SIEMENS

ICs for Communications

Segmentation and Reassembly Element SARE

PXB 4110 Version 1.1

Preliminary Data Sheet 01.97

T4110-XV11-P2-7600

PXB 4110 **Revision History:** **Current Version: 01.97**		
Previous Version: Preliminary Data Sheet 08.95 (Version 1.1)		
Page (in previous Version)	Page (in new Version)	Subjects (changes since last revision)
9-10, 25-29, 32-37, 40, 54-56	10-11, 24-28, 31-36, 39, 55-57	Up to 32 simultaneous connections/links changed to typically 60 (or at least 32) simultaneous virtual connections/links throughout document.
10-11, 13, 142	10-13, 144	Package changed from P-FQFP-208-2 to P-FQFP-208-4.
12-13, 17, 21, 24	13, 19, 23	UCMOD added to UTOPIA Interface, Pin 31 changed from NC. UTOPIA Master/Slave, M/S, inverted high/low, Pin 187.
13, 26	13, 25	Pin 147, PINT, changed to $\overline{\text{PINT}}$.
18	17	Pin 32, $\overline{\text{PERR}}$, changed from Output (O) to Open Drain (OD).
19	18	Max. clock frequency changed from 33 (66) MHz to 33 MHz.
9, 30	11, 30	ATM NIC application: deleted reference to UTPT.
36	35	“With or without CRC32/Multiplexilng PDUs” moved from AAL5 to AAL3/4.
38-41, 98	37-41, 95	Correction and clarification of RVF registers, VPCI and RVTB, including text, tables, equations, and figures.
43, 58, 85, 112-114	43, 58, 86, 110-112	Timestamp changed to Reserved or reference to Timestamp removed.
63	64	Bit rate of virtual channel changed from r = 155.52 Mbit/s × TiK0/TiL to 8 × (frequency of UTOPIA TxCLK) × TiK0/TiL.
76	76	External local memory address access was changed from AD(19-0) to AD(21-0).
81	80	Additional description regarding location of control data for proper device operation.
88, 89	87	Table 31: Vector ID reset value changed from 1029_H to $110A_H$; Interrupt line reset value changed from 00_H to FF_H; Interrupt pin reset value changed from 00_H to 01_H; Min_Lat changed to Max_Lat.
91	88	TmK0 Access from SARE changed from R/W to R.
109-111	104-109	Table 35: C_TCKMOD changed to Reserved; C_TCKDIV description updated; DELCON function removed; C_BPARK added.
126-127	123-125	Ambient temperature changed from 0 to 70 °C to – 40 to + 85 °C; Supply voltage, 3 V, changed from – 0.5 to 4.125 V.
126-127	124-125	Added DC characteristics: High-Level output voltage for PCI pins; Power consumption on V_{DD3} and V_{DD5}; Input and output leakage current for PIC pins.
134-135	133	UTOPIA Interface signal characteristics: Delay time values updated.

PXB 4110 Revision History: Current Version: 01.97		
Previous Version: Preliminary Data Sheet 08.95 (Version 1.1)		
Page (in previous Version)	Page (in new Version)	Subjects (changes since last revision)
139-140	139	Table 42: CLK to signal valid delay values changed; Input setup time values changed.
140-141	140-141	Local Memory/Control Interface Timing: figure 31 and table 44 replaced; figure 32 and table 44 added.

Edition 01.97

This edition was realized using FrameMaker® and Designer software.

**Published by Siemens AG,
Bereich Halbleiter, Marketing-
Kommunikation, Balanstraße 73,
81541 München**

Attention please!

As far as patents or other rights of third parties are concerned, liability is only assumed for components, not for applications, processes and circuits implemented within components or assemblies.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

For questions on technology, delivery and prices please contact the Semiconductor Group Offices in Germany or the Siemens Companies and Representatives worldwide (see address list).

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Siemens Office, Semiconductor Group.

Siemens AG is an approved CECC manufacturer.

Packing

Please use the recycling operators known to you. We can also help you – get in touch with your nearest sales office. By agreement we will take packing material back, if it is sorted. You must bear the costs of transport.

For packing material that is returned to us unsorted or which we are not obliged to accept, we shall have to invoice you for any costs incurred.

Components used in life-support devices or systems must be expressly authorized for such purpose!

Critical components[1] of the Semiconductor Group of Siemens AG, may only be used in life-support devices or systems[2] with the express written approval of the Semiconductor Group of Siemens AG.

1 A critical component is a component used in a life-support device or system whose failure can reasonably be expected to cause the failure of that life-support device or system, or to affect its safety or effectiveness of that device or system.

2 Life support devices or systems are intended (a) to be implanted in the human body, or (b) to support and/or maintain and sustain human life. If they fail, it is reasonable to assume that the health of the user may be endangered.

Table of Contents **Page**

Table of Contents **Page**

Table of Contents **Page**

List of Figures **Page**

List of Tables **Page**

1 Overview

The Segmentation And Reassembly Element SARE PXB 4110 is a member of the Siemens ATM chip set. Following the modular chip concept, the SARE has the UTOPIA industry-standard interface which is also used by other components of the ATM chip set, including SDHT (PXB 4240), IWE (PXB 4220), ASP-up (PXB 43201) and ASP-down (PXB 43202).

The SARE conforms to the most advanced architectural requirements in equipment that terminates AAL connections in ATM networks. It supports processing of the Segmentation/Reassembly and ATM Adaptation Layers for up to 64K simultaneous connections. The type of AAL (either AAL5, AAL3/4, SMDS, OAM cell mode or transparent mode; or cell FIFO mode for constant bit rate) is individually programmable per Virtual Connection. The interface to the system is formed by a shared packet memory via the on-chip PCI compliant interface.

For packets to be segmented, the integrated DMA controller autonomously transfers user data from the central shared packet memory to its internal segmentation unit on a cell-by-cell basis. Received cells are checked for correctness and likewise transferred to the central memory by DMA via the PCI Interface.

For optimum usage of memory space, the on-chip DMA controller implements scatter/ gather data storage using linked list data buffers, with programmable buffer length. The descriptors of the data buffers in segmentation/reassembly are stored locally for each active virtual connection. Typically, at least 60 simultaneously active virtual connections are supported without external local memory, because the active descriptors and link context data are stored in on-chip memory. In the case of more than 60 simultaneously active virtual connections, the buffer descriptors and link context data for up to 64K connections can be resident in a local RAM connected to the SARE through a separate 32-bit parallel bus.

The interface of the SARE to the ATM Physical Layer is a UTOPIA Level 1 Interface, with 8-bit data buses. In the case where the SARE is used in an ATM Switch/Hub, it can be configured to operate as a clock slave with respect to the UTOPIA Interface (instead of master, as in ATM network attachment applications) to directly connect to an ATM Switch Preprocessor (ASP).

Direct access to SARE internal registers is provided via the PCI Interface. The SARE directly provides the interface to control and monitor other devices (like PHY devices) from the PCI Interface without the need of a separate control bus, thus minimizing the component cost in particular in Network Interface Card applications. For testing purposes and for applications with local packet memory, access to local memory from the PCI is also possible via the SARE.

Overall, because of these features, the SARE enables the implementation of glueless, cost-optimized solutions for AAL terminating equipment in ATM networks.

SIEMENS

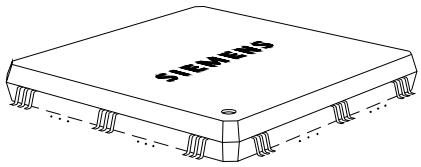
Segmentation and Reassembly Element SARE

PXB 4110

Version 1.1

CMOS

1.1 Features



P-FQFP-208-4

- **General Features**
  - Processing of ATM Adaptation Layer Segmentation/Reassembly and Convergence Sublayer protocols
  - Programmable for AAL5, AAL3/4, transparent and Cell FIFO mode (for CBR support) on a connection-by-connection basis
  - Implementation of segmentation/reassembly for at least 60 (typical) simultaneous virtual connections without external RAM
  - Implementation of segmentation/reassembly for up to 64K simultaneous virtual connections with additional external local RAM
  - Optimized load-sharing between local memory bus (for context and overhead data) and PCI system bus (only payload data)
  - Random VPI/VCI values for 32 virtual connections without external RAM (or 64K virtual connections with external RAM)
  - CRC-32 calculation/check programmable on a connection-by-connection basis
  - Cell rate shaping with priorities, up to 8 (dual) leaky buckets
  - VC level OAM cell detection and CRC-10 calculation
  - Transmission and reception of ATM cells with bit rate up to 155 Mbit/s
- **Interfaces**
  - UTOPIA Interface (Master/Slave modes) Level 1 according to newest standards; output buffer of 4 ATM cell depth in transmit direction for UTOPIA cell level handshake
  - PCI bus interface to central packet memory/host
  - 32-bit optional bus interface for local memory and/or for control of peripheral (e.g. PHY) devices
  - Programmable clock frequency (maximum operating frequency 33 MHz)

Type	Ordering Code	Package
PXB 4110	Q67101-H6589	P-FQFP-208-4

- **System Bus Interface**
  - 32-bit PCI bus
  - Output buffer for 128 double words for compensation of PCI bus latency (≈ 27 µs)
  - On-chip bus master DMA controller with packet scatter/gather capability for up to 64K simultaneous virtual connections
  - Flexible linked list buffer structures with programmable buffer length
- **Control Functions**
  - Control via PCI bus (Master/Slave modes)
- **Miscellaneous**
  - JTAG boundary scan according to IEEE 1149.1
  - Built-in data path loop for test
  - P-FQFP-208-4 package
  - Low power, 3.3 V 0.5 µ CMOS technology
  - 5 V PCI Interface
  - 3.3 V UTOPIA Interface (5 V tolerant for 5 V environment)
  - All other interfaces: 3.3 V compatible
- **Applications**
  - ATM End User equipment
  - Motherboards for Multimedia PCs and Workstations
  - Network Interface Cards (NICs) for Workstations/PCs and Servers
  - Routers and Gateways; SMDS networks
  - ATM Hubs and Switches

1.2 Logic Symbol

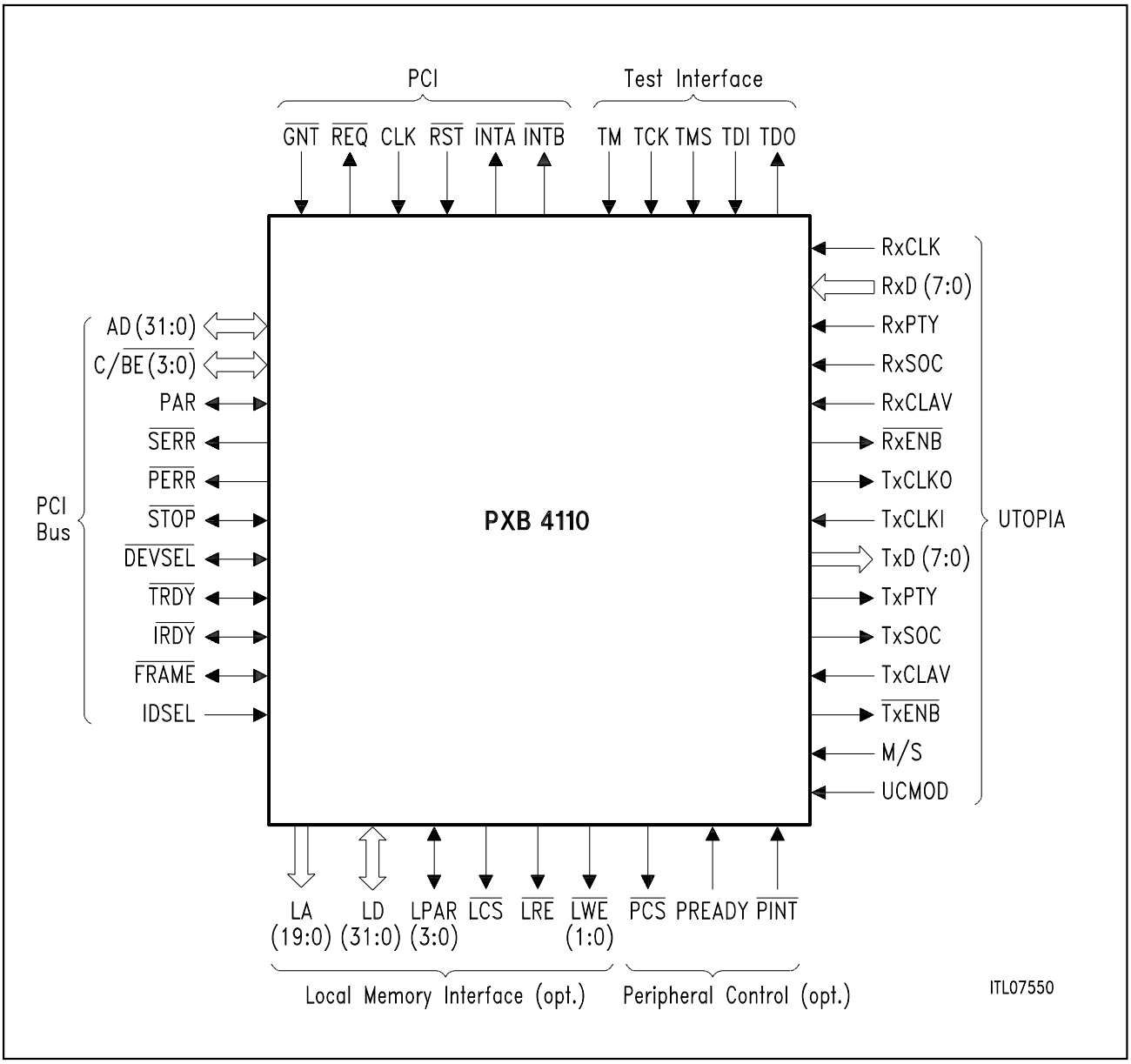


Figure 1 PXB 4110 Segmentation and Reassembly Element Logic Symbol

1.3 Pin Configuration

(top view)

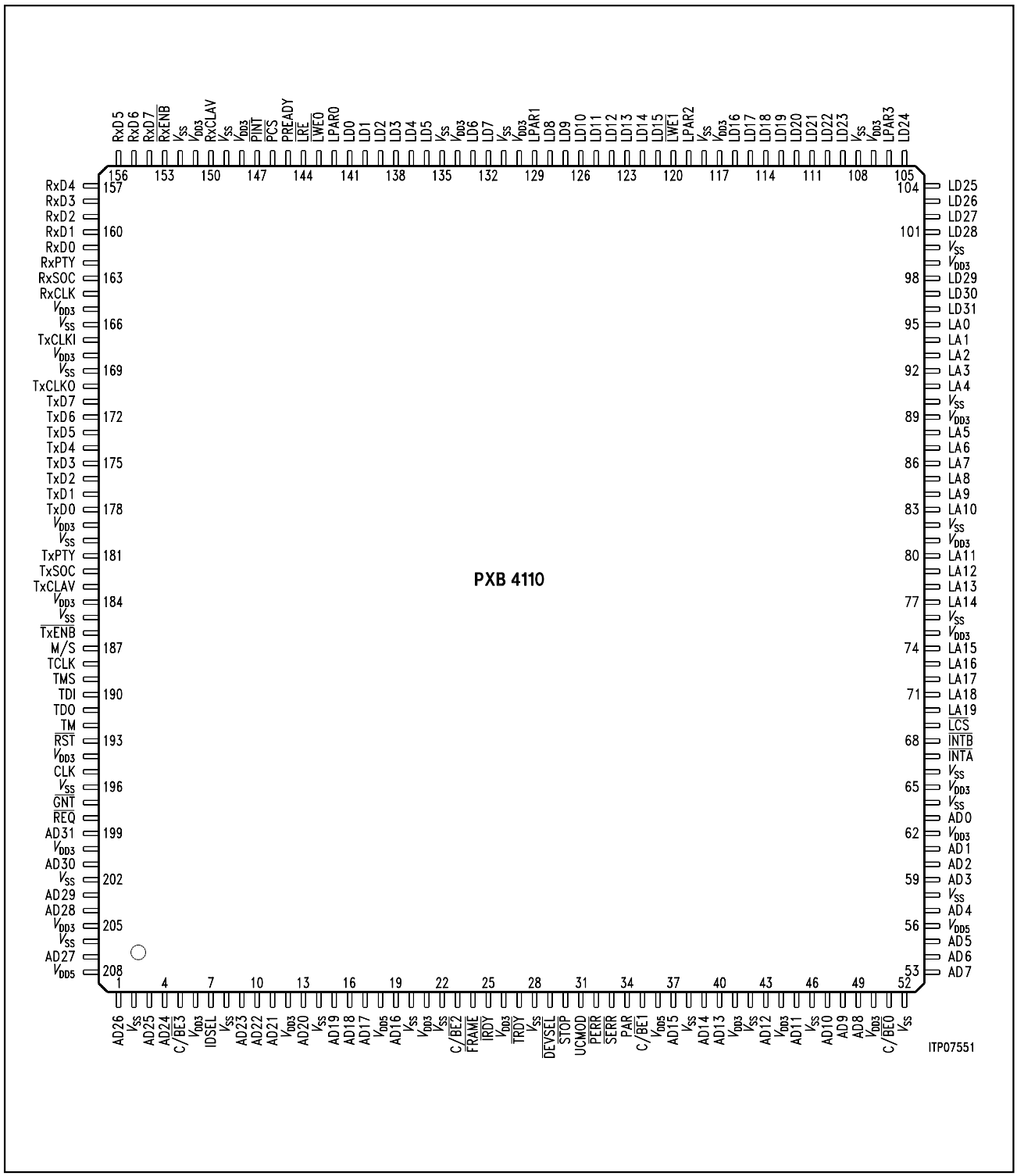


Figure 2 SARE Pin Configuration

1.4 Pin Definitions and Functions

1.4.1 Parallel Host Interface

In **table 1**: The terms “Master” (M) and “Slave” (S) are sometimes used interchangeably with “initiator” or “target” respectively to refer to the SARE’s role in a bus transaction; and (OD) in the Input/Output column indicates an Open Drain signal.

Table 1 Parallel Host Interface

Pin No.	Symbol	Input (I) Output (O)	Function
63, 61, 60, 59, 57, 55, 54, 53, 49, 48, 47, 45, 43, 40, 39, 37, 19, 17, 16, 15, 13, 11, 10, 9, 4, 3, 1, 207, 204, 203, 201, 199	AD0, AD1, AD2, AD3, AD4, AD5, AD6, AD7, AD8, AD9, AD10, AD11, AD12, AD13, AD14, AD15, AD16, AD17, AD18, AD19, AD20, AD21, AD22, AD23, AD24, AD25, AD26, AD27, AD28, AD29, AD30, AD31	I/O	**Address/Data Bus** A bus transaction consists of an address phase followed by one or more data phases. When SARE is Master, AD(31:0) are outputs in the address phase of a transaction. During the data phase, AD(31:0) remain outputs for write transactions, and become inputs for read transactions.When SARE is Slave, AD(31:0) are inputs in the address phase of a transaction. During the data phase, AD(31:0) remain inputs for write transactions, and become outputs for read transactions. AD(31:0) are updated and sampled on the rising edge of CLK.
51, 35, 23, 5	C/$\overline{BE0}$, C/$\overline{BE1}$, C/$\overline{BE2}$, C/$\overline{BE3}$	I/O	**Command/Byte Enable** During the address phase of a transaction, C/$\overline{BE(3:0)}$ define the bus command. During the data phase C/$\overline{BE(3:0)}$ are used as Byte Enables. The Byte Enables are valid for the entire data phase and determine which byte lanes carry meaningful data. C/$\overline{BE0}$ applies to byte 0 (lsb) and C/$\overline{BE3}$ applies to byte 3 (msb). When SARE is Master, C/$\overline{BE(3:0)}$ are outputs. When SARE is Slave, C/$\overline{BE(3:0)}$ are inputs. C/$\overline{BE(3:0)}$ are updated and sampled on the rising edge of CLK.

Table 1 Parallel Host Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
34	PAR	I/O	**Parity** PAR is even parity across AD(31:0) and C/$\overline{\text{BE}}$(3:0). PAR is stable and valid one clock after the address phase. PAR has the same timing as AD(31:0) but is delayed by one clock. When SARE is Master, PAR is output during address phase and write data phases. When SARE is Slave, PAR is output during read data phases. Parity errors detected by the SARE are indicated on $\overline{\text{PERR}}$ output. PAR is updated and sampled on the rising edge of CLK.
24	$\overline{\text{FRAME}}$	I/O	**Cycle Frame** $\overline{\text{FRAME}}$ indicates the beginning and duration of an access. $\overline{\text{FRAME}}$ is asserted to indicate a bus transaction is beginning. While $\overline{\text{FRAME}}$ is asserted, data transfers continue. When $\overline{\text{FRAME}}$ is deasserted, the transaction is in the final phase. When SARE is Master, $\overline{\text{FRAME}}$ is an output. When SARE is Slave, $\overline{\text{FRAME}}$ is an input. $\overline{\text{FRAME}}$ is updated and sampled on the rising edge of CLK.

Table 1 Parallel Host Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
25	$\overline{\text{IRDY}}$	I/O	**Initiator Ready** $\overline{\text{IRDY}}$ indicates the bus master's ability to complete the current data phase of the transaction. It is used in conjunction with $\overline{\text{TRDY}}$. A data phase is completed on any clock where both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are sampled asserted. During a write, $\overline{\text{IRDY}}$ indicates that valid data is present on AD(31:0). During a read, it indicates the master is prepared to accept data. Wait cycles are inserted until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted together. When SARE is Master, $\overline{\text{IRDY}}$ is an output. When SARE is Slave, $\overline{\text{IRDY}}$ is an input. $\overline{\text{IRDY}}$ is updated and sampled on the rising edge of CLK.
27	$\overline{\text{TRDY}}$	I/O	**Target Ready** $\overline{\text{TRDY}}$ indicates a Slave's ability to complete the current data phase of the transaction. During a read, $\overline{\text{TRDY}}$ indicates that valid data is present on AD(31:0). During a write, it indicates the target is prepared to accept data. When SARE is Master, $\overline{\text{TRDY}}$ is an input. When SARE is Slave, $\overline{\text{TRDY}}$ is an output. $\overline{\text{TRDY}}$ is updated and sampled on the rising edge of CLK.
30	$\overline{\text{STOP}}$	I/O	**STOP** $\overline{\text{STOP}}$ is used by a Slave to request the current Master to stop the on-going bus transaction. When SARE is Master, $\overline{\text{STOP}}$ is an input. When SARE is Slave, $\overline{\text{STOP}}$ is an output. $\overline{\text{STOP}}$ is updated and sampled on the rising edge of CLK.

Table 1 Parallel Host Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
7	IDSEL	I	**Initialization Device Select** When SARE is Slave in a transaction, if IDSEL is active in the address phase and C/$\overline{BE}$(3:0) indicates a register read or write, the SARE assumes a read or write to a configuration register. In response, the SARE asserts $\overline{DEVSEL}$ in the subsequent CLK period. IDSEL is sampled on the rising edge of CLK.
29	$\overline{DEVSEL}$	I/O	**Device Select** When activated by a Slave, indicates to the current bus master that the Slave has decoded its address as the target of the current transaction, i.e. that the address is valid. If no bus slave activates $\overline{DEVSEL}$ within six bus clock cycles, the Master should abort the transaction. When SARE is Master, $\overline{DEVSEL}$ is input. If $\overline{DEVSEL}$ is not activated within six clock cycles after an address is output on AD(31:0), the SARE aborts the transaction and generates an interrupt status. When SARE is Slave, $\overline{DEVSEL}$ is output.
32	$\overline{PERR}$	O (OD)	**Parity Error** When activated, indicates a parity error over the AD(31:0) and C/$\overline{BE}$(3:0) signals when compared to the PAR input. It has a delay of two CLK cycles with respect to AD and C/$\overline{BE}$, i.e. it is activated for the cycle immediately following the corresponding PAR cycle. $\overline{PERR}$ is activated on the rising edge of CLK.
33	$\overline{SERR}$	O (OD)	**System Error** The SARE activates this signal for indicating a fatal system error. $\overline{SERR}$ is activated on the rising edge of CLK.

Table 1 Parallel Host Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
198	$\overline{\text{REQ}}$	O	**Request** This signal is used by the SARE to request control of the PCI bus. $\overline{\text{REQ}}$ is activated on the rising edge of CLK.
197	$\overline{\text{GNT}}$		**Grant** This signal is activated to grant control of the PCI to the SARE in response to a bus request via $\overline{\text{REQ}}$. After $\overline{\text{GNT}}$ is activated, the SARE may begin a bus transaction only after $\overline{\text{FRAME}}$ has been deactivated. $\overline{\text{GNT}}$ is sampled on the rising edge of CLK.
195	CLK	I	**Clock** Provides timing for all PCI transactions, the DMA controller and the AAL processor. Generally, all PCI signals are sampled and output on the rising edge of CLK. The maximum CLK frequency is 33 MHz.
193	$\overline{\text{RST}}$	I	**Reset** An active $\overline{\text{RST}}$ signal brings all PCI registers, sequencers and signals into a consistent state. All PCI output signals are driven to their begin state.
67	$\overline{\text{INTA}}$	O (OD)	**Interrupt Request A** When an interrupt status is active and the interrupt source is unmasked, the SARE activates this open-drain output. Examples of interrupt sources are alarms, or events pertaining to the DMA controller. SARE deactivates $\overline{\text{INTA}}$ when the interrupt status is acknowledged via an appropriate action (e.g. specific register read) and no other unmasked interrupt statuses are active. $\overline{\text{INTA}}$ is activated on the rising edge of CLK.

Table 1 Parallel Host Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
68	$\overline{\text{INTB}}$	O (OD)	**Interrupt Request B** See above. The possible interrupt sources (statuses) of the SARE may be programmed to generate an interrupt on either $\overline{\text{INTA}}$ or $\overline{\text{INTB}}$ (or none). This feature can be used to assign interrupt sources e.g. in two groups of different priorities ($\overline{\text{INTA}}$: higher priority interrupts; $\overline{\text{INTB}}$: lower priority interrupts), as determined via the software requirements. $\overline{\text{INTB}}$ is activated on the rising edge of CLK.

1.4.2 UTOPIA Interface

For the sake of convenience, the signals on the UTOPIA Interface are named according to their significance in the Master mode (NIC applications, i.e. the SARE is connected to a PHY device). In the case of the Slave mode (Switch/Hub) applications, the UTOPIA Interface should be configured (via the M/S pin) as a Slave with respect to clocking. Then the significance of the pins changes as follows:

Master mode (signal name)	Function in Slave mode
RxCLAV	Transmit Enable
$\overline{\text{RxENB}}$	Transmit Cell Space Available
TxCLAV	Receive Enable
$\overline{\text{TxENB}}$	Receive Cell Available.

See **table 2** for a description of the pins on the UTOPIA Interface.

Table 2 UTOPIA Interface

Pin No.	Symbol	Input (I) Output (O)	Function
31	UCMOD	I	In the case where the SARE operates as a UTOPIA interface clock master, UCMOD determines the mode of generation of the UTOPIA TxCLK clock as follows: If UCMOD is "low" ,the clock (TxCLKO output) is derived from TxCLKI input (same frequency). If UCMOD is "high", the clock (TxCLKO output) is derived from the PCI clock by division by a programmable factor.

Table 2 UTOPIA Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
167	TxCLKI	I	**Transmit Clock In** Used to clock the UTOPIA Interface output bus when the SARE operates as a UTOPIA Interface Slave, e.g. when the SARE is connected to an ATM Switch Preprocessor ASP (hub applications). In the case where the SARE operates as a UTOPIA Interface Master, the master transmit clock on TxCLKO may optionally be derived from the signal provided on TxCLKI (e.g. this may be connected externally to RxCLK). The signals of the interface are evaluated with the rising edge of this clock.
170	TxCLKO	O	**Transmit Clock Out** Used to clock the UTOPIA Interface output bus when the SARE operates as a UTOPIA Interface Master (NIC applications). TxCLKO is either derived from TxCLKI (with the same frequency), or from the PCI clock (divided by a programmable factor equal to 1, 2, 4, 8 or 16). The signals of the interface are evaluated with the rising edge of this clock.
164	RxCLK	I	**Receive Clock** Used to clock the UTOPIA Interface input bus. It should be connected to the RxCLK line of the UTOPIA Interface (according to UTOPIA nomenclature) when the SARE operates as a UTOPIA Interface Master (e.g. in NIC applications). It should be connected to the UTOPIA TxCLK line when the SARE operates as a UTOPIA Interface Slave. Its maximum frequency is 33 MHz. The signals of the interface are evaluated with the rising edge of this clock.

Table 2 UTOPIA Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
161, 160, 159, 158, 157, 156, 155, 154	RxD0, RxD1, RxD2, RxD3, RxD4, RxD5, RxD6, RxD7	I	**Receive Data** In NIC applications, this bus delivers the cells from a PHY level device according to the UTOPIA Interface specification. RxD(7) is the most significant bit. (It is the first bit received on an ATM serial interface.) In ATM Switch/Hub applications, this bus carries the cells from the switch (TxD bus according to UTOPIA nomenclature).
162	RxPTY	I	**Receive Parity** Optional parity bit for RxD(7:0) bus. If used, it delivers a bit value to complement the bits on RxD(7:0) to an odd parity. Checking of parity via the SARE may be disabled by software.
163	RxSOC	I	**Receive Start of Cell** This signal is set to "high" when the first octet of a cell is present on RxD(7:0) and "low" otherwise.
150	RxCLAV	I	**Receive Cell Available** When the SARE operates as a UTOPIA Interface Master (NIC applications). It is influenced by two events. If the SARE enables cell transfer by asserting $\overline{\text{RxENB}}$ and cells are available, RxCLAV is asserted and transfer of cells on RxD(7:0) is started. If no valid cells are available, the RxCLAV is deactivated by the PHY layer device. **Transmit Enable** ($\overline{\text{TxENB}}$, according to UTOPIA nomenclature) When the SARE operates as a UTOPIA Interface Slave (in ATM Switch/Hub applications).

Table 2 UTOPIA Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
153	$\overline{\text{RxENB}}$	O	**Receive Enable** When the SARE operates as a UTOPIA Interface Master (NIC applications). This signal is deasserted by the SARE if it is not ready to accept more data. In this case the other device should stop data transfer on RxD(7:0) after a delay of one clock cycle. **Transmit Cell Space Available** (TxCLAV, according to UTOPIA nomenclature) When the SARE operates as a UTOPIA Interface Slave (in ATM Switch/Hub applications).
178, 177, 176, 175, 174, 173, 172, 171	TxD0, TxD1, TxD2, TxD3, TxD4, TxD5, TxD6, TxD7	O	**Transmit Data** In NIC applications, this bus delivers the cells from SARE to a PHY level device according to the UTOPIA interface specification. TxD(7) is the most significant bit. (It is the first bit transmitted on an ATM serial interface.) In ATM Switch/Hub applications, this bus carries the cells towards the switch (RxD bus according to UTOPIA nomenclature).
181	TxPTY	O	**Transmit Parity** Parity bit for TxD(7:0) bus. It delivers a bit value to complement the bits on TxD(7:0) to an odd parity.
182	TxSOC	O	**Transmit Start of Cell** This signal is set to "high" when the first octet of a cell is present on TxD(7:0) and "low" otherwise.

Table 2 UTOPIA Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
183	TxCLAV	I	**Transmit Cell Space Available** When the SARE operates as a UTOPIA Interface Master (NIC applications). In cell level handshake this signal is activated by the device that receives TxD(7:0) if there is enough storage space for a whole cell. Then the SARE transfers a whole cell in 53 cycles without a break. If the receiving device cannot accept a subsequent cell, it should deassert TxCLAV at the latest 4 cycles before the end of the cell (with the clock edge that outputs the 44th octet). Otherwise the SARE is allowed to send the next cell without pause (back-to-back cell transmission). **Receive Enable** ($\overline{\text{RxENB}}$, according to UTOPIA nomenclature) When the SARE operates as a UTOPIA Interface Slave (in ATM Switch/Hub applications).
186	$\overline{\text{TxENB}}$	O	**Transmit Enable** When the SARE operates as a UTOPIA Interface Master (NIC applications). It is asserted if valid data is transferred on TxD(7:0). **Receive Cell Available** (RxCLAV, according to UTOPIA nomenclature) When the SARE operates as a UTOPIA Interface Slave (in ATM Switch/Hub applications).
187	M/S	I	**Master/Slave** When M/S is "low", the SARE operates as a UTOPIA Interface Master (NIC applications). When M/S is "high", the SARE operates as a UTOPIA interface Slave (in ATM Switch/Hub applications).

1.4.3 Local Memory/Control Interface

The Local Memory Interface is used by the SARE to store and retrieve descriptors and context data of the currently active virtual connections when their number exceeds the internal storage capacity of the SARE (typically, 60 links in segmentation and reassembly). The SARE operates as the bus master.

The Local Memory Interface can also be used to control peripheral devices via the PCI Interface, such as PHY devices. This allows for the implementation of NIC applications without a local controller on the NIC.

See **table 3** for Local Memory and Control Interface pin descriptions.

Table 3 Local Memory and Control Interface

Pin No.	Symbol	Input (I) Output (O)	Function
95, 94, 93, 92, 91, 88, 87, 86, 85, 84, 83, 80, 79, 78, 77, 74, 73, 72, 71, 70	LA0, LA1, LA2, LA3, LA4, LA5, LA6, LA7, LA8, LA9, LA10, LA11, LA12, LA13, LA14, LA15, LA16, LA17, LA18, LA19	O	**Local Address Bus**
141, 140, 139, 138, 137, 136, 133, 132, 128, 127, 126, 125, 124, 123, 122, 121, 116, 115, 114, 113, 112, 111, 110, 109, 105, 104, 103, 102, 101, 98, 97, 96	LD0, LD1, LD2, LD3, LD4, LD5, LD6, LD7, LD8, LD9, LD10, LD11, LD12, LD13, LD14, LD15, LD16, LD17, LD18, LD19, LD20, LD21, LD22, LD23, LD24, LD25, LD26, LD27, LD28, LD29, LD30, LD31	I/O	**Local Data Bus**

Table 3 Local Memory and Control Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
142, 129, 119, 106	LPAR0, LPAR1, LPAR2, LPAR3	I/O	**Local Bus Parity** Optional parity bits for the octets on LD(31:24), LD(23:16), LD(15:8) and LD(7:0), respectively. Parity can be even or odd (programmable).
143, 120	$\overline{\text{LWE0}}$, $\overline{\text{LWE1}}$	O	**Local Bus Write Enable** Write enable signals for the half-words on LD(31:16), and LD(15:0), respectively. If the lower half of the bus, LD(15:0), for example, is used to control a peripheral device (like a PHY device), the corresponding write signal $\overline{\text{LWE0}}$ should be connected to the $\overline{\text{WE}}$ input of the PHY device.
144	$\overline{\text{LRE}}$	O	**Local Bus Read Enable** To be connected to the Read Enable (or $\overline{\text{OE}}$) input of a local memory and/ or of a peripheral device (if present).
69	$\overline{\text{LCS}}$	O	**Local Bus Chip Select**
146	$\overline{\text{PCS}}$	O	**Peripheral Bus Chip Select**
145	PREADY	I	**Peripheral Bus Ready** A peripheral device indicates the end of a read or write transaction by setting this signal "high".
147	$\overline{\text{PINT}}$	I	**Peripheral Interrupt** A peripheral device may activate this signal to indicate a special event or active interrupt status. It is an active low signal.

1.4.4 Test Interface

See **table 4** for the SARE Test Interface pin descriptions.

Table 4 Test Interface

Pin No.	Symbol	Input (I) Output (O)	Function
192	TM	I	**Test Mode** When TM is low, this sets the SARE in a mode that allows the on-chip memories to be tested.
188	TCLK	I	**Test Clock** For boundary scan according to IEEE Std. 1149.1.
189	TMS	I	**Test Mode Select** For boundary scan according to IEEE Std. 1149.1.
190	TDI	I	**Test Data Input** For boundary scan according to IEEE Std. 1149.1.
191	TDO	O	**Test Data Output** For boundary scan according to IEEE Std. 1149.1.

1.4.5 Power Supply Interface

See **table 5** for the Power Supply Interface pin descriptions.

