ACT7  |    |    |    | CT_ACT6  |    |    |    | CT_ACT5  |    |    |    | CT_ACT4  |    |    |    | CT_ACT3  |    |    |    | CT_ACT2  |    |   |   | CT_ACT1 |   |   |   | CT_ACT0 |   |   |   |
| 1    | CT_ACT15 |    |    |    | CT_ACT14 |    |    |    | CT_ACT13 |    |    |    | CT_ACT12 |    |    |    | CT_ACT11 |    |    |    | CT_ACT10 |    |   |   | CT_ACT9 |   |   |   | CT_ACT8 |   |   |   |

**Table 6-23. ABR Cell Type Actions**

| Value     | Description   |
|-----------|---|
| 0000      | Reserved (no action).   |
| 0001      | Send data cell without reporting I_MAN and I_EXP in status entry. |
| 0010-1000 | Reserved (no action).   |
| 1001      | Send data cell with reporting of I_MAN and I_EXP in status entry. |
| 1010      | Send in-rate Forward RM cell.                                     |
| 1011      | Send in-rate Backward RM cell.                                    |
| 1110-1111 | Reserved (no action).   |

**Table 6-24. ABR Cell Type Decision Vector (ACDV)**

| Bit | Name    | Description                                    |
|-----|---------|--|
| 3   | TRM_EXP | Time since last Forward RM transmitted >= TRM. |
| 2   | TA_PND  | TA_PND bit set in VCC Table entry.             |
| 1   | TA_XMIT | TA_XMIT bit set in VCC Table entry.            |
| 0   | RUN     | RUN bit set in VCC Table entry.                |

Table 6-25. ABR Rate Decision Block (ARDB)

| 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    | ACR_EN    | CONG_EN | TA_OOR | SET_OOR | L_EXP |    |    |    | L_MAN |    |    |    |    |    |    |    | I_EXP     |    |    |    | I_MAN |    |   |   |   |   |   |   |   |   |   |   |
| 1    | CONG_ER   |         |        |         |       |    |    |    |       |    |    |    |    |    |    |    | ACR       |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
| 2    | ACR_INDEX |         |        |         |       |    |    |    |       |    |    |    |    |    |    |    | ACR_TO    |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
| 3    | Rsvd      |         |        |         |       |    |    |    |       |    |    |    |    |    |    |    |           |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
| 4    | NEW_RATE1 |         |        |         |       |    |    |    |       |    |    |    |    |    |    |    | NEW_RATE0 |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
| 5    | NEW_RATE3 |         |        |         |       |    |    |    |       |    |    |    |    |    |    |    | NEW_RATE2 |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
| 6    | NEW_RATE5 |         |        |         |       |    |    |    |       |    |    |    |    |    |    |    | NEW_RATE4 |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
| 7    | NEW_RATE7 |         |        |         |       |    |    |    |       |    |    |    |    |    |    |    | NEW_RATE6 |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |

Table 6-26. ARDB Field Descriptions

| Field Name            | Description  |
|-----------------------|--|
| ACR_EN                | ACR "Use-It-Or-Lose-It" behavior enabled on rate.  |
| CONG_EN               | Enable congestion rate adjustment on rate.   |
| TA_OOR                | Out of rate RM cell turnaround enabled on rate.  |
| SET_OOR               | Rate is below schedule table min rate. Enable out-of-rate Forward RM generation to restart VCC.  |
| 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. I and L parameters are copied to VCC structure.   |
| CONG_ER               | New ER for congestion.   |
| ACR                   | ACR field for transmitted forward RM cell.   |
| ACR_INDEX             | New rate index for ACR lose-it behavior triggered. The new rate decision block is located at byte address SCH_ERB*128 + 32*ACR_INDEX.                                      |
| ACR_TO                | ACR Lose-it behavior trigger time.   |
| NEW_RATE7 - NEW_RATE0 | New rate indexes. The correct new rate index is selected with a 3-bit rate decision vector. The rate decision block is located at byte address SCH_ERB*128 + 32*NEW_RATEn. |



**Table 6-27. ABR Rate Decision Vector (ARDV)**

| Value | Description  |
|-------|--|
| 000   | Reserved/Unused, or<br>Transmit Forward RM cell, ADTF not expired, CRM not expired |
| 001   | Transmit Forward RM cell, ADTF not expired, CRM expired                            |
| 010   | Reserved/Unused, or<br>Transmit Forward RM cell, ADTF expired, CRM not expired     |
| 011   | Reserved/Unused, or<br>Transmit Forward RM cell, ADTF expired, CRM expired         |
| 100   | Received Backward RM cell with CI = 0, NI = 0                                      |
| 101   | Received Backward RM cell with CI = 0, NI = 1                                      |
| 110   | Received Backward RM cell with CI = 1, NI = 0                                      |
| 111   | Received Backward RM cell with CI = 1, NI = 1                                      |

An exponent table maps an explicit rate to a rate decision block index. The table is indexed by the ER exponent.

**Table 6-28. Exponent Table**

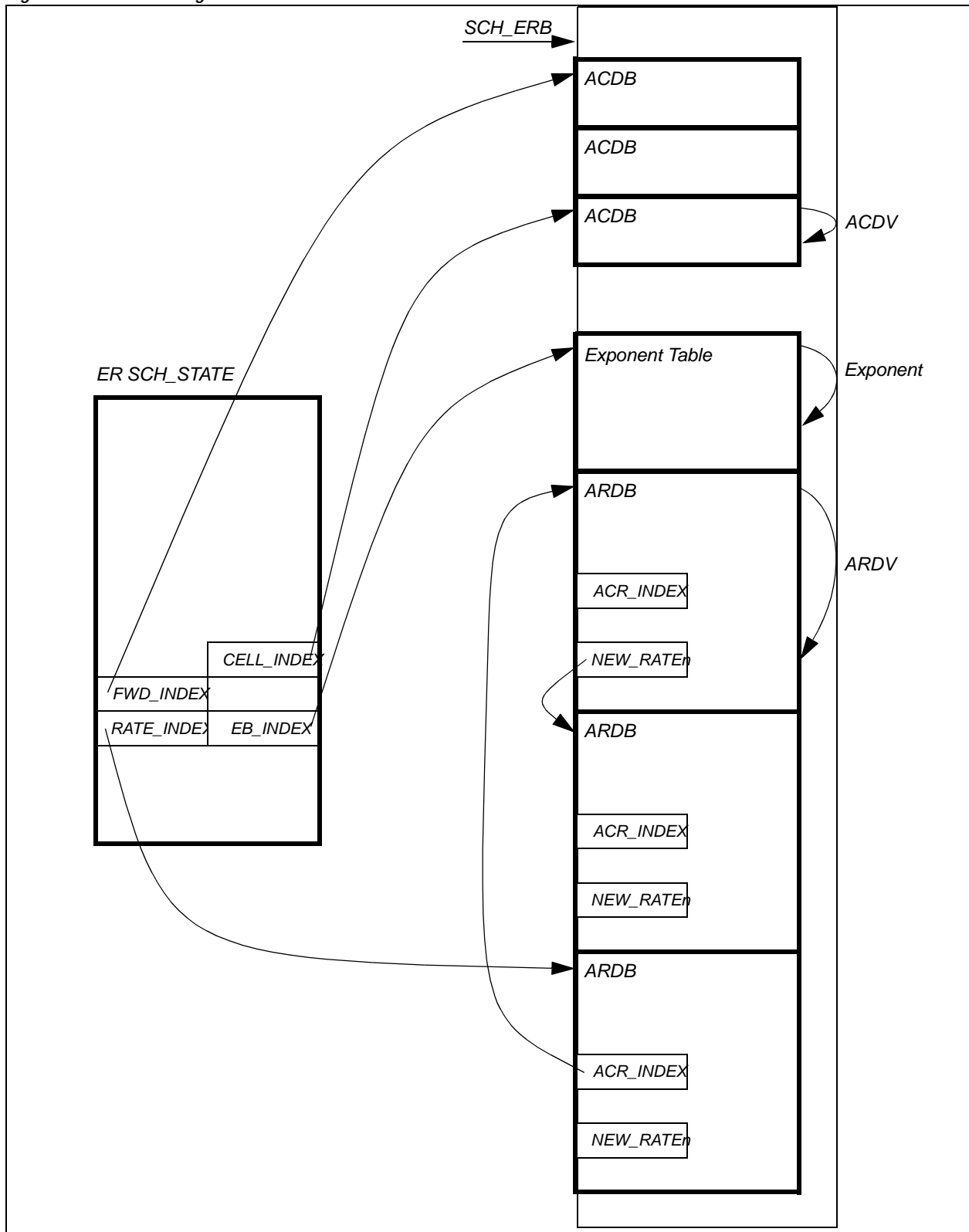
| Word | 31   | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19      | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0    | Rsvd |    |    |    |    |    |    |    |    |    |    |    | SHIFT0  |    |    |    | EXP_BASE0  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| :    | Rsvd |    |    |    |    |    |    |    |    |    |    |    | :       |    |    |    | :          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 31   | Rsvd |    |    |    |    |    |    |    |    |    |    |    | SHIFT31 |    |    |    | EXP_BASE31 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

**Table 6-29. Exponent Table Field Descriptions**

| Field Name           | Description  |
|----------------------|--|
| EXP_BASE31-EXP_BASE0 | Base rate decision block index.  |
| SHIFT31-SHIFT0       | Shift of ER mantissa. Rate index is EXP_BASEn + mantissa>>SHIFTn.<br>If the ER NZ bit is not set, the rate index is EXP_BASE0. |

Figure 6-23 illustrates the linkage pattern between the components of the ABR Instruction Tables.

Figure 6-23. ABR Linkage



## 6.5.7 RS\_QUEUE

**Table 6-30. RS\_QUEUE Entry—OAM-PM Reporting Information Ready for Transmission**

| 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    | 0000<br>OAM_PM |    |    | Reserved |    |    |    | BLER |    |    |    | SEG_VCC_INDEX |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 1    | BCK_TUC0       |    |    |          |    |    |    |      |    |    |    |               |    |    |    |    | BCK_TUC01 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 2    | TRCC0          |    |    |          |    |    |    |      |    |    |    |               |    |    |    |    | TRCC0+1   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 3    | Reserved       |    |    |          |    |    |    |      |    |    |    |               |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

**Table 6-31. RS\_QUEUE Entry—Forward ER RM Cell Received**

| 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    | 0100<br>ER_FWD |    |    | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SEG_VCC_INDEX |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |
| 1    | Reserved       |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |

**Table 6-32. RS\_QUEUE Entry—Backward ER RM Cell Received**

| 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    | 0101<br>ER_BCK |    |    | Reserved |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    | SEG_VCC_INDEX |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |
| 1    | Reserved       |    |    |          |    |    |    |    |    |    | BN | CI | NI | Reserved |    |    |    | ER |    |               |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |

**Table 6-33. RS\_QUEUE Field Descriptions**

| Field Name    | Description   |
|---------------|---|
| OAM_PM        | OAM PM backward reporting information ready for transmission.                           |
| BLER          | Block Error Result.   |
| SEG_VCC_INDEX | Segmentation VCC index for backward reporting OAM PM cell.                              |
| BCK_TUC0      | TUC0 field in received forward monitoring cell. Written directly from Forward_RM cell.  |
| BCK_TUC01     | TUC01 field in received forward monitoring cell. Written directly from Forward_RM cell. |
| TRCC0         | Total Received Cell Count with CLP = 0.   |
| TRCC0+1       | Total Received Cell Count with CLP = 0+1.   |
| ER_FWD        | Forward ER RM cell received.  |
| SEG_VCC_INDEX | Segmentation VCC index for transmitted Backward ER RM cell.                             |
| ER_BCK        | Backward ER RM cell received.   |
| BN            | Backward Explicit Congestion Notification bit from received RM cell.                    |
| CI            | Congestion Indication bit from received RM cell.  |
| NI            | No Increase bit from received RM cell.  |
| ER            | Explicit Cell Rate field from received RM cell.   |

### 6.5.8 Scheduler Internal SRAM Registers

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

Application program should ignore these registers.

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

Figure 6-24 shows the layout for the head and tail pointers. Table 6-34 describes the memory map of these registers.

Figure 6-24. Head and Tail Pointers

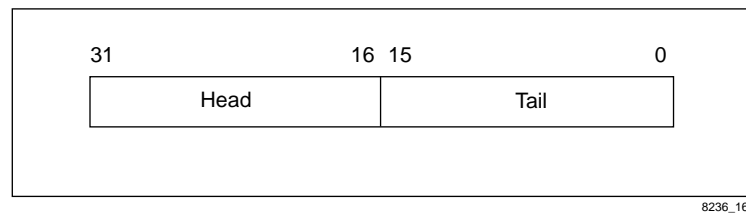


Table 6-34. Scheduler Internal SRAM Memory Map (Head/Tail Pointers)

| 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.  |
| 0x1660–0x1663 | PRI_PNTR8  | Global priority 8.  |
| 0x1664–0x1667 | PRI_PNTR9  | Global priority 9.  |
| 0x1668–0x166B | PRI_PNTR10 | Global priority 10. |
| 0x166C–0x166F | PRI_PNTR11 | Global priority 11. |
| 0x1670–0x1673 | PRI_PNTR12 | Global priority 12. |
| 0x1674–0x1677 | PRI_PNTR13 | Global priority 13. |
| 0x1678–0x167B | PRI_PNTR14 | Global priority 14. |
| 0x167C–0x167F | PRI_PNTR15 | Global priority 15. |
| 0x1680–0x17FF | Reserved   | Not implemented.    |





# 7.0 OAM Functions

---

## 7.1 OAM Overview

OAM cells are ATM Layer management messages. These are generated by the host on the segmentation side of the CN8236, and are detected and either monitored or processed on the reassembly side of the CN8236.

ATM's OAM capabilities differentiate it from other less robustly managed communication technologies. The CN8236 provides internal support for the detection and generation of OAM traffic, including Performance Monitoring (PM) OAM.

The CN8236 supports the F4 and F5 OAM flows according to *ITU-T Recommendation I.610*. It also monitors the performance of up to 128 channels, generating PM-OAM cells according to the same specification.

### 7.1.1 OAM Functions Supported

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

[Table 7-1](#) lists the OAM functions of the F4 and F5 OAM flows, as defined in *Recommendation I.610*.

*Table 7-1. OAM Functions of the ATM Layer*

| OAM Function                   | Main Application  |
|--------------------------------|---|
| AIS (Alarm Indication Signal)  | Reports defect indications in the forward direction.  |
| RDI (Remote Defect Indication) | Reports remote defect indications in the backward direction.  |
| CC (Continuity Check)          | Continuously monitors the continuity of a connection. A continuity cell is used periodically to check whether a connection is idle or has failed. |
| Loopback                       | Used for<br>on-demand connectivity monitoring<br>fault localization<br>pre-service connectivity verification                                      |
| PM (Performance Monitoring)    | Estimates performance and reports performance estimations in the backward direction.  |
| Activation/Deactivation        | Activating/deactivating performance monitoring and continuity check.  |
| System Management              | Used by end-systems only.   |

The CN8236 internally supports the following functions as described in *ITU-T Recommendation I.610*:

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

Implementation of the full range of functions in processing OAM cells are 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.

There are two kinds of F4 OAM flows which can simultaneously exist in a VPC:

- End-to-end F4 flow (used for end-to-end VPC operations communications)
- Segment F4 flow (used for communicating operations information within the bounds of one VPC link, such segment having its source and sink defined by the user)

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

F4 flow OAM cells are identified as F4 flow cells by a pre-assigned VCI value of three (segment flow cell) or four (end-to-end flow cell) as shown in [Table 7-2](#). The same pre-assigned VCI value is used for both directions of the F4 flow.

*Table 7-2. VCI Values for F4 OAM Flows*

| VCI Value | Interpretation              |
|-----------|-----------------------------|
| 3         | Segment OAM F4 flow cell    |
| 4         | End-to-end OAM F4 flow cell |

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

There are two kinds of F5 OAM flows which can simultaneously exist in a VCC:

- End-to-end F5 flow (used for end-to-end VCC operations communications)
- Segment F5 flow (used for communicating operations information within the bounds of one VCC link, such segment having its source and sink defined by the user)

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 PTI code value of 100 (segment flow cell) or 101 (end-to-end flow cell) and shown in [Table 7-3](#). The same pre-assigned PTI value is used for both directions of the F5 flow.

*Table 7-3. PTI Values for F5 OAM Flows*

| PTI Code | Interpretation              |
|----------|-----------------------------|
| 100      | Segment OAM F5 flow cell    |
| 101      | End-to-end OAM F5 flow cell |

### 7.1.2.3 Performance Monitoring (PM)

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

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

The CN8236 performs PM on up to 128 user-assigned channels. The SAR enables PM for a channel by setting the PM enable bits (PM\_EN) in the Segmentation and Reassembly VCC State table entries for that channel.

The CN8236 performs PM processing on any channel by monitoring blocks of user cells on that channel. The size of this block of cells is set by the user, and can have the value (N) of 128, 256, 512, or 1024 cells. The CN8236 inserts a PM cell after every N user cells on that channel. A block size of 0 is valid when the SAR is a destination point ONLY for PM processing. In this case, the segmentation coprocessor only generates Backward reporting cells in response to reassembly performance monitoring calculations.

PM as performed by the CN8236 calculates the following:

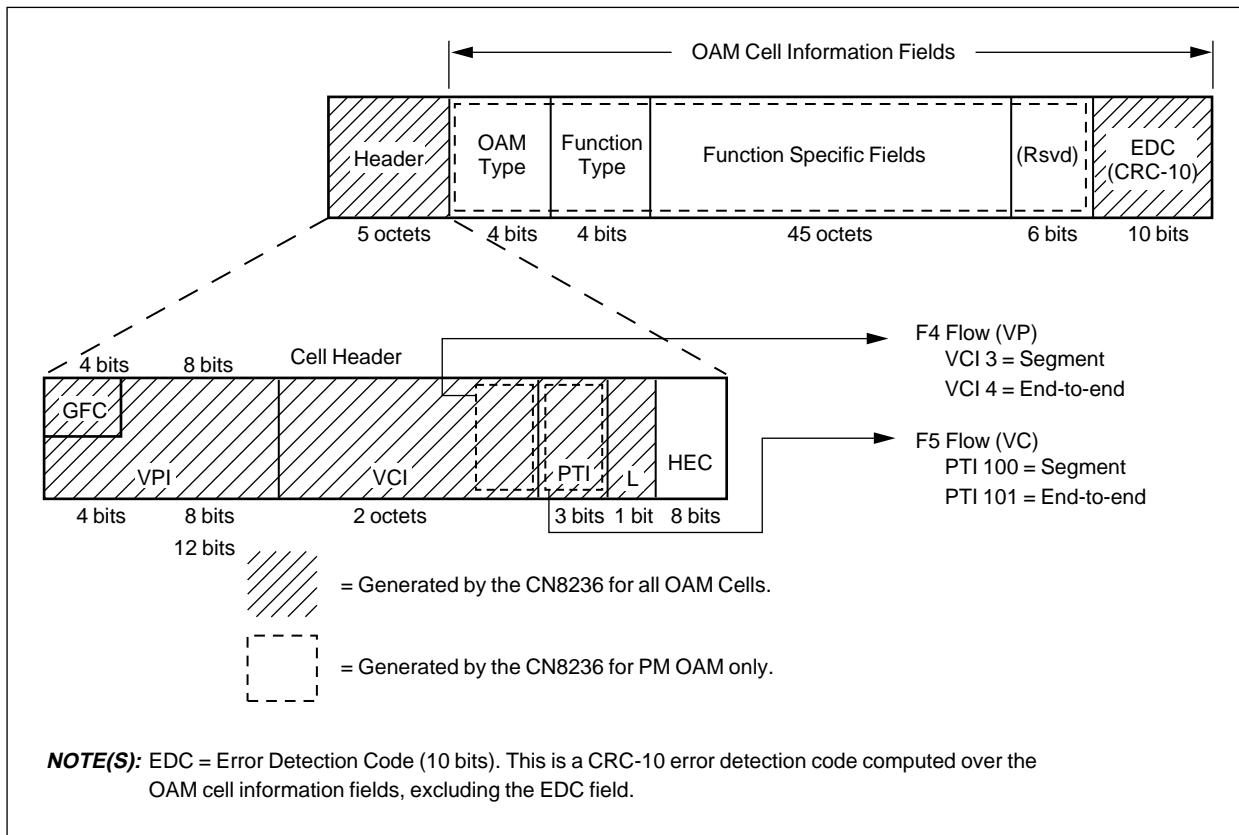
- Errored blocks (by means of a BIP-16 error detection code generated over the payloads of the user cells in the PM block)
- A count of mis-inserted PM forward monitoring cells

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

### 7.1.3 OAM Cell Format

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

Figure 7-1. OAM Cell Format



8236\_110

The OAM Type and Function Type identifiers as specified by I.610, are listed in [Table 7-4](#).

**Table 7-4. OAM Type and Function Type Identifiers**

| OAM Type <sup>(1)</sup>  | Coding | Function Type <sup>(2)</sup>               | Coding                       |
|--|--------|--|------------------------------|
| Fault Management   | 0001   | AIS<br>RDI<br>Continuity Check<br>Loopback | 0000<br>0001<br>0100<br>1000 |
| Performance Management   | 0010   | Forward monitoring<br>Backward reporting   | 0000<br>0001                 |
| Activation/Deactivation  | 1000   | Performance Monitoring<br>Continuity Check | 0000<br>0001                 |
| <p><b>NOTE(S):</b><br/>           (1) OAM Type indicates the type of management function performed by this cell (for example, fault management, performance management, etc.).<br/>           (2) Function Type indicates the actual function performed by this cell within the management type indicated by the OAM Type field.</p> |        |  |                              |

### 7.1.4 Local vs. Host Processing of OAM

ATM carries with it significant network management overhead. The CN8236 supports those robust OAM features described previously. Due to the breadth of ATM applications and the continuing evolution of ATM management standards, it is essential that a range of flexibility be provided in the processing of network management overhead. To provide this flexibility, the CN8236 includes an optional local processor interface.

Since the CN8236 is capable of SAR-shared memory segmentation and reassembly, it can route OAM traffic including PM traffic to and from this local processor, thereby off-loading ATM network management from the host. Thus, host processing power is focused on the user applications specifically concerned with processing ATM user data traffic.

To accomplish this, the CN8236 provides global OAM buffer and status queues (OAM\_BFR\_QU and OAM\_STAT\_QU, addresses for both assigned in the RSM\_CTRL1 register). If the OAM\_QU\_EN bit in the RSM\_CTRL1 register is set to a logic high, the CN8236 routes OAM traffic to these global queues. With SAR-shared memory addresses assigned to these global queues, the CN8236 processes OAM traffic through the local processor, thereby freeing the host from these management functions. In addition, the global OAM segmentation status queue, SEG\_CTRL(OAM\_STAT\_ID), is used if the OAM\_STAT bit in the segmentation buffer descriptor is a logic high.

## 7.2 Segmentation of OAM Cells

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

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

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

### 7.2.1 Key OAM-Related Fields for OAM Segmentation

- 7.2.1.1 Segmentation Buffer Descriptors** Several fields in the SBD entry 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 CN8236 now reports 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 1. This activates the WR\_PTI and WR\_VCI bits in the buffer descriptor, which signal the CN8236 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 CN8236.
  - 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 0.
  - 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 CN8236 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 CN8236 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 CN8236 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 CN8236 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

- |   |  |
|---|--|
| <b>7.3.1.1 Reassembly VCC State Table</b> | The reassembly VCC State table field, SEG_VCC_INDEX, should be written to point to the channel index of the corresponding segmentation channel. This is necessary for PM-OAM and ABR channels.   |
| <b>7.3.1.2 Reassembly Status Queue</b>    | The 3-bit OAM field in the RSM status queue entry must be set to the value indicated in the RSM status queue structure description for that field. A non-0 value in the OAM field indicates that the cell is an OAM cell.<br>If the CRC-10 error detection code computation on the OAM cell shows an error, the CN8236 sets the CRC_ERROR bit in the status queue entry to a logic high. |
| <b>7.3.1.3 F4 Flow</b>                    | For F4 flow operation, a separate VCC table entry must be configured.  |

### 7.3.2 OAM Reassembly Operation

Received OAM traffic should be detected and routed to the global OAM buffer and status queues (OAM\_BFR\_QU and OAM\_STAT\_QU).

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

Once an OAM cell is detected, the RSM coprocessor checks the cell to determine whether it is a PM cell by seeing if the OAM\_TYPE field in the cell is set to value 0010. That PM detection is only performed on F4 and F5 OAM cells.

If the RSM coprocessor finds the cell is a PM cell, it is processed following the guidelines described in [Section 7.4](#).

*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 each received OAM cell's information fields. If an error is detected (for example, the computed CRC value does not match the CRC-10 value written in the cell), the SAR sets the CRC\_ERROR bit in the status queue entry to a logic high.

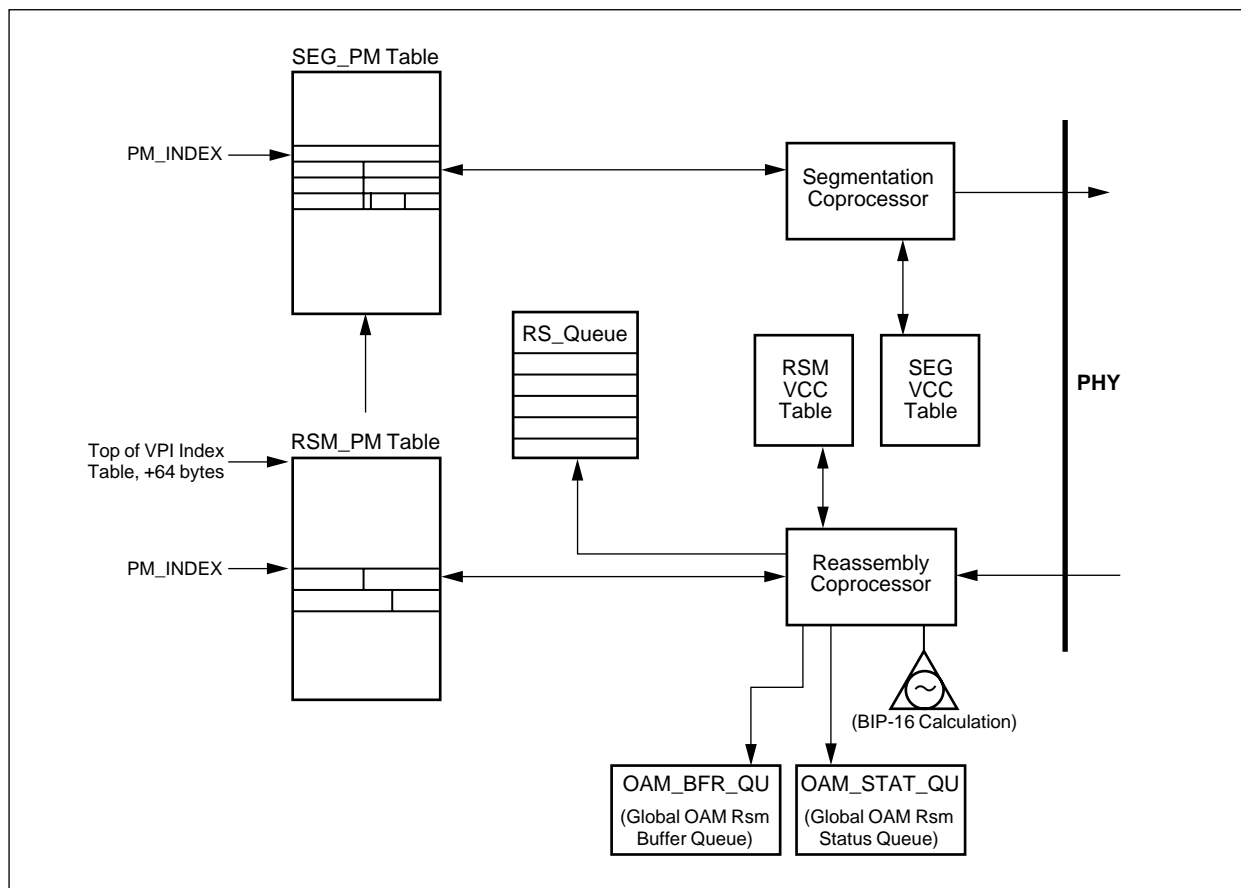


## 7.4 PM Processing

OAM Performance Monitoring can be enabled for up to 128 VCCs by setting the PM\_EN bits in the RSM VCC table entry and the SEG VCC table entry for that channel.

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

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



8236\_054

For any channel on which PM processing is enabled, the CN8236 performs these functions:

- The RSM coprocessor reassembles received PM-OAM cells in the global reassembly OAM buffer queue (OAM\_BFR\_QU).
- A reassembly status record is written to the global reassembly OAM status queue for each received PM-OAM cell. The status entry for a forward monitoring PM-OAM cell includes the Block Error Result (BIPV) calculation, and both Total Received Cell Counts (TRCC0 and TRCC0+1). The two Total User Cell number fields (TUC0 and TUC0+1) can be extracted directly from the cell payload.
- The RSM coprocessor performs the BIP-16 calculation on each received data cell in the defined PM block and writes this data to the RSM\_PM table entry set aside for that PM\_INDEX.
- For each forward monitoring PM cell received, the RSM coprocessor writes an entry for a backward reporting PM cell in the RS\_Queue, causing the CN8236 to generate and transmit a backward reporting PM cell.
- The segmentation coprocessor automatically generates a forward monitoring PM cell at the end of each PM block. It gets the data for these cells from the SEG\_PM table entry for that PM\_INDEX. This can be optionally disabled by setting the FWD\_MON field equal to 0.

### 7.4.1 Initializing PM Operation

The user must initialize the fields described in [Table 7-5](#) before starting PM processing.

**Table 7-5. PM-OAM Field Initialization For Any PM\_INDEX**

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

## 7.4.2 Setting Up Channels for PM Operation

When the CN8236 receives a PM activation cell (OAM Type = 1000, Function Type = 0000), or when the user decides to activate PM processing on a channel, the host (or local) processor must enable PM on the applicable channel by setting the PM\_EN bits in the RSM and SEG VCC table entries to logic high and selecting an unused PM\_INDEX (0-127).

The host then initializes the corresponding SEG\_PM and RSM\_PM table entries. The format of each entry of the SEG\_PM table is illustrated in [Table 7-6](#), while the format of each entry of the RSM\_PM table is illustrated in [Table 7-8](#).

The assigned PM\_INDEX value for that channel has to be written to both the RSM VCC table entry and the SEG VCC table entry for that channel.

For F4 flows, each VCI channel in the VPI group must have the PM\_EN bits in both the RSM VCC table entry and the SEG VCC table entry set high, and the PM\_INDEX pointing to the same location. In addition, a RSM and SEG VCC table entry must be configured corresponding to VCI = 3 or VCI = 4.

To initialize backward reporting without forward monitoring on any channel, set FWD\_MON = 0.

Once the SEG\_PM and RSM\_PM table entries have been initialized, the processor sends an activation confirmed OAM cell to the originator.

## 7.4.3 PM Operation

PM processing operates automatically on any channel until stopped by clearing the PM\_EN fields in the SEG VCC table entry and the RSM VCC table entry.

PM-OAM cells are not included in the NRM cell count, as part of ER processing. See [Chapter 6.0](#), for details.

### 7.4.3.1 Generation of Forward Monitoring PM Cells

The segmentation coprocessor generates a forward monitoring PM cell at the end of each PM block, as defined by the BLOCK\_SIZE field in the SEG\_PM table for any PM\_INDEX. It determines the point to generate a forward monitoring cell by following these processes:

- At the point of initialization of PM processing or when a forward monitoring cell is sent, the SEG coprocessor sets the BLOCK\_COUNT field to 0. It also increments Monitoring Cell Sequence Number (MSN), and re-initializes the BIP field to 0.
- As each data cell is segmented, the SEG coprocessor increments the BLOCK\_COUNT number and updates the BIP field.
- When the BLOCK\_COUNT number reaches the block size specified by the BLOCK\_SIZE field, signifying the end of the PM block, the SEG coprocessor generates a new forward monitoring PM cell and starts these processes again.

#### 7.4.3.2 Reassembly of Forward Monitoring PM Cells

When the CN8236 receives a forward monitoring PM cell, the RSM coprocessor reads the RSM\_PM table word pointed to by the PM\_INDEX field in the RSM VCC table entry. The location of the RSM\_PM table is above the LECID table. The BIPV, TRCC0, and TRCC0+1 fields are written to a special RSM-PM forward monitoring status queue entry. The TUC0 and TUC0+1 fields can be extracted directly from the RSM\_PM cell payload.

When a new buffer is needed, the reassembly coprocessor uses the global OAM buffer queue if RSM\_CTRL1(OAM\_QU\_EN) is a logic high. Otherwise, it uses the BFR0 pool identification number in the RSM VCC table to point to the appropriate free buffer queue.

A RSM status entry is written for each OAM cell reassembled.

The reassembly coprocessor uses the global OAM status queue if the RSM\_CTRL1(OAM\_QU\_EN) bit is a logic high. Otherwise, it uses the STAT field in the RSM VCC table to determine which status queue to use for that channel.

#### 7.4.3.3 Reassembly of Backward Reporting PM Cells

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

#### 7.4.3.4 Turnaround and Segmentation of Backward Reporting PM Cells

For each forward monitoring cell received, the CN8236 also writes the BIPV, TRCC0, TRCC0+1, TUC0, and TUC0+1 fields to the RS\_Queue for further processing by the segmentation coprocessor. The segmentation coprocessor generates a backward reporting cell.

#### 7.4.3.5 Turnaround of Backward Reporting PM Cells ONLY

To enable turnaround of backward reporting PM cells without generation of forward monitoring PM cells, set the segmentation PM\_EN bit in the SEG VCC table entry to a logic high, set the PM\_INDEX, and set the FWD\_MON field to 0.

### 7.4.4 Error Conditions During PM Processing

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

If the RS\_Queue fills, the PM-OAM information is dropped, and the RS\_QUEUE\_FULL status indication is set.

### 7.4.5 PASS\_OAM Function

A PASS\_OAM only function is activated by setting bit 22 of word 0 in the RSM VCC table. When active, OAM cells are processed, and data cells are discarded without incrementing any discard counters.

## 7.5 OAM Control and Status Structures

Refer to [Section 4.3](#) and [Section 5.7](#) for information on VCC Tables, buffer descriptors, the transmit queue, and status queues, as they apply to generating and processing OAM cells.

The base address of the SEG\_PM table is given by the SEG\_PMB field in the SEG\_PMBASE register. The address of each entry is located at byte address,  $SEG\_PMB \times 128 + (PM\_INDEX) \times 32$ .

The RSM\_PM table is located above the LECID table, which is located above the VPI table. The address of each entry is located at byte address, if  $RSM\_CTRL0(VPI\_MASK)$   
 $RSM\_ITB \times 128 + 1024 + 64 + (PM\_INDEX) \times 16$   
 else  
 $RSM\_ITB \times 128 + 16384 + 64 + (PM\_INDEX) \times 16$

### 7.5.1 SEG\_PM Structure

[Tables 7-6](#) and [7-7](#) describe the structure and field definitions of the SEG\_PM table.

Table 7-6. SEG\_PM Structure

| Word | 31         | 30         | 29      | 28       | 27          | 26 | 25 | 24 | 23 | 22 | 21 | 20   | 19 | 18 | 17 | 16 | 15        | 14 | 13 | 12      | 11  | 10 | 9 | 8 | 7 | 6 | 5 | 4       | 3 | 2 | 1 | 0 |  |  |  |  |
|------|------------|------------|---------|----------|-------------|----|----|----|----|----|----|------|----|----|----|----|-----------|----|----|---------|-----|----|---|---|---|---|---|---------|---|---|---|---|--|--|--|--|
| 0    | ATM_HEADER |            |         |          |             |    |    |    |    |    |    |      |    |    |    |    |           |    |    |         |     |    |   |   |   |   |   |         |   |   |   |   |  |  |  |  |
| 1    | FWD_TUC0   |            |         |          |             |    |    |    |    |    |    |      |    |    |    |    | FWD_TUC01 |    |    |         |     |    |   |   |   |   |   |         |   |   |   |   |  |  |  |  |
| 2    | FWD_PM     | BLOCK_SIZE | FWD_MON | Rsvd     | BLOCK_COUNT |    |    |    |    |    |    |      |    |    |    |    |           |    |    |         | BIP |    |   |   |   |   |   |         |   |   |   |   |  |  |  |  |
| 3    |            |            |         | Reserved |             |    |    |    |    |    |    | BLER |    |    |    |    |           |    |    | BCK_MSN |     |    |   |   |   |   |   | FWD_MSN |   |   |   |   |  |  |  |  |
| 4    | BCK_TUC0   |            |         |          |             |    |    |    |    |    |    |      |    |    |    |    | BCK_TUC01 |    |    |         |     |    |   |   |   |   |   |         |   |   |   |   |  |  |  |  |
| 5    | TRCC0      |            |         |          |             |    |    |    |    |    |    |      |    |    |    |    | TRCC01    |    |    |         |     |    |   |   |   |   |   |         |   |   |   |   |  |  |  |  |
| 6    | Reserved   |            |         |          |             |    |    |    |    |    |    |      |    |    |    |    |           |    |    |         |     |    |   |   |   |   |   |         |   |   |   |   |  |  |  |  |
| 7    | Reserved   |            |         |          |             |    |    |    |    |    |    |      |    |    |    |    |           |    |    |         |     |    |   |   |   |   |   |         |   |   |   |   |  |  |  |  |

Table 7-7. SEG\_PM Field Descriptions

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

## 7.5.2 RSM\_PM Table

Tables 7-8 and 7-9 describe the structure and definitions of the RSM\_PM table.

**Table 7-8. RSM\_PM Table Entry**

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

**Table 7-9. RSM\_PM Table Field Descriptions**

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





## 8.0 DMA Coprocessor

---

### 8.1 Overview

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

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

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

### 8.2 DMA Read

For outgoing messages, DMA read cycles move data from host memory to the segmentation coprocessor using a gather DMA method. The maximum burst size is thirteen 32-bit words, which 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

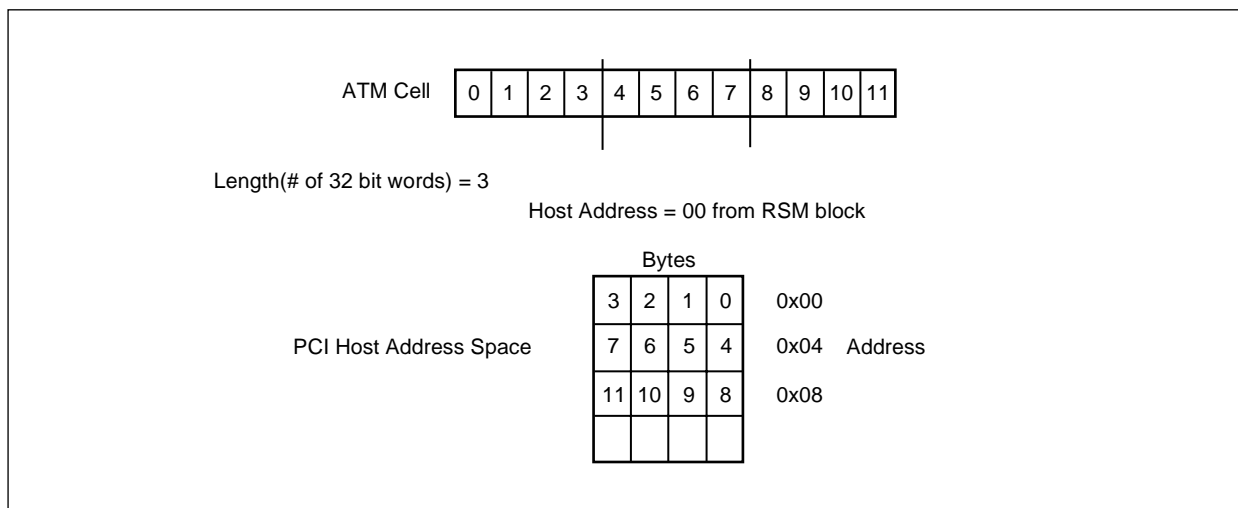
For incoming messages, DMA write cycles move data from the reassembly coprocessor to host memory using a scatter DMA method. The maximum burst size is fourteen 32-bit words, which correspond to one ATM cell and a status word appended to PM cells. The burst size can be reduced by setting the MAX\_BURST\_LEN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

## 8.4 Misaligned Transfers

The reassembly and segmentation coprocessors handle data internally on word addresses. The DMA coprocessor must be capable of handling transfers from the PCI bus without the same constraint, that is, 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 CN8236. When the CN8236 specifies a host address with the Least Significant Bits (LSBs) = 00, it is implied that the data is byte aligned. [Figure 8-1](#) illustrates 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].

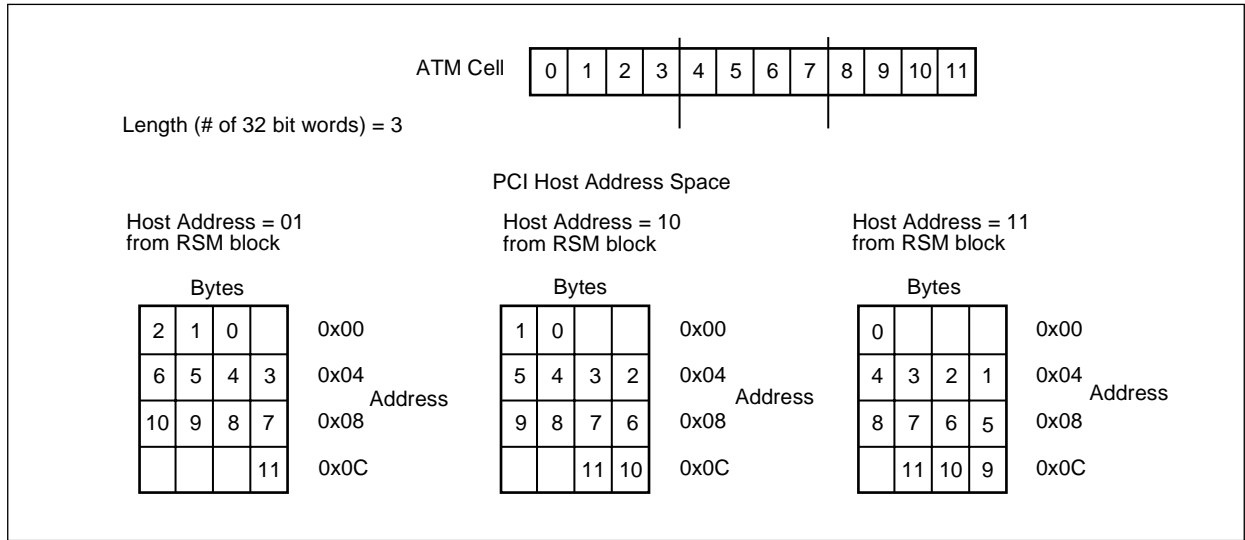
Figure 8-1. Little Endian Aligned Transfer



8236\_055

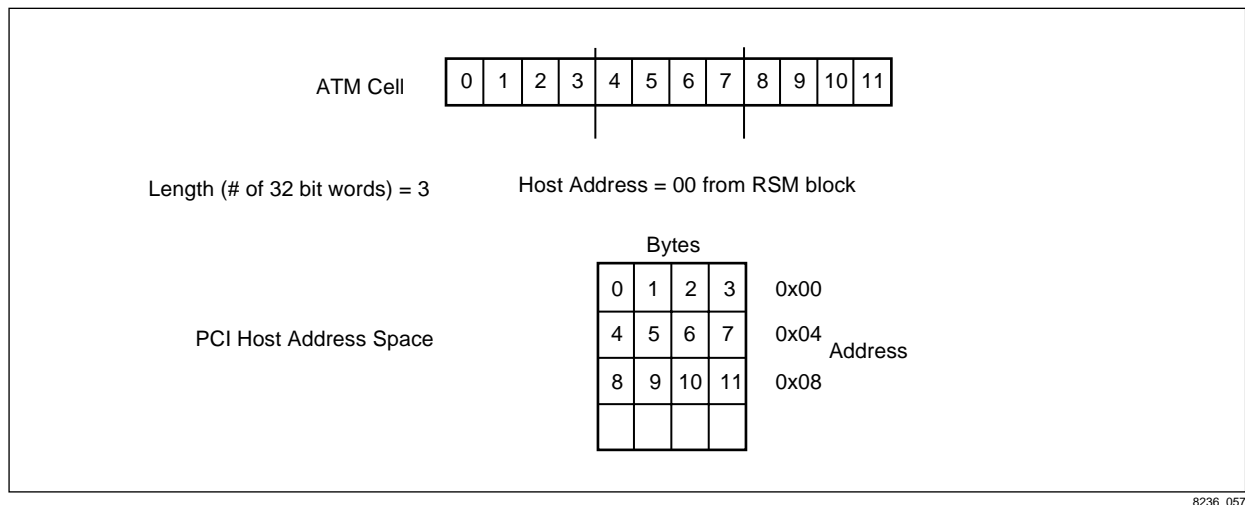
When the CN8236 specifies a host address with the LSBs not equal to 00, it is implied that the data is misaligned. [Figure 8-2](#) illustrates how a misaligned address would map into the PCI host address space for a little endian system.

Figure 8-2. Little Endian Misaligned Transfer



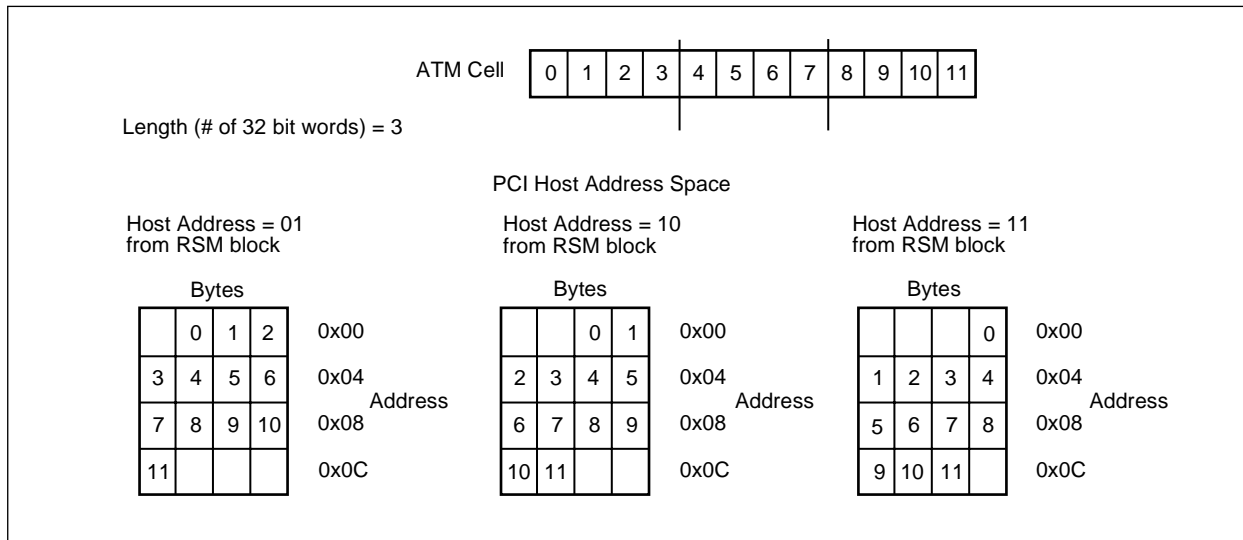
[Figure 8-3](#) illustrates how a byte-aligned address would map into the PCI host address space for a big endian system.

Figure 8-3. Big Endian Aligned Transfer



When the CN8236 specifies a host address with the LSBs not equal to 00, it is implied that the data is not aligned. Figure 8-4 illustrates 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\* causes these bits to be a logic low.