Table 5 Power Supply Interface

Pin No.	Symbol	Input (I) Output (O)	Function
2, 8, 14, 20, 22, 28, 38, 42, 46, 52, 58, 64, 66, 76, 82, 90, 100, 108, 118, 131, 135, 149, 152, 166, 169, 180, 185, 196, 202, 206	V_{SS}	I	**Ground**

Table 5 Power Supply Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
18, 36, 56, 208	V_{DD5}	I	**Positive Power Supply Voltage** 4.75-5.25 V
6, 12, 21, 26, 41, 44, 50, 62, 65, 75, 81, 89, 99, 107, 117, 130, 134, 148, 151, 165, 168, 179, 184, 194, 200, 205	V_{DD3}	I	**Positive Power Supply Voltage** 3.135-3.465 V

1.5 Functional Block Diagram

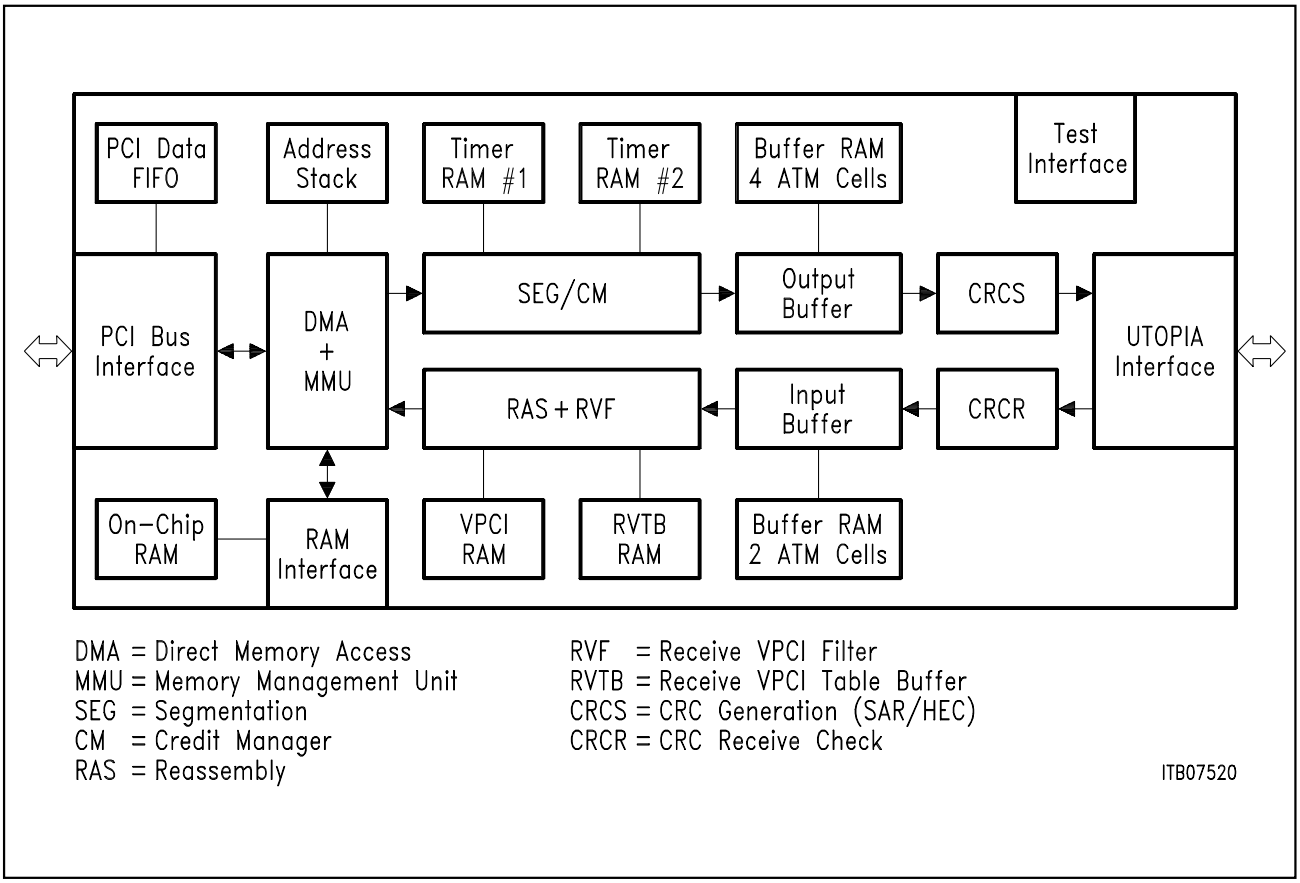


Figure 3 SARE Block Diagram

1.6 System Integration

The SARE has been designed with the two following objectives:

1. Offer very low-cost solutions – important for ATM attachment cards and for ATM interfaces on PC motherboards.
2. Offer high performance in cases where that performance is needed – for example in applications for ATM servers, gateways or switches).

The first objective is realized through the very high level of integration of the SARE. The on-chip memory and the interfaces allow a seamless integration of the SARE with the rest of the system without external components.

– Through its on-chip DMA controller, the SARE is able to transfer ATM cell payload directly to and from the central system memory, e.g. on a motherboard, via its PCI bus, without necessity for glue logic.
– No local microprocessor is needed to control the SARE, since it can be controlled via the PCI. Typically, at least 60 simultaneous receive and transmit links can be processed without external memory.
– Through an auxiliary parallel interface on the SARE, control of other devices (e.g. PHY device) on the NIC can be effected via the PCI Interface, thus allowing NIC system integration without a local controller.

These features are especially useful for implementing so-called passive NICs and inexpensive ATM attachments on motherboards, e.g. of Multimedia PCs.

The second objective above is realized via the architectural choices made in the SARE. High performance is required, for example, in ATM routers where a large number of virtual connections are to be simultaneously processed.

– For such applications, an external RAM can be connected to the SARE to serve as a local memory extension for storing buffer descriptors, segmentation timers and other context data for a large number of simultaneously active virtual connections (maximum 64k virtual connections).
– In the most common system configuration, ATM cell payload is transferred via the on-chip DMA controller directly to and from the central system memory (to minimize total transmission delay) whereas connection data, descriptors etc. are held in local memory. This optimizes the use of the central resources (system memory and system bus).

1.6.1 ATM Network Interface Cards

One typical application for the SARE is in the design of Network Interface Cards (NICs) for attachment of workstations and PCs to an ATM local area or public network.

As shown in **figure 4**, the SDHT (SONET/SDH Transceiver for 155 Mbit/s) can be used to implement the PHY functions for 155 Mbit/s STM-1/STS-3c/OC-3.

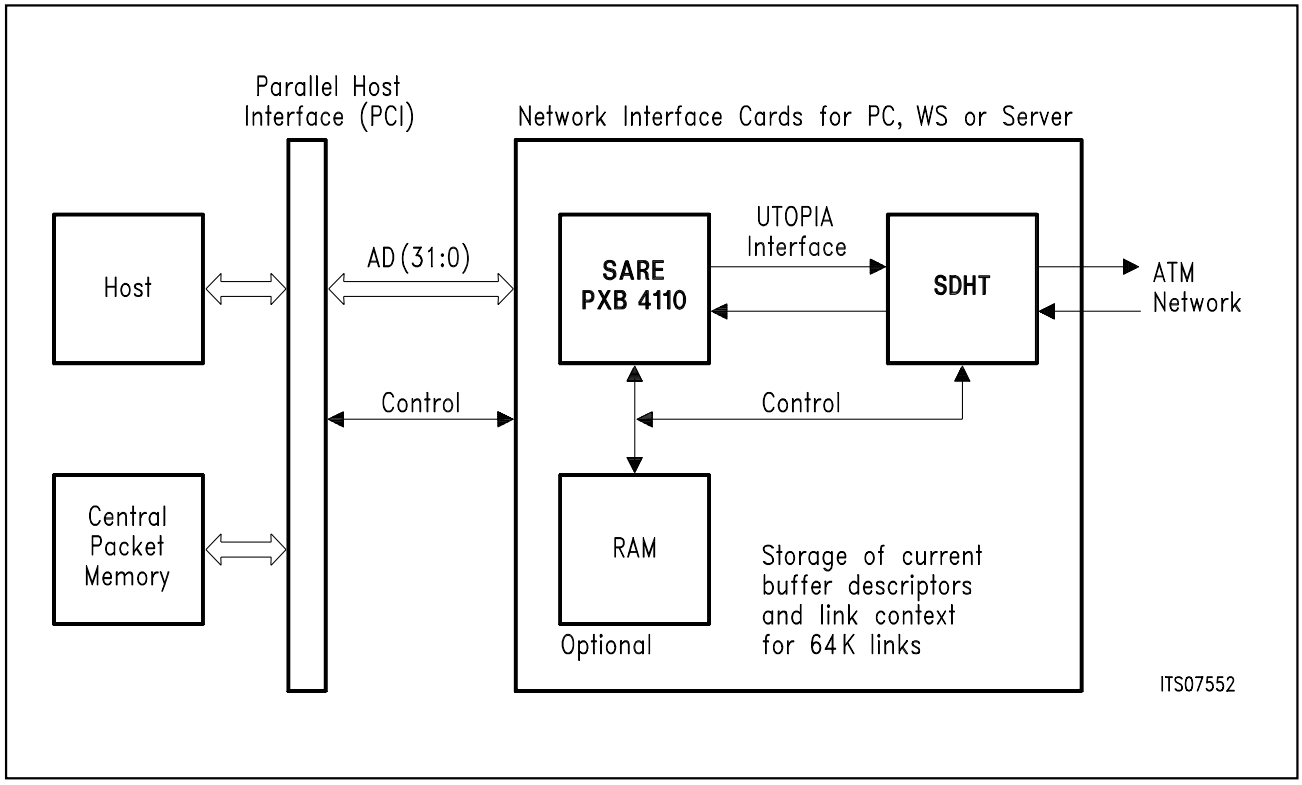


Figure 4 Network Interface Card (NIC) Architecture

1.6.2 Applications in ATM Switches/Hubs

The other application for the SARE is in ATM Switches or Hubs.

The SARE can be directly connected to an ATM Switching Preprocessor (ASP) via its UTOPIA interface. This is possible due to the fact that the UTOPIA interface can be configured such that the SARE operates as a UTOPIA Interface Clock Slave (and not as Master), in this case with respect to the ASP. In an important class of applications, the SARE is used in an ATM switch or LAN hub to terminate all VCs that are needed for signalling, multicasting and OAM functions, controlled by an attached microprocessor that operates as a Group Controller.

2 Functional Description

2.1 Block Diagram

Figure 5 shows the main functional blocks of the SARE.

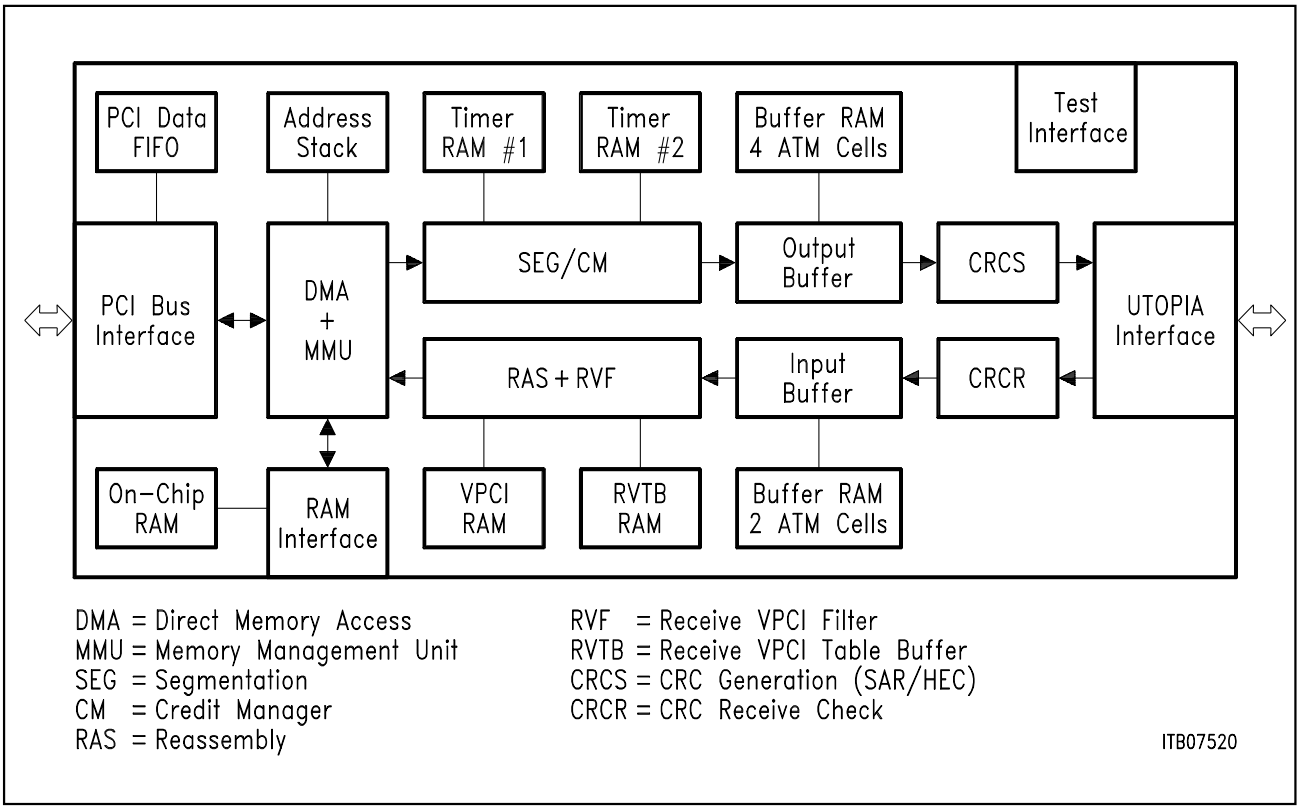


Figure 5 SARE Function Block Diagram

2.2 Interfaces

2.2.1 UTOPIA Interface

The SARE contains a universal UTOPIA Interface module. This interface has the following features:

- Compatible to 8-bit Level 1 UTOPIA standard
- Master (NIC applications) or Slave (Hub/Switch applications) mode for clocking
- Operates in cell level handshake only
- Operates up to 25 MHz (33 MHz)
- Parity generation and optional check
- Status reporting

The interface consists of 28 lines.

The maximum clock frequency of 33 MHz results from the fact that in the case where the SARE operates in master mode the clock can be optionally generated from the PCI

master clock (via a programmable division factor of 1, 2, 4, 8 or 16), instead of deriving the transmit clock from the transmit clock input.

Note: If the PCI clock is directly used to clock the UTOPIA Interface (i.e. division factor 1), it cannot be guaranteed that the SARE will have no cell loss at all times under full load conditions on UTOPIA (when no gaps or idle cells occur in the payload). Therefore care should be exercised in this particular case.

After power-up, the UTOPIA Interface is inactive and has to be enabled via a special configuration bit.

2.2.2 System Bus Interface

The System Bus Interface of the SARE is compliant with the industry standard PCI interface. The interface can be adapted to other bus systems via external glue logic. This interface has the following features:

- Master and Slave capability
- 32-bit address/data bus
- Programmable Master Latency timer
- Two interrupt lines
- The interface consists of a total of 51 lines.

2.2.3 Local Memory Interface

For the case where the total number of simultaneously active virtual channels in segmentation/reassembly is such that the on-chip memory of the SARE cannot accommodate all the necessary context data (typically more than 60 full duplex simultaneously active connections), the Local Memory Interface is used by the SARE for storing context information, buffer descriptors and other data necessary for the segmentation/reassembly process. This interface has the following features:

- 20-bit address and 32-bit data buses
- Parity lines for each data bus octet
- The interface consists of 60 lines.

With additional 3 lines ($\overline{\text{PCS}}$, PREADY, $\overline{\text{PINT}}$) the local memory interface address (8 LSB lines) and data bus (16 lines) can be used to control peripheral devices (e.g. a PHY device) via the PCI Interface without the necessity of a local controller.

2.2.4 Boundary Scan and Test Interface

This interface complies with the IEEE Standard 1149.1 Test Access Port and Boundary Scan Architecture. The Test Access Port (TAP) allows boundary scan to be used for testing of the SARE and the board on which it is installed. The TAP is comprised of four pins that are used to interface serially with the TAP controller within the SARE.

The SARE can be set to a special test mode (for testing on-chip memories) via the Test Mode (TM) pin.

2.2.4.1 Additional Test Capabilities

In addition, the whole chip can be tested via special test configuration bits that loop back data from the transmit to the receive lines. In this case one (receiver or transmitter) operates as UTOPIA Master and the other as UTOPIA Slave, according to **table 6**.

Table 6 Loopback and UTOPIA Mode Descriptions

Loopback	UTOPIA	Description
0	0	Normal mode, UTOPIA is Master
0	1	Normal mode, UTOPIA is Slave
1	0	Test mode; Transmitter is Master, Receiver is Slave
1	1	Test mode; Transmitter is Slave, Receiver is Master

The SARE also includes a built-in self-test (BIST) which can be activated via a bit in the Configuration Register. This built-in self-test consists of writing patterns in an external memory via the PCI Interface using the on-chip DMA controller. BIST can be exploited on the system level by implementing the check of the correctness of the patterns via a host. The Boundary Scan Interface, the UTOPIA test loop, the on-chip memory test mode, and the BIST in combination provide extensive diagnostic capabilities of the SARE in a real system environment.

2.3 Functional Overview

2.3.1 Functions in Receive Direction

From the UTOPIA Interface the cells are input bytewise to the CRC Receive Check module (CRCR) where the HEC and possibly CRC-10 checks are performed. From here the cells are input to the Input Buffer via an 8-bit bus. This includes cells with HEC error or CRC-10 error (cells with CRC-10 error are only marked as such, the subsequent Reassembly Unit (RAS) is responsible for any further actions).

The Input Buffer filters out idle and unassigned cells (if any) and collects statistics. Further, it performs the separation of cell header and cell payload. The cell header is separately output to the Receive VPCI Filter module on a 32-bit bus. The Input Buffer is able to store two cells (24×32-bit buffer), thereby performing the adaptation, if need be, of the UTOPIA and PCI bus interface clocks in the system. The payload of the cells is output to the Reassembly Unit via a 32-bit bus.

The Receive VPCI Filter checks to which of 32 predetermined groups of virtual links the received cell belongs, if any. The RAS includes a state machine to control the reassembly process, intermediate storage buffer where the payload from the Input Buffer section is stored, and a CRC-32 check module. In the case that the cell belongs to one of the 32 groups of virtual links, the cell header is forwarded by the RVF to the

RAS state machine via a 32-bit bus and the VPCI address on a separate 24-bit bus. In the case where the cell does not belong to one of the groups, it is discarded.

From the information received from the RVF, the RAS control block checks whether the received cell belongs to an active (open) connection. Cells not belonging to an open connection are discarded. In the case where the cell belongs to an open connection, it is reassembled. In the usual case, the payload of cells being reassembled is stored directly in the central memory via the PCI Interface by the on-chip DMA controller (thus avoiding the overhead of copying assembled packets from the local memory in the case where the reassembly is performed in the local memory).

To this effect, the DMA controller contains an address FIFO (16 32-bit words) and a data FIFO. A new DMA transfer command is issued by the RAS to the DMA controller by writing a start address in the address FIFO and consecutive data words in the data FIFO. The data FIFO size of the DMA controller (128 32-bit words) implies that up to around 10 cells can be stored simultaneously internally before an overflow occurs, thus allowing for possibly long PCI latency times. When the data FIFO reaches a certain filling threshold (96 words), the DMA controller outputs a control signal to the UTOPIA interface module to prevent further cells from being accepted (translated into $\overline{\text{RxENB}}$ being deactivated). This applies to the Master mode. In the Slave mode, the cell is completely received, after which the pin $\overline{\text{RxENB}}$ (which in this case has the significance of Transmit Cell Available) is deasserted.

In addition to performing transfer of cell payload in reassembly, the DMA controller also performs transfer of other data to and from the central memory (e.g. new buffer descriptors) or the local memory (e.g. link context data in the case of many simultaneously active links) as instructed by the RAS.

2.3.2 Functions in Transmit Direction

When a packet (Protocol Data Unit, PDU) is ready for segmentation, this is indicated via an entry in the Transmit Waiting Queue (TWQ). In the general case an entry in the TWQ contains a pointer to a Transmit Virtual Channel descriptor in the Transmit VPCI Table (TVT) and a pointer to a PDU descriptor in the Transmit PDU Table (TPT). The TWQ is polled every cell cycle by the Credit Manager (CM) which is responsible for incorporating a new packet into one of eight segmentation queues of the PDUs currently in segmentation. Each segmentation queue corresponds to a particular bit rate at which the PDUs in that queue are to be segmented. Two timers (peak cell rate and average cell rate timers) attached to the eight queues are dynamically managed so that the rate of segmentation in each traffic class (segmentation queue) of every PDU is independent of the number of PDUs in that class. Based on these timers the CM determines, once per cell cycle, the PDU from which the next cell is to be segmented and issues this information to the Segmentation Unit (SEG).

The SEG fetches the context information pertaining to that PDU from the Transmit PDU Table (TPT). The descriptor contains, among other information, the current address in

the current data buffer from which the next cell for that PDU is to be read, and programs the DMA controller accordingly with this value as start address. The updated context for that PDU is subsequently written by the SEG into the TPT. (For typically 60 simultaneously active links, the TPT is on-chip; for simultaneously active links exceeding this number, the TPT information is located in a local memory accessed via the Local Memory Interface). This information includes the current CRC-32 value, the calculation of which the SEG is also responsible for. The payload and header of the cell are forwarded to the Output Buffer via a 32-bit bus (13 double words per cell).

The Output Buffer contains a buffer of size 52 x 32 bits for four complete cells. This block is responsible for the generation of idle cells if no cell from SEG is available. The idle cell format is programmable (via registers included in the Output Buffer). Optionally, the idle cell generation can be disabled, in the case where the generation is left to the PHY device. The Output Buffer also constitutes the border between the PCI bus interface clock and UTOPIA Interface clock systems. The cells are forwarded to the CRC Generation module (CRCS) via an eight-bit bus (52 bytes per cell, i.e. excluding HEC).

The CRCS calculates the HEC and (for cells corresponding to AAL3/4 and SMDS, and for OAM cells) CRC-10 error check sequences and outputs the complete 53-byte cell to the UTOPIA interface module via an 8-bit bus.

3 Receive Data Structures

3.1 Receive Procedure Summary

For each PDU in reception, the type of the PDU is individually programmable. The following types of PDU are implemented by the SARE (this information is contained in the Receive VPCI Table, which is initialized by the CPU and described in more detail below):

Table 7 Receive Connection Types Implemented in the SARE

Type of Connection	Characteristics/Applications
No connection	VC not active
AAL5	No timing relation required, Variable Bit Rate, connection oriented e.g. data communication
AAL3/4; With or without CRC-32 With or without multiplexing of PDUs	No timing relation required, Variable Bit Rate, connection oriented/connectionless e.g. SMDS
Transparent	Like AAL5, but Single Segment Messages OAM
Cell FIFO Mode	Timing relation required Constant Bit Rate, audio, video (AAL0/1) OAM

3.1.1 Receive Procedure Example

In the case of AAL5, the receive procedure is summarized in **figure 6**.

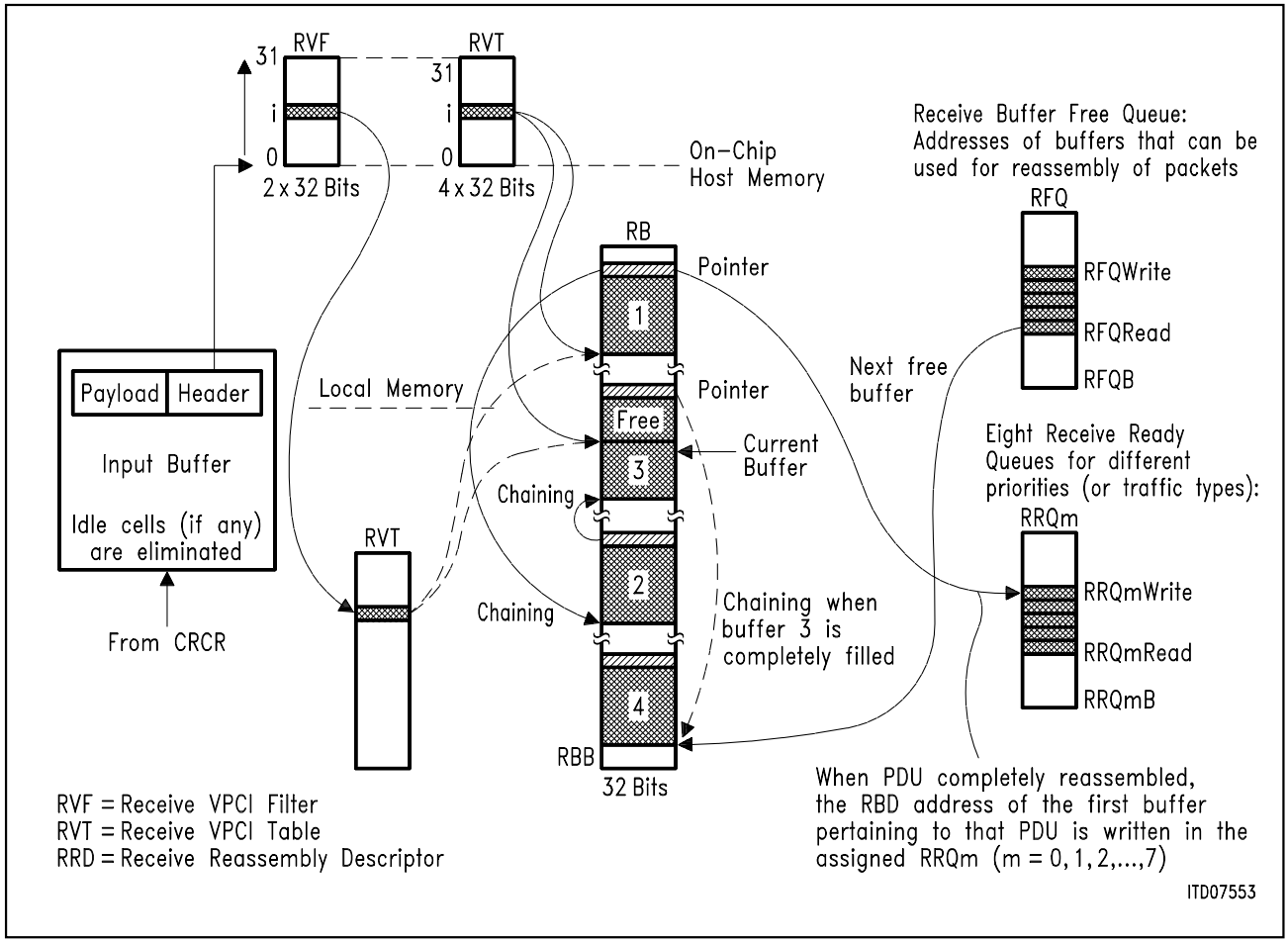


Figure 6 AAL5 Receive Procedure

When a valid cell is received, the SARE first performs a fetch of the context associated with the corresponding virtual connection (identified by the VPI and VCI values in the header of the received cell) from its on-chip memory or from off-chip local memory (for a number of simultaneous connections typically exceeding 60). This context is located in the Receive VPCI Table (RVT). The RVT contains a pointer to the buffer where the payload for the cell in the host memory is to be stored. Transfer of the payload is done by the on-chip DMA controller via the PCI bus, after which the updated context for that connection is stored again in the on-chip memory (or off-chip local memory, as the case may be). The data buffers are organized in a linked list chain structure, and have a programmable length between 64 and 8192 bytes (in powers of 2). Every time a buffer becomes full and the PDU has not been completely received, the address for the next buffer is fetched from a Receive Buffer Free Queue (RFQ) located (usually) in the local memory. Thus, the CPU has to ensure that there is always empty buffer addresses in the RFQ in order that a data overflow does not occur. The RFQ is common to all virtual connections.

When a PDU has been completely received, the status of the PDU is stored in the first three words of the first buffer associated with that PDU, an entry in one of the eight Receive Ready Queues is performed, and, if programmed, a maskable interrupt status is generated to indicate that a PDU has been correctly received.

Many special features, including:

– Streaming mode/early warning reception (indication of the beginning for a received PDU after the reception of the first cell)
– Handling of Constant Bit Rate traffic via the Cell FIFO mode
– Handling of virtual connections in the AAL3/4 PDU multiplexing protocol (PDUs with Multiplexing Identification, MID) mode
– Transparent mode (e.g. for OAM cells)

are not accounted for by the above description, and will be explicated in the course of the more detailed description in the following paragraphs.

3.2 General Description of Receive Data Structures (AAL5)

3.2.1 Initialization of the Reassembly for a PDU

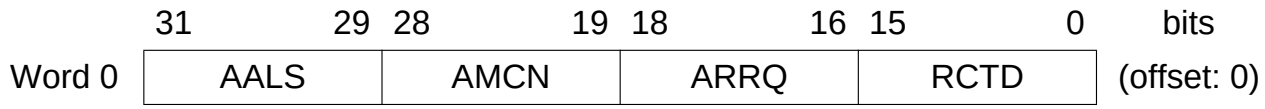
In order that cells in a given virtual connection can be reassembled into a PDU, it is only required that the host has:

1. initialized a Receive VPCI Table (RVT) entry that defines the characteristics of that virtual connection, and
2. activated that virtual connection by defining the corresponding VPI/VCI combination as a valid one in the Receive VPCI Filter (RVF), with a pointer to the properly initialized RVT.

The RVT is initialized by the host in the on-chip or the external local memory. The Receive VPCI Filter is resident on the SARE.

3.2.2 Initialization of the Receive VPCI Table (RVT)

The Receive VPCI Table (RVT) is initialized by the host with the following parameters that specify the characteristics of a Virtual Circuit:



AALS AAL Select

This field determines the manner in which the reassembly of the cells is to be performed in this virtual connection, in accordance with the table below:

AALS	Description	Type of Connection
000	NONE	No connection
001	TRANSPARENT	Transparent (PDU = like AAL5, but Single Segment Message)
010	CELL_MODE	Cell FIFO Mode
011	AALTYPE5	AAL5
100	AALTYPE34	AAL3/4, no CRC-32, no Multiplexing Identification
101	AALTYPE34M	AAL3/4, no CRC-32, Multiplexing Identification
110	AALTYPE34C	AAL3/4, CRC-32 evaluation, no Multiplexing Identification
111	AALTYPE34CM	AAL3/4, CRC-32 evaluation, Multiplexing Identification (SMDS)

AMCN Active MID Count

Count in this virtual connection (if applicable) currently in reassembly. This value is initialized to 0 by the host and updated by the SARE thereafter.

ARRQ Assigned Receive Ready Queue

Determines in which queue the completion of the reassembly of a PDU in the connection is to be reported.

RCTD Receive Context Descriptor

Contains a Receive MID-Table Descriptor (RMD) if reassembly with Multiplexing Identifiers (MID) is to be performed (AALS = 101 or 111). If reassembly without MID is to be performed, this field contains a Receive Buffer Descriptor (RBD), which gives the address of the first linked buffer for the PDU in reassembly. RCTD = 0 means that the corresponding virtual connection has no PDU currently in reassembly. The host is expected to initialize RCTD to 0.

3.2.3 Receive VPCI Filter (RVF)

The incoming cell VPCI is checked by the RVF to determine the VPCI group to which the cell belongs. Each group has associated on-chip registers VPCI and RVTB that have the following formats.

VPCI

Entry i (i = 0,...,31):

	8	8	16	# of bits
VPCI	VPILO	VPIHI	VCILO	

RVTB

Entry i (i = 0,...,31):

	3	1	4	24	# of bits
RVTB	Rsvd	OnChip	VCIRANGE	RVTB	

If VCIRANGE = 15, this implies that the corresponding entry is not valid (virtual connection or group not active). In the following it is assumed that VCIRANGE is different from 15.

If OnChip is set to 1, this means that the RVT entry for the corresponding virtual connection is in an on-chip RAM (starting at internal byte address $16 \times i$ for i = 0,...,31). In this case VPI and VCI have to match VPILO (= VPIHI) and VCILO (VCIRANGE = 0), respectively. This means that the parameters for at least 32 simultaneous virtual connections may be located on-chip, each virtual connection being identified by a random VPI, VCI combination independent of the other virtual connection identifiers.

Next, consider the case where OnChip is set to 0. In order to belong to VPCI group i (i = 0,...,31), the values of VPI, VCI in the header of the incoming cell have to satisfy the condition:

$$VPIHI \geq VPI \geq VPILO$$

$$VCILO + 2^{VCIRANGE} > VCI \geq VCILO \qquad \text{for one i (i = 0,...,31).}$$

When this condition is satisfied, RVTB gives the base address for the corresponding RVT.

The byte address of the RVT entry for the connection inside this group is given by:

$$RVT\text{-}Address = 256 \times RVTB + 2^{VCIRANGE+4} \times (VPI\text{-}VPILO) + 16 \times (VCI\text{-}VCILO).$$

Depending on the value yielded by this expression, the RVT entry may be located either on-chip or off-chip. Also see **figure 7**.

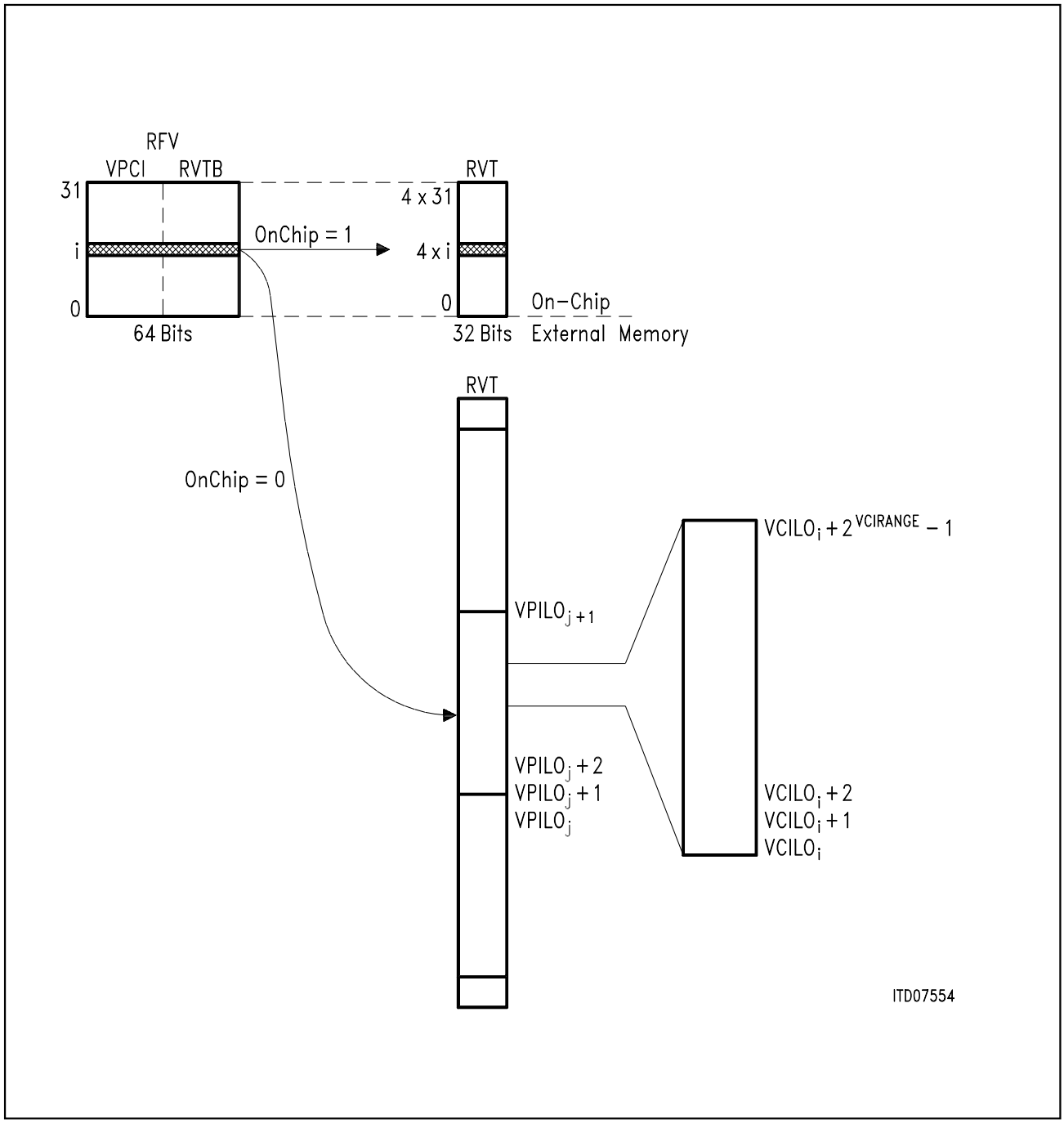
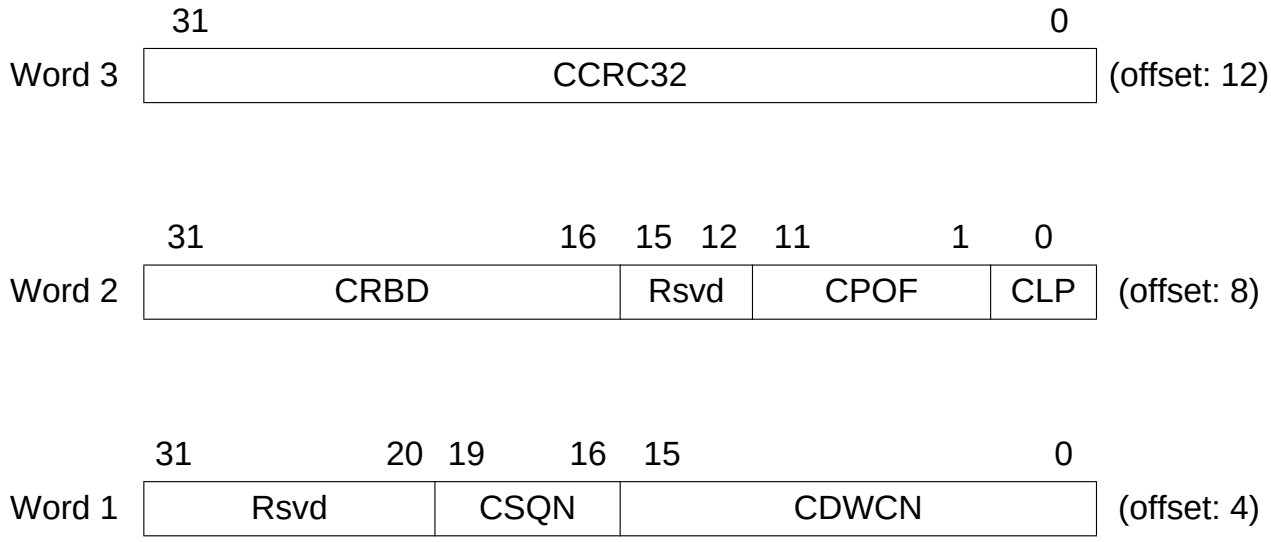


Figure 7 SARE Receiver Data Structure

3.2.4 Context Information During Reassembly

When the received cell is not the first cell of a PDU, the context information for that virtual connection is retrieved from the RVT (on-chip, or in external local memory, as the case may be). This context information is comprised of three 32-bit words, and has the following format:

Context Information



Word 0 of the RVT entry was discussed in **section 3.2.2**.

Note: For AAL3/4 type connections (with Multiplexing of PDUs), this information is stored in the Receive MID Table RMT instead of in the RVT.

CCRC32 **Current value of the CRC-32 checksum**

CRBD **Current value of the RBD**

CPOF **Current Payload Offset**

Current offset (in 32-bit words) of the payload in the current buffer

CLP **Cell Loss Priority**

OR'd among the CLP fields of the received cells

CSQN **Current Sequence Number**

For AAL3/4 only

CDWCN **Number of received 32-bit words**

After the reassembly of the current cell, the updated context information is stored back into the RVT.