## 9.0 Local Memory Interface

---

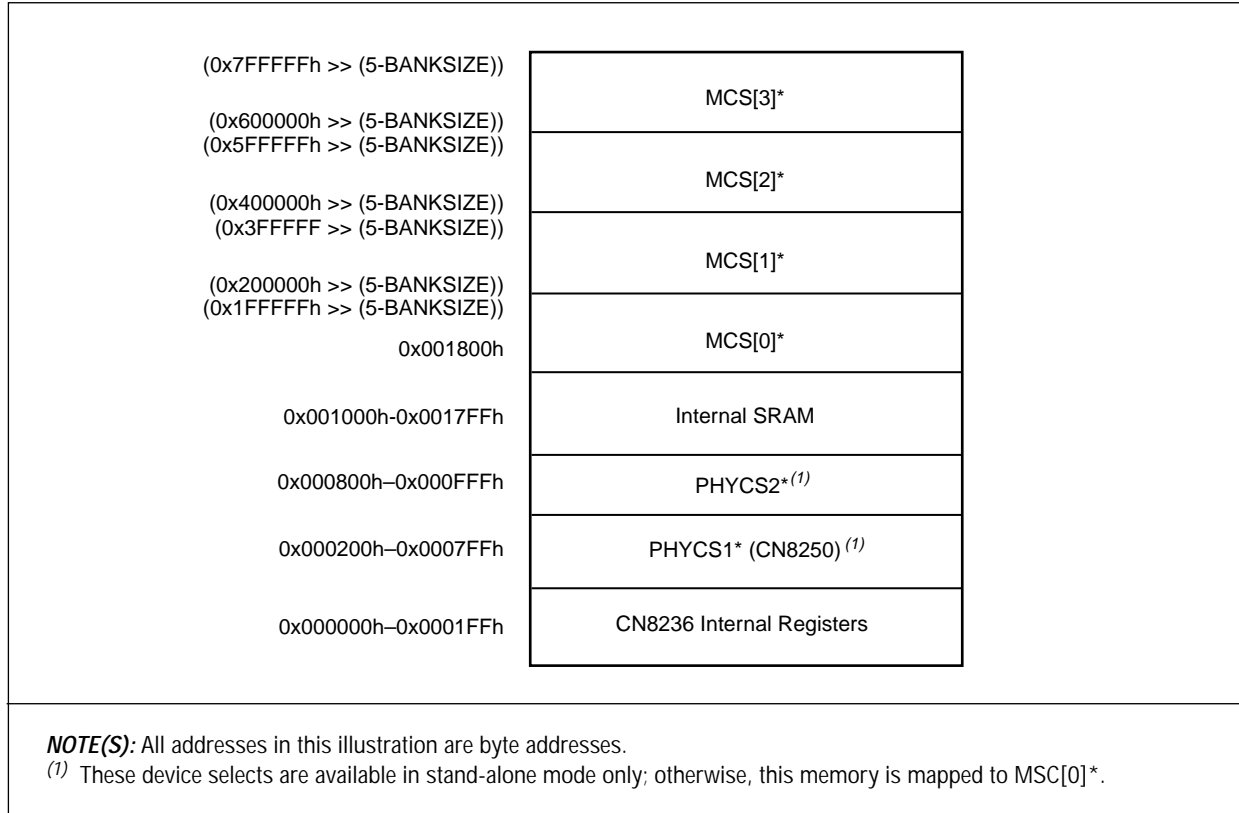
### *9.1 Overview*

To simplify system implementations, the CN8236 integrates a complete memory controller, designed for direct interface to common SRAMs. The control and status registers and 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 8 MB of external memory using SRAM devices can be accessed by the CN8236. The amount of memory required is heavily dependent on the number of VCCs implemented and the number of VCCs that are currently active.

Figure 9-1 provides the CN8236 address map.

Figure 9-1. CN8236 Memory Map



8236\_059

## 9.2 Memory Bank Characteristics

The external memory is organized in one to four banks of up to 2 MB each. The system can use any number of banks to fulfill the memory requirements; the only stipulation is that the banks must be of the same size and organization. The local processor and host processor select between the banks using the PBSEL[1:0] inputs. PBSEL[1:0] is sourced by local processor Host processor sources PCI Address Bits. BANKSIZE[2:0] (bits 23–21) in the CONFIG0 register denote the size of the memory banks and allow the CN8236 to incorporate the various bank sizes into contiguous memory. [Table 9-1](#) gives the coding of the BANKSIZE[2:0] control bits.

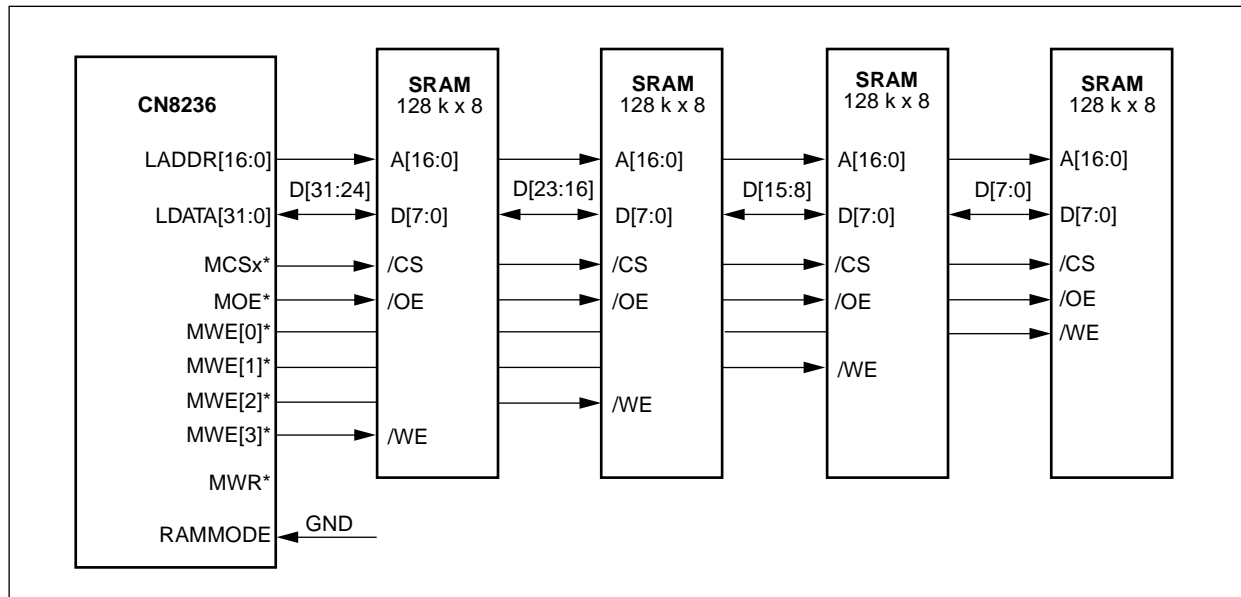
**Table 9-1. Memory Bank Size**

| BANKSIZE | Bank Memory Organization | Total Bank Size (Bytes) | PBSEL[1:0] or PCI Address Bits | Typical Implementation          |
|----------|--------------------------|-------------------------|--------------------------------|---------------------------------|
| 111      | 2 M × 32                 | 8 M                     | —                              | Future expansion                |
| 110      | 1 M × 32                 | 4 M                     | —                              | Two 1 M × 16, one 1 M × 32      |
| 101      | 512 K × 32               | 2 M                     | A[22:21]                       | Four 512 K × 8                  |
| 100      | 256 K × 32               | 1 M                     | A[21:20]                       | Two 256 K × 16, Eight 256 K × 4 |
| 011      | 128 K × 32               | 512 K                   | A[20:19]                       | Four 128 K × 8                  |
| 010      | 64 K × 32                | 256 K                   | A[19:18]                       | Two 64 K × 16, Eight 64 K × 4   |
| 001      | 32 K × 32                | 128 K                   | A[18:17]                       | Four 32 K × 8                   |
| 000      | 16 K × 32                | 64 K                    | A[17:16]                       | Two 16 K × 16, Eight 16 K × 4   |

The memory controller is designed to work with standard by\_8 devices, by\_4 SRAM devices, and by\_16 devices. Grounding the RAMMODE input selects the by\_4 or by\_8 mode of operation, while pulling RAMMODE to a logic 1 selects by\_16 operation. When by\_16 operation is selected, the MWE[3:0]\* outputs become byte enables for both reads and writes. When reading local memory, entire 32-bit words are always read, regardless of memory type. [Figure 9-2](#) shows a typical 500-KB bank implementation using by\_8 SRAM devices. [Figure 9-3](#) shows a typical bank using by\_16 RAM. To connect different sized RAM banks, simply use more or less address bits; all other control remains the same.

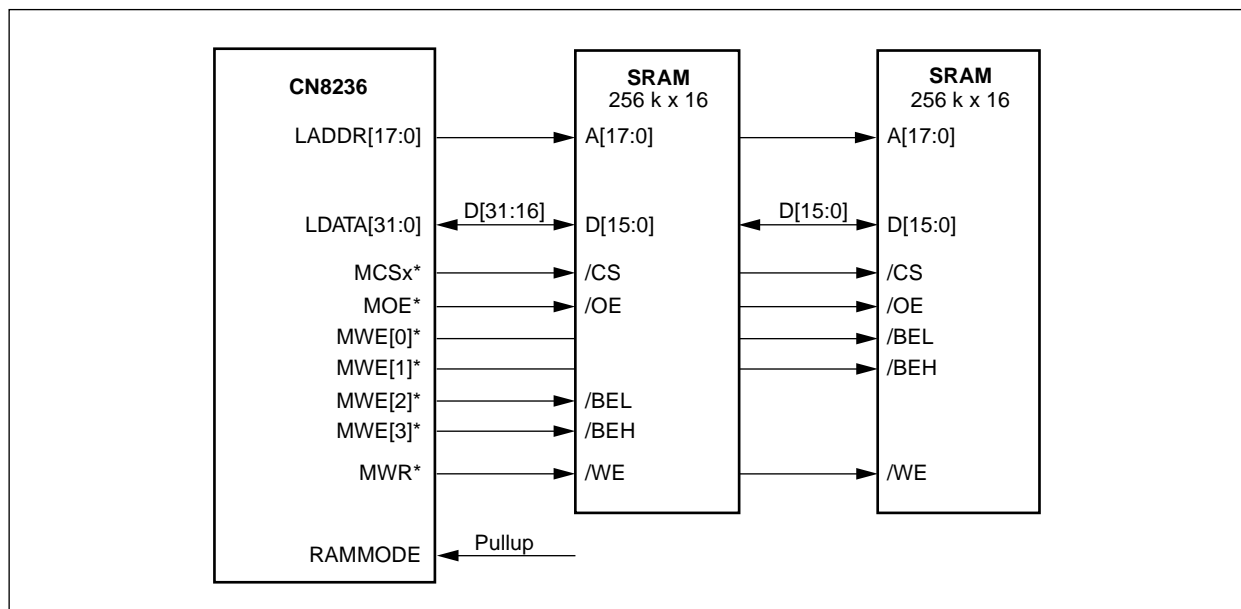
**NOTE:** The number and type of SRAM chips used affect the address and data bus capacitance and, therefore, the AC timing specifications and the required SRAM speed. The use of by\_4 devices causes more address bus loading than the use of by\_8 or by\_16 devices. See [Chapter 16.0](#), for detailed timing information.

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



8236\_060

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



8236\_061



The memory map contains space allocated to the RS825x (physical layer device), and to a future second Mindspeed PHY device. This mapping is only valid when the PROCMODE input pin is pulled high, indicating standalone operation with no local processor present. [Section 10.6](#) details standalone operation. When PROCMODE is logic low and the local processor is present, addresses 0x1FFh through 0xFFFh are available for general use and are mapped to MCS[0]\*.

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

Accesses made to the control registers and internal SRAM by the local processor follow the convention for SRAM accesses; that is, either 0 or 1 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 0 wait state. When the CN8236 decodes a PCI slave read to its address space, the CN8236 performs a prefetch of four subsequent (contiguous) word locations.

SRAM access time requirements are directly proportional to the system clock speed and 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 16.0](#) for timing information.)

## 9.3 Memory Size Analysis

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

### SEGMENTATION

1. The 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. 32 OAM PM channels active.
4. Transmit queues configured for 256 entries.
5. No status queues in SAR local memory.
6. Each active channel has an average of one active segmentation buffer.
7. RS\_Queue size is 1,024 entries.

### REASSEMBLY

1. Eight VPIs per 1,024 channels.
2. 1,024 entries preallocation on VPI = 0 only.
3. UNI VPI space.
4. 32 OAM PM channels active.
5. Free buffer queues configured for 256 entries.
6. No status queues in SAR local memory.
7. FBQ pools 0-15 have 4-word entries, and pools 16-31 have 2-word entries.

Table 9-2. Memory Size in Bytes

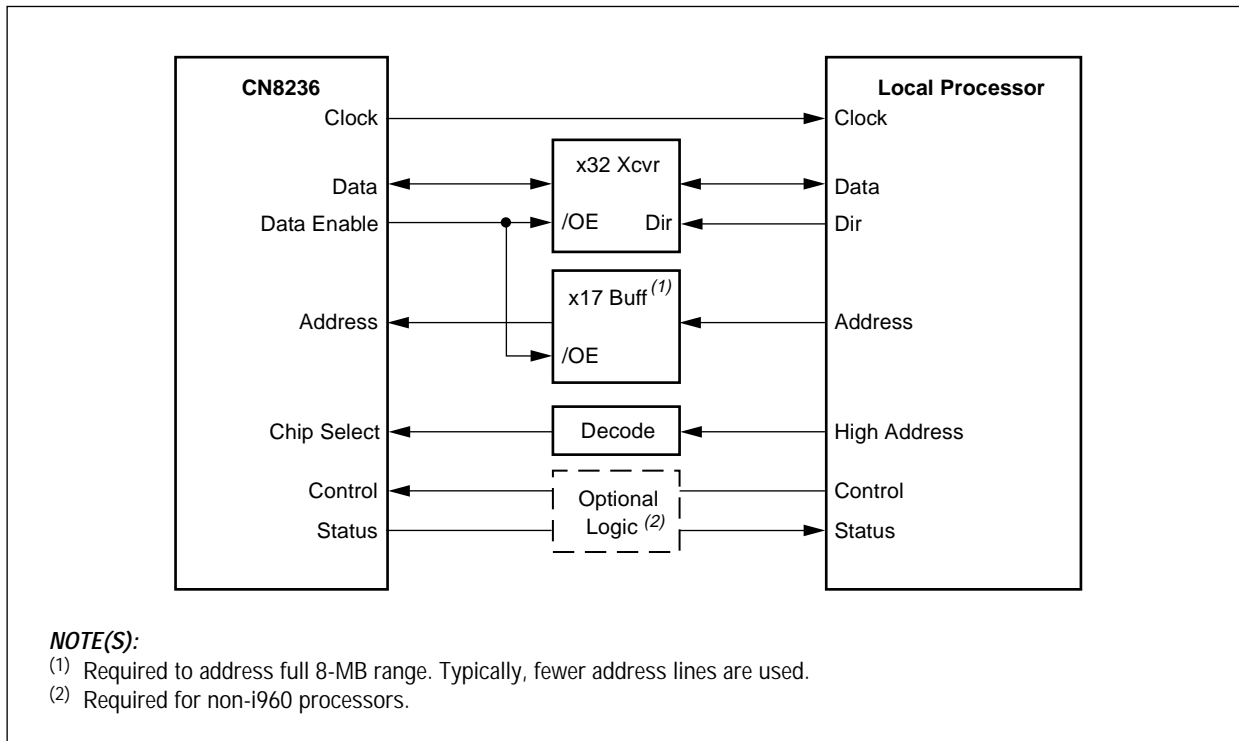
| Data Structure         | One Peer | 16 Peers | 32 Peers |
|------------------------|----------|----------|----------|
| SEG Schedule Table     | 16896    | 16896    | 16896    |
| SEG SEG_PM             | 1024     | 1024     | 1024     |
| RS Queue               | 8192     | 8192     | 8192     |
| SEG Transmit Queues    | 1024     | 16384    | 32768    |
| SEG ER Tables          | 67200    | 67200    | 67200    |
| RSM Free Buffer Queues | 4096     | 65536    | 98304    |
| RSM PM-OAM             | 512      | 512      | 512      |
| RSM VPI Index Table    | 1024     | 1024     | 1024     |
| Total Fixed            | 99200    | 176000   | 225152   |
| SEG VCC Table          | 40960    | 40960    | 40960    |
| SEG Buffer Descriptors | 20480    | 20480    | 20480    |
| RSM VCC Table          | 49152    | 49152    | 49152    |
| RSM VCI Index Table    | 92       | 92       | 92       |
| Total Incremental      | 110684   | 110684   | 110684   |
| Grand Total            | 209884   | 286684   | 335836   |

# 10.0 Local Processor Interface

## 10.1 Overview

The CN8236 integrated circuit can be used in conjunction with an external processor that performs initialization, link management, monitoring, and control functions. The local processor interface consists of a loosely coupled architecture that interfaces to the CN8236 through bidirectional transceivers and buffers controlled by the local processor and the CN8236, as shown in Figure 10-1. This architecture allows the processor access to all of the CN8236 SAR-shared memory and control registers, while insulating the CN8236 from processor instruction and data cache fills. This also allows the local processor the option to control multiple CN8236 and/or physical interface devices.

Figure 10-1. CN8236—Local Processor Interface



**NOTE(S):**

- (1) Required to address full 8-MB range. Typically, fewer address lines are used.
- (2) Required for non-i960 processors.

8236\_111

The processor interface is a generic synchronous interface based on the Intel i960CA 32-bit architecture and is completely compatible with the i960CA/CF and the new i960Jx processors. Other synchronous and asynchronous processors (for example, from Motorola, AMD, IDT) can be interfaced using external circuitry. The only requirement is that the processor have a 32-bit bus and that the control signals be synchronized to SYSCLK.

To access the CN8236 SAR-shared memory or control registers, the processor must arbitrate with the CN8236 for access to the memory controller. Due to the requirements of reassembly and segmentation access to SRAM and the implications of PCI bus utilization, the local processor has the lowest priority in the memory arbitration scheme. Since the local processor is typically used for low bandwidth supervision and maintenance functions, this should be acceptable.

When the local processor accesses the CN8236's control registers, internal SRAM, or SAR-shared memory, a local processor memory request is generated internal to the CN8236. The memory arbiter then coordinates this request with requests from other memory consumers and grants the memory bus to the local processor at the appropriate time. The local processor is held off during this process by the insertion of a variable number of wait states, accomplished by the i960 withholding *READY\** or *RDYRCV\**. Once the local processor is granted the memory system, the transceivers are enabled to allow the local processor's address and data to access the SRAM or control registers. The conclusion of the data transaction is signaled by the assertion of *PRDY\**. Wait states can be inserted by the processor at any time by asserting *PWAIT\**. The last data cycle in a burst is indicated by the *PBLAST\** signal. In this manner, non-i960 processor half-speed buses or slow transceivers can be accounted for.

The *LP\_BWAIT* bit in the *CONFIG0* register automatically adds a single wait state between the first access in a burst and subsequent accesses. This can be used to simplify the design of memory controllers for processors that do not produce a wait output and which require more time between data cycles in a burst.

## 10.2 Interface Pin Descriptions

The local processor bus interface consists of the control, address, and status signals described in [Table 10-1](#). Refer to [Table 2-1](#) and [Figure 10-7](#) for further information on these interface pins.

**Table 10-1. Processor Interface Pins (1 of 2)**

| Signal     | Dir <sup>(1)</sup> | Description  |
|------------|--------------------|--|
| PROCMODE   | I                  | Processor interface mode select input—A logic low on this input enables the local processor mode of operation.   |
| PCS*       | I                  | Processor interface chip select—A logic low on this signal in conjunction with a logic low on PAS* at the rising edge of SYSCLK initiates a memory request to the memory controller.               |
| PAS*       | I                  | Processor address strobe— A logic low on this signal in conjunction with a logic low on PCS* latches the value of PWNR, PBSEL[1:0], PADDR[1:0], and PBE[3:0]* at the rising edge of SYSCLK.        |
| PWNR       | I                  | Processor write/read select—A logic 1 on this input indicates a write cycle; a logic 0 indicates a read cycle. Latched at rising edge of SYSCLK when PAS* and PCS* are active.                     |
| PADDR[1:0] | I                  | Word select address inputs—Indicates the word address for a single cycle access, or the first word for a multi-cycle burst access. Latched at rising edge of SYSCLK when PAS* and PCS* are active. |
| PBSEL[1:0] | I                  | Bank select inputs—Decode to select MCS[3:0]*. Latched at rising edge of SYSCLK when PAS* and PCS* are active.   |

Table 10-1. Processor Interface Pins (2 of 2)

| Signal   | Dir <sup>(1)</sup> | Description   |
|--|--------------------|---|
| PBE[3:0]*  | I                  | Byte select inputs—Active low. Allows individual bytes of selected word to be written. Not active on reads. Latched at rising edge of SYSCLK when PAS* and PCS* active. PBE[3]* controls writes to LDATA[31:24]; PBE[2]* controls LDATA[23:16]; etc.  |
| PWAIT*   | I                  | Processor wait input—Allows the processor to insert a variable number of wait states to extend memory transaction. Must be active on rising edge of SYSCLK with PRDY* active to insert wait cycle. Can be used to interface to half speed or slow processor bus or to allow the use of slow transceivers. If insertion of wait states is not required, set this input to a logic high. This signal can only be active, logic low, when PBLAST* is a logic high. |
| PBLAST*  | I                  | Processor burst last input—Indicates the last word of a cycle. Must be active on rising edge of SYSCLK with PRDY* active to indicate last cycle. If burst accesses and wait cycles generated by PWAIT* are not required, this signal should be set to a logic low.  |
| PRDY*  | O                  | Processor interface ready signal—A logic low on this signal at rising edge of SYSCLK indicates that the present cycle has been completed. If a read cycle, the data is valid to latch by the processor; if a write cycle, the data has been written and can be removed from the bus. When PRDY* is active, wait states can be inserted with PWAIT*, or a single or burst cycle can be terminated by PBLAST* <sup>(2)</sup> .                                    |
| PFAIL*   | I                  | The local processor can indicate a failure of its internal self-test or initialization processes by asserting the PFAIL* input to the CN8236.   |
| <p><b>NOTE(S):</b></p> <p>(1) Direction given with respect to the CN8236.</p> <p>(2) This output corresponds to the READY* or RDYRCV* input in the i960 architecture.</p> <p>3. The processor system is responsible for controlling the direction of the bidirectional data bus transceiver. In the i960 architecture, this can be controlled by the DT/R* signal.</p> |                    |   |

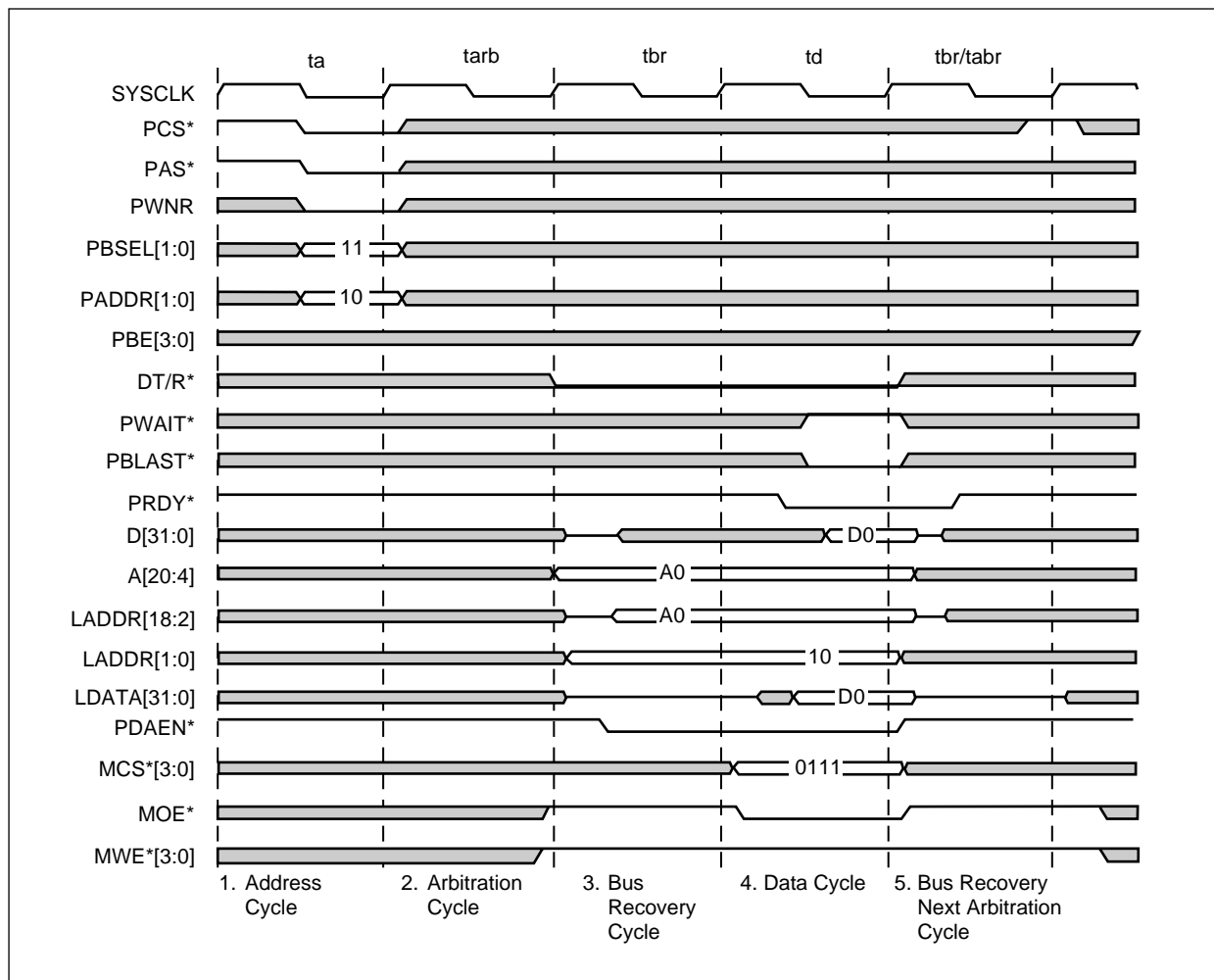
## 10.3 Bus Cycle Descriptions

Throughout the bus cycle descriptions, cycle refers to a single SYSCLK cycle ending with a rising edge. An arbitration cycle is one in which the memory requests from the local processor and internal memory consumers are compared, and the one with the highest priority is granted the memory access on the next cycle. A memory access that was previously arbitrated might occur on an arbitration cycle. Once the local processor has successfully acquired the memory controller, it holds the bus until it is relinquished by the assertion of PBLAST\* on the last data cycle. Therefore, local processor burst transfers are always completed, and can theoretically be of arbitrary length. However, in practice, burst transfers should be limited to four or less. The maximum arbitration delay for a local processor access is on the order of 20 cycles; however, it is typically from one to four cycles. This parameter is heavily influenced by the SYSCLK frequency, line rate, number of VCCs, idle cell ratio, and SRAM access speed. Therefore, a system design in which local processor accesses must occur within a fixed time period is not recommended.

### 10.3.1 Single Read Cycle, Zero Wait State Example

Figure 10-2 illustrates a single read cycle with 0 wait states. During the address cycle (cycle 1) at the rising edge of SYSCLK with PCS\* and PAS\* active, a memory request is generated by the processor interface circuitry. Also at this time, the read/write select, bank select, and word select inputs (PWNR, PBSEL[1:0], and PADDR[1:0]) are internally latched. The byte enables (PBE[3:0]\*) are Don't-Cares during reads. During cycle 2, this local processor memory request is processed by the memory arbitration circuitry. If no other memory consumers request an access on the same cycle, the local processor is granted access on cycle 3. However, to take into account bus transceiver turnaround, cycle 3 is always a wait or bus recovery state, which gives sufficient time for the address from the processor to access the SRAM. For 0 wait state SRAM, unless a wait state is inserted by the processor, the data is available to be latched into the processor on cycle 4, which is indicated by the assertion of PRDY\*. Cycle 5 is an arbitration cycle for the internal memory consumers, which might have requested access during the processor access. It also serves as a bus recovery cycle for the processor. Once the PCS\*, PAS\*, PWNR, PBSEL[1:0], and PADDR[1:0] are sampled at cycle 1, they are Don't-Cares for the remainder of the access. DT/R\* is an output supplied by the local processor to indicate the direction of the data transceivers. The CN8236 PDAEN\* signal is active to enable data and address.

Figure 10-2. Local Processor Single Read Cycle



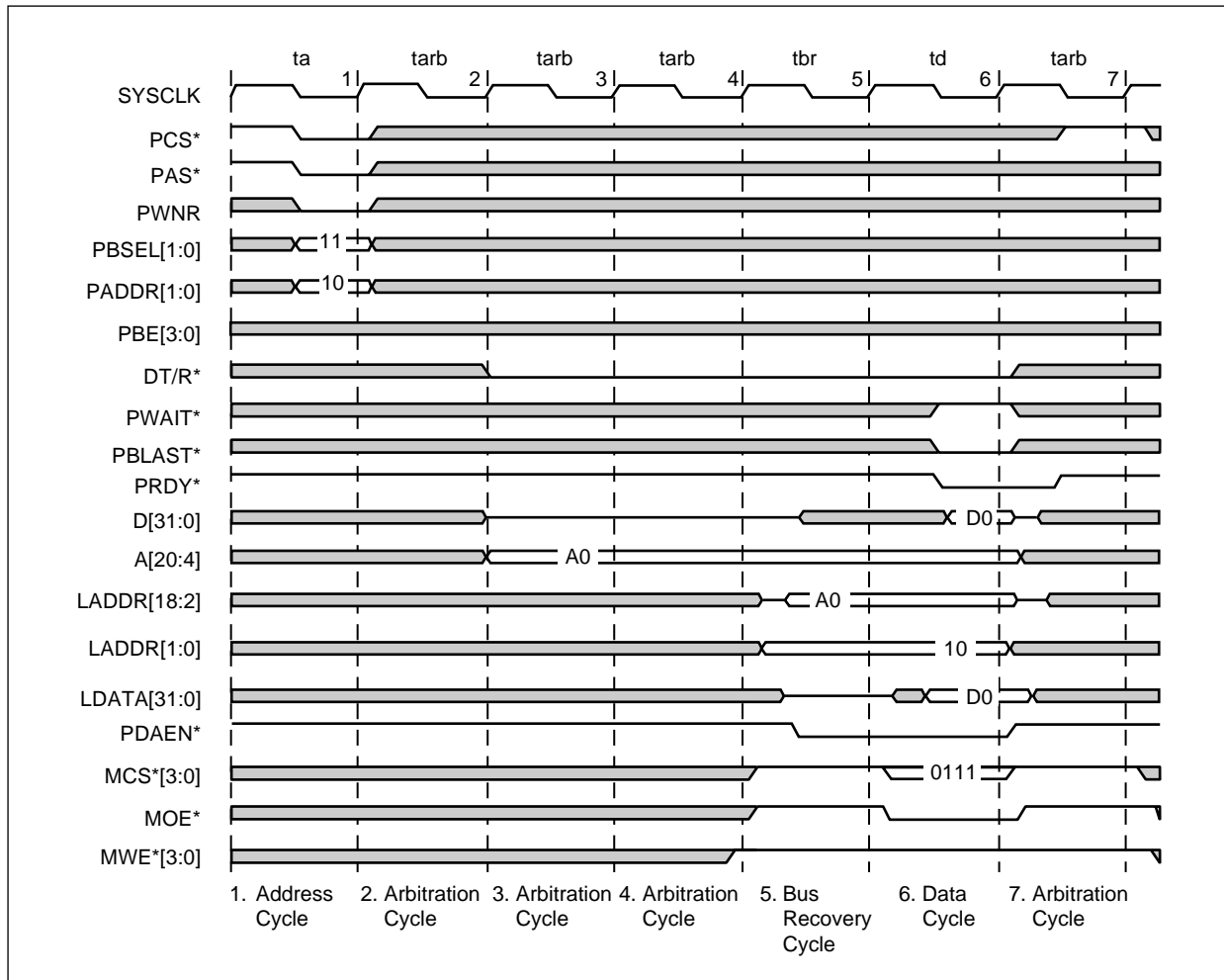
8236\_062



### 10.3.2 Single Read Cycle, Wait States Inserted By Memory Arbitration

Figure 10-3 illustrates a local processor single read cycle with arbitration wait states. This example is similar to the preceding one, except that here the local processor is not able to access the RAM immediately because of higher priority memory requests on cycles 2 and 3. On cycle 4, the memory controller allows the local processor access to the address and data bus, and the transaction takes place at the end of cycle 6.

Figure 10-3. Local Processor Single Read Cycle with Arbitration Wait States

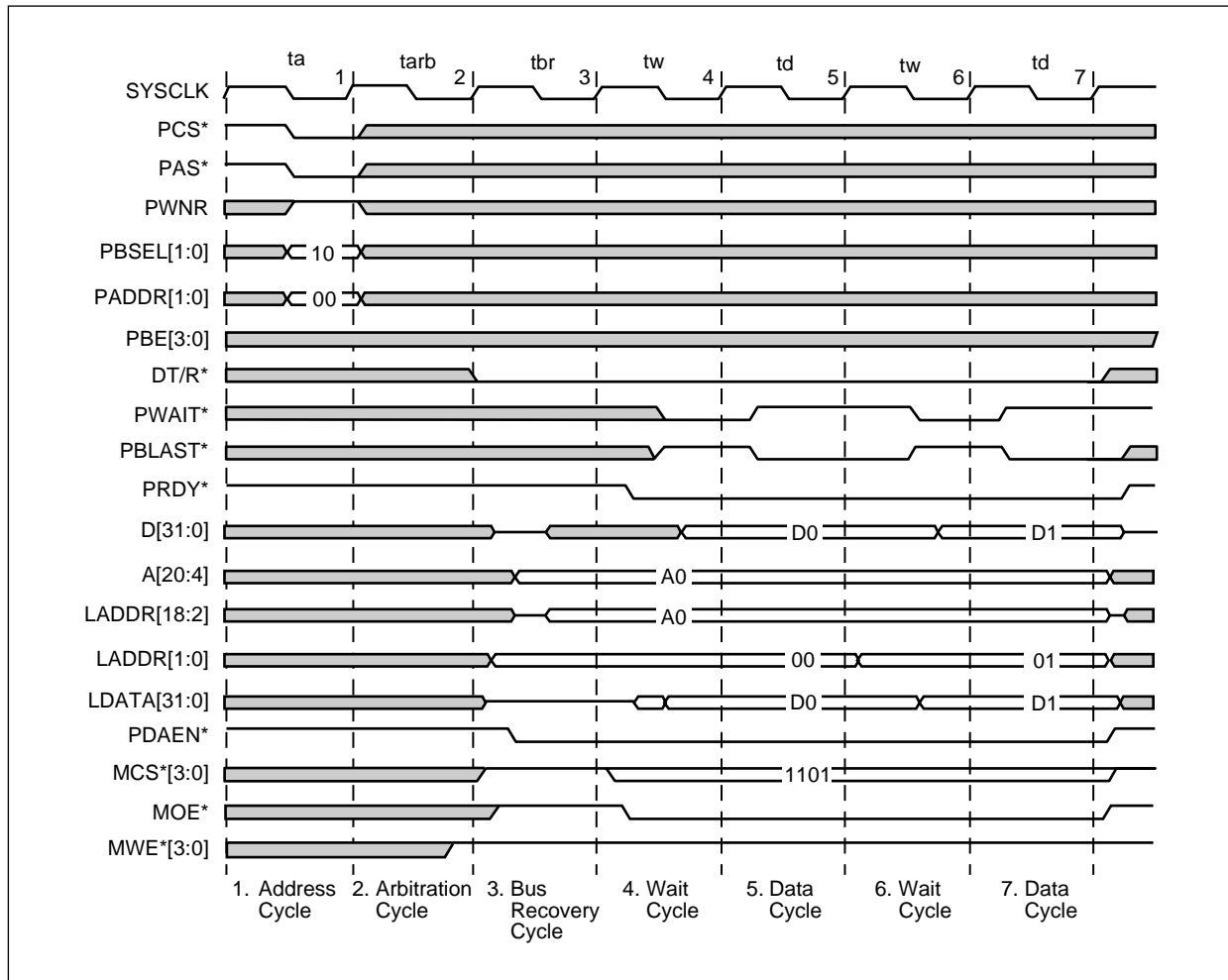


8236\_063

### 10.3.3 Double Read Burst With Processor Wait States

In Figure 10-4, the processor inserts wait states on cycle 4 and cycle 6 to allow additional time for the reads to occur. At the rising edge of SYSCLK on cycle 4 and cycle 6, the combination of PWAIT\* low and PRDY\* low extends the read by one more cycle. The local processor word select inputs (PADDR[1:0]) are latched at cycle 1. The CN8236 word select address lines, LADDR[1:0], are incremented automatically at the beginning of cycle 6.

Figure 10-4. Local Processor Double Read with Wait States Inserted

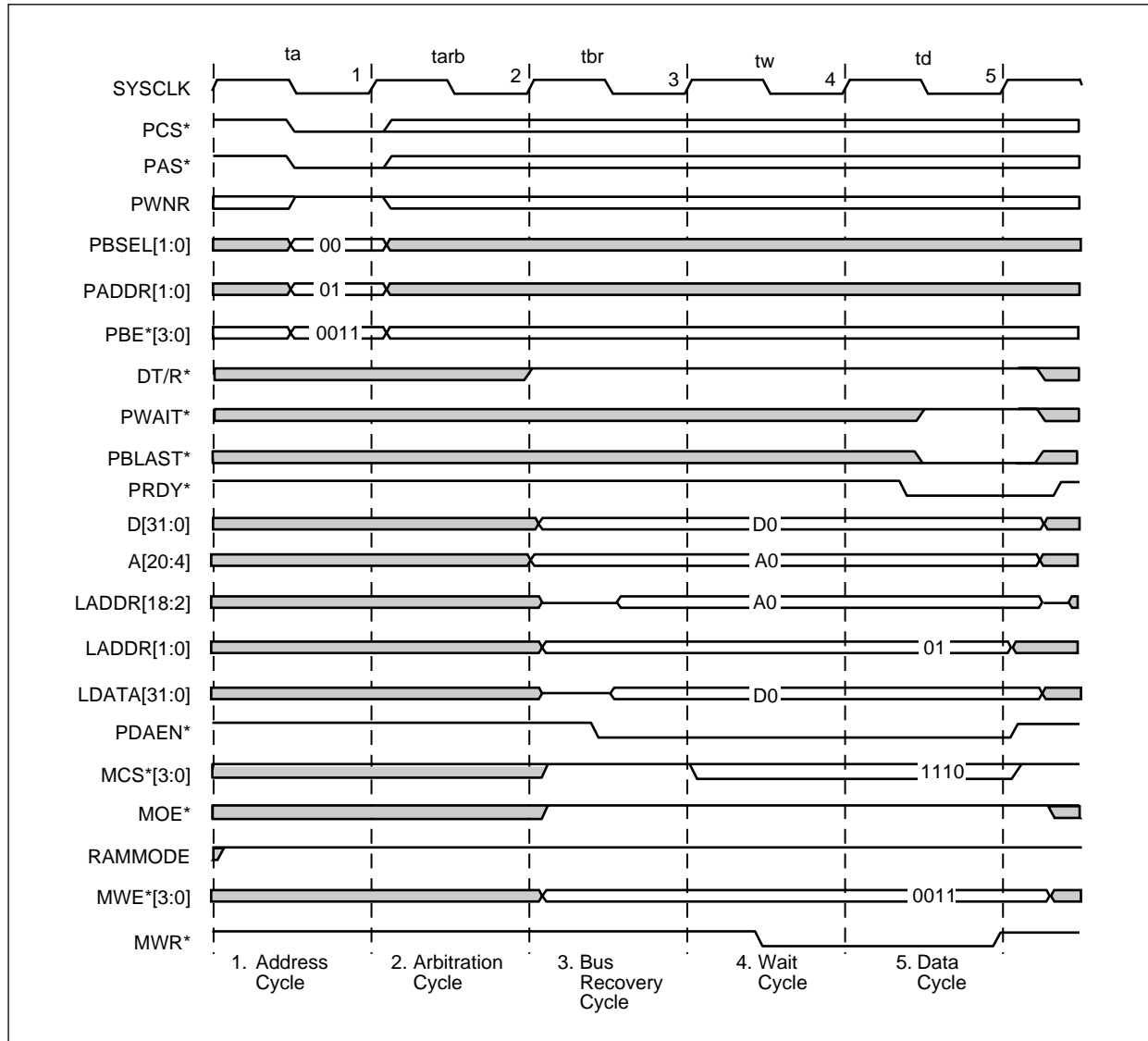


8236\_064

### 10.3.4 Single Write With One-Wait-State Memory

In Figure 10-5, the local processor performs a single write into one-wait-state memory. There is no arbitration delay. RAMMODE is a logic high, indicating that by\_16 RAM is used. Here the PBE[3:0]\* inputs are latched at cycle 1, and are used to select the byte enables that are active during the cycle when MWR\* is active. In this case, the two most significant bits are active, indicating a 16-bit write to the two most significant bytes.

Figure 10-5. Local Processor Single Write with One Wait State by\_16 SRAM

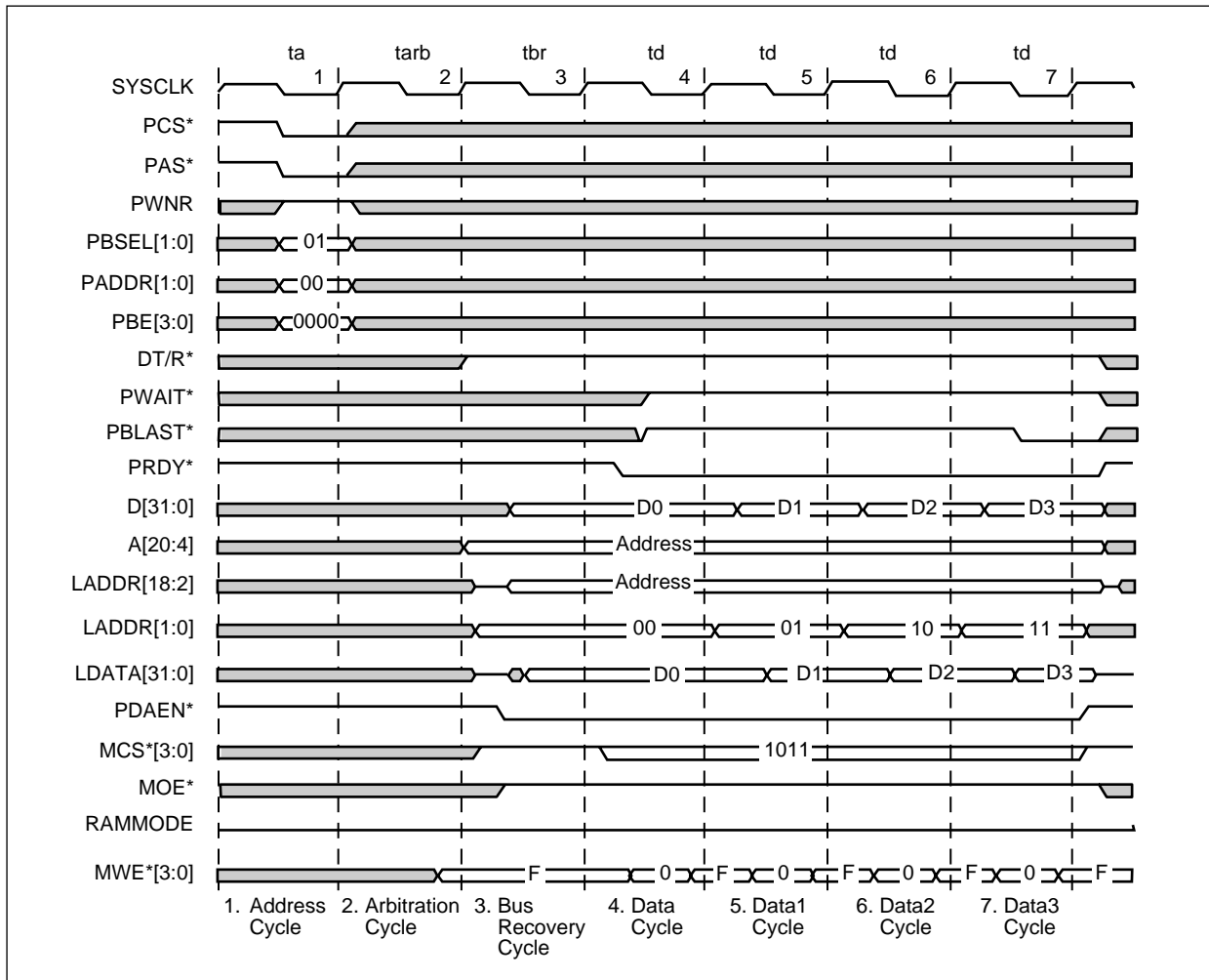


8236\_065

### 10.3.5 Quad Write Burst, No Wait States

In Figure 10-6, a quad burst write access to 0-wait-state memory is illustrated. RAMMODE is logic low, selecting by\_8 or by\_4 RAM mode. Here PBE[3:0]\* is latched on cycle 1, indicating that the write is active on all bytes, and the MWE\*[3:0] outputs are active as write strobes while MWR\* is not used. The SAR-shared memory word select addresses, LADDR[1:0], are incremented automatically by the CN8236 on each successive write cycle. Although the i960 architecture has the limitation that a quad word transfer must start on a quad word boundary, the CN8236 does not have that limitation. Thus, the PADDR[1:0] bits can be any value and are incremented as long as the burst transfer proceeds.

Figure 10-6. Local Processor Quad Write, No Wait States



8236\_066

## 10.4 Processor Interface Signals

Figure 10-7 illustrates the signal interface between the CN8236 device and the i960CA/CF processor. The memory region decoded for PCS\* should be set for  $N_{RAD}$  and  $N_{WAD} = 2$ ,  $N_{RDD}$  and  $N_{WDD} = 0$  or  $1$  (depending on the use of 0 or 1 wait state SRAM), and  $N_{XDA} = 1$ . In addition, external ready control must be enabled, and burst can be enabled or disabled at the system designer's option. Pulling up the i960 CLKMODE input to a logic 1 selects the divide-by-one clock mode, making i960 PCLK synchronous to SYSCLK.



This configuration is for addressing the entire 8 MB of SRAM. In the majority of systems, the SRAM requirements is considerably less. The implications of this are that the PBSEL[1:0] inputs can be driven by lower order address lines, and there are less than 17 address lines to buffer. Therefore, in most applications, the data transceivers can utilize two x\_16 parts, such as a 74ABT16245, and the address buffer can utilize a single x\_16 74ABT16244.

**NOTE:** The i960CA/CF signals a failure of its internal self-test upon reset or power-up, by asserting its FAIL\* output. This line is connected to the PFAIL\* pin of the CN8236, and the status of this pin is reflected in the Host Interrupt Status register [HOST\_ISTAT0; 0xC0].

## 10.5 Local Processor Operating Mode

The major difference between the i80960Jx processor and the i80960CA is that the i80960Jx utilizes a multiplexed address/data bus structure, while the i80960CA/CF is non-multiplexed. However, in the CN8236 system, the de-multiplexing of address/data takes place on the processor side of the address buffers and, therefore, does not affect the CN8236. Otherwise, the i80960Jx has the same bus control signals as the i80960CA/CF with the exception of the WAIT\* signal, which the i80960Jx does not possess. The insertion of wait states, if required, must be accomplished by an external memory controller, which in any case, is required for a i80960Jx implementation.

## 10.6 Standalone Operation

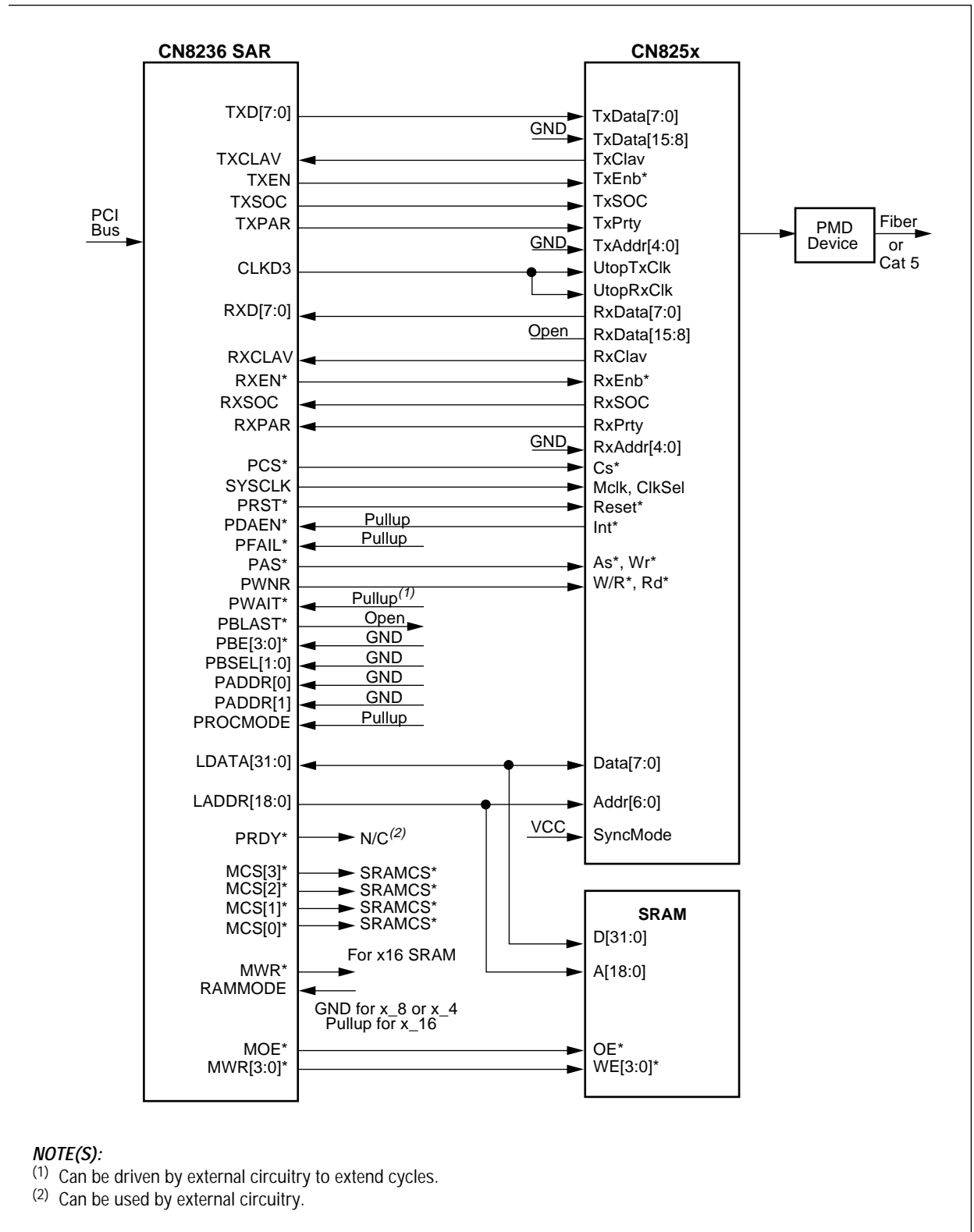
Standalone interface pins and descriptions are given in [Table 10-2](#). [Figure 10-8](#) shows the signal interface between the CN8236 and the RS825x ATM receiver/transmitter device with no local processor. The PCS\*, PAS\*, and PWNR pins are now outputs providing chip select, address strobe, and write/read control to the RS825x. PDAEN\* is now an input connected to the interrupt sources of the RS825x. PBLAST\* is a second chip select, which can be used to connect a future second Mindspeed PHY device. The PRDY\* output is active and indicates the cycles in which the data transaction occurs. The PWAIT\* input is active and can be used to prolong the cycle (as shown in [Figure 10-9](#)). Physical interface devices other than the RS825x can be connected by using PWAIT\* to extend the read or write cycle, and by using external logic to translate the CN8236 control signals.

**Table 10-2. Standalone Interface Pins**

| Signal   | Dir <sup>(1)</sup> | Description   |
|--|--------------------|---|
| PROCMODE   | I                  | Processor interface mode select input. A logic 1 enables standalone operation without a local processor.                          |
| PCS*   | O                  | Chip select output for PHY device number 1. Synchronous to SYSCLK.  |
| PBLAST*  | O                  | Chip select output for PHY device number two. Synchronous to SYSCLK.  |
| PAS*   | O                  | PHY address strobe. Synchronous to SYSCLK.  |
| PWNR   | O                  | PHY write/read select. A logic 1 on this output indicates a write cycle, a logic 0 indicates a read cycle. Synchronous to SYSCLK. |
| PRDY*  | O                  | PHY interface ready signal. A logic low on this signal at rising edge of SYSCLK indicates that the data cycle has been completed. |
| PWAIT*   | I                  | PHY wait input. Allows external logic to insert wait states to extend data cycles. Only active when PRDY* is active.              |
| PDAEN*   | I                  | PHY interrupt input, active low, level sensitive <sup>(2)</sup> .   |
| PADDR[1:0]   | I                  | Not used, pull to logic 0.  |
| PBSEL[1:0]   | I                  | Not used, pull to logic 0.  |
| PBE[3:0]*  | I                  | Not used, pull to logic 0.  |
| PFAIL*   | I                  | Not used, pull to logic 1.  |
| <p><b>NOTE(S):</b><br/> <sup>(1)</sup> Direction given with respect to the CN8236.<br/> <sup>(2)</sup> See the HOST_ISTATO register for details.</p> |                    |   |



Figure 10-8. CN825x and SAR (CN8236) Interface (Standalone Operation)

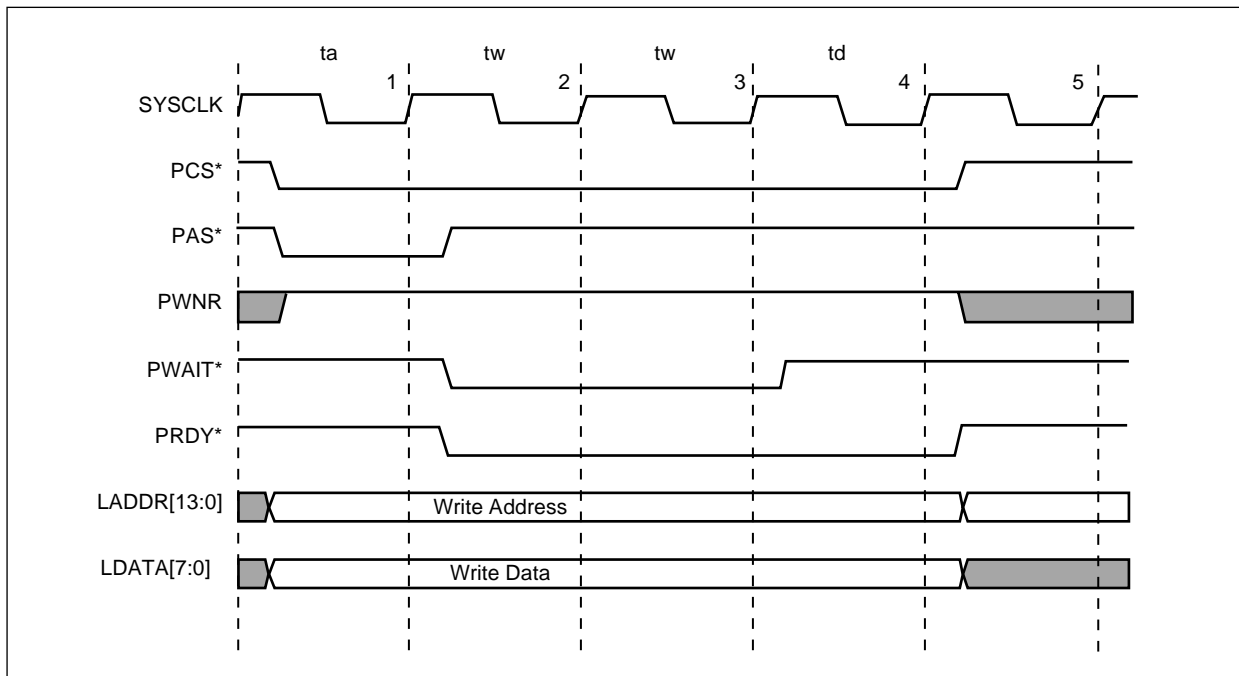


**NOTE(S):**

- (1) Can be driven by external circuitry to extend cycles.
- (2) Can be used by external circuitry.

8236\_113

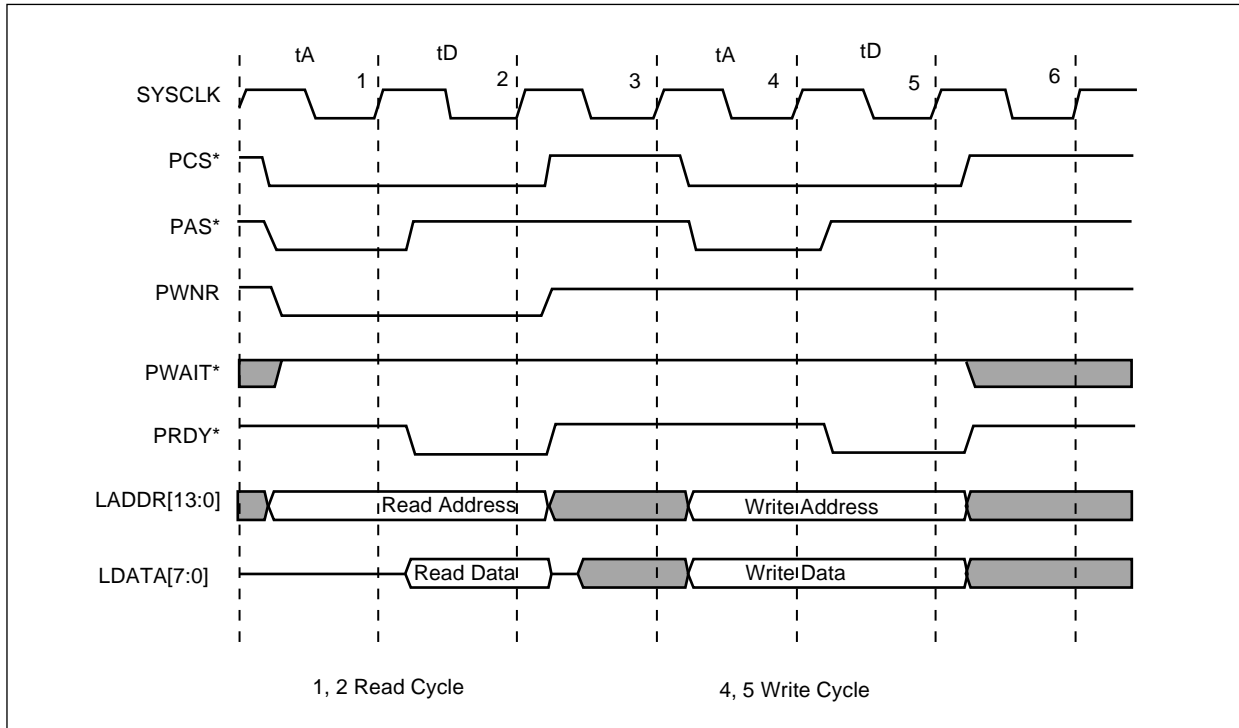
Figure 10-9. CN8236/PHY Functional Timing with Inserted Wait States



8236\_067

Figure 10-10 shows a read and write RS825x access. At cycle 1 (the rising edge of SYSCLK) the RS825x samples PCS\*, PAS\*, and PWNR low, indicating a read cycle. By the next rising edge of SYSCLK at cycle 2, the data is output by the RS825x to be latched by the CN8236. The same procedure occurs for a write except that at cycle 4, PWNR is sampled high. The RS825x then latches the data to the appropriate internal register on the next SYSCLK rising edge at cycle 5.

Figure 10-10. CN8236/RS825x Read/Write Functional Timing

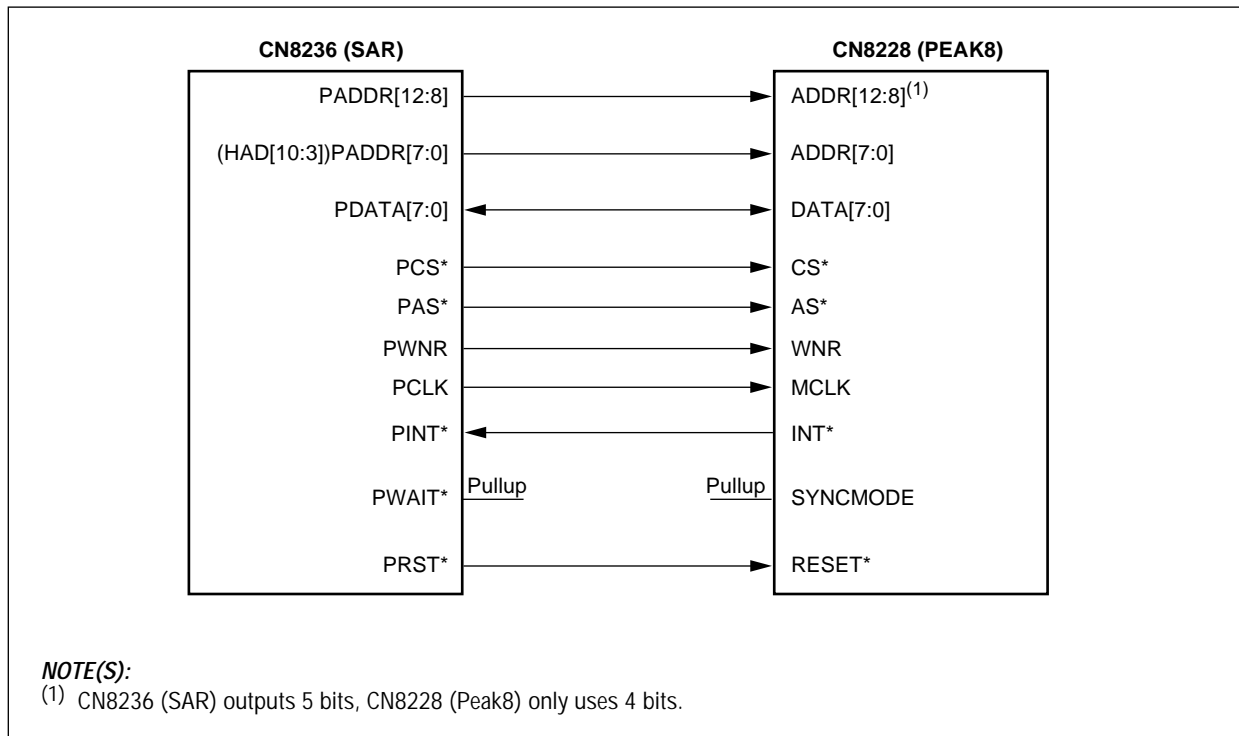


8236\_068

### 10.6.1 Microprocessor Interface for Multiple Physical Devices

Multi-phy or extended addressing for a PHY is provided through a PHY page mechanism. This allows address bits to be appended to PHY control access without increasing the size of the PHY memory map as seen by the PCI. The PHY BANK field in the CONFIG1 control register provides up to 5 bits of page addressing for a total of 14 bits of PHY addressing. The contents of PHY BANK are placed on LADDR[13:9] during PHYCS1 or PHYCS2 accesses. Figure 10-11 illustrates control connections between the SAR and a PHY device.

Figure 10-11. SAR/Peak 8 Control Connections



8236\_114

## 10.7 System Clocking

The CN8236 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. In either case, the clock signal would be looped externally to RxCLK and TxCLK (if MULT\_CLR set low, TxCLK is not used). For example, if CLK2X is 66 MHz, then CLKD3 is 22 MHz, and is suitable for the UTOPIA interface. If CLK2X is 50 MHz, then SYSCLK is 25 MHz and is suitable for the UTOPIA interface.

The CLK2X frequency required for a given application is a function of the physical line rate, number of VCCs, active concurrent VCCs, and the SRAM cycle time.

## 10.8 Real-Time Clock Alarm

A real-time clock counter and alarm registers are built into the CN8236. This real-time clock consists simply of a 7-bit pre-scaler (configured via the DIVIDER field in the CONFIG0 register) that accepts the SYSCLK input and outputs a constant (nominally 1 MHz) pulse train, and a 32-bit read/write counter (the Real-Time Clock register [CLOCK;0x00]), that counts the number of pulses output by the pre-scaler since the system was initialized. When the pre-scaler is set to generate a 1 MHz pulse train, the CLOCK counter counts in 1  $\mu$ s intervals. An interrupt is generated when the CLOCK counter overflows, that is, more than  $2^{32}$  pulses have occurred since it was cleared to 0. If this happens, the CLOCK counter simply wraps around to 0 and starts counting over. The control processor or host software is responsible for noting the overflow.

One simple real-time alarm is implemented in the CN8236. This is Alarm register 1 [ALARM1;0x04], which is continuously compared to the Clock register. When a match is detected, the corresponding interrupt is generated to the local or host processors. Either processor can then respond to this interrupt and reload a new value into the ALARM1 register.

## 10.9 CN8236 Reset

The CN8236 must be reset by the host processor prior to system initialization for proper operation. This can be done in one of two ways: by asserting the external HRST\* pin (which is normally connected to the system power-up reset circuitry), or by setting the GLOBAL\_RESET bit in the Configuration register 0 [CONFIG0;0x14]. The HRST\* pin must be deasserted and GLOBAL\_RESET must be cleared before beginning the CN8236 initialization process.

Asserting the HRST\* input pin automatically causes the local processor reset pin to be driven active. The reset to the local processor stays active until the LP\_ENABLE bit in the CONFIG0 register is set to a logic high. When using the GLOBAL\_RESET bit, the processor must manually set or clear the appropriate bits in the CONFIG0 register.

# 11.0 PCI Bus Interface

---

## 11.1 Overview

The PCI bus interface is compliant with *PCI Local Bus Specification*, Revision 2.1. 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 CN8236 to be valid before the next rising edge of HCLK.

The maximum PCI bus clock rate supported by the CN8236 is 33 MHz. The PCI bus interface logic is clocked directly from the PCI bus clock, while the remainder of the CN8236 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 CN8236 can also act 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. 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 as follows:

- 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 CN8236 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 CN8236 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.



## 11.2 Unimplemented PCI Bus Interface Functions

The PCI bus interface on the CN8236 does not implement all 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, and 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 CN8236, 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 0 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 0, the slave logic signals 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.

Implement Sections 3.1.8 and Section 7.2 of *Compact PCI Hot Swap Specification*. Assume that a logic low on the HSWITCH\* input is switch locked and a logic high is switch unlocked.

**NOTE:** The arming/disarming of the INS/EXT bits provides a switch debounce function.

## 11.3 PCI Configuration Space

In accordance with the *PCI Bus Specification*, Revision 2.1, the CN8236 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. (The descriptions and definitions of these register fields as implemented in the CN8236 are provided in [Chapter 14.0](#).)

The incoming DMA FIFO size is programmable and can be set to 2 KB or 8 KB depending on the value of the INCFIFO\_SZ bit in the CONFIG1 register.

## 11.4 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 that 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 terminates the current burst, maintain its bus request, and restarts 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 permanently terminates the transaction, flags an error, and ceases to process any more commands.

1. **Target Abort**—The PCI transaction terminates if the addressed target signals a target abort. In this case, the RTA and MERROR bits in the PCI Configuration register space are set, and the PCI\_BUS\_STATUS[4] bit in the SYS\_STAT register is set.
2. **Master Abort**—If the addressed target does not respond with an HDEVSEL\* assertion, a master abort is flagged. In this case, the RMA and MERROR bits in the PCI Configuration register space are set, and the PCI\_BUS\_STATUS[3] bit in the SYS\_STAT register is 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 are set and the PCI\_BUS\_STATUS[2] bit in the SYS\_STAT register is set.
4. **Interface Disabled**—If the driver or application software on the PCI host CPU has been disabled, the CN8236 PCI bus master logic (using the M\_EN bit in the Command field of the PCI bus configuration registers), any attempt to perform a DMA transaction to the PCI bus results in an error. In this case, the MERROR and INTF\_DIS bits in the PCI configuration space are set, and the PCI\_BUS\_STATUS[1] bit in the SYS\_STAT register is set.
5. **Internal Failure**—Upon a synchronization error between the DMA coprocessor and the PCI master logic, an internal failure is flagged. In this case, the MERROR and INT\_FAIL bits in the PCI configuration space are set, and the PCI\_BUS\_STATUS[0] bit in the SYS\_STAT register is set.

**NOTE:** 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 (RTA, RMA, or DPR) or with a reset of the PCI bus master logic using the HRST\* input pin. For example, a master abort error can be cleared by writing a logic 1 to the RMA status bit in the PCI Configuration register space, causing the status bit to be cleared. Internal failures (attempting to initiate a master transaction with the interface disabled, or loss of synchronization with the DMA controller) can only be reset by applying the global reset, CONFIG0 (GLOBAL\_RESET), or by asserting the HRST\* signal.

Next, the MERROR bit must be cleared. The MERROR bit in the PCI Configuration register drives the PCI\_BUS\_ERROR interrupt. To clear this interrupt, a logic high must be written to the MERROR bit location. The MERROR bit can also be cleared by a logic low on the HRST\* input pin.

The local processor can clear the error bits by setting CONFIG0 (PCI\_ERR\_RESET) to a logic high. After the errors have been cleared, the SAR must 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 PCI\_READ\_MULTI bit 22 in the CONFIG0 register.

## 11.5 Burst FIFO Buffers

Two small FIFO buffers are implemented to support PCI slave burst-mode operation (Read =  $8 \times 32$ , Write =  $64 \times 32$ ), to allow synchronization between the CN8236 internal logic and the PCI bus interface, and to carry commands from the DMA coprocessor to the PCI bus logic. The incoming master FIFO is  $512 \times 32$  or  $2 \text{ K} \times 32$  bits, the outgoing Master FIFO is  $16 \times 36$  bits.

## 11.6 PCI Bus Slave Logic

The PCI slave logic permits the host CPU on the PCI bus to access and modify CN8236 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 can 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 CN8236 system clock. Also, small FIFOs are used to speed up burst reads ( $8 \times 32$ ) and writes ( $64 \times 32$ ) 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. The slave interface responds 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 are reported to be asserting HSERR\* in the normal fashion, if SE\_EN and PE\_EN in the command register are both set.
- Parity errors during the data phase are ignored.

## 11.7 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 causes both of these bits to be logic low.

## 11.8 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 CN8236 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 CN8236 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 [Section 14.7](#) for the detailed information on the PCI Configuration space and PCI registers concerned with implementing the Power Management functions.

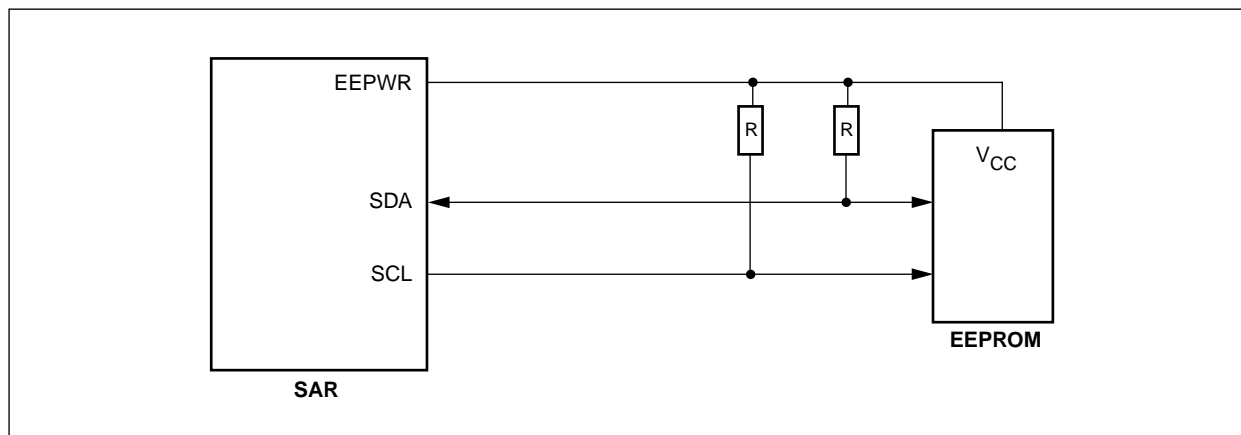
Power Management is enabled by default. This capability can be disabled via bit 2 of the EEPROM (Disable Capability register), which sets bit 20 of the PCI Status register to a logic low. See the *PCI Bus Power Management Interface Specification*, Revision 1.0, for specific information on the functions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

## 11.9 Interface Module to Serial EEPROM

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

A condition can arise where the protocol is violated and unknown operation of the EEPROM can occur. If HRST\* is applied while the EEPROM is being accessed, that is, right after HRST\* goes inactive, the access is abnormally terminated with indeterminate effects on the EEPROM. In order to reset the EEPROM, its power must be cycled. A 12 mA pin, EEPWR, is added to supply power to the EEPROM if the above condition cannot be guaranteed from happening. Whenever HRST\* is active, EEPWR is a logic low. When HRST\* goes to a logic high (inactive), EEPWR immediately goes to a logic high. [Figure 11-1](#) illustrates a suggested connection of the EEPROM.

Figure 11-1. EEPROM Connection



8236\_069

## 11.9.1 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 11-1](#).

**Table 11-1. EEPROM Fields**

| Address Offset   | Name             | Description  |
|--|------------------|--|
| 0x00   | FIELD_ENABLES    | Bit 5–Load Memory Size Mask from EEPROM.<br>Bit 4–Load Latency Timer from EEPROM.<br>Bit 3–Load General Enables from EEPROM.<br>Bit 2–Disable Capability registers (for Power Management).<br>Bit 1–Load Subsystem ID (SID) from EEPROM.<br>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).<br>Bit 3–Special Status register Bit 30 (MSTR_CTRL_SWAP, Master Control Byte Swap).<br>Bit 2–PCI Command register Bit 6 (PE_EN, Enable Detection of Parity Errors).<br>Bit 1–PCI Command register Bit 2 (M_EN, Master Enable). <sup>(1)</sup><br>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:<br>Bit 7 6 5 4 3 2 1 0 = Size<br>x 0 0 0 0 0 0 0 = 8 M  |
| <p><b>NOTE(S):</b><br/><sup>(1)</sup> System BIOS is typically responsible for setting these bits after programming the PCI Base address register.</p> |                  |  |

## 11.9.2 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 0, 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 0, the SUBSYSTEM\_ID and/or SUBSYSTEM\_VENDOR\_ID fields defaults to all 0s.

### 11.9.3 Accessing the EEPROM

The EEPROM is accessed through the PCI Configuration space at offset 0x4C, the EEPROM register. See [Section 14.7](#) for a description of the EEPROM register's contents. See [Table 14-16](#) for a description of register 4C.

Before starting an EEPROM operation, the BUSY bit must be a logic 0 (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 is a logic 1 (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 indicates the status of the operation. A logic 1 (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 0. The module then transfers 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 then polls the register until the BUSY bit is read as 0 (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 might 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 1. 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.

Explanation of series EEPROM clock:

$$\text{SCL} = (\text{PCI}/84)/4 = 98.2 \text{ KHz} (@\text{PCI} = 33\text{MHz})$$

### 11.9.4 Using the Subsystem ID Without an EEPROM

For applications that can utilize BIOS or boot code to initialize devices before loading high-level operating system software, the CN8236 allows for the programming of the PCI Configuration space fields, SUBSYSTEM\_ID and SUBSYSTEM\_VENDOR\_ID. This feature allows a user to employ the CN8236 without an EEPROM, but still allows for unique Subsystem IDs.

To program the SUBSYSTEM\_ID and SUBSYSTEM\_VENDOR\_ID fields, BIOS must first write a logic 1 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 0. When bit 31 is set to 0, writes to these two fields are ignored.



## ***11.10 PCI Host Address Map***

The address map of the CN8236 resources seen by the PCI bus is the same as that seen by the host processor. The base address of the CN8236 resource mapping is defined in the `BASE_ADDRESS_REGISTER_0` field, located in the PCI configuration space.

Burst reads of the Control and Status registers only return valid data for the first address. Subsequent data words read during a burst read are indeterminate.



# 12.0 ATM UTOPIA Interface

---

## 12.1 Overview of ATM UTOPIA Interface

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

The ATM UTOPIA interface for the CN8236 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 buffer to be used by the reassembly coprocessor.

The ATM UTOPIA interface is responsible for communicating with and controlling the ATM link interface device, 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 UTOPIA interface logic. The ATM UTOPIA interface accommodates Mindspeed RS825x, RS8228, or Bt8223 physical layer devices, 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 CN8236 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 buffer. 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 buffer into 8- or 16-bit (plus parity) streams, generates appropriate cell delineation pulses for use by the transmit ATM UTOPIA interface, and inserts the blank HEC byte into the ATM header of each cell prior to transferring it to the UTOPIA interface.
- Generates and checks odd parity on the octet transmit and receive data buses.
- Programmable single/separate UTOPIA clock via CONFIG1 register bit 23.

## 12.2 ATM UTOPIA Interface Logic

The CN8236 ATM UTOPIA 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] and UTOPIA1 determine whether the UTOPIA or slave UTOPIA protocol are used and whether it is level 1 or level 2.

## 12.3 ATM Physical I/O Pins

The operational mode desired is indicated to the CN8236 by appropriately driving the FRCFG[1:0] and UTOPIA1 inputs according to [Tables 12-1](#) and [12-2](#).

**Table 12-1. ATM Physical Interface Mode Select (FRCFG[1:0])**

| 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         |

**Table 12-2. ATM Physical Interface Mode Select (UTOPIA1)**

| UTOPIA1 | ATM Physical Interface Mode |
|---------|-----------------------------|
| 0       | UTOPIA Level 2              |
| 1       | UTOPIA Level 1              |

The interpretation of the ATM UTOPIA interface pins on the CN8236 and the actual signals generated or received by the framer in UTOPIA mode are shown in [Table 12-3](#). Both the TxClk and RxClk signals of the UTOPIA interface can be derived from the CLKD3 output of the CN8236.

**Table 12-3. UTOPIA Mode Signals**

| CN8236 Signal | PHY Signal   | Active Polarity | CN8236 Direction |
|---------------|--------------|-----------------|------------------|
| TxDATA[15:0]  | TxDATA[15:0] | —               | Out              |
| TxPAR         | TxPRTY       | —               | Out              |
| TxSOC         | TxSOC        | High            | Out              |
| TxEN*         | TxENB*       | Low             | Out              |
| TxCLAV        | TxFULL*      | Low             | In               |
| TxADD[4:0]    | TxADDR[4:0]  | —               | Out              |
| TxCLK         | TxCLK        | Rising Edge     | In               |
| RxDATA[15:0]  | RxDATA[15:0] | —               | In               |
| RxPAR         | RxPRTY       | —               | In               |
| RxSOC         | RxSOC        | High            | In               |
| RxEN*         | RxENB*       | Low             | Out              |
| RxCLAV        | RxEMPTY*     | Low             | In               |
| RxCLK         | RxCLK/TxCLK  | Rising Edge     | In               |
| RxADD[4:0]    | RxADDR[4:0]  | —               | Out              |

The interpretation of the ATM UTOPIA interface pins on the CN8236 and the actual signals generated or received by the framer in slave UTOPIA mode is shown in [Table 12-4](#).

**Table 12-4. Slave UTOPIA Mode Interface Signals**

| CN8236 Signal   | PHY Signal   | Active Polarity | CN8236 Direction |
|---|--------------|-----------------|------------------|
| TxDATA[15:0]  | TxDATA[15:0] | —               | Out              |
| TxPAR   | TxPRTY       | —               | Out              |
| TxSOC   | TxSOC        | High            | Out              |
| TxEN*   | TxENB*       | Low             | In               |
| TxCLAV  | TxEMP*       | Low             | Out              |
| TxCLK   | TxCLK        | Rising Edge     | In               |
| RxDATA[15:0]  | RxDATA[15:0] | —               | In               |
| RxPAR   | RxPRTY       | —               | In               |
| RxSOC   | RxSOC        | High            | In               |
| RxEN*   | RxENB*       | Low             | In               |
| RxCLAV  | RxFULL*      | Low             | Out              |
| RxCLK   | RxCLK        | Rising Edge     | In               |
| TxADDR[4:0]   | TxADDR[4:0]  | —               | In               |
| RxADDR[4:0]   | RxADDR[4:0]  | —               | In               |
| <b>NOTE:</b> When operating the CN8236 in 8-bit mode, pull-ups need to be added to the unused high order bits (RxDATA[15:8]). |              |                 |                  |

### 12.3.1 UTOPIA Interface

In addition to the current UTOPIA level 1 support, this interface provides complete support for UTOPIA level 2, 8/16 bit mode in both master and slave modes, including support for multi-PHY in both master and slave modes. Multi-PHY port shaping is provided by tunnels through one TxFIFO buffer. A 3-bit PORT\_ID is used to provide port identification for up to 8 ports in master mode (UTOPIA address 0 through 7) and 32 ports in slave mode. The UTOPIA output drives are 8 mA to meet the multi-PHY specification.

Multi-PHY mode is enabled by setting CONFIG1(MULTI\_PHY) to a logic high. In master mode, the PORT\_ID field is used as a port identification. On the segmentation side, the appropriate port identification is written in the PORT\_ID field in the VCC table entry. The maximum allowable PORT\_ID is CONFIG1(NUM\_PORTS). On the reassembly side, the UTOPIA interface polls addresses 0 through CONFIG1(NUM\_PORTS) for a cell. When a cell is detected in a PHY, the cell along with the PORT\_ID is transferred to the reassembly block. An expanded lookup mechanism is used to find the appropriate state table entry. (See [Section 5.2.2.5](#) for more details.)

In slave mode, the UTOPIA block responds only when the UTOPIA address equals CONFIG1(SLAVE\_ADDR). In non-multi-PHY master mode, CONFIG1(SLAVE\_ADDR) is output on the RxAddr and TxAddr busses.

Also, routing tag prepending is supported with programmable cell size up to 64 bytes in single PHY master mode and single/multi-PHY slave mode. On the reassembly side, the tag is discarded by the UTOPIA interface.

When F4 PMOAM operation or VP ABR operation is enabled, the routing tag content of all cells including OAM within the VP must be identical.



## 12.4 UTOPIA Level 2 Interface

The CN8236 supports both UTOPIA Level 1 and Level 2 interfaces. These are described in general terms below with an emphasis on their differences.

- UTOPIA Level 1: This is an 8- or 16-bit interface designed for data rates up to 200 Mbps at a clock rate of 25 MHz. Both Octet-level and Cell Level handshaking are supported.
- UTOPIA Level 2: This interface defines the Multi-port support features that allow up to 31 UTOPIA devices to multiplex on one UTOPIA bus. The SAR only supports up to eight PHY devices. It allows either 8 or 16 bit data buses and uses only cell level handshaking. It can transfer up to 800 Mbps when running at 50 MHz in the 16 bit mode.

Both Level 1 and Level 2 support odd parity over the width of the data bus.

### 12.4.1 Cell Tagging

In addition to the standard 53-octet cell formats, given in [Tables 12-5](#) and [12-6](#), the CN8236 supports user programmable cell sizes for Routing Tag applications. This allows a maximum cell size of 64 bytes. These cell formats are shown in [Tables 12-7](#) and [12-8](#). The number of octets added for the tagging function is programmed into the TAG\_SIZE bits of CONFIG1.

*Table 12-5. Cell Format 8 Bit Mode*

| Bit 7 | ... | 0                   |
|-------|-----|---------------------|
|       |     | Header 1            |
|       |     | Header 2            |
|       |     | Header 3            |
|       |     | Header 4            |
|       |     | UDF1 (HEC) (byte 5) |
|       |     | Payload 1           |
|       |     | :                   |
|       |     | Payload 48          |

**Table 12-6. Cell Format 16 Bit Mode**

| Bit 15              | ... | Bit 8 | Bit 7             | ... | 0 |
|---------------------|-----|-------|-------------------|-----|---|
| Header 1            |     |       | Header 2          |     |   |
| Header 3            |     |       | Header 4          |     |   |
| UDF1 (HEC) (byte 5) |     |       | UDF2 (0) (byte 6) |     |   |
| Payload 1           |     |       | Payload 2         |     |   |
| ⋮                   |     |       | ⋮                 |     |   |
| Payload 47          |     |       | Payload 48        |     |   |

**Table 12-7. Cell Format, Tagging Enabled, 8 Bit Mode**

| 7          | 0                            |                   |
|------------|------------------------------|-------------------|
| Tag 1      | 1 <sup>st</sup> byte of Cell |                   |
| ⋮          |                              |                   |
| ⋮          |                              |                   |
| Tag n      |                              |                   |
| Header 1   |                              |                   |
| Header 2   |                              |                   |
| Header 3   |                              |                   |
| Header 4   |                              |                   |
| HEC        |                              |                   |
| Payload 1  |                              |                   |
| Payload 2  |                              |                   |
| ⋮          |                              |                   |
| ⋮          |                              |                   |
| ⋮          |                              |                   |
| Payload 48 |                              | Last byte of Cell |

Table 12-8. Cell Format, Tagging Enabled, 16 Bit Mode

| 15         | 8 | 7          | 0 |                              |                   |
|------------|---|------------|---|------------------------------|-------------------|
| Tag 1      |   | Tag 2      |   | 1 <sup>st</sup> word of Cell |                   |
| ⋮          |   | ⋮          |   |                              |                   |
| ⋮          |   | ⋮          |   |                              |                   |
| Tag n-1    |   | Tag n-2    |   |                              |                   |
| Header 1   |   | Header 2   |   |                              |                   |
| Header 3   |   | Header 4   |   |                              |                   |
| HEC        |   | Not Used   |   |                              |                   |
| Payload 1  |   | Payload 2  |   |                              |                   |
| ⋮          |   | ⋮          |   |                              |                   |
| ⋮          |   | ⋮          |   |                              |                   |
| Payload 47 |   | Payload 48 |   |                              | Last word of Cell |

### 12.4.2 UTOPIA Configuration Control

Most options for the CN8236 UTOPIA interface are configured by control bits in the Config0 and Config1 registers. However, the selection of UTOPIA Level 1 versus Level 2 and of Master or Slave mode is controlled by the input pins FRCFG0, FRCFG1, and UTOPIA1. Normally, these inputs are hardwired as required by the system design.

UTOPIA Level 1/Level 2: Tying the UTOPIA1 input pin to ground selects UTOPIA level 2, connecting it to +3.3V selects UTOPIA Level 1. When selecting UTOPIA Level 2 be sure to program the UTOPIA\_MODE bit to logic high to enable Cell level handshaking (octet level handshaking is not valid in UTOPIA Level 2).

Master/Slave selection: Controlled by hardware inputs FRCFG[1:0] as defined in [Table 12-1](#).

Octet or Cell Handshaking: This is controlled by the UTOPIA\_MODE bit in the Config0 register. Setting this bit high selects cell-level handshaking. In this mode, both the CN8236 and the selected port only begin a transfer when there is room in the FIFO buffers for an entire cell. When UTOPIA Level 2 mode is selected, this bit MUST be set to logic high.

UTOPIA clocks: In the default state after a reset, the UTOPIA transmit and receive blocks both use the RxClk input. By setting the MULTI\_CLK bit of Config1 to logic high, the transmit side uses the TxClk input. The UTOPIA receive block continues to use the RxClk.

Multi-port operations: By default, the CN8236 expects a single device on the UTOPIA bus. By setting the MULTI\_PHY bit of Config1 to logic high, up to 8 devices can share the bus.

UTOPIA data bus width: The width of the UTOPIA data bus is selectable by the UTOP16 bit of the Config1 register. Setting this bit high enables a 16-bit data path.

**NOTE:** Number of ports on the bus: When running in multi-port mode the total number of ports to poll is programmed into the NUM\_PORTS bits of Config1.

Since port numbering begins with 0, this value equals –1 (total number of ports).

Slave address: When the CN8236 is configured as a slave on the UTOPIA bus, its port address must be programmed into the SLAVE\_ADDR bits of Config1. Software must be aware that all CN8236s on the bus default to the same address, (00), and therefore, must reprogram each device to a unique value.

### 12.4.3 UTOPIA Level 2 Multi-Port Operation

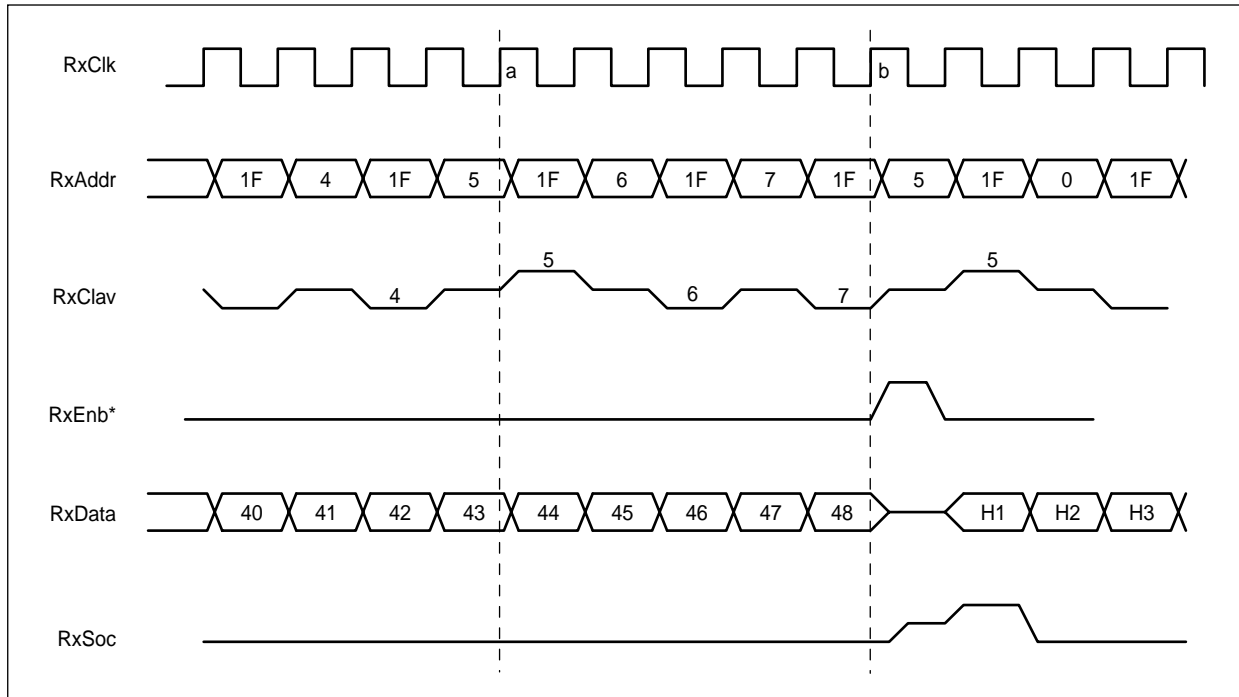
The CN8236 supports multi-port (up to eight) operations as described in the *UTOPIA Level 2 Specification af-phy-0039.000* (see the ATM forum web site for details: [www.atmforum.com](http://www.atmforum.com)). Two primary functions are involved in transferring data on the UTOPIA bus: polling the ports to determine which ones have data ready and then selecting which port transfers its data. The receive side is discussed below. Refer to the above referenced web page for more details.

The CN8236 starts by polling port 0 and continue to NUM\_PORTS as stored in the CONFIG1 register. Polling is accomplished by outputting the desired port number on the UTOPIA address bus. If that port has data ready it asserts the RxClav line. The controller can ignore this line and continue polling other ports or it can initiate the data transfer by holding the port address on the address lines while RxEnb\* is deasserted. Once the data transfer has begun, the CN8236 can continue polling other ports.

The polling and selection process waveforms are shown in [Figure 12-1](#). This diagram assumes that 8 ports are present. At clock cycle a, port 5 has just been polled. It asserts the RxClav line to indicate that it has a complete cell to send. The CN8236 continues to poll ports 6 and 7 (both indicate that they do not have cells at this time). During this entire polling operation, data is being transferred across the data bus.

At clock cycle b, the CN8236 has again output port 5 on the UTOPIA address bus but this time has raised the TxEnb\* line. This selects port 5. Port 5 acknowledges that it is ready by again asserting RxClav. Port 5 then asserts the RxSOC line and begins to transfer data.

Figure 12-1. UTOPIA Level 2 Receive Timing

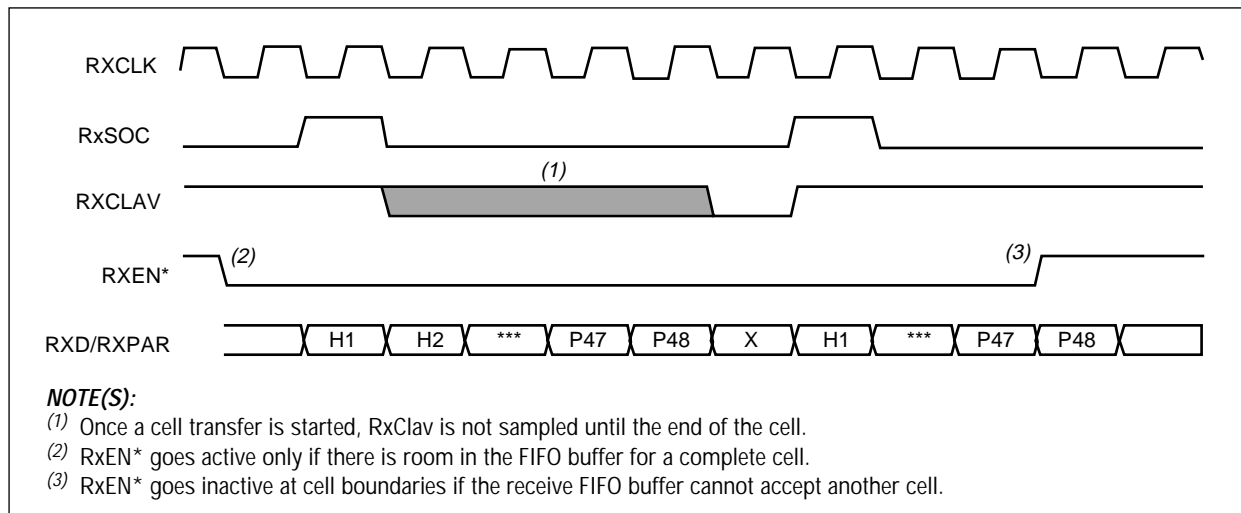


8236\_070

## 12.5 UTOPIA Level 1 Mode Cell Handshake Timing

If high, the UTOPIA\_MODE bit in the CONFIG0 register selects cell-level handshaking. Received data is latched from the RxDATA[15:0] and RxPar lines on the rising edge of RxClk after RxEN\* is sampled active (see Figure 12-2). The odd parity computed over the RxData[15:0] lines is compared to the RxPar input. If in error, the FR\_PAR\_ERR bit is set in the HOST\_ISTAT0/LP\_ISTAT0 registers. Data is discarded upon a parity error if the RSM\_PHALT bit in the RSM\_CTRL register is set to a logic high, and the reassembly coprocessor halts. The RxSOC signals the start of cell. The RxClav input is the physical layer FIFO buffer empty signal. When it is active, a complete cell is not present in the physical receive FIFO buffer. The physical layer device sets RxClav inactive when it has a complete cell to transfer. The CN8236 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 CN8236 samples the RxClav input. If low, the physical device does not have a cell to transfer. If RxClav is high, the physical device has another cell to transfer and the CN8236 immediately starts receiving the next cell if it can accept a complete cell. The FR\_RMODE bit in the CONFIG0 register should be set to a logic low in this mode.

Figure 12-2. Receive Timing in UTOPIA Level 1 Mode with Cell Handshake

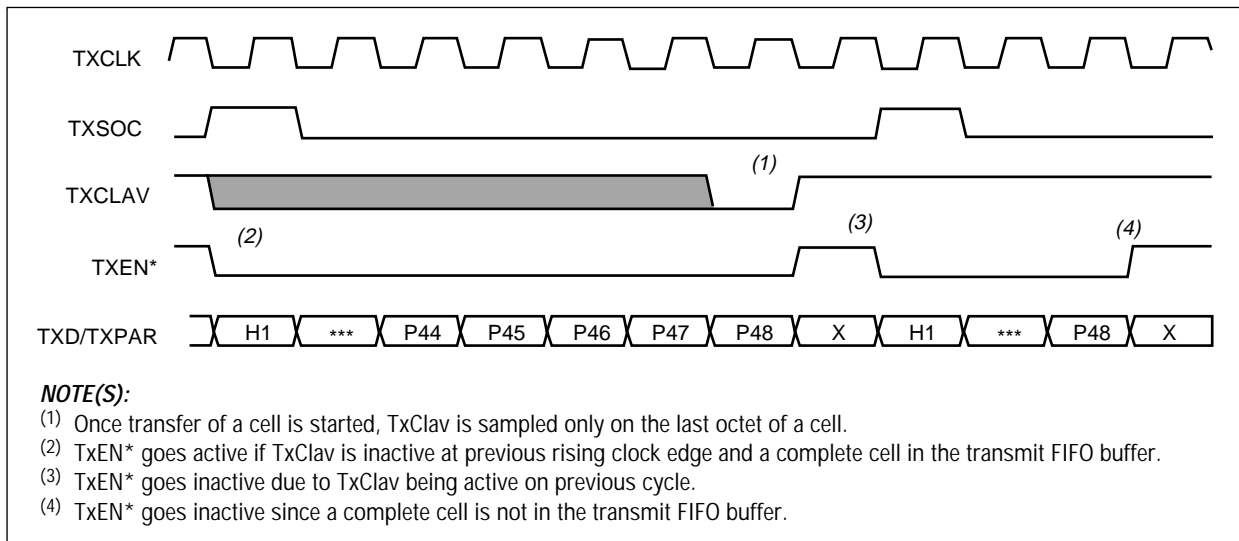


8236\_071

Transmit data is driven on TxData[15:0] on the rising edge of TxClk when TxEN\* is asserted. TxEN\* is only asserted when there is data in the CN8236 transmit FIFO buffer. Simultaneously, the odd parity computed over the TxData[15:0] 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 12-3.)

In UTOPIA mode, the TxClk input can be connected to the CN8236 CLKD3 output; a 50% duty cycle clock derived by dividing CLK2X by three.