3.2.5 Receive Buffer

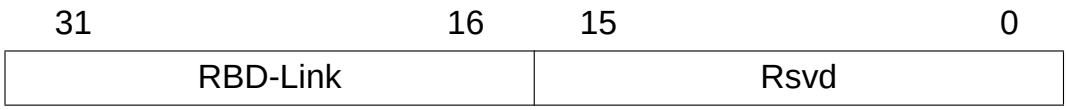
The payloads of cells are stored in buffers of programmable length. Linked list chaining is performed on the receive data. The SARE Reassembly Unit fetches the address of a free buffer from the Receive Free Queue (RFQ) and maintains this buffer ready for any PDU that fills its current buffer. The buffer length is a system parameter programmed via BUFSIZE, common for both segmentation and reassembly. **Table 8** shows the buffer size options available.

Table 8 Available Buffer Sizes

Value of BUFSIZE	Buffer Size (bytes), bufsiz
000	64
001	128
010	256
011	512
100	1024
101	2048
110	4096
111	8192

The chaining of the consecutive buffers for a given PDU is effected by the SARE by appending a Receive Chain Buffer Trailer (RCBT) in the last 4 bytes of the buffer. This word has the following format:

RCBT



RBD-Link Receive Buffer Descriptor-Link

Pointer to the next buffer. The effective byte address of the beginning of the buffer is then given by

$$\text{RB-Address} = 256 \times \text{RBB} + \text{bufsiz} \times \text{RBD-Link},$$

where RBB is a 24-word Receive Buffer Base address (RBB), a configurable system parameter.

Rsvd Reserved

During reassembly, the current byte address inside the current buffer is determined by the CPOF parameter in the RVT (or RMT, as described above):

Current-Address in RB = 256 × RBB + bufsiz × RBD + 4 × CPOF.

When a buffer becomes full, a pointer of a free buffer (RBD) is fetched from the RFQ and written into the last double word of the buffer. Empty (unused) buffers are characterized by the fact that the last double word in the buffer is “all 0s”: RBD-Link = 0.

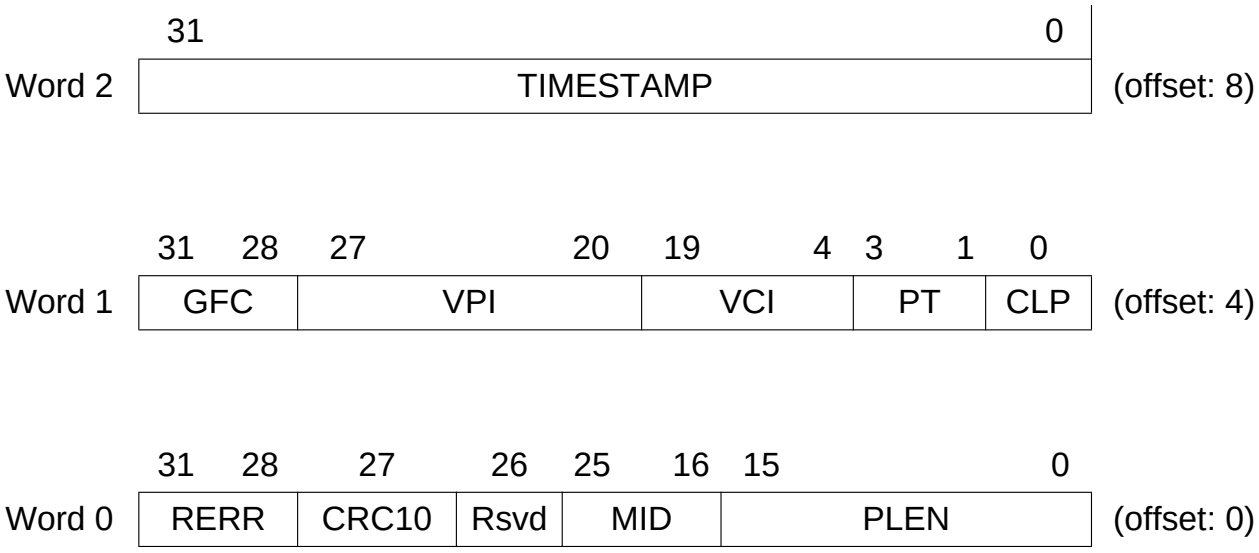
Buffers, the pointer of each is obtained from the RFQ, are checked for this condition. If the condition does not hold true, an indication is given to the host via interrupt and the PDU in reassembly is aborted with an error message.

3.2.6 Receive Ready Queues (RRQ)

When the last part of a PDU has been reassembled, an indication is issued to the host via one of eight Receive Ready Queues (RRQm, m = 0,...,7). This consists of the RBD value of the first buffer for that PDU. The queue where the indication is to be issued is determined by the ARRQ parameter in the RVT (See above).

Simultaneously, a Receive Reassembly Header (RRH) is entered in the first three 32-bit words of the first buffer of the PDU. This header enables the host to read the information pertaining to the virtual connection, and to check the status of the PDU. It has the following format:

RRH



TIMESTAMP 32-Bit Timestamp

For further processing by the host.

GFC Generic Flow Control

Value of GFC in last received cell.

VPI **Virtual Path Identifier (UNI)**

VCI **Virtual Channel Identifier**

PT **Payload Type**

Value of PT in last received cell.

CLP **Cell Loss Priority**

OR'd among the CLP fields of the received cells.

RERR **Error code**

In case reception of PDU was aborted

CRC10 **CRC-10 check**

0 = correct; 1 = error

Rsvd **Reserved**

MID **Message Identifier**

For SMDS: Multiplexing

PLEN **Total number of received 32-bit words in PDU**

Length of the complete CPCS-PDU. In the case of AAL5, this implies the payload, the Padding bytes and the CPCS trailer (n cells $\times$ 12 double-words per cell). In the case of AAL3/4 this is the sum of the SAR LI-fields, which gives the total length of the CPCS-PDU (CPCS-PDU header, CPCS-PDU trailer, CPCS-PDU payload, plus padding bytes).

RERR **Error Codes**

See **table 9**.

Table 9 Receive Ready Queue Error Codes

RERR	Meaning
0000	No error
0001	Begin Of Message (BOM) or Single Segment Message (SSM) without End Of Message (EOM)
0010	Overflow of receive buffer
0011	Abort
0100	Incorrect CPCS-CRC-32
0101	CPCS-PDU length overflow
0110	Sequence number error
0111	Incorrect SAR length field

Table 9 Receive Ready Queue Error Codes (cont'd)

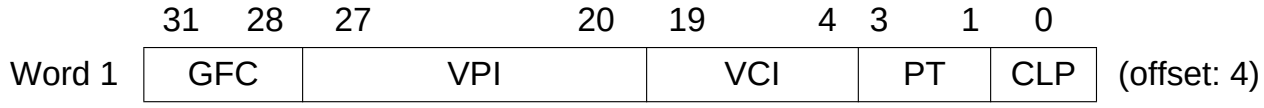
RERR	Meaning
1000	Incorrect SAR CRC-10 field
1001	Chaining error
1010 to 1111	Reserved

Note: Because of possible latencies on the PCI bus that may delay the transfer of the last cell's payload and the Receive Reassembly Header (RRH) to the host memory, it may be that a Receive Ready Queue (RRQ) interrupt status is generated before all the payload data, and thus, RRH, is transferred. The host software should initialize the start of receive buffers (where RRH is stored) to all zeros and poll the RRH location before taking any action subsequent to an RRQ interrupt status.

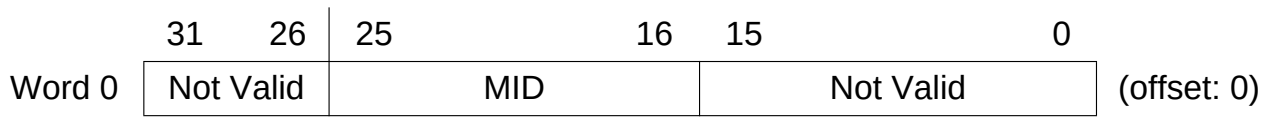
3.2.7 Streaming Mode Operation

The streaming mode receive, or "early warning receive" feature can be used, e.g. to begin processing the received PDU data as soon as the first cell of that PDU has arrived.

If this mode is programmed, the SARE stores the parameters of the first received cell of a PDU in the beginning of the Receive Buffer, in the same position as it normally stores the final Receive Reassembly Header after the arrival of a complete PDU. After that, it reports the arrival of the first cell of a PDU via interrupt in a Receive Ready Queue. The following information on the virtual connection is stored in the second 32-bit word of the receive buffer.



In addition, in the case of a virtual connection with a AAL3/4 protocol, the MID is stored in the first word:



In this mode of operation, the Receive Ready Queues 4 to 7 can be used by the SARE to issue the indication of the beginning of a PDU (after reception of the first cell), such that RRQ4 is used for indicating the beginning of reception of a PDU for a virtual connection normally programmed to use RRQ0, RRQ5 for a virtual connection which uses RRQ1, RRQ6 for a virtual connection which uses RRQ2, and RRQ7 for a virtual

connection which uses RRQ3. Thus, the number of Receive Ready Queues for completely received PDUs is effectively reduced from 8 to 4 (namely, RRQ0 to RRQ3).

Note: Even if the streaming mode has been enabled, some or any of the queues RRQ4-7 can still be available for the indication of reception of a complete PDU (and not the beginning of a PDU), since each queue has an individual "streaming mode enable" bit. Thus, if the virtual connections for which an "early warning indication" is desired are programmed on one queue (say, RRQ4), six queues (namely RRQ1-3 and RRQ5-7) are available for normal "complete PDU" indication for other virtual connections.

When the host receives an early warning of the reception of a PDU, it can start processing on the data as soon as the indication is received via a Receive Ready Queue. Except for the payload of the first cell, it may be that the first buffer is not completely filled with valid data when the early warning indication is received, however. To enable the host to know whether a buffer is completely filled, it may enable the generation of an interrupt status at the completion of every buffer. This interrupt status is also generated in one of the queues RRQ4-7: in each of these queues, the host can enable generation of interrupt status after the reception of the first cell and/or after the completion of a receive buffer, or both. For details, see **chapter 6** Register Descriptions.

3.3 Other Modes of Reception

The description above used as example the reassembly in the case of a virtual connection according to the ATM Adaptation Layer 5 (see example above). Besides AAL5, the three other types of protocols will be briefly described in the following, with regard to the most important differences to AAL5.

3.3.1 Reassembly on Virtual Connections According to AAL3/4

In the case of AAL3/4, PDUs may be reassembled either without or with MID (Multiplexing Identifier), as determined by the global system parameter C_MIDMUX (i.e. 0: no MID; 1: MID), set by the host.

In the case of reassembly without MID, RCTD = 0 in the first RVT word means that the cell must be a Begin Of Message (BOM) or a Single Segment message (SSM). If this is not the case, the cell is ignored. If a cell representing a BOM or SSM is received but RCTD<>0, the reassembly of the "old" PDU is interrupted with an error indication, and reassembly of a new PDU is started.

In the case of reassembly using MID, the Receive Context Descriptor (RCTD) in the RVT does not contain an RBD to the receive buffer, but a Receive MID-Table Descriptor (RMD).

RVT

	31 29	28 19	18 16	15 0
Word 0	AALS	AMCN	ARRQ	RMD

This points to an auxiliary data structure, the Receive MID Table (RMT) that contains the RBD values for the different PDUs in reassembly for that MID connection. See **figure 8**.

RMD **Free RMDs**

The RMDs are fetched by the SARE from a Receive MID Queue (RMQ) which contains the addresses of free RMDs. The number of significant bits in the MIDs is given by a system parameter SMID (Size of MID) which simultaneously gives the size of the section in the RMT per virtual connection.

AMCN **Active MID Count**

Determines the number of active MID counts in the corresponding virtual connection. If AMCN > 0, the RCTD is the RMD pointer in the RMT. If AMCN = 0 (in which case RCTD = 0 also), the SARE fetches a new RMD from the Receive MID Queue.

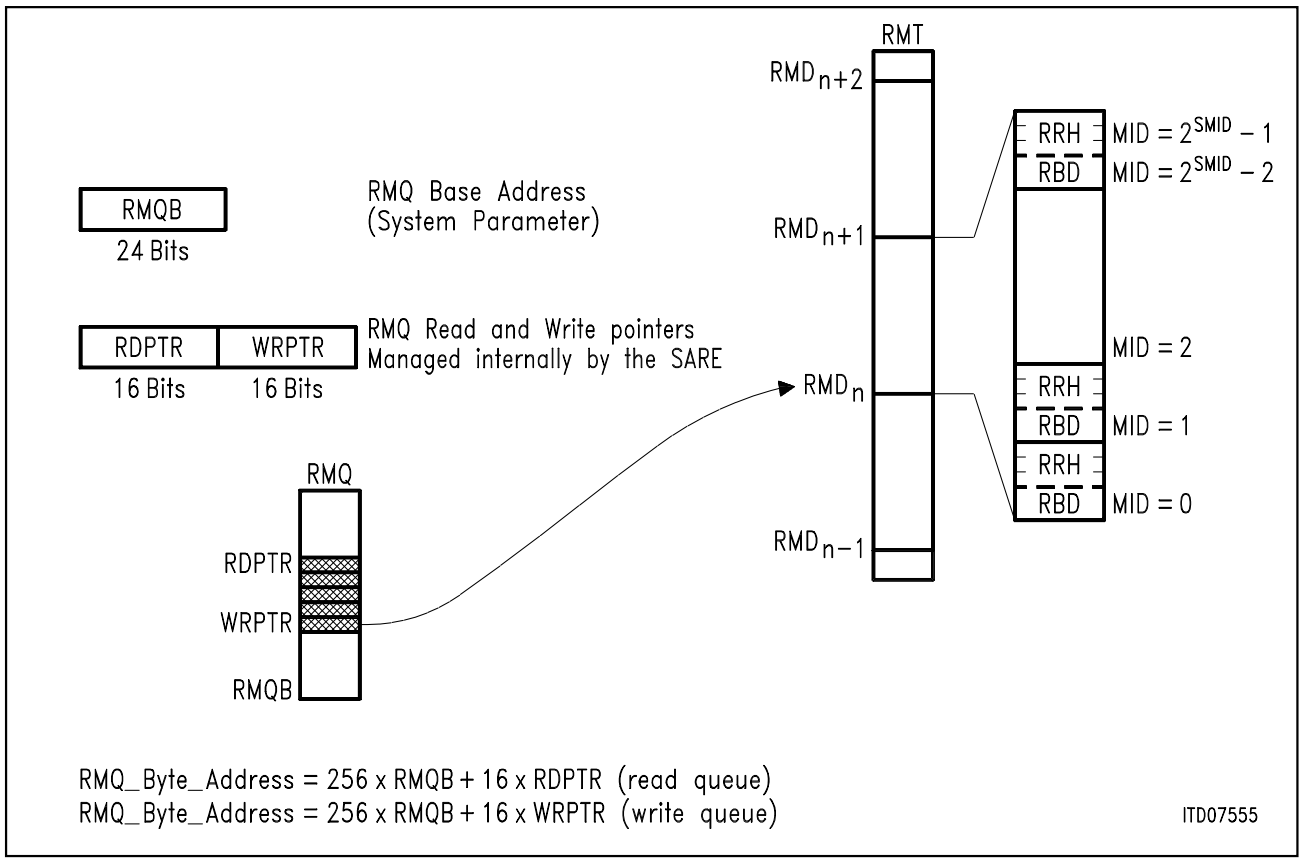


Figure 8 Receive MID Table

In the case of AAL3/4, the burst size for transferring the payload data is 11 double words. Before transferring the payload, the sequence number and the SAR-LI field are checked for consistency. Upon error the PDU is aborted and reported to the host.

3.3.2 Transparent Mode Reception

This mode is similar to the AAL5 mode, except that every received cell is considered as a Single Segment Message (SSM), and CRC-32 is not calculated. After the cell payload has been stored in a receive buffer, the indication is immediately issued by the SARE by writing the RBD in the corresponding Receive Ready Queue. Thus, only one cell is stored per receive buffer.

3.3.3 Reception in Cell FIFO Mode

This mode is suitable for handling Constant Bit Rate traffic, since the cells in the virtual connection are in this case written in a circular FIFO which can be read by the data sink (e.g. host) at a constant pace. The size of the circular FIFO is programmable, thus it can be used as an “elastic buffer” to compensate for the difference (jitter) between the arrival of the cells and the rate of the data sink. In multimedia applications, the size of the FIFO determines the “acceptable” rate of slip, i.e. loss or duplication of data (video image or voice samples) in the application. The FIFO is formed of a concatenation of receive buffers chained together, where the last buffer points to the first data buffer. The circular FIFO is a static data structure, i.e. the individual buffers in the ring are not managed via the Receive Buffer Free Queue.

In the Cell FIFO Mode, the RCTD field of the RVT entry is initialized by the host. It points to a first buffer, where the storage of cells is to be started. The cells are transparently stored, including the header (except HEC), and are always written on 64-byte boundaries in the FIFO. As shown in **figure 9**, each cell has a timestamp stored after the payload (Cell Mode Trailer CMT). The timestamp is derived from a 16-bit counter which is incremented every 60 ns (for 33 MHz PCI clock frequency).

When a buffer becomes full, the pointer at the end of the buffer is read and used as the new buffer descriptor. The current RBD pointer CRBD in RVT is set to this value and the current offset CPOF is set to 0. In addition to the pointer to the next buffer, the Receive Chain Buffer Trailer RCBT at the end of a buffer contains a timestamp associated with that buffer.

No interrupt status is generated after the reception of a cell or after completion of a cell buffer, and the software is responsible for detecting any buffer over- or underflow by checking the pointers CRBD/CPOF in RVT.

The number of cells in a single buffer depends of the BUFSIZE system parameter (viz.,1, 2, 4, 8,..., 128 cells per buffer).

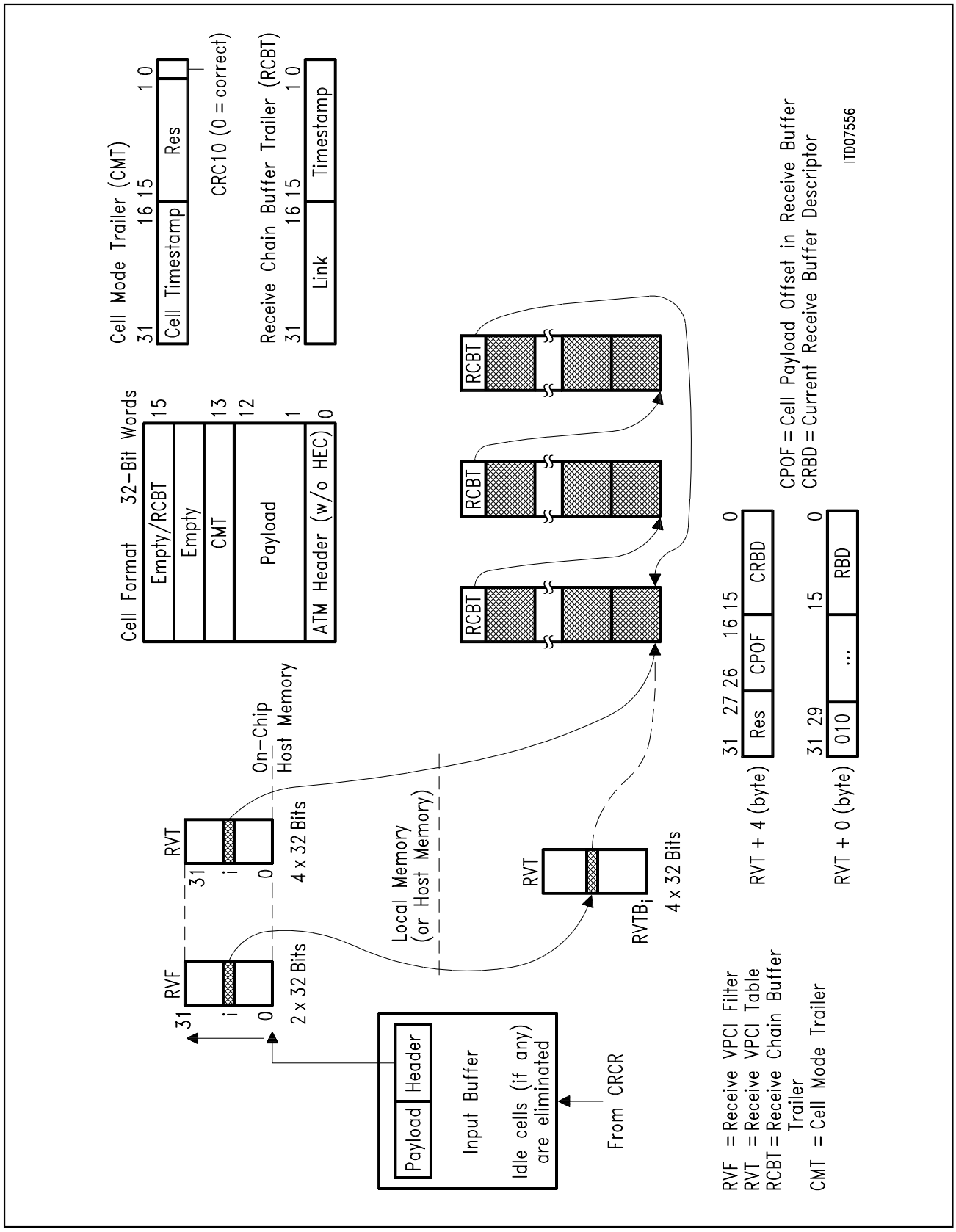


Figure 9 Cell FIFO Mode Receive

3.3.3.1 OAM and Signalling Support

“Idle Cells” are always discarded in the receive direction.

Depending on the value of the configuration register bit C_UACEN, “Unassigned Cells” can be either discarded by the receiver, or they can be let through to the RVF: in the last case, if the combination VPI = 0/VCI = 0 is recognized, the cell is accepted.

Via the configuration register bit C_PHOAMEN the handling of the other physical layer cells, OAM F1-F3 and “PHY layer reserved” cells, can be programmed: they are either discarded or accepted (if VPI = 0 and VCI = 0).

Meta-signalling cells and OAM F4 cells are recognized from their VPI/VCI combination. The same entry in RVF can be used for them as for OAM F1-F3 and Unassigned Cells.

OAM F5 cells, Resource Management cells and cells reserved to ATM layer use the same VPI/VCI combination as corresponding user cells. They are recognized by their PT field and, depending on the value of the C_OAMCF in the configuration register, they are handled either in the Transparent or the Cell FIFO mode. If handled in Transparent mode, they are reported via RRQ0.

In keeping with the ATM Forum UNI Specification, cells with GFC<>0 are counted during 32768 cell times, and an interrupt status can be generated when the number of cells with GFC<>0 exceeds 10.

4 Transmit Data Structures

4.1 Transmit Procedure Summary

Figure 10 shows the segmentation data structures in their general form.

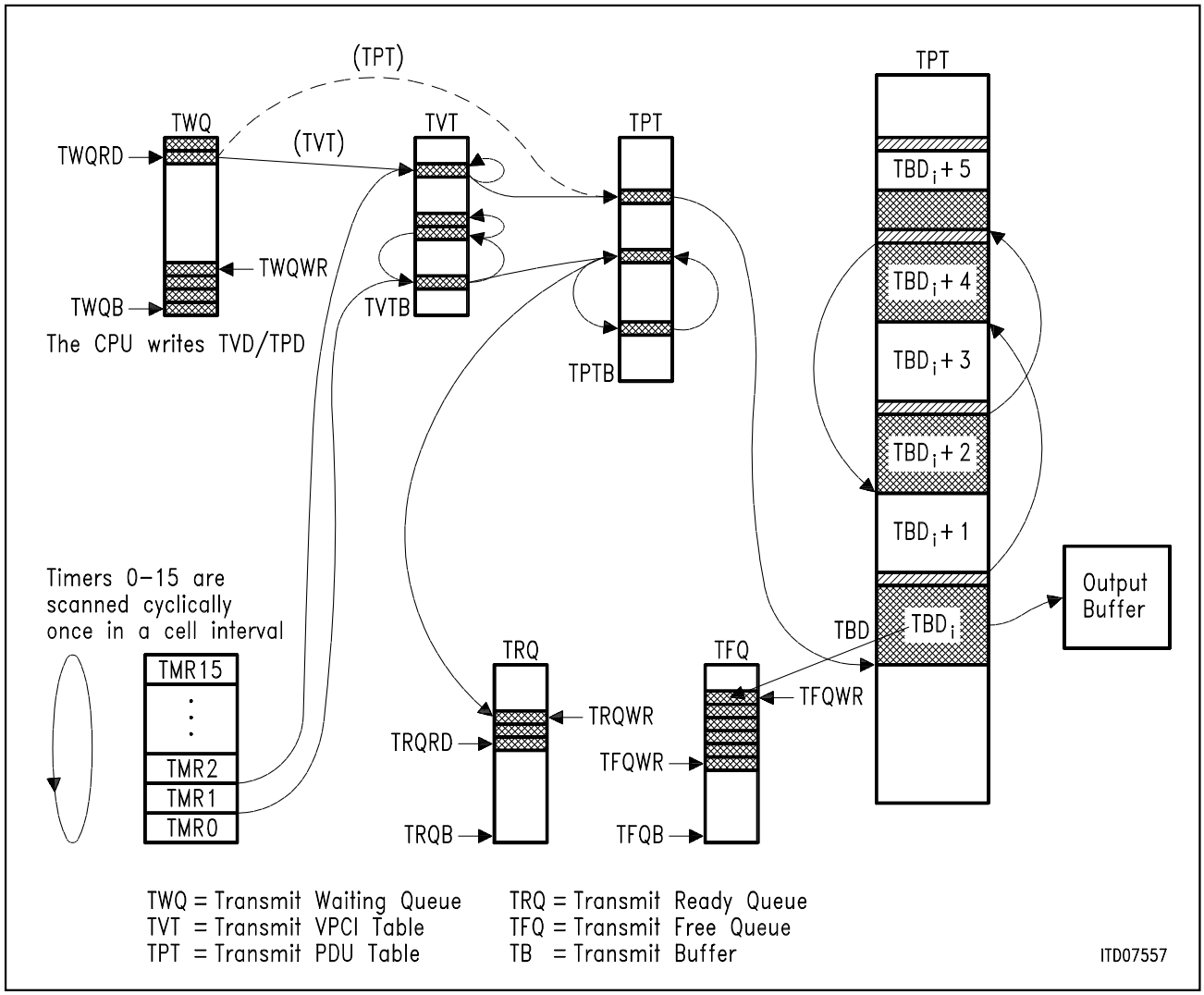


Figure 10 Transmitter Procedure

In order to prepare a virtual connection for segmentation, the host has to initialize an entry for that VC in the Transmit VCI Table (TVT). This entry consists of four 32-bit words and contains, among other things, the value of VPI/VCI, and the effective bit-rate at which this channel is to be segmented. However, as long as no PDU has been released for segmentation, the SARE is not going to use this entry.

To release a PDU for segmentation, the host makes an entry into the Transmit PDU Table (TPT) that contains PDU specific information such as length of the PDU, the variable part of the ATM header, the type of the AAL, and, in the case of AAL3/4, the MID and the starting value of the sequence number. The payload of the PDU is entered

separately in linked list buffers (TBs). At the end of every TB a pointer to the next buffer in the list is entered. Refer to **figure 11**.

TVT and TPT are located either on-chip, or in external local memory, whereas the data buffers are usually located in the host memory.

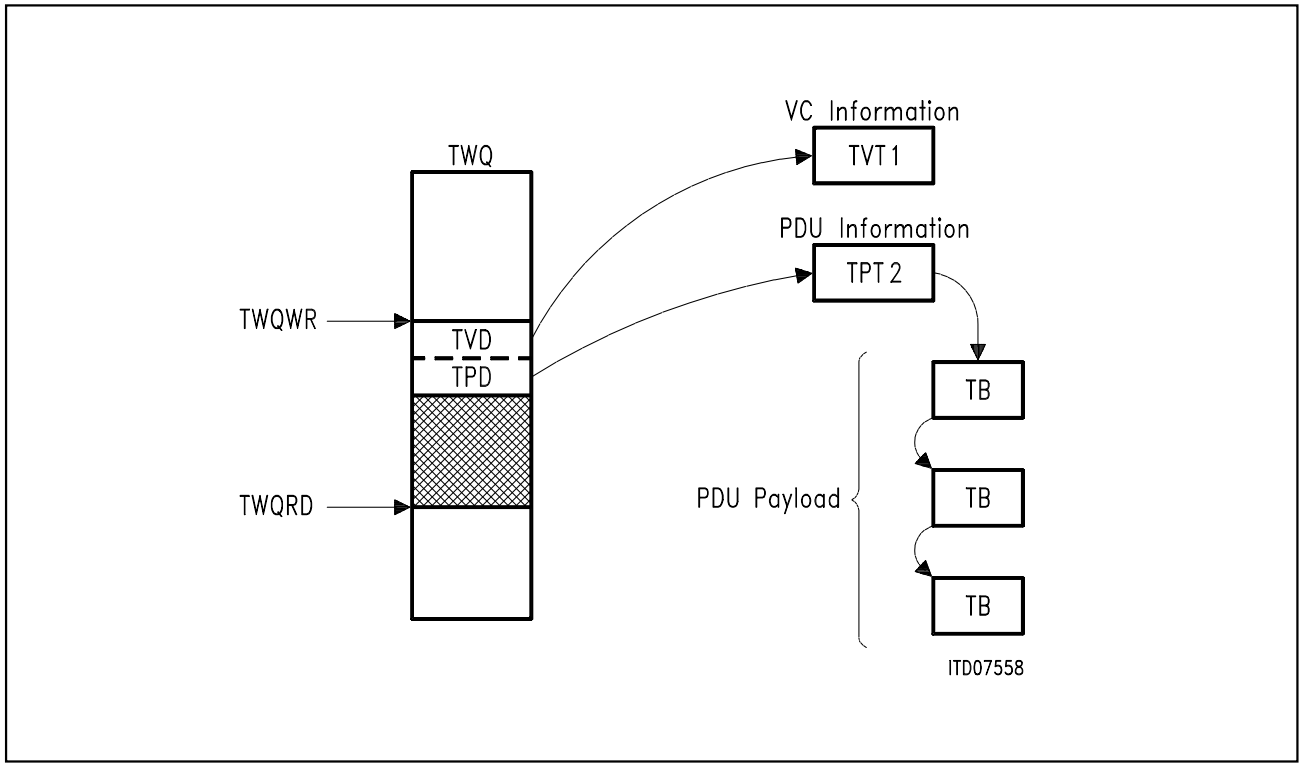


Figure 11 Transmitter Procedure

Finally the host makes an entry into the Transmit Waiting Queue (TWQ) that contains the pointers to the TVT and TPT, called TVT Descriptor (TVD) and TPT Descriptor (TPD), respectively. When the SARE recognizes a new entry in the TWQ, the Credit Manager will include the virtual channel to the proper segmentation scheduler (bit rate timers) and integrate the TPT entry into a linked list chain with other, already active PDUs (if any) in the same virtual channel.

Note: The SARE polls the TWQ once in a cell time (derived from UTOPIA transmit clock divided by 53) to detect a new entry. However, the queue read and write pointers being located in on-chip registers, the system bus will not be loaded by the polling of the TWQ.

The Credit Manager decides, once in a cell cycle, from which PDU (if any) a cell is to be segmented next. The Segmentation Unit (SEG) performs the necessary PDU header and trailer generation (dependent on the particular AAL type) and programs the internal DMA controller to fetch the payload from a TB in the host memory. Completely segmented TBs are released by the SARE for further use via a separate queue (similar to TWQ) called Transmit Free Queue (TFQ). Once a PDU has been completely segmented, this information is issued to the Credit Manager, which takes further action

by removing that PDU from its timer. After a cell from an incompletely segmented PDU has been sent, the corresponding TPT entry is updated.

A special system parameter in a SARE configuration register determines whether AAL3/4 or SMDS type PDUs inside a virtual channel are to be multiplexed (using MID). If this is the case, one cell from each multiplexed PDU is segmented in turn. See figure below.

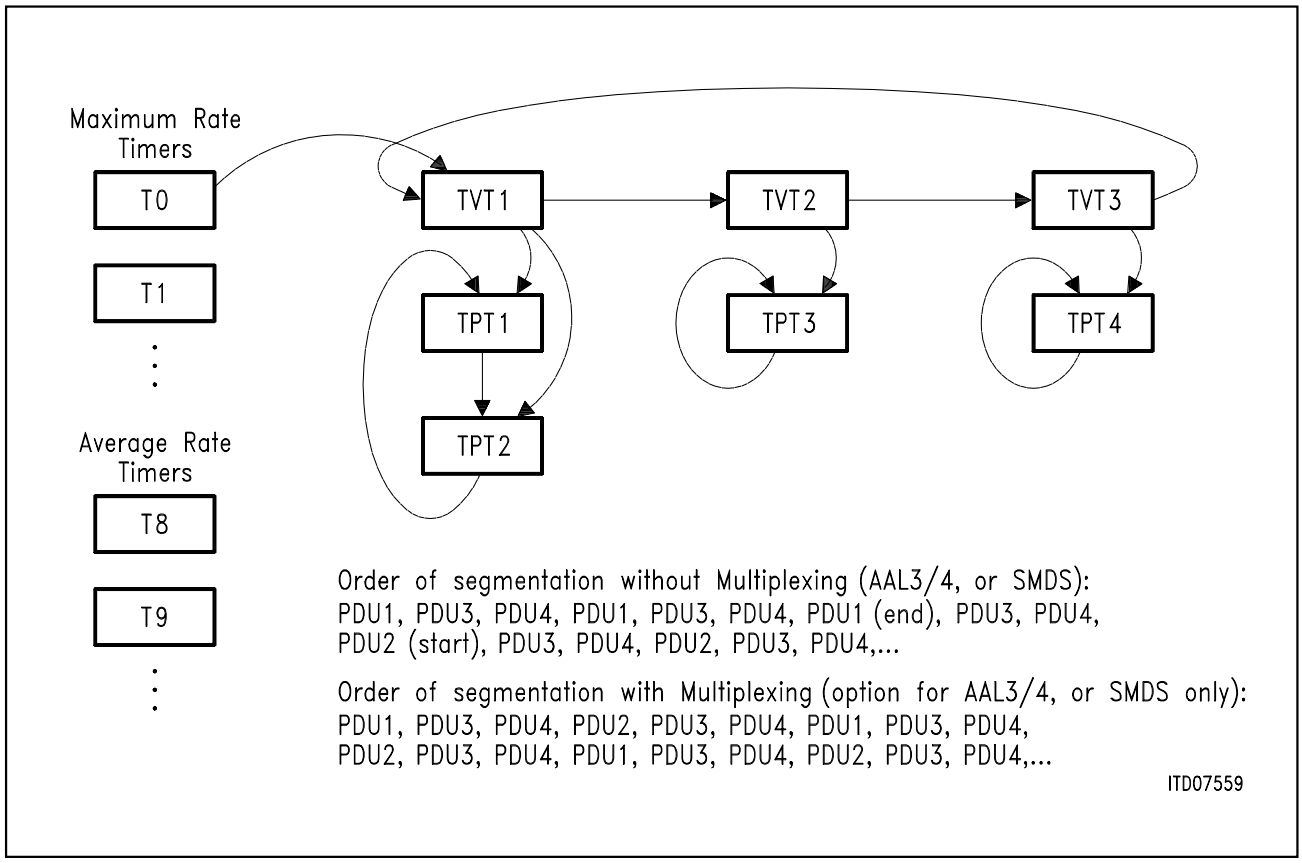


Figure 12 Transmitter Procedure

4.2 Modes of Segmentation

The SARE implements five different segmentation protocols. The mode is individually programmable per virtual connection, and is programmed for each PDU via three MODE bits in the TPT.

In addition to these modes, a special sixth mode, the Cell FIFO Mode, is reserved for Constant Bit Rate traffic. In this special mode, the Credit Manager does not interfere with the scheduling of the segmentation, but the segmentation rate is essentially determined by the data source. All different modes of segmentation can be simultaneously implemented, and the number of simultaneous channels using any of the protocols (including CBR) is arbitrary. See **table 10**.

Table 10 Modes of Segmentation

MODE	Protocol	Credit Manager Involved Yes/No
001	AAL5	Yes
010	AAL3/4	Yes
011	SMDS	Yes
100	OAM	Yes
101	Transparent	Yes
–	Cell FIFO Mode	No

4.3 General Description of Transmit Data Structures

This chapter provides the general description of the data structures. Differences due to the type of AAL and mode-specific differences (especially for the transparent and Cell FIFO modes) are pointed out in the last part of this chapter.

The data structures are summarized in **table 11**.

Table 11 Transmit Data Structures

Name	Abbreviation	Contents
Transmit VCI Table	TVT	VCI specific information
Transmit PDU Table	TPT	PDU specific information
Transmit Buffers	TBs	Payload data
Transmit Waiting Queue	TWQ	Pointers to TPT of new PDUs to be segmented
Transmit Ready Queue	TRQ	Pointers to TPT of completely segmented PDUs
Transmit Free Queue	TFQ	Pointers of released (empty) TBs

4.3.1 Transmit VCI Table (TVT)

Table 12 shows the Transmit VCI Table, located in the external local memory (or, for a number of links, typically up to 60, in on-chip memory) and its information content pertaining to each virtual channel.

Table 12 Transmit VCI Table

Addr	Byte															
	31/15	**30/14**	**29/13**	**28/12**	**27/11**	**26/10**	**25/9**	**24/8**	**23/7**	**22/6**	**21/5**	**20/4**	**19/3**	**18/2**	**17/1**	**16/0**
00_H	CRDECE(2:0)			CGSEL		NTMR			VPI							
02_H	VCI															
04_H	CRMAXE				CRDECE (3)	CGCNT			CREDIT							
06_H	CREDIT															
08_H	CRMAXM								CRDECM							
$0A_H$	TVD_NEXT															
$0C_H$	TPD_PREV															
$0E_H$	TPD															

CRDECE Exponent of Credit Decrement

CGSEL Congestion Control Select

VC-specific throttle factor (1, 1/2, 1/4 or 1/8) for slowing down segmentation in case of congestion

NTMR Timer Number

Determines the cell rate at which this VC is to be segmented

VPI, VCI Virtual Path Identifier, Virtual Circuit Identifier

Corresponding fields in the segmented ATM cells

CRMAXE Exponent of Maximum Credit

CGCNT Congestion Control Counter

CREDIT Current Credit

For the Leaky Bucket Algorithm

CRMAXM Mantissa of Maximum Credit

CRDECM Mantissa of Credit Decrement

TVD_NEXT, TPD_PREV, TPD

Internal pointers to corresponding TVT or TPT table entries

To throttle the segmentation on a virtual connection-by-virtual connection basis, the bits CGSEL can be set by the host as shown in **table 13**.