Figure 12-3. Transmit Timing in UTOPIA Level 1 Mode with Cell Handshake



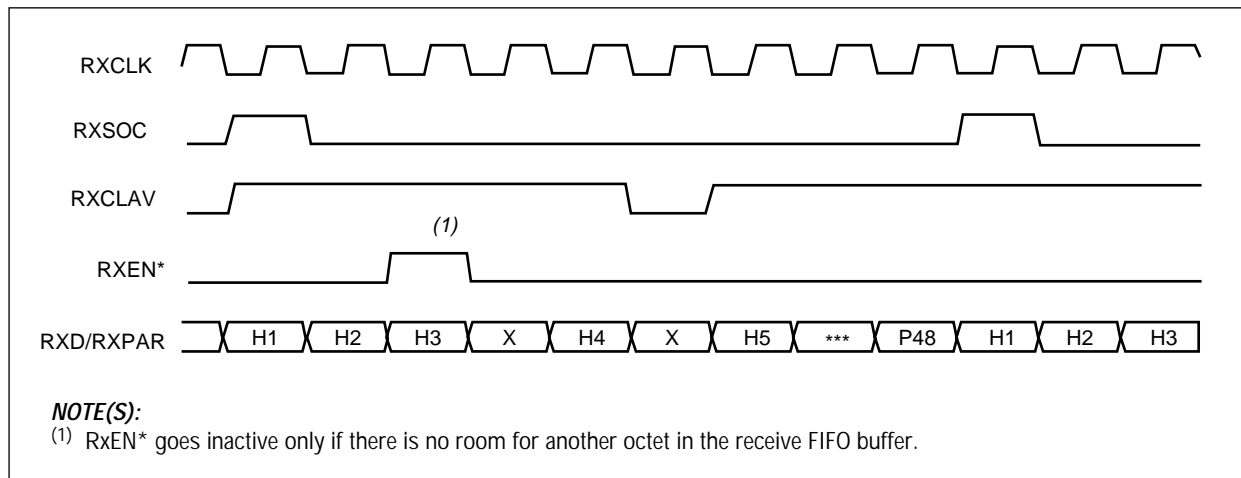
8236\_072

## 12.6 UTOPIA Level 1 Mode Octet Handshake Timing

If low, the UTOPIA\_MODE bit in the CONFIG0 register selects octet-level handshaking. Received data is latched from the RxData[15:0] and RxPar lines on the rising edge of RxClk after RxEN\* is sampled active (see Figure 12-4). The odd parity computed over the RxData[15:0] lines is compared to the RxPar input. If in error, FR\_PAR\_ERR in the HOST\_ISTAT0/LP\_ISTAT0 registers is set. Data is discarded upon a parity if the RSM\_PHALT bit in the RSM\_CTRL register is set to a logic high, and the reassembly coprocessor halts.

The RxSOC signals the start of cell to the CN8236. The RxClav input is the physical layer FIFO buffer empty signal. When it is active, no data is present in the physical receive FIFO buffer. The physical layer device sets RxClav inactive when it has an octet to transfer. The CN8236 sets RxEN\* to a logic low if it can accept an octet in the next clock cycle. The FR\_RMODE bit in the CONFIG0 register should be set to a logic low in this mode.

Figure 12-4. Receive Timing in UTOPIA Level 1 Mode with Octet Handshake



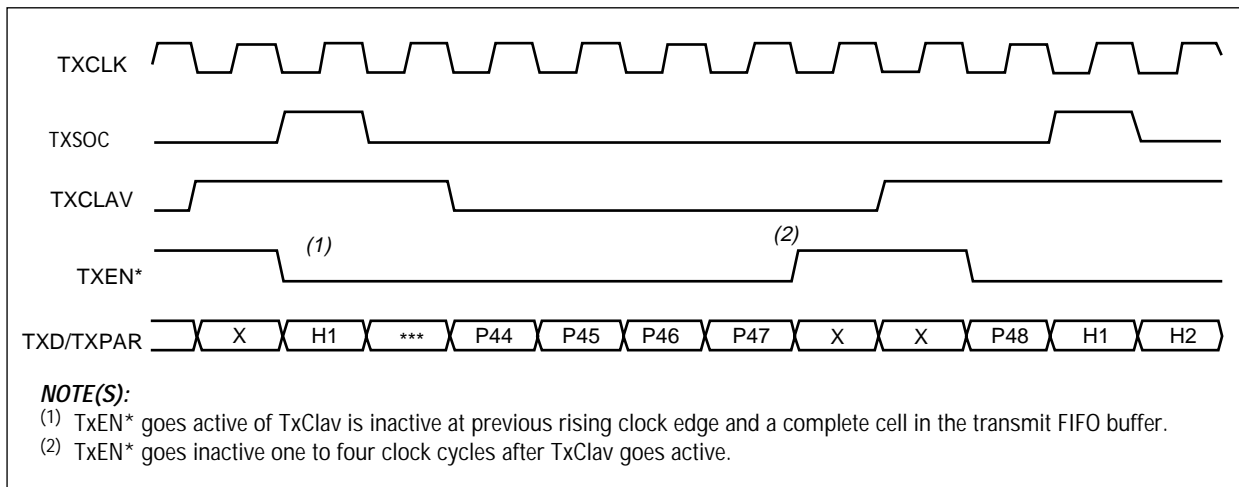
8236\_073



Transmit data is driven on TxData[15:0] on the rising edge of TxClk when TxEN\* is asserted. TxEN\* is only asserted when there is data in the CN8236 transmit FIFO buffer. Simultaneously, odd parity computed over the TxData[15:0] 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 12-5.)

In UTOPIA mode, the TxClk input can be connected to the CN8236 CLKD3 output, which is a 50% duty cycle clock derived by dividing the CLK2X input by three.

Figure 12-5. Transmit Timing in UTOPIA Level 1 Mode with Octet Handshake



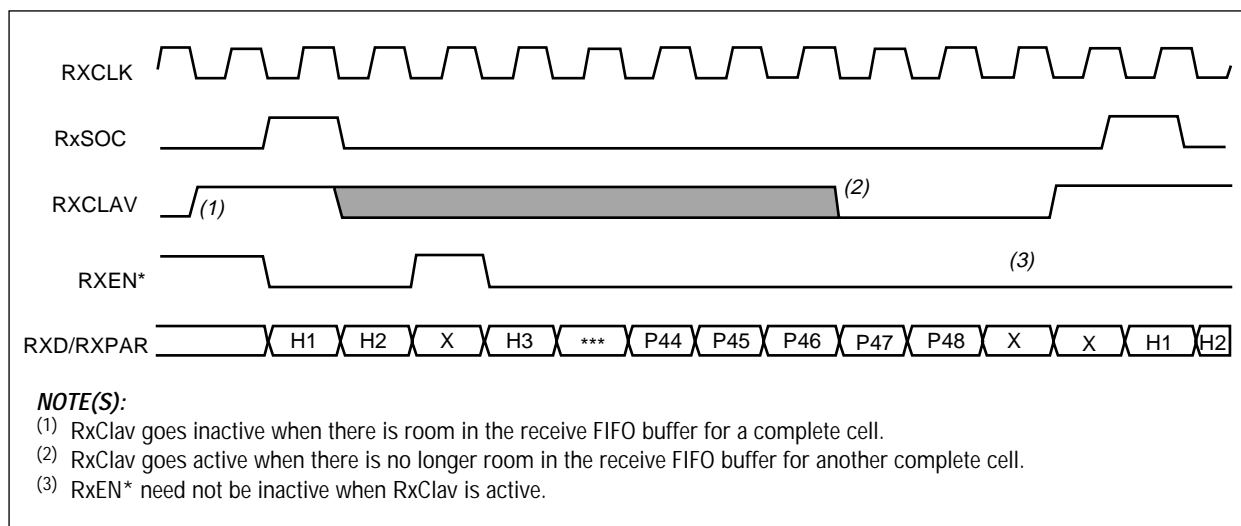
8236\_074

## 12.7 Slave Level 1 UTOPIA Mode

The slave UTOPIA mode is similar to the UTOPIA mode, except the direction of the enable signals and FIFO buffer 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 CN8236. The TxFull\* signal is changed to the TxEmpty\* signal and is an output of the CN8236. The RxEmpty\* signal is changed to the RxFull\* signal, and is also an output of the CN8236. This mode supports only a cell-level handshake protocol.

Received data is latched from the RxData[15:0] and RxPar lines on the rising edge of RXCLK when RxEN\* is active (see Figure 12-6). The odd parity computed over the RxData[15:0] 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. Data is discarded upon a parity error if 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 CN8236 the start of cell. The RxClav output is the receive FIFO buffer full signal. When it is active, the CN8236 cannot accept another cell. The CN8236 sets RxClav inactive when it has room in the receive FIFO buffer for another cell. The physical device sets RxEN\* to a logic low if it can transfer an octet. The FR\_RMODE bit in the CONFIG0 register should be set to a logic low in this mode.

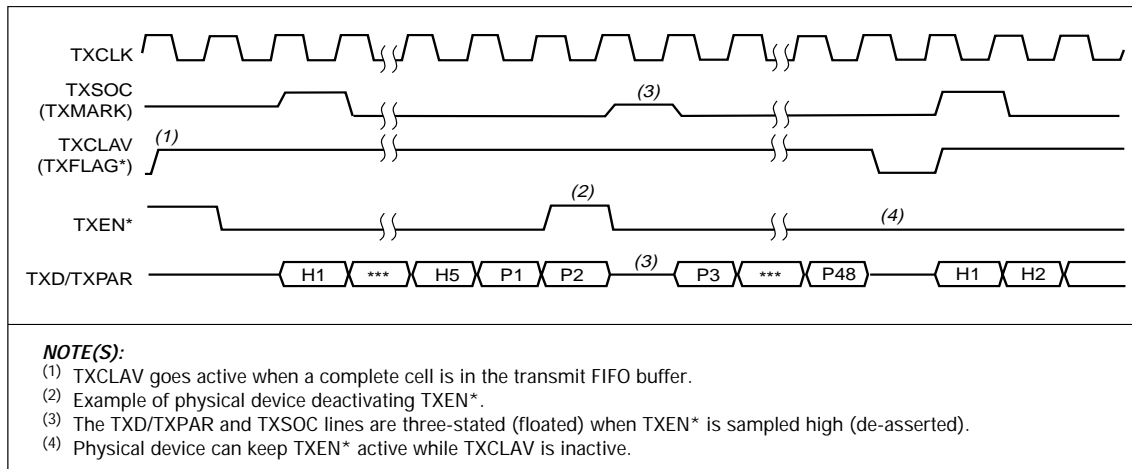
Figure 12-6. Receive Timing in Slave UTOPIA Level 1 Mode



8236\_075

Transmit data is driven on TxData[15:0] on the rising edge of TXCLK after TxEN\* is sampled asserted. Simultaneously, the 8-bit odd parity computed over the TxData[15:0] lines is driven on to the TxPar output. The TxSOC line is driven by the SAR to indicate start of cell. If the TxClav output is asserted by the CN8236, the transmit FIFO buffer does not contain a complete cell. (See [Figure 12-7.](#))

Figure 12-7. Transmit Timing in Slave UTOPIA Level 1 Mode



8236\_076

## 12.8 Loopback Mode

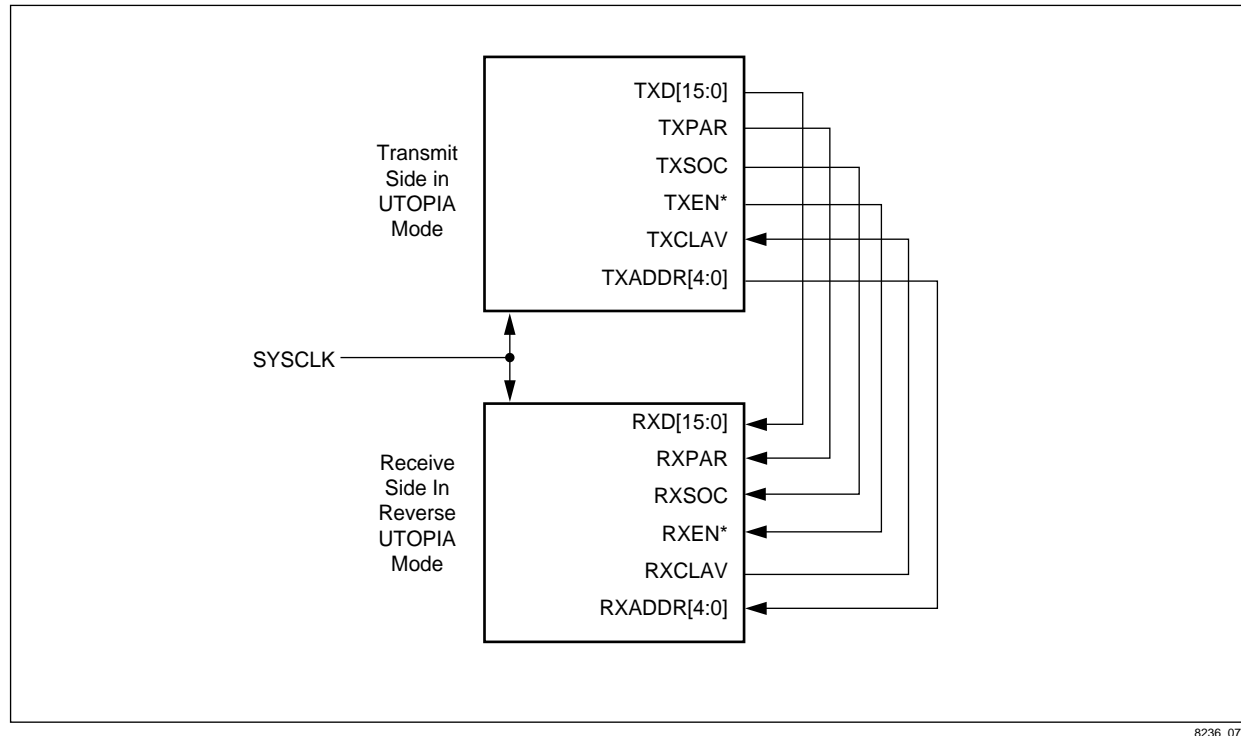
The physical interface can be internally looped by setting the FR\_LOOP bit in the CONFIG0 register to a logic high. This mode uses the internal system clock for operation; therefore, a framer clock is not needed during loopback operation.

When the FR\_SRC\_LOOP signal equals 1, the interface loops back the data and control signals so the path from the segmentation coprocessor to the reassembly coprocessor can be tested. The internal connections of the PHY device interface signals are connected, as Figure 12-8 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, TxData, 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 12-8. Source Loopback Mode Diagram



## 12.9 Receive Cell Synchronization Logic

The receive cell synchronization logic accepts a stream of octets (together with error and cell boundary indications) from the receive ATM UTOPIA 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 UTOPIA 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), that is, the first byte received is the MSB of the word.
- Ensures that a complete cell (exactly 52 bytes) is always written to the FIFO buffer. If a synchronization error occurs, the FR\_SYNC\_ERR bit in the HOST\_ISTAT0/LP\_ISTAT0 registers is set. The ATM UTOPIA interface attempts to re-synchronize 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 buffer being full.

## 12.10 Transmit Cell Synchronization Logic

The transmit cell synchronization logic copies cell data from the transmit cell FIFO buffer to the transmit ATM UTOPIA interface while performing the following functions:

- Reads 32-bit words from the transmit cell FIFO buffer 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-0) 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 UTOPIA 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 UTOPIA interface unit, with cell delineation pulses at the starting byte of every cell. Only complete 53-byte cells are supplied to the ATM UTOPIA interface. If the transmit cell FIFO buffer is empty, the transmit cell synchronization logic indicates that no more data can be transferred to the framer.

## 13.0 AALx Interworking

---

The AALx is a programmable platform that supports development of various voice processing algorithms. Header processing is required since the cells from various channels are streamed to or from one FIFO buffer in the AALx. The function of the SAR is to integrate AALx and Host Processor traffic (that is, management) and to provide traffic shaping in network centric scheduling applications. This solution supports up to six AALx parts.

PCI Masters communicate with AALxs through either a FIFO buffer or mailbox registers. In order for the SAR to communicate with the AALx FIFO buffer, the RSM and SEG need to be configured in 52-octet logical FIFO buffer mode, and the FIFO buffer on the AALx must be in slave access mode. On the ingress side, the RSM block performs ATM header lookup on each cell and steers the voice data cells to the appropriate AALx. In addition, it determines the state of the AALx ingress FIFO buffer and drop cells if full.

On the egress side, the SEG block supports two scheduling modes, network centric scheduling and voice centric scheduling. In network centric scheduling, the scheduling table can be locked to the network cell rate. CBR entries are written for each AALx. When a transmit opportunity is detected, the SEG block determines the state of the appropriate AALx egress FIFO buffer and retrieves a cell if available. In voice centric scheduling, the SEG block constantly determines the state of each AALx egress FIFO buffer and retrieves and transmits a cell as soon as one is available.

The transmission of a cell bumps management cells scheduled by the SAR. To better support network-centric scheduling, an external scheduler reference clock is supplied. In addition, the PHYs add a cell slot reference output clock. In order for the RSM to determine the state of the AALx ingress FIFO buffer, the AALx toggles a GPIO output every time it reads a cell from the FIFO buffer. This output is connected to the HFIFORDx input of the SAR. The RSM block maintains a shadow FIFO buffer read counter and can determine if there is room to accept another cell. Similarly, the AALx toggles another GPIO output every time it writes a cell to the egress FIFO buffer. This output is connected to the HFIFOWRx input. The SEG maintains a shadow FIFO buffer write counter to determine that a cell is available. The ingress and egress FIFO buffer shadow counters have a programmable FIFO buffer depth up to 16 cells via the AALx\_CTRL register.

The header word of each contains the VCC\_INDEX of the channel. The format of the word is as follows in Tables 13-1 and 13-2:

SEG:

Reserved (12 bits) | SEG\_VCC\_INDEX (16) | PTI (3) | CLP (1)

Table 13-1. SEG\_VCC\_INDEX Format Table

| 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 |    |    |    |    |    |    |    |    |    |    |    | SEG_VCC_INDEX |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | PTI |   |   | CLP |

RSM:

RSM\_ROUTE\_TAG (28) | PTI (3) | CLP (1)

Table 13-2. RSM\_ROUTE\_TAG Format Table

| 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    | RSM_ROUTE_TAG |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | PTI |   |   | CLP |

On reassembly, the VPI and VCI are stripped from the ATM header word, and the RSM\_ROUTE\_TAG is added. This eliminates the need for a lookup algorithm in the AALx and still allows the AALx to perform AAL5 processing since the PTI field is passed. On the segmentation side, the same header word is passed from the AALx to the SAR. The SEG block can use the VCC\_INDEX to determine the routing tag and also to support PM-OAM in the future. The PTI field is passed to the cell without change; however, the CLP bit is ORed with the CLP bit in the ATM HEADER field of the VCC table entry.

In order to generate management traffic and to configure the AALx PCI registers, a central PCI resource is required. This consists of a processor and central PCI bridge chip.

*NOTE:* PM-OAM operation does not work with this configuration.



## 13.1 AALx RSM Operation

To support AALx operation, HPORT\_ID and AALx\_EN are added to the reassembly VCC table entry as follows: AALx\_EN is added to bit 21 of word 0. When this bit is set high, the DPRI field is used as the HPORT\_ID. For proper operation with the AALx part, the channel must be set up per logical FIFO buffer mode (FIFO\_EN = 1), that is, AAL0 fixed termination mode with termination length of one cell. Also, M52\_EN must be a logic high. In addition, a RSM\_ROUTE\_TAG is written into the upper 28 bits of the BOM\_BD\_PNTR word on the RSM VCC table entry.

Six shadow FIFO buffer counters, corresponding to each HFIFORDx input, is maintained. When the ingress FIFO buffer is written, the counter is incremented. An edge detection circuit decrements each counter. The counter size is programmable via the AALx\_CTRL(INGRESS\_DEPTH) register up to 16 cells. Each counter is reset by either RSM\_RESET or a separate reset per port.

When a write occurs on a full condition, HRSMOVFLx output pulses to a logic high for one clock period. When a read occurs on an empty condition, HRSMUNFLx output pulses to a logic high for one clock period. Rollover or rollunder of the counter must be prevented.

When a cell is received, the normal cell lookup process occurs. If AALx\_EN is active in the VCC table entry, DPRI is used as the HPORT\_ID. The RSM block checks if room is available in the appropriate AALx FIFO buffer. If so, the FIFO buffer address in the VCC table entry is used to transfer the complete cell (52 octet mode). If not, the cell is discarded. The ATM header word is modified to pass the RSM\_ROUTE\_TAG as follows: RSM\_ROUTE\_TAG (28 bits) | PTI (3) | CLP (1).

## 13.2 AALx SEG Operation

To support AALx operation, HPORT\_ID and AALx\_EN are added to the segmentation VCC table entry as follows: AALx\_EN is added to bit 21 of word 5 and HPORT\_ID is added to word 5 (of a CBR mode connection) bits 26 through 30. HPORT\_ID is only used in network centric scheduling (that is, EXTERNAL\_SCH = 0). (The PCI addresses for each AALx are stored in internal memory as shown in [Table 4-6](#).)

Six shadow FIFO buffer counters, corresponding to each HFIFOWRx input, are maintained. An edge detection circuit increments each counter. The counter is decremented after a read of the cell. A non 0 count indicates a cell is available for transmit from the corresponding AALx. The counter size is programmable up to 16 cells via the AALx\_CTRL(EGRESS\_DEPTH) register.

Each counter is reset by either SEG\_RESET or a separate reset per port. When a write occurs on a full condition, HSEGOVFLx output pulses to a logic high for one clock period. The counter should prevent rollover or roll-under.

### 13.2.1 AALx Network Centric Operation—(EXTERNAL\_SCH = 0)

In this mode, the AALx traffic stream is scheduled using a CBR connection, with bandwidth reserved in the schedule table. The Segmentation block, processing a CBR connection from the Scheduler with AALx\_EN set high, looks up the HPORT\_ID field to identify which HFIFOWRx counter check to determine if a cell is available from that AALx. If it is, the PCI address for that AALx is looked up from the corresponding internal memory address and the entire cell (52 bytes) with modified ATM Header word is read across the PCI bus. This process is similar to the virtual FIFO buffer scheduling mode as exists today. However, no rate matching using the SCH\_OPT bit is required, the CURR\_PNTR field is a Don't Care, and the header is included in the cell read across the PCI bus, as opposed to being read from the VCC table.

To set up a connection for the AALx traffic stream in this mode, the user does the following:

- Populates the schedule table with the connection's VCC index.
- Sets the connection's VCC table entries: AALx\_EN = 1, HPORT\_ID, and SCH\_MODE = CBR.
- Writes the AALx's FIFO buffer PCI address to internal memory.
- Writes the ATM header into VCC entry.
- To enable the AALx traffic stream to be transmitted, sets the connection's VCC table RUN bit high.

### 13.2.2 AALx Voice Centric Operation—(EXTERNAL\_SCH = 1)

No scheduling is performed by the Scheduler block in this mode. For each processing loop through its state machine, the Segmentation block monitors the cell-available counters (priority is round-robin) to determine if any have data available for transmission. If so, the PCI address for the corresponding AALx is looked up from internal memory, and the entire cell (52 bytes) with modified ATM Header word is read across the PCI bus.

No connection setup is required in this mode; the user must do the following:

- Write the AALx's FIFO buffer PCI address to internal memory.
- Write ATM header into VCC entry.

The AALx traffic stream is enabled as the AALx strobes HFIFOWRx.



# 14.0 CN8236 Registers

---

## 14.1 Control and Status Registers

Each CN8236 register description is prefaced with the appropriate abbreviated access types described in [Table 14-1](#). Detailed control and status register (CSR) descriptions are listed in [Table 14-2](#).

*NOTE:* The byte-enables are ignored by the CN8236 when writing to control and status registers.

The access type terminology given in [Table 14-1](#) applies to all registers in this section.

*Table 14-1. Type Abbreviation Description*

| Abbreviation | Description  |
|--------------|--|
| R/W          | Read and write access for both host and local processors               |
| R/W H        | Read and write access for host; read only access for local processor   |
| R/W L        | Read and write access for local; read only access for host processor   |
| R/W R        | Read and write access for both processors with restrictions            |
| R/O-W/O B    | Part of this register is read only; part write only by both processors |
| R/O          | Read only access for both processors                                   |

## Register Terminology

Table 14-2. CN8236 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  | Basic Configuration and Control register 0             |
| 0x18      | CONFIG1       | R/W  | Basic Configuration and Control register 1             |
| 0x1c      | INT_DELAY     | R/W  | Interrupt Delay register                               |
| 0x20      | AALx_CTRL     | R/W  | AALx Control register                                  |
| 0x24–0x7c | Reserved      | —    | Not Implemented  |
| 0x80      | SEG_CTRL      | R/W  | Segmentation Control register                          |
| 0x84      | SEG_VBASE     | R/W  | SEG VCC Table and Schedule Table Base Address register |
| 0x88      | SEG_PMBASE    | R/W  | SEG PM Table and Bucket Table Base Address register    |
| 0x8c      | SEG_TXBASE    | R/W  | Segmentation Transmit Queue Base register              |
| 0x90      | SEG_TAGBASE   | R/W  | Base Address of Routing Tag Table                      |
| 0x94–0x9c | Reserved      | —    | Not Implemented  |
| 0xa0      | SCH_PRI       | R/WB | Schedule Priority Queue Control register 1             |
| 0xa4      | SCH_PRI_2     | R/W  | Schedule Priority Queue Control register 2             |
| 0xa8      | SCH_SIZE      | R/W  | Schedule Size and Slot Minimum Drain Rate register     |
| 0xac      | SCH_CTRL      | R/W  | Scheduler Control register                             |
| 0xb0      | SCH_ABR_MAX   | R/W  | Maximum ABR VCC_INDEX register                         |
| 0xb4      | SCH_ABR_CON   | R/W  | Schedule ABR Constant register                         |
| 0xb8      | SCH_ABRBASE   | R/W  | ABR Decision Table Lookup Base register                |
| 0xbc      | SCH_CNG       | R/W  | ABR Congestion Notification 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_FQBASE    | R/W  | Reassembly Free Buffer Queue Base register             |
| 0xfc      | RSM_FQCTRL    | R/W  | Reassembly Free Buffer Queue Control register          |
| 0x100     | RSM_TBASE     | R/W  | Reassembly Table Base register                         |

Table 14-2. CN8236 Control and Status Registers (2 of 2)

| Address     | Name          | Type  | Description                                 |
|-------------|---------------|-------|---|
| 0x104       | RSM_TO        | R/W   | Reassembly Time-out register                |
| 0x108       | RS_QBASE      | R/W   | Reassembly/Segmentation Queue Base register |
| 0x10C       | ERS_BASE      | R/W   | Reassembly ER_SHIFT Tables Base register    |
| 0x110       | VPI_SIZE      | R/W   | Variable VPI Size register                  |
| 0x114       | TX_FIFO_CTRL  | R/W   | Transmit Port Register                      |
| 0x118       | TX_PORT_CTRL  | R/W   | Transmit Port Control register              |
| 0x11C–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_DSC_CNT  | R/O   | AAL5 PDUs Discarded Counter                 |
| 0x170–0x19C | Reserved      | —     | Not Implemented                             |
| 0x1a0       | HOST_MBOX     | R/W L | Host Mailbox register                       |
| 0x1a4       | HOST_ST_WR    | R/O   | Host Status Write register                  |
| 0x1a8       | AALx_STAT     | R/O   | AALx Counter Status register                |
| 0x1ac       | TX_STATUS     | R/O   | Tx Port Cell Discard Status register        |
| 0x1b0       | LP_MBOX       | R/W H | Local Processor Mailbox register            |
| 0x1b4–0x1bc | 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 H | Host Interrupt Mask register 0              |
| 0x1d4       | HOST_IMASK1   | R/W H | Host Interrupt Mask register 1              |
| 0x1d8–0x1dc | Reserved      | —     | Not Implemented                             |
| 0x1e0       | LP_ISTAT0     | R/O   | Local Processor Interrupt Status register 0 |
| 0x1e4       | LP_ISTAT1     | R/O   | Local Processor Interrupt Status register 1 |
| 0x1e8–0x1ec | Reserved      | —     | Not Implemented                             |
| 0x1f0       | LP_IMASK0     | R/W L | Local Processor Interrupt Mask register 0   |
| 0x1f4       | LP_IMASK1     | R/W L | Local Processor Interrupt Mask register 1   |
| 0x1f8–0x1fc | Reserved      | —     | Not Implemented                             |

## 14.2 System Registers

### 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 each processor, and can generate an interrupt on the RTC\_OVFL status bit in the LP\_ISTAT0 register when it overflows from 0xFFFFFFFF to 0x0.

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

### 0x04—Alarm Register 1 (ALARM1)

This register contains a 32-bit value which is compared against the CLOCK register. When the two registers are equal, the ALARM1 status bit in the LP\_ISTAT0 register is latched and can be enabled to cause an interrupt to the local processor. To implement a periodic interrupt, add a constant value to this register after each interrupt.

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



## 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, and 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 and LP\_ISTAT0 registers.

| Bit   | Field Size | Name                 | Description  |
|-------|------------|----------------------|--|
| 31–17 | 15         | Reserved             | Not implemented at this time.  |
| 16–12 | 5          | PCI_BUS_STATUS [4:0] | The status bits are as follows:<br>4 = Target Abort<br>3 = Master Abort<br>2 = Parity Error<br>1 = Interface Disabled<br>0 = Internal Failure<br>Reflects corresponding error bits in the PCI Configuration register. Bits are reset by either a write to the PCI Configuration register by the host, or by setting CONFIG0 (PCI_ERR_RESET) bit. |
| 11    | 1          | RAMMODE              | Reflects the state of the RAMMODE input pin.   |
| 10    | 1          | PROCMODE             | Reflects the state of the PROCMODE input pin.  |
| 9, 8  | 2          | FRCFG[1:0]           | Reflects the state of the FRCFG[1:0] input pins.   |
| 7–4   | 4          | VERSION [3:0]        | Version number for the CN8236: Rev A = 0 and Rev B = 2.  |
| 3–0   | 4          | DEVICE[3:0]          | Device ID for the CN8236; set to 4.  |

## 0x14—Configuration Register 0 (CONFIG0)

This register provides all control and configuration bits that are not associated with the reassembly and segmentation coprocessors. The majority of these configuration bits are set at initialization time and are not changed dynamically. The assertion of the HRST\* system reset pin clears all of the bits in the CONFIG0 register except for MEMCTRL, which is 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 and 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    | 1          | Reserved               | Always set to 0.   |
| 26    | 1          | 8223_MODE              | When set, enables modified microprocessor interface timing to PHY.   |
| 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 or local processor.                 |
| 23    | 1          | PCI_WR_RD              | When this bit is high, the PCI Master Arbitration Scheme is set to write priority over read. This bit takes precedence over PCI_ARB (bit 21).  |
| 22    | 1          | PCI_READ_MULT          | 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<br>(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 buffer 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.<br>0 = Octet handshake<br>1 = Cell handshake   |
| 12    | 1          | ENDIAN                 | Selects between Little and Big Endian host data structures.<br>0 = Little Endian<br>1 = Big Endian   |

| Bit  | Field Size | Name          | Description   |
|--|------------|---------------|---|
| 11   | 1          | LP_BWAIT      | Selects 0 or 1 wait states between consecutive data cycles during local processor burst accesses. Set to logic low for standalone operation mode. |
| 10   | 1          | MEMCTRL       | Selects 0 or 1 wait states SAR-shared memory (1- or 2-cycle). <sup>(1)</sup>  |
| 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]  | Pre-scaler for SYSCLK which advances the counter in the CLOCK register. SYSCLK is divided by the divider value; if 0, divided by 128.             |
| <b>NOTE(S):</b><br><sup>(1)</sup> If 1 wait state is selected, PWAIT function does not work. |            |               |   |

## 0x18—Configuration Register 1 (CONFIG1)

This register provides system control and configuration bits that are not directly associated with the reassembly and segmentation coprocessors. The majority of these bits are configuration bits (which occur at initialization time) and are not changed dynamically. The assertion of the HRST\* system reset pin clears all bits in the CONFIG1 register for backward compatibility.

| Bit   | Field Size | Name             | Description  |
|-------|------------|------------------|--|
| 31–24 | 8          | Reserved         | Set to 0.  |
| 23    | 1          | MULTI_CLK        | If logic high, selects separate UTOPIA transmit and receive clocks. If logic low, both clocks use the RxClk input.   |
| 22    | 1          | MULTI_PHY        | If logic high, multi-PHY operation is enabled.   |
| 21    | 1          | UTOP16           | If logic high, UTOPIA 16 bit interface is enabled; otherwise 8-bit interface.  |
| 20–18 | 3          | NUM_PORTS[2:0]   | Number of PHY ports to poll when in Master UTOPIA Mode starting with address 0. Number of ports = (NUM_PORTS + 1)  |
| 17–13 | 5          | SLAVE_ADDR[4:0]  | When in Slave UTOPIA Multi-PHY Mode, this is the UTOPIA device address of the SAR. When in Master Non-Multi-PHY Mode, this is the address present on both TxAddr and RxAddr. |
| 12–9  | 4          | TAG_SIZE[3:0]    | Select Tag size. Valid range is 0 to 11. In 16 bit mode (UTOP16 = 1), only even size tags are valid.   |
| 8–4   | 5          | PHYBANK[4:0]     | Physical Chip Bank Select. This value is placed on LADDR[13:9] when a PHY1 or PHY2 control access occurs.  |
| 3     | 1          | Reserved         | Set to 0.  |
| 2     | 1          | TX_FIFO_FLUSH_EN | Enables TX_FIFO_FLUSH mechanism. This mechanism is valid only in UTOPIA master and multi-PHY modes.  |
| 1     | 1          | INCFIFO_SZ       | Incoming DMA FIFO buffer size. Logic high sets the FIFO buffer to 8 KB, logic low to 2 KB.   |
| 0     | 1          | NEW_PMOAM        | When a logic high, the NEW_PMOAM mechanism is enabled.   |

**0x1c—Interrupt Delay Register (INT\_DELAY)**

| Bit   | Field Size | Name          | Description  |
|-------|------------|---------------|--|
| 31–11 | 21         | Reserved      | Set to 0.  |
| 10    | 1          | TIMER_LOC     | If logic high, interrupt hold-off timer used with local interrupt; else with host interrupt. |
| 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. |

**0x20—AALx Control Register (AALx\_CTRL)**

| Bit   | Field Size | Name                      | Description  |
|-------|------------|---------------------------|--|
| 31–30 | 2          | Reserved                  | Set to 0.  |
| 29–24 | 6          | RST_INGRESS_FIFO<br>[5:0] | If logic high, resets the AALx ingress FIFO buffer shadow counter.                                   |
| 23–22 | 2          | Reserved                  | Set to 0.  |
| 21–16 | 6          | RST_EGRESS_FIFO<br>[5:0]  | If logic high, resets the AALx egress FIFO buffer shadow counter.                                    |
| 15–12 | 4          | Reserved                  | Set to 0.  |
| 11–8  | 4          | INGRESS_DEPTH             | Depth of all ingress FIFO buffers in terms of number of cells.<br>Actual depth = register value + 1. |
| 7–4   | 4          | Reserved                  | Set to 0.  |
| 3–0   | 4          | EGRESS_DEPTH              | Depth of all egress FIFO buffers in terms of number of cells.<br>Actual depth = register value + 1.  |

## 14.3 Segmentation Registers

### 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 causes 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 halts on a cell boundary.  |
| 30    | 1          | SEG_RESET     | Segmentation Reset—resets segmentation coprocessor and pointers.   |
| 29–27 | 3          | VBR_OFFSET    | Offset from schedule slot priority to general priority. (VBR_OFFSET + (# VBR / ABR priorities) ≤ 7.) Not active if USE_SCH_CTRL is asserted.   |
| 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. Not active if USE_SCH_CTRL is asserted.   |
| 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    | 1          | USE_SCH_CTRL  | Activate the use of SLOT_DEPTH, the 4-bit VBR_OFFSET field, and TUN_PRIQ_OFFSET from the SCH_CTRL register. Deactivate the use of DBL_SLOT and the 3-bit VBR_OFFSET field from the SEG_CTRL register. This bit cannot be set to a logic high in 8235 mode.   |
| 21–16 | 6          | Reserved      | Program and read as 0.   |
| 15–12 | 4          | TX_FIFO_LEN   | Depth of transmit FIFO buffer in cells. Valid range is 3–9. To ensure optimum performance, the depth of the FIFO buffer should be at least three.  |
| 11    | 1          | CLPO_EOM      | Set CLP in ATM header to 0 on last cell of CPCS-PDU.   |
| 10–6  | 5          | OAM_STAT_ID   | Status queue ID for buffer descriptors with OAM_STAT set.  |
| 5     | 1          | SEG_ST_HALT   | Enables a status queue entry for a VCC halted with a partially segmented 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 does not segment any cells for a VCC assigned to a full SAR-shared memory status queue. This bit can be used to disable overflow checking when the queues are sized large enough to prevent overflow. |

| Bit | Field Size | Name         | Description   |
|-----|------------|--------------|---|
| 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 does not segment any cells for a VCC assigned to a full PCI memory status queue. This bit can be used to disable overflow checking when the queues are sized large enough to prevent overflow. |
| 2   | 1          | TX_RND       | Set for transmit queue servicing 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.<br>00 = 64<br>01 = 256<br>10 = 1,024<br>11 = 4,096  |

### 0x84—Segmentation VCC Table and 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. |

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

The SEG\_PMBASE register sets the base address in SAR-shared memory for the VBR bucket table and the performance monitoring table. Both base addresses are 128-byte aligned, and only the 16 most significant bits of the address are specified in the SEG\_PMBASE register.

| Bit   | Field Size | Name           | Description   |
|-------|------------|----------------|---|
| 31–16 | 16         | SEG_BCKB[15:0] | Base address for the VBR bucket table. (See <a href="#">Section 6.2.4.4</a> , for details on loading bucket table entries.) |
| 15–0  | 16         | SEG_PMB[15:0]  | Base address for the performance monitoring table.  |

### 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 0.                           |
| 12–5  | 8          | XMIT_INTERVAL[7:0] | Interval for transmit queue READ_UD_PNTR update. |
| 4–0   | 5          | TX_EN              | Transmit queues 0-TX_EN are enabled.             |

### 0x90—Segmentation Routing Tag Table Base Register (SEG\_TAGBASE)

| Bit   | Field Size | Name           | Description                             |
|-------|------------|----------------|---|
| 31–16 | 16         | Reserved       | Always set to 0.                        |
| 15–0  | 16         | SEG_TAGB[15:0] | Base address for the routing tag table. |

## 14.4 Scheduler Registers

### 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 0.                          |
| 22  | 1          | TUN_ENA7  | Enable tunnel on global priority pointer 7.     |
| 21  | 1          | GFC7      | Enable GFC on priority pointer 7.               |
| 20  | 1          | Reserved  | Program and read as 0.                          |
| 19  | 1          | TUN_ENA6  | Enable tunnel on global priority pointer 6.     |
| 18  | 1          | GFC6      | Enable GFC on priority pointer 6.               |
| 17  | 1          | Reserved  | Program and read as 0.                          |
| 16  | 1          | TUN_ENA5  | Enable tunnel on global priority pointer 5.     |
| 15  | 1          | GFC5      | Enable GFC on priority pointer 5.               |
| 14  | 1          | Reserved  | Program and read as 0.                          |
| 13  | 1          | TUN_ENA4  | Enable tunnel on global priority pointer 4.     |
| 12  | 1          | GFC4      | Enable GFC on priority pointer 4.               |
| 11  | 1          | Reserved  | Program and read as 0.                          |
| 10  | 1          | TUN_ENA3  | Enable tunnel on global priority pointer 3.     |
| 9   | 1          | GFC3      | Enable GFC on priority pointer 3.               |
| 8   | 1          | Reserved  | Program and read as 0.                          |
| 7   | 1          | TUN_ENA2  | Enable tunnel on global priority pointer 2.     |
| 6   | 1          | GFC2      | Enable GFC on priority pointer 2.               |
| 5   | 1          | Reserved  | Program and read as 0.                          |



| Bit | Field Size | Name     | Description                                 |
|-----|------------|----------|---|
| 4   | 1          | TUN_ENA1 | Enable tunnel on global priority pointer 1. |
| 3   | 1          | GFC1     | Enable GFC on priority pointer 1.           |
| 2   | 1          | Reserved | Program and read as 0.                      |
| 1   | 1          | TUN_ENA0 | Enable tunnel on global priority pointer 0. |
| 0   | 1          | GFC0     | Enable GFC on priority pointer 0.           |

### Oxa4—Schedule Priority Control Register 2 (SCH\_PRI\_2)

The SCH\_PRI\_2 register sets the enables for GFC, CBR tunneling, and PCR limits on global priority queues 8 through 15.

| Bit | Field Size | Name        | Description                                      |
|-----|------------|-------------|--|
| 31  | 1          | QPCR_ENA_15 | Enable PCR limits on global priority pointer 15. |
| 30  | 1          | QPCR_ENA_14 | Enable PCR limits on global priority pointer 14. |
| 29  | 1          | QPCR_ENA_13 | Enable PCR limits on global priority pointer 13. |
| 28  | 1          | QPCR_ENA_12 | Enable PCR limits on global priority pointer 12. |
| 27  | 1          | QPCR_ENA_11 | Enable PCR limits on global priority pointer 11. |
| 26  | 1          | QPCR_ENA_10 | Enable PCR limits on global priority pointer 10. |
| 25  | 1          | QPCR_ENA_9  | Enable PCR limits on global priority pointer 9.  |
| 24  | 1          | QPCR_ENA_8  | Enable PCR limits on global priority pointer 8.  |
| 23  | 1          | Reserved    | Program and read as 0.                           |
| 22  | 1          | TUN_ENA_15  | Enable tunnel on global priority pointer 15.     |
| 21  | 1          | GFC15       | Enable GFC on global priority pointer 15.        |
| 20  | 1          | Reserved    | Program and read as 0.                           |
| 19  | 1          | TUN_ENA_14  | Enable tunnel on global priority pointer 14.     |
| 18  | 1          | GFC14       | Enable GFC on global priority pointer 14.        |
| 17  | 1          | Reserved    | Program and read as 0.                           |
| 16  | 1          | TUN_ENA_13  | Enable tunnel on global priority pointer 13.     |
| 15  | 1          | GFC13       | Enable GFC on global priority pointer 13.        |
| 14  | 1          | Reserved    | Program and read as 0.                           |
| 13  | 1          | TUN_ENA_12  | Enable tunnel on global priority pointer 12.     |
| 12  | 1          | GFC12       | Enable GFC on global priority pointer 12.        |
| 11  | 1          | Reserved    | Program and read as 0.                           |
| 10  | 1          | TUN_ENA_11  | Enable tunnel on global priority pointer 11.     |
| 9   | 1          | GFC11       | Enable GFC on global priority pointer 11.        |

| Bit | Field Size | Name       | Description                                  |
|-----|------------|------------|--|
| 8   | 1          | Reserved   | Program and read as 0.                       |
| 7   | 1          | TUN_ENA_10 | Enable tunnel on global priority pointer 10. |
| 6   | 1          | GFC10      | Enable GFC on global priority pointer 10.    |
| 5   | 1          | Reserved   | Program and read as 0.                       |
| 4   | 1          | TUN_ENA_9  | Enable tunnel on global priority pointer 9.  |
| 3   | 1          | GFC9       | Enable GFC on global priority pointer 9.     |
| 2   | 1          | Reserved   | Program and read as 0.                       |
| 1   | 1          | TUN_ENA_8  | Enable tunnel on global priority pointer 8.  |
| 0   | 1          | GFC8       | Enable GFC on global priority pointer 8.     |

### 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 0.  |
| 13–0  | 14         | SLOT_PER[13:0] | Number of system clocks per schedule slot. The value written to the register should be (SLOT_PER – 1). Minimum bound for SLOT_PER break value = 70. |

## 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   |
|-------|------------|-----------------|---|
| 31–27 | 5          | Reserved        | Program and read as 0.  |
| 26    | 1          | USE_SCHREF      | If logic high, the SCHREF input is used as the clock for defining a schedule table slot period in conjunction with SLOT_PER. If logic low, SYSClk is used.<br><br><b>NOTE: Must be set to 0 during initialization unless using an external scheduler clock.</b> |
| 25    | 1          | EXTERNAL_SCH    | If logic high, scheduling priority is granted to externally scheduled traffic. See <a href="#">Chapter 13.0</a> .   |
| 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          | SLOT_DEPTH      | Depth of the schedule slot is set to 1 + SLOT_DEPTH words. Active only if USE_SCH_CTRL is asserted.   |
| 11–10 | 2          | Reserved        | Program and read as 0.  |
| 9–6   | 4          | TUN_PRI0_OFFSET | Offset from the TUN_PRI0 field in the schedule table and CBR VCC table. Active only if USE_SCH_CTRL is asserted.  |
| 5–4   | 2          | Reserved        | Program and read as 0.  |
| 3–0   | 4          | VBR_OFFSET      | Offset from schedule slot priority to general priority. Active only if USE_SCH_CTRL is asserted.  |

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

This register sets the maximum number of ABR VCCs being used and specifies the MAX\_BEHIND value. The MAX\_BEHIND value specifies the number of slots in the schedule table; the schedule table entry pointer can fall behind. MAX\_BEHIND should be set to 0 for all applications, except special multi-PHY DSL applications.

| Bit   | Field Size | Name     | Description |
|-------|------------|----------|-------------|
| 31–24 | 8          | Reserved | Set to 0.   |

| Bit   | Field Size | Name       | Description   |
|-------|------------|------------|---|
| 23–16 | 8          | MAX_BEHIND | Number of slots Scheduler can fall behind. For normal operations, this value should be 0. (A value of 0 results in the default value of 255 internally used.) For multi-PHY DSL operations, this value should be set to the number of port used e.g., if 8 ports are used, this MAX_BEHIND should be set to 16. |
| 15–0  | 16         | VCC_MAX    | Maximum ABR VCC index.  |

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

The SCH\_ABR\_CON register sets the ABR TRM and ADTF time-out. Time-out values are in units of 64 cell slots.

| Bit   | Field Size | Name     | Description   |
|-------|------------|----------|---|
| 31–16 | 16         | ABR_TRM  | ABR TRM parameter.<br>Parameter value is SYSCLK period × SLOT_PER × ABR_TRM × 64.   |
| 15–0  | 16         | ABR_ADTF | ABR ADTF parameter.<br>Parameter values is SYSCLK period × SOT_PER × ABR_ADTF × 64. |

Access types: R/W

### 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 0.  |
| 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.  |

Access Types: R/W

### 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. |

Access Types: R/W

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

The PCR\_QUE\_INT01 register fields are used to store the two lower PCR values used in PCR shaping on 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  $1 / (\text{QPCR\_INTx} \times \text{SYSCLK\_period} \times \text{SLOT\_PER})$ .

| 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). |

### 14.4.1 0xc4—PCR Queue Interval 2 and 3 Register (PCR\_QUE\_INT23)

The PCR\_QUE\_INT23 register fields are used to store the two highest PCR values used in PCR shaping on 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  $1 / (\text{QPCR\_INTx})$ .

| 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). |

## 14.5 Reassembly Registers

### 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 clears 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, that is, 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 is also 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 resumes normal operation.   |
| 29    | 1          | Reserved      | Program and read as 0.   |
| 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 4096.   |
| 27–24 | 4          | Reserved      | Program and read as 0.   |
| 23–18 | 6          | Reserved      | Program and read as 0.   |
| 17    | 1          | RSM_PHALT     | Reassembly coprocessor halt on parity error detect. The reassembly coprocessor halts the incoming channel logic if a parity error is detected and the RSM_PHALT bit is set.  |
| 16    | 1          | PREPEND_INDEX | Causes VCC_INDEX to be prepended to the BOM cell transfer.   |
| 15    | 1          | FWALL_EN      | Firewall enable. Enables free buffer queue firewalling of user cells. If set, this bit enables the per connection free buffer queue firewall. Each connection that firewall is active in must have the FW_EN bit set to a logic high in the VCC table.   |
| 14    | 1          | RSM_FBQ_DIS   | Free Buffer Queue Underflow Protection Disable. When a logic high, the reassembly coprocessor ignores the VLD bit in the free buffer queue when a new buffer is required. The periodic writing of the read index pointer to host/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 × 1024 is the maximum allowable size in bytes of an AAL5 CPCS-PDU, including overhead.  |
| 4-0  | 5          | GDIS_PRI | Global Discard Priority. Used by the Frame Relay and CLP discard functions. If the individual channel priority number is less than or equal to GDIS_PRI, PDUs on that channel can be discarded. |



## 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" data-bbox="829 667 1265 1136" 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> <tr> <td>110</td> <td>64</td> </tr> <tr> <td>111</td> <td>not valid.</td> </tr> </tbody> </table> | VALUE | VCC BLOCK_SIZE | 000 | 1 | 001 | 2 | 010 | 4 | 011 | 8 | 100 | 16 | 101 | 32 | 110 | 64 | 111 | not valid. |
| VALUE | VCC BLOCK_SIZE |                |  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 000   | 1              |                |  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 001   | 2              |                |  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 010   | 4              |                |  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 011   | 8              |                |  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 100   | 16             |                |  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 101   | 32             |                |  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 110   | 64             |                |  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 111   | not valid.     |                |  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 27    | 1              | EN_VPI_SIZE    | If logic high, enables an expanded, more flexible VPI Index table configuration. The VPI_SIZE register and PORT_ID from the framer are used to calculate an index into the VPI Index table.  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 26–24 | 3              | Reserved       | Program and read as 0.   |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 23–20 | 4              | Reserved       | Program and read as 0.   |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 19–17 | 3              | Reserved       | Program and read as 0.   |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 16–13 | 4              | Reserved       | Program and read as 0.   |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 12    | 1              | OAM_FF_DSC     | OAM FIFO buffer Full Discard. When a logic high and OAM_QU_EN is a logic high, an OAM cell is discarded if the incoming DMA FIFO buffer 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 uses the global OAM free buffer queue and status queue instead of per-channel resources.  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 9–5   | 5              | OAM_BFR_QU     | OAM Free Buffer Queue. When OAM_QU_EN is a logic high, OAM cells uses, buffers from the free buffer queue identified by OAM_BFR_QU.  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |
| 4–0   | 5              | OAM_STAT_QU    | OAM Status Queue. When OAM_QU_EN is a logic high, OAM cells posts status to the status queue identified by OAM_STAT_QU.  |       |                |     |   |     |   |     |   |     |   |     |    |     |    |     |    |     |            |