Table 13 Segmentation Throttle Factor

CGSEL	Throttle Factor
00	1
01	0.5
10	0.25
11	0.125

The three congestion control counter bits CGCNT are used by the SARE internally only.

When writing a new entry into the TVT, the host should initialize the following fields.

TVD_NEXT	Set to TVD of this entry (= The TVT entry points onto itself)
TPD	Should be set to 0 (= No PDUs in segmentation)
CREDIT, CRMAX, CRDEC, CGSEL	Initialized to desired values by the host
CRMAXE	Exponent of Maximum Credit
CGCNT	Set to 0
NTMR	Maximum rate timer number
VPI, VCI	Values for the corresponding virtual connection

4.3.2 Transmit PDU Table (TPT)

Table 14 shows the Transmit PDU Table, located in the external local memory (or, for a number of links typically at least 60, in on-chip memory) and its information content pertaining to each PDU to be segmented.

Table 14 Transmit PDU Table

Addr	Byte															
	31/15	**30/14**	**29/13**	**28/12**	**27/11**	**26/10**	**25/9**	**24/8**	**23/7**	**22/6**	**21/5**	**20/4**	**19/3**	**18/2**	**17/1**	**16/0**
00_H	TBD															
02_H	MODE			STATE		TBD_OFFSET										
04_H	CRC32															
06_H	CRC32															
08_H	PDUSIZE															
$0A_H$	GFC				PT			CLP	x	x	x	x	SN			
$0C_H$	x	x	x	x	x	x	MID									
$0E_H$	TPD_NEXT															

TBD **TB Descriptor**

Pointer to the current data buffer

MODE **Type of protocol**

001 AAL5

010 AAL34

011 SMDS

100 OAM

101 Transparent

STATE **Internal flag**

00 First cell of PDU

10 For all other cells

TBD_OFFSET Double-word offset in the current data buffer

CRC32 **Current intermediate value of CPCS-CRC**

For AAL5, SMDS

PDUSIZE **Number of payload bytes remaining to be segmented**

GFC, PT, CLP, SN, MID

Corresponding fields in the ATM or SAR header

TPD_NEXT Internal chaining pointer

x Not used (reserved)

The host initializes these fields as shown in **table 15**.

Table 15 TPT Initialization Setup by Host

Address_Name	Address_Description
TBD	Pointer to the first transmit buffer for the payload
MODE	Type of protocol required (the type, AAL5, AAL34, SMDS, or OAM, should be the same for all PDUs in a given virtual connection)
STATE	Should be set to 00
TBD_OFFSET	Should be set to 0
CRC32	Should be set to –1 ($0FFFFFFFF_H$)
PDUSIZE	Number of bytes in payload
GFC, PT, CLP	Set to the desired values
MID	MID of the PDU (for AAL3/4 and SMDS only)
SN	Starting value for the sequence number (normally 0)
TPD_NEXT	Points to this TPT entry (TPD of this entry)

4.3.3 Transmit Buffers (TBs)

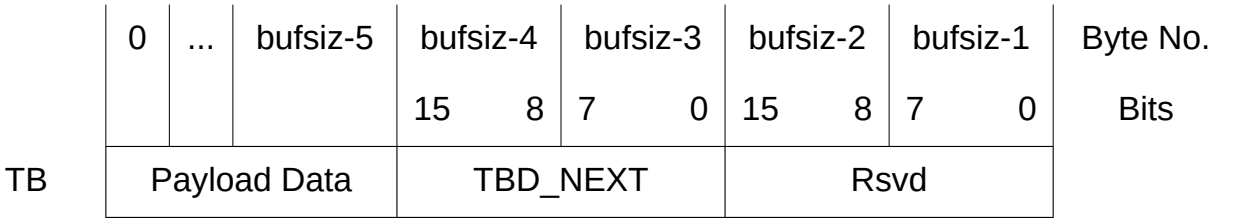
Similar to the payload in reassembly, the payload in segmentation is stored in buffers of programmable length, Transmit Buffers (TBs). The same system parameter, BUFSIZE, determines the buffer length of data in segmentation as for the data in reassembly. Thus the available buffer size options are shown in **table 16**.

Table 16 Buffer Size

Value of BUFSIZE	Buffer Size (bytes), bufsiz
000	64
001	128
010	256
011	512
100	1024
101	2048
110	4096
111	8192

Every TB has a pointer called TBD (Transmit Buffer Descriptor). Linking of the buffers is implemented via a pointer, TBD_NEXT, to the next buffer in the chain, located in the next-to-last 16-bit word of the TB.

The rest of the TB contains payload data, i.e. each TB contains (bufsiz-4) bytes of data, for a total TB length of bufsiz. Thus the TB format is as follows:



The actual current byte address inside a TB is given by:

$$\text{Current-Address in TB} = 256 \times \text{TBB} + \text{bufsiz} \times \text{TBD} + 4 \times \text{TBD\_OFFSET},$$

where TBB is the Transmit Buffer Base address (24-bit system parameter) and TBD and TBD_OFFSET the Transmit Buffer Descriptor and Offset, respectively, both found in the Transmit PDU Table.

As buffers are fully segmented and become free, their pointers, TBDs, are written by the SARE in the Transmit Buffer Free Queue (TFQ).

4.3.4 Transmit Waiting Queue (TWQ)

To release a PDU for segmentation, the host makes an entry into the Transmit Waiting Queue which has the format shown in **table 17**.

Table 17 Transmit Waiting Queue

Address	Byte															
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00_H	TVD															
02_H	TPD															

TVD **TVT Descriptor**

Set to point to the TVT entry of the virtual channel to which the PDU belongs

TPD **TPT Descriptor**

Set to point to the TPT entry for the PDU in question.

4.3.5 Transmit Ready Queue (TRQ)

When the PDU is completely segmented, the SARE makes an entry in the Transmit Ready Queue (TRQ). The format of the TRQ is shown in **table 18**.

Table 18 Transmit Ready Queue

Address	Byte															
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00_H	TPD															

TPD **TPT Descriptor**

Points to the TPT of the completely segmented PDU.

4.4 Operation of the Credit Manager

The Credit Manager is the scheduler that instructs the segmentation unit when and from which virtual connection a cell is next to be segmented.

Eight different cell rates are implemented by the Credit Manager. To each cell rate two timers are attached, called “Maximum rate time” (T0,...,T7) and “Average rate timer” (T8,...,T15). Each virtual connection is individually assigned to one of the eight cell rates, independently of the other virtual connections. Credit is accumulated and decreased by the Credit Manager separately for each virtual connection according to the leaky bucket algorithm: depending on this credit, cells from this virtual connection are segmented either using the “Maximum rate” or the “Average rate” timers. As previously shown, the credit information for each virtual connection is contained in the TVT entry.

The Credit Manager operates autonomously, once the desired maximum and average cell rate values for the eight traffic categories have been programmed. Intervention from the host is only required if the rates are to be changed (e.g. because of congestion), or a virtual channel has to be segmented at a modified rate because of ABR Traffic Management.

Note: For congestion control purposes, it is also possible to use the CGSEL (Congestion Control Select) field in the TVT to throttle the segmentation individually for any virtual connection (throttle factor 1, 1/2, 1/4 or 1/8).

The operation of the Credit Manager is based on the following principles:

– Every one of the eight traffic categories (bit rates) can have an arbitrary number of virtual channels – up to 65535 – simultaneously being segmented at any one time.

– Any AAL3/4 type virtual connection may have up to 1024 PDUs simultaneously in segmentation

– All virtual connections belonging to one traffic category (bit rate) shall be segmented at fairly equal intervals - i.e. not all at once, after a time-out.

The Credit Manager manages the segmentation with the two-dimensional chained data structure as shown in **figure 13**. The TVT entries for the virtual connections are chained together, on the one hand, and, inside a virtual channel, the TPT entries, on the other. A timer, at time-out, points to a certain entry in the TVT, namely to the entry of the virtual channel from which a cell is next to be segmented. This TVT in turn contains a pointer to the TPT entry of the PDU from which the segmentation is to be performed. When the cell has been transmitted, the context for that PDU is updated and re-entered in TPT, the pointer in the TVT is made to point to the next PDU in the circular chain (for MID only; for non-MID type protocols the pointer to the TPT remains constant until the PDU has been completely segmented), and the pointer in the timer made to point to the next element in the chained list of TVT entries.

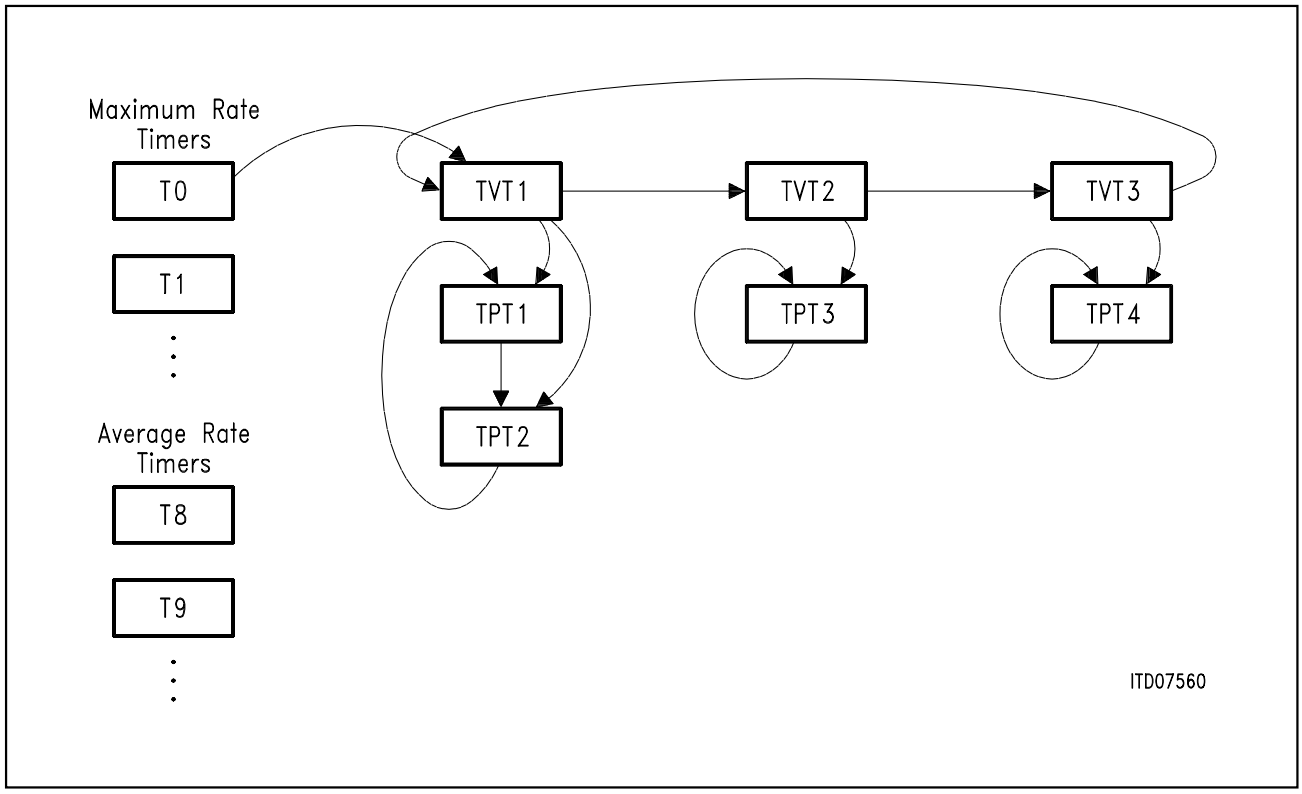


Figure 13 Credit Manager's Two-Dimensional Chained Data Structure

The timer values are dynamically and automatically adjusted by the Credit Manager, since the interval of segmentation of a cell from a given virtual connection is to be held constant irrespective of the number of virtual connections that belong to that traffic category and the associated timer. Thus, if, at any time, m active virtual connections belong to timer Ti – i.e. m PDUs are in the process of being segmented – and the cell interval for that PDU is set to T seconds (= 1/Cell rate for that PDU), the timer will expire every T/m seconds. When a PDU has been completely segmented, it is removed from the chain and the timer value is adjusted to T/(m-1) until a new PDU in a virtual channel in that traffic class is released for segmentation.

During the time that a virtual channel is not active (no PDU in segmentation), the credit for that channel is accumulated at a predefined rate and up to a predefined maximum value. This accumulated credit is calculated as soon as the virtual channel becomes active again. In order to measure the time during which the channel has remained inactive, a timestamp is stored when the last cell of the last PDU in that channel is segmented, and this value is subtracted from the "current time" when the first cell of a new PDU in that channel is segmented.

When a virtual channel belonging to timer Ti has exhausted its credit, it is automatically assigned to the "Average rate timer" Ti+8 by the Credit Manager.

Thus each virtual connection may be in one of the three states. See **table 19** for a description of each state and its associated actions.

Table 19 Potential States of a Virtual Connection

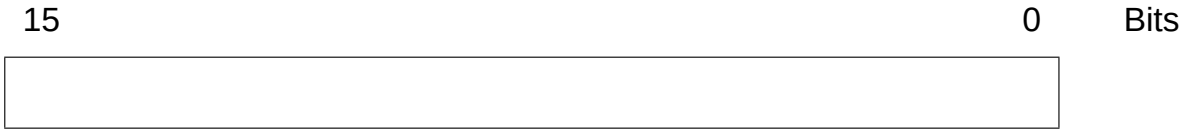
State	Description	Action
State1	No PDU in segmentation	Credit accumulates up to a maximum value
State 2	PDU in segmentation, Credit exhausted	Segmentation using Average Rate timer
State 3	PDU in segmentation, Credit not exhausted	Segmentation using Maximum Rate timer, Credit decrements

State 1 is invisible to the Credit Manager, since the TVT entry is not part of a chain linked to a timer. When State 1 is left, the accumulated Credit is calculated from an expression involving the old credit, the length of time spent in State 1 and the maximum Credit specified by the host.

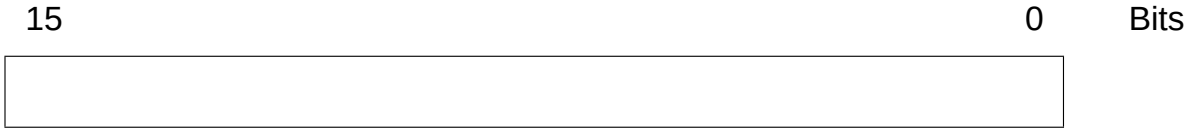
To enable the above operations by Credit Manager, each of the timers Ti (i = 0,...,15) has the following registers associated with it:

TiTVD

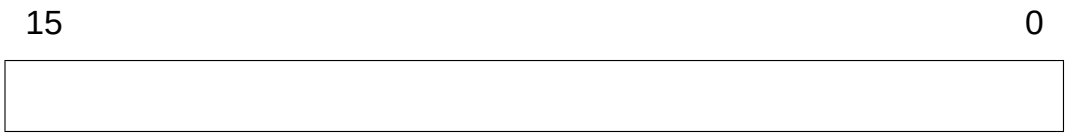
TiTVDPREV



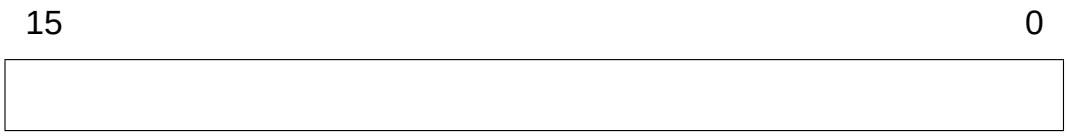
TiPHASE



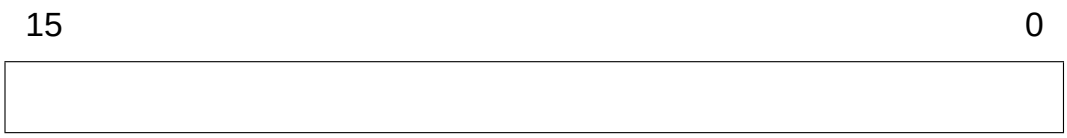
TiK



TiL



TiK0



TiTVD **TVD Descriptor**

Pointer to the next TVT entry in chained list

TiTVDPREV **TVD Previous**

Pointer to the previously processed TVT entry in chained list

TiPHASE **Internal state of the bit-rate timer**

TiK **Nominator of the bit-rate quotient**

TiL **Denominator of the bit-rate quotient**

TiK0 **Increment/Decrement for TiK**

When virtual channel is added/removed, respectively.

The host should initialize registers TiK, TiL and TiK0. The resulting bit-rate of a virtual channel amounts to

$r = 8 \times (\text{Frequency of UTOPIA TXCLK}) \times TiK0/TiL.$

These registers should be changed during operation only when the timer is not used (when TiK = 0).

The other registers should not be written via the host, but are reserved for the Credit Manager. They can be read for testing purposes, however.

4.5 Description of the Specific Modes of Segmentation

In this chapter, the different modes of segmentation and frame formats are described, as determined by the 3 MODE bits in the TPT entry.

As mentioned in the beginning, the Credit Manager schedules the segmentation in all modes except the Cell FIFO mode.

4.5.1 AAL5

For AAL5, the CPCS-PDU, the padding bytes and the first four bytes (CPCS-UU, CPI and Length) of the CPCS-PDU trailer are stored in the TB(s), in this order. The total length should amount to n $\times$ 48 + 44 bytes (n integer), so that after CRC-32 is added by the SARE, the total length of the payloads amounts to a multiple of 48 bytes. n may be a value between 0 and 1365, giving a total data length of 44 to 65564 bytes. The CPCS-PDU payload can have a length of up to 65535 bytes.

Added by SARE

CRC-32

Always end of an ATM cell payload

PDUSIZE Bytes	0-47 Bytes	1 Byte	1 Byte	2 Bytes
CPCS-PDU Payload	PAD	CPCS-UU	CPI	Length

The PDUSIZE field in the TPT entry is to be set to the length of the CPCS-PDU payload (in bytes), not the total data length. Thus, 16 bits for the PDUSIZE are sufficient. A PDUSIZE value of zero is not allowed. The value should conform to the Length parameter in the CPCS-PDU trailer.

The data are transparently segmented, so that PAD, CPCS-UU, CPI and Length are under the exclusive responsibility of the software and are not checked by the SARE.

4.5.2 AAL3/4

CPCS-PDU header (CPI, Btag, BASize = 4 bytes), the CPCS-PDU payload (1 to 65535 bytes), the PAD field (0 to 3 bytes) and the CPCS-PDU trailer (AL, Etag and Length = 4 bytes) are stored in the TB(s), in this order. The total length of the data is thus between 12 and 65544 bytes.

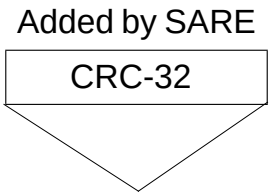
1 Byte	1 Byte	2 Bytes	PDUSIZE Bytes	0-3 Bytes	1 Byte	1 Byte	2 Bytes
CPI	Btag	BASize	CPCS-PDU Payload	PAD	AL	Etag	Length

The PDUSIZE field in the TPT entry is to be set to the length of the CPCS-PDU payload (in bytes), not the total data length. Thus, 16 bits for the PDUSIZE are sufficient. A PDUSIZE value of zero is not allowed. The value should conform to the Length parameter in the CPCS-PDU trailer and should not exceed the value of the BASize field in the CPCS-PDU Header.

The data are transparently segmented, so that the whole PDU Overhead is the exclusive responsibility of the software. This data is not checked or modified by the SARE.

4.5.3 SMDS

The format of the data in the memory is identical to the format in the AAL3/4 case. The only difference is that the SARE inserts a CRC- checksum between the PAD field and the CPCS trailer. This CRC-32 is calculated over the fields DA to PAD. In the TBs, the AL field should be contiguous with the PAD field, though.



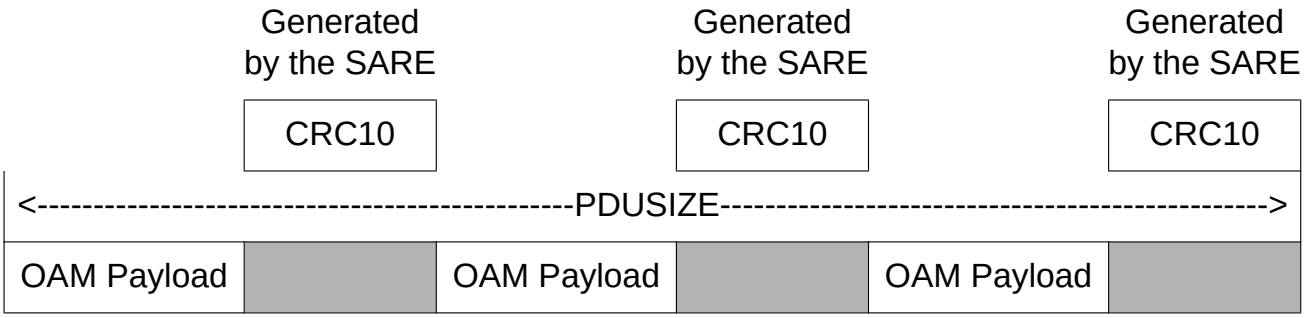
1 Byte	1 Byte	2 Bytes	PDUSIZE Bytes	0-3 Bytes	1 Byte	1 Byte	2 Bytes
CPI	Btag	BASize	CPCS-PDU Payload	PAD	AL	Etag	Length

The payload can in principle be longer than the SMDS limit of 9188 bytes.

In order to generate SMDS-PDUs without CRC-32 the AAL3/4 mode should be used.

4.5.4 OAM

OAM cells have a payload of 48 bytes, of which the last 10 bits are occupied by the CRC-10 checksum, generated by the SARE. The 48 bytes are transparently sent by the SARE, with the exception of the last 10 bits. OAM cells can be sent either one by one (i.e., one TPT entry and one TB per OAM cell), or n OAM cells can be grouped together in a single TPT entry and concatenated in one or more TBs (“message mode”), to be sent at the prescribed bit rate, managed by the Credit Manager.

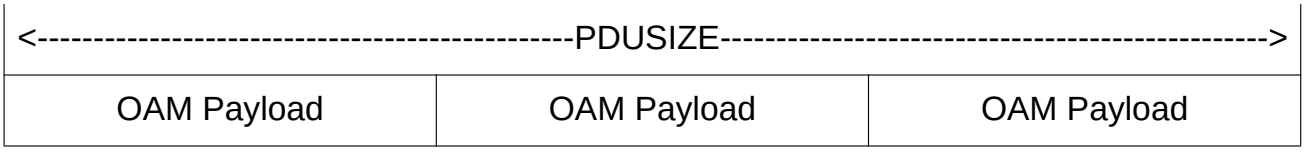


48 to 65520 Bytes

The PDUSIZE field in the TPT entry is to be set equal to the sum of the payloads (48 or 48 × n bytes).

4.5.5 Transparent Mode

The transparent mode is similar to the OAM mode. However, the last 10 bits are also sent transparently from the TB.



48 to 65520 Bytes

4.5.6 Cell FIFO Mode

The Cell FIFO mode accounts for the needs of non-packet oriented ATM traffic. Transmission of cells in this mode is not scheduled by the Credit Manager. Also, data structures for the virtual connections that operate in this mode differs from that of the other modes of operation, to fulfill the following requirements:

– Generation of header (except HEC) completely under the control of the user

– Support for both CBR and VBR traffic

– Buffer size (= FIFO size) determined by the user (arbitrarily large)

– Efficient communication between host and SARE (minimal overhead)
– Cells ready to be transmitted in Cell FIFO mode always have higher priority than virtual connections in other modes.

The last two criteria mean that minimal latency between the data source (user) and the network is achieved.

The principle of operation of the Cell FIFO is shown in **figure 14**. The Transmit Waiting Queue is polled once per cell time by the Credit Manager. If a request to segment a cell in the Cell FIFO mode has been entered into the TWQ, this request is immediately satisfied, i.e. the Cell FIFO mode has priority over all other connections (which are managed by the Credit Manager scheduler).

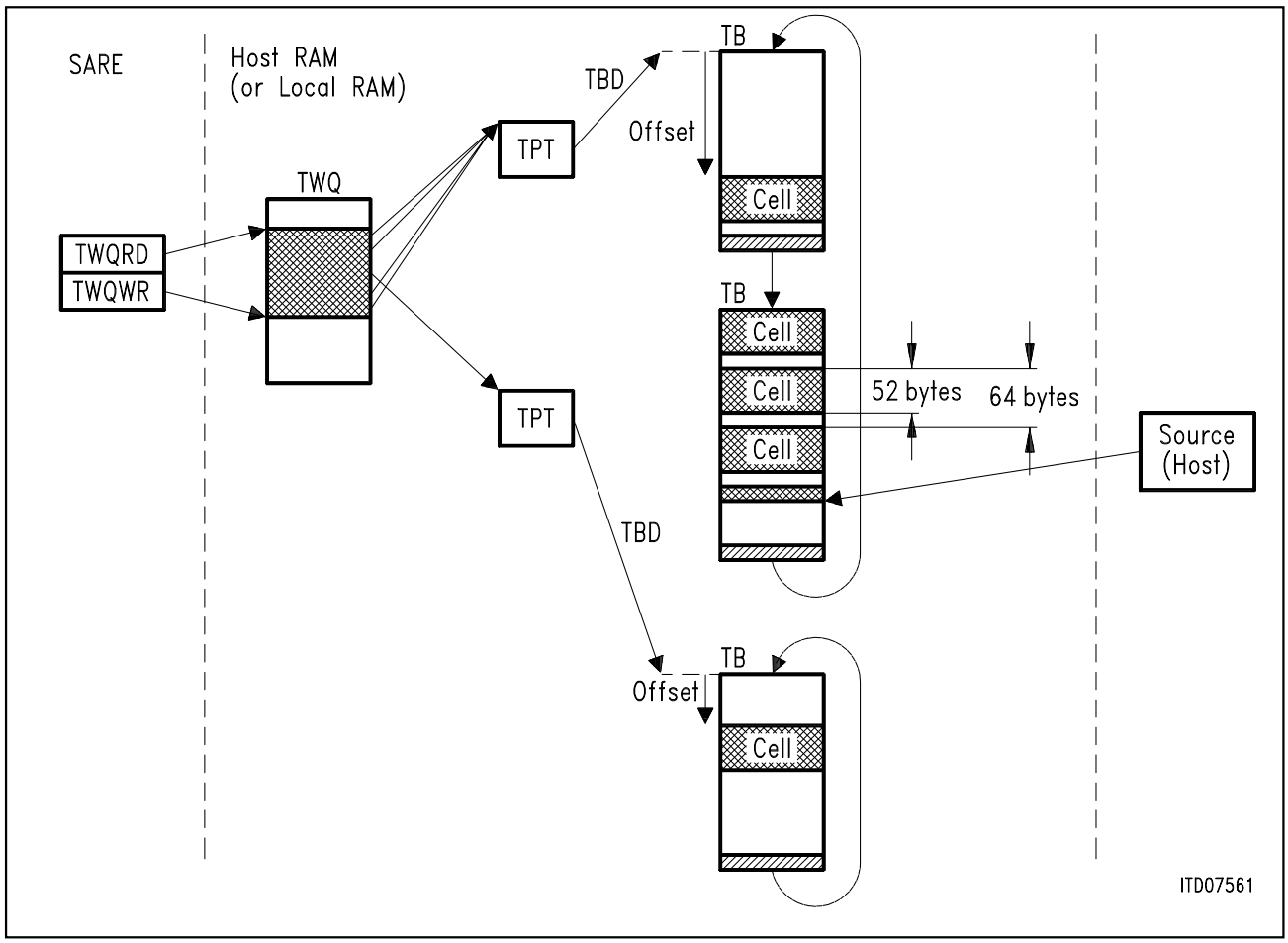


Figure 14 Cell FIFO Operation Principle

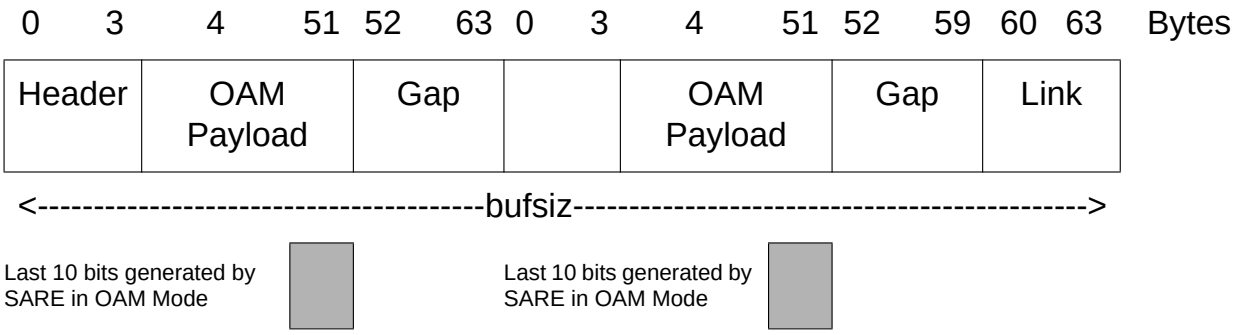
The entry in the TWQ contains a pointer to a corresponding TPT entry. The TPT entry in turn contains a pointer (Transmit Buffer Descriptor, TBD) to a Transmit Buffer, along with an offset inside that buffer (TBD_OFFSET). The TBs are statically assigned, i.e. they are not managed via the Transmit Buffer Free Queue (TFQ) (to be used by possibly other virtual connections), but should be organized in a ring buffer structure, with the last buffer pointing to the first. The number of buffers linked together is arbitrary. Because of the

high priority given in the processing to cells is Cell FIFO mode, normally a small ring buffer chain should be enough (depending on the traffic type, of course). The host should prepare the ring list buffers before entering a TPT for that virtual connection in cell FIFO mode into the TWQ.

The buffers are linked together via a pointer contained in the last word of the buffer. The size of each buffer is the same as in the other modes, and is determined by the BUFSIZE system parameter.

Cells are entered in the TBs always on 64-byte boundaries, 52 bytes per cell, including header (except the HEC). The cells can be transmitted in either of two modes, as determined by the MODE parameter in the TPT entry: either in the transparent mode or in the OAM mode. The only difference between these is that in the OAM mode the last 10 bits of the cell are replaced by the CRC-10 checksum generated by the SARE, whereas in the transparent mode these bits are transmitted transparently from the buffer. Thus the format of the cell in the TB is as follows:

TB



For every cell to be transmitted in the Cell FIFO mode, one entry into the TWQ is made. The entry in the TWQ for Cell FIFO is characterized by the fact that the TVD is zero. See **table 20**.

Table 20 TWQ in Cell FIFO Mode

Address	Byte															
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00_H	0000000000000000															
02_H	TPD[1)]															

1) Where TPD (TPT Descriptor) points to the entry in the TPT Table.

Note: Writing a new entry into TWQ by the host involves also updating the TWQ Write Pointer register of the SARE (1 read + 1 write in addition to writing TWQ itself). The software should ensure that these operations are "locked" together so that the Write pointer is updated correctly (only one segmentation process can perform these operations at any one time). As an alternative, use the register TWQPUT to perform the incrementation by 1 of the pointer in just one read operation.

The entry in the TPT has the simplified format shown in **table 21**.

Table 21 TPT in Cell FIFO Mode

Address	Byte															
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00_H	TBD															
02_H	MODE			x	x	TBD_OFFSET										
04_H	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
06_H	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
08_H	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
$0A_H$	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
$0C_H$	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
$0E_H$	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

TBD **TB Descriptor**

Pointer to the current data buffer to be initialized by the host to the address of the first buffer, thereafter updated by the SARE automatically

MODE **Type of protocol**

100 OAM

101 Transparent

(The difference is that in OAM mode CRC10 is calculated, and in the transparent case not.)

TBD_OFFSET **Offset in 32-bit words in the current data buffer**

Set to 0 by the host in the beginning, updated automatically by the SARE thereafter.

x **Unused**

When a new TWQ entry is made by the host, the SARE first checks the value of the first word of the entry (TVD):

1. If it is zero, a cell is segmented from the cell FIFO pointed to by the corresponding TPT entry. Thereafter the offset (TBD_OFFSET) is incremented by 64 (bytes), and, if necessary, the link pointer to the next TB is read from the TB and stored in the TPT as TBD.
2. If it is not non-zero, a cell may be segmented from one of the eight average rate and peak rate queues (in the case of a pending time-out), as described in the previous sections.

4.5.7 Additional Functions

4.5.7.1 Generation of an Incorrect HEC (for testing)

In the generation of the HEC the contents of a programmable register, COSETS, is exclusively OR'ed with the result of the HEC polynomial ($x^8 + x^2 + x + 1$). In keeping with the ATM UNI standard, a correct HEC is obtained when COSETS is set to 55_H, which is also the default value after reset.

For testing purposes, it is possible to generate an incorrect HEC for the cells by setting the COSETS parameter to a value different from 55_H. However, the HEC of every transmitted cell will be affected.

For reference purposes, the CRC generation characteristics are shown in **table 22**.

Table 22 CRC Generation Characteristics

Name	Polynomial	Start Value	XOR'd with
HEC	x^8+x^2+x+1	0	COSETS
CRC10	$x^{10}+x^9+x^5+x^4+x+1$	0	0
CRC32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$	$0FFFFFFFF_H$	$0FFFFFFFF_H$

4.5.7.2 Empty Cell Generation

If generation of empty cells is desired, the pattern of the empty cells can be programmed. The registers ECH (Empty Cell Header) and ECPAT (Empty Cell Pattern) contain the header and the payload of the empty cell:

ECH

31 24	23 16	15 8	7 0	Bits
Header (Byte 1)	Header (Byte 2)	Header (Byte 3)	Header (Byte 4)	

ECPAT

7 0
Payload Byte
(Bytes 6-53)

The HEC is generated as described above, as the 5th byte of the cell header.

In keeping with the standards, the default values of the registers ECH and ECPAT (e.g. after a Reset) are:

ECH = 0000001_H
ECPAT = $6A_H$.

5 System Interface and Operational Description

5.1 DMA Controller and Memory Management Unit

5.1.1 General Structure

The internal DMA controller of the SARE performs all the data transfers necessary for the operation of the segmentation and reassembly. The associated Memory Management Unit arbitrates between the different accesses.

The DMA controller is connected to the PCI Interface, on the one hand, and to the local memory interface, on the other. See **figure 15**.

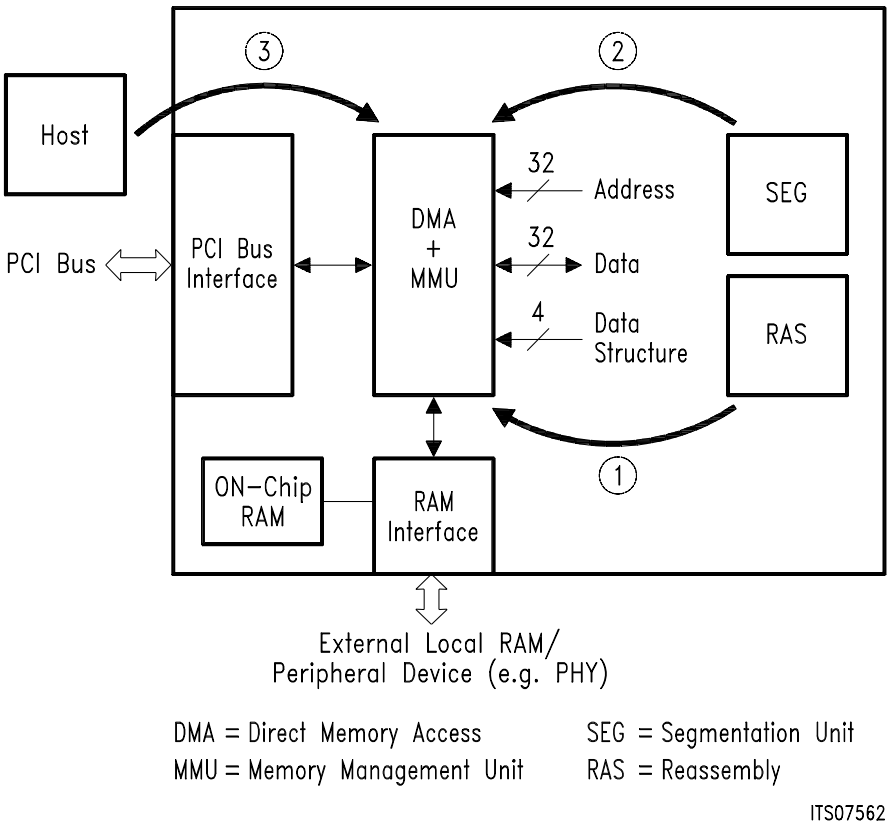


Figure 15 DMA/MMU Connections

The DMA controller gets its instructions from the two blocks RAS (Reassembly Unit) and SEG (Segmentation Unit).

Arbitration is performed internally between the following access types:

1. Accesses initiated by RAS, to transfer payload data to an external memory (host memory or local external memory), and to retrieve and store control data from/to an external memory.
2. Accesses initiated by SEG, to transfer payload data from an external memory (host memory or local external memory), and to retrieve and store control data from/to an external memory.
3. Accesses from the PCI interface (host) to the SARE on-chip RAM, SARE registers, or the external local memory interface (either to external memory or to an external peripheral, e.g. a PHY device).

The order of priority of these accesses is: 1, 3, 2. Thus, transfers initiated by the RAS have the highest priority (in order to prevent overflow of internal data buffers).

In type 3 accesses the SARE operates as a PCI slave device. These are handled in a later part of this chapter. The rest of this section deals only with the DMA controller (access types 1 and 2).

5.1.2 DMA Operation

Different data types have to be transferred between the SARE and the external memory. For each data type (payload, RVT, TPT etc.) an individual configuration bit is programmed in the MMU in order to determine whether the corresponding data is to be transferred between the SARE and the PCI, or between the SARE and the external local memory interface:

control bit = 1: PCI Interface

control bit = 0: external local memory interface.

Every data transfer originated by the RAS or the SEG carries the information of the “Data Structure Type” to the MMU/DMA controller, which performs the transfer according to the configuration setting for this data type.

5.1.2.1 DMA Controller Write

In the case of a transfer to the external local memory, no buffering (because of bus latencies) is required since the SARE is the only master on this bus.

In the case of a transfer to the PCI interface, the possible bus arbitration time (latency) makes it necessary to implement buffering of data at the PCI output. Additionally, queuing of DMA transfer commands is necessary. This is implemented via two FIFOs (see **figure 16**):

1. Address FIFO with up to 16 consecutive data transfer commands, which makes the DMA controller appear as a 16-channel controller. The FIFO contains the start addresses for 16 consecutive transfers, each with a number (from 1 to 128) of double-words.
2. At the PCI interface a data FIFO of 128 entries, associated with the address FIFO. Each DMA transfer may involve an arbitrary number of words from the data FIFO (from 1 to 128). Neglecting any transfer of control data, this corresponds to the storage space for approximately 10 ATM cell payloads, which, in the worst case of a 155 Mbit/s cell burst leads to an approximate maximum bus latency of 30 µs before a receive cell overflow may occur.

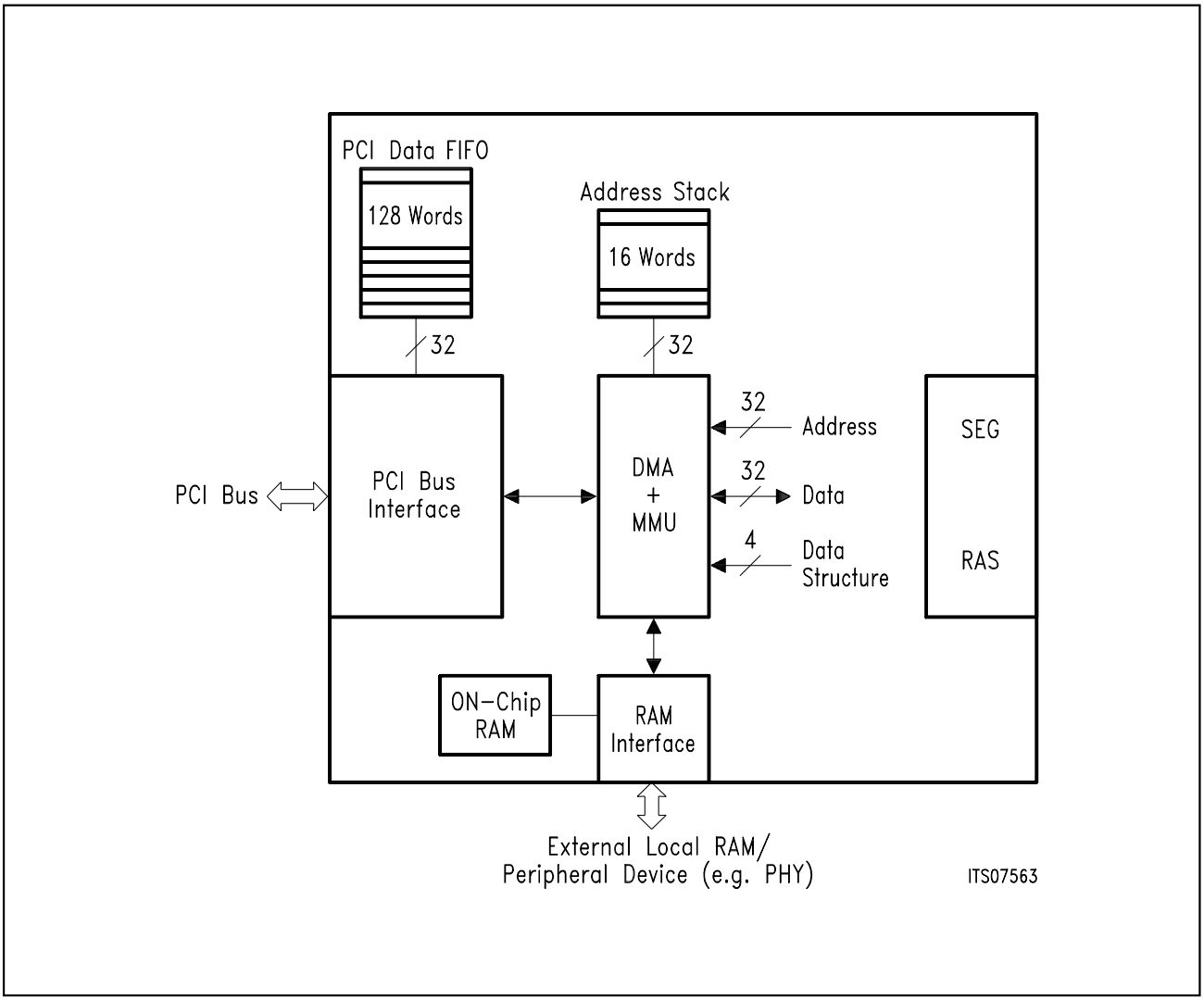


Figure 16 DMA/MMU Operation

The actual transfer of data to the PCI Interface via DMA starts when one of the following conditions is fulfilled:

1. The last data word belonging to the DMA instruction is moved to the FIFO. This is recognized internally via address comparison: thus, when the address for DMA makes a “jump”, this causes a new entry to be made into the address FIFO i.e. a new DMA channel to be opened and the execution of the previous DMA-request to be started.
2. A programmable threshold in the FIFO has been reached. This threshold may be set to be either 16, 32, 64 or 96 words (32-bit words).

The first condition is necessary to avoid seizing the PCI bus for a very short time (and often). The second condition handles the case where cells belonging to one PDU are received back-to-back, and payloads (12 double-words long each) belonging to these cells are transferred to the same buffer in the host memory: In this case the DMA burst size could conceivably even exceed the data FIFO size of 128 words.

The FIFO threshold can be adjusted to be optimal for every possible system, taking into account the load on the PCI, allowed burst size and retry-wait-time.

When the PCI data FIFO reaches the threshold of 96 double-words, reception of further data is automatically inhibited by disabling the UTOPIA handshake.

5.1.2.2 DMA Controller Read

Since the internal working clock of the SARE is the PCI clock, no buffering of data in this direction is needed.

5.2 Host Accesses

The types of accesses on the PCI interface from the host are shown in **table 23**, with the SARE in the PCI slave mode. In the case of PCI memory accesses, the PCI address lines AD23 and AD22 are decoded to distinguish between an access to an External local memory, SARE on-chip memories and a peripheral device connected to the External local memory interface.

Table 23 Host Access Types

Host Access to	AD23	AD22	Type of PCI Access
SARE Registers			I/O
External local RAM	0	0	Memory
SARE on-chip RAM	0	1	Memory
Peripheral device on external local interface	1	x	Memory

The SARE Registers (Configuration, Control and Status registers) allow the direct control and monitoring of the SARE via the PCI bus. They (as well as the PCI configuration registers) are described in the following chapter.

Accesses to the on-chip memory are effected via PCI memory accesses with AD23,22 = 01. The SARE on-chip RAM occupies a memory space of 4 Kbytes. When an on-chip RAM access from the PCI is recognized via AD23,22 by the MMU, the internal RAM address is decoded from the low significant address bits, thereby ignoring the bits
AD12-21. (Thus, the memory range is multiplied inside the total address range of 2 Mbytes.)

The base addresses for all data and control structures are individually programmable via the SARE configuration and control registers. Thus, for on-chip memory as for external memory, the host has complete control over the address ranges it chooses for the different data structures. Whether an access initiated by the SARE is to an on-chip memory or not is “indirectly” determined by the base address. More specifically, the internal absolute address i_addr (for any given data structure) is obtained from the following expression:

$$i\_addr = base\_address \times 64 + offset \quad \text{(all values in 32 bit words)}$$

where base_address is the value (a 24-bit value) in the base address register for that data structure and offset is a 16-bit pointer. Numbering the bits in i_addr from i_addr0 (least significant bit) upwards, an access to an on-chip RAM is performed if bits i_addr21,20 = 01. This is consistent with the decoding on the PCI interface.

The simplest way to select a data structure to be on-chip is to program bits a15,14 = 01 in the base address register (assuming a0 to be the least significant bit in the value written in the base address register) and ensuring that the value of the base address register and the size of the data structure are such that the expression given above always gives a result where i_addr21,20 = 01.

For a limited number of simultaneous virtual connections, all control and context data can be resident in on-chip RAM. This allows for low-cost implementations of AAL terminating equipment without any external local memory, on the one hand, and optimizes the use of the system bus by eliminating transfer of control and context data via the PCI Interface, on the other (case a in **figure 17**).

Accesses to an external local memory via the PCI are effected by PCI memory accesses with AD23,22 = 00. The lines AD(21-2) in the address cycle are directly mapped to LA(19-0) outputs using a non-multiplexed address/data bus timing, and AD(31-0) in the data cycle are directly mapped to LD(31-0). External memory accesses allow for the implementation of systems with a large number of simultaneous virtual connections. Usually a local memory is recommended to be used to store control and context data for PDUs currently in segmentation or in reassembly. In this case, access to the external local memory via the PCI is mainly necessary for the host to initialize the different data structures (case b in **figure 17**). Systems with local memory can also be designed, where the PDUs are assembled in the local memory (store-and-forward architectures, case c in **figure 17**) – although in this case the local memory interface may turn out to be a bottleneck for high cell rates.

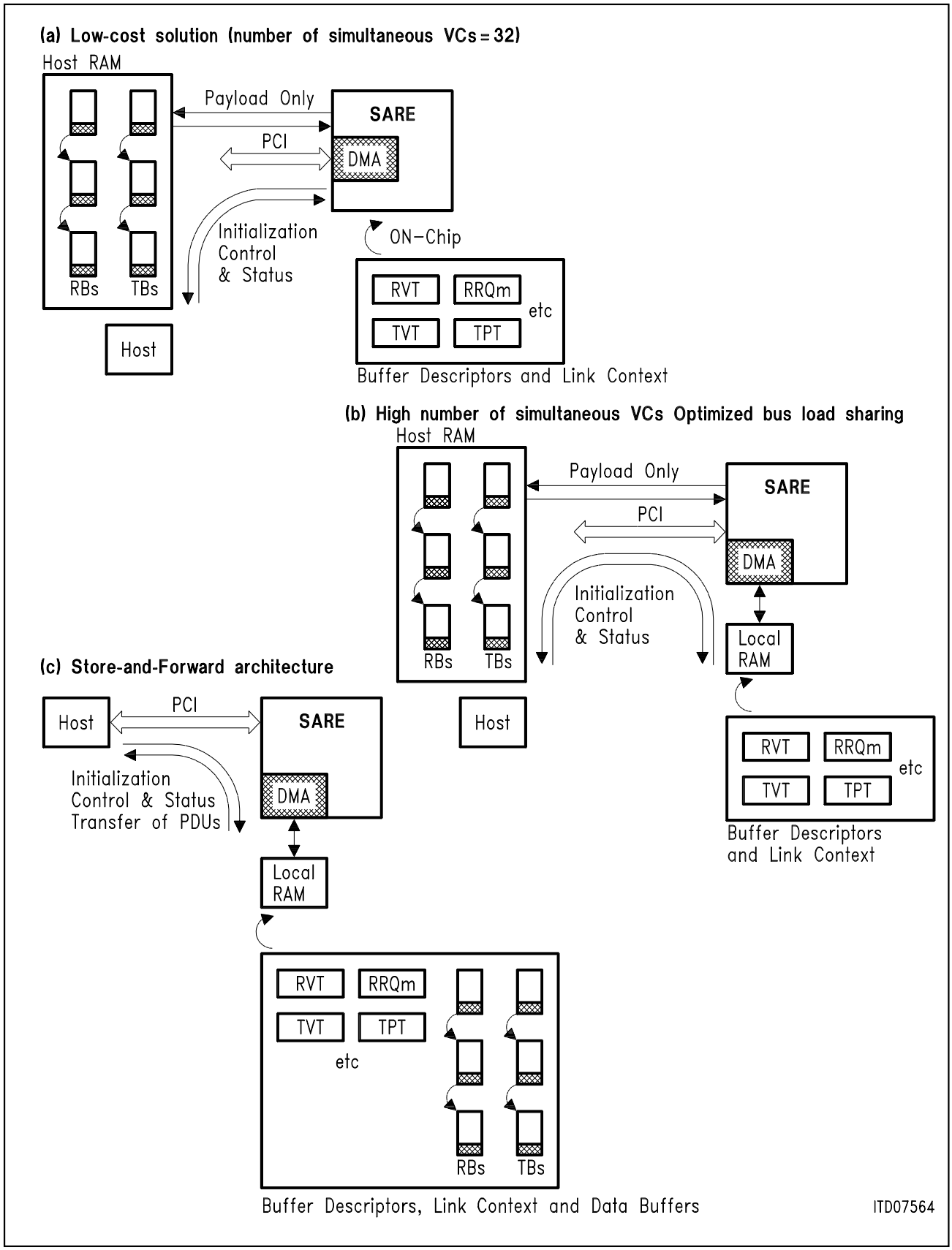


Figure 17 Configuration Options

Accesses to a peripheral device via the PCI are effected via PCI memory accesses with AD23,22 = 1x. As in the case of local memory accesses, the lines AD(21-2) in the address cycle are mapped to LA(19-0) using a non-multiplexed address/data bus timing, and AD(31-0) in the data cycle are mapped to LD(31-0). The address on LA is always a double-word address. Thus, in the case of a peripheral device that supports only a 16-bit data bus with a word address, say, on A(x to 0), line LA(0) should be connected to A(0) of the peripheral device, line LA(1) to A(1) etc., and the data bus connected e.g. to the 16 least significant bits of LD(31-0). Instead of the local memory chip select line $\overline{\text{LCS}}$, the peripheral device chip select line $\overline{\text{PCS}}$ is activated. In addition, and contrarily to memory accesses, peripheral accesses are asynchronous in the sense that the PREADY input is monitored and wait cycles are inserted as long as PREADY is not activated. For peripheral devices able to run at the speed of the local bus without wait states (maximum cycle time 60 ns), pin PREADY may be tied permanently to “high”. A write or read operation to a peripheral (typically a PHY device) can be aborted by a transfer request originated by the SARE Reassembly Unit, which has higher access priority, in which case a special interrupt status is generated.

In the rest of this chapter, information is given for system designers that enables to use the SARE so as to minimize cost and optimize the loading of the system bus. Guidelines are given to calculate the total memory needs, as well as examples of recommended memory configurations for various system requirements.

5.3 Memory Requirements

Memory requirements can be calculated to optimize the system architecture. **Tables 24** and **26** show the types of accesses from the host or by the SARE, for the purpose of bus load optimization and the variables used for calculation, respectively. **Table 26** shows the data structures and associated equations for calculating optimized memory requirements.

Table 24 Access Type Nomenclature

Abbreviation	Meaning	Description
i	Initialization	Access from host is only required at initialization. (After initialization by the host, only accesses by the SARE are performed.)
r and/or w	Read and/or write	Read or write accesses by the entity (host or SARE) are called for during normal operation.

Table 25 Variables Used in Calculation of Required Storage Space

Abbreviation	Description
N_{RAS}	Number of simultaneous virtual connections in reassembly
N_{PDU_R}	Number of PDUs in reassembly
L_{PDU_R}	Average length of payload of PDUs in reassembly (bytes)
L_{RB}	Length of chained receive buffers
N_{SEG}	Number of simultaneous virtual connections in segmentation
N_{PDU_S}	Number of PDUs in segmentation
L_{PDU_S}	Average length of payload of PDUs in segmentation (bytes)
L_{TB}	Length of chained transmit buffers (bytes)
NRMID	Number of virtual connections (AAL3/4) with MID multiplexing
SMID	Number of significant MID bits (between 1 and 10)

Table 26 Storage Space Required by Each Data Structure

Data Structure	Access from Host	Access by RAS	Access by SEG	Size (Bytes)
RVT	i	r/w	–	$16 \times N_{RAS}$
RMQ	i	r/w	–	$2 \times$ NRMID
RMT	i	r/w	–	$16 \times 2^{SMID} \times$ NRMID
RB	r	w	–	$L_{RB} \times N_{PDU\_R} \times ((L_{PDU\_R} + 7 + L_{RB})//(L_{RB} - 4))$
RFQ	i/w	r	–	$2 \times N_{PDU\_R} \times ((L_{PDU\_R} + 7 + L_{RB})//(L_{RB} - 4))$
RRQ	r	w	–	$2 \times N_{PDU\_R}$
TB	w	–	r	$L_{TB} \times N_{PDU\_S} \times ((L_{PDU\_S} - 5 + L_{TB})//(L_{TB} - 4))$
TVT	w	–	r	$16 \times N_{SEG}$
TPT	w	–	r	$16 \times N_{PDU\_S}$
TWQ	w	–	r	$4 \times N_{PDU\_S}$
TFQ	i/r	–	w	$2 \times N_{PDU\_S} \times ((L_{PDU\_S} - 5 + L_{TB})//(L_{TB} - 4))$
TRQ	r	–	w	$2 \times N_{PDU\_S}$

Note 1 // denotes integer division.

Note 2 The above calculations have been simplified by leaving the transparent mode and the cell FIFO mode out of consideration.

Note 3 In the above table only the number of effectively used memory locations are listed, as functions of the parameters. However, the size required by some tables may be considerably higher, as in the case of RVT, where the actual size is heavily dependent on the random values taken by VPI/VCI, when more than 32 simultaneous virtual connections are processed. Also, it is recalled that the sizes of the queues are always powers of 2.

L_{RB} and L_{TB} are determined by the BUFSIZE system parameter and are both equal to 64, 128, 256, 512, 1024, 2048, 4096 or 8192 bytes.

To illustrate the memory requirements, the following simplifications are used:

– No multiplexing of PDUs within one virtual connection in reassembly or segmentation (AAL3/4 MID) is performed, so that:

$N_{PDU\_S} = N_{SEG}$ and $N_{PDU\_R} = N_{RAS}$.

– An equal number of PDUs are prepared for segmentation, or are currently in reassembly

$N_{SEG} = N_{RAS} = N$.

Tables 27-29 give the memory requirements for data (total of receive + transmit data buffers RB + TB) and for control structures.

In the calculations, the memory requirements use the notion of "number of PDUs" which, in the case of segmentation, can be superior to the number of virtual connections, even in the case of AAL5. The reason is obviously that the number of PDUs per virtual connection prepared for segmentation can be arbitrary for the SARE, e.g. to obtain a sustained cell rate in a high speed connection.

All values in **tables 27-29** are in Bytes. The first value, shown in italics, is the total size of control data. The second value, shown plain, is the total memory space taken by data buffers (RBs and TBs).

Table 27 Buffer Size (RB and TB) = 64 Bytes

Average PDU Size	Number of PDUs						
	16	**32**	**64**	**128**	**256**	**512**	**1024**
32 Bytes	*960* 2048	*1920* 4096	*3840* 8192	*7680* 16384	*15360* 32768	*30720* 65536	*61440* 131072
160 Bytes	*1088* 6144	*2176* 12288	*4352* 24576	*8704* 49152	*17408* 98304	*34816* 196608	*69632* 393216
800 Bytes	*1792* 28672	*3584* 57344	*7168* 114688	*14336* 229376	*28672* 458752	*57344* 917504	*114688* 1835008
4000 Bytes	*5184* 137216	*10368* 274432	*20736* 548864	*41472* 1097728	*82944* 2195456	*165888* 4390912	*331776* 8781824

Table 28 Buffer Size (RB and TB) = 256 Bytes

Average PDU Size	Number of PDUs						
	16	**32**	**64**	**128**	**256**	**512**	**1024**
32 Bytes	*960* 8192	*1920* 16384	*3840* 32768	*7680* 65536	*15360* 131072	*30720* 262144	*61440* 524288
160 Bytes	*960* 8192	*1920* 16384	*3840* 32768	*7680* 65536	*15360* 131072	*30720* 262144	*61440* 524288
800 Bytes	*1152* 32768	*2304* 65536	*4608* 131072	*9216* 262144	*18432* 524288	*36864* 1048576	*73720* 2097152
4000 Bytes	*1920* 65536	*3840* 131072	*7680* 262144	*15360* 524288	*30720* 1048576	*61440* 2097152	*122880* 4194304

Table 29 Buffer Size (RB and TB) = 2048 Bytes

Average PDU Size	Number of PDUs						
	16	**32**	**64**	**128**	**256**	**512**	**1024**
32 Bytes	*960* 65536	*1920* 131072	*3840* 262144	*7680* 524288	*15360* 1048576	*30720* 2097152	*61440* 4194304
160 Bytes	*960* 65536	*1920* 131072	*3840* 262144	*7680* 524288	*15360* 1048576	*30720* 2097152	*61440* 4194304
800 Bytes	*960* 65536	*1920* 131072	*3840* 262144	*7680* 524288	*15360* 1048576	*30720* 2097152	*61440* 4194304
4000 Bytes	*1024* 131072	*2048* 262144	*4096* 524288	*8192* 1048576	*16384* 2097152	*32768* 4194304	*65536* 8388608

Note: To ensure correct operation of the SARE in spite of PCI bus latencies, the following control data should be located on-chip or in external local memory: RFQ, RVT, RFQ, TVT, and TPT (and RMT in the case of AAL3/4 with channel multiplexing).

5.4 Operational Description of Reassembly Procedures

5.4.1 Initialization

After Reset the SARE reception is blocked, so that reception and spurious writing in the memory by the SARE is not possible before the initialization procedure is complete. Reassembly should only be enabled (via bit C_RASEN in CONFIG register) when initialization is completed.

- Initialize hardware registers:
  - Set BUFSIZE register
  - Initialize the base address registers (RMTB, RBB, RFQB, RMQB and RRQmB)
  - Set the queue length registers (RFQL, RMQL and RRQmL); this also sets the queue pointers automatically to 0
  - Set COSETR register
  - Write a Receive Buffer Descriptor pointer in OAMF5D register
- Initialize RFQ and RB:
  - Write RFQ with free RBDs
  - Form a linked list of RBDs destined to OAM-F5, Resource Management and ATM Layer reserved cells
  - Write WRPTR for RFQ into RFQWR
  - Write last double-word of every RB to 0.

(For AAL3/4 or SMDS:)

- Initialize RMQ and RMT:
  - Initialize RMQ
  - Fill RMT with 0s
  - Initialize SMID.
- Initialize RVF and RVT
  - Set VCIRANGE = 15 for unused RVF groups
  - Set VCILO, VCIRANGE, VPILO and VPIHI to desired values in used RVF groups
  - Initialize RVT entries of used RVF groups
- Initialize Interrupt registers
  - Mask register settings
  - Program desired interrupt interval
- Initialize CONFIG register

5.4.2 Setting up a Virtual Connection

- Initialize RVT entry:
  - Read RDPTR from RFQWR
  - Use base address RFQB and RDPTR to read RBD from RFQ
  - Read required RVF and calculate from RVTB, VPILO, VPIHI, VCILO and VCIRANGE the address of the RVT entry
  - Write RBD along with other parameters into RVT entry

5.4.3 PDU Receive

When a PDU has been reassembled, an entry is made by the SARE in RRQm and an I_RRQm_READY interrupt status is generated. Software reaction:

- Get RBD from RRQm:
  - Jump to interrupt service routine
  - Read RDPTR from RRQmGET (auto-incremented), or
  - Read RDPTR from RRQmRD, increment RDPTR and write back
  - Use base address RRQmB and RDPTR to read RBD from RRQm
- Read RRH:
  - Use base address RBB and RBD to read RRH (3 32-bit words) from RB
  - Check RERR and CRC10
  - Pass ATM cell header to internal software structures
  - (For AAL3/4 or SMDS: Check MID)
  - Pass ATM cell header to internal software structures
- Read RB data:
  - Read data from first RB and pass them on
  - Read RBD from end of RB, go to next RB
- When PDU data completely read (indicated by PLEN), release RBs:
  - Read WRPTR from RFQWR
  - Use base address RFQB and WRPTR to write RBDs to RFQ
  - Increment WRPTR and write back to RFQWR

5.4.4 Tearing Down a Virtual Connection

- Deactivate RVT entry
  - Read required RVF entry
  - Calculate from RVTB, VPILO, VPIHI, VCILO and VCIRANGE the address of the RVT entry
  - Set AALS to 0.

5.5 Operational Description of Segmentation Procedures

5.5.1 Initialization

Before a virtual connection is established and segmentation is started, the software has to:

- initialize the base address registers (TPTB, TVTB, TBB, TRQB, TWQB and TFQB) and configure the STRUCT register;
- set up a list of free TBs in the Transmit Buffer Free Queue TFQ
- set up a list of free TPTs in the Transmit Ready Queue TRQ
- mark the entries in TVT as free:
  - set up a list of free TVDs (internally)
  - set TVT.TPD = 0
    TVT.CGCNT = 0
- mark the entries in TPT as free:
  - set up a list of free TPDs (internally)
  - set TPT.STATE = 0
    TPT.TBD_OFFSET = 0
    CRC32 = – 1.

At the end of the initialization, segmentation is enabled by setting C_SEGEN in CONFIG register to “1”.

5.5.2 Setting up a Virtual Connection

When call set-up signalling is completed, the TVT entry is initialized:

- Initialize Manager parameters (CREDIT, CRMAX, CRDEC, CGSEL, CRMAXE)
- Set congestion control parameter CGSEL
- Select Credit Manager Timer via NTMR
- Program VCI, VPI
- Include the TVT entry in the linked list of the selected Timer

5.5.3 PDU send

- Write PDU payload in TBs:
  - Get TB from TFQ:
    Read RDPTR from TFQGET (auto-incremented), or

    Read RDPTR from TFQRD, increment RDPTR and write back

    Use base address TFQB and RDPTR to read TBD from TFQ
  - Write TB (address via base address TBB and TBD) with data
  - Repeat for as many TBs as necessary
  - Link TBs via TBD_NEXT pointer

- Write TPT for the PDU:
  - Get TPD from internal list of free TPDs
  - Write TPT entry (address via base address TPTB and TPD):
    Set TBD to address of first TB in list

    Set TPT.MODE to desired AAL type

    Set PDUSIZE

    Set ATM Cell Header (except VPI/VCI)

    (for AAL3/4 or SMDS: set MID; set SN = 0)
- Release PDU for segmentation via Transmit Waiting Queue TWQ:
  - Read WRPTR from TWQPUT/TWQWR
  - Use base address TWQB and WRPTR to write TVD/TPD into TWQ
  - In case WRPTR was obtained from TWQWR, increment WRPTR and write back

5.5.4 End of PDU Segmentation

When segmentation of a PDU is completed, the SARE writes the TPD address into TRQ and, optionally, generates an interrupt (I_TRQ_READY). Possible software reaction:

- Get TPD from TRQ:
  - Read RDPTR from TRQGET (auto-incremented), or
  - Read RDPTR from TRQRD, increment RDPTR and write back
  - Use base address TRQB and RDPTR to read TPD from TRQ
  - Include TPD into internal list of free TPDs
  - Update internal list of PDUs still in segmentation in that virtual connection (book-keeping)

5.5.5 Tearing Down a Virtual Connection

- Check that all PDUs for the virtual connection are completely segmented (book-keeping via TRQ, as described above)
- Mark the TVT entry as free:
  - include the TVT entry in the internal list of free TVDs
  - set TVT.TPD = 0
    TVT.CGCNT = 0

6 Register Descriptions

6.1 PCI Interface Registers

Table 30 gives the list of configuration registers in the PCI interface and provides a brief description for each.

Table 30 PCI Interface Register Descriptions

Register	Description
Vendor ID	Identifies the manufacturer of the PCI compatible part
Device ID	Identifies the device.
Cache line size	Size of the cache line (not relevant for the SARE)
Latency timer	Maximum burst duration in Double Words. Set by the system controller. Typical value according to PCI Design Guide is 66 Double Words, or 1.98 µs at 33 MHz. Theoretical maxim um value is 256 Double Words, or 7.68 µs. For example, for 5 devices and a Latency Timer set to 66 DW, the device with lowest priority will have access to the bus within 4 × 1.98 µs = 7.92 µs max.
Header type	According to PCI Specification, only a value = 00_H is allowed.
Base address register n/ Mapping address register n	Registers for defining the mapping of address space n. The address range and thereby the number of valid bits for com parison are fixed for every address space.
Local base address register	Base address for user defined registers
Interrupt line	Defined by the system, defines the correspondence between the interrupt pins of the device and the pins of the interrupt controller.
Interrupt pin	Selects one or several of the interrupt pins $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$ or $\overline{\text{INTD}}$ to be used. Programmed for $\overline{\text{INTA}}$ in the SARE.
Min_Gnt	Contains the desired value for the Latency Timer (in units of 250 ns).
Max_Lat	Maximum waiting time between activation of $\overline{\text{REQ}}$ and activation of $\overline{\text{GNT}}$.
Burst duration	Same as Min_Gnt, but units = bus cycles (Double-Word)
Back-Off	Time between two retries.

Table 31 provides a complete specification for each of the PCI Interface Registers, including its address, initialization value, and reset value.

Table 31 PCI Interface Register Specifications

Register Name	Byte Address	Byte 3	Byte 2	Byte 1	Byte 0	Init. Value	Reset Value	Comment
Vendor ID	00-01$_H$					–	110A$_H$	Hardwired to 110A$_H$
Device ID	02-03$_H$					–	3583$_H$	Hardwired to 3583$_H$
Command	04-05$_H$					032F	0000$_H$	Bit 4 hardwired to 0
Status	06-07$_H$					0000$_H$	0000$_H$	
Revision ID	08$_H$					–	01$_H$	Hardwired
Class code	09-0B$_H$					–	028000$_H$	Hardwired to 028000$_H$
Cache line size	–							Not implemented (only for special write)
Latency timer	0D$_H$					60$_H$	00$_H$	Bits 0-2 hardwired to 0
Header type	0E$_H$					–	00$_H$	Hardwired
BIST	–							Not implemented
Base address register 0	10-13$_H$					Variable	00000000$_H$	Note 1
Base Address Register 1	14-17$_H$					Variable	00000000$_H$	Note 2
Local base address register	18-1B$_H$							Not implemented
Expansion ROM base address	–							Not implemented
Interrupt line	3C$_H$						FF$_H$	
Interrupt pin	3D$_H$					01$_H$	01$_H$	Use $\overline{\text{INTA}}$
Min_Gnt	3E$_H$					0D$_H$	00$_H$	Typical: 3.25 µs
Max_Lat	3F$_H$					27$_H$	00$_H$	Typical: 9.75 µs
Mapping address register 0	40-43$_H$					Variable	00000000$_H$	
Mapping address register 1	44-47$_H$					Variable	00000000$_H$	
Address step size	48$_H$					–	00$_H$	Hardwired
Data step size	49$_H$					–	00$_H$	Hardwired
Burst duration	4A$_H$					60$_H$	00$_H$	Programmable
Back-off	4B$_H$					00$_H$	00$_H$	

Note 1 The Base Address Register 0 (BAR0) is the PCI base address for on-chip SARE memory, external local memory and peripheral device (connected to the SARE external local mem ory interface) memory space. The site of this space is 4 MDW (16 MBytes); thus the location is determined by bits 25-31 of the BAR0 register.

Note 2 The Base Address Register 1 (BAR1) is the PCI base address for the on-chip SARE configuration and status registers. The size of this space is 256 DW (1024 Bytes), thus the location is determined by bits 10-31 of the BAR1 register.

6.2 Configuration, Control and Status Registers

The control, configuration and status registers are read and/or written via 32-bit accesses from the PCI interface (with the SARE as a PCI slave device). They are located in a contiguous area starting at a programmable address contained in the PCI configuration register "Base Address Register 0":

Register address = (Bits 31 to 2 of Base Address Register 0) + Offset,

all in 32-bit words. The offset addresses of the registers are summarized in **table 32**.

Generally, the values of all "reserved" register bits are undefined when read, and, unless indicated otherwise, should be written at "0" to ensure proper operation.

Table 32 Configuration, Control, and Status Register Descriptions

Register Name	Double-Word Address hex (dec)	Access from Host	Access from SARE	Description/Notes
TmTVD m = 0 to 15	00-0F_H (0-15)	R/W	R/W	SARE has access priority over the host
TmTVDPREV m = 0 to 15	10-1F_H (16-31)	R/W	R/W	See TmTVD
TmPHASE m = 0 to 15	20-2F_H (32-47)	R/W	R/W	See TmTVD
TmK m = 0 to 15	30-3F_H (48-63)	R/W	R/W	See TmTVD
TmL m = 0 to 15	40-4F_H (64-79)	R/W	R/W	See TmTVD
TmK0 m = 0 to 15	50-5F_H (80-95)	R/W	R	See TmTVD
RVTBx x = 0 to 15	60-6F_H (96-111)	R/W	R	SARE has access priority over the host
VPCIx x = 0 to 15	70-7F_H (112-127)	R/W	R	See RVTBx

Table 32 Configuration, Control, and Status Register Descriptions (cont'd)

Register Name	Double-Word Address hex (dec)	Access from Host	Access from SARE	Description/Notes
ECH	80_H (128)	R/W	R	Value 00000001_H after Reset
ECPAT	81_H (129)	R/W	R	Value $6A_H$ after Reset
OAMF5D	82_H (130)	R/W	R	
RMTB	85_H (133)	R/W	R	
RBB	86_H (134)	R/W	R	
RFQB	87_H (135)	R/W	R	
RMQB	88_H (136)	R/W	R	
RRQmB m = 0 to 7	89- 90_H (137-144)	R/W	R	
BUFSIZE	91_H (145)	R/W	R	
SMID	92_H (146)	R/W	R	
RFQL	93_H (147)	R/W	R	A write from host clears RFQWR (address $9D_H$/157)
RMQL	94_H (148)	R/W	R	A write from host clears RMQWR (address $9E_H$/158)
RRQmL m = 0 to 7	95-$9C_H$ (149-156)	R/W	R	A write from host clears RRQm GET (address 9E-$A6_H$/159-166) and RRQmRD (address A7-AE_H/167-174)
RFQWR	$9D_H$ (157)	R/W	R/W	The host can write the lower 16 bits (at a maximum, depending on the RFQ length, as determined by RFQL), the upper 16 bits have no effect. SARE can only write the higher 16 bits.
RMQWR	$9E_H$ (158)	R/W	R/W	See RFQWR (number of writable bits determined by RMQL)

Table 32 Configuration, Control, and Status Register Descriptions (cont'd)

Register Name	Double-Word Address hex (dec)	Access from Host	Access from SARE	Description/Notes
RRQmGET m = 0 to 7	9F-A6$_H$ (159-166$_H$)	R	R/W	Host read-only. A write from host has no effect. After a read from the host the contents of RRQmGET (Read Pointer value) are automatically increm ented. The SARE can only write the higher 16 bits.
RRQmRD m = 0 to 7	A7-AE$_H$ (167-174)	R/W	R/W	As RRQmGET, but: - Write from host possible on the lower 16 bits (number of significant bits determined by RRQmL). No automatic incrementation on read from host.
CONF	AF$_H$ (175)	R/W	R/W	Cleared after Reset. Reserved bits should be written at "0". When bit 31 (Reset) is written with a "1" by the host, it is autom atically cleared by the SARE 3 clock cycles thereafter.
COSETR	B0$_H$ (176)	R/W	R	Initialized to 55$_H$ after Reset
COSETS	B1$_H$ (177)	R/W	R	Initialized to 55$_H$ after Reset.
TIMESTAMP	B2$_H$ (178)	R	R/W	Counter – Read-only from host. Initialized to 1 after Reset.
IMASKA	B3$_H$ (179)	R/W	R	Cleared after Reset.
IMASKB	B4$_H$ (180)	R/W	R	Cleared after Reset.
INTSTATE	B5$_H$ (181)	R/W	R/W	Written by SARE when an interrupt status occurs. Reserved bits are read at "0". Cleared after Reset.
I_MIN_PERIOD	B7$_H$ (183)	R/W	R	
S_CELLS	B8$_H$ (184)	R/W	R/W	32-bit counter. Cleared after Reset.
S_CELLSEG	B9$_H$ (185)	R/W	R/W	See S_CELLS.
S_HECE	BA$_H$ (186)	R/W	R/W	See S_CELLS.

Table 32 Configuration, Control, and Status Register Descriptions (cont'd)

Register Name	Double-Word Address hex (dec)	Access from Host	Access from SARE	Description/Notes
S_CRC10E	BB_H (187)	R/W	R/W	See S_CELLS.
S_RCVD	BC_H (188)	R/W	R/W	See S_CELLS.
TPTB	BD_H (189)	R/W	R	
TVTB	BE_H (190)	R/W	R	
TBB	BF_H (191)	R/W	R	
TWQB	$C0_H$ (192)	R/W	R	
TRQB	$C1_H$ (193)	R/W	R	
TFQB	$C2_H$ (194)	R/W	R	
TWQL	$C3_H$ (195)	R/W	R	A read of TWQL clears TWQPUT (address $C6_H$/198) and TWQWR ($C9_H$/201).
TRQL	$C4_H$ (196)	R/W	R	A read of TRQL clears TRQGET (address $C7_H$/199) and TRQRD (address CA_H/202).
TFQL	$C5_H$ (197)	R/W	R	A read of TFQL clears TFQGET (address $C8_H$/200) and TFQRD (address CB_H/203).
TWQPUT	$C6_H$ (198)	R	R/W	Host read-only. A write from host has no effect. After a read from the host, the contents of TWQPUT are automatically incremented (if upper 16 bits are not all zero). The SARE can only write the higher 16 bits.
TRQGET	$C7_H$ (199)	R/W	R/W	After a read from the host the contents of TWQGET are autom atically incremented (if upper 16 bits are not all zero). If required, the host can set the higher 16 bits (which contain the SARE internal value of the TRQ write pointer) via a write of TRQGET.
TFQGET	$C8_H$ (200)	R/W	R/W	See TRQGET.