### 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. |

### 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.<br>0 = 64<br>1 = 256<br>2 = 1,024<br>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 – 15 are four words independent of the value of FWD_EN; otherwise, they are two words. |
| 11–8  | 4          | FWD_EN     | Forward Processing Enable. Selects the number of free buffer queues in bank 0 that have buffer return processing enabled. Starting with free buffer queue 0, a value of 0 in FWD_EN selects only one queue, and a value of 15 selects 16 queues.             |
| 7–0   | 8          | FBQ_UD_INT | Free Buffer Queue Update Interval. This value determines how many buffers are taken off the free buffer queue before the reassembly coprocessor writes the current read index pointer to host or SAR-shared memory.  |

### 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.      |

### 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. A value of 0 divides by 65,538. |
| 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.      |

### 0x108—Reassembly/Segmentation Queue Base Address 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       | Program and read as 0.  |
| 17–16 | 2          | RS_SIZE[1:0]   | Size of the RS Queue. Each RS Queue entry is eight octets in length.<br>00 = 256<br>01 = 1,024<br>10 = 4,096<br>11 = 16,384 |
| 15–0  | 16         | RS_QBASE[15:0] | Base address of the reassembly/segmentation queue.  |

### 0x10C—Reassembly ER\_SHIFT Tables Base Register (ERS\_BASE)

This register sets the base address in SRC local memory for the ER\_SHIFT tables used in implicit host congestion ER reduction. The base addresses are 128-byte aligned, and only the 16 most significant bits of the address are specified in the ERS\_BASE register.

| Bit   | Field Size | Name             | Description                           |
|-------|------------|------------------|---------------------------------------|
| 31–16 | 16         | REserved         | Program and read as 0.                |
| 15–0  | 16         | ER_SHIFT_B[15:0] | Base address for the ER_SHIFT tables. |

### 0x110—Variable VPI Size Register (VPI\_SIZE)

This register determines the maximum VPI allowed per port.

| Bit   | Field Size    | Name     | Description   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
|-------|---------------|----------|---|-------|---------------|---|-----|---|---|---|---|---|---|---|----|---|----|---|----|---|-----|---|-----|---|-----|---|-------|---|-------|---|-------|---|----------|---|----------|---|----------|
| 31–28 | 4             | MAX_VPI7 | Maximum VPI for port.<br><table border="0"> <tr> <td>Value</td> <td>max VPI value</td> </tr> <tr> <td>0</td> <td>OFF</td> </tr> <tr> <td>1</td> <td>1</td> </tr> <tr> <td>2</td> <td>3</td> </tr> <tr> <td>3</td> <td>7</td> </tr> <tr> <td>4</td> <td>15</td> </tr> <tr> <td>5</td> <td>31</td> </tr> <tr> <td>6</td> <td>63</td> </tr> <tr> <td>7</td> <td>127</td> </tr> <tr> <td>8</td> <td>255</td> </tr> <tr> <td>9</td> <td>511</td> </tr> <tr> <td>a</td> <td>1,023</td> </tr> <tr> <td>b</td> <td>2,047</td> </tr> <tr> <td>c</td> <td>4,095</td> </tr> <tr> <td>d</td> <td>not used</td> </tr> <tr> <td>e</td> <td>not used</td> </tr> <tr> <td>f</td> <td>not used</td> </tr> </table> | Value | max VPI value | 0 | OFF | 1 | 1 | 2 | 3 | 3 | 7 | 4 | 15 | 5 | 31 | 6 | 63 | 7 | 127 | 8 | 255 | 9 | 511 | a | 1,023 | b | 2,047 | c | 4,095 | d | not used | e | not used | f | not used |
| Value | max VPI value |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 0     | OFF           |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 1     | 1             |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 2     | 3             |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 3     | 7             |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 4     | 15            |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 5     | 31            |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 6     | 63            |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 7     | 127           |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 8     | 255           |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 9     | 511           |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| a     | 1,023         |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| b     | 2,047         |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| c     | 4,095         |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| d     | not used      |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| e     | not used      |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| f     | not used      |          |   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 27–24 | 4             | MAX_VPI6 | Maximum VPI for port 6.   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 23–20 | 4             | MAX_VPI5 | Maximum VPI for port 5.   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 19–16 | 4             | MAX_VPI4 | Maximum VPI for port 4.   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 15–12 | 4             | MAX_VPI3 | Maximum VPI for port 3.   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 11–8  | 4             | MAX_VPI2 | Maximum VPI for port 2.   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 7–4   | 4             | MAX_VPI1 | Maximum VPI for port 1.   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |
| 3–0   | 4             | MAX_VPI0 | Maximum VPI for port 0.   |       |               |   |     |   |   |   |   |   |   |   |    |   |    |   |    |   |     |   |     |   |     |   |       |   |       |   |       |   |          |   |          |   |          |

### 0x114—Transmit Port Register (TX\_FIFO\_CTRL)

This register sets a counter compare value that is used for the head of line flushing mechanism of the transmit FIFO. If head of line flushing is enabled in Configuration register 1, a counter is reset when the UTOPIA master in multi-PHY mode puts out the address of a PHY device, and the counter is increased based on UTOPIA tx\_clk. Once the counter reaches the TX\_CNTR value and no UTOPIA CLAV signal have been received from the PHY device, the cell in the FIFO is discarded. TX\_CNTR must not have a value of 0 if the head of line flushing mechanism is enabled.

| Bit   | Field Size | Name     | Description                                |
|-------|------------|----------|--|
| 31-16 | 16         | Reserved | Set to 0.                                  |
| 15-0  | 16         | TX_CNTR  | Counter Values to Tx FIFO flush mechanism. |

### 0x118—Transmit Port Control Register

This register disables UTOPIA ports if the TX\_FIFO\_FLUSH\_EN bit in Configuration register 1 is set. If bit x of the TX\_PORT\_DIS bitmap is set, cells belonging to PHY x are being discarded. No TX\_STATUS bit are set.

| Bit  | Field Size | Name             | Description          |
|------|------------|------------------|----------------------|
| 31-8 | 24         | Reserved         | Set to 0.            |
| 7-0  | 8          | TX_PORT_DIS[7:0] | Disables PHY port x. |

## 14.6 Counters and Status Registers

### 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 0 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. |

### 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 0 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. |

### 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 0 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. |

### 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 0 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. |

### 0x1a0—Host Mailbox Register (HOST\_MBOX)

This register implements a mailbox for communication between the host and local processors. The register is written by the local processor and read by the host to pass messages in that direction. Writes to this register can interrupt the host, while reads can interrupt the local processor.

| Bit  | Field Size | Name            | Description                                 |
|------|------------|-----------------|---|
| 31–0 | 32         | HOST_MBOX[31:0] | Messages flow from local processor to host. |

### 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. |

### 0x1a8—AALx\_STAT Register (AALx\_STAT)

This register provides the status of the AALx shadow counters. All bits are latched until read. Any bit set to a logic high causes the AALx\_STAT bit to be set in the HOST\_ISTAT1 and LP\_ISTAT1 registers.

| Bit   | Field Size | Name               | Description   |
|-------|------------|--------------------|---|
| 31–22 | 10         | Reserved           | Always read as 0.   |
| 21–16 | 6          | AALx_RSM_OVFL[5:0] | A cell was discarded since the ingress FIFO buffer of port x was full.              |
| 15–14 | 2          | Reserved           | Always read as 0.   |
| 13–8  | 6          | AALx_RSM_UNFL[5:0] | A HFIFORDx edge was detected when the ingress FIFO buffer shadow counter was empty. |
| 7–6   | 2          | Reserved           | Always read as 0.   |
| 5–0   | 6          | AALx_SEG_OVFL[5:0] | A HFIFOWRx edge was detected when the egress FIFO buffer shadow counter was full.   |

### 0x1ac—Transmit Port Cell Discard Status Register (TX\_STATUS)

This register indicates the status of a PHY device. If the head of line flushing mechanism is enabled (TX\_FIFO\_FLUSH\_EN set in Configuration register 1), this register indicates if a cell has been discarded due to the head of line flushing mechanism. If a bit x of the TX\_STAT bitmap is set, PHY x discarded a cell. If 1 or more bits of TX\_STAT is set, an interrupt is generated (if enabled). The TX\_STAT bitmap is latched until the TX\_STATUS register is read by the host.

| Bit  | Field Size | Name         | Description          |
|------|------------|--------------|----------------------|
| 31–8 | 24         | Reserved     | Set to 0.            |
| 7–0  | 8          | TX_STAT[7:0] | Status of Tx port x. |

### 0x1b0—Local Processor Mailbox Register (LP\_MBOX)

This register implements a mailbox for communication between the host and local processors. LP\_MBOX is written by the host processor and read by the local processor to pass messages in that direction. Writes to this register can interrupt the local processor while reads can interrupt the host processor.

| Bit  | Field Size | Name          | Description   |
|------|------------|---------------|---|
| 31–0 | 32         | LP_MBOX[31:0] | Local processor mailbox register. Messages flow from host processor to local processor. |



### 14.6.1 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 follows:

- L Level-sensitive status—A logic 1 on the status bit causes an interrupt when enabled by the corresponding IMASK bit. Reading the status does not clear the status or interrupt. The source of the condition causing the status must be cleared before the status or interrupt is cleared.
- E Event driven status—A 0 > 1 transition on the status bit causes an interrupt when enabled. Reading the status register clears the status bit and the interrupt.
- DE Dual Event status—A 0 > 1 and 1 > 0 transition on the status bit can be enabled to cause an interrupt. Reading the status register clears the status bit and the interrupt.

*NOTE:* Only host reads reset the status bits in the HOST\_ISTAT0 register.

Table 14-3. 0x1c0—Host Processor 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 0.  |
| 28    | 1          | E    | HOST_MBOX_WRITTEN | This bit is set upon a write to the HOST_MBOX register by the local processor, and cleared by a read of the HOST_MBOX register.   |
| 27    | 1          | E    | LP_MBOX_READ      | This bit is set upon the read of the LP_MBOX register by the local processor.   |
| 26    | 1          | —    | Reserved          | Read as 0.  |
| 25–24 | 2          | —    | Reserved          | Read as 0.  |
| 23    | 1          | —    | Reserved          | Read as 0. Reserved for future status page expansion.   |
| 22    | 1          | L    | HSTAT1            | This bit is set when any bit in HOST_ISTAT1 is set.   |
| 21–19 | 3          | —    | Reserved          | Read as 0.  |
| 18    | 1          | E    | GFC_LINK          | Set when three consecutive received cells have GFC SET_A, SET_B, or HALT bits set.  |
| 17    | 1          | L    | RSM_RUN           | Set when the reassembly machine is running. Is high when the RSM Coprocessor is processing a cell.  |
| 16    | 1          | L    | RSM_HS_WRITE      | Indicates reassembly host status has been written by CN8236 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 CN8236.   |
| 14–12 | 3          | —    | Reserved          | Read as 0.  |
| 11    | 1          | L    | SEG_RUN           | Set when the segmentation machine is running. Is 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 CN8236 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 CN8236.  |
| 8–4   | 5          | —    | Reserved          | Read as 0.  |
| 3     | 1          | E    | AAL5_DSC_RLOVR    | Set on the occurrence of an AAL5_DSC_CNT rollover.  |
| 2     | 1          | E    | CELL_DSC_RLOVR    | Set on the occurrence of a CELL_DSC_CNT rollover.   |
| 1     | 1          | E    | CELL_RCVD_RLOVR   | Set on the occurrence of a CELL_RCVD_CNT rollover.  |
| 0     | 1          | E    | CELL_XMT_RLOVR    | Set on the occurrence of a CELL_XMIT_CNT rollover.  |

Table 14-4. 0x1c4—Host Processor 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    | 1          | L    | AALx_STAT     | Whenever any bit in AALx_STAT is a logic 1, AALx_STAT is a logic 1.   |
| 29    | 1          | L    | TX_DISCARD    | When any bit in TX_STATUS is a logic 1, TX_DISCARD is a logic 1. TX_DISCARD is reset to 0 after the host reads the TX_STATUS register.  |
| 28–27 | 2          | —    | Reserved      | Read as 0.  |
| 26    | 1          | E    | DMA_AFULL     | Set when the incoming DMA burst FIFO buffer becomes almost full.  |
| 25    | 1          | E    | FR_PAR_ERR    | Set on the occurrence of a parity error on the reassembly ATM physical interface.   |
| 24    | 1          | E    | FR_SYNC_ERR   | Set on the occurrence of a synchronization error on the reassembly ATM physical interface.  |
| 23–16 | 8          | —    | Reserved      | Read as 0.  |
| 15    | 1          | E    | RS_QUEUE_FULL | Reassembly/segmentation queue full condition.   |
| 14    | 1          | E    | RSM_OVFL      | Reassembly overflow. Indicates that a cell was lost due to a FIFO buffer full condition.  |
| 13    | 1          | E    | RSM_HS_FULL   | Set on the occurrence of a host status queue full condition.  |
| 12    | 1          | E    | RSM_LS_FULL   | Set on the occurrence of a local status queue full condition.   |
| 11    | 1          | E    | RSM_HF_EMPT   | Set on the occurrence of a host free buffer queue empty condition.  |
| 10    | 1          | E    | RSM_LF_EMPT   | Set on the occurrence of a local free buffer queue empty condition.   |
| 9–3   | 7          | —    | Reserved      | Read as 0.  |
| 2     | 1          | E    | SEG_UNFL      | Segmentation underflow indicates that a scheduled cell could not be sent due to lack of PCI bandwidth.  |
| 1     | 1          | E    | SEG_HS_FULL   | Indicates that the segmentation host status queue is full.  |
| 0     | 1          | E    | SEG_LS_FULL   | Indicates that the segmentation local status queue is full.   |

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 clears all of the HOST\_IMASK0 interrupt enables.

**Table 14-5. 0x1d0—Host Interrupt Mask Register 0 (HOST\_IMASK0)**

| Bit   | Field Size | Name                 | Description   |
|-------|------------|----------------------|---|
| 31    | 1          | EN_PFAIL             | Enables interrupt when PFAIL status is a logic 1.   |
| 30    | 1          | EN_PHY_INTR          | Enables interrupt when PHY_INTR status is a logic 1.  |
| 29    | 1          | Reserved             | Set to 0.   |
| 28    | 1          | EN_HOST_MBOX_WRITTEN | Enables interrupt when HOST_MBOX WRITTEN status is a logic 1.   |
| 27    | 1          | EN_LP_MBOX_READ      | Enables interrupt when LP_MBOX_READ status is a logic 1.  |
| 26    | 1          | Reserved             | Set to 0.   |
| 25–24 | 2          | Reserved             | Set to 0.   |
| 23    | 1          | Reserved             | Set to 0. Reserved for future status page expansion.  |
| 22    | 1          | EN_HSTAT1            | Global interrupt enable for HOST_ISTAT1 status register. Individual interrupts in HOST_ISTAT1 are enabled in HOST_IMASK1. |
| 21–19 | 3          | Reserved             | Set to 0.   |
| 18    | 1          | EN_GFC_LINK          | Enables interrupt when GFC_LINK status is a logic 1.  |
| 17    | 1          | EN_RSM_RUN           | Enables interrupt when RSM_RUN status is a logic 1.   |
| 16    | 1          | EN_RSM_HS_WRITE      | Enables interrupt when RSM_HS_WRITE status is a logic high.   |
| 15    | 1          | EN_RSM_LS_WRITE      | Enables interrupt when RSM_LS_WRITE status is a logic high.   |
| 14–12 | 3          | Reserved             | Set to 0.   |
| 11    | 1          | EN_SEG_RUN           | Enables interrupt when SEG_RUN status is a logic high.  |
| 10    | 1          | EN_SEG_HS_WRITE      | Enables interrupt when SEG_HS_WRITE status is a logic high.   |
| 9     | 1          | EN_SEG_LS_WRITE      | Enables interrupt when SEG_LS_WRITE status is a logic high.   |
| 8–4   | 5          | Reserved             | Set to 0.   |
| 3     | 1          | EN_AAL5_DSC_RLOVR    | Enables an interrupt when AAL5_DSC_RLOVR status is a logic high.  |
| 2     | 1          | EN_CELL_DSC_RLOVR    | Enables an interrupt when CELL_DSC_RLOVR status is a logic high.  |
| 1     | 1          | EN_CELL_RCVD_RLOVR   | Enables an interrupt when CELL_RCVD_RLOVR status is a logic high.   |
| 0     | 1          | EN_CELL_XMIT_RLOVR   | Enables an interrupt when CELL_XMIT_RLOVR status is a logic high.   |

This register contains the interrupt enables that correspond to the status in the HOST\_ISTAT1 register. The assertion of the HRST\* system reset pin clears all of the HOST\_IMASK1 interrupt enables.

**Table 14-6. 0x1d4—Host Interrupt Mask Register 1 (HOST\_IMASK1)**

| Bit   | Field Size | Name             | Description   |
|-------|------------|------------------|---|
| 31    | 1          | EN_PCI_BUS_ERROR | Enables interrupt when PCI_BUS_ERROR status is a logic 1.     |
| 30    | 1          | EN_AALx_STAT     | Enables interrupt when HOST_ISTAT1 (AALx_STAT) is a logic 1.  |
| 29    | 1          | EN_TX_DISCARD    | Enables interrupt when HOST_ISTAT1 (TX_DISCARD) is a logic 1. |
| 28–27 | 2          | Reserved         | Set to 0.   |
| 26    | 1          | EN_DMA_AFULL     | Enabled interrupt when DMA_AFULL status is a logic 1.         |
| 25    | 1          | EN_FR_PAR_ERR    | Enables interrupt when FR_PAR_ERR status is a logic 1.        |
| 24    | 1          | EN_FR_SYNC_ERR   | Enables interrupt when FR_SYNC_ERR status is a logic 1.       |
| 23–16 | 8          | Reserved         | Set to 0.   |
| 15    | 1          | EN_RSQUEUE_FULL  | Enables interrupt when RSQUEUE_FULL status is a logic 1.      |
| 14    | 1          | EN_RSM_OVFL      | Enables interrupt when RSM_OVFL status is a logic 1.          |
| 13    | 1          | EN_RSM_HS_FULL   | Enables interrupt when RSM_HS_FULL status is a logic high.    |
| 12    | 1          | EN_RSM_LS_FULL   | Enables interrupt when RSM_LS_FULL status is a logic high.    |
| 11    | 1          | EN_RSM_HF_EMPT   | Enables interrupt when RSM_HF_EMPT status is a logic high.    |
| 10    | 1          | EN_RSM_LF_EMPT   | Enables interrupt when RSM_LF_EMPT status is a logic high.    |
| 9–3   | 7          | Reserved         | Set to 0.   |
| 2     | 1          | EN_SEG_UNFL      | Enables interrupt when SEG_UNFL status is a logic high.       |
| 1     | 1          | EN_SEG_HS_FULL   | Enables interrupt when SEG_HS_FULL status is a logic high.    |
| 0     | 1          | EN_SEG_LS_FULL   | Enables interrupt when SEG_LS_FULL status is a logic high.    |

## 14.6.2 Local Processor Interrupt Status Registers

The Local Processor Interrupt Status registers contain all the interruptible status bits for the local processor. The corresponding interrupt enables are located in the LP\_IMASK<sub>x</sub> registers. Status types are defined as follows:

- L Level sensitive status—A logic 1 on the status bit causes an interrupt when enabled by the corresponding IMASK bit. Reading the status does not clear the status or interrupt. The source of the condition causing the status must be cleared before the status or interrupt is cleared.
- E Event driven status—A 0 > 1 transition on the status bit causes an interrupt when enabled. Reading the status register clears the status bit and the interrupt.
- DE Dual event status—A 0 > 1 and 1 > 0 transition on the status bit can be enabled to cause an interrupt. Reading the status register clears the status bit and the interrupt.

*NOTE:* Only local processor reads reset the status bits in the LP\_ISTAT0 register.

Table 14-7. 0x1e0—Local Processor Interrupt Status Register 0 (LP\_ISTAT0)

| Bit   | Field Size | Type | Name            | Description  |
|-------|------------|------|-----------------|--|
| 31    | 1          | E    | RTC_OVFL        | Clock register overflow.   |
| 30    | 1          | E    | ALARM1          | Set when ALARM1 register matches CLOCK register.   |
| 29    | 1          | —    | Reserved        | Read as 0.   |
| 28    | 1          | E    | LP_MBOX_WRITTEN | This bit is set upon a write to the LP_MBOX register by the host processor.  |
| 27    | 1          | E    | HOST_MBOX_READ  | This bit is set upon the read of the HOST_MBOX register by the host processor.   |
| 26    | 1          | —    | Reserved        | Read as 0.   |
| 25–24 | 2          | —    | Reserved        | Read as 0.   |
| 23    | 1          | —    | Reserved        | Read as 0. Reserved for future status page expansion.  |
| 22    | 1          | L    | LSTAT1          | This bit is set when any bit in LP_ISTAT1 is set.  |
| 21–19 | 3          | —    | Reserved        | Read as 0.   |
| 18    | 1          | E    | GFC_LINK        | Set when three consecutive received cells have GCF SET_A, SET_B, or HALT bits set.   |
| 17    | 1          | L    | RSM_RUN         | Set when the reassembly machine is running. Is high when the RSM coprocessor is processing a cell.   |
| 16    | 1          | L    | RSM_HS_WRITE    | Indicates reassembly host status has been written by the CN8236 to status queues 0 through 15. For queue number, read HOST_ST_WR which must be read in order to clear status bit.    |
| 15    | 1          | E    | RSM_LS_WRITE    | Indicates that a reassembly local status queue has been written by the CN8236.   |
| 14–12 | 3          | —    | Reserved        | Read as 0.   |
| 11    | 1          | L    | SEG_RUN         | Set when the segmentation machine is running. Is high when SEG_ENABLE bit in SEG_CTRL is high or when processing the last cell after SEG_ENABLE is set low.                          |
| 10    | 1          | L    | SEG_HS_WRITE    | Indicates segmentation host status has been written by the CN8236 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 CN8236.   |
| 8–4   | 5          | —    | Reserved        | Read as 0.   |
| 3     | 1          | E    | AAL5_DSC_RLOVR  | Set on the occurrence of an AAL5_DSC_CNT rollover.   |
| 2     | 1          | E    | CELL_DSC_RLOVR  | Set on the occurrence of a CELL_DSC_CNT rollover.  |
| 1     | 1          | E    | CELL_RCVD_RLOVR | Set on the occurrence of a CELL_RCVD_CNT rollover.   |
| 0     | 1          | E    | CELL_XMIT_RLOVR | Set on the occurrence of a CELL_XMIT_CNT rollover.   |

Table 14-8. 0x1e4—Local Processor Interrupt Status Register 1 (LP\_ISTAT1)

| Bit   | Field Size | Type | Name          | Description   |
|-------|------------|------|---------------|---|
| 31    | 1          | L    | PCI_BUS_EROR  | This bit is set if the MERROR bit in the PCI configuration register is set. The MERROR bit is reset by either writing a logic 1 to the MERROR bit in the PCI configuration register, or setting the CONFIG0(PCI_ERR_RESET) bit to a logic high. |
| 30    | 1          | L    | AALx_STAT     | Whenever any bit in AALx_STAT is a logic 1, AALx_STAT is a logic 1.   |
| 29    | 1          | L    | TX_DISCARD    | When any bit in TX_STATUS is a logic 1, TX_DISCARD is reset to 0 after the host reads the TX_STATUS register.   |
| 28–27 | 2          | —    | Reserved      | Read as 0.  |
| 26    | 1          | E    | DMA_AFULL     | Set when the incoming DMA burst FIFO buffer becomes almost full.  |
| 25    | 1          | E    | FR_PAR_ERR    | Set on the occurrence of a parity error on the reassembly ATM physical interface.   |
| 24    | 1          | E    | FR_SYNC_ERR   | Set on the occurrence of a synchronization error on the reassembly ATM physical interface.  |
| 23–16 | 8          | —    | Reserved      | Read as 0.  |
| 15    | 1          | E    | RS_QUEUE_FULL | Reassembly/segmentation queue full condition.   |
| 14    | 1          | E    | RSM_OVFL      | Reassembly overflow. Indicates that a cell was lost due to a FIFO buffer full condition.  |
| 13    | 1          | E    | RSM_HS_FULL   | Set on the occurrence of a host status queue full condition.  |
| 12    | 1          | E    | RSM_LS_FULL   | Set on the occurrence of a local status queue full condition.   |
| 11    | 1          | E    | RSM_HF_EMPT   | Set on the occurrence of a host free buffer queue empty condition.  |
| 10    | 1          | E    | RSM_LF_EMPT   | Set on the occurrence of a local free buffer queue empty condition.   |
| 9–3   | 7          | —    | Reserved      | Read as 0.  |
| 2     | 1          | E    | SEG_UNFL      | Segmentation underflow indicates that a scheduled cell could not be sent due to lack of PCI bandwidth.  |
| 1     | 1          | E    | SEG_HS_FULL   | Indicates that the segmentation host status queue is full.  |
| 0     | 1          | E    | SEG_LS_FULL   | Indicates that the segmentation local status queue is full.   |



This register contains the interrupt enables that correspond to the status in the LP\_ISTAT0 register. The assertion of the HRST\* system reset pin clears all of the LP\_IMASK0 interrupt enables.

**Table 14-9. 0x1f0—Local Processor Interrupt Mask Register 0 (LP\_IMASK0)**

| Bit   | Field Size | Name               | Description   |
|-------|------------|--------------------|---|
| 31    | 1          | EN_RTC_OVFL        | Enables an interrupt when RTC_OVFL status is a logic high.  |
| 30    | 1          | EN_ALARM1          | Enables an interrupt when ALARM1 status is a logic 1.   |
| 29    | 1          | Reserved           | Set to 0.   |
| 28    | 1          | EN_LP_MBOX_WRITTEN | Enables an interrupt when LP_MBOX_WRITTEN status is a logic 1.  |
| 27    | 1          | EN_HOST_MBOX_READ  | Enables an interrupt when HOST_MBOX_READ status is a logic 1.   |
| 26    | 1          | Reserved           | Set to 0.   |
| 25–24 | 2          | Reserved           | Set to 0.   |
| 23    | 1          | Reserved           | Set to 0. Reserved for future status page expansion.  |
| 22    | 1          | EN_LSTAT1          | Global interrupt enable for LP_ISTAT1 status register. Individual interrupts of LP_ISTAT1 are enabled in LP_IMASK1. |
| 21–19 | 3          | Reserved           | Set to 0.   |
| 18    | 1          | EN_GFC_LINK        | Enables an interrupt when GFC_LINK status is a logic 1.   |
| 17    | 1          | EN_RSM_RUN         | Enables an interrupt when RSM_RUN status is a logic 1.  |
| 16    | 1          | EN_RSM_HS_WRITE    | Enables an interrupt when RSM_HS_WRITE status is a logic high.  |
| 15    | 1          | EN_RSM_LS_WRITE    | Enables an interrupt when RSM_LS_WRITE status is a logic high.  |
| 14–12 | 3          | Reserved           | Set to 0.   |
| 11    | 1          | EN_SEG_RUN         | Enables an interrupt when SEG_RUN status is a logic high.   |
| 10    | 1          | EN_SEG_HS_WRITE    | Enables an interrupt when SEG_HS_WRITE status is a logic high.  |
| 9     | 1          | EN_SEG_LS_WRITE    | Enables an interrupt when SEG_LS_WRITE status is a logic high.  |
| 8–4   | 5          | Reserved           | Set to 0.   |
| 3     | 1          | EN_AAL5_DSC_RLOVR  | Enables an interrupt when 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.   |

This register contains the interrupt enables that correspond to the statuses in the LP\_ISTAT1 register. The assertion of the HRST\* system reset pin clears all of the LP\_IMASK1 interrupt enables.

**Table 14-10. 0x1f4—Local Processor Interrupt Mask Register 1 (LP\_IMASK1)**

| Bit   | Field Size | Name             | Description   |
|-------|------------|------------------|---|
| 31    | 1          | EN_PCI_BUS_ERROR | Enables an interrupt when PCI_BUS_ERROR status is a logic 1.  |
| 30    | 1          | EN_AALx_STAT     | Enables interrupt when HOST_ISTAT1(AALx_STAT) is a logic 1.   |
| 29    | 1          | EN_TX_DISCARD    | Enables an interrupt when LP_STAT1(TX_DISCARD) is a logic 1.  |
| 28–27 | 2          | Reserved         | Set to 0.   |
| 26    | 1          | EN_DMA_AFULL     | Enables an interrupt when DMA_AFULL status is a logic 1.      |
| 25    | 1          | EN_FR_PAR_ERR    | Enables an interrupt when FR_PAR_ERR status is a logic 1.     |
| 24    | 1          | EN_FR_SYNC_ERR   | Enables an interrupt when FR_SYNC_ERR status is a logic 1.    |
| 23–16 | 8          | Reserved         | Set to 0.   |
| 15    | 1          | EN_RSQUEUE_FULL  | Enables an interrupt when RSQUQUQ_FULL status is a logic 1.   |
| 14    | 1          | EN_RSM_OVFL      | Enables an interrupt when RSM_OVFL status is a logic 1.       |
| 13    | 1          | EN_RSM_HS_FULL   | Enables an interrupt when RSM_HS_FULL status is a logic high. |
| 12    | 1          | EN_RSM_LS_FULL   | Enables an interrupt when RSM_LS_FULL status is a logic high. |
| 11    | 1          | EN_RSM_HS_EMPT   | Enables an interrupt when RSM_HS_EMPT status is a logic high. |
| 10    | 1          | EN_RSM_LS_EMPT   | Enables an interrupt when RSM_LS_EMPT status is a logic high. |
| 9–3   | 7          | Reserved         | Set to 0.   |
| 2     | 1          | EN_SEG_UNFL      | Enables an interrupt when SEG_UNFL status is a logic high.    |
| 1     | 1          | EN_SEG_HS_FULL   | Enables an interrupt when SEG_HS_FULL status is a logic high. |
| 0     | 1          | EN_SEG_LS_FULL   | Enables an interrupt when SEG_LS_FULL status is a logic high. |

## 14.7 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 CN8236 is shown in [Table 14-11](#). [Table 14-12](#) provides descriptions of fields and other registers within the PCI configuration space.

**Table 14-11. 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 | 0 |
| 0x00      | DEVICE_ID (0x8236)                |    |    |    |    |    |    |    |                    |    |    |    |    |    |    |    | 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                      |    |    |    |    |    |   |   |                        |   |   |   |   |   |   |   |
| 0x58–0x7c | Reserved (0x00000000)             |    |    |    |    |    |    |    |                    |    |    |    |    |    |    |    |                            |    |    |    |    |    |   |   |                        |   |   |   |   |   |   |   |

[Table 14-13](#) details the PCI Command register and [Table 14-14](#) shows the PCI Status register breakdown. The PCI Special Status register is detailed in [Table 14-15](#). [Table 14-16](#) details the EPROM register.

Table 14-12. PCI Register Configuration Register Field Descriptions (1 of 2)

| Bit                        | Description  |
|----------------------------|--|
| DEVICE_ID                  | 16-bit device identifier. Serves to uniquely identify the SAR to the host operating system. Set to 0x8236.   |
| 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 Status register 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 Command register 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 CN8236 chip: 0 = Rev A and 2 = Rev B.   |
| HEADER_TYPE                | This field identifies the layout of the second part of the predefined header of the PCI Configuration space (beginning at 0x10). See <i>PCI Local Bus Specification</i> , 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 cacheline 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 CN8236 (as seen and assigned by the host processor). Value after HRST* active is 0x00.   |
| 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 may 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 can 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.  |
| MAX_LATENCY                | This read-only register specifies how often the CN8236 device 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 $\mu$ s).  |
| MIN_GRANT                  | This read-only register specifies how long of a burst period the CN8236 device needs to gain access to the PCI bus. Set to 0x02 (a period of time in units of 1/4 $\mu$ s).  |
| INTERRUPT_PIN              | This read-only register tells which interrupt pin the device (or device function) uses. Set to 0x01 (corresponds to interrupt pin INTA#).  |
| 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.   |

**Table 14-12. PCI Register Configuration Register Field Descriptions (2 of 2)**

| Bit                     | Description   |
|-------------------------|---|
| SPECIAL_STATUS_REGISTER | Device status not defined by the PCI specification. (The field is further subdivided into subfields as described in Table 14-15.) Detailed descriptions of these subfields can be found in the PCI bus specification. 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 Table 14-16 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 <i>PCI Bus Power Management Interface Specification</i> 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 <i>PCI Bus Power Management Interface Specification</i> 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, and to enable and monitor power management events. See the <i>PCI Bus Power Management Interface Specification</i> Revision 1.0 for specific information related to this register.  |

Table 14-13. PCI Command Register

| Bit   | Field Size | Name                 | Description  |
|-------|------------|----------------------|--|
| 15–10 | 6          | Reserved             | Set to 0x0b000000.   |
| 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 0.  |
| 6     | 1          | PE_EN <sup>(1)</sup> | Parity Error detection and report enable.  |
| 5–3   | 3          | Reserved             | Set to 0.  |
| 2     | 1          | M_EN <sup>(1)</sup>  | Master enable. M_EN must be asserted (that is, set to 1) before the CN8236 can act as master on the PCI bus.   |
| 1     | 1          | MS_EN <sup>(1)</sup> | Memory Space enable. MS_EN must be asserted (that is, set to 1) before the CN8236 address space (registers and memory) can be accessed across the PCI interface. |
| 0     | 1          | —                    | I/O Space enable. (Not used.) Set to 0.  |

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

Table 14-14. PCI Status Register

| 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).  |
| 23    | 1          | —        | Fast Back-to-back supported as Slave. Set to 1.   |
| 22    | 1          | —        | UDF Support. Set to 0.  |
| 21    | 1          | —        | 66 MHz Support. Set to 0.   |
| 20    | 1          | NEW_CAP  | Capability List Support. This bit indicates whether this function implements a list of extended capabilities defined in <i>PCI Bus Specification</i> , Revision 2.1, such as PCI Power Management. When set to 1, 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 0x0.   |

**NOTE(S):**  
<sup>(1)</sup> Can be set to 0 from EEPROM.

Table 14-15. PCI Special Status Register

| 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.<br>0000000 = 8 M  |
| 15–12  | 4          | Reserved                        | Set to 0.  |
| 11   | 1          | INTF_DIS                        | Master Interface Disabled. If the M_EN bit in the COMMAND field is a logic low, any attempt by the CN8236 to perform a DMA transaction to the PCI bus results in an error. INTF_DIS and MERROR bits 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 is also 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><br><sup>(1)</sup> Can be loaded from EEPROM. |            |                                 |  |

Table 14-16. EEPROM Register (1 of 2)

| Bit | Field Size | Name | Description   |
|-----|------------|------|---|
| 31  | 1          | BUSY | Indicates that an EEPROM operation is currently in progress. This bit must be read as a 0 before initiating an EEPROM transfer. |

Table 14-16. EEPROM Register (2 of 2)

| Bit   | Field Size | Name          | Description   |
|-------|------------|---------------|---|
| 30    | 1          | NO_ACK        | When set to a 1, indicates that the previous transfer failed (that is, 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.   |



## 15.0 SAR Initialization—Example Tables

The following tables provide an example CN8236 SAR initialization of control registers, internal memory control structures, and external memory control structures.

### 15.1 Segmentation Initialization

#### 15.1.1 Segmentation Control Registers

Before segmentation is enabled, the host must allocate and initialize all of the segmentation control registers. [Table 15-1](#) lists the initialized values for each field.

*Table 15-1. Table of Values for Segmentation Control Register Initialization (1 of 2)*

| Register                                    | Field         | Initialized Value                      | Notes   |
|---|---------------|--|---|
| SEG_CTRL<br>(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 buffer depth to four cells.   |
|   | CLP0_EOM      | 0                                      | Disable CLP on EOM processing.  |
|   | OAM_STAT_ID   | 0x10                                   | OAM global status queue set to 16.  |
|   | SEG_ST_HALT   | 0                                      | Disable status queue entry for a VCC halted with a partially segmented buffer.  |
|   | SEG_LS_DIS    | 0                                      | Enable local status queue full check.   |
|   | SEG_HS_DIS    | 0                                      | Enable host status queue full check.  |
|   | TX_RND        | 1                                      | Round-robin transmit queue processing selected.   |
| TR_SIZE                                     | 0x0           | Transmit queue size set to 64 entries. |   |

Table 15-1. Table of Values for Segmentation Control Register Initialization (2 of 2)

| Register  | Field         | Initialized Value | Notes  |
|---|---------------|-------------------|--|
| SEG_VBASE<br>(SEG Virtual Channel<br>Connection Base<br>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_PMBASE<br>(SEG PM Base Address<br>Register)                           | SEG_BCKB      | 0x219             | VBR bucket table starts at 0x10C80 in SAR-shared memory. |
|   | SEG_PMB       | 0x1AD             | PM table starts at 0xD680 in SAR-shared memory.          |
| SEG_TXBASE<br>(Segmentation<br>Transmit Queue Base<br>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.                 |

### 15.1.2 Segmentation Internal Memory Control Structures

Before segmentation is enabled, the host must allocate and initialize all of the segmentation internal memory control structures. Table 15-2 lists the initialized values for each field.

Table 15-2. Table of Values for Segmentation Internal Memory Initialization

| Table  | Field        | Initialized Value | Notes                                     |
|--|--------------|-------------------|---|
| SEG_SQ_BASE Table<br>Entry 0<br>(SEG Status Queue<br>Base Table Entry 0)   | BASE_PNTR    | 0x1C00            | Base address of status queue 0 is 0x1C00. |
|  | LOCAL        | 0                 | Status queue 0 resides in host memory.    |
|  | SIZE         | 0x0               | Size of status queue 0 is 64 entries.     |
|  | WRITE        | 0x0               | Must be initialized to 0.                 |
|  | READ_UD      | 0x0               | Must be initialized to 0.                 |
|  | Reserved     | 0x0               | Must be initialized to 0.                 |
| SEG_TQ_BASE Table<br>Entry 0<br>(SEG Transmit Queue<br>Base Table Entry 0) | READ_UD_PNTR | 0x40              | Location of READ_UD is at 0x100.          |
|  | LOCAL        | 0                 | READ_UD located in host memory.           |
|  | UPDATE       | 0x0               | Must be initialized to 0.                 |
|  | READ         | 0x0               | Must be initialized to 0.                 |
|  | Reserved     | 0x0               | Must be initialized to 0.                 |

### 15.1.3 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 15-3](#) lists the initialized values for each field.

**Table 15-3. Table of Values for Segmentation SAR Shared Memory Initialization (1 of 2)**

| Table                                 | Field       | Initialized Value         | Notes   |
|---------------------------------------|-------------|---------------------------|---|
| SEG VCC Table Entry 0 – (Words 0 – 6) | PM_INDEX    | 0x0                       | PM table index = 0.                                   |
|                                       | LAST_PNTR   | 0x0                       | Must be initialized to 0.                             |
|                                       | ATM_HEADER  | 0x00100100                | ATM Header, VPI = 0x1, VCI = 0x10.                    |
|                                       | CRC_REM     | 0xFFFF_FFFF               | Must be initialized to 0xFFFF_FFFF for AAL5 channels. |
|                                       | BETAG       | 0                         | First AAL3/4 BTag/ETag pair is 0.                     |
|                                       | SN          | 0                         | First AAL3/4 SN = 0.                                  |
|                                       | MID         | 5                         | AAL3/4 MID = 5.                                       |
|                                       | STM_MODE    | 0                         | Status Message Mode enabled.                          |
|                                       | STAT        | 0x2                       | Channel uses status queue 2.                          |
|                                       | PM_EN       | 1                         | PM OAM processing enabled.                            |
|                                       | CURR_PNTR   | 0x0                       | Must be initialized to 0.                             |
|                                       | 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.       |   |
| Reserved                              | 0x0         | Must be initialized to 0. |   |
| SEG Buffer Descriptors                |             |                           | (No initialization required.)                         |
| SEG Transmit Queue Entries            | VLD         | 0                         | Must be initialized to 0.                             |
|                                       | LINK_HEAD   | 0                         | Must be initialized to 0.                             |
|                                       | FND_CHAIN   | 0                         | Must be initialized to 0.                             |
|                                       | SEG_BD_PNTR | 0x0                       | Must be initialized to 0.                             |
|                                       | Reserved    | 0x0                       | Must be initialized to 0.                             |

Table 15-3. Table of Values for Segmentation SAR Shared Memory Initialization (2 of 2)

| Table                    | Field         | Initialized Value         | Notes  |
|--------------------------|---------------|---------------------------|--|
| SEG Status Queue Entries | USER_PNTR     | 0x0                       | Must be initialized to 0.                    |
|                          | VLD           | 0                         | Must be initialized to 0.                    |
|                          | STOP          | 0                         | Must be initialized to 0.                    |
|                          | DONE          | 0                         | Must be initialized to 0.                    |
|                          | SINGLE        | 0                         | Must be initialized to 0.                    |
|                          | OVFL          | 0                         | Must be initialized to 0.                    |
|                          | I_EXP         | 0x0                       | Must be initialized to 0.                    |
|                          | I_MAN         | 0x0                       | Must be initialized to 0.                    |
|                          | SEG_VCC_INDEX | 0x0                       | Must be initialized to 0.                    |
| Reserved                 | 0x0           | Must be initialized to 0. |  |
| SEG_PM Table Entry 0     | ATM_HEADER    | 0x00100108                | VPI = 0x1, VCI = 0x10, PTI = 4 (F5 segment). |
|                          | FWD_TUC0      | 0x0                       | Must be initialized to 0.                    |
|                          | FWD_TUC01     | 0x0                       | Must be initialized to 0.                    |
|                          | BCK_MSN       | 0x0                       | Must be initialized to 0.                    |
|                          | BCK_TUC0      | 0x0                       | Must be initialized to 0.                    |
|                          | BCK_TUC01     | 0x0                       | Must be initialized to 0.                    |
|                          | TRCC0         | 0x0                       | Must be initialized to 0.                    |
|                          | TRCC0+1       | 0x0                       | Must be initialized to 0.                    |
|                          | BIP           | 0x0                       | Must be initialized to 0.                    |
|                          | FWD_MON       | 1                         | Forward Monitoring cell generation enabled.  |
|                          | BLOCK_SIZE    | 10                        | PM OAM block size of 512.                    |
|                          | FWD_MSN       | 0x3                       | Initial sequence number is 3.                |
|                          | Reserved      | 0x0                       | Must be initialized to 0.                    |

## 15.2 Scheduler Initialization

### 15.2.1 Scheduler Control Registers

Before segmentation is enabled, the host must allocate and initialize all of the Scheduler control registers. [Table 15-4](#) lists the initialized values for each field.

**Table 15-4. Table of Values for Scheduler Control Register Initialization**

| Register  | Field       | Initialized Value | Notes  |
|---|-------------|-------------------|--|
| SCH_PRI<br>(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<br>(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<br>(Schedule Maximum ABR Register)          | VCC_MAX     | 0x63              | Enable 50 channels of ABR processing.                    |
| PCR_QUE_INT01<br>(PCR Queue Interval 0 and 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<br>(PCR Queue Interval 2 and 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<br>(Schedule ABR Constant Register)         | ABR_TRM     | 0x23C             | Set TRM to <i>TM 4.1</i> default of 100 ms.              |
|   | ABR_ADTF    | 0xB2E             | Set ADTF to <i>TM 4.1</i> default of 0.5 second.         |
| SCH_ABRBASE<br>(ABR Decision Table Lookup Base Reg)     | OOR_ENA     | 1                 | Enable out-of-rate ABR RM cells.                         |
|   | OOR_INT     | 0x1C9D            | Produces an out-of-rate interval of one cell per second. |
|   | SCH_ABRB    | 0x1D1             | ABR table starts at 0xE880 in SAR-shared memory.         |
| SCH_CNG<br>(ABR Congestion Register)                    | FBO_CNG     | 0x0               | No congestion experienced.                               |

## 15.2.2 Scheduler Internal Memory Control Structures

No internal memory scheduler structures need to be initialized.

## 15.2.3 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 15-5](#) lists the initialized values for each field.

**Table 15-5. Table of Values for Sch 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.            |
|  | Reserved      | 0xF               | Set all reserved bits to 1.                           |
| Schedule Table (Tunnel slot)   | CBR           | 0                 | Slot will schedule a tunnel.                          |
|  | PRI           | 0x3               | Tunnel Priority Queue = 3.                            |
|  | Reserved      | 0xF               | Set all reserved bits to 1.                           |
| Schedule Table (DEFAULT)   | Reserved      | 0xF               | Set all reserved bits to 1.                           |
| 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.   |
|  | Reserved      | 0x0               | Must be initialized to 0.                             |
| Bucket Table Entry   | L2_EXP        | 0xA               | —   |
|  | L2_MAN        | 0x0               | GCRA L = 1.   |
|  | I2_EXP        | 0xB               | —   |
|  | I2_MAN        | 0x0               | GCRA I = 2.   |
|  | Reserved      | 0x0               | Must be initialized to 0.                             |
| SEG VCC Table Entry for GFR (Words 7 – 9)<br>(NOTE: 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.)                 |
|  | Reserved      | 0x0               | Must be initialized to 0.                             |

Table 15-5. Table of Values for Sch SAR Shared Memory Initialization (2 of 2)

| Table                                      | Field              | Initialized Value | Notes   |
|--|--------------------|-------------------|---|
| SEG VCC Table Entry for ABR (Words 7 – 19) | CONG_ID            | 0x1               | Channel associated with RSM free buffer queue 1 for host congestion indication. |
|  | OOD_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 1.  |
|  | FWD_DIR            | 0                 | Forward RM cell DIR field set to 0.   |
|  | FWD_BN             | 0                 | Forward RM cell BN field set to 0.  |
|  | FWD_CI             | 0                 | Forward RM cell CI field set to 0.  |
|  | FWD_NI             | 0                 | Forward RM cell NI field set to 0.  |
|  | FWD_RA             | 0                 | Forward RM cell RA field set to 0.  |
|  | FWD_ER             | 0x64BF            | Forward RM cell ER field set to approximately 360000 cells per second.          |
|  | Reserved           | 0x0               | Must be initialized to 0.   |
|  | (ALL OTHER FIELDS) | —                 | Direct output of ABR template.  |
| ABR Cell Decision Block                    | (ALL FIELDS)       | —                 | Direct output of ABR template.  |
| ABR Rate Decision Block                    | (ALL FIELDS)       | —                 | Direct output of ABR template.  |
| Exponent Table                             | (ALL FIELDS)       | —                 | Direct output of ABR template.  |

## 15.3 Reassembly Initialization

### 15.3.1 Reassembly Control Registers

Before reassembly is enabled, the host must allocate and initialize all of the reassembly control registers. [Table 15-6](#) lists the initialization values for each field.