Table 32 Configuration, Control, and Status Register Descriptions (cont'd)

Register Name	Double-Word Address hex (dec)	Access from Host	Access from SARE	Description/Notes
TWQWR	$C9_H$ (201)	R/W	R/W	As TWQPUT; but, write from host possible on the lower 16 bits (number of significant bits determined by TWQL). No automatic incrementation on read from host.
TRQRD	CA_H (202)	R/W	R/W	As TRQGET; but, write from host possible on the lower 16 bits (number of significant bits determined by TRQL). No automatic incrementation on read from host.
TFQRD	CB_H (203)	R/W	R/W	As TFQGET; but, write from host possible on the lower 16 bits (number of significant bits determined by TFQL). No automatic incrementation on read from host.
CELLCNT	CC_H (204)	R	R/W	Internal counter. Cleared after Reset.
STRUCT	CD_H (205)	R/W	R	DMA Controller Structure Register for the different data types.
RVTBx x = 16 to 31	$E0\text{-}EF_H$ (224-239)	R/W	R	SARE has access priority over the host.
VPCIx x = 16 to 31	$F0\text{-}FF_H$ (240-255)	R/W	R	See RVTBx.

6.2.1 Credit Manager Timer Registers

Table 33 provides a brief description of each of the Credit Manager Timer Registers. It is followed by the specification for each.

Table 33 Credit Manager Timer Register Summary

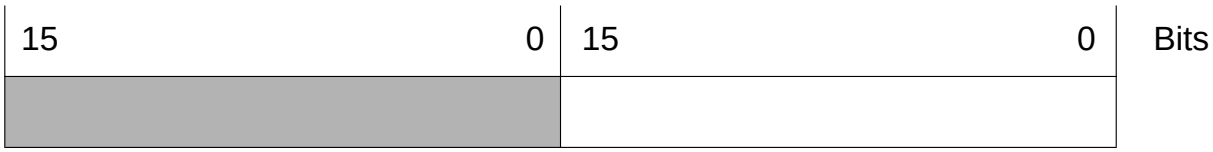
Name	Description
TmTVD	TVD Descriptor/Pointer to the next TVT entry in chained list
TmTVDPREV	TVD Previous/Pointer to the previously processed TVT entry in chained list
TmPHASE	Internal state of the bit-rate timer
TmK	Nominator of the bit-rate quotient
TmL	Denominator of the bit-rate quotient
TmK0	Increment/Decrement for TmK when virtual channel is added/removed, respectively

TmTVD m = 0 to 15

Address: 00-0F$_H$ (0-15)

Access: R/W

Register Bits:

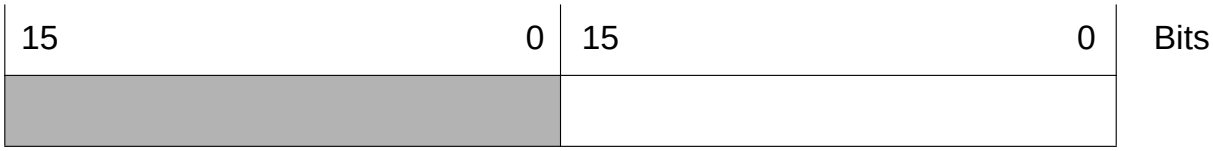


TmTVDPREV m = 0 to 15

Address: 10-1F$_H$ (16-31)

Access: R/W

Register Bits:

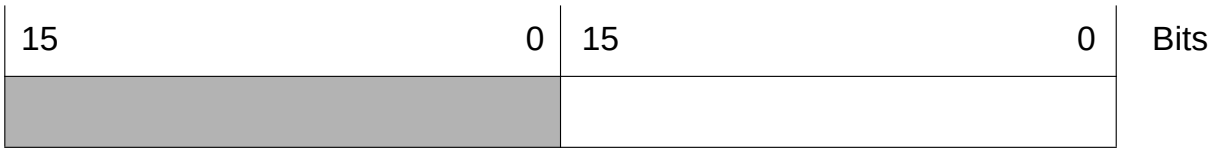


TmPHASE m = 0 to 15

Address: 20-2F$_H$ (32-47)

Access: R/W

Register Bits:

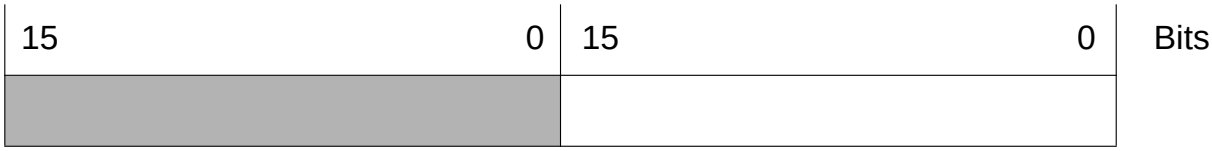


TmK m = 0 to 15

Address: 30-3F$_H$ (48-63)

Access: R/W

Register Bits:

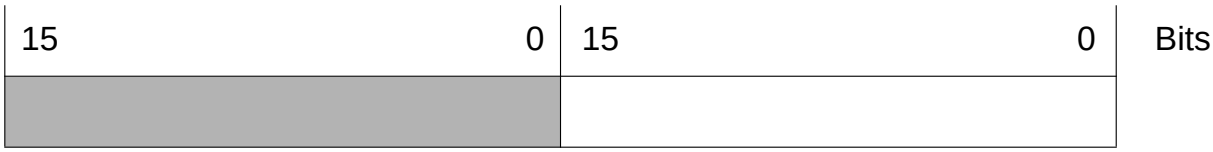


TmL m = 0 to 15

Address: 40-4F$_H$ (64-79)

Access: R/W

Register Bits:

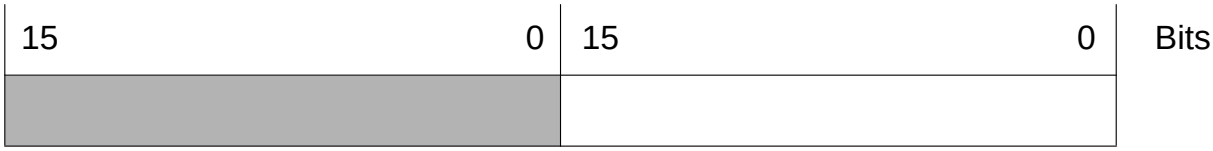


TmK0 m = 0 to 15

Address: 50-5F$_H$ (80-95)

Access: R/W

Register Bits:



6.2.2 Receive VPCI Filter Registers

The RVTBx and the VPCIx comprise the Receive Filter Registers.

RVTBx x = 0 to 31

Address: 60-6F$_H$ (96-111), E0-EF$_H$ (224-239)

Access: R/W

Register Bits:

31 29	28	27 24	23 0	Bits
Rsvd	OnChip	VCIRANGE	RVTB	

RVTB **RVT Base Address**

VCIRANGE **Range of Values taken from VCI**

Determines the range of values that can be taken by VCI (for a given VPI), and the size of the VCI table within the RVT for any given VPI.If VCIRANGE = 15, this implies that the corresponding entry is not valid (virtual connection or group not active).

OnChip **RVT On-Chip**

OnChip = 1 The RVT entry for this virtual connection is on-chip

OnChip = 0 The RVT entry for this virtual connection is in external memory (or, possibly, on-chip, depending on the result of the calculation of the expression for the RVT_Address below).

Rsvd **Reserved**

When read, the value of this bit position is undefined. It should always be written at “0”.

VCPIx x = 0 to 31

Address: 70-7F$_H$ (112-127), F0-FF$_H$ (240-255)

Access: R/W

Register Bits:

31 24	23 16	15 0	Bits
VPILO	VPIHI	VCILO	

VCILO **VCI Low**

VPIHI **VPI High**

VPILO **VPI Low**

The byte address of the RVT entry for the connection inside this group is given by:

$$\text{RVT-Address} = 256 \times \text{RVTB} + 2^{\text{VCIRANGE+4}} \times (\text{VPI-VPILO}) + 16 \times (\text{VCI-VCILO}).$$

6.2.3 Empty (Idle) Cell Registers

ECH

Address: 80_H (128)

Access: R/W

Register Bits:

31 24	23 16	15 8	7 0	Bits
Header (Byte 1)	Header (Byte 2)	Header (Byte 3)	Header (Byte 4)	

Header Byte 1 **1st header byte of idle cell**

Default value: 00_H

Header Byte 2 **2nd header byte of idle cell**

Default value: 00_H

Header Byte 3 **3rd header byte of idle cell**

Default value: 00_H

Header Byte 4 **4th header byte of idle cell**

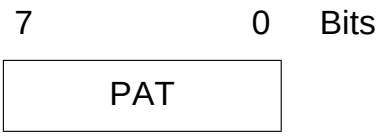
Default value: 01_H

ECPAT

Address: 81_H (129)

Access: R/W

Register Bits:



PAT **Empty cell payload pattern**

Default value: $6A_H$

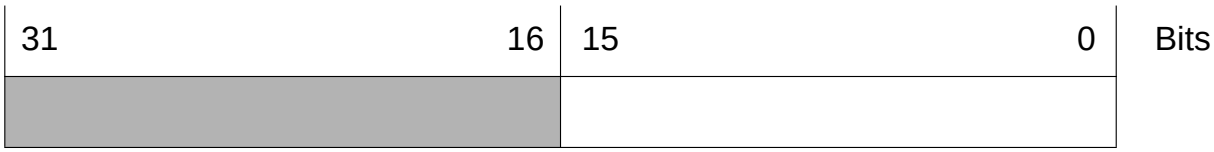
6.2.4 OAM-F5 Cell FIFO Descriptor

OAMF5D

Address: 82_H (130)

Access: R/W

Register Bits:



This register gives the Receive Buffer Descriptor (Pointer) of the memory area where the OAM-F5, Resource Management and ATM-Layer reserved cells are to be stored, if C_OAMM = 0 (i.e., if these cells are received in the cell FIFO mode and not in the transparent mode).

6.2.5 Base Addresses for Receive Data Structures

The base addresses are 24 bits wide and represent the 24 most significant bits of the address to which an offset is added to get the final address in the data structure:

byte_address = base_address $\times$ 256 + offset.

RMTB

Address: 85_H (133)

Access: R/W

Register Bits:

The byte address in the Receive MID Table as a function of the Size of MID (SMID) parameter, the Receive MID Table Descriptor (RMD) and the received MID is:

RMT Byte Address = $256 \times$ RMTB + $2^{SMID+4} \times$ RMD + $16 \times$ MID.

RBB

Address: 86_H (134)

Access: R/W

Register Bits:

The byte address in the Receive Buffer as a function of the Receive Buffer Descriptor (RBD from RVT or from RFQ), BUFSIZE (size of receive/transmit buffers) and of the current payload offset CPOF (in RRH) is:

RB Byte Address = $256 \times$ RBB + $2^{BUFSIZE + 6} \times$ RBD + $4 \times$ CPOF.

RFQB

Address: 87_H (135)

Access: R/W

Register Bits:

The byte address in the Receive Buffer Free Queue as a function of the read and write pointers in RFQWR is:

RFQ Byte Address = 256 × RFQB + 2 × RDPTR (read queue)

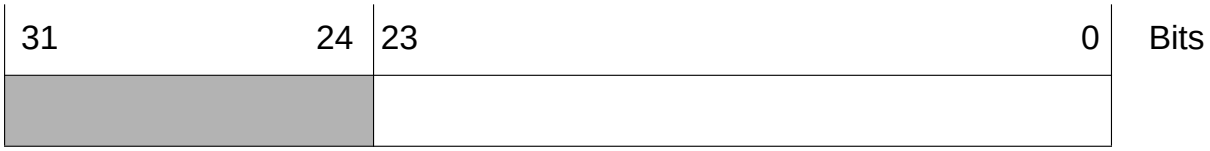
RFQ Byte Address = 256 × RFQB + 2 × WRPTR (write queue).

RMQB

Address: 88_H (136)

Access: R/W

Register Bits:



The byte address in the Receive MID Table Queue as a function of the read and write pointers in RMQWR is:

RMQ Byte Address = 256 × RMQB + 2 × RDPTR (read queue)

RMQ Byte Address = 256 × RMQB + 2 × WRPTR (write queue).

RRQmB m = 0 to 7

Address: $89\text{-}90_H$ (137-144)

Access: R/W

Register Bits:

The byte address in the Receive Ready Queue m (m = 0,...,7) as a function of the read and write pointers in RRQmGET/RRQmRD is:

RRQm Byte Address = 256 × RRQmB + 2 × RDPTR (read queue)

RRQm Byte Address = 256 × RRQmB + 2 × WRPTR (write queue).

6.2.6 Buffer Size and MID Table Size Registers

BUFSIZE

Address: 91_H (145)

Access: R/W

Register Bits:

The size of the receive and transmit buffers as a function of BUFSIZE is given in the **table 34**.

Table 34 Buffer Sizes

Value of BUFSIZE	Buffer Size (Bytes)
000	64
001	128
010	256
011	512
100	1024
101	2048
110	4096
111	8192

SMID

Address: 92_H (146)

Access: R/W

Register Bits:

SMID (Size of MID, value between 0 and 10) gives the number of significant bits in the MIDs. It simultaneously gives the size of the section in the RMT per virtual connection.

6.2.7 Receive Queue Management Registers

RFQL

Address: 93_H (147)

Access: R/W

Register Bits:

31 … 4	3 … 0	Bits
	RFQL	

The maximum length of the Receive Buffer Free Queue is given by:

$N_{RFQ} = 2^{RFQL+1} - 1$ 16-bit entries,

where the maximum value of RFQL is 15.

RFQWR

Address: $9D_H$ (157)

Access: R/W

Register Bits:

31 … 16	15 … 0	Bits
RDPTR	WRPTR	

A write to RFQL clears RFQWR.

RDPTR contains the pointer to the current SARE read location in RFQ (updated by SARE), and WRPTR the pointer of the current host write location (to be updated by the host). When the host enters pointers (RBDs) to released Receive Buffers in the RFQ, it first reads WRPTR from RFQWR. After having entered the RBD(s) into the RFQ, it increments WRTPR by the number of new RFQ entries. Bits 31 to 16 have no significance when writing RFQWR.

RMQL

Address: 94_H (148)

Access: R/W

Register Bits:

31 … 4	3 … 0	Bits
	RMQL	

The maximum length of the Receive MID Table Queue is given by:

$N_{RMQ} = 2^{RMQL+1} – 1$ 16-bit entries,

where the maximum value of RMQL is 15.

RMQWR

Address: $9E_H$ (158)

Access: R/W

Register Bits:

31 … 16	15 … 0	Bits
RDPTR	WRPTR	

A write to RMQL clears RMQWR.

When a new Receive MID Table Descriptor (RMD) is needed, the SARE reads the next entry from RMQ from address RDPTR if $\overline{\text{RDPTR}}$ WRPTR, and increments RDPTR by 1 (modulo N_{RMQ}+1). The SARE enters into RMQ at address WRPTR RMDs which are no more needed (i.e. if no PDU in a given virtual connection is being segmented) and increments WRPTR correspondingly (modulo N_{RMQ}+1).

The host can read RMQWR for testing purposes or for statistics, but should not write to this register when the SARE is in operation. Before operation, RMQWR should be initialized by the host to contain the actual number of RMDs.

RRQmL m = 0 to 7

Address: 95-9C$_H$ (149-156)

Access: R/W

Register Bits:

31	4	3	0	Bits

The maximum length of a Receive Ready Queue is given by:

$N_{RRQm} = 2^{RRQmL+1} – 1$ 16-bit entries,

where the maximum value of RRQmL is 15.

RMQmGET m = 0 to 7

Address: 9F-A6$_H$ (159-166$_H$)

Access: R

Register Bits:

31	16	15	0	Bits
WRPTR		RDPTR		

RRQmRD m = 0 to 7

Address: A7-AE$_H$ (167-174)

Access: R

Register Bits:

31	16	15	0	Bits
WRPTR		RDPTR		

Both RRQmGET and RRQmRD contain the read and write pointers of the corresponding Receive Ready Queue. The SARE enters the Receive Buffer Descriptor (pointer) of the first buffer of a completely reassembled PDU in these queues. Alternatively, if queue n has been programmed in the streaming (or “early warning reception”) mode of operation (for n between 0 and 3), then RRQnGET/RRQnRD are used to report completely reassembled PDUs in queue RRQn, whereas RRQ(n+4)GET/RRQ(n+4)RD are used to

report the arrival of the first cell of a PDU, or the completion of each RB, or both (see CONF register description).

After the entry of the RBD in RRQm at the address given by WRPTR, the write pointer WRPTR is incremented by 1 (modulo N_{RRQm} + 1). When

WRPTR - $\overline{RDPTR}$ 0 (mod N_{RRQm} + 1),

the host can read the RBD(s) using the RDPTR pointer either from RRQmGET or from RRQmRD. The only difference is that when RRQmGET is read, the read pointer RDPTR is automatically incremented by 1. When the read pointer RDPTR is read from RRQm RD, on the other hand, it is not automatically incremented, but the host is required to increment the RDPTR by the number of RBDs taken from the corresponding RRQ (m odulo N_{RRQm} + 1).

6.2.8 Configuration Register

CONF

Address: AF_H (175)

Access: R/W

Register Bits: See also **table 35**.

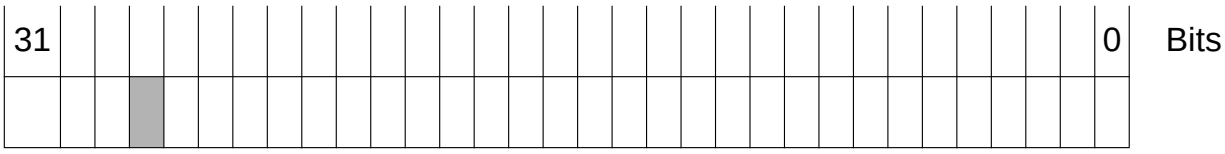


Table 35 CONF Register Bit Descriptions

Bit	Name	Description
0	C_SEGEN	**Segmentation Enable** When this bit is set to “1”, segmentation is enabled.
1	C_RASEN	**Reassembly Enable** When this bit is set to “1”, reassembly is enabled.
2	C_UTOPEN	**UTOPIA Enable** When this bit is set to “1”, the UTOPIA interface is enabled.
3	C_INTEN	**Interrupt Enable** When this bit is set to “1”, the generation of interrupts is enabled.
4	C_HECC	**HEC Control** HEC check is performed on the cell header of incoming cells if C_HECC = 1. No HEC check is performed if C_HECC = 0.

Table 35 CONF Register Bit Descriptions (cont'd)

Bit	Name	Description
5	C_UACEN	**Unassigned Cells Enable** If C_UACEN is "0", unassigned cells (or empty cells, Cf ATM Forum UNI Specification) are unconditionally ignored. If C_UACEN is "1", they are checked by the receiver: if the address matches the combination VPI = 0/VCI = 0 (enabled in the Receive VPCI Filter RVF), they are accepted and stored in the RBs.
6	C_PHOAMEN	**PHY Layer and OAM Cell Enable** If C_PHOAMEN is "0", OAM F1-F3 and "PHY layer reserved" cells are discarded. If C_PHOAMEN is "1", these cells are checked by the receiver: if the address matches the combination VPI = 0/VCI = 0 (enabled in the Receive VPCI Filter RVF), they are accepted and stored in the RBs.
7	C_OAMM	**OAM Mode** When C_OAMM is "0", OAM F5, Resource Management and ATM Layer reserved cells (which use the same address combination as the user cells, but are recognized by their PT field) are handled in the Cell FIFO mode. When C_OAMM is "1", these cells are handled in the Transparent mode: in this case the Receive Ready Queue 0 (RRQ0) is used to indicate their arrival.
8	C_TSRES	**Timestamp Resolution** The resolution of the timestamps is either 100 µs (if C_TSRES = 0) or 10 µs (if C_TSRES = 1).
9	C_LOOPBACK	**Loopback Control** When C_LOOPback is set to "1", segmented cells are looped back to the receiver (transparent loop, i.e. cells also transmitted on the UTOPIA interface), and cells received on the UTOPIA interface are ignored.
10	C_PEODD	**Parity Even/Odd** Determines the polarity of the byte parity bits LPAR(3:0) on the Local Memory Interface (0: even, 1: odd). A parity error can be indicated via an interrupt status.

Table 35 CONF Register Bit Descriptions (cont'd)

Bit	Name	Description
11	C_MIDMUX	**MID Multiplexing Enable** MID Multiplexing on (C_MIDMUX = 1) or off (C_MIDMUX = 0). This global bit affects the segmentation of all those virtual connection types that allow MID m ultiplexing (AAL3/4 and SMDS). If C_MIDMUX is set to "1", PDUs belonging to one and the same virtual connection are segmented interleaved, i.e. one cell from each PDU one after the other, in a cyclical manner. (The software has to make sure that no more than one PDU with the same MID value in one virtual connection are sim ultaneously being segmented.) If C_MIDMUX is "0", only one PDU in a given virtual connection is segmented at a time (i.e. a PDU belonging to one virtual connection is completely segmented before segmentation of another PDU in the same virtual connection is started).
12	C_UTMS	**UTOPIA Master/Slave Configuration** Determines whether the SARE is to operate as a UTOPIA interface master (C_UTMS = 0; NIC applications) or as a UTOPIA interface slave (C_UTMS = 1; ATM Switch/Hub applications). This configuration bit is internally XOR'd with the M/S pin.
13	C_BIST	**Built-In Self Test** When set to 1, this bit activates the built-in self test of the SARE. (This test consists of writing certain bursts of patterns by the on-chip DMA controller to an external mem ory via the PCI Interface.) This bit should always be left at 0 except when diagnostics are to be performed. *Note: The SARE does not check the results of the self-test.*
14	C_LBE	**Little/Big Endian** Indicates the type of host system on the PCI bus and affects the ordering of data bytes in the host memory. 1 = Little Endian; 0 = Big Endian

Table 35 CONF Register Bit Descriptions (cont'd)

Bit	Name	Description
16-15	C_THDMA	**Threshold on the Data FIFO for DMA Transfer** Determines the filling threshold of the 128-double word data FIFO at which execution of DMA transfer will start even if all the data for this DMA has not yet been transferred to the FIFO: C_THDMA = 00 Threshold = 16 Double Words C_THDMA = 01 Threshold = 32 Double Words C_THDMA = 10 Threshold = 64 Double Words C_THDMA = 11 Threshold = 96 Double Words
17	–	**Reserved**
20-18	C_TCKDIV	**TxCLK Division Factor** In the case where the SARE operates as a UTOPIA interface clock master and TxCLK is derived internally from the PCI clock (i.e., pin UCMOD = "high"), C_TCKDIV determines the division factor: C_TCKDIV = 000 Division factor = 1 C_TCKDIV = 001 Division factor = 2 C_TCKDIV = 010 Division factor = 4 C_TCKDIV = 011 Division factor = 8 C_TCKDIV = 100 Division factor = 16 All other values: Reserved
21	C_IDLGEN	**Idle Cell Generation Enable** If C_IDLGEN = 1, idle (empty) cells are generated and transmitted on the UTOPIA interface according to the UTOPIA interface handshake procedure when no other cells are to be transmitted. If C_IDLGEN = 0, no empty cells are generated.
22	C_STMEN0	**Streaming Mode Enable 0** When C_STRMEN0 = 1, the streaming mode (or "early warning reception") for the virtual connections using Receive Ready Queue 0 is enabled. This means that RRQ4 is used to report the arrival of PDUs (either the arrival of the first cell in the PDU, or the completion of every Receive Buffer) belonging to such a virtual connection, whereas RRQ0 is used to report the arrival of the complete PDU.

Table 35 CONF Register Bit Descriptions (cont'd)

Bit	Name	Description
23	C_STMEN1	**Streaming Mode Enable 1** When C_STRMEN1 = 1, the streaming mode (or "early warning reception") for the virtual connections using Receive Ready Queue 1 is enabled. This means that RRQ5 is used to report the arrival of PDUs (either the arrival of the first cell in the PDU, or the completion of every Receive Buffer) belonging to such a virtual connection, whereas RRQ1 is used to report the arrival of the complete PDU.
24	C_STMEN2	**Streaming Mode Enable 2** When C_STRMEN2 = 1, the streaming mode (or "early warning reception") for the virtual connections using Receive Ready Queue 2 is enabled. This means that RRQ6 is used to report the arrival of PDUs (either the arrival of the first cell in the PDU, or the completion of every Receive Buffer) belonging to such a virtual connection, whereas RRQ2 is used to report the arrival of the complete PDU.
25	C_STMEN3	**Streaming Mode Enable 3** When C_STRMEN3 = 1, the streaming mode (or "early warning reception") for the virtual connections using Receive Ready Queue 3 is enabled. This means that RRQ7 is used to report the arrival of PDUs (either the arrival of the first cell in the PDU, or the completion of every Receive Buffer) belonging to such a virtual connection, whereas RRQ3 is used to report the arrival of the complete PDU.
26	C_CLIEN	**Cell Interrupt Enable** When C_CLIEN = 1, an interrupt status after the arrival and storage of the first cell belonging to a PDU in a virtual connection in the streaming mode is generated.
27	C_RBIEN	**Receive Buffer Interrupt Enable** When C_RBIEN = 1, an interrupt status after the completion of every Receive Buffer by data from a PDU in a virtual connection in the streaming mode is generated.
28, 30	–	**Reserved** When read, the value of this bit position is undefined. It should always be written at "0".

Table 35 CONF Register Bit Descriptions (cont'd)

Bit	Name	Description
29	C_BPARK	**Bus Parking** When C_BPARK = 0, the PCI bus is driven during bus parking. When C_BPARK = 1, the PCI bus is not driven during bus parking.
31	C_RESET	**Software Reset** By writing a "1" in this bit, the host causes a software reset of the SARE to occur. This bit is automatically written to "0" again by the SARE when the reset has been performed.

6.2.9 CRC Registers

COSETR

Address: $B0_H$ (176)

Access: R/W

Register Bits:

The pattern programmed in COSETR is subtracted from the calculated CRC before a comparison with the received HEC is made. The value after initialization (Reset) is 55_H, in keeping with the ITU-TS recommendation. Bits 8 to 31 are arbitrary.

COSETS

Address: $B1_H$ (177)

Access: R/W

Register Bits:

The pattern programmed in COSETS is added to the calculated CRC to yield the HEC that is transmitted. The value after initialization (Reset) is 55_H, in keeping with the ITU-TS recommendation. Bits 8 to 31 are arbitrary.

6.2.10 Interrupt Registers

The PCI interface of the SARE has an interrupt line $\overline{\text{INTA}}$ and an auxiliary interrupt line $\overline{\text{INTB}}$. Each line has an individual interrupt mask register, IMASKA and IMASKB, respectively. One of the main purposes of this is to give additional flexibility so that the host software could, e.g. define two interrupt status groups of different priority levels. Every interrupt status source can be made to generate an interrupt on either $\overline{\text{INTA}}$ or $\overline{\text{INTB}}$ (or even both) by setting the corresponding bit in either IMASKA or IMASKB to “1”. The reset status of both mask registers is “all zeros”, i.e. all interrupts are masked.

The interrupt status register INTSTATE is not affected by the mask registers: any active interrupt status can thus be read in INTSTATE at any time. This allows the host to im plement polling of interrupt status, if needed. The bits in the interrupt mask registers only affect the activation of the interrupt line $\overline{\text{INTA}}$ or $\overline{\text{INTB}}$ when the corresponding interrupt status occurs. An interrupt is acknowledged (i.e. the interrupt status bit in INTSTATE is cleared) when a “1” is written into the corresponding bit position in the INTSTATE Register.

IMASKA

Address: $B3_H$ (179)

Access: R/W

Register Bits:

31	0	Bits

IMASKB

Address: $B4_H$ (180)

Access: R/W

Register Bits:

31	0	Bits

INTSTATE

Address: $B5_H$ (181)

Access: R/W

Register Bits:

The meanings of the bits in these registers are shown in **table 36**.

Table 36 Interrupt Register Bit Descriptions

Bit	Name	Description
0-7	I_RRQm_READY m = 0 to 7	**RRQm Ready** An entry has been made in RRQm
8-15	I_RRQm_OVER m = 0 to 7	**RRQm Overflow** RRQm full
16	I_RFQ_UNDER	**RFQ Underflow** RFQ empty/RB full
17	I_RMQ_UNDER	**RMQ Underflow** RMQ empty
18	RESERVED	**Reserved**
19	I_GFC	**GFC Statistics** Number of cells with GFC<>0 exceeded 10 (from 32768)
20	I_PCI_WFULL	**PCI Write FIFO Full**
21	I_PCI_TAB	**PCI Target Abort**
22	I_DMA_OVER	**DMA Channel Overflow**
23	I_UT_PAR	**UTOPIA Parity Error**
24	I_PERI_AB	**Peripheral Device Transfer Abort** A write or read operation to a peripheral (typically PHY) device has been aborted because the external bus – either PCI of the Local External Memory Interface – is needed for a higher priority transfer (originated by the SARE Reassembly Unit).
25	I_RAM_PAR	**Parity Error on External Local Memory Bus**

Table 36 Interrupt Register Bit Descriptions (cont'd)

Bit	Name	Description
26	I_PERI_INT	**Peripheral Interrupt** A peripheral device (typically a PHY device) has activated the PINT input.
27	I_TRQ_Ready	**TRQ Entry**
28	I_TRQ_OVER	**TRQ Overflow**
29	I_TRQ_UNDER	**TRQ Underflow**
30	I_TWQ_OVER	**TWQ Overflow**
31	I_TFQ_UNDER	**TFQ Underflow**

When an interrupt status has been acknowledged, a renewed activation of the interrupt line – due to an already pending unmasked interrupt status, or an interrupt status occurring immediately after the acknowledgment of the previous one – may be delayed by a programmable time value. (This may be useful e.g. in applications where one wishes to continuously guarantee a certain percentage of host processing time to tasks outside the communication task.)

I_MIN_PERIOD

Address: $B7_H$ (183)

Access: R/W

Register Bits:

31 … 5	4	3 … 0	Bits
	I_B_PER	I_N_PER	

I_N_PER, I_B_PER Interrupt Period

Determine the minimum time interval between two consecutive interrupts on either $\overline{\text{INTA}}$ or $\overline{\text{INTB}}$, as follows:
Minimum interval = 7.5 µs × $2^{\text{I\_N\_PER}}$ if I_B_PER = 1
Minimum interval = 75 µs × $2^{\text{I\_N\_PER}}$ if I_B_PER = 0

6.2.11 Statistics Registers

All of the Statistics Registers are 32-bit count-up registers which are cleared upon Reset. **Table 37** provides a description of each register and the specifications for each follow.

Table 37 Summary of Statistic Registers

Name	Description
S_CELLS	Cell clock counter
S_CELLSEG	Number of transmitted non-empty cells
S_HECE	Number of received cells discarded because of an HEC error
S_CRC10E	Number of SAR-CRC-10 errors in AAL3/4 type cells (CRC-10 errors in OAM cells are excluded)
S_RCVD	Number of received cells that belong to an open connection

S_CELLS

Address: $B8_H$ (184)

Access: R/W

Register Bits:

31	0	Bits

S_CELLSEG

Address: $B9_H$ (185)

Access: R/W

Register Bits:

31	0	Bits

S_HECE

Address: BA_H (186)

Access: R/W

Register Bits:

31	0	Bits

S_CRC10E

Address: BB_H (187)

Access: R/W

Register Bits:

31	0	Bits

S_RCVD

Address: BC_H (188)

Access: R/W

Register Bits:

31	0	Bits

6.2.12 Base Addresses for Transmit Data Structures

The base addresses are 24 bits wide and represent the 24 most significant bits of the address to which an offset is added to get the final address in the data structure:

byte_address = base_address × 256 + offset.

TPTB

Address: BD_H (189)

Access: R/W

Register Bits:

The byte address in the Transmit PDU Table as a function of the TPT Descriptor (TPD) from TVT and the offset inside the TPT entry is:

TPT Byte Address = 256 × TPTB + 16 × TPD + Offset.

TVTB

Address: BE_H (190)

Access: R/W

Register Bits:

The byte address in the Transmit VPCI Table as a function of the Transmit VPCI Table Descriptor TVD (current TVD in Credit Manager timer register TmTVD) and the offset inside the TVT entry is:

TVT Byte Address = 256 × TVTB + 16 × TVD + Offset.

TBB

Address: BF_H (191)

Access: R/W

Register Bits:

The byte address in the Transmit Buffer as a function of the Transmit Buffer Descriptor (TBD: from Transmit PDU Table TPT), BUFSIZE (size of receive/transmit buffers) and of the current payload offset TBD_OFFSET (from TPT) is:

$$\text{TB Byte Address} = 256 \times \text{TBB} + 2^{\text{BUFSIZE} + 6} \times \text{TBD} + 4 \times \text{TBD\_OFFSET}.$$

TWQB

Address: $C0_H$ (192)

Access: R/W

Register Bits:

The byte address in the Transmit Waiting Queue as a function of the read and write pointers in TWQWR/TWQPUT is:

$$\text{TWQ Byte Address} = 256 \times \text{TWQB} + 4 \times \text{RDPTR} \quad \text{(read queue)}$$

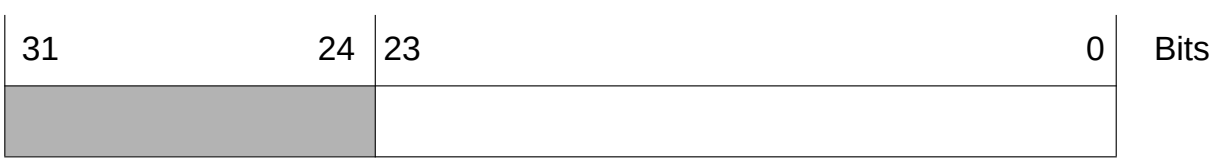
$$\text{TWQ Byte Address} = 256 \times \text{TWQB} + 4 \times \text{WRPTR} \quad \text{(write queue).}$$

TRQB

Address: $C1_H$ (193)

Access: R/W

Register Bits:



The byte address in the Transmit Ready Queue as a function of the read and write pointers in TRQGET/TRQRD is:

TRQ Byte Address = 256 × TRQB + 2 × RDPTR (read queue)

TRQ Byte Address = 256 × TRQB + 2 × WRPTR (write queue).

TFQB

Address: $C2_H$ (194)

Access: R/W

Register Bits:

The byte address in the Transmit Free Queue as a function of the read and write pointers

in TFQGET/TFQRD is:

TFQ Byte Address = 256 × TFQB + 2 × RDPTR (read queue)

TFQ Byte Address = 256 × TFQB + 2 × WRPTR (write queue).

6.2.13 Transmit Queue Management Registers

TWQL

Address: $C3_H$ (195)

Access: R/W

Register Bits:

The maximum length of the Transmit Waiting Queue is given by:

$NTWQ = 2^{TWQL+1} - 1$ 32-bit entries,

where the maximum value of TWQL is 15.

TWQPUT

Address: $C6_H$ (198)

Access: R

Register Bits:

31	16	15	0	Bits
RDPTR		WRPTR		

TWQWR

Address: $C9_H$ (201)

Access: R/W

Register Bits:

31	16	15	0	Bits
RDPTR		WRPTR		

A write to TWQL clears both the read and the write pointer.

To write a new entry in TWQ, the host first can check whether space is available in the queue, i.e. that

WRPTR + 1 $\overline{(\text{mod } (NTWQ+1))}$ RDPTR.

After having entered the command (TVD/TPT pair) into the TWQ (Byte Address = 256 $\times$ TWQB + 4 $\times$ WRPTR) the value of WRPTR should be incremented and written back to TWQWR. Several TWQ entries can be made if WRPTR is correctly updated. Bits 31 to 16 have no significance when writing TWQWR. Registers TWQWR and TWQPUT contain the same information. The only difference is that TWQWR is automatically incremented by 1 (modulo (N_{TWQ}+1)) when TWQPUT is read.

TRQL

Address: $C4_H$ (196)

Access: R/W

Register Bits:

31	4	3	0	Bits

The maximum length of the Transmit Ready Queue is given by:

NTRQ = $2^{TRQL+1} - 1$ 16-bit entries,

where the maximum value of TRQL is 15.

TRQGET

Address: $C7_H$ (199)

Access: R/W

Register Bits:

31 16	15 0	Bits
WRPTR	RDPTR	

TRQRD

Address: CA_H (202)

Access: R/W

Register Bits:

31 16	15 0	Bits
WRPTR	RDPTR	

A write to TRQL clears both the read and the write pointer.

The SARE writes TPT Descriptors of completely segmented PDUs into the Transmit Ready Queue (to byte address 256 × TRQB + 2 × WRPTR) and increments WRPTR. If $\overline{\text{RDPTR}}$ WRPTR, the host can obtain released TPT Descriptors by reading the TRQ at byte address 256 × TRQB + 2 × RDPTR. When TRQGET is read, the read pointer RDPTR is automatically incremented by “1” (modulo (N_{TRQ}+1)) as long as $\overline{\text{RDPTR}}$ WRPTR. If RDPTR is read from TRQRD, its value has to be incremented by the number of withdrawn TPDs and written back to TRQRD (no auto-incrementation of RDPTR). Bits 31 to 16 have no significance when writing TRQRD.

The write pointer WRPTR can be set to a specific value by writing TRQGET. However, this is not normally required.

TFQL

Address: $C5_H$ (197)

Access: R/W

Register Bits:

31	4	3	0	Bits

The maximum length of the Transmit Free Queue is given by:

NTFQ = $2^{TFQL+1} - 1$ 16-bit entries,

where TFQL can take a value up to 15.

TFQGET

Address: $C8_H$ (200)

Access: R/W

Register Bits:

31	16	15	0	Bits
WRPTR		RDPTR		

TFQRD

Address: CB_H (203)

Access: R/W

Register Bits:

31	16	15	0	Bits
WRPTR		RDPTR		

A write to TFQL clears both the read and the write pointer.