*Table 15-6. Table of Values for Reassembly Control Register Initialization (1 of 2)*

| Register  | Field        | Initialized Value         | Notes   |
|---|--------------|---------------------------|---|
| RSM_CTRL0<br>(Reassembly Control Register 0)        | RSM_ENABLE   | 0 – 1                     | Must be set to a logic low until initialization of all reassembly structures is complete. Set to a logic high to commence reassembly process. |
|   | RSM_RESET    | 0                         | Use CONFIG0(GLOBAL_RESET) to initialize the SAR.  |
|   | VPI_MASK     | 1                         | UNI VPI space selected.   |
|   | RSM_PHALT    | 0                         | RSM halt on receive parity error disabled.  |
|   | FWALL_EN     | 1                         | Firewall function enabled.  |
|   | RSM_FBQ_DIS  | 0                         | Free buffer queue underflow protection enabled.   |
|   | RSM_STAT_DIS | 0                         | Status queue overflow protection enabled.   |
|   | GTO_EN       | 1                         | Enable internal time-out interrupt.   |
|   | MAX_LEN      | 0x10                      | AAL5 max CPCS-PDU length = 16384 bytes.   |
|   | GDPRI        | 0x4                       | Global Service Discard Priority = 4.  |
| Reserved  | 0x0          | Must be initialized to 0. |   |
| RSM_CTRL1<br>(Reassembly Control Register 1)        | OAM_FF_DSC   | 1                         | OAM cells discarded when Incoming DMA FIFO buffer almost full.  |
|   | OAM_EN       | 1                         | OAM detection enabled.  |
|   | OAM_QU_EN    | 1                         | Global OAM FB and Stat queues enabled.  |
|   | OAM_BFR_QU   | 0x4                       | Global OAM free buffer queue = 4.   |
|   | OAM_STAT_QU  | 0x4                       | Global OAM status queue = 4.  |
|   | Reserved     | 0x0                       | Must be initialized to 0.   |
| RSM_FQBASE<br>(RSM Free Buffer Queue Base Register) | FQ01_BASE    | 0xB0                      | Free buffer queue bank 1 starts at 0x5800 in SAR-shared memory.   |
|   | FQ00_BASE    | 0x30                      | Free buffer queue bank 0 starts at 0x1800 in SAR-shared memory.   |



Table 15-6. Table of Values for Reassembly Control Register Initialization (2 of 2)

| Register   | Field      | Initialized Value | Notes  |
|--|------------|-------------------|--|
| RSM_FQCTRL<br>(RSM Free Buffer Queue Control Register) | FBQ_SIZE   | 0x0               | Free buffer queue size is 64 entries.                                      |
|  | FWD_RND    | 1                 | Free buffer queues are processed in round-robin fashion for credit return. |
|  | FBQ0_RTN   | 0                 | Free buffer queue bank 0 selected for buffer return processing.            |
|  | FWD_EN     | 0x4               | Free buffers queues 0 through 4 enabled for buffer return processing.      |
|  | FBQ_UD_INT | 0x20              | Free buffer queue update interval set to 32.                               |
|  | Reserved   | 0x0               | Must be initialized to 0.  |
| RSM_TBASE<br>(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<br>(RSM Time-Out Register)                      | RSM_TO_PER | 0x100             | Internal time-out interrupt every 256 SYSCLK periods.                      |
|  | RSM_TO_CNT | 0x10              | RSM VCC table entries 0 through 16 enabled for time-out processing.        |
| RS_QBASE<br>(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.                       |
|  | Reserved   | 0x0               | Must be initialized to 0.  |

### 15.3.2 Reassembly Internal Memory Control Structures

Before reassembly is enabled, the host must allocate and initialize all of the reassembly internal memory control structures. [Table 15-7](#) lists the initialized values for each field.

**Table 15-7. Table of Values for Reassembly Internal Memory Initialization**

| Table  | Field        | Initialized Value | Notes  |
|--|--------------|-------------------|--|
| RSM_SQ_BASE Table<br>Entry 0<br>(RSM Status Queue<br>Base Table Entry 0)       | BASE_PNTR    | 0x1800            | Base address of RSM status queue 0 is 0x6000.          |
|  | LOCAL        | 0                 | Status queue 0 resides in host memory.                 |
|  | SIZE         | 0x0               | Size of status queue 0 is 64 entries.                  |
|  | WRITE        | 0x0               | Must be initialized to 0.                              |
|  | READ_UD      | 0x0               | Must be initialized to 0.                              |
|  | Reserved     | 0x0               | Must be initialized to 0.                              |
| RSM_FBQ_BASE Table<br>Entry 0<br>(RSM Free Buffer Queue<br>Base Table Entry 0) | READ_UD_PNTR | 0x0               | Location of READ_UD is at 0x0.                         |
|  | BD_LOCAL     | 0                 | Buffer descriptors and READ_UD located in host memory. |
|  | BFR_LOCAL    | 0                 | Buffers located in host memory.                        |
|  | EMPT         | 0                 | Must be initialized to 0.                              |
|  | UPDATE       | 0x0               | Must be initialized to 0.                              |
|  | READ         | 0x0               | Must be initialized to 0.                              |
|  | FORWARD      | 0x20              | 32 free buffers initially put on free buffer queue.    |
|  | LENGTH       | 0x200             | Buffer lengths in free buffer queue 0 are 200 bytes.   |
|  | Reserved     | 0x0               | Must be initialized to 0.                              |
| Global Time-Out Table  | TERM_TOCNT0  | 0x400             | Provides a 133 ms time-out period.                     |
|  | TERM_TOCNT1  | 0xFFFF            | Provides an 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 0.                              |
|  | Reserved     | 0x0               | Must be initialized to 0.                              |

### 15.3.3 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 15-8](#) lists the initialized values for each field.

**Table 15-8. Table of Values for Reassembly SAR Shared Memory Initialization (1 of 4)**

| Table                   | Field          | Initialized Value                            | Notes   |
|-------------------------|----------------|--|---|
| VPI Index Table Entry 0 | VP_EN          | 1  | VPI = 0 enabled. NOTE: All entries in the VPI Index table must be initialized. If VPI is disabled, set VP_EN = 0.   |
|                         | VCI_RANGE      | 0x10   | Allowable VCI range is 0 to 0x43F.  |
|                         | VCI_IT_PNTR    | 0x2014                                       | VCI Index table is located at 0x8050 in SAR-shared memory.  |
| VCI Index Table Entry 0 | VCC_BLOCK_PNTR | 0x0  | Initial block of 64 VCC table entries is 0.   |
|                         | Reserved       | 0x0  | Must be initialized to 0.   |
| RSM VCC Table Entry 0   | FF_DSC         | 1  | Enable FIFO buffer Full EPD.  |
|                         | VC_EN          | 1  | Table Entry is enabled. All VCC table entries that have a path through the VPI/VCI lookup space must be initialized. If entry is disabled, set VC_EN = 0. |
|                         | AAL_TYPE       | 0x0  | Channel configured for AAL5.  |
|                         | DPRI           | 0x2  | Channel service discard priority set to 2.  |
|                         | TO_INDEX       | 0x1  | Point to TERM_TOCNT1 global time-out value.   |
|                         | PM_INDEX       | 0x2  | Channel used entry 2 of PM_OAM table.   |
|                         | AAL_EN         | 0x082  | Message Status mode and Frame Relay Discard enabled.  |
| RSM VCC Table Entry 0   | TO_LAST        | 0  | Not last VCC entry processed for time-out.  |
|                         | TO_EN          | 1  | Time-out processing enabled on this channel.  |
|                         | CUR_TOCNT      | 0x0  | Must be initialized to 0.   |
|                         | ABR_CTRL       | 0x0  | No ABR service enabled on this channel.   |
|                         | PDU_FLAGS      | 0x002  | Must be initialized to 2.   |
|                         | TOT_PDU_LEN    | 0x0  | Must be initialized to 0.   |
|                         | CRC_REM        | 0xFFFF_FFFF                                  | Must be initialized to 0xFFFF_FFFF for AAL5 channels.   |
|                         | BASIZE         | 0x0  | Must be initialized to 0. (For AAL3/4 only.)  |
|                         | NEXT_ST        | 0x2  | Must be initialized to 2. (For AAL3/4 only.)  |
| NEXT_SN                 | 0x0            | Must be initialized to 0. (For AAL3/4 only.) |   |

Table 15-8. Table of Values for Reassembly SAR Shared Memory Initialization (2 of 4)

| Table                           | Field         | Initialized Value         | Notes   |
|---------------------------------|---------------|---------------------------|---|
| RSM VCC Table Entry 0           | BTAG          | 0x0                       | Must be initialized to 0. (For AAL3/4 only.)              |
|                                 | STAT          | 0x1                       | Channel uses status queue 1.                              |
|                                 | BFR1          | 0x11                      | Channel uses free buffer queue 17 for COM buffers.        |
|                                 | BFR0          | 0x1                       | Channel uses free buffer queue 1 for BOM buffers.         |
|                                 | SEG_VCC_INDEX | 0x4                       | Corresponding SEG channel = 4 for PM_OAM and ABR service. |
|                                 | SERV_DIS      | 0x0                       | Must be initialized to 0.                                 |
|                                 | RX_COUNTER    | 0x100                     | Initial firewall credit = 256.                            |
|                                 | Reserved      | 0x0                       | Must be initialized to 0.                                 |
| RSM AAL3/4 Head VCC Table Entry | VC_EN         | 1                         | Table entry is enabled.                                   |
|                                 | AAL_TYPE      | 0x2                       | Channel configured for AAL3/4.                            |
|                                 | FF_DSC        | 1                         | Enable FIFO buffer Full EPD.                              |
|                                 | PM_INDEX      | 0x2                       | Channel used entry 2 of PM_OAM table.                     |
|                                 | AAL_EN        | 0x084                     | Enable buffer descriptor linking.                         |
|                                 | TO_LAST       | 0                         | Not last VCC entry processed for time-out.                |
|                                 | TO_EN         | 0                         | Must be initialized to 0.                                 |
|                                 | MID0          | 1                         | Allow MID value of 0.                                     |
|                                 | MID_BITS      | 0x5                       | Allow MID values up to 31.                                |
|                                 | CRC10_EN      | 1                         | Enable CRC10 field checking and error counting.           |
|                                 | LI_EN         | 1                         | Enable LI field checking and error counting.              |
|                                 | ST_EN         | 1                         | Enable ST field checking and error counting.              |
|                                 | SN_EN         | 1                         | Enable SN field checking and error counting.              |
|                                 | CPI_EN        | 1                         | Enable CPI field checking.                                |
|                                 | BAT_EN        | 1                         | Enable BASIZE = Length checking.                          |
|                                 | BAH_EN        | 0                         | Do not check for BASIZE < 37.                             |
| ER_EFCI                         | 0             | Must be initialized to 0. |   |

Table 15-8. Table of Values for Reassembly SAR Shared Memory Initialization (3 of 4)

| Table                              | Field                  | Initialized Value | Notes   |
|------------------------------------|------------------------|-------------------|---|
| RSM AAL3/4 Head<br>VCC Table Entry | ABR_CTRL               | 0x0               | No ABR service enabled on this channel.   |
|                                    | VCC_INDEX              | 0x0010            | AAL3/4 block located at VCC table entry #16.  |
|                                    | STAT                   | 0x1               | OAMs use status queue 1.  |
|                                    | BFR1                   | —                 | (not applicable.)   |
|                                    | BFR0                   | 0x1               | OAMs use free buffer queue 1.   |
|                                    | CRC10_ERR              | 0                 | Must be initialized to 0.   |
|                                    | MID_ERR                | 0                 | Must be initialized to 0.   |
|                                    | LI_ERR                 | 0                 | Must be initialized to 0.   |
|                                    | SN_ERR                 | 0                 | Must be initialized to 0.   |
|                                    | BOM_SSM_ERR            | 0                 | Must be initialized to 0.   |
|                                    | EOM_ERR                | 0                 | Must be initialized to 0.   |
|                                    | SEG_VCC_INDEX          | 0x4               | Corresponding SEG channel = 4 for PM_OAM and ABR service.   |
|                                    | RX_COUNTER/VPC_INDEX   | 0x100             | Initial firewall credit = 256.  |
| RSM Buffer<br>Descriptors          | NEXT_PTR               | 0x0               | Initialization optional.  |
|                                    | BUFF_PNTR              | 0x2000            | Buffer address of 0x2000.   |
| RSM Free Buffer<br>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 0x80.  |
|                                    | FWD_VLD                | 0                 | Must be initialized to 0.   |
|                                    | VCC_INDEX              | 0                 | Must be initialized to 0.   |
|                                    | Reserved               | 0x0               | Must be initialized to 0.   |
| RSM Status Queue<br>Entries        | VLD                    | 0                 | Must be initialized to 0.   |
|                                    | (ALL OTHER<br>ENTRIES) | 0                 | Must be initialized to 0.   |
| LECID Table                        | LECID0-31              | 0x20              | LANE LECID = 32.  |

Table 15-8. Table of Values for Reassembly SAR Shared Memory Initialization (4 of 4)

| Table                | Field    | Initialized Value | Notes  |
|----------------------|----------|-------------------|--|
| RSM_PM Table Entry 0 | BCNT     | 0x0               | Must be initialized to 0.                              |
|                      | BIP16    | 0x0               | Must be initialized to 0.                              |
|                      | MSN      | 0x4               | First expected MSN in forward monitoring cell is four. |
|                      | Reserved | 0x0               | Must be initialized to 0.                              |
|                      | TRCC0    | 0x0               | Must be initialized to 0.                              |
|                      | TRCC01   | 0x0               | Must be initialized to 0.                              |

## 15.4 General Initialization

### 15.4.1 General Control Registers

Before the SAR is enabled, the host must allocate and initialize all of the general SAR control registers. [Table 15-9](#) lists the initialized values for each field.

**Table 15-9. Table of Values for General Control Register Initialization (1 of 2)**

| Register                                | Field          | Initialized Value         | Notes   |
|---|----------------|---------------------------|---|
| CONFIG0<br>(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 0.   |
|   | INT_LBANK      | 0 – 1                     | Should be set to 0 during initialization, but set to one after system reset.  |
|   | PCI_READ_MULTI | 1                         | PCI Read Multiple Command used.   |
|   | PCI_ARB        | 1                         | Round-robin arbitration of internal read/write PCI master.  |
|   | STATMODE       | 0x0                       | Selects BOM sync hardware mode.   |
|   | FR_RMODE       | 0                         | Early RSM header processing enabled.  |
|   | FR_LOOP        | 0                         | Internal ATM physical interface disabled.   |
|   | UTOPIA_MODE    | 1                         | Cell handshake mode selected.   |
|   | LP_BWAIT       | 0                         | Selects 0 wait states between consecutive data cycles during local processor.   |
|   | MEMCTRL        | 0                         | Selects 0 wait states SAR-shared memory.  |
|   | BANKSIZE       | 0x3                       | 512 kB banks selected.  |
|   | DIVIDER        | 0x0                       | Divide by 128 selected for CLOCK prescaler.   |
| Reserved                                | 0x0            | Must be initialized to 0. |   |
| INT_DELAY<br>(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 15-9. Table of Values for General Control Register Initialization (2 of 2)

| Register   | Field                   | Initialized Value | Notes  |
|--|-------------------------|-------------------|--|
| HOST_ST_WR<br>(Host Status Write Register)         | RSM_HS_WRITE            | —                 | Read twice after SAR reset.  |
|  | RSM_LS_WRITE            | —                 | Read twice after SAR reset.  |
| HOST_I_STAT1<br>(Host Interrupt Status Register 1) | (ALL)                   | —                 | Read twice HOST_ST_WR reset.   |
| HOST_I_STAT0<br>(Host Interrupt Status Register 0) | (ALL)                   | —                 | Read twice HOST_I_STAT1 reset.                                       |
| LP_I_STAT1<br>(Local Interrupt Status Register 1)  | (ALL)                   | —                 | Read twice SAR reset.  |
| LP_I_STAT0<br>(Local Interrupt Status Register 0)  | (ALL)                   | —                 | Read twice LP_I_STAT1 reset.   |
| HOST_IMASK1<br>(Host Interrupt Mask Register 1)    | (ALL)                   | 0x8700FC07        | Enable all errors to cause an interrupt.                             |
| HOST_IMASK0<br>(Host Interrupt Mask Register 0)    | (ALL)                   | 0x4040000F        | Enable interrupts in errors, counter relievers and framer interrupt. |
| LP_IMASK1<br>(Local Interrupt Mask Register 1)     | (ALL)                   | 0x0               | Local processor not used.  |
| LP_IMASK0<br>(Local Interrupt Mask Register 0)     | (ALL)                   | 0x0               | Local processor not used.  |
| 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.  |
|  | (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 CN8236 device master on the PCI bus.                          |
|  | MS_EN                   | 1                 | Enable CN8236 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.)  |



## 16.0 Electrical and Mechanical Specifications

### 16.1 Timing

#### 16.1.1 PCI Bus Interface Timing

All PCI bus interface signals are synchronous to the PCI bus clock, HCLK, except for HRST\* and HINT\*. [Table 16-1](#) provides the PCI bus interface timing parameters. [Figures 16-1](#) and [16-2](#) illustrate this timing.

*Table 16-1. PCI Bus Interface Timing Parameters (1 of 2)*

| Symbol   | Parameter   | Min | Max | Units |
|--|---|-----|-----|-------|
| $t_{cyc}$                                      | HCLK Cycle Time <sup>(1)</sup>                      | 25  | —   | ns    |
| $t_{high}$                                     | HCLK High Time <sup>(1)</sup>                       | 10  | 15  | ns    |
| $t_{low}$                                      | HCLK Low Time <sup>(1)</sup>                        | 10  | 15  | 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    |

## 16.1 Timing

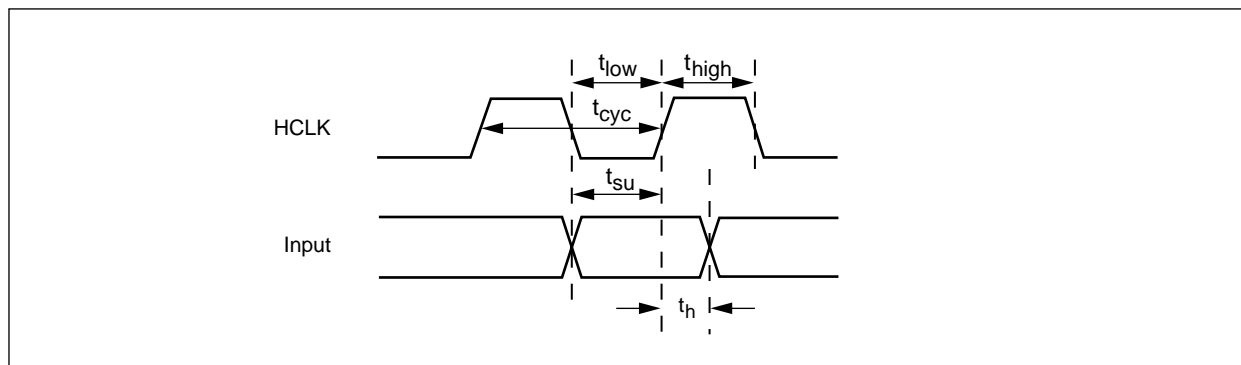
ATM ServiceSAR Plus with xBR Traffic Management

Table 16-1. 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    |
| $t_{on}$      | 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> | —   | 23  | ns    |
| $t_{rst-off}$ | Reset Active to Output Float Delay                           | —   | 40  | ns    |

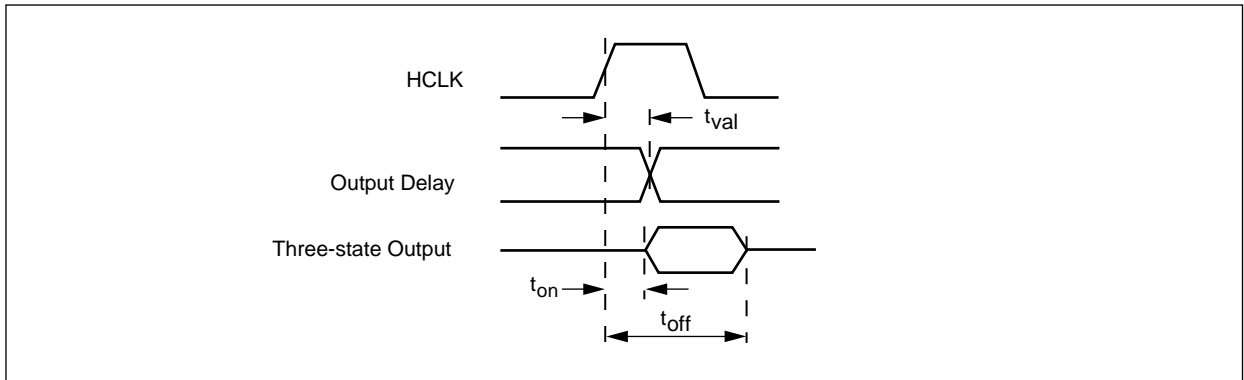
**NOTE(S):**  
 (1) See Figure 16-1 for waveforms and definitions.  
 (2) See Figure 16-2 for waveforms and definitions. The maximum output delays are measured with a 50 pF load, and the minimum delays are measured with a 0 pF load.  
 (3) Applicable when STAT outputs configured as BOM cell synchronization signals.

Figure 16-1. PCI Bus Input Timing Measurement Conditions



8236\_078

Figure 16-2. PCI Bus Output Timing Measurement Conditions



8236\_079

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

All ATM physical interface signals are synchronous to interface clocks, RxCLK and/or TxCLK, except for TxClav and RxClav in the slave UTOPIA mode.

Timing parameters for the UTOPIA interface are provided in [Table 16-2](#).

[Table 16-3](#) provides the timing parameters for the slave UTOPIA interface.

Timing diagrams for both interfaces are provided in [Figures 16-3](#) and [16-4](#).

**Table 16-2. UTOPIA Interface Timing Parameters**

| Symbol     | Parameter  | Min | Max | Units |
|------------|--|-----|-----|-------|
| $t_{cyc}$  | RxCLK/TxCLK Cycle Time <sup>(1)</sup>                  | 20  | —   | 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  | %     |
| $t_{su}$   | RxDATA Input Setup Time to RxCLK <sup>(1)</sup>        | 4   | —   | ns    |
|            | RxPAR Input Setup Time to RxCLK <sup>(1)</sup>         | 4   | —   | ns    |
|            | RxSOC Input Setup Time to RxCLK <sup>(1)</sup>         | 4   | —   | ns    |
|            | RxCLAV Input Setup Time to RxCLK <sup>(1)</sup>        | 4   | —   | ns    |
|            | TxCLAV Input Setup Time to TxCLK <sup>(1)</sup>        | 4   | —   | ns    |
| $t_h$      | RxDATA 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 TxDATA Valid Delay <sup>(2)</sup>             | 2   | 12  | ns    |
|            | TxCLK to TxPAR Valid Delay <sup>(2)</sup>              | 2   | 12  | ns    |
|            | TxCLK to TxSOC Valid Delay <sup>(2)</sup>              | 2   | 12  | ns    |
|            | TxCLK to TxEN* Valid Delay <sup>(2)</sup>              | 2   | 12  | ns    |
|            | RxCLK to RxEN* Valid Delay <sup>(2)</sup>              | 2   | 12  | ns    |

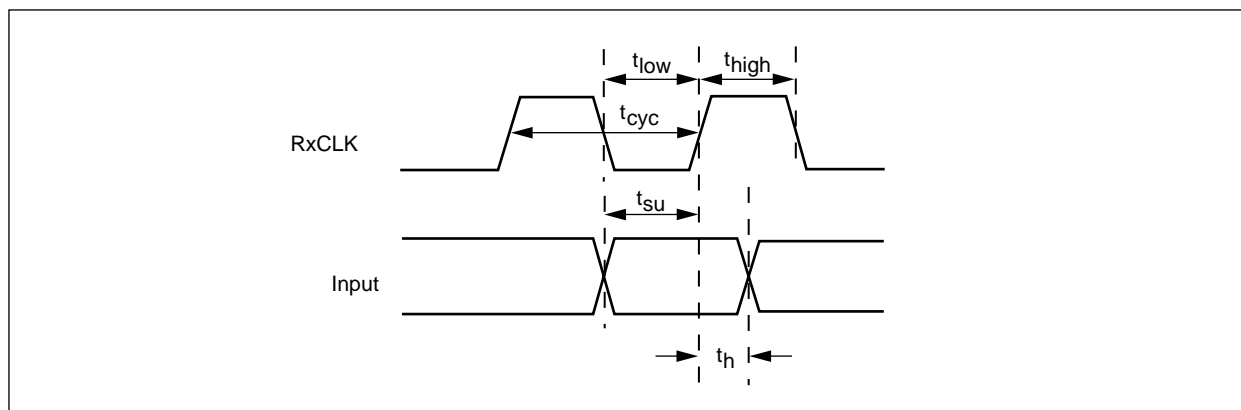
**NOTE(S):**  
 (1) See [Figure 16-3](#) for waveforms and definitions.  
 (2) See [Figure 16-4](#) for waveforms and definitions. The output delays are measured with a 50 pF load.

Table 16-3. Slave UTOPIA Interface Timing Parameters

| Symbol     | Parameter  | Min | Max | Units |
|------------|--|-----|-----|-------|
| $t_{cyc}$  | RxCLK/TxCLK Cycle Time <sup>(1)</sup>                  | 20  | —   | 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  | %     |
| $t_{su}$   | RxDATA Input Setup Time to RxCLK <sup>(1)</sup>        | 4   | —   | ns    |
|            | RxPAR Input Setup Time to RxCLK <sup>(1)</sup>         | 4   | —   | ns    |
|            | RxSOC Input Setup Time to RxCLK <sup>(1)</sup>         | 4   | —   | ns    |
|            | RxEN* Input Setup Time to RxCLK <sup>(1)</sup>         | 4   | —   | ns    |
|            | TxEN* Input Setup Time to TxCLK <sup>(1)</sup>         | 4   | —   | ns    |
| $t_h$      | RxDATA 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 TxDATA Valid Delay <sup>(2)</sup>             | 2   | 12  | ns    |
|            | TxCLK to TxPAR Valid Delay <sup>(2)</sup>              | 2   | 12  | ns    |
|            | TxCLK to TxCLAV Valid Delay <sup>(2)</sup>             | 2   | 12  | ns    |
|            | RxCLK to RxCLAV Valid Delay <sup>(2)</sup>             | 2   | 12  | ns    |
|            | TxCLK to TxSOC Valid Delay <sup>(2)</sup>              | 2   | 12  | ns    |

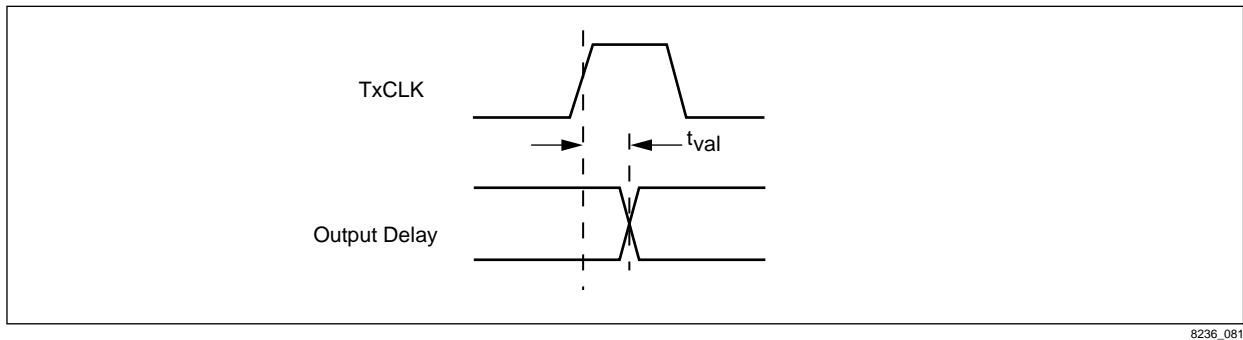
**NOTE(S):**  
 (1) See Figure 16-3 for waveforms and definitions.  
 (2) See Figure 16-4 for waveforms and definitions. The output delays are measured with a 50 pF load.

Figure 16-3. UTOPIA and Slave UTOPIA Input Timing Measurement Conditions



8236\_080

Figure 16-4. UTOPIA and Slave UTOPIA Output Timing Measurement Conditions



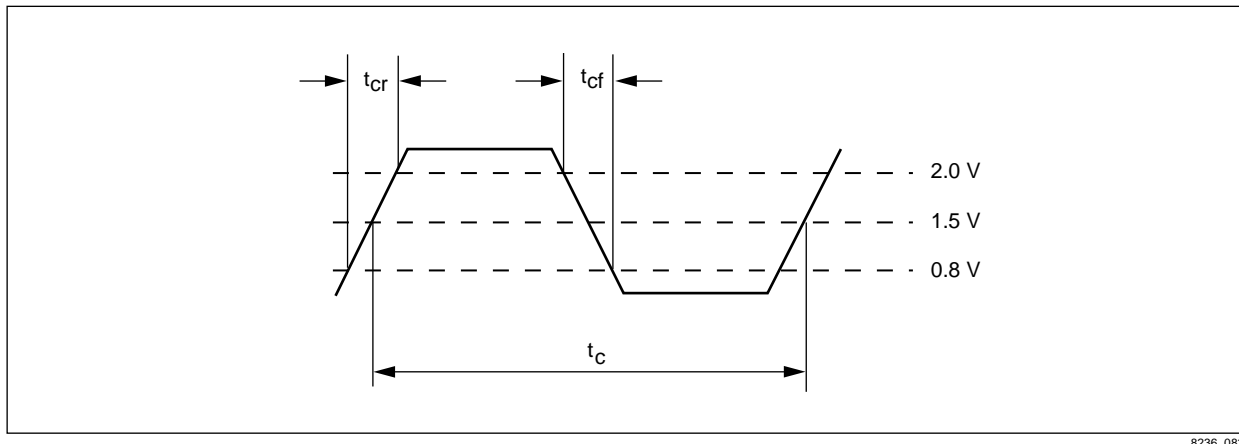
### 16.1.3 System Clock Timing

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, SAR-shared memory, and processor read and write timing. The CLKD3 output can be used as the ATM physical interface clock. There is no defined skew relationship between the CLK2X input and SYSCLK and CLKD3 outputs. [Table 16-4](#) specifies the timing parameters of the three clocks. [Figures 16-5](#) and [16-6](#) illustrate this timing.

**Table 16-4. System Clock Timing**

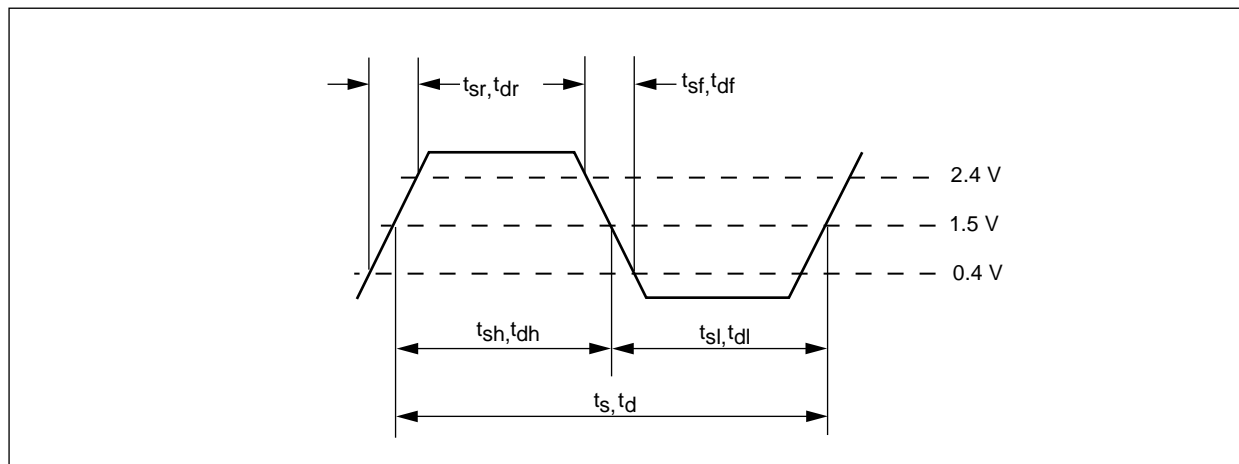
| Symbol  | Parameter        | Min           | Max           | Units |
|---|------------------|---------------|---------------|-------|
| <b>Input Clocks<sup>(1)</sup></b>   |                  |               |               |       |
| $t_{cf}$  | CLK2X Frequency  | 0             | 66            | MHz   |
| $t_c$   | 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   |
| $t_d$   | 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    |
| <b>NOTE(S):</b>   |                  |               |               |       |
| (1) See <a href="#">Figure 16-5</a> for waveforms and definitions.  |                  |               |               |       |
| (2) See <a href="#">Figure 16-6</a> for waveforms and definitions. The outputs are measured with a load of 35 pF. |                  |               |               |       |

Figure 16-5. Input System Clock Waveform



8236\_082

Figure 16-6. Output System Clock Waveform



8236\_083



### 16.1.4 CN8236 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 16-5 lists the number of loads per bank for the different SRAM organizations, while Table 16-6 lists the capacitive loading used in the timing specifications given for the three typical implementations. CN8236 memory interface timing is provided in Table 16-7. (See Section 9.2, for details of memory bank organization.)

Table 16-5. 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         |
| MWE <sup>x</sup> * <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 16-6. 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 |
|---|----------------------|----------------------|---------------------|-------|
| Memory Interface Loading <sup>(1)</sup> |                      |                      |                     |       |
| 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.

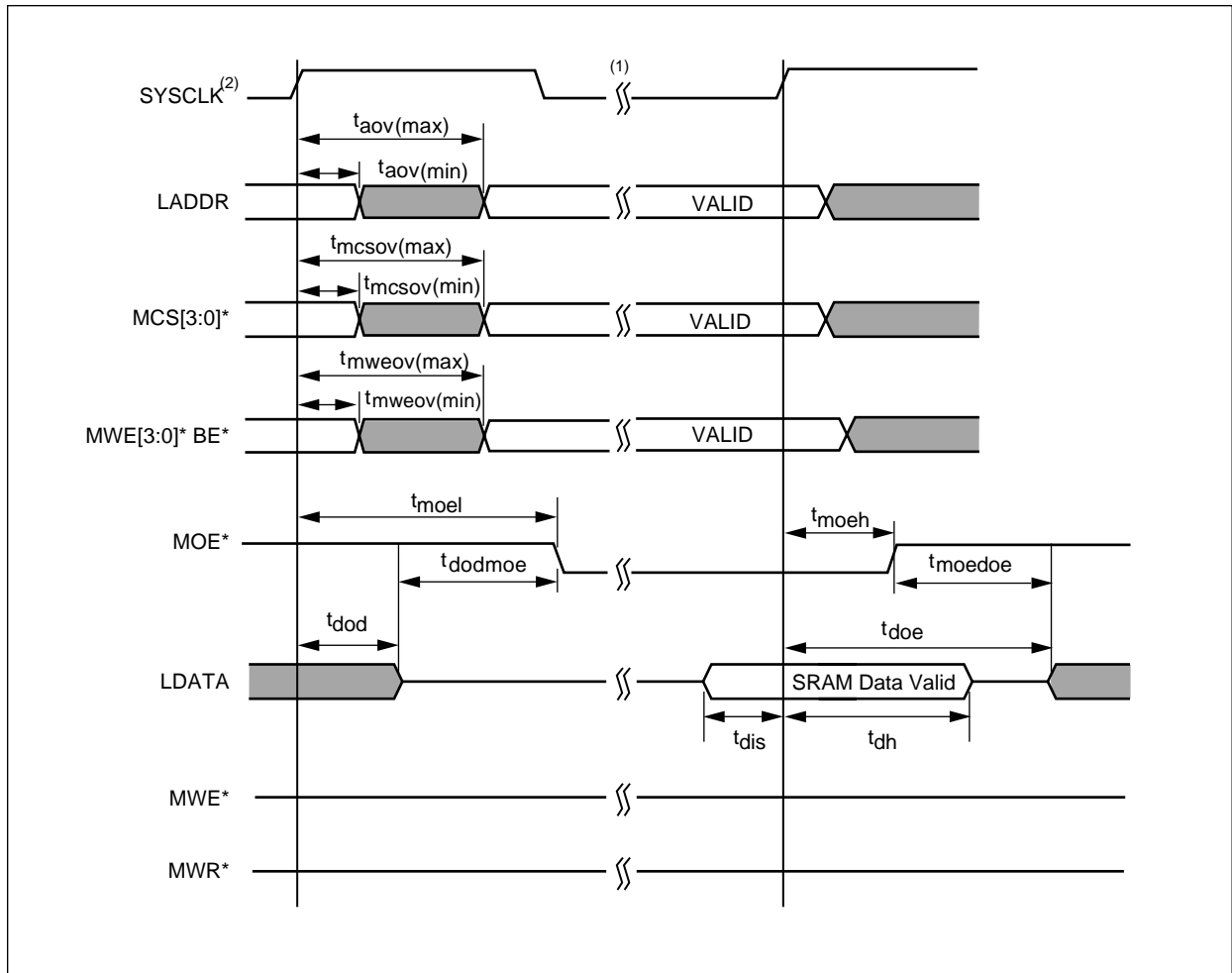
## 16.1 Timing

## ATM ServiceSAR Plus with xBR Traffic Management

Table 16-7. CN8236 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><br>(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><br>(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    |
| $t_{mwel}$   | 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 <a href="#">Figure 16-7</a> for waveforms.   |  |                      |     |                      |     |                     |     |       |
| (2) T = $t_s$ for single cycle memory (no wait states), T = $2t_s$ for 2-cycle memory, (1 wait state). |  |                      |     |                      |     |                     |     |       |
| (3) Insert 1 clock cycle for 2-cycle memory (1 wait state).  |  |                      |     |                      |     |                     |     |       |
| (4) See <a href="#">Figure 16-8</a> for waveforms.   |  |                      |     |                      |     |                     |     |       |
| 5. SYSCLK shown for reference only.  |  |                      |     |                      |     |                     |     |       |

Figure 16-7. CN8236 Memory Read Timing

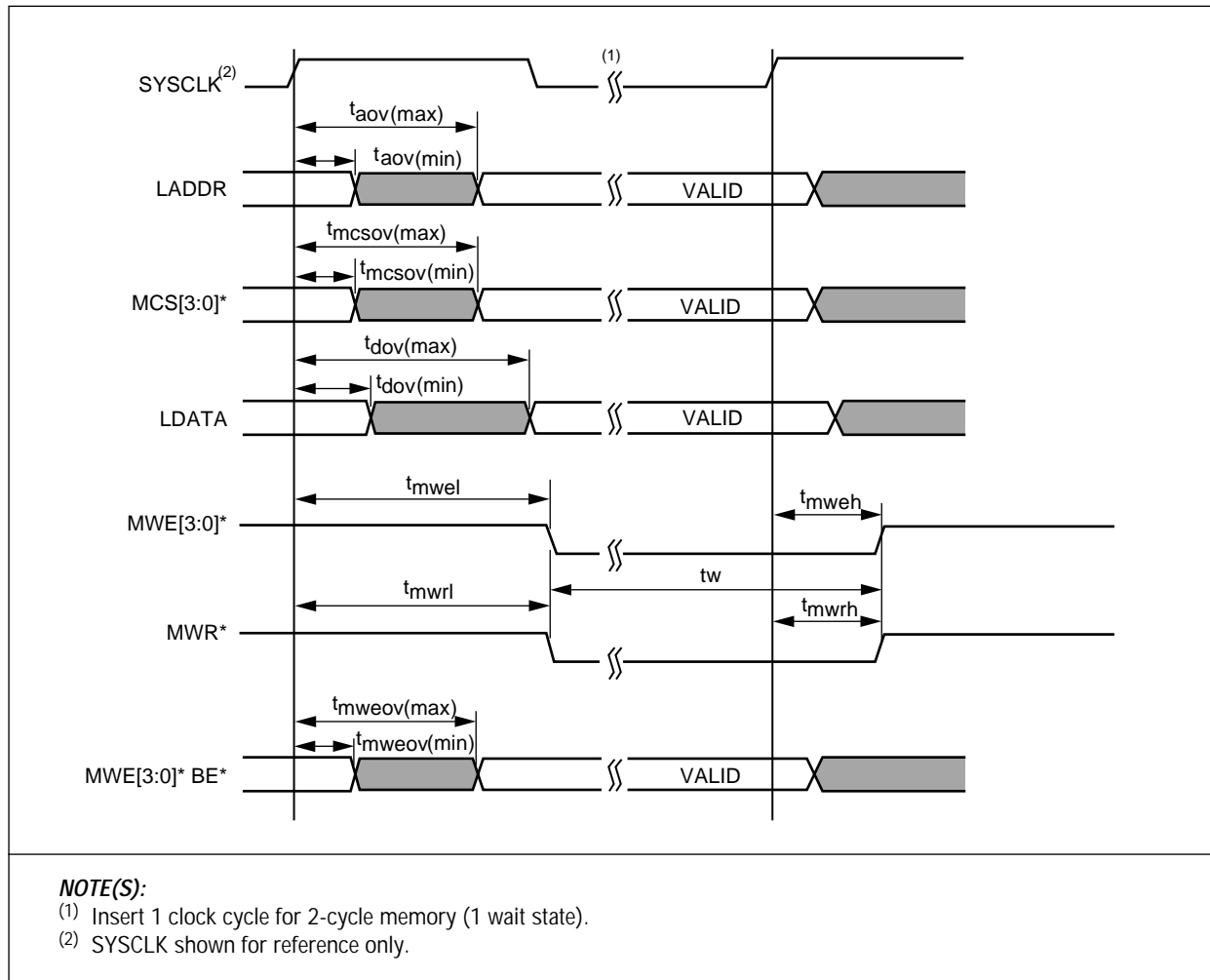


**NOTE(S):**

- (1) Insert one clock cycle for two cycle memory (one wait state).
- (2) SYSCLK shown for reference only.

8236\_117

Figure 16-8. CN8236 Memory Write Timing



8236\_085

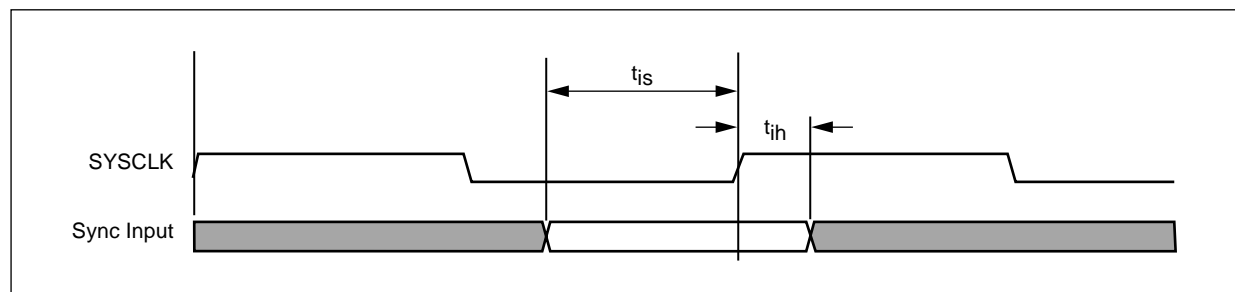
### 16.1.5 PHY Interface Timing (Standalone Mode)

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 CN8236 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 provided in [Table 16-8](#) and [Figures 16-9](#) and [16-10](#). (See [Section 10.6](#) for details.)

**Table 16-8. PHY Interface Timing (PROCMODE = 1)**

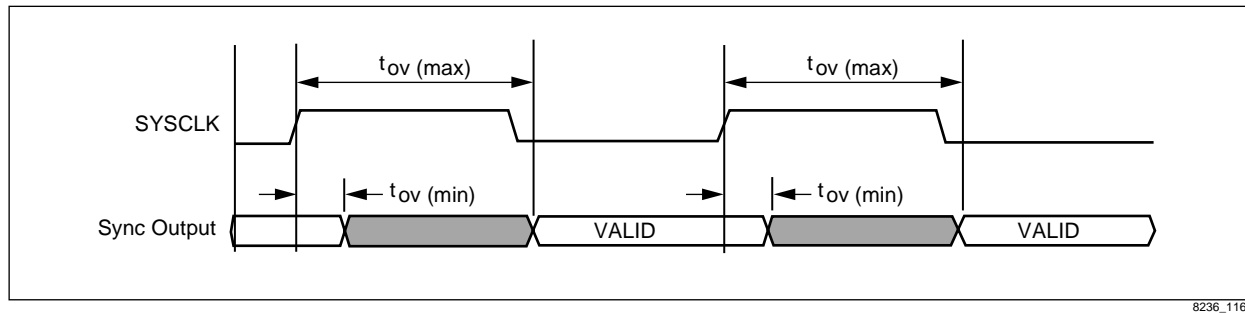
| Symbol   | Parameter                                 | Min | Max | Units |
|--|---|-----|-----|-------|
| <b>Synchronous Inputs</b>  |   |     |     |       |
| $t_{is}$   | PWAIT* Input Setup <sup>(1)</sup>         | 14  | —   | ns    |
|  | LDATA Input Setup <sup>(1)</sup>          | 5   | —   | 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    |
|  | PBLAST* 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 16-9</a> for waveforms and definitions.   |   |     |     |       |
| (2) See <a href="#">Figure 16-10</a> for waveforms and definitions. The outputs are measured with a load of 35 pF. |   |     |     |       |

**Figure 16-9. Synchronous PHY Interface Input Timing**



8236\_115

Figure 16-10. Synchronous PHY Interface Output Timing



8236\_116

### 16.1.6 Local Processor Interface Timing

Timing for the local processor interface can be broken into two sections. The first is the synchronous-to-SYSCLK interface of processor control signals to and from the CN8236. The second is the interface to the SAR-shared memory and the CN8236 control and status registers, which involves both SRAM and transceiver, and buffer timing parameters that are not specified and are left up to the system designer.

All of the synchronous interface signals are inputs to the CN8236 except for SYSCLK and the ready output of the CN8236 (PRDY\*). The output loading of these signals is 35 pF for the timing provided in Table 16-9. Synchronous processor interface timing is provided in Figures 16-11 and 16-12. Memory interface timing is provided in Table 16-10. Local interface processor and interfacing timing is provided in Figures 16-13 and 16-14.

Table 16-9. Synchronous Processor Interface Timing (1 of 2)

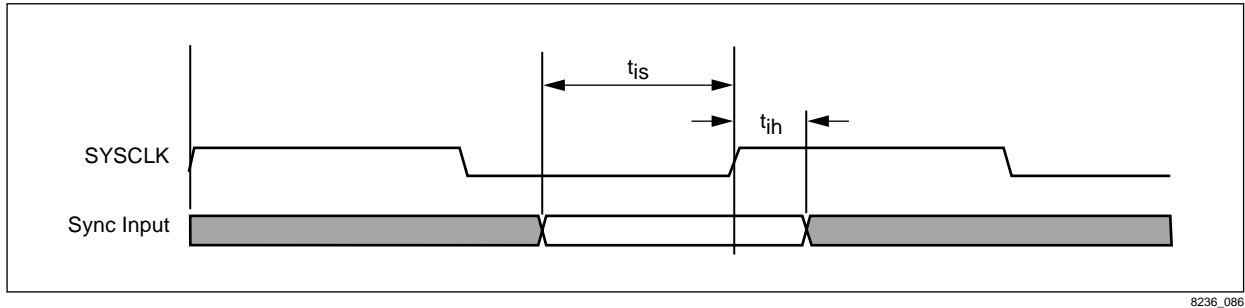
| Symbol             | Parameter                             | Min | Max | Units |
|--------------------|---------------------------------------|-----|-----|-------|
| Synchronous Inputs |                                       |     |     |       |
| $t_{is}$           | PCS* Input Setup <sup>(1)</sup>       | 8   | —   | ns    |
|                    | PAS* Input Setup <sup>(1)</sup>       | 10  | —   | ns    |
|                    | PBLAST* Input Setup <sup>(1)</sup>    | 10  | —   | ns    |
|                    | PWAIT* Input Setup <sup>(1)</sup>     | 10  | —   | ns    |
|                    | PADDR[1,0] Input Setup <sup>(1)</sup> | 10  | —   | ns    |
|                    | PBSEL[1,0] Input Setup <sup>(1)</sup> | 8   | —   | ns    |
|                    | PBE[3:0]* Input Setup <sup>(1)</sup>  | 8   | —   | ns    |
|                    | PWNR Input Setup <sup>(1)</sup>       | 10  | —   | ns    |

Table 16-9. Synchronous Processor Interface Timing (2 of 2)

| Symbol                     | Parameter                               | Min | Max | Units |
|----------------------------|---|-----|-----|-------|
| $t_{ih}$                   | PCS* Input Hold <sup>(1)</sup>          | 0   | —   | ns    |
|                            | PAS* Input Hold <sup>(1)</sup>          | 0   | —   | ns    |
|                            | PBLAST* Input Hold <sup>(1)</sup>       | 0   | —   | ns    |
|                            | PWAIT* Input Hold <sup>(1)</sup>        | 0   | —   | ns    |
|                            | PADDR[1,0] Input Hold <sup>(1)</sup>    | 0   | —   | ns    |
|                            | PBSEL[1,0] Input Hold <sup>(1)</sup>    | 0   | —   | ns    |
|                            | PBE[3:0]* Input Hold <sup>(1)</sup>     | 0   | —   | ns    |
|                            | PWNR Input Hold <sup>(1)</sup>          | 0   | —   | ns    |
| <b>Synchronous Outputs</b> |   |     |     |       |
| $t_{ov}$                   | PRDY* Output Valid Delay <sup>(2)</sup> | 5   | 15  | ns    |

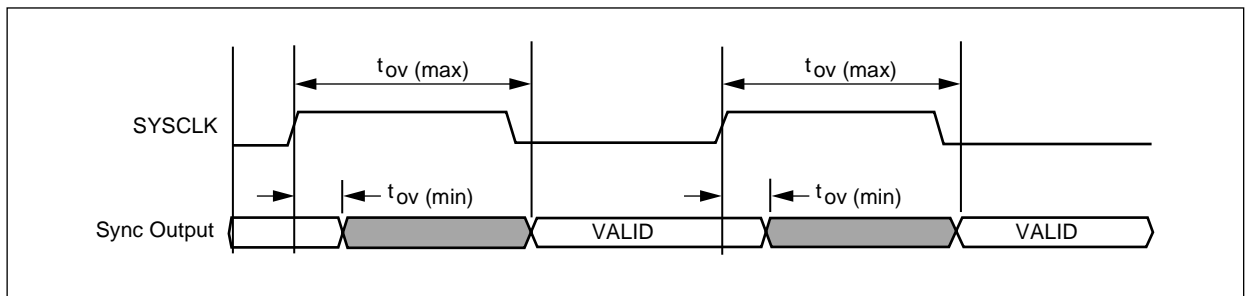
**NOTE(S):**  
 (1) See Figure 16-11 for waveforms and definitions.  
 (2) See Figure 16-12 for waveforms and definitions. The outputs are measured with a load of 35 pF.