The SARE writes Transmit Buffer pointers (i.e. TBDs) of completely segmented Transm it Buffers into the Transmit Free Queue (to byte address 256 × TFQB + 2 × WRPTR) and increments WRPTR. If $\overline{\text{RDPTR}}$ WRPTR, the host can obtain released TBDs by reading the TFQ at byte address 256 × TFQB + 2 × RDPTR. When TFQGET is read, the read pointer RDPTR is automatically incremented by “1” (modulo (N_{TFQ}+1)) as long as $\overline{\text{RDPTR}}$ WRPTR. If RDPTR is read from TFQRD, its value has to be incremented by the

number of withdrawn TBDs and written back to TFQRD (no auto-incrementation of RDPTR). Bits 31 to 16 have no significance when writing TFQRD.

The write pointer WRPTR can be set to a specific value by writing TFQGET. However, this is normally not called for.

6.2.14 Cell Time Base Register

CELLCNT

Address: CC_H (204)

Access: R

Register Bits:

This counter register is incremented every cell interval (with UTOPIA transmit cell clock). It is used by the Credit Manager for the calculation of the current credit accumulated by a virtual connection, and also by the host for proper initialization of the CREDIT field in the Transmit VPCI Table entry.

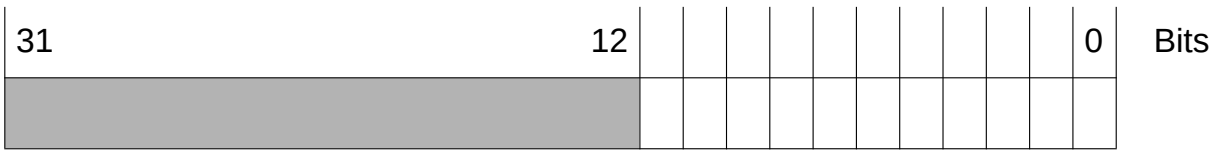
6.2.15 STRUCT Register

STRUCT

Address: CD_H (205)

Access: R/W

Register Bits:



The meaning of the bits is shown in **table 38**. The logical value of the bit determines which type of memory is used as follows:

bit = 0 This data type is in a local memory

bit = 1 This data type is in the host (PCI) memory.

Table 38 STRUCT Register Bit Descriptions

Bit	Name	Description
0	S_RB	Receive Buffer
1	S_TB	Transmit Buffer
2	S_RFQ	Receive Buffer Free Queue
3	S_RMQ	Receive MID Table Queue
4	S_RRQ	Receive Ready Queue
5	S_TWQ	Transmit Waiting Queue
6	S_TRQ	Transmit Ready Queue
7	S_TFQ	Transmit Buffer Free Queue
8	S_TPT	Transmit PDU Table
9	S_TVT	Transmit VPCI Table
10	S_RVT	Receive VPCI Table
11	S_RMT	Receive MID Table

Note: To optimize bus load sharing and minimize overhead, it is desirable to keep often used context information (for both segmentation and reassembly) either on-chip or in an external local memory. This is especially important in the case of high traffic loads, and when the PCI bus forms the application backbone interface of the system, the usage of which by the ATM communication process should be optimized.
The data types:
Receive VPCI Table
Receive MID Table
Receive Free Queue (RFQ)
Transmit VPCI Table
Transmit PDU Table
are needed at every cell reception or transmission. It is therefore recommended that these be kept on-chip, or, where the number of virtual connections exceeds 32, in local memory (STRUCT bit = 0).

7 Electrical Characteristics

7.1 Absolute Maximum Ratings

Parameter	Symbol	Limit Values	Unit
Ambient temperature under bias	T_A	– 40 to + 85	°C
Storage temperature	T_{stg}	– 65 to + 125	°C
Supply voltage, 3 V	V_{DD3}	– 0.5 to + 4.125	V
Supply voltage, 5 V	V_{DD5}	– 0.5 to + 6.0	V
Voltage on PCI and UTOPIA interface pins (with respect to ground)	V_S	If V_{DD5} < 3 V: – 0.4 to V_{DD3} + 0.5 If V_{DD5} > 3 V: – 0.4 to V_{DD5} + 0.5	 V V
Voltage on any other pin (with respect to ground)	V_S	– 0.4 to V_{DD3} + 0.5	V

Note: Stresses above those listed here may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

7.2 Recommended Operating Conditions

Parameter	Symbol	Limit Values	Unit
Supply voltage, 3 V	V_{DD3}	3.0 to 3.6	V
Supply voltage, 5 V	V_{DD5}	4.5 to 5.5	V
Voltage	V_{SS}	0	V

7.3 DC Characteristics

(T_A = – 40 to + 85 °C, see also **section 7.2**)

Parameter	Symbol	Limit Values			Unit	Test Condition
		min.	typ.	max.		
High-Level input voltage	V_{IH}	2.0			V	
Low-Level input voltage	V_{IL}			0.8	V	
High-Level output voltage	V_{OH}	2.4			V	I_{OH} = – 2 mA for PCI interface pins; I_{OH} = – 400 µA for all other pins
Low-Level output voltage	V_{OL}			0.55	V	I_{OL} = 2 mA-6 mA based on the pin.
Power consumption on V_{DD3}			150	300	mA	*Note: The power consumption is heavily dependent on the traffic load, operating parameters, and other factors.*
Power consumption on V_{DD5}			1	3	mA	
Input leakage current, PCI pins	I_{LIp}	– 70		+ 70	µA	0 V < V_{IN} < V_{DD3}
Input leakage current, all other pins	I_{LI}	– 1		1	µA	0 V < V_{IN} < V_{DD3}
Output leakage current, PCI pins	I_{LOp}	– 70		+ 70	µA	0 V < V_{OUT} < V_{DD3}
Output leakage current, all other pins	I_{LO}	– 1		1	µA	0 V < V_{OUT} < V_{DD3}

Note: When applying power to the SARE, the following rule should be observed: The voltage on V_{DD5} should never exceed V_{DD3} by more than 3.6 V at anytime. Applying voltages to signal pins when power supply is not active (circuit not under bias) may cause damage – refer to paragraph "Absolute Maximum Ratings". When power supply is switched on, the pads do not reach their stable bias until after 2 µs (maximum).

7.4 Capacitance

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input capacitance	C_{IN}		12 8 10	pF pF pF	CLK IDSEL pin All other inputs
I/O capacitance	$C_{I/O}$		10	pF	
Load capacitance (Os and I/Os)	C_L		56 66 37 27 20 25 50	pF pF pF pF pF pF pF	$\overline{\text{LCS}}$ LA(19:0), $\overline{\text{LRE}}$ LD(31:0), $\overline{\text{LWE}}$(1:0) LPAR(3:0), $\overline{\text{PCS}}$, TxCLKO, $\overline{\text{RxENB}}$, TxD(7:0), TxPTY, TxSOC, $\overline{\text{TxENB}}$ All others

7.5 AC Characteristics

7.5.1 AC Measurement Conditions

Measurement conditions for the power supply voltage are as specified above under Recommended Operating Conditions, at T_A = – 40 to + 85 °C.

7.5.2 UTOPIA Interface Timing

The AC testing input/output waveforms are shown in **figure 18**.

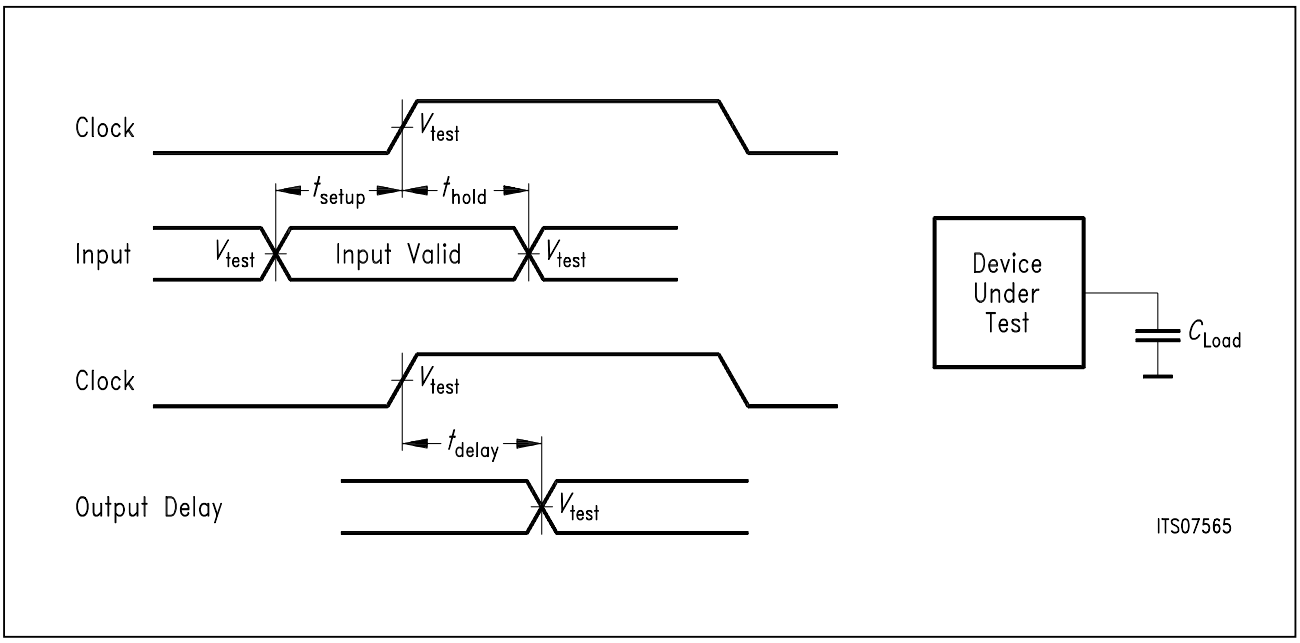


Figure 18 UTOPIA Interface Timing Measurement Waveforms

Table 39 UTOPIA Interface Timing Measurement Conditions

Parameter	Value	Unit
Input capacitance of input signal (pin)	10	pF
Input and output timing reference Level (V_{test})	1.4	V

The UTOPIA Interface timing of the SARE is compliant with the ATM Forum Universal Test & Operations PHY Interface for ATM (Level 1) specification. To simplify operation in a system that runs with a PCI Interface at 33 MHz, the UTOPIA Interface is specified to run at up to 33 MHz (instead of up to 25 MHz only).

Note: In the case that the clock frequency f used on the UTOPIA interface is higher than 25 MHz (for a PCI CLK clock frequency of 33 MHz), caution should be exercised. when long bursts of cells are received: in some cases bursts of cells received at a rate of 8 × f bits/s may lead to a loss of cell at full load.

As explained earlier, the UTOPIA Interface of the SARE has two modes of operation, depending on whether the SARE operates as a clock master or clock slave: “high”

– If pin M/S = “high” (Master mode), the SARE is normally connected to a PHY device (applications in ATM Terminal equipment).
– If pin M/S = “low” (Slave mode), the SARE may be connected to a switch (e.g. via an ATM Switching Pre/Post-Processor ASP), for instance for implementing signalling, OAM or testing in an ATM network.

The pin names of the SARE UTOPIA Interface are according to the Master mode. They change their meaning in the Slave mode according to **figure 19**.

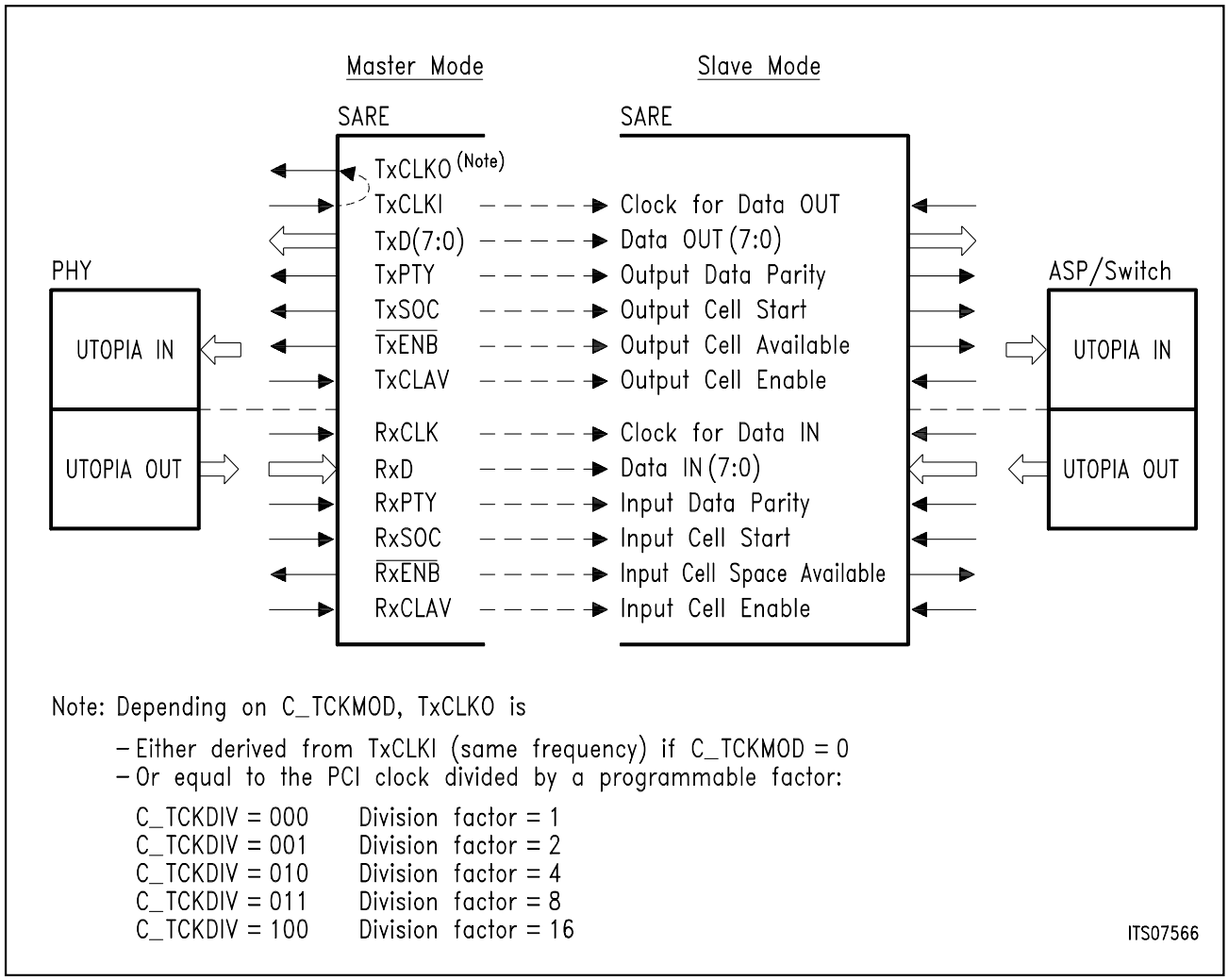


Figure 19 Significance of SARE Pins in Master and Slave Modes

The pin connections in the two cases are thus shown in **figure 20**.

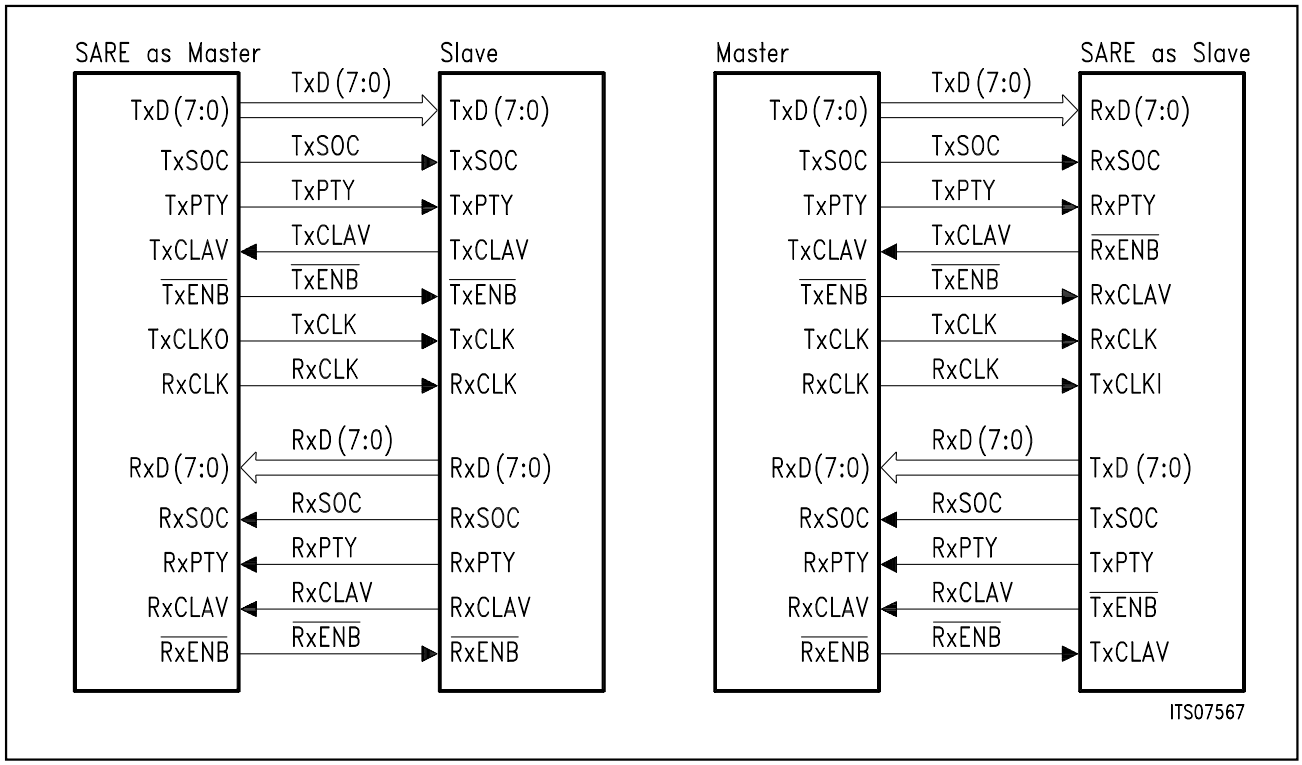


Figure 20 Connections in Master and Slave Modes

In the following description, the first name on the timing diagram is the name of the SARE pin in the Master mode: this is also the name of the signal according to the UTOPIA Interface specification nomenclature. The second name, in parentheses, is the corresponding SARE pin when it operates in the Slave mode.

For details on the UTOPIA Interface protocol, see “UTOPIA Level 2 Specification, Version 0.8”.

7.5.2.1 Transmit Handshake Protocol Timing

The timing sequences may be summarized as: $\overline{\text{TxENB}}$ asserted indicates valid ATM available in the current TxCLK cycle; TxCLAV deasserted indicates the entity that receives TxD is unable to accept another cell transfer after the current one.

Figures 21 to **25** show the cell-level handshaking protocol implemented by the SARE.

In the first example (**figure 21**), the SARE recognizes on clock edge 2 that TxCLAV has been asserted by the PHY (assuming that SARE is in Master mode), and starts to transmit a complete cell. The PHY indicates 4 cycles before the end of the cell (i.e. on clock edge 51) whether it can accept an additional cell. In this example, the PHY cannot accept an additional cell and, as a consequence, the SARE deasserts $\overline{\text{TxENB}}$ on clock edge 55. Once TxCLAV indicates that the PHY can accept a cell, it has to stay asserted until the PHY recognizes that the SARE has transmitted payload byte 43 of that cell. As the PHY must indicate that it cannot accept another cell 4 write cycles before the end of the cell, the high-to-low transition of TxCLAV has to take place exactly on clock edge 50.

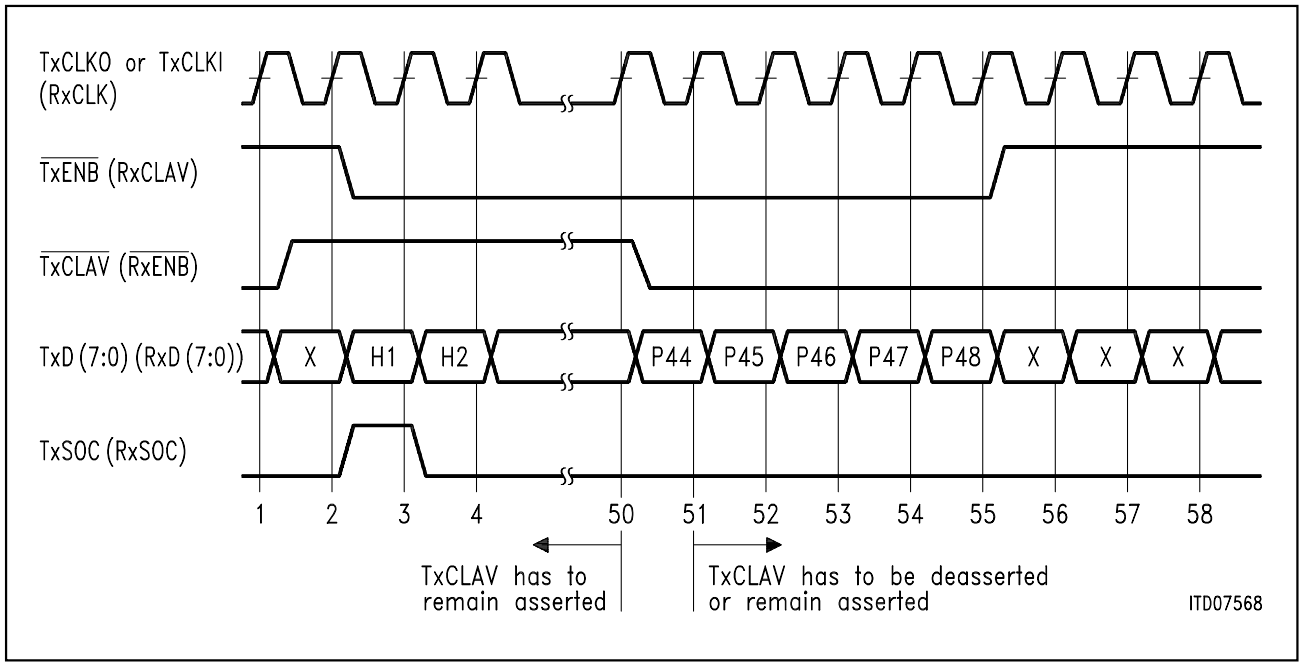


Figure 21 UTOPIA I/F Transmit Handshake Protocol (Master Mode), Example 1

Figure 22 shows an example where the PHY indicates on clock edge 3 that it can accept another cell from the SARE. If a complete cell is available, the SARE starts transmitting the next cell immediately after P48.

Note: According to the UTOPIA protocol, the device that outputs $\overline{TxENB}$ can interrupt data transmission at any time by deasserting $\overline{TxENB}$, as shown in this example after H4. However, this will never occur when the SARE outputs $\overline{TxENB}$ (i.e. the SARE is in Master mode), since the SARE does not start transmitting until a complete cell is ready for transmission. When the SARE is in the Slave mode, it behaves itself according to the (i.e. it ignores the byte following H4).

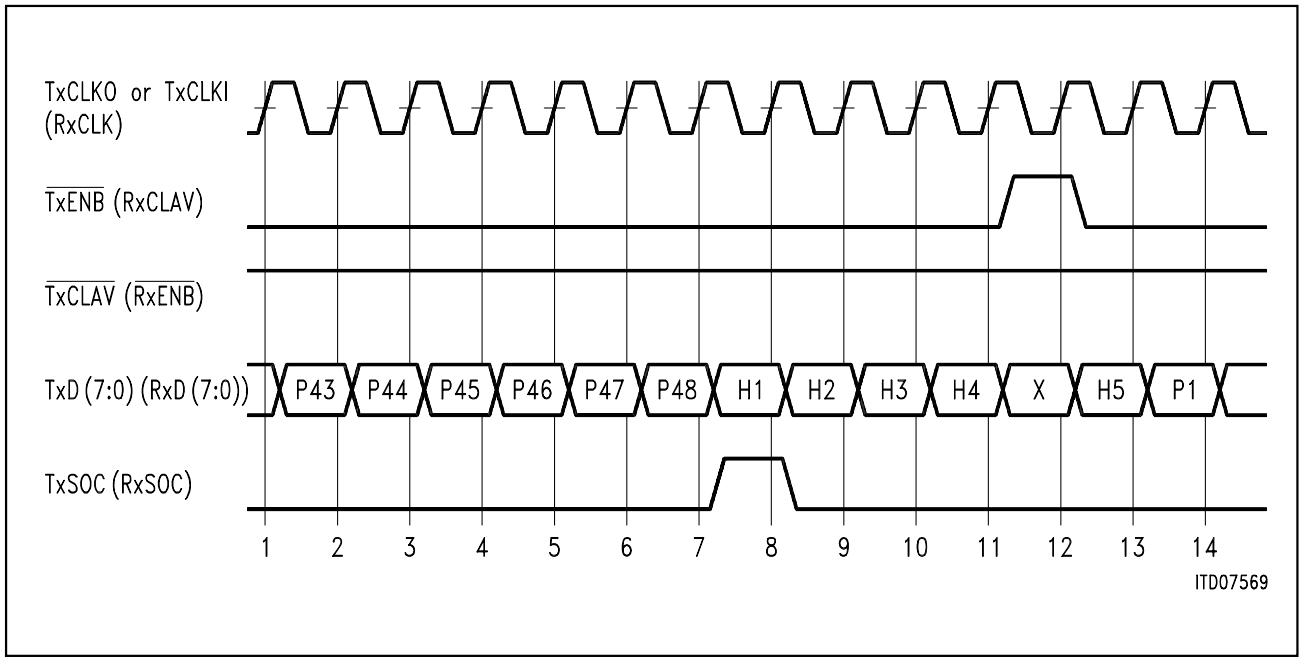


Figure 22 UTOPIA I/F Transmit Handshake Protocol (Master Mode), Example 2

Figure 23 shows an example where the PHY indicates on clock edge 3 that it cannot accept another cell from the SARE. Before the SARE can interrupt data transmission by deasserting TxENB on clock edge 7, it detects on clock edge 5 that the PHY can again accept another cell. Therefore, the SARE keeps TxENB asserted and transmits the next cell immediately after P48 (if it has a cell to transmit).

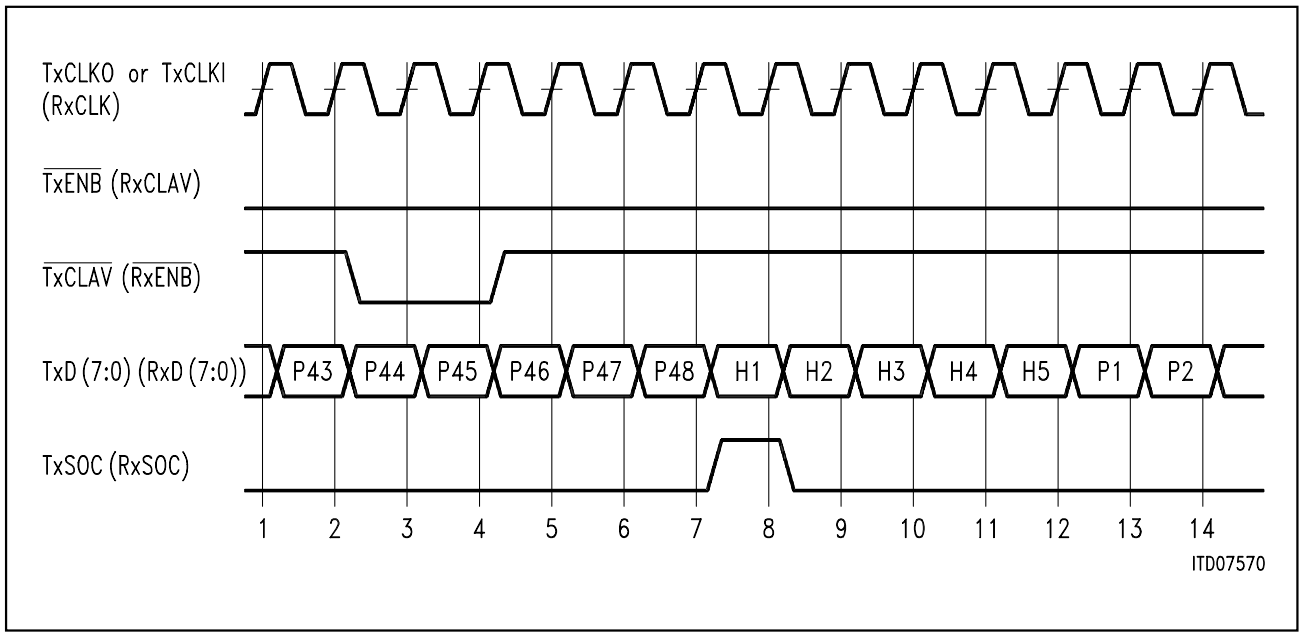


Figure 23 UTOPIA I/F Transmit Handshake Protocol (Master Mode), Example 3

7.5.2.2 Receive Handshake Protocol Timing

The timing sequences may be summarized as: $\overline{\text{RxENB}}$ forces a data read from the entity that outputs RxD when RxCLAV is asserted. Data is output during a cycle after one at which $\overline{\text{RxENB}}$ was asserted and RxCLAV was asserted.

In the first example below (**figure 24**), the Slave asserts RxCLAV to indicate that a new cell is available. When the master is ready to accept the cell, it asserts $\overline{\text{RxENB}}$ (on clock edge 3). Data transfer starts on the next clock edge. RxCLAV is held active by the slave for the duration of the cell. If another cell is immediately available thereafter, RxCLAV stays active. RxSOC is activated for the cycle where the first byte of every cell is transferred. When the transfer of the second cell is completed, the slave drives RxCLAV low to indicate invalid data on RxD (on clock edge 110). The cell transfer must have been completed because the slave should not output invalid data during cell transfer. If it can accept a new cell (determined by the filling level of its internal buffers), the master does not react on the indication, but simply keeps $\overline{\text{RxENB}}$ asserted. On clock edge 163 the master deasserts $\overline{\text{RxENB}}$ while P48 is on RxD because it knows that P48 is the last octet of the cell. After the transfer of the last byte, the slave deasserts RxCLAV to indicate that no cell is available. Simultaneously, RxD and RxSOC return to high-impedance.

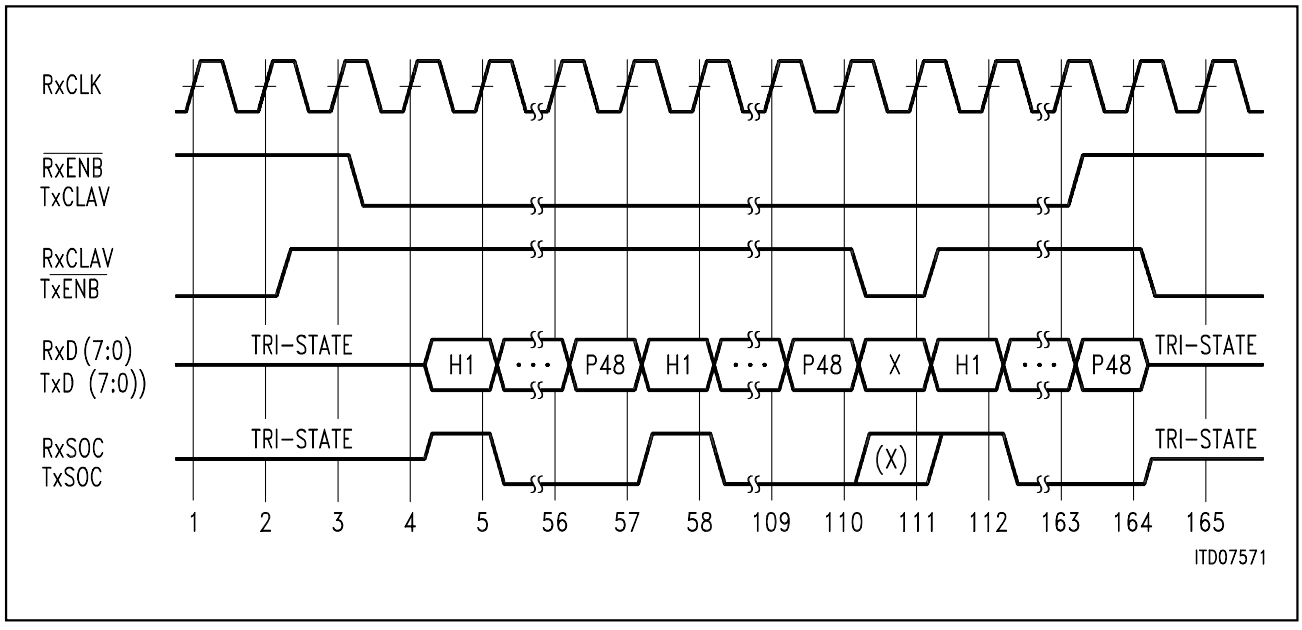


Figure 24 UTOPIA Interface Receive Handshake Protocol, Example 1

In the second example (**figure 25**), the Slave indicates the availability of a new cell on clock edge 1. When the Master is ready to accept the cell it asserts $\overline{\text{RxENB}}$ (on clock edge 10 in the example). Note that, when RxCLAV has been deasserted and $\overline{\text{RxENB}}$ is deasserted, $\overline{\text{RxENB}}$ is not reasserted before RxCLAV is asserted (and the master is ready to accept a new cell). At the end of the first cell the master deasserts $\overline{\text{RxENB}}$ on clock edge 63. RxCLAV stays active to indicate the availability of a new cell. If the master deasserts $\overline{\text{RxENB}}$ during the transfer of the next cell (on clock edge 101), the slave returns RxD and RxSOC to high-impedance in the next cycle.

(Note: This case will not occur when the SARE is in the master mode, since the SARE will not activate $\overline{\text{RxENB}}$ unless it has space for a complete cell.)

Cell transfer resumes when RxENB is asserted again.
The meaning of RxCLAV depends on $\overline{\text{RxENB}}$: On clock edge 1 ($\overline{\text{RxENB}}$ is inactive) and on clock edge 64 ($\overline{\text{RxENB}}$ has been deasserted in the previous cycle) RxCLAV indicates the availability of a new cell. From clock edge 11 on ($\overline{\text{RxENB}}$ has been asserted in the previous cycle), RxCLAV indicates valid data on RxD.

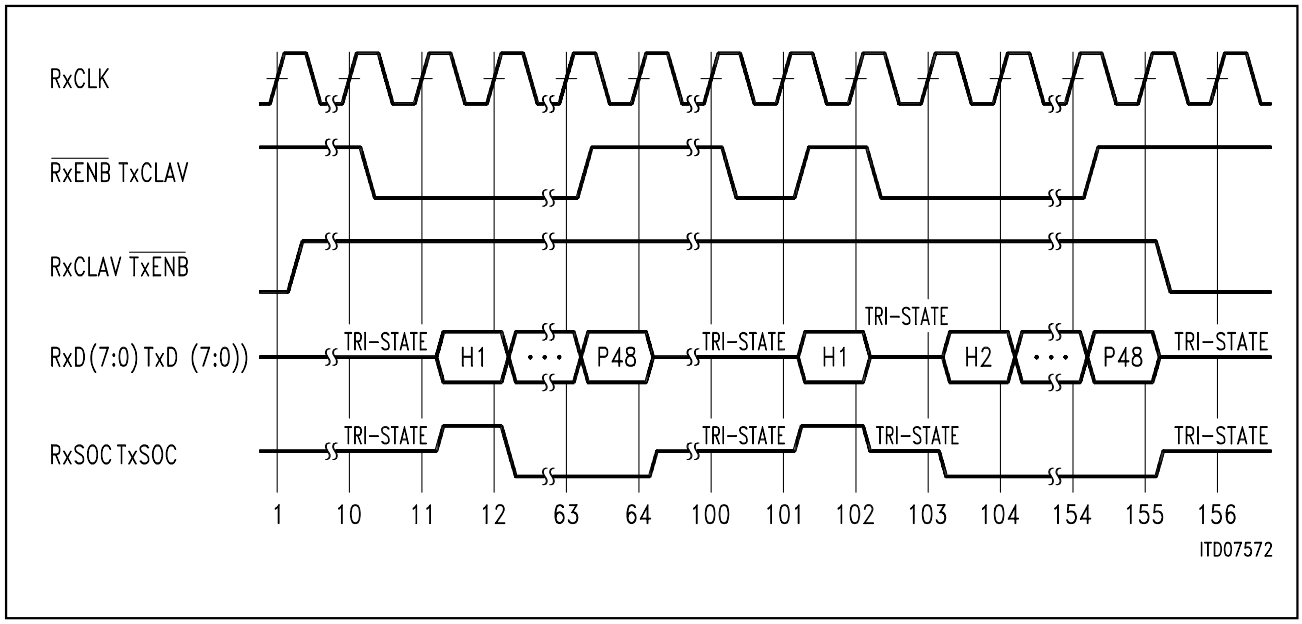


Figure 25 UTOPIA Interface Receive Handshake Protocol, Example 2

The timing characteristics for the UTOPIA Interface signals are given in the table below.

7.6 UTOPIA Interface Signal Characteristics

Parameter	Limit Values			Unit	Remarks
	min.	typ.	max.		
TxCLKI and RxCLKI frequency (nominal)	0		33	MHz	
TxCLKI and RxCLKI duty cycle	40%		60%		
TxCLKI and RxCLKI peak-to-peak jitter (Note 1)	CL		5%		
TxCLKI and RxCLKI rise/fall time (Note 2)			3	ns	
TxCLKO duty cycle					Master mode only
TxCLKO peak-to-peak jitter (Note 1)					Master mode only
TxCLKO rise/fall time (Note 2)				ns	Master mode only
Setup time of TxCLAV to TxCLKO	10			ns	Master mode only
Hold time of TxCLAV from TxCLKO	1			ns	Master mode only
Setup time of TxCLAV to TxCLKI	10			ns	Slave mode
Hold time of TxCLAV from TxCLKI	1			ns	Master mode
Delay time of TxD(7:0), TxPTY, TxSOC, $\overline{\text{TxENB}}$ from TxCLKO		10		ns	Master mode
Delay time of TxD(7:0), TxPTY, TxSOC, $\overline{\text{TxENB}}$ from TxCLKI		17		ns	Slave mode
Setup time of RxD(7:0), RxPTY, RxSOC, RxCLAV to RxCLK	10			ns	
Hold time of RxD(7:0), RxPTY, RxSOC, RxCLAV from RxCLK	1			ns	
Delay time of $\overline{\text{RxENB}}$ from RxCLK		15		ns	

Note 1 Measured from one rising edge to the next rising edge.