Figure 16-11. Synchronous Local Processor Input Timing



8236\_086

Figure 16-12. Synchronous Local Processor Output Timing



8236\_087

## 16.1 Timing

ATM ServiceSAR Plus with xBR Traffic Management

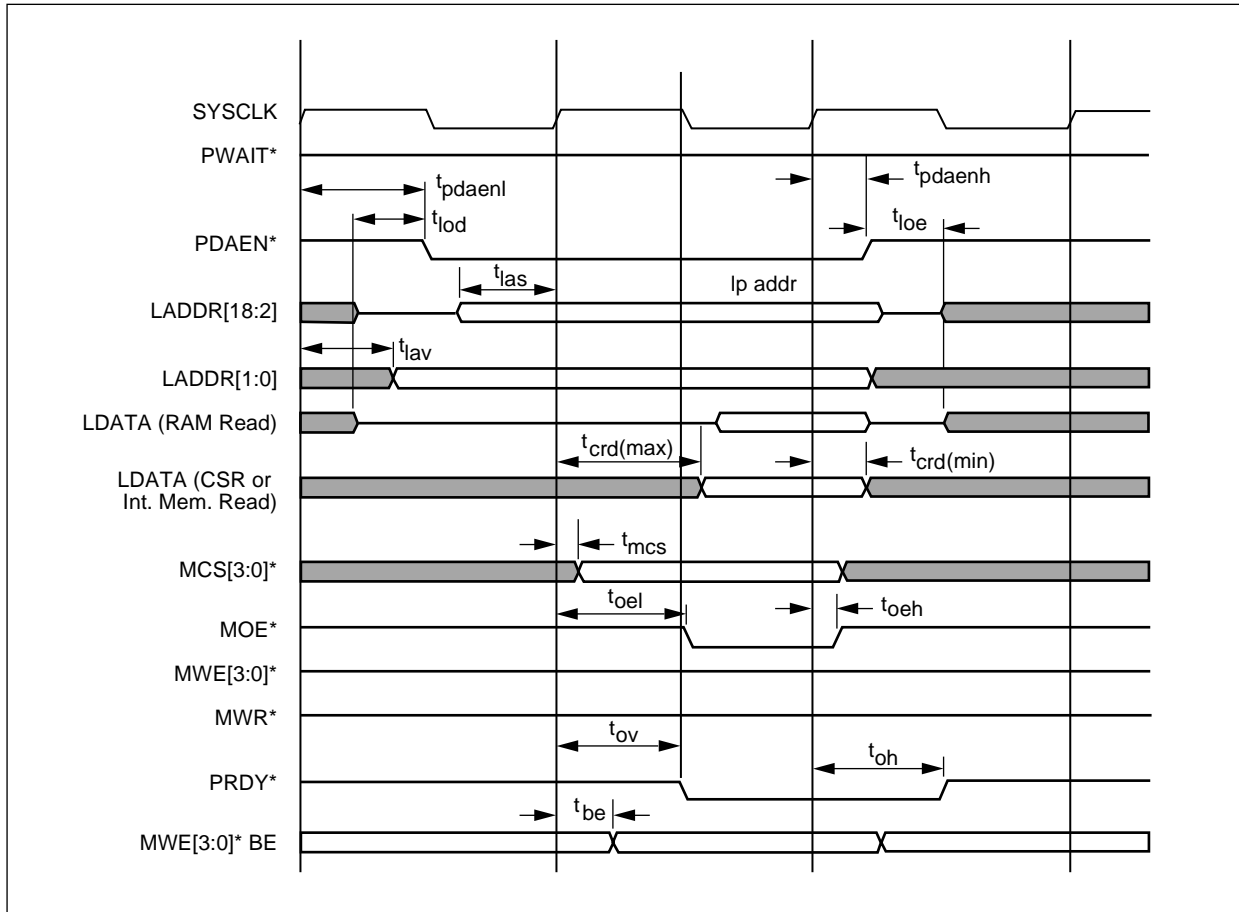
Table 16-10. Local Processor Memory Interface Timing

| Symbol       | Parameter  | Min | Max | Units |
|--------------|--|-----|-----|-------|
| $t_{pdaenl}$ | SYSCCLK to PDAEN* Low, Bus Recovery Cycle <sup>(1)</sup>                           | —   | 12  | ns    |
| $t_{pdaenh}$ | SYSCCLK to PDAEN* High, Bus Recovery Cycle <sup>(1)</sup>                          | 2   | —   | ns    |
| $t_{lod}$    | LADDR[18:2], LDATA Output Disable to PDAEN* Low, Bus Recovery Cycle <sup>(1)</sup> | 4   | —   | ns    |
| $t_{loe}$    | PDAEN* High to LADDR[18:2], LDATA Output Enable, Bus Recovery Cycle <sup>(1)</sup> | 9   | —   | ns    |
| $t_{mcs}$    | SYSCCLK to MCS*[3:0] Valid <sup>(1)</sup>  | 1   | 6   | ns    |
| $t_{lav}$    | SYSCCLK to LADDR[1,0] Valid <sup>(1, 2)</sup>                                      | 1   | 7   | ns    |
| $t_{oel}$    | MOE* Active from SYSCCLK <sup>(1, 3)</sup>   | 8   | 20  | ns    |
| $t_{oeh}$    | MOE* Inactive from SYSCCLK <sup>(1, 3)</sup>                                       | 1   | 10  | ns    |
| $t_{wl}$     | MWR*, MWE[3:0] Active from SYSCCLK <sup>(1, 3)</sup>                               | —   | 16  | ns    |
| $t_{wh}$     | MWR*, MWE[3:0]* Inactive to SYSCCLK <sup>(1, 3)</sup>                              | —   | 1   | ns    |
| $t_{be}$     | MWE[3:0]* Byte Enables Valid from SYSCCLK (RAMMODE = 1) <sup>(1)</sup>             | 1   | 7   | ns    |
| $t_{crd}$    | CSR Read Data Output Valid   | 2   | 20  | ns    |
| $t_{cwds}$   | CSR Write Data Setup to SYSCCLK  | 8   | —   | ns    |
| $t_{cwdh}$   | CSR Write Data Hold from SYSCCLK   | 0   | —   | ns    |
| $t_{las}$    | LADDR Setup to SYSCCLK   | 12  | —   | ns    |

**NOTE(S):**  
(1) See Figures 16-13 and 16-14 for waveforms and definitions.  
(2)  $t_{lav}$  is valid for second and subsequent accesses during burst transfers. See functional timing diagrams.  
(3) In the case of two cycle memory, or when inserting wait states by PWAIT\*, MOE\*, MWE[3:0]\*, and MWR\* are extended across 2 or more clock cycles with the same relative timing to SYSCCLK. See the functional timing diagrams in Section 10.6.

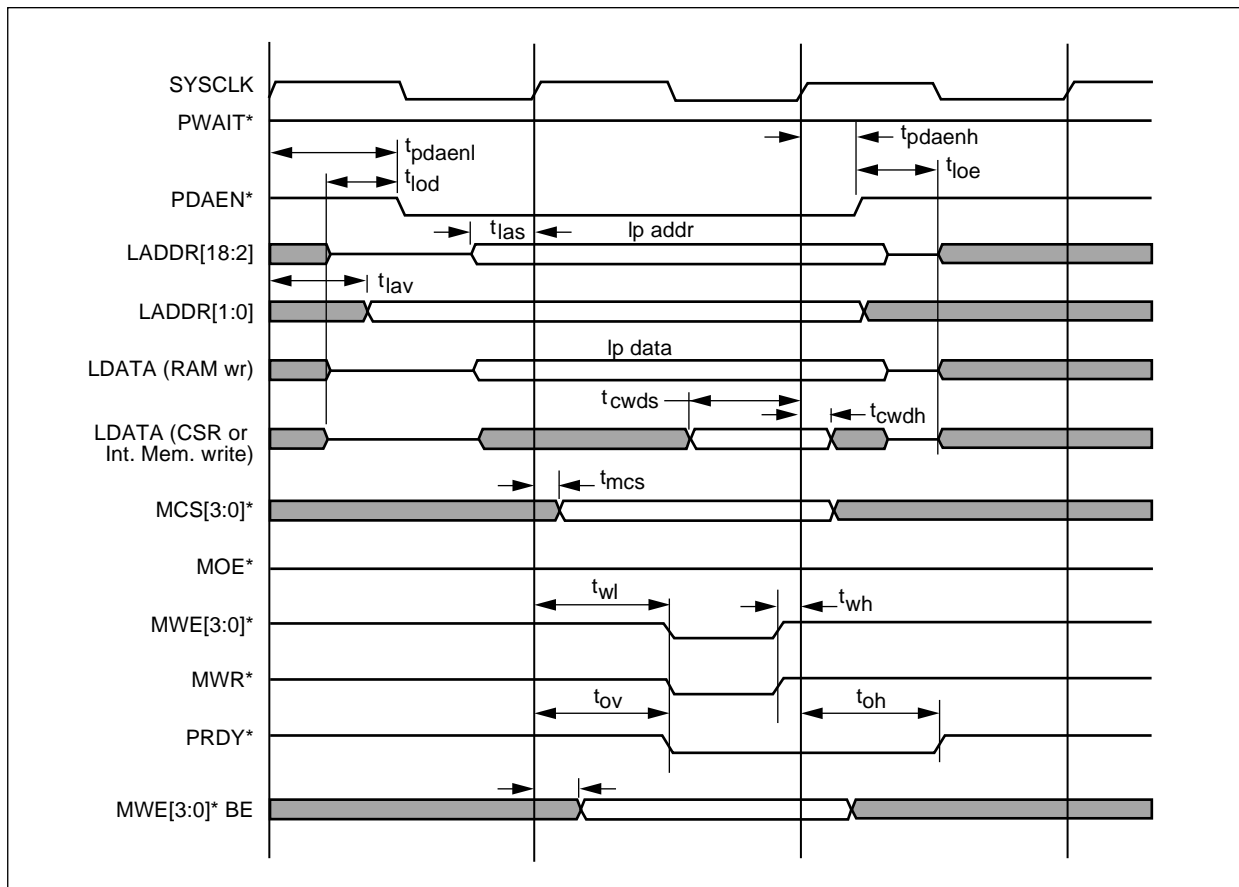


Figure 16-13. Local Processor Read Timing



8236\_088

Figure 16-14. Local Processor Write Timing



8236\_089

## 16.2 Absolute Maximum Ratings

Stresses above those listed as absolute maximum ratings (Table 16-11) can 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 exceeding 5.5 V can induce destructive latchup.

**Table 16-11. Absolute Maximum Ratings**

| Parameter  | Value               | Unit |
|--|---------------------|------|
| Supply Voltage (Vdd)                                       | -0.5 to +4          | V    |
| Input Voltage  | -0.5 to +5.5        | V    |
| Output Voltage   | -0.5 to (Vdd + 0.3) | V    |
| Storage Temperature  | -40 to 125          | C    |
| Maximum Current @ Max Clock Frequencies and 3.6 V          | 400                 | mA   |
| Moisture Sensitivity Level (MSL)                           | 4                   | —    |
| FIT @ 55 °C  | 55                  | —    |
| Theta Ja (T <sub>a</sub> = 85 °C, T <sub>j</sub> = 125 °C) | 17                  | C/W  |

## 16.3 DC Characteristics

The DC electrical characteristics are listed in [Table 16-12](#).

**Table 16-12. DC Characteristics (1 of 2)**

| Parameter                                   | Conditions  | Min                | Max                | Unit          |
|---|---|--------------------|--------------------|---------------|
| Operating Supply Voltage (Vdd)              | —   | 3.0                | 3.6                | V             |
| Output Voltage High                         | $I_{OH} = -500 \mu\text{A}$<br>(for all 3.3 V PCI signaling)                        | $0.9 \cdot V_{dd}$ | —                  | V             |
|   | $I_{OH} = -2 \text{ mA}$<br>(for all 5 V PCI signaling)                             | 2.4                | —                  | V             |
|   | $I_{OH} = 4.0 \text{ mA}$<br>(for all other signals)                                | 2.4                | —                  | V             |
| Output Voltage Low                          | $I_{OL} = 1500 \mu\text{A}$<br>(for all 3.3 V PCI signals)                          | —                  | $0.1 \cdot V_{dd}$ | V             |
|   | $I_{OH} = 3 \text{ mA}, 6 \text{ mA}$ <sup>(1)</sup><br>(for all 5 V PCI signaling) | —                  | 0.55               | V             |
|   | $I_{OL} = 4.0 \text{ mA}$<br>(for all other signals)                                | —                  | 0.4                | V             |
| Input Voltage High<br>(PCI)                 | (for 3.3 V PCI signaling)   | $0.5 \cdot V_{dd}$ | 5.25               | V             |
|   | (for 5 V PCI signaling)   | 2.0                | 5.25               | V             |
| Input Voltage High<br>(CLK2X, RxClk, TxClk) | —   | $0.7 \cdot V_{dd}$ | 5.25               | V             |
| Input Voltage High<br>(all others)          | —   | 2.0                | 5.25               | V             |
| Input Voltage Low<br>(PCI)                  | (for 3.3 V PCI signaling)   | -0.5               | $0.3 \cdot V_{dd}$ | V             |
|   | (for 5 V PCI signaling)   | -0.5               | 0.8                | V             |
| Input Voltage Low<br>(CLK2X, RxClk, TxClk)  | —   | 0                  | $0.3 \cdot V_{dd}$ | V             |
| Input Voltage Low<br>(all others)           | —   | 0                  | 0.8                | V             |
| Input Leakage Current                       | $V_{in} = V_{dd} \text{ or } GND$   | -10                | 10                 | $\mu\text{A}$ |

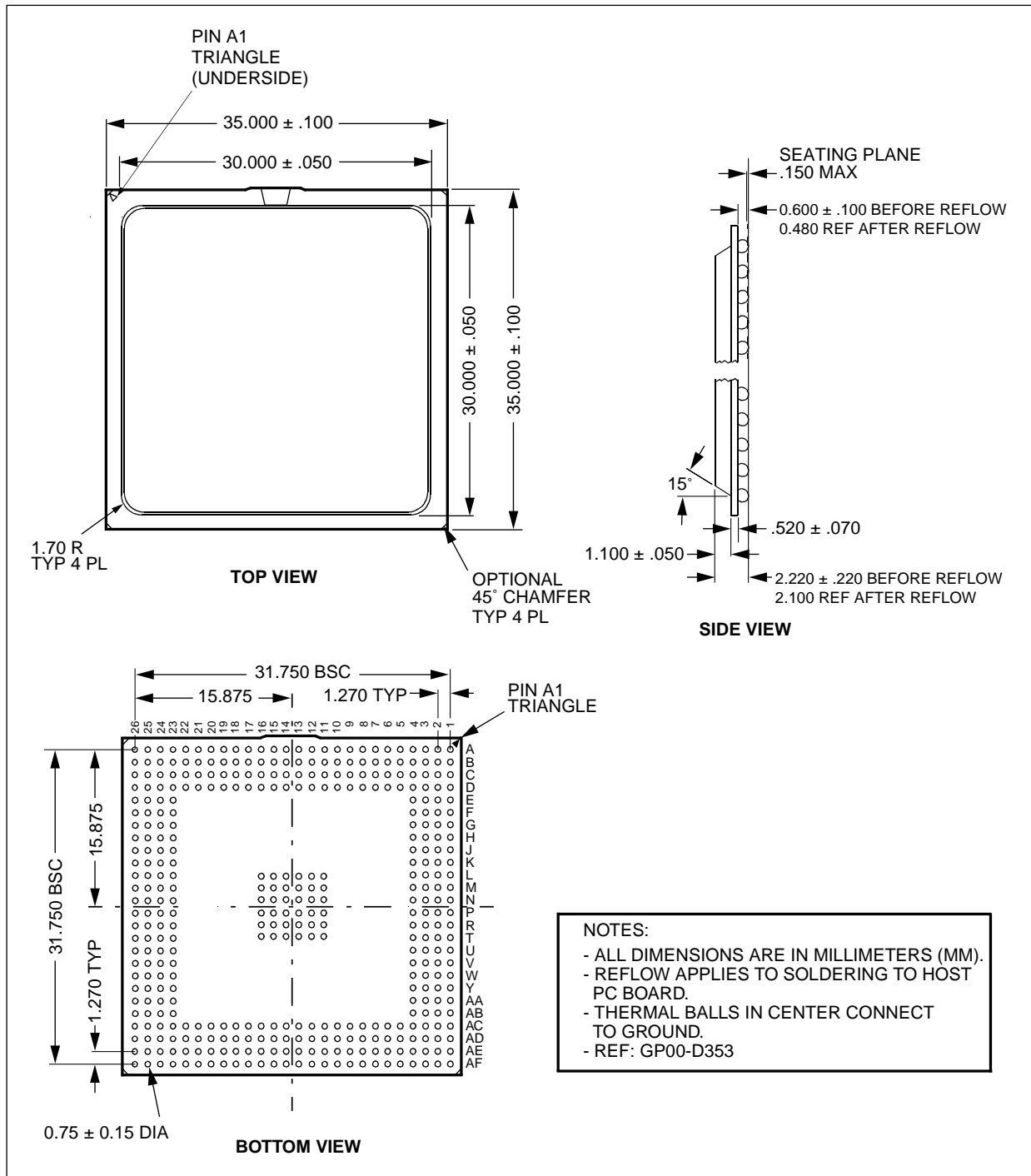
Table 16-12. DC Characteristics (2 of 2)

| Parameter  | Conditions        | Min | Max | Unit |
|--|-------------------|-----|-----|------|
| Three-state Output Leakage Current   | VOUT = Vdd or GND | -10 | 10  | μA   |
| Pullup Current   | —                 | 30  | 100 | μA   |
| Pulldown Current   | —                 | 30  | 100 | μA   |
| Input Capacitance  | —                 | —   | 7   | pF   |
| Output Capacitance   | —                 | —   | 7   | pF   |
| <p><b>NOTE(S):</b><br/>           All outputs are CMOS drive levels, and can be used with CMOS or TTL logic.<br/>           (1) Per PCI Specification Rev 2.1, signals without pullup resistors must have 3 mA low output current. Signals requiring pullup must have 6 mA. The latter include: FRAME*, TRDY*, IRDY*, DEVSEL*, STOP*, SERR*, PERR*, and LOCK*.</p> |                   |     |     |      |

## ***16.4 Mechanical Specifications***

The CN8236 388-pin BGA package is illustrated in [Figure 16-15](#). [Figure 16-16](#) illustrates a pinout configuration of the CN8236, and pin listings are provided in [Tables 16-13](#) and [16-14](#).

Figure 16-15. 388-Pin Ball Grid Array Package (BGA)



8236\_090





Table 16-13. Pin Description (Numeric List) (1 of 4)

| PIN | PIN LABEL | I/O | PIN | PIN LABEL | I/O | PIN | PIN LABEL | I/O |
|-----|-----------|-----|-----|-----------|-----|-----|-----------|-----|
| A1  | GND       | —   | B10 | PADDR1    | I   | C19 | spare     | —   |
| A2  | VDD       | —   | B11 | VDD       | —   | C20 | GND       | —   |
| A3  | LADDR0    | I/O | B12 | PBE3_     | I   | C21 | spare     | —   |
| A4  | LADDR3    | I/O | B13 | PRDY_     | O   | C22 | spare     | —   |
| A5  | LADDR7    | I/O | B14 | GND       | —   | C23 | HFIFORD5  | I   |
| A6  | LADDR9    | I/O | B15 | PDAEN_    | I/O | C24 | HFIFORD2  | I   |
| A7  | LADDR13   | I/O | B16 | PROCMODE  | I   | C25 | HFIFORD0  | I   |
| A8  | LADDR15   | I/O | B17 | spare     | —   | C26 | GND       | —   |
| A9  | VDD       | —   | B18 | spare     | —   | D1  | HFIFOWR3  | I   |
| A10 | PBSEL0    | I   | B19 | VDD       | —   | D2  | HFIFOWR2  | I   |
| A11 | GND       | —   | B20 | spare     | —   | D3  | HFIFOWR1  | I   |
| A12 | VDD       | —   | B21 | spare     | —   | D4  | spare     | —   |
| A13 | PWAIT_    | I   | B22 | spare     | —   | D5  | LADDR4    | I/O |
| A14 | VDD       | —   | B23 | HENUM_    | OD  | D6  | VDD       | —   |
| A15 | PCS_      | I/O | B24 | GND       | —   | D7  | LADDR10   | I/O |
| A16 | PRST_     | O   | B25 | VDD       | —   | D8  | VDD       | —   |
| A17 | spare     | —   | B26 | HFIFORD1  | I   | D9  | LADDR16   | I/O |
| A18 | GND       | —   | C1  | GND       | —   | D10 | GND       | —   |
| A19 | spare     | —   | C2  | HFIFOWR0  | I   | D11 | PBSEL1    | I   |
| A20 | spare     | —   | C3  | spare     | —   | D12 | PBE1_     | I   |
| A21 | spare     | —   | C4  | LADDR1    | I/O | D13 | GND       | —   |
| A22 | spare     | —   | C5  | LADDR5    | I/O | D14 | PAS_      | I/O |
| A23 | GND       | —   | C6  | GND       | —   | D15 | PINT_     | OD  |
| A24 | VDD       | —   | C7  | LADDR11   | I/O | D16 | GND       | —   |
| A25 | HFIFORD4  | I   | C8  | GND       | —   | D17 | spare     | —   |
| A26 | GND       | —   | C9  | LADDR17   | I/O | D18 | spare     | —   |
| B1  | spare     | —   | C10 | PADDR0    | I   | D19 | spare     | —   |
| B2  | VDD       | —   | C11 | PBE0_     | I   | D20 | spare     | —   |
| B3  | GND       | —   | C12 | PBE2_     | I   | D21 | VDD       | —   |
| B4  | LADDR2    | I/O | C13 | PWNR      | I/O | D22 | EPPWR     | O   |
| B5  | LADDR6    | I/O | C14 | PBLAST_   | I/O | D23 | HFIFORD3  | I   |
| B6  | LADDR8    | I/O | C15 | PFAIL_    | I   | D24 | HAD0      | I/O |
| B7  | LADDR12   | I/O | C16 | VDD       | —   | D25 | HAD1      | I/O |
| B8  | LADDR14   | I/O | C17 | spare     | —   | D26 | HAD2      | I/O |
| B9  | LADDR18   | I/O | C18 | spare     | —   | E1  | VDD       | —   |

Table 16-13. Pin Description (Numeric List) (2 of 4)

| PIN | PIN LABEL | I/O | PIN | PIN LABEL | I/O | PIN | PIN LABEL | I/O |
|-----|-----------|-----|-----|-----------|-----|-----|-----------|-----|
| E2  | GND       |     | J23 | VDD       | —   | N26 | HCBE2_    | I/O |
| E3  | HFIFOWR4  | I   | J24 | HAD13     | I/O | P1  | LDATA30   | I/O |
| E4  | VGG       | I   | J25 | HAD14     | I/O | P2  | LDATA29   | I/O |
| E23 | HAD3      | I/O | J26 | HAD15     | I/O | P3  | LDATA28   | I/O |
| E24 | VDD       | —   | K1  | VDD       | —   | P4  | VDD       | —   |
| E25 | GND       | —   | K2  | GND       | —   | P23 | GND       | —   |
| E26 | HAD4      | I/O | K3  | MCS0_     | O   | P24 | HAD18     | I/O |
| F1  | STAT0     | O   | K4  | MCS1_     | O   | P25 | HAD17     | I/O |
| F2  | STAT1     | O   | K23 | HCBE1_    | I/O | P26 | HAD16     | I/O |
| F3  | HSWITCH_  | I   | K24 | HPAR      | I/O | R1  | LDATA27   | I/O |
| F4  | HFIFOWR5  | I   | K25 | VDD       | —   | R2  | LDATA26   | I/O |
| F23 | HAD5      | I/O | K26 | GND       | —   | R3  | LDATA25   | I/O |
| F24 | HAD6      | I/O | L1  | VDD       | —   | R4  | LDATA24   | I/O |
| F25 | HAD7      | I/O | L2  | CLKD3     | O   | R23 | HAD22     | I/O |
| F26 | HCBE0_    | I/O | L3  | VDD       | —   | R24 | HAD21     | I/O |
| G1  | GND       |     | L4  | HLED_     | OD  | R25 | HAD20     | I/O |
| G2  | MWE2_     | O   | L23 | HSERR_    | OD  | R26 | HAD19     | I/O |
| G3  | MWE3_     | O   | L24 | HPERR_    | I/O | T1  | GND       | —   |
| G4  | RAMMODE   | I   | L25 | HSTOP_    | I/O | T2  | VDD       | —   |
| G23 | VDD       | —   | L26 | HDEVSEL_  | I/O | T3  | LDATA23   | I/O |
| G24 | GND       | —   | M1  | GND       | —   | T4  | LDATA22   | I/O |
| G25 | HAD8      | I/O | M2  | SYSCCLK   | O   | T23 | HCBE3_    | I/O |
| G26 | HAD9      | I/O | M3  | GND       | —   | T24 | HIDSEL    | I   |
| H1  | MOE_      | O   | M4  | spare     | —   | T25 | HAD23     | I/O |
| H2  | MWE0_     | O   | M23 | HTRDY_    | I/O | T26 | VDD       | —   |
| H3  | MWE1_     | O   | M24 | HIRDY_    | I/O | U1  | LDATA21   | I/O |
| H4  | VDD       | —   | M25 | HFRAME_   | I/O | U2  | VDD       | —   |
| H23 | HAD10     | I/O | M26 | GND       |     | U3  | LDATA20   | I/O |
| H24 | HAD11     | I/O | N1  | LDATA31   | I/O | U4  | LDATA19   | I/O |
| H25 | HAD12     | I/O | N2  | GND       | —   | U23 | HAD25     | I/O |
| H26 | GND       | —   | N3  | VDD       | —   | U24 | HAD24     | I/O |
| J1  | MCS2_     | O   | N4  | CLK2X     | I   | U25 | GND       | —   |
| J2  | MCS3_     | O   | N23 | HCLK      | I   | U26 | VDD       | —   |
| J3  | VDD       | —   | N24 | VDD       | —   | V1  | LDATA18   | I/O |
| J4  | MWR_      | O   | N25 | GND       | —   | V2  | GND       | —   |

Table 16-13. Pin Description (Numeric List) (3 of 4)

| PIN  | PIN LABEL | I/O | PIN  | PIN LABEL | I/O | PIN  | PIN LABEL | I/O |
|------|-----------|-----|------|-----------|-----|------|-----------|-----|
| V3   | LDATA17   | I/O | AB24 | SDA       | I/O | AD7  | RXD8      | I   |
| V4   | LDATA16   | I/O | AB25 | spare     | —   | AD8  | RXD4      | I   |
| V23  | VDD       | —   | AB26 | PCI5V     | I   | AD9  | GND       | —   |
| V24  | HAD28     | I/O | AC1  | LDATA2    | I/O | AD10 | RXEN_     | I/O |
| V25  | HAD27     | I/O | AC2  | LDATA1    | I/O | AD11 | spare     | —   |
| V26  | HAD26     | I/O | AC3  | LDATA0    | I/O | AD12 | TXCLK     | I   |
| W1   | LDATA15   | I/O | AC4  | GND       | —   | AD13 | GND       | —   |
| W2   | LDATA14   | I/O | AC5  | VDD       | —   | AD14 | UTOPIA1   | I   |
| W3   | LDATA13   | I/O | AC6  | RXD13     | I   | AD15 | spare     | —   |
| W4   | GND       | —   | AC7  | RXD9      | I   | AD16 | VDD       | —   |
| W23  | GND       | —   | AC8  | RXD5      | I   | AD17 | TXD12     | O   |
| W24  | HAD31     | I/O | AC9  | RXD1      | I   | AD18 | TXD9      | O   |
| W25  | HAD30     | I/O | AC10 | RXSOC     | I   | AD19 | GND       | —   |
| W26  | HAD29     | I/O | AC11 | spare     | —   | AD20 | TXD3      | O   |
| Y1   | LDATA12   | I/O | AC12 | VDD       | —   | AD21 | VDD       | —   |
| Y2   | LDATA11   | I/O | AC13 | RXCLK     | I   | AD22 | TXADDR1   | I/O |
| Y3   | LDATA10   | I/O | AC14 | TXEN_     | I/O | AD23 | spare     | —   |
| Y4   | LDATA9    | I/O | AC15 | spare     | —   | AD24 | spare     | —   |
| Y23  | HRST_     | I   | AC16 | TXD15     | O   | AD25 | TRST_     | I   |
| Y24  | GND       | —   | AC17 | TXD11     | O   | AD26 | TMS       | I   |
| Y25  | HGNT_     | I   | AC18 | TXD8      | O   | AE1  | SCHREF    | I   |
| Y26  | HREQ_     | O   | AC19 | VDD       | —   | AE2  | VDD       | —   |
| AA1  | LDATA8    | I/O | AC20 | TXD2      | O   | AE3  | RXADDR3   | I/O |
| AA2  | GND       | —   | AC21 | TXADDR4   | I/O | AE4  | RXADDR0   | I/O |
| AA3  | LDATA7    | I/O | AC22 | TXADDR0   | I/O | AE5  | RXD15     | I   |
| AA4  | LDATA6    | I/O | AC23 | TCLK      | I   | AE6  | RXD11     | I   |
| AA23 | GND       | —   | AC24 | TDO       | O   | AE7  | RXD7      | I   |
| AA24 | VDD       | —   | AC25 | TDI       | I   | AE8  | RXD3      | I   |
| AA25 | HINT_     | OD  | AC26 | VDD       | —   | AE9  | VDD       | —   |
| AA26 | VDD       | —   | AD1  | GND       | —   | AE10 | GND       | —   |
| AB1  | LDATA5    | I/O | AD2  | VDD       | —   | AE11 | RXCLAV    | I/O |
| AB2  | LDATA4    | I/O | AD3  | spare     | —   | AE12 | FRCFG0    | I   |
| AB3  | VDD       | —   | AD4  | RXADDR1   | I/O | AE13 | VDD       | —   |
| AB4  | LDATA3    | I/O | AD5  | RXPARG    | I   | AE14 | VDD       | —   |
| AB23 | SCL       | O   | AD6  | RXD12     | I   | AE15 | spare     | —   |

Table 16-13. Pin Description (Numeric List) (4 of 4)

| PIN  | PIN LABEL | I/O |
|------|-----------|-----|
| AE16 | TXPAR     | 0   |
| AE17 | TXD13     | 0   |
| AE18 | TXD10     | 0   |
| AE19 | TXD6      | 0   |
| AE20 | TXD4      | 0   |
| AE21 | TXD0      | 0   |
| AE22 | TXADDR2   | I/O |
| AE23 | VDD       | —   |
| AE24 | spare     | —   |
| AE25 | VDD       | —   |
| AE26 | VDD       | —   |
| AF1  | GND       | —   |
| AF2  | RXADDR4   | I/O |

| PIN  | PIN LABEL | I/O |
|------|-----------|-----|
| AF3  | RXADDR2   | I/O |
| AF4  | GND       | —   |
| AF5  | RXD14     | I   |
| AF6  | RXD10     | I   |
| AF7  | RXD6      | I   |
| AF8  | RXD2      | I   |
| AF9  | RXD0      | I   |
| AF10 | spare     | —   |
| AF11 | spare     | —   |
| AF12 | FRCFG1    | I   |
| AF13 | spare     | —   |
| AF14 | GND       | —   |
| AF15 | TXSOC     | I/O |

| PIN  | PIN LABEL | I/O |
|------|-----------|-----|
| AF16 | TXCLAV_   | I/O |
| AF17 | TXD14     | 0   |
| AF18 | GND       | —   |
| AF19 | TXD7      | 0   |
| AF20 | TXD5      | 0   |
| AF21 | TXD1      | 0   |
| AF22 | TXADDR3   | I/O |
| AF23 | GND       | —   |
| AF24 | VDD       | —   |
| AF25 | spare     | —   |
| AF26 | GND       | —   |

Table 16-14. Pin Description (Alphabetic List) (1 of 4)

| PIN LABEL | PIN  | I/O | PIN LABEL | PIN  | I/O | PIN LABEL | PIN | I/O |
|-----------|------|-----|-----------|------|-----|-----------|-----|-----|
| CLK2X     | N4   | I   | GND       | T1   | —   | HAD15     | J26 | I/O |
| CLKD3     | L2   | O   | GND       | U25  | —   | HAD16     | P26 | I/O |
| EPPWR     | D22  | O   | GND       | V2   | —   | HAD17     | P25 | I/O |
| FRCFG0    | AE12 | I   | GND       | W23  | —   | HAD18     | P24 | I/O |
| FRCFG1    | AF12 | I   | GND       | W4   | —   | HAD19     | R26 | I/O |
| RXCLK     | AC13 | I   | GND       | Y24  | —   | HAD20     | R25 | I/O |
| GND       | A1   | —   | GND       | AA2  | —   | HAD21     | R24 | I/O |
| GND       | A11  | —   | GND       | AA23 | —   | HAD22     | R23 | I/O |
| GND       | A18  | —   | GND       | AC4  | —   | HAD23     | T25 | I/O |
| GND       | A23  | —   | GND       | AD1  | —   | HAD24     | U24 | I/O |
| GND       | A26  | —   | GND       | AD13 | —   | HAD25     | U23 | I/O |
| GND       | B3   | —   | GND       | AD19 | —   | HAD26     | V26 | I/O |
| GND       | B14  | —   | GND       | AD9  | —   | HAD27     | V25 | I/O |
| GND       | B24  | —   | GND       | AE10 | —   | HAD28     | V24 | I/O |
| GND       | C1   | —   | GND       | AF1  | —   | HAD29     | W26 | I/O |
| GND       | C6   | —   | GND       | AF4  | —   | HAD30     | W25 | I/O |
| GND       | C8   | —   | GND       | AF14 | —   | HAD31     | W24 | I/O |
| GND       | C20  | —   | GND       | AF18 | —   | HCBE0_    | F26 | I/O |
| GND       | C26  | —   | GND       | AF23 | —   | HCBE1_    | K23 | I/O |
| GND       | D10  | —   | GND       | AF26 | —   | HCBE2_    | N26 | I/O |
| GND       | D13  | —   | HAD0      | D24  | I/O | HCBE3_    | T23 | I/O |
| GND       | D16  | —   | HAD1      | D25  | I/O | HCLK      | N23 | I   |
| GND       | E2   | —   | HAD2      | D26  | I/O | HDEVSEL_  | L26 | I/O |
| GND       | E25  | —   | HAD3      | E23  | I/O | HENUM_    | B23 | OD  |
| GND       | G1   | —   | HAD4      | E26  | I/O | HFIFORD0  | C25 | I   |
| GND       | G24  | —   | HAD5      | F23  | I/O | HFIFORD1  | B26 | I   |
| GND       | H26  | —   | HAD6      | F24  | I/O | HFIFORD2  | C24 | I   |
| GND       | K2   | —   | HAD7      | F25  | I/O | HFIFORD3  | D23 | I   |
| GND       | K26  | —   | HAD8      | G25  | I/O | HFIFORD4  | A25 | I   |
| GND       | M1   | —   | HAD9      | G26  | I/O | HFIFORD5  | C23 | I   |
| GND       | M3   | —   | HAD10     | H23  | I/O | HFIFOWR0  | C2  | I   |
| GND       | M26  | —   | HAD11     | H24  | I/O | HFIFOWR1  | D3  | I   |
| GND       | N2   | —   | HAD12     | H25  | I/O | HFIFOWR2  | D2  | I   |
| GND       | N25  | —   | HAD13     | J24  | I/O | HFIFOWR3  | D1  | I   |
| GND       | P23  | —   | HAD14     | J25  | I/O | HFIFOWR4  | E3  | I   |

Table 16-14. Pin Description (Alphabetic List) (2 of 4)

| PIN LABEL | PIN  | I/O | PIN LABEL | PIN | I/O | PIN LABEL | PIN  | I/O |
|-----------|------|-----|-----------|-----|-----|-----------|------|-----|
| HFIFOWR5  | F4   | I   | LDATA1    | AC2 | I/O | MOE_      | H1   | O   |
| HFRAME_   | M25  | I/O | LDATA2    | AC1 | I/O | MWE0_     | H2   | O   |
| HGNT_     | Y25  | I   | LDATA3    | AB4 | I/O | MWE1_     | H3   | O   |
| HIDSEL    | T24  | I   | LDATA4    | AB2 | I/O | MWE2_     | G2   | O   |
| HINT_     | AA25 | OD  | LDATA5    | AB1 | I/O | MWE3_     | G3   | O   |
| HIRDY_    | M24  | I/O | LDATA6    | AA4 | I/O | MWR_      | J4   | O   |
| HLED_     | L4   | OD  | LDATA7    | AA3 | I/O | PADDR0    | C10  | I   |
| HPAR      | K24  | I/O | LDATA8    | AA1 | I/O | PADDR1    | B10  | I   |
| HPERR_    | L24  | I/O | LDATA9    | Y4  | I/O | PAS_      | D14  | I/O |
| HREQ_     | Y26  | O   | LDATA10   | Y3  | I/O | PBE0_     | C11  | I   |
| HRST_     | Y23  | I   | LDATA11   | Y2  | I/O | PBE1_     | D12  | I   |
| HSERR_    | L23  | OD  | LDATA12   | Y1  | I/O | PBE2_     | C12  | I   |
| HSTOP_    | L25  | I/O | LDATA13   | W3  | I/O | PBE3_     | B12  | I   |
| HSWITCH_  | F3   | I   | LDATA14   | W2  | I/O | PBLAST_   | C14  | I/O |
| HTRDY_    | M23  | I/O | LDATA15   | W1  | I/O | PBSEL0    | A10  | I   |
| LADDR0    | A3   | I/O | LDATA16   | V4  | I/O | PBSEL1    | D11  | I   |
| LADDR1    | C4   | I/O | LDATA17   | V3  | I/O | PCI5V     | AB26 | I   |
| LADDR2    | B4   | I/O | LDATA18   | V1  | I/O | PCS_      | A15  | I/O |
| LADDR3    | A4   | I/O | LDATA19   | U4  | I/O | PDAEN_    | B15  | I/O |
| LADDR4    | D5   | I/O | LDATA20   | U3  | I/O | PFAIL_    | C15  | I   |
| LADDR5    | C5   | I/O | LDATA21   | U1  | I/O | PINT_     | D15  | OD  |
| LADDR6    | B5   | I/O | LDATA22   | T4  | I/O | PRDY_     | B13  | O   |
| LADDR7    | A5   | I/O | LDATA23   | T3  | I/O | PROCMODE  | B16  | I   |
| LADDR8    | B6   | I/O | LDATA24   | R4  | I/O | PRST_     | A16  | O   |
| LADDR9    | A6   | I/O | LDATA25   | R3  | I/O | PWAIT_    | A13  | I   |
| LADDR10   | D7   | I/O | LDATA26   | R2  | I/O | PWNR      | C13  | I/O |
| LADDR11   | C7   | I/O | LDATA27   | R1  | I/O | RAMMODE   | G4   | I   |
| LADDR12   | B7   | I/O | LDATA28   | P3  | I/O | RXADDR0   | AE4  | I/O |
| LADDR13   | A7   | I/O | LDATA29   | P2  | I/O | RXADDR1   | AD4  | I/O |
| LADDR14   | B8   | I/O | LDATA30   | P1  | I/O | RXADDR2   | AF3  | I/O |
| LADDR15   | A8   | I/O | LDATA31   | N1  | I/O | RXADDR3   | AE3  | I/O |
| LADDR16   | D9   | I/O | MCS0_     | K3  | O   | RXADDR4   | AF2  | I/O |
| LADDR17   | C9   | I/O | MCS1_     | K4  | O   | RXD0      | AF9  | I   |
| LADDR18   | B9   | I/O | MCS2_     | J1  | O   | RXD1      | AC9  | I   |
| LDATA0    | AC3  | I/O | MCS3_     | J2  | O   | RXD2      | AF8  | I   |

Table 16-14. Pin Description (Alphabetic List) (3 of 4)

| PIN LABEL | PIN  | I/O | PIN LABEL | PIN  | I/O | PIN LABEL | PIN  | I/O |
|-----------|------|-----|-----------|------|-----|-----------|------|-----|
| RXD3      | AE8  | I   | spare     | C21  | —   | TXCLK     | AD12 | I   |
| RXD4      | AD8  | I   | spare     | C22  | —   | TXD0      | AE21 | O   |
| RXD5      | AC8  | I   | spare     | D4   | —   | TXD1      | AF21 | O   |
| RXD6      | AF7  | I   | spare     | D17  | —   | TXD2      | AC20 | O   |
| RXD7      | AE7  | I   | spare     | D18  | —   | TXD3      | AD20 | O   |
| RXD8      | AD7  | I   | spare     | D19  | —   | TXD4      | AE20 | O   |
| RXD9      | AC7  | I   | spare     | D20  | —   | TXD5      | AF20 | O   |
| RXD10     | AF6  | I   | spare     | M4   | —   | TXD6      | AE19 | O   |
| RXD11     | AE6  | I   | spare     | AB25 | —   | TXD7      | AF19 | O   |
| RXD12     | AD6  | I   | spare     | AC11 | —   | TXD8      | AC18 | O   |
| RXD13     | AC6  | I   | spare     | AC15 | —   | TXD9      | AD18 | O   |
| RXD14     | AF5  | I   | spare     | AD11 | —   | TXD10     | AE18 | O   |
| RXD15     | AE5  | I   | spare     | AD15 | —   | TXD11     | AC17 | O   |
| RXEN_     | AD10 | I/O | spare     | AD23 | —   | TXD12     | AD17 | O   |
| RXCLAV    | AE11 | I/O | spare     | AD24 | —   | TXD13     | AE17 | O   |
| RXSOC     | AC10 | I   | spare     | AD3  | —   | TXD14     | AF17 | O   |
| RXPAR     | AD5  | I   | spare     | AE15 | —   | TXD15     | AC16 | O   |
| SCHREF    | AE1  | I   | spare     | AE24 | —   | TXEN_     | AC14 | I   |
| SCL       | AB23 | O   | spare     | AF10 | —   | TXCLAV_   | AF16 | I/O |
| SDA       | AB24 | I/O | spare     | AF11 | —   | TXSOC     | AF15 | I/O |
| spare     | A17  | —   | spare     | AF13 | —   | TXPAR     | AE16 | O   |
| spare     | A19  | —   | spare     | AF25 | —   | UTOPIA1   | AD14 | I   |
| spare     | A20  | —   | STAT0     | F1   | O   | VDD       | A2   | —   |
| spare     | A21  | —   | STAT1     | F2   | O   | VDD       | A9   | —   |
| spare     | A22  | —   | SYSCLK    | M2   | O   | VDD       | A12  | —   |
| spare     | B1   | —   | TCLK      | AC23 | I   | VDD       | A14  | —   |
| spare     | B17  | —   | TDI       | AC25 | I   | VDD       | A24  | —   |
| spare     | B18  | —   | TDO       | AC24 | O   | VDD       | B2   | —   |
| spare     | B20  | —   | TMS       | AD26 | I   | VDD       | B11  | —   |
| spare     | B21  | —   | TRST_     | AD25 | I   | VDD       | B19  | —   |
| spare     | B22  | —   | TXADDR0   | AC22 | I/O | VDD       | B25  | —   |
| spare     | C3   | —   | TXADDR1   | AD22 | I/O | VDD       | C16  | —   |
| spare     | C17  | —   | TXADDR2   | AE22 | I/O | VDD       | D21  | —   |
| spare     | C18  | —   | TXADDR3   | AF22 | I/O | VDD       | D6   | —   |
| spare     | C19  | —   | TXADDR4   | AC21 | I/O | VDD       | D8   | —   |

Table 16-14. Pin Description (Alphabetic List) (4 of 4)

| PIN LABEL | PIN | I/O | PIN LABEL | PIN  | I/O | PIN LABEL | PIN  | I/O |
|-----------|-----|-----|-----------|------|-----|-----------|------|-----|
| VDD       | E1  | —   | VDD       | T26  | —   | VDD       | AE2  | —   |
| VDD       | E24 | —   | VDD       | U2   | —   | VDD       | AE9  | —   |
| VDD       | G23 | —   | VDD       | U26  | —   | VDD       | AE13 | —   |
| VDD       | H4  | —   | VDD       | V23  | —   | VDD       | AE14 | —   |
| VDD       | J3  | —   | VDD       | AA24 | —   | VDD       | AE23 | —   |
| VDD       | J23 | —   | VDD       | AA26 | —   | VDD       | AE25 | —   |
| VDD       | K1  | —   | VDD       | AB3  | —   | VDD       | AE26 | —   |
| VDD       | K25 | —   | VDD       | AC5  | —   | VDD       | AF24 | —   |
| VDD       | L1  | —   | VDD       | AC12 | —   | VGG       | E4   | I   |
| VDD       | L3  | —   | VDD       | AC19 | —   |           |      |     |
| VDD       | N24 | —   | VDD       | AC26 | —   |           |      |     |
| VDD       | N3  | —   | VDD       | AD2  | —   |           |      |     |
| VDD       | P4  | —   | VDD       | AD16 | —   |           |      |     |
| VDD       | T2  | —   | VDD       | AD21 | —   |           |      |     |

The pins listed in [Table 16-15](#) have been selected as future inputs. When designing the printed circuit board, tie these pins to ground for backward compatibility of future parts.

Table 16-15. Spare Pins Reserved for Inputs

| Pin  | Comments   |
|------|--|
| AF11 | Tied to ground on PCB for forward compatibility. |
| AF13 | Tied to ground on PCB for forward compatibility. |
| AD23 | Tied to ground on PCB for forward compatibility. |
| AE24 | Tied to ground on PCB for forward compatibility. |



# Appendix A: Boundary Scan

---

The CN8236 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 CN8236. It is assumed that the reader is familiar with boundary scan terminology.

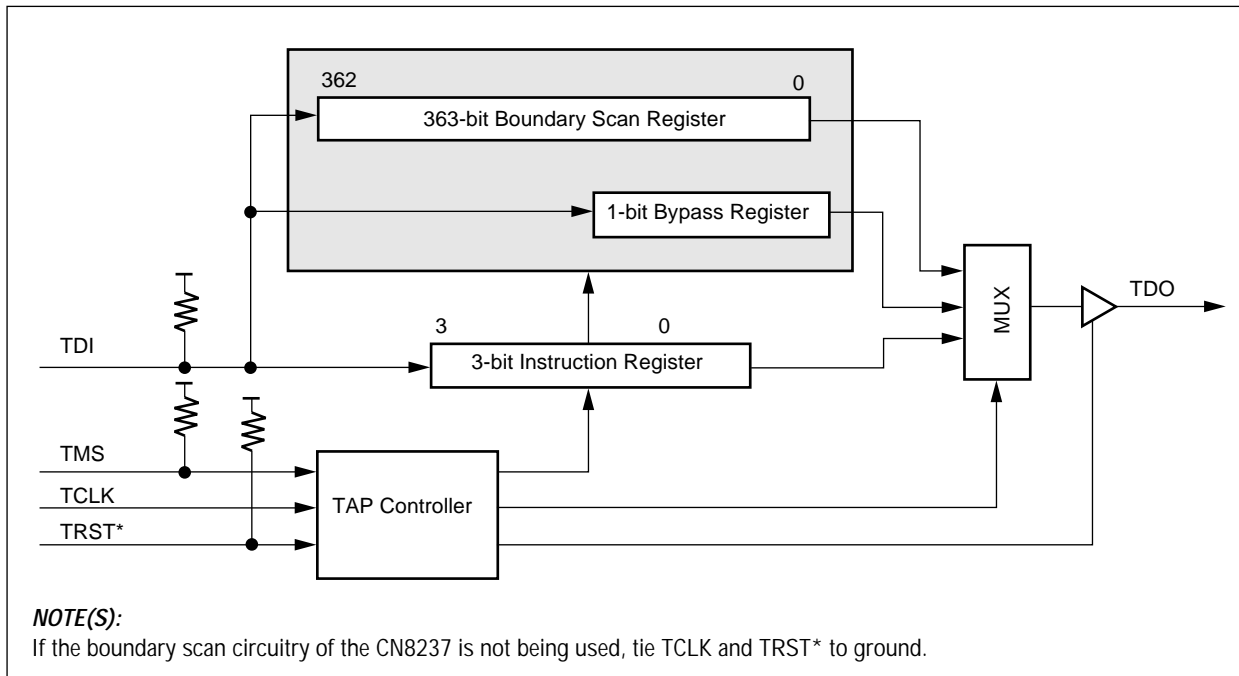
The Boundary Scan section of the CN8236 provides access to all external I/O signals of the device for board and system-level testing. This circuitry also conforms to IEEE Std 1149.1-1990. The boundary scan test logic is accessed through five dedicated pins on the CN8236 (see [Table A-1](#)).

**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 (IR), and the Tap Access Port (TAP) controller (see [Figure A-1](#)).

Figure A-1. Test Circuitry Block Diagram



8236\_092

## A.1 Instruction Register

The Instruction register 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 sixteen instructions include three IEEE 1149.1 mandatory public instructions (BYPASS, EXTEST, and SAMPLE/PRELOAD) and thirteen private instructions for manufacturing use only. Bit-0 (LSB) is shifted into the instruction register first. [Table A-2](#) shows the bit settings for this register.

**NOTE:** The implementation of this register as described is applicable only to Rev. A of CN8236 device.

**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     | T38VBFIFO - 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 CN8236.

## ***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-tri-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 1.

Table A-3. Boundary Scan Register Cells (1 of 8)

| Cell | Related Pin Name | Cell Type | Controlling Cell |
|------|------------------|-----------|------------------|
| 0    | STAT[1]          | output3   | 30               |
| 1    | LADDR[0]         | output3   | 30               |
| 2    | LADDR[1]         | output3   | 30               |
| 3    | LADDR[2]         | bidir     | 19               |
| 4    | LADDR[3]         | bidir     | 19               |
| 5    | LADDR[4]         | bidir     | 19               |
| 6    | LADDR[5]         | bidir     | 19               |
| 7    | LADDR[6]         | bidir     | 19               |
| 8    | LADDR[7]         | bidir     | 19               |
| 9    | LADDR[8]         | bidir     | 19               |
| 10   | LADDR[9]         | bidir     | 19               |
| 11   | LADDR[10]        | bidir     | 19               |
| 12   | LADDR[11]        | bidir     | 19               |
| 13   | LADDR[12]        | bidir     | 19               |
| 14   | LADDR[13]        | bidir     | 19               |
| 15   | LADDR[14]        | bidir     | 19               |
| 16   | LADDR[15]        | bidir     | 19               |
| 17   | LADDR[16]        | bidir     | 19               |
| 18   | LADDR[17]        | bidir     | 19               |
| 19   | —                | control   | —                |

Table A-3. Boundary Scan Register Cells (2 of 8)

| Cell | Related Pin Name | Cell Type | Controlling Cell |
|------|------------------|-----------|------------------|
| 20   | LADDR[18]        | bidir     | 19               |
| 21   | PADDR[0]         | input     | —                |
| 22   | PADDR[1]         | input     | —                |
| 23   | PBSEL[0]         | input     | —                |
| 24   | PBSEL[1]         | input     | —                |
| 25   | PBE[0]_NEG       | input     | —                |
| 26   | PBE[1]_NEG       | input     | —                |
| 27   | PBE[2]_NEG       | input     | —                |
| 28   | PBE[3]_NEG       | input     | —                |
| 29   | PWNR             | bidir     | 35               |
| 30   | —                | control   | —                |
| 31   | PRDY_NEG         | output3   | 30               |
| 32   | PWAIT_NEG        | input     | —                |
| 33   | PBLAST_NEG       | bidir     | 35               |
| 34   | PAS_NEG          | bidir     | 35               |
| 35   | —                | control   | —                |
| 36   | PCS_NEG          | bidir     | 35               |
| 37   | —                | control   | —                |
| 38   | PDAEN_NEG        | bidir     | 37               |
| 39   | PFAIL_NEG        | input     | —                |
| 40   | —                | control   | —                |
| 41   | PINT_NEG         | output3   | 40               |
| 42   | PRST_NEG         | output3   | 30               |
| 43   | PROCMODE         | input     | —                |
| 44   | HAD[0]           | bidir     | 60               |
| 45   | HAD[1]           | bidir     | 60               |
| 46   | HAD[2]           | bidir     | 60               |
| 47   | HAD[3]           | bidir     | 60               |
| 48   | HAD[4]           | bidir     | 60               |
| 49   | HAD[5]           | bidir     | 60               |
| 50   | EEPWR            | output3   | 30               |
| 51   | —                | control   | —                |
| 52   | HENUM_NEG        | output3   | 51               |
| 53   | HFIFORD[5]       | input     | —                |

Table A-3. Boundary Scan Register Cells (3 of 8)

| Cell | Related Pin Name | Cell Type | Controlling Cell |
|------|------------------|-----------|------------------|
| 54   | HFIFORD[4]       | input     | —                |
| 55   | HFIFORD[3]       | input     | —                |
| 56   | HFIFORD[2]       | input     | —                |
| 57   | HFIFORD[1]       | input     | —                |
| 58   | HFIFORD[0]       | input     | —                |
| 59   | HAD[6]           | bidir     | 60               |
| 60   | —                | control   | —                |
| 61   | HAD[7]           | bidir     | 60               |
| 62   | HCBE[0]_NEG      | bidir     | 102              |
| 63   | HAD[8]           | bidir     | 70               |
| 64   | HAD[9]           | bidir     | 70               |
| 65   | HAD[10]          | bidir     | 70               |
| 66   | HAD[11]          | bidir     | 70               |
| 67   | HAD[12]          | bidir     | 70               |
| 68   | HAD[13]          | bidir     | 70               |
| 69   | HAD[14]          | bidir     | 70               |
| 70   | —                | control   | —                |
| 71   | HAD[15]          | bidir     | 70               |
| 72   | HCBE[1]_NEG      | bidir     | 102              |
| 73   | —                | control   | —                |
| 74   | HPAR             | bidir     | 73               |
| 75   | —                | control   | —                |
| 76   | HSERR_NEG        | bidir     | 75               |
| 77   | —                | control   | —                |
| 78   | HPERR_NEG        | bidir     | 77               |
| 79   | —                | control   | —                |
| 80   | HSTOP_NEG        | bidir     | 79               |
| 81   | —                | control   | —                |
| 82   | HDEVSEL_NEG      | bidir     | 81               |
| 83   | —                | control   | —                |
| 84   | HTRDY_NEG        | bidir     | 83               |
| 85   | —                | control   | —                |
| 86   | HIRDY_NEG        | bidir     | 85               |
| 87   | —                | control   | —                |

Table A-3. Boundary Scan Register Cells (4 of 8)

| Cell | Related Pin Name | Cell Type | Controlling Cell |
|------|------------------|-----------|------------------|
| 88   | HFRAME_NEG       | bidir     | 87               |
| 89   | HCLK             | input     | —                |
| 90   | —                | internal  | —                |
| 91   | HCBE[2]_NEG      | bidir     | 102              |
| 92   | HAD[16]          | bidir     | 99               |
| 93   | HAD[17]          | bidir     | 99               |
| 94   | HAD[18]          | bidir     | 99               |
| 95   | HAD[19]          | bidir     | 99               |
| 96   | HAD[20]          | bidir     | 99               |
| 97   | HAD[21]          | bidir     | 99               |
| 98   | HAD[22]          | bidir     | 99               |
| 99   | —                | control   | —                |
| 100  | HAD[23]          | bidir     | 99               |
| 101  | HIDSEL           | input     | —                |
| 102  | —                | control   | —                |
| 103  | HCBE[3]_NEG      | bidir     | 102              |
| 104  | HAD[24]          | bidir     | 111              |
| 105  | HAD[25]          | bidir     | 111              |
| 106  | HAD[26]          | bidir     | 111              |
| 107  | HAD[27]          | bidir     | 111              |
| 108  | HAD[28]          | bidir     | 111              |
| 109  | HAD[29]          | bidir     | 111              |
| 110  | HAD30            | bidir     | 111              |
| 111  | —                | control   | —                |
| 112  | HAD[31]          | bidir     | 111              |
| 113  | —                | control   | —                |
| 114  | HREQ_NEG         | bidir     | 113              |
| 115  | HGNT_NEG         | input     | —                |
| 116  | HRST_NEG         | input     | —                |
| 117  | TXDATA[8]        | output3   | 145              |
| 118  | TXDATA[9]        | output3   | 145              |
| 119  | TXDATA]10[       | output3   | 145              |
| 120  | TXDATA[11]       | output3   | 145              |
| 121  | TXDATA[12]       | output3   | 145              |



Table A-3. Boundary Scan Register Cells (5 of 8)

| Cell | Related Pin Name | Cell Type | Controlling Cell |
|------|------------------|-----------|------------------|
| 122  | TXDATA[13]       | output3   | 145              |
| 123  | TXDATA[14]       | output3   | 145              |
| 124  | TXDATA[15]       | output3   | 145              |
| 125  | —                | control   | —                |
| 126  | HINT_NEG         | bidir     | 125              |
| 127  | PCI5V            | input     | —                |
| 128  | —                | control   | —                |
| 129  | SDA              | bidir     | 128              |
| 130  | —                | control   | —                |
| 131  | SCL              | bidir     | 130              |
| 132  | TXADDR[0]        | bidir     | 136              |
| 133  | TXADDR[1]        | bidir     | 136              |
| 134  | TXADDR[2]        | bidir     | 136              |
| 135  | TXADDR[3]        | bidir     | 136              |
| 136  | —                | control   | —                |
| 137  | TXADDR[4]        | bidir     | 136              |
| 138  | TXDATA[0]        | output3   | 145              |
| 139  | TXDATA[1]        | output3   | 145              |
| 140  | TXDATA[2]        | output3   | 145              |
| 141  | TXDATA[3]        | output3   | 145              |
| 142  | TXDATA[4]        | output3   | 145              |
| 143  | TXDATA[5]        | output3   | 145              |
| 144  | TXDATA[6]        | output3   | 145              |
| 145  | —                | control   | —                |
| 146  | TXDATA[7]        | output3   | 145              |
| 147  | TXPAR            | output3   | 145              |
| 148  | —                | control   | —                |
| 149  | TXCLAV_NEG       | bidir     | 148              |
| 150  | —                | control   | —                |
| 151  | TXSOC            | output3   | 150              |
| 152  | —                | control   | —                |
| 153  | TXEN_NEG         | bidir     | 152              |
| 154  | UTOPIA[1]        | input     | —                |
| 155  | FRCTRL           | input     | —                |

Table A-3. Boundary Scan Register Cells (6 of 8)

| Cell | Related Pin Name | Cell Type | Controlling Cell |
|------|------------------|-----------|------------------|
| 156  | FRCFG[1]         | input     | —                |
| 157  | FRCFG[0]         | input     | —                |
| 158  | TXCLK            | input     | —                |
| 159  | —                | control   | —                |
| 160  | RXCLAV_NEG       | bidir     | 159              |
| 161  | —                | control   | —                |
| 162  | RXEN_NEG         | bidir     | 161              |
| 163  | RXSOC            | input     | —                |
| 164  | RXDATA[0]        | input     | —                |
| 165  | RXDATA[1]        | input     | —                |
| 166  | RXDATA[2]        | input     | —                |
| 167  | RXDATA[3]        | input     | —                |
| 168  | RXDATA[4]        | input     | —                |
| 169  | RXDATA[5]        | input     | —                |
| 170  | RXDATA[6]        | input     | —                |
| 171  | RXDATA[7]        | input     | —                |
| 172  | RXPAR            | input     | —                |
| 173  | RXADDR[0]        | bidir     | 177              |
| 174  | RXADDR[1]        | bidir     | 177              |
| 175  | RXADDR[2]        | bidir     | 177              |
| 176  | RXADDR[3]        | bidir     | 177              |
| 177  | —                | control   | —                |
| 178  | RXADDR[4]        | bidir     | 177              |
| 179  | SCHREF           | input     | —                |
| 180  | LDATA[0]         | bidir     | 195              |
| 181  | LDATA[1]         | bidir     | 195              |
| 182  | LDATA[2]         | bidir     | 195              |
| 183  | LDATA[3]         | bidir     | 195              |
| 184  | RXDATA[8]        | input     | —                |
| 185  | RXDATA[9]        | input     | —                |
| 186  | RXDATA[10]       | input     | —                |
| 187  | RXDATA[11]       | input     | —                |
| 188  | RXDATA[12]       | input     | —                |
| 189  | RXDATA[13]       | input     | —                |

Table A-3. Boundary Scan Register Cells (7 of 8)

| Cell | Related Pin Name | Cell Type | Controlling Cell |
|------|------------------|-----------|------------------|
| 190  | RXDATA[14]       | input     | —                |
| 191  | RXDATA[15]       | input     | —                |
| 192  | LDATA[4]         | bidir     | 195              |
| 193  | LDATA[5]         | bidir     | 195              |
| 194  | LDATA[6]         | bidir     | 195              |
| 195  | —                | control   | —                |
| 196  | LDATA[7]         | bidir     | 195              |
| 197  | LDATA[8]         | bidir     | 204              |
| 198  | LDATA[9]         | bidir     | 204              |
| 199  | LDATA[10]        | bidir     | 204              |
| 200  | LDATA[11]        | bidir     | 204              |
| 201  | LDATA[12]        | bidir     | 204              |
| 202  | LDATA[13]        | bidir     | 204              |
| 203  | LDATA[14]        | bidir     | 204              |
| 204  | —                | control   | —                |
| 205  | LDATA[15]        | bidir     | 204              |
| 206  | LDATA[16]        | bidir     | 213              |
| 207  | LDATA[17]        | bidir     | 213              |
| 208  | LDATA[18]        | bidir     | 213              |
| 209  | LDATA[19]        | bidir     | 213              |
| 210  | LDATA[20]        | bidir     | 213              |
| 211  | LDATA[21]        | bidir     | 213              |
| 212  | LDATA[22]        | bidir     | 213              |
| 213  | —                | control   | —                |
| 214  | LDATA[23]        | bidir     | 213              |
| 215  | LDATA[24]        | bidir     | 222              |
| 216  | LDATA[25]        | bidir     | 222              |
| 217  | LDATA[26]        | bidir     | 222              |
| 218  | LDATA[27]        | bidir     | 222              |
| 219  | LDATA[28]        | bidir     | 222              |
| 220  | LDATA[29]        | bidir     | 222              |
| 221  | LDATA[30]        | bidir     | 222              |
| 222  | —                | control   | —                |
| 223  | LDATA[31]        | bidir     | 222              |

Table A-3. Boundary Scan Register Cells (8 of 8)

| Cell | Related Pin Name | Cell Type | Controlling Cell |
|------|------------------|-----------|------------------|
| 224  | CLK2X            | input     | —                |
| 225  | SYCLK            | output3   | 30               |
| 226  | —                | internal  | —                |
| 227  | CLKD3            | output3   | 30               |
| 228  | —                | control   | —                |
| 229  | HLED_NEG         | output3   | 228              |
| 230  | MCS[0]_NEG       | output3   | 30               |
| 231  | MCS[1]_NEG       | output3   | 30               |
| 232  | MCS[2]_NEG       | output3   | 30               |
| 233  | MCS[3]_NEG       | output3   | 30               |
| 234  | MWR_NEG          | output3   | 30               |
| 235  | MOE_NEG          | output3   | 30               |
| 236  | MWE[0]_NEG       | output3   | 30               |
| 237  | MWE[1]_NEG       | output3   | 30               |
| 238  | MWE[2]_NEG       | output3   | 30               |
| 239  | MWE[3]_NEG       | output3   | 30               |
| 240  | RAMMODE          | input     | —                |
| 241  | STAT[0]          | output3   | 30               |
| 242  | HSWITCH_NEG      | input     | —                |
| 243  | HFIFOWR[5]       | input     | —                |
| 244  | HFIFOWR[4]       | input     | —                |
| 245  | HFIFOWR[3]       | input     | —                |
| 246  | HFIFOWR[2]       | input     | —                |
| 247  | HFIFOWR[1]       | input     | —                |
| 248  | HFIFOWR[0]       | input     | —                |

## A.5 Electrical Characteristics

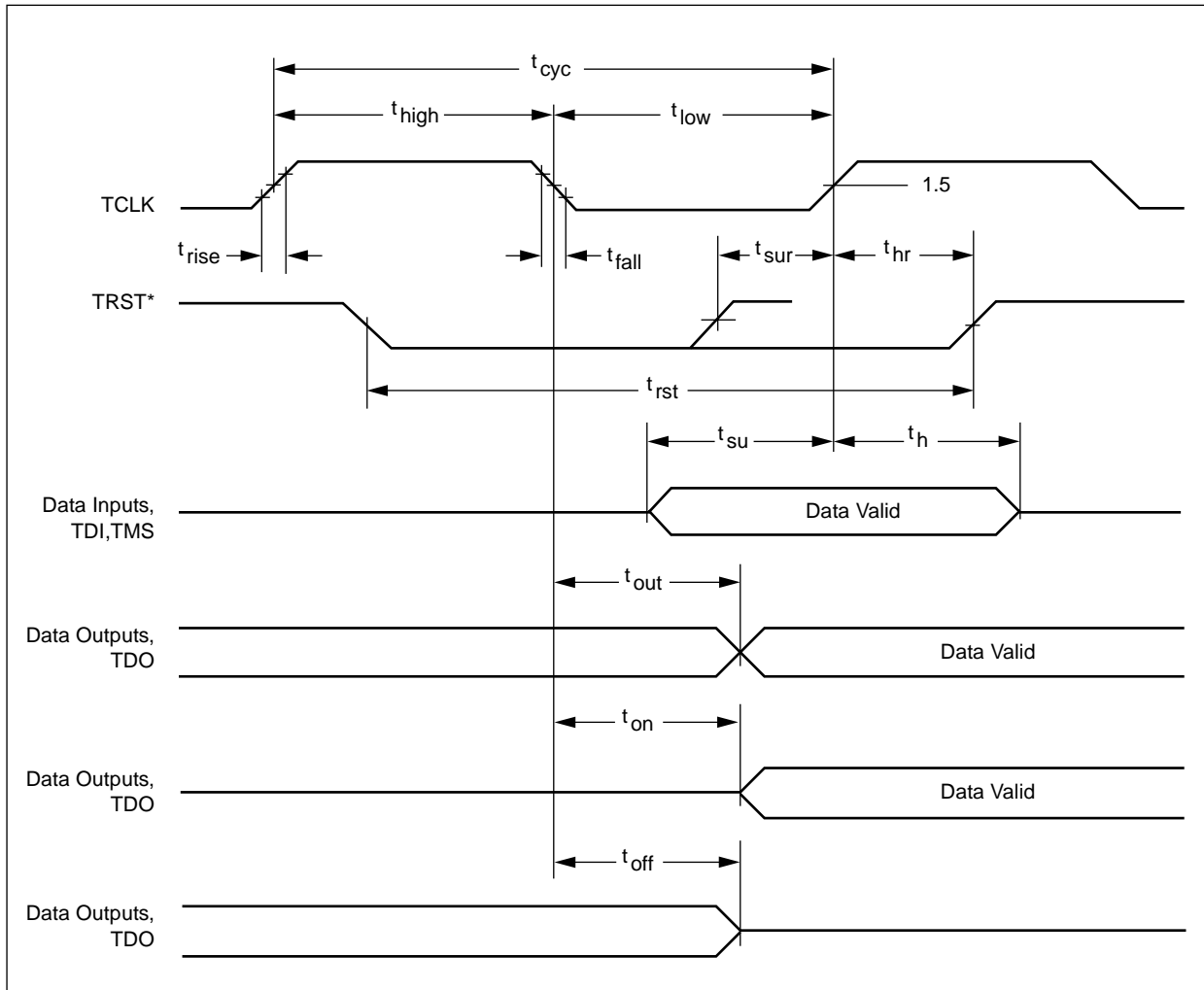
This section describes the electrical characteristics for boundary scan. [Table A-4](#) provides timing specifications and [Figure A-2](#) shows a timing diagram.

**Table A-4. Timing Specifications**

| Symbol     | Description  | Min | Max | Unit      |
|------------|--|-----|-----|-----------|
| —          | TCLK Frequency                                       | —   | 20  | MHz       |
| $t_{cyc}$  | TCLK Cycle   | 50  | —   | ns        |
| $t_{high}$ | TCLK High Pulse Width                                | 20  | 30  | ns        |
| $t_{low}$  | TCLK Low Pulse Width                                 | 20  | 30  | ns        |
| $t_{rise}$ | TCLK Rise  | 1   | 4   | ns        |
| $t_{fall}$ | TCLK Fall  | 1   | 4   | ns        |
| $t_{sur}$  | TRST* Inactive to TCLK Rising Edge <sup>(1)</sup>    | 5   | —   | ns        |
| $t_{hr}$   | TRST* Inactive after TCLK Rising Edge <sup>(1)</sup> | 5   | —   | ns        |
| $t_{rst}$  | TRST* Active   | 1   | —   | $t_{cyc}$ |
| $t_{su}$   | TDI, TMS, and Data Inputs Setup                      | 5   | —   | ns        |
| $t_h$      | TDI, TMS, and Data Inputs Hold                       | 5   | —   | ns        |
| $t_{out}$  | TDO and Data Outputs Valid                           | —   | 10  | ns        |
| $t_{on}$   | TDO and Data Outputs Float to Valid                  | —   | 10  | ns        |
| $t_{off}$  | TDO and Data Outputs Valid to Float                  | —   | 10  | ns        |

**NOTE(S):**  
<sup>(1)</sup> TRST\* going inactive timing only required if TMS = 0 at the rising edge of TCLK.

Figure A-2. Timing Diagram



8236\_093

## 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).

```
-----
-- Boundary Scan Description Language (IEEE 1149.1b)
-- BSDL for Device CN8236
-- BSDL File Created by 'topgen' revision 2.1
-- (c) Mindspeed, Inc.
-----
```

```
entity CN8236 is generic (PHYSICAL_PIN_MAP : string := "BGA_352");
```

```
port (
    HFIFOWR0      : in   bit;
    HFIFOWR1      : in   bit;
    HFIFOWR2      : in   bit;
    HFIFOWR3      : in   bit;
    HFIFOWR4      : in   bit;
    HFIFOWR5      : in   bit;
    HSWITCH_NEG   : in   bit;
    STAT0         : out  bit;
    RAMMODE       : in   bit;
    MWE3_NEG      : out  bit;
    MWE2_NEG      : out  bit;
    OUTGND0       : linkage bit;
    OUTVDD0       : linkage bit;
    MWE1_NEG      : out  bit;
    MWE0_NEG      : out  bit;
    MOE_NEG       : out  bit;
    MWR_NEG       : out  bit;
    OUTVDD1       : linkage bit;
    MCS3_NEG      : out  bit;
    MCS2_NEG      : out  bit;
    MCS1_NEG      : out  bit;
    MCS0_NEG      : out  bit;
    OUTGND1       : linkage bit;
    OUTVDD2       : linkage bit;
    HLED_NEG      : out  bit;
    OUTVDD3       : linkage bit;
    CLKD3         : out  bit;
    OUTVDD4       : linkage bit;
    OUTGND2       : linkage bit;
    SYSCLK        : out  bit;
```

```

OUTGND3      : linkage bit;
CLK2X        : in   bit;
OUTVDD5      : linkage bit;
OUTGND4      : linkage bit;
LDATA31      : inout bit;
LDATA30      : inout bit;
LDATA29      : inout bit;
LDATA28      : inout bit;
OUTVDD6      : linkage bit;
LDATA27      : inout bit;
LDATA26      : inout bit;
LDATA25      : inout bit;
LDATA24      : inout bit;
OUTGND5      : linkage bit;
OUTVDD7      : linkage bit;
LDATA23      : inout bit;
LDATA22      : inout bit;
LDATA21      : inout bit;
OUTVDD8      : linkage bit;
LDATA20      : inout bit;
LDATA19      : inout bit;
LDATA18      : inout bit;
OUTGND6      : linkage bit;
LDATA17      : inout bit;
LDATA16      : inout bit;
LDATA15      : inout bit;
LDATA14      : inout bit;
LDATA13      : inout bit;
OUTGND7      : linkage bit;
LDATA12      : inout bit;
LDATA11      : inout bit;
LDATA10      : inout bit;
LDATA9       : inout bit;
LDATA8       : inout bit;
OUTGND8      : linkage bit;
LDATA7       : inout bit;
LDATA6       : inout bit;
LDATA5       : inout bit;
LDATA4       : inout bit;
OUTVDD9      : linkage bit;
RXDATA15     : in   bit;
RXDATA14     : in   bit;
RXDATA13     : in   bit;
RXDATA12     : in   bit;
RXDATA11     : in   bit;
RXDATA10     : in   bit;
RXDATA9      : in   bit;
RXDATA8      : in   bit;

```



```

LDATA3      : inout  bit;
LDATA2      : inout  bit;
LDATA1      : inout  bit;
LDATA0      : inout  bit;
OUTGND9     : linkage bit;
OUTVDD10    : linkage bit;
SCHREF      : in     bit;
CGND0       : linkage bit;
CVDD0       : linkage bit;
OUTGND10    : linkage bit;
RXADDR4     : inout  bit;
RXADDR3     : inout  bit;
RXADDR2     : inout  bit;
RXADDR1     : inout  bit;
RXADDR0     : inout  bit;
OUTGND11    : linkage bit;
OUTVDD11    : linkage bit;
RXPAR       : in     bit;
RXDATA7     : in     bit;
RXDATA6     : in     bit;
RXDATA5     : in     bit;
RXDATA4     : in     bit;
RXDATA3     : in     bit;
RXDATA2     : in     bit;
RXDATA1     : in     bit;
OUTGND12    : linkage bit;
OUTVDD12    : linkage bit;
RXDATA0     : in     bit;
RXSOC       : in     bit;
RXEN_NEG    : inout  bit;
OUTGND13    : linkage bit;
RXCLAV_NEG  : inout  bit;
OUTVDD13    : linkage bit;
TXCLK       : in     bit;
FRCFG0      : in     bit;
FRCFG1      : in     bit;
FRCTRL      : in     bit;
OUTGND14    : linkage bit;
OUTVDD14    : linkage bit;
OUTGND15    : linkage bit;
OUTVDD15    : linkage bit;
UTOPIA1     : in     bit;
TXEN_NEG    : inout  bit;
TXSOC       : out    bit;
TXCLAV_NEG  : inout  bit;
TXPAR       : out    bit;
TXDATA7     : out    bit;
TXDATA6     : out    bit;

```

```

OUTGND16      : linkage bit;
OUTVDD16      : linkage bit;
TXDATA5       : out  bit;
TXDATA4       : out  bit;
TXDATA3       : out  bit;
TXDATA2       : out  bit;
TXDATA1       : out  bit;
TXDATA0       : out  bit;
OUTVDD17      : linkage bit;
TXADDR4       : inout bit;
TXADDR3       : inout bit;
TXADDR2       : inout bit;
TXADDR1       : inout bit;
TXADDR0       : inout bit;
OUTGND17      : linkage bit;
OUTVDD18      : linkage bit;
OUTVDD19      : linkage bit;
CGND1         : linkage bit;
CVDD1         : linkage bit;
TCLK          : in   bit;
OUTVDD20      : linkage bit;
TRST_NEG      : in   bit;
TMS           : in   bit;
TDO           : out  bit;
TDI           : in   bit;
OUTVDD21      : linkage bit;
SCL           : inout bit;
SDA           : inout bit;
PCI5V         : in   bit;
OUTGND18      : linkage bit;
OUTVDD22      : linkage bit;
HINT_NEG      : inout bit;
OUTVDD23      : linkage bit;
TXDATA15      : out  bit;
TXDATA14      : out  bit;
TXDATA13      : out  bit;
TXDATA12      : out  bit;
TXDATA11      : out  bit;
OUTGND19      : linkage bit;
TXDATA10      : out  bit;
TXDATA9       : out  bit;
TXDATA8       : out  bit;
OUTVDD24      : linkage bit;
HRST_NEG      : inout bit;
OUTGND20      : linkage bit;
HGNT_NEG      : inout bit;
HREQ_NEG      : inout bit;
OUTGND21      : linkage bit;

```

```

HAD31      : inout  bit;
HAD30      : inout  bit;
HAD29      : inout  bit;
OUTVDD25   : linkage bit;
HAD28      : inout  bit;
HAD27      : inout  bit;
HAD26      : inout  bit;
HAD25      : inout  bit;
HAD24      : inout  bit;
OUTGND22   : linkage bit;
OUTVDD26   : linkage bit;
HCBE3_NEG  : inout  bit;
HIDSEL     : inout  bit;
HAD23      : inout  bit;
OUTVDD27   : linkage bit;
HAD22      : inout  bit;
HAD21      : inout  bit;
HAD20      : inout  bit;
HAD19      : inout  bit;
OUTGND23   : linkage bit;
HAD18      : inout  bit;
HAD17      : inout  bit;
HAD16      : inout  bit;
HCBE2_NEG  : inout  bit;
OUTGND24   : linkage bit;
OUTVDD28   : linkage bit;
HCLK       : inout  bit;
OUTGND25   : linkage 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  : inout  bit;
OUTGND26   : linkage bit;
OUTVDD29   : linkage bit;
HPAR       : inout  bit;
HCBE1_NEG  : inout  bit;
HAD15      : inout  bit;
HAD14      : inout  bit;
HAD13      : inout  bit;
OUTVDD30   : linkage bit;
OUTGND27   : linkage bit;
HAD12      : inout  bit;
HAD11      : inout  bit;
HAD10      : inout  bit;
HAD9       : inout  bit;

```

```

HAD8           : inout  bit;
OUTGND28      : linkage bit;
OUTVDD31      : linkage bit;
HCBE0_NEG     : inout  bit;
HAD7           : inout  bit;
HAD6           : inout  bit;
HFIFORD0      : in     bit;
HFIFORD1      : in     bit;
HFIFORD2      : in     bit;
HFIFORD3      : in     bit;
HFIFORD4      : in     bit;
HFIFORD5      : in     bit;
HENUM_NEG     : out    bit;
EEPWR         : out    bit;
HAD5           : inout  bit;
HAD4           : inout  bit;
OUTGND29      : linkage bit;
OUTVDD32      : linkage bit;
HAD3           : inout  bit;
HAD2           : inout  bit;
HAD1           : inout  bit;
HAD0           : inout  bit;
OUTGND30      : linkage bit;
OUTGND31      : linkage bit;
OUTVDD33      : linkage bit;
OUTGND32      : linkage bit;
OUTVDD34      : linkage bit;
OUTGND33      : linkage bit;
OUTVDD35      : linkage bit;
OUTGND34      : linkage bit;
OUTVDD36      : linkage bit;
OUTGND35      : linkage bit;
CGND2         : linkage bit;
CVDD2         : linkage bit;
PROCMODE      : in     bit;
PRST_NEG      : out    bit;
PINT_NEG      : out    bit;
PFAIL_NEG     : in     bit;
OUTGND36      : linkage bit;
OUTVDD37      : linkage bit;
PDAEN_NEG     : inout  bit;
PCS_NEG       : inout  bit;
PAS_NEG       : inout  bit;
PBLAST_NEG    : inout  bit;
PWAIT_NEG     : in     bit;
PRDY_NEG      : out    bit;
PWNR          : inout  bit;
OUTGND37      : linkage bit;

```

```

OUTVDD38      : linkage bit;
PBE3_NEG      : in   bit;
PBE2_NEG      : in   bit;
PBE1_NEG      : in   bit;
OUTGND38      : linkage bit;
OUTVDD39      : linkage bit;
PBE0_NEG      : in   bit;
PBSEL1        : in   bit;
PBSEL0        : in   bit;
PADDR1        : in   bit;
PADDR0        : in   bit;
OUTGND39      : linkage bit;
OUTVDD40      : linkage bit;
LADDR18       : inout bit;
LADDR17       : inout bit;
LADDR16       : inout bit;
LADDR15       : inout bit;
LADDR14       : inout bit;
OUTGND40      : linkage bit;
OUTVDD41      : linkage bit;
LADDR13       : inout bit;
LADDR12       : inout bit;
LADDR11       : inout bit;
LADDR10       : inout bit;
LADDR9        : inout bit;
LADDR8        : inout bit;
OUTGND41      : linkage bit;
OUTVDD42      : linkage bit;
LADDR7        : inout bit;
LADDR6        : inout bit;
LADDR5        : inout bit;
LADDR4        : inout bit;
LADDR3        : inout bit;
LADDR2        : inout bit;
OUTGND42      : linkage bit;
OUTVDD43      : linkage bit;
LADDR1        : out   bit;
LADDR0        : out   bit;
OUTGND43      : linkage bit;
OUTVDD44      : linkage bit;
OUTGND44      : linkage bit;
PVGGO        : linkage bit;
CGND3         : linkage bit;
CVDD3         : linkage bit;
STAT1         : out   bit
);

```

```

-- Libraries
-- bcad won't recognize the reference to the work library.
-- use work.STD_1149_1_1994.all;
use STD_1149_1_1994.all;

attribute COMPONENT_CONFORMANCE of CN8236 : entity is
"STD_1149_1_1993";

attribute PIN_MAP of CN8236 : entity is PHYSICAL_PIN_MAP;

constant BGA_352: PIN_MAP_STRING :=

"HFIFOWR0      : C2," &
"HFIFOWR1      : D3," &
"HFIFOWR2      : D2," &
"HFIFOWR3      : D1," &
"HFIFOWR4      : E3," &
"HFIFOWR5      : F4," &
"HSWITCH_NEG: F3," &
"STAT0         : F1," &
"RAMMODE       : G4," &
"MWE3_NEG      : G3," &
"MWE2_NEG      : G2," &
"OUTGND0       : G1," &
"OUTVDD0       : H4," &
"MWE1_NEG      : H3," &
"MWE0_NEG      : H2," &
"MOE_NEG       : H1," &
"MWR_NEG       : J4," &
"OUTVDD1       : J3," &
"MCS3_NEG      : J2," &
"MCS2_NEG      : J1," &
"MCS1_NEG      : K4," &
"MCS0_NEG      : K3," &
"OUTGND1       : K2," &
"OUTVDD2       : K1," &
"HLED_NEG      : L4," &
"OUTVDD3       : L3," &
"CLKD3         : L2," &
"OUTVDD4       : L1," &
"OUTGND2       : M3," &
"SYSCLK        : M2," &
"OUTGND3       : M1," &
"CLK2X         : N4," &
"OUTVDD5       : N3," &
"OUTGND4       : N2," &
"LDATA31       : N1," &
"LDATA30       : P1," &
"LDATA29       : P2," &

```

```

"LDATA28      : P3," &
"OUTVDD6      : P4," &
"LDATA27      : R1," &
"LDATA26      : R2," &
"LDATA25      : R3," &
"LDATA24      : R4," &
"OUTGND5      : T1," &
"OUTVDD7      : T2," &
"LDATA23      : T3," &
"LDATA22      : T4," &
"LDATA21      : U1," &
"OUTVDD8      : U2," &
"LDATA20      : U3," &
"LDATA19      : U4," &
"LDATA18      : V1," &
"OUTGND6      : V2," &
"LDATA17      : V3," &
"LDATA16      : V4," &
"LDATA15      : W1," &
"LDATA14      : W2," &
"LDATA13      : W3," &
"OUTGND7      : W4," &
"LDATA12      : Y1," &
"LDATA11      : Y2," &
"LDATA10      : Y3," &
"LDATA9       : Y4," &
"LDATA8       : AA1," &
"OUTGND8      : AA2," &
"LDATA7       : AA3," &
"LDATA6       : AA4," &
"LDATA5       : AB1," &
"LDATA4       : AB2," &
"OUTVDD9      : AB3," &
"RXDATA15     : AE5," &
"RXDATA14     : AF5," &
"RXDATA13     : AC6," &
"RXDATA12     : AD6," &
"RXDATA11     : AE6," &
"RXDATA10     : AF6," &
"RXDATA9      : AC7," &
"RXDATA8      : AD7," &
"LDATA3       : AB4," &
"LDATA2       : AC1," &
"LDATA1       : AC2," &
"LDATA0       : AC3," &
"OUTGND9      : AD1," &
"OUTVDD10     : AD2," &
"SCHREF       : AE1," &

```

```

"CGND0      : AF1," &
"CVDD0      : AE2," &
"OUTGND10   : AC4," &
"RXADDR4    : AF2," &
"RXADDR3    : AE3," &
"RXADDR2    : AF3," &
"RXADDR1    : AD4," &
"RXADDR0    : AE4," &
"OUTGND11   : AF4," &
"OUTVDD11   : AC5," &
"RXPAR      : AD5," &
"RXDATA7    : AE7," &
"RXDATA6    : AF7," &
"RXDATA5    : AC8," &
"RXDATA4    : AD8," &
"RXDATA3    : AE8," &
"RXDATA2    : AF8," &
"RXDATA1    : AC9," &
"OUTGND12   : AD9," &
"OUTVDD12   : AE9," &
"RXDATA0    : AF9," &
"RXSOC      : AC10," &
"RXEN_NEG   : AD10," &
"OUTGND13   : AE10," &
"RXCLAV_NEG : AE11," &
"OUTVDD13   : AC12," &
"TXCLK      : AD12," &
"FRCFG0     : AE12," &
"FRCFG1     : AF12," &
"FRCTRL     : AC13," &
"OUTGND14   : AD13," &
"OUTVDD14   : AE13," &
"OUTGND15   : AF14," &
"OUTVDD15   : AE14," &
"UTOPIA1    : AD14," &
"TXEN_NEG   : AC14," &
"TXSOC      : AF15," &
"TXCLAV_NEG : AF16," &
"TXPAR      : AE16," &
"TXDATA7    : AF19," &
"TXDATA6    : AE19," &
"OUTGND16   : AD19," &
"OUTVDD16   : AC19," &
"TXDATA5    : AF20," &
"TXDATA4    : AE20," &
"TXDATA3    : AD20," &
"TXDATA2    : AC20," &
"TXDATA1    : AF21," &

```



```

"TXDATA0      : AE21," &
"OUTVDD17     : AD21," &
"TXADDR4      : AC21," &
"TXADDR3      : AF22," &
"TXADDR2      : AE22," &
"TXADDR1      : AD22," &
"TXADDR0      : AC22," &
"OUTGND17     : AF23," &
"OUTVDD18     : AE23," &
"OUTVDD19     : AF24," &
"CGND1        : AF26," &
"CVDD1        : AE25," &
"TCLK         : AC23," &
"OUTVDD20     : AE26," &
"TRST_NEG     : AD25," &
"TMS          : AD26," &
"TDO          : AC24," &
"TDI          : AC25," &
"OUTVDD21     : AC26," &
"SCL          : AB23," &
"SDA          : AB24," &
"PCI5V        : AB26," &
"OUTGND18     : AA23," &
"OUTVDD22     : AA24," &
"HINT_NEG     : AA25," &
"OUTVDD23     : AD16," &
"TXDATA15     : AC16," &
"TXDATA14     : AF17," &
"TXDATA13     : AE17," &
"TXDATA12     : AD17," &
"TXDATA11     : AC17," &
"OUTGND19     : AF18," &
"TXDATA10     : AE18," &
"TXDATA9      : AD18," &
"TXDATA8      : AC18," &
"OUTVDD24     : AA26," &
"HRST_NEG     : Y23," &
"OUTGND20     : Y24," &
"HGNT_NEG     : Y25," &
"HREQ_NEG     : Y26," &
"OUTGND21     : W23," &
"HAD31        : W24," &
"HAD30        : W25," &
"HAD29        : W26," &
"OUTVDD25     : V23," &
"HAD28        : V24," &
"HAD27        : V25," &
"HAD26        : V26," &

```

```

"HAD25      : U23," &
"HAD24      : U24," &
"OUTGND22   : U25," &
"OUTVDD26   : U26," &
"HCBE3_NEG  : T23," &
"HIDSEL     : T24," &
"HAD23      : T25," &
"OUTVDD27   : T26," &
"HAD22      : R23," &
"HAD21      : R24," &
"HAD20      : R25," &
"HAD19      : R26," &
"OUTGND23   : P23," &
"HAD18      : P24," &
"HAD17      : P25," &
"HAD16      : P26," &
"HCBE2_NEG  : N26," &
"OUTGND24   : N25," &
"OUTVDD28   : N24," &
"HCLK       : N23," &
"OUTGND25   : M26," &
"HFRAME_NEG : M25," &
"HIRDY_NEG  : M24," &
"HTRDY_NEG  : M23," &
"HDEVSEL_NEG: L26," &
"HSTOP_NEG  : L25," &
"HPERR_NEG  : L24," &
"HSERR_NEG  : L23," &
"OUTGND26   : K26," &
"OUTVDD29   : K25," &
"HPAR       : K24," &
"HCBE1_NEG  : K23," &
"HAD15      : J26," &
"HAD14      : J25," &
"HAD13      : J24," &
"OUTVDD30   : J23," &
"OUTGND27   : H26," &
"HAD12      : H25," &
"HAD11      : H24," &
"HAD10      : H23," &
"HAD9       : G26," &
"HAD8       : G25," &
"OUTGND28   : G24," &
"OUTVDD31   : G23," &
"HCBE0_NEG  : F26," &
"HAD7       : F25," &
"HAD6       : F24," &
"HFIFORD0   : C25," &

```

```

"HFIFORD1      : B26," &
"HFIFORD2      : C24," &
"HFIFORD3      : D23," &
"HFIFORD4      : A25," &
"HFIFORD5      : C23," &
"HENUM_NEG     : B23," &
"EEPWR         : D22," &
"HAD5          : F23," &
"HAD4          : E26," &
"OUTGND29     : E25," &
"OUTVDD32     : E24," &
"HAD3          : E23," &
"HAD2          : D26," &
"HAD1          : D25," &
"HAD0          : D24," &
"OUTGND30     : C26," &
"OUTGND31     : A26," &
"OUTVDD33     : B25," &
"OUTGND32     : B24," &
"OUTVDD34     : A24," &
"OUTGND33     : A23," &
"OUTVDD35     : D21," &
"OUTGND34     : C20," &
"OUTVDD36     : B19," &
"OUTGND35     : A18," &
"CGND2        : D16," &
"CVDD2        : C16," &
"PROCMODE     : B16," &
"PRST_NEG     : A16," &
"PINT_NEG     : D15," &
"PFAIL_NEG    : C15," &
"OUTGND36     : B14," &
"OUTVDD37     : A14," &
"PDAEN_NEG    : B15," &
"PCS_NEG     : A15," &
"PAS_NEG     : D14," &
"PBLAST_NEG   : C14," &
"PWAIT_NEG   : A13," &
"PRDY_NEG    : B13," &
"PWNR        : C13," &
"OUTGND37     : D13," &
"OUTVDD38     : A12," &
"PBE3_NEG     : B12," &
"PBE2_NEG     : C12," &
"PBE1_NEG     : D12," &
"OUTGND38     : A11," &
"OUTVDD39     : B11," &
"PBE0_NEG     : C11," &

```

```

"PBSEL1      : D11," &
"PBSEL0      : A10," &
"PADDR1      : B10," &
"PADDR0      : C10," &
"OUTGND39    : D10," &
"OUTVDD40    : A9," &
"LADDR18     : B9," &
"LADDR17     : C9," &
"LADDR16     : D9," &
"LADDR15     : A8," &
"LADDR14     : B8," &
"OUTGND40    : C8," &
"OUTVDD41    : D8," &
"LADDR13     : A7," &
"LADDR12     : B7," &
"LADDR11     : C7," &
"LADDR10     : D7," &
"LADDR9      : A6," &
"LADDR8      : B6," &
"OUTGND41    : C6," &
"OUTVDD42    : D6," &
"LADDR7      : A5," &
"LADDR6      : B5," &
"LADDR5      : C5," &
"LADDR4      : D5," &
"LADDR3      : A4," &
"LADDR2      : B4," &
"OUTGND42    : C3," &
"OUTVDD43    : D3," &
"LADDR1      : C4," &
"LADDR0      : A3," &
"OUTGND43    : B3," &
"OUTVDD44    : C1," &
"OUTGND44    : D2," &
"PVGG0      : E4," &
"CGND3      : E2," &
"CVDD3      : E1," &
"STAT1      : F2";

```

-- TAP Port Name Attributes

```

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 (1.0e+07, BOTH);
attribute TAP_SCAN_RESET of TRST_NEG : signal is true;

```

-- Instruction Register Attributes

```

attribute INSTRUCTION_LENGTH of CN8236 : entity is 4;

```

attribute INSTRUCTION\_OPCODE of CN8236 : entity is

```
"EXTEST      (0000)," &
"RSTHIGH     (0001)," &
"SAMPLE      (0010)," &
"TMWAFIFO    (0011)," &
"TMWBFIFO    (0100)," &
"TMRFIFO     (0101)," &
"TSEGFIFO    (0110)," &
"TRSMFIFO    (0111)," &
"T38AFIFO    (1000)," &
"T38BFIFO    (1001)," &
"IDCODE      (1010)," &
"HIGHZ       (1011)," &
"RSVD0       (1100)," &
"RSVD1       (1101)," &
"RSVD2       (1110)," &
"BYPASS      (1111)";
```

attribute INSTRUCTION\_CAPTURE of CN8236 : entity is "0001";

attribute INSTRUCTION\_PRIVATE of CN8236 : entity is

```
"RSTHIGH," &
"TMWAFIFO," &
"TMWBFIFO," &
"TMRFIFO," &
"TSEGFIFO," &
"TRSMFIFO," &
"T38AFIFO," &
"T38BFIFO," &
"RSVD0," &
"RSVD1," &
"RSVD2";
```

end CN8236;



## Appendix B: List of Acronyms

---

### A

|     |                            |
|-----|----------------------------|
| AAL | ATM Adaption Layer         |
| ABR | Available Bit Rate         |
| ACR | Allowed Cell Rate          |
| ATM | Asynchronous Transfer Mode |

### B

|        |                                    |
|--------|------------------------------------|
| BASIZE | Buffer Allocation Size             |
| BGA    | Ball Grid Array                    |
| BOM    | Beginning of Message               |
| BSDL   | Boundary Scan Description Language |
| BTAG   | Beginning Tag                      |

### C

|      |                                  |
|------|----------------------------------|
| CBR  | Constant Bit Rate                |
| CCR  | Current Cell Rate                |
| CDV  | Cell Delay Variation             |
| CDVT | Cell Delay Variation Tolerance   |
| CLP  | Cell Loss Priority               |
| CLR  | Cell Loss Ratio                  |
| CI   | Congestion Indication            |
| COM  | Continuation of Message          |
| CPCS | Common Part Convergence Sublayer |
| CPI  | Common Part Indicator            |
| CPU  | Central Processing Unit          |
| CRC  | Cyclic Redundancy Check          |
| CSR  | Control and Status Registers     |
| CTD  | Cell Transfer Delay              |

### D

|     |                      |
|-----|----------------------|
| DE  | Discard Eligibility  |
| DMA | Direct Memory Access |

### E

|      |                      |
|------|----------------------|
| EDC  | Error Detection Code |
| EOM  | End of Message       |
| EPD  | Early Packet Discard |
| ER   | Explicit Rate        |
| ETAG | Ending Tag           |
| EVS  | Evaluation System    |

**F**

FIFO First In First Out

**G**GCRA Generic Cell Rate Algorithm  
GFC Generic Flow Control  
GFR Guaranteed Frame Rate  
GPIO General Purpose Input/Output**I**IETF Internet Engineering Task Force  
ILMI Interim Local Management Interface**J**

JTAG Joint Test Action Group

**L**LANE LAN Emulation  
LEC LAN Emulation Client  
LECID LAN Emulation Client ID  
LEN Length  
LSB Least Significant Bit**M**Mbit Megabit  
Mbps Megabits Per Second  
MBS Maximum Burst Size  
MIB Management Information Base  
MSB Most Significant Bit**N**NI No Increase  
NIC Network Interface Card  
NNI Network-to-Network Interface**O**

OAM Operation And Maintenance

**P**PCI Peripheral Component Interconnect  
PCR Peak Cell Rate  
PDU Protocol Data Unit  
PFE Package Forward Engine  
PHY Physical Layer Device  
PM Performance Monitoring  
PTI Payload Type Identifier  
PVC Permanent Virtual Connections (Circuit)**Q**

QoS Quality of Service



**R**

|     |                        |
|-----|------------------------|
| RDB | Rate Decision Block    |
| RDF | Rate Decrease Factor   |
| RM  | Resource Management    |
| RR  | Relative Rate          |
| RSM | Reassembly Coprocessor |

**S**

|            |  |
|------------|--|
| SAM        | Service Access Multiplexer                     |
| SBD        | Segmentation Buffer Descriptor                 |
| SCR        | Sustainable Cell Rate                          |
| SDU        | Service Data Unit                              |
| SEG        | Segmentation Coprocessor                       |
| ServiceSAR | Service Segmentation and Reassembly Controller |
| SMDS       | Switched Multimegabit Data Service             |
| SNMP       | Simple Network Management Protocol             |
| SRAM       | Static Random Access Memory                    |
| SRC        | Segmentation and Reassembly Controller Chip    |
| SSM        | Single Segment Message                         |
| ST         | Segment Type                                   |
| SVC        | Switched Virtual Circuit (Connection)          |

**U**

|        |   |
|--------|---|
| UNI    | User-to-Network Interface                               |
| UTOPIA | Universal Test and Operation Physical Interface for ATM |
| UU     | User-to-User  |

**T**

|     |                             |
|-----|-----------------------------|
| TCR | Tagged Cell Rate            |
| TTL | Transistor-Transistor Logic |

**U**

|     |                      |
|-----|----------------------|
| UBR | Unspecified Bit Rate |
|-----|----------------------|

**V**

|     |                            |
|-----|----------------------------|
| VBR | Variable Bit Rate          |
| VC  | Virtual Circuit            |
| VCC | Virtual Channel Connection |
| VCI | Virtual Channel Identifier |
| VP  | Virtual Path               |
| VPI | Virtual Path Identifier    |



**MINDSPEED™**

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