Note 2 Measured at transmit side (driver), unloaded, between 10% and 90% / 90% and 10% levels of V_{OH}.

7.6.1 PCI Interface Timing

The AC testing input/output waveforms are shown in **figures 26** and **27** below.

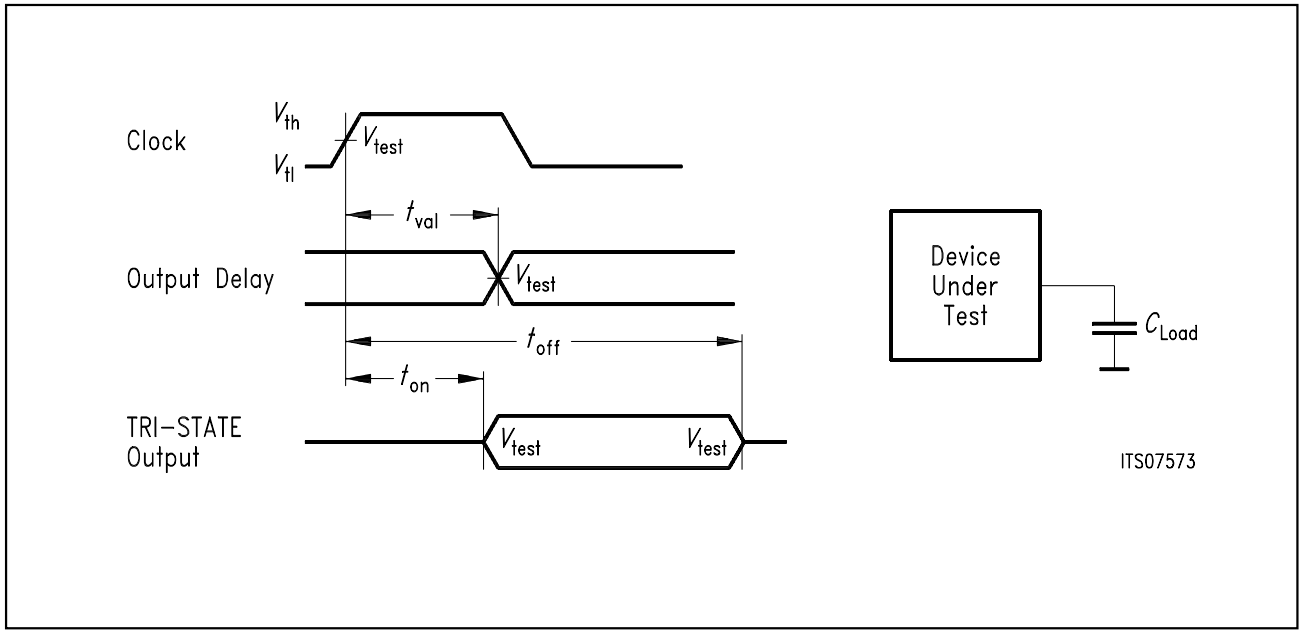


Figure 26 PCI Output Timing Measurement Waveforms

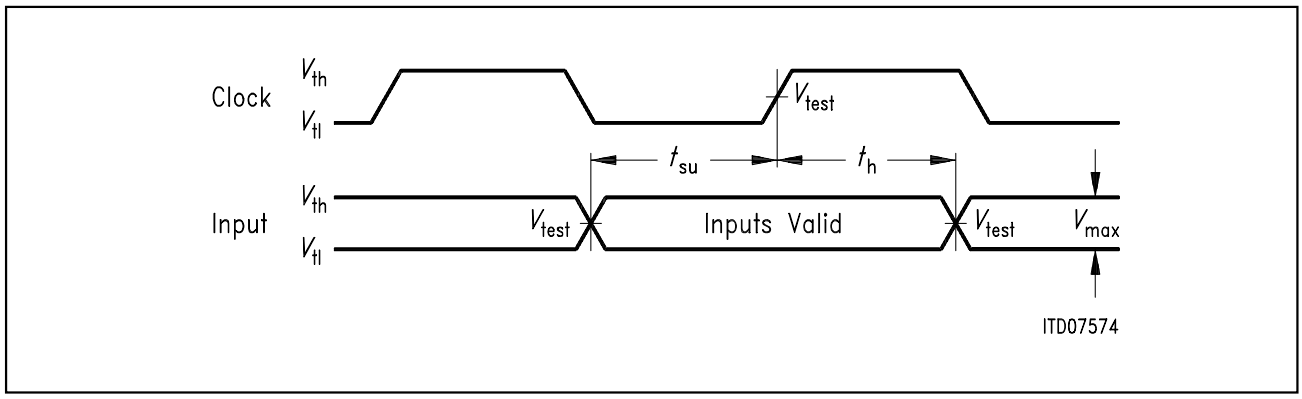


Figure 27 PCI Input Timing Measurement Waveforms

Table 40 PCI Input and Output Measurement Conditions

Symbol	Value	Unit
V_{th}	2.4	V
V_{tl}	0.4	V
V_{test}	1.5	V
V_{max}	2.0	V

The timings below show the basic read and write transaction between an initiator (Master) and a target (Slave) device.

The SARE is able to work both as master and slave. The former mode is used by the SARE to write and read data to/from the host memory using the integrated DMA controller and the burst capability of the PCI interface. The latter mode enables for an external entity (a host) to read and write the SARE registers and memories, and to access the local memory and/or a peripheral device connected to the SARE, via the SARE.

7.6.1.1 PCI Read Transaction

The transaction starts with an address phase which occurs during the first cycle when $\overline{\text{FRAME}}$ is activated (clock 2 in **figure 28**). During this phase the bus master (initiator) outputs a valid address on AD(31:0) and a valid bus command on $\overline{\text{C/BE}}$(3:0). The first clock of the first data phase is clock 3. During the data phase $\overline{\text{C/BE}}$ indicate which byte lanes on AD(31:0) are involved in the current data phase.

The first data phase on a read transaction requires a turn-around cycle. In **figure 28** the address is valid on clock 2 and then the master stops driving AD. The target drives the AD lines following the turnaround when $\overline{\text{DEVSEL}}$ is asserted. ($\overline{\text{TRDY}}$ cannot be driven until $\overline{\text{DEVSEL}}$ is asserted.) The earliest the target can provide valid data is clock 4. Once enabled, the AD output buffers of the target stay enabled through the end of the transaction.

A data phase may consist of a data transfer and wait cycles. A data phase completes when data is transferred, which occurs when both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. When either is deasserted a wait cycle is inserted. In the example below, data is successfully transferred on clocks 4, 6 and 8, and wait cycles are inserted on clocks 3, 5 and 7. The first data phase completes in the minimum time for a read transaction. The second data phase is extended on clock 5 because $\overline{\text{TRDY}}$ is deasserted. The last data phase is extended because $\overline{\text{IRDY}}$ is deasserted on clock 7.

The Master knows at clock 7 that the next data phase is the last. However, the master is not ready to complete the last transfer, so $\overline{\text{IRDY}}$ is deasserted on clock 7, and $\overline{\text{FRAME}}$ stays asserted. Only when $\overline{\text{IRDY}}$ is asserted can $\overline{\text{FRAME}}$ be deasserted, which occurs on clock 8.

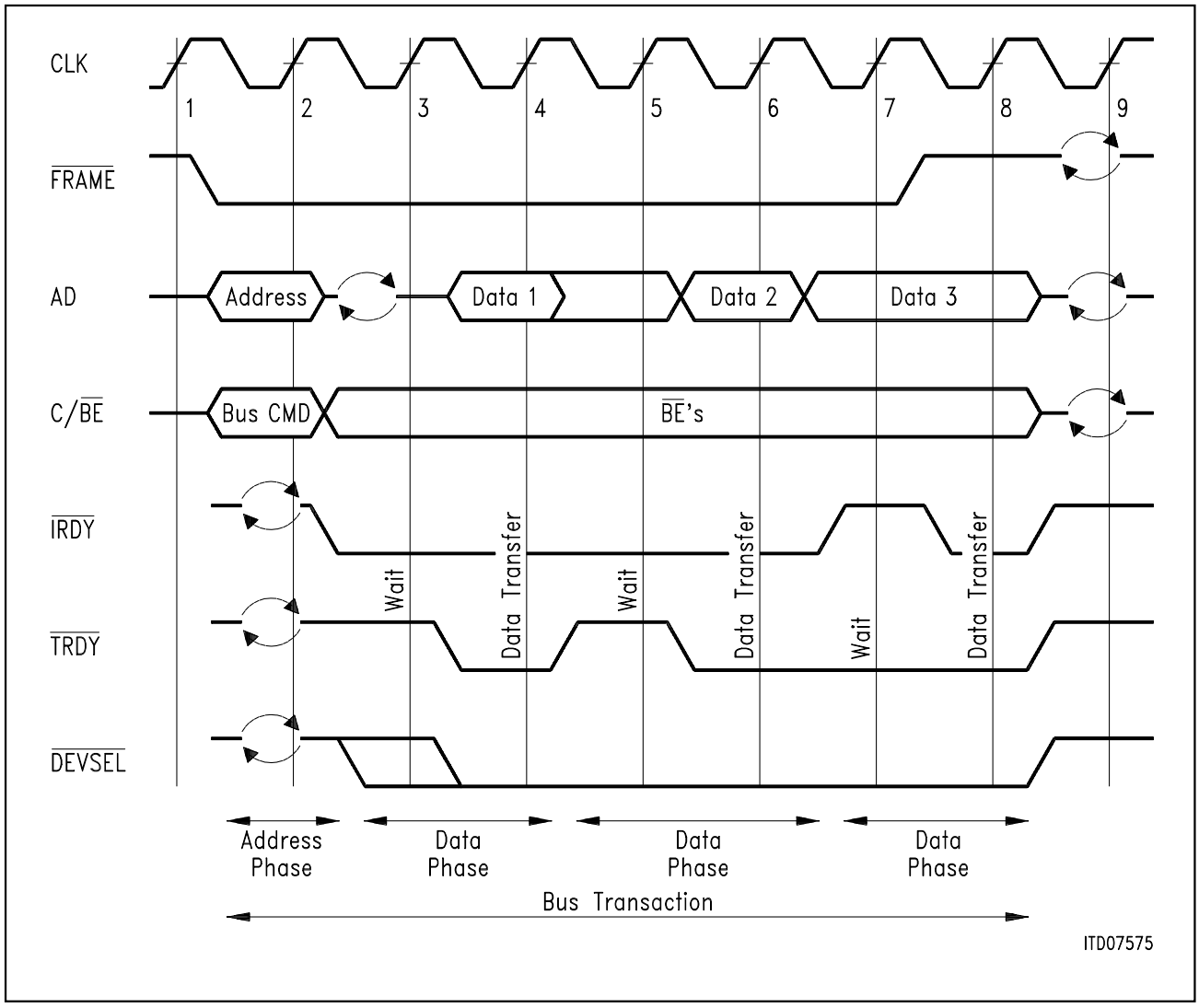


Figure 28 PCI Read Transaction

7.6.1.2 PCI Write Transaction

The transaction starts when $\overline{\text{FRAME}}$ is activated (clock 2 in **figure 29**). A write transaction is similar to a read transaction except no turnaround cycle is required following the address phase. In the example, the first and second data phases complete with zero wait cycles. The third data phase has three wait cycles inserted by the target. Both initiator and target insert a wait cycle on clock 5. In the case where the initiator inserts a wait cycle (clock 5), the data is held on the bus, but the byte enables are withdrawn. The last data phase is characterized by $\overline{\text{IRDY}}$ being asserted while the $\overline{\text{FRAME}}$ signal is deasserted. This data phase is completed when $\overline{\text{TRDY}}$ goes active (clock 8).

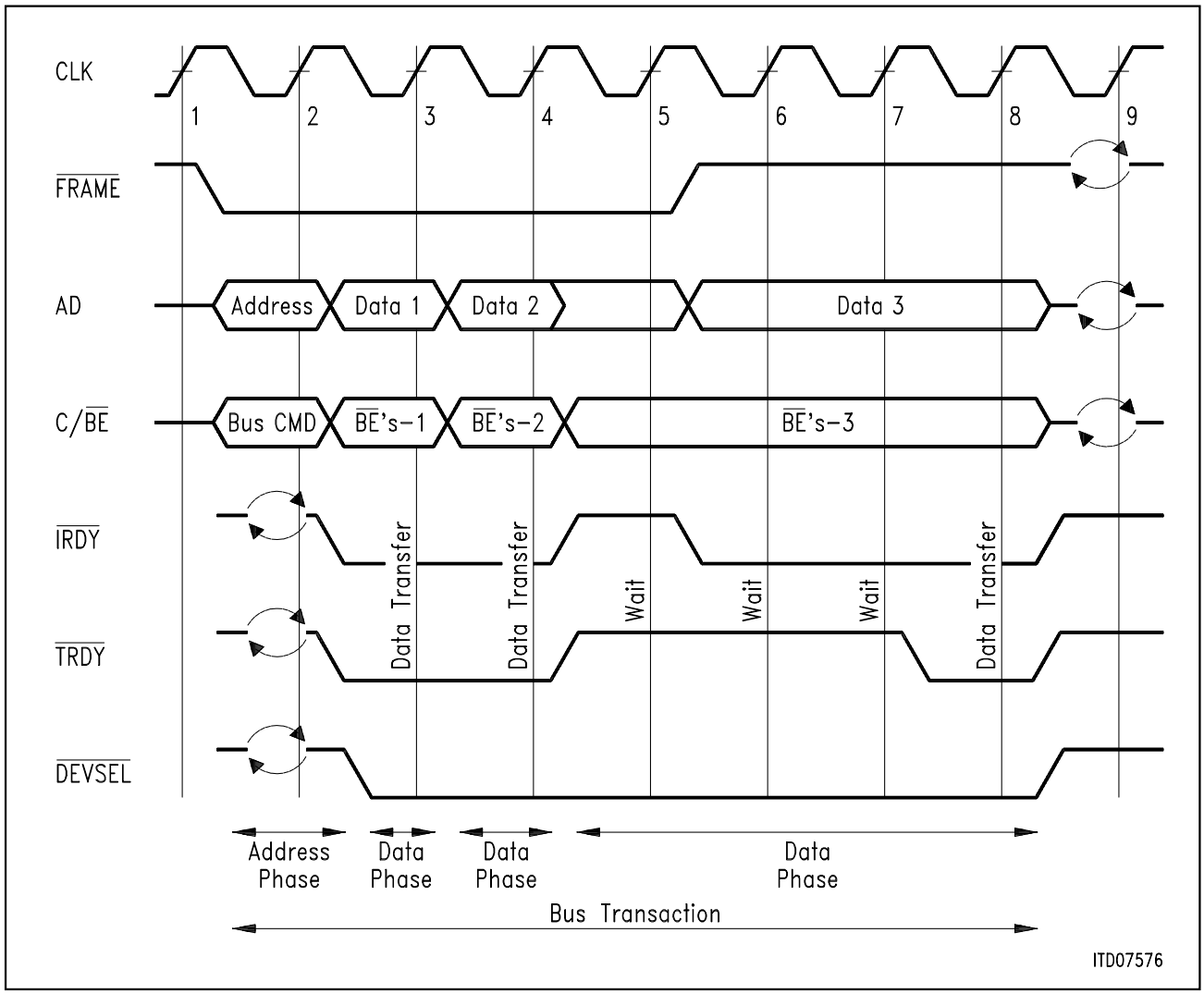


Figure 29 PCI Write Transaction

When the SARE operates as a PCI Master (initiator) and it either reads or writes a burst – as controlled by the on-chip DMA controller – it does not deactivate $\overline{\text{IRDY}}$ between consecutive data. In other words, no wait states are inserted by the SARE as a transaction initiator. When the SARE operates as a PCI Slave (target), it inserts wait cycles by deactivating $\overline{\text{TRDY}}$ for 4 (read) or 2 (write) cycles typically. The number of wait states inserted is not critical for two reasons. One, because accesses to/via the SARE are usually kept to a minimum in a system. And two, because they are dependent on the type of access (e.g. for an access to a peripheral device on the PREADY signal).

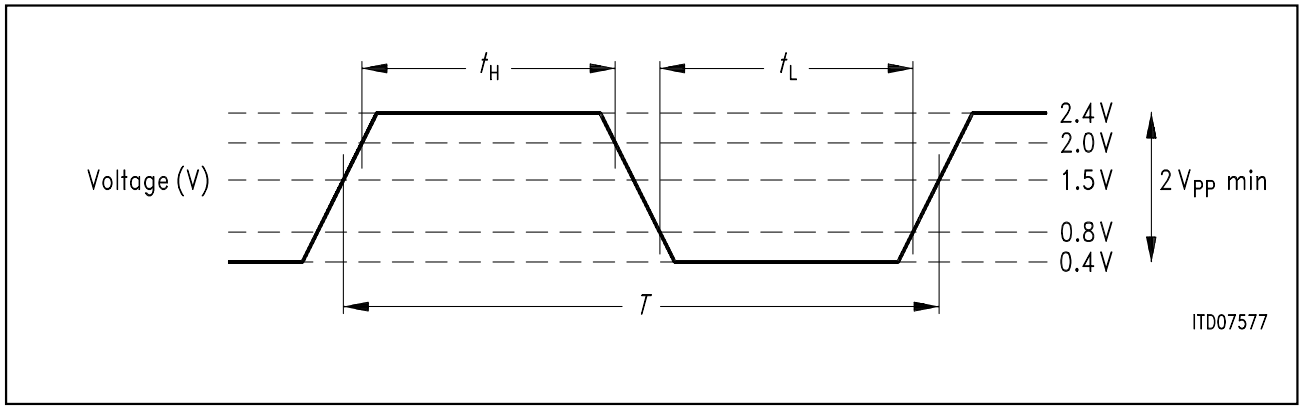


Figure 30 PCI Clock Specification

Table 41 PCI Clock Characteristics

Parameter	Symbol	Limit Values			Unit
		min.	typ.	max.	
CLK cycle time	T	30		∞	ns
CLK high time	t_H	12			ns
CLK low time	t_L	12			ns
CLK slew rate (see note)		1		4	V/ns

Note: Rise and fall times are specified in terms of the edge rate measured in V/ns. This slew rate must be met across the minimum peak-to-peak portion of the clock waveform as shown in ***figure 30****.*

Table 42 PCI Interface Signal Characteristics

Parameter	Limit Values			Unit	Remarks
	min.	typ.	max.		
CLK to signal valid delay bussed signals		11		ns	Notes 1, 2
CLK to signal valid delay point-to-point		12		ns	Notes 1, 2
Float to active delay		3		ns	
Active to float delay		20		ns	
Input setup time to CLK bussed signals		7		ns	Note 2
Input setup time to CLK point-to-point		8		ns	Note 2
Input hold time from CLK		0		ns	

Note 1 Minimum times are measured with 0 pF equivalent load; maximum times are measured with 50 pF equivalent load.

Note 2 $\overline{REQ}$ and $\overline{GNT}$ are point-to-point signals. All other signals are bussed.

7.6.2 Local Memory/Control Interface Timing

The frequency of this interface is given by the PCI clock CLK, in that the cycle time on the Local Memory/Control Interface is equal to twice the CLK cycle (i.e. 60 ns for a 33 MHz PCI clock). **Figure 31** illustrates the timing for reading or writing an external local memory (when the PREADY signal *is not* used).

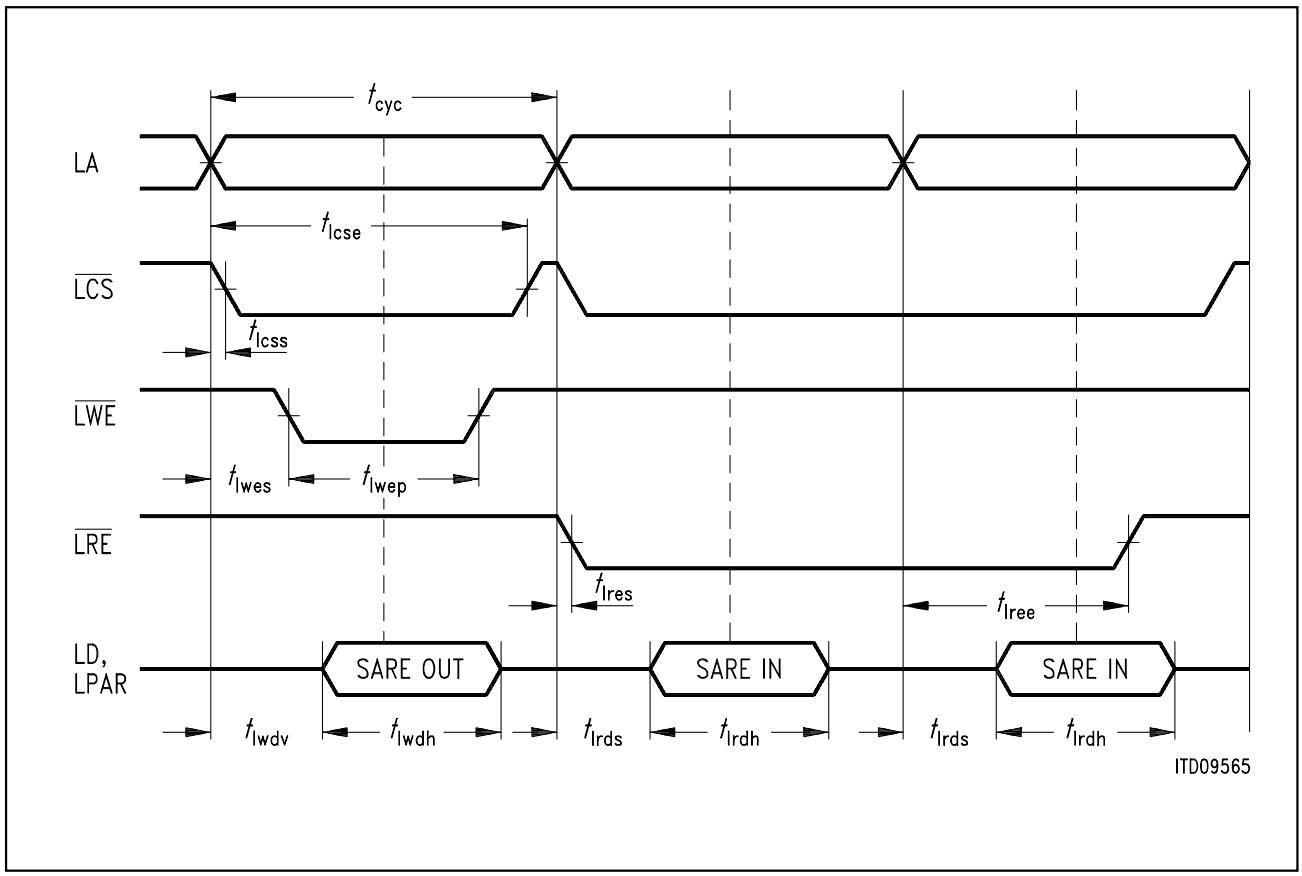


Figure 31 Local Memory/Control Interface Timing

In **table 43**, the PCI Interface cycle period is denoted by T.

Table 43 Local Memory/Control Interface Timing Characteristics

Parameter	Symbol	Limit Values			Unit
		min.	typ.	max.	
Access cycle time	t_{cyc}		$2T$		ns
Chip select start time from LA	t_{lcss}		0		ns
Chip select end time from LA	t_{lcse}		55		ns
Write enable start time from LA	t_{lwes}	12	15	18	ns
Write enable pulse time	t_{lwep}		T		ns
Write data valid from LA	t_{lwdv}	15	20	25	ns
Write data hold	t_{lwdh}		25		ns
Read enable start time from LA	t_{lres}		0		ns
Read enable end time from last Read cycle start	t_{lree}	40	43	46	ns
Read data start from LA	t_{lrds}			26	ns
Read data hold after start	t_{lrdh}	8			ns

7.6.3 Peripheral Device Timing Characteristics

In the case of a peripheral device access where PREADY is inactive, the read or write cycle is extended: the address and $\overline{PCS}$ are held on the bus, and $\overline{LWE0}$ or $\overline{LRE}$ remains low. The cycle is terminated ($\overline{LRE}$ or $\overline{LWE}$ returned to “high”) between 1T and 2T from the instant when PREADY becomes active. **Figure 32** illustrates the timing for reading or writing a peripheral device (when the PREADY signal *is* used).

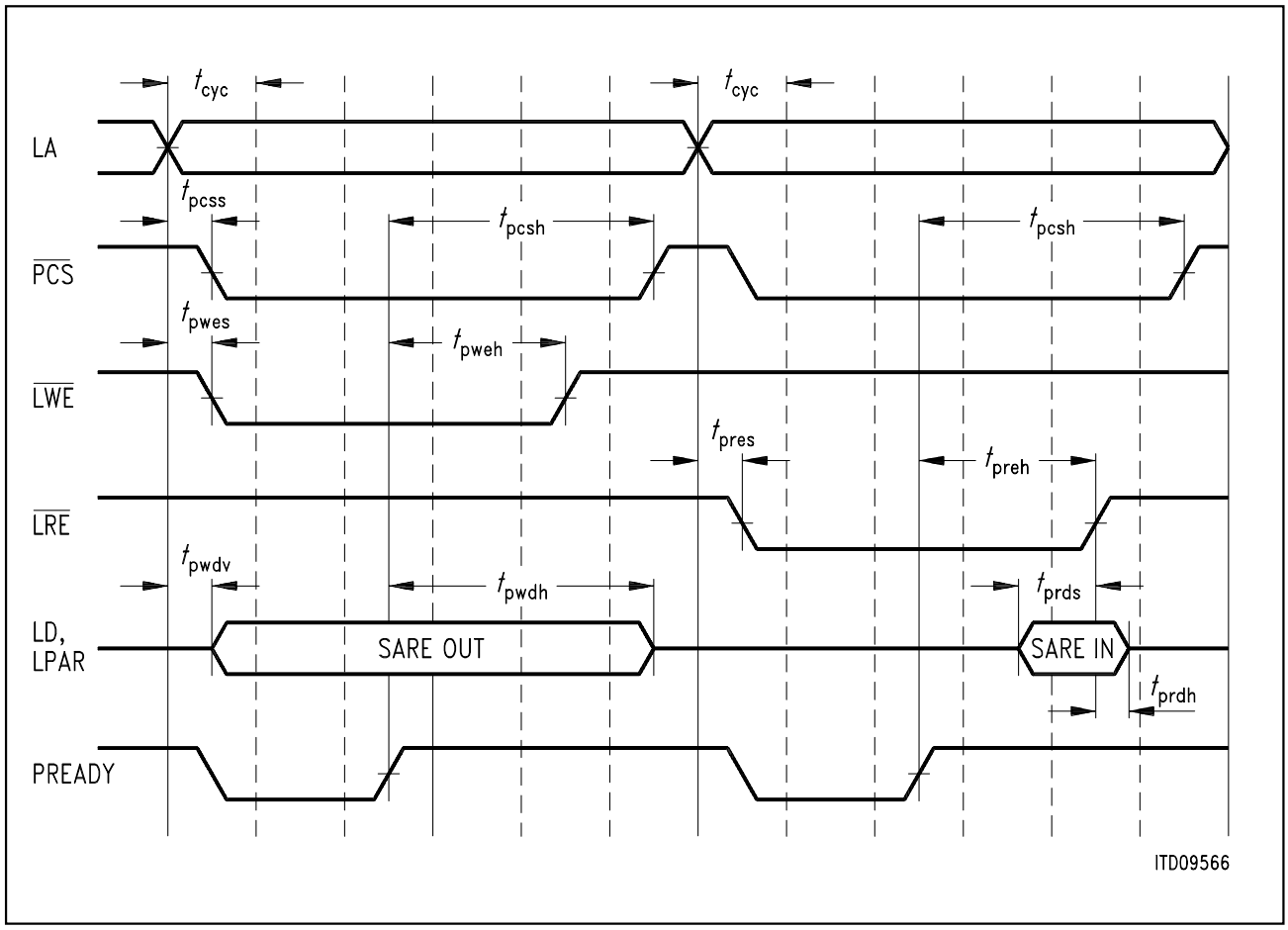


Figure 32 Peripheral Device Access Timing

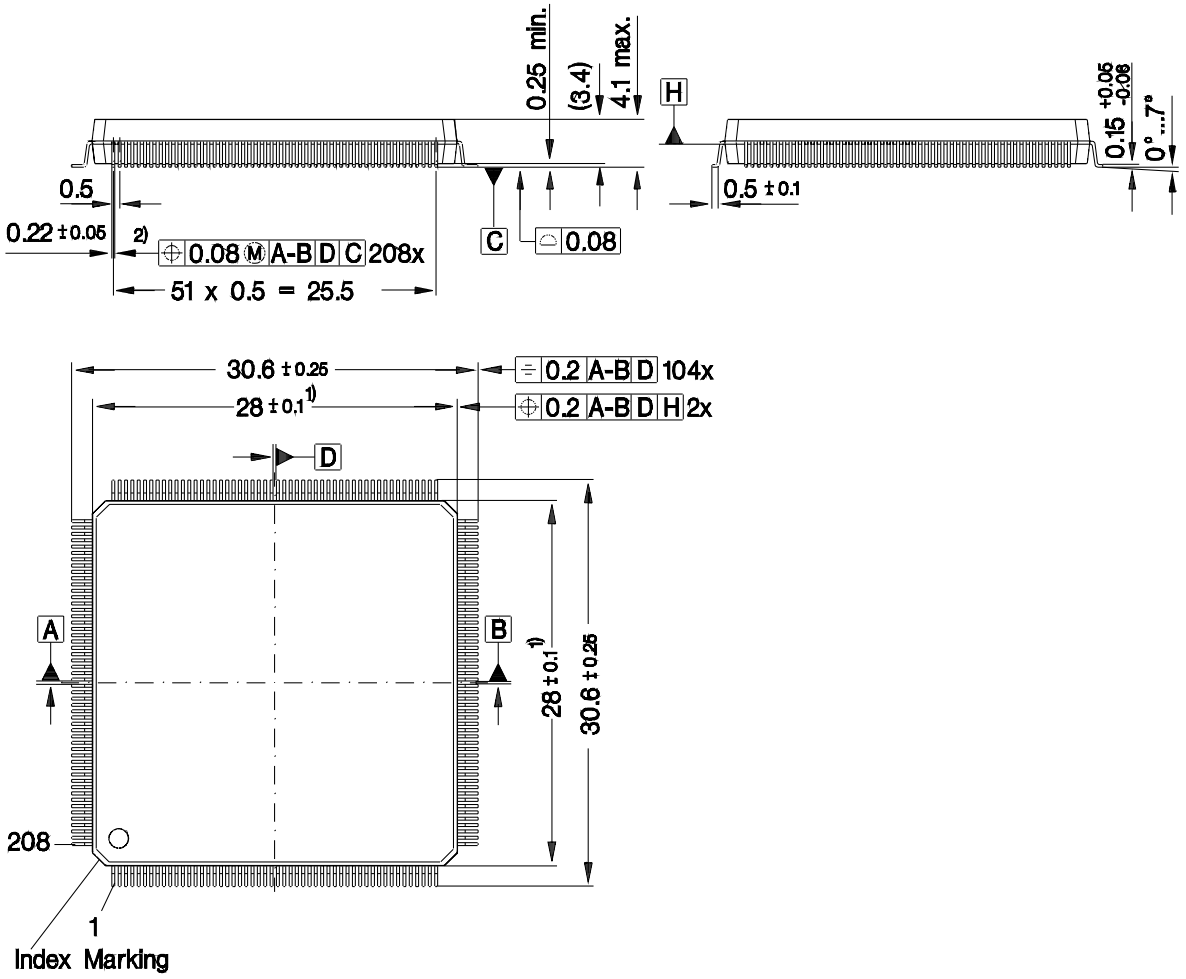
In **table 44**, the PCI Interface cycle period is denoted by T.

Table 44 Peripheral Device Access Timing Characteristics

Parameter	Symbol	Limit Values			Unit
		min.	typ.	max.	
Access cycle time	t_{cyc}		$1T$		ns
Chip select start time from LA	t_{pcss}	12	15	18	ns
Chip select hold time from PREADY	t_{pcsh}	$2.5T$	$3T$	$3.5T$	ns
Write enable start time from LA	t_{pwes}	11	14	17	ns
Write enable hold time from PREADY	t_{pweh}	$1.5T$	$2T$	$2.5T$	ns
Write data valid from LA	t_{pwdv}	10	15	20	ns
Write data hold time from PREADY	t_{pwdh}	$2.5T$	$3T$	$3.5T$	ns
Read enable start time from LA	t_{pres}	11	14	17	ns
Read enable hold time from PREADY	t_{preh}	$1.5T$	$2T$	$2.5T$	ns
Read data setup time from $\overline{\text{LRE}}$	t_{prds}	16			ns
Read data hold time from $\overline{\text{LRE}}$	t_{prdh}	0			ns

8 Package Outline

Plastic Package, P-FQFP-208-4
(Plastic Fine Pitch Quad Flat Package)



2) Does not include dambar protrusion of 0.08 max. per side

1) Does not include plastic or metal protrusion of 0.25 max. per side

Sorts of Packing

Package outlines for shipping tubes, trays, etc. are contained in Siemens' Package Information Data Book.

SMD = Surface Mounted Device

Dimensions in mm

9 Appendix

9.1 General ATM Acronyms

Abbreviation	Meaning
AAL	ATM Adaptation Layer
ATM	Asynchronous Transfer Mode
ABR	Available Bit Rate
BOC	Begin of Cell
BOM	Begin of Message
CBR	Constant Bit Rate
CLP	Cell Loss Priority
COM	Continuation of Message
CPCS	Common Part Convergence Sublayer
CPCS-UU	CPCS User-to-User Indication
CPI	Common Part Indicator
CRC	Cyclic Redundancy Check
DW	Double Word (32 bits)
EOM	End of Message
GFC	Generic Flow Control
HDR	Header
HEC	Header Error Control
IOM	ISDN Oriented Modular
IRQ	Interrupt Request
ISDN	Integrated Services Digital Network
JTAG	Joint Test Action Group
LAN	Local Area Network
LI	Length Indication
LSB	Least Significant Bit / Least Significant Byte
MSB	Most Significant Bit / Most Significant Byte
NIC	Network Interface Card (terminal adapter card)
NNI	Network-to-Network Interface
OAM	Operation, Administration and Maintenance

Abbreviation	Meaning
PAD	Padding
PDU	Protocol Data Unit
PHY	Physical Layer Device
PMD	Physical Media Dependent (Layer)
PLL	Phase Locked Loop
PT	Payload Type
SAR	Segmentation & Reassembly
SARE	Segmentation And Reassembly Element, Siemens chip, PXB 4110
SDH	Synchronous Digital Hierarchy
SDU	Service Data Unit
SMDS	Switched Multi-megabit Data Service
SONET	Synchronous Optical NETwork
SSM	Single Segment Message
ST	Segment Type
STM-n	Synchronous Transport Module of Level n
TC	Transmission Convergence (Layer)
UBR	Undefined Bit Rate
UNI	User Network Interface
UTOPIA	Universal Test & Operations PHY Interface for ATM
VBR	Variable Bit Rate
VC	Virtual Circuit
VCI	Virtual Circuit Identifier
VP	Virtual Path
VPCI	Virtual Path/Channel Identifier
VPI	Virtual Path Identifier

9.2 Data Structure Acronyms

Abbreviation	Meaning
RB	Receive Buffer
RBD	Receive Buffer Descriptor
RFQ	Receive Buffer Free Queue
RMQ	Receive MID Table Queue
RMT	Receive MID Table
RRH	Reassembly Header
RRQ	Receive Ready Queue
RVT	Receive VPCI Table
TB	Transmit Buffer
TBD	Transmit Buffer Descriptor
TFQ	Transmit Buffer Free Queue
TPD	TPT Descriptor
TPT	Transmit PDU table
TRQ	Transmit Ready Queue
TVD	TVT Descriptor
TVT	Transmit VPCI Table
TWQ	Transmit Waiting Queue

9.3 Parameter/Variable Acronyms

Abbreviation	Meaning
AALS	AAL Select
AMCN	Active MID Count
CRDEC	Credit Decrement Value (M = Mantissa, E = Exponent)
CRMAX	Maximum Credit (M = Mantissa, E = Exponent)
PLEN	Payload Length
RCTD	Reassembly Context Descriptor
RERR	Reassembly Error Flags
SN	Sequence Number

9.4 Register Acronyms

Abbreviation	Meaning
CONF	Configuration Register
COSETR	COSET Reassembly Register
COSETS	COSET Segmentation Register
ECH	Empty Cell Header
ECPAT	Empty Cell Pattern
IMASK A/B	Interrupt Status Mask Register A/B
INTSTATE	Interrupt Status Register
RBB	RB Base Register
RFQB	RFB Base Register
RFQL	RFQ Length Register
RFQWR	RFQ Write Pointer
RMQB	RMQ Base Register
RMQL	RMQ Length Register
RMQWR	RMQ Write Pointer
RMTB	RMT Base Register
RRQmB	RRQm Base Register
RRQmL	RRQm Length Register
RRQmRD	RRQm Read Pointer
RVTBx	VPCI Group x RVT Base Entry
SMID	Selective MID Register
TBB	TB Base Register
TFQB	TFQ Base Register
TFQL	TFQ Length
TFQRD	TFQ Read Pointer
TIMESTAMP	Timestamp Register
TPTB	TPT Base Register
TRQB	TRQ Base Register
TRQL	TRQ Length Register
TRQRD	TRQ Read Pointer
TVTB	TVT Base Register
TWQB	TWQ Base Register
TWQL	TWQ Length Register
TWQWR	TWQ Write Pointer
VPCIx	VPCI Group Limit Register