- **Peripheral Component Interconnect (PCI) Device That Provides Asynchronous Transfer-Mode (ATM) Interface**
- **Single-Chip Segmentation and Reassembly (SAR) for Full-Duplex ATM Adaptation-Layer (AAL) Processing**
- **On-Chip PCI Host Interface Allows Use of Host Memory for Packet SAR**
- **53-Byte ATM Cells Are Transparent to the User**
- **Provides Complete Encapsulation and Termination of AAL5 and Limited AAL3/4**
- **Features a Null AAL That Provides Functions for Constant-Bit-Rate Services**
- **Supports 1023 Unique Virtual Circuits (VCs) on Receive Side**
- **Explicit Cell-Level Interleaving Between Groups of VCs**
- **Packet Interface Is Managed by Efficient Descriptor Rings**
- **Physical (PHY)-Layer Interface Is Full Duplex and Compliant to the ATM Forum UTOPIA Contribution**
- **Supports PHY-Layer Data Rates in the Range of 25.6 Mbit/s to 155.52 Mbit/s**
- **Interfaces Directly to the TNETA1500 SONET ATM BiCMOS Receiver/Transmitter (SABRE)**
- **Recognizes ATM-Layer Operation and Maintenance (OAM) Cells**
- **No External Logic Required for Host or Local Buses to Ensure Simple Design**
- **Supports an External EPROM for System Configuration Code**
- **Packaged in 240-Pin Plastic Quad Flatpack**

description

The TNETA1561 (PCI SAR) is an asynchronous transfer mode (ATM) segmentation and reassembly (SAR) device with a peripheral component interconnect (PCI)-bus interface. This device incorporates ATM adaptation-layer (AAL) processing, ATM SAR processing for full-duplex operation up to STS-3c rate of 155.52 Mbit/s, and the controls for the register interface on the physical (PHY) layer. The TNETA1561 provides a packet interface that is managed by descriptor rings, making the 53-byte ATM-framing format transparent to the user. The device passes the payload of 48 bytes, constituting the payload of each cell, across the PCI-host interface. All packets are segmented and reassembled in host memory and accessed by the chip via the descriptor-ring mechanism. The device reduces the memory requirements for network-interface cards (NICs). The TNETA1561 requires no local processor on the card, which enables very compact solutions.

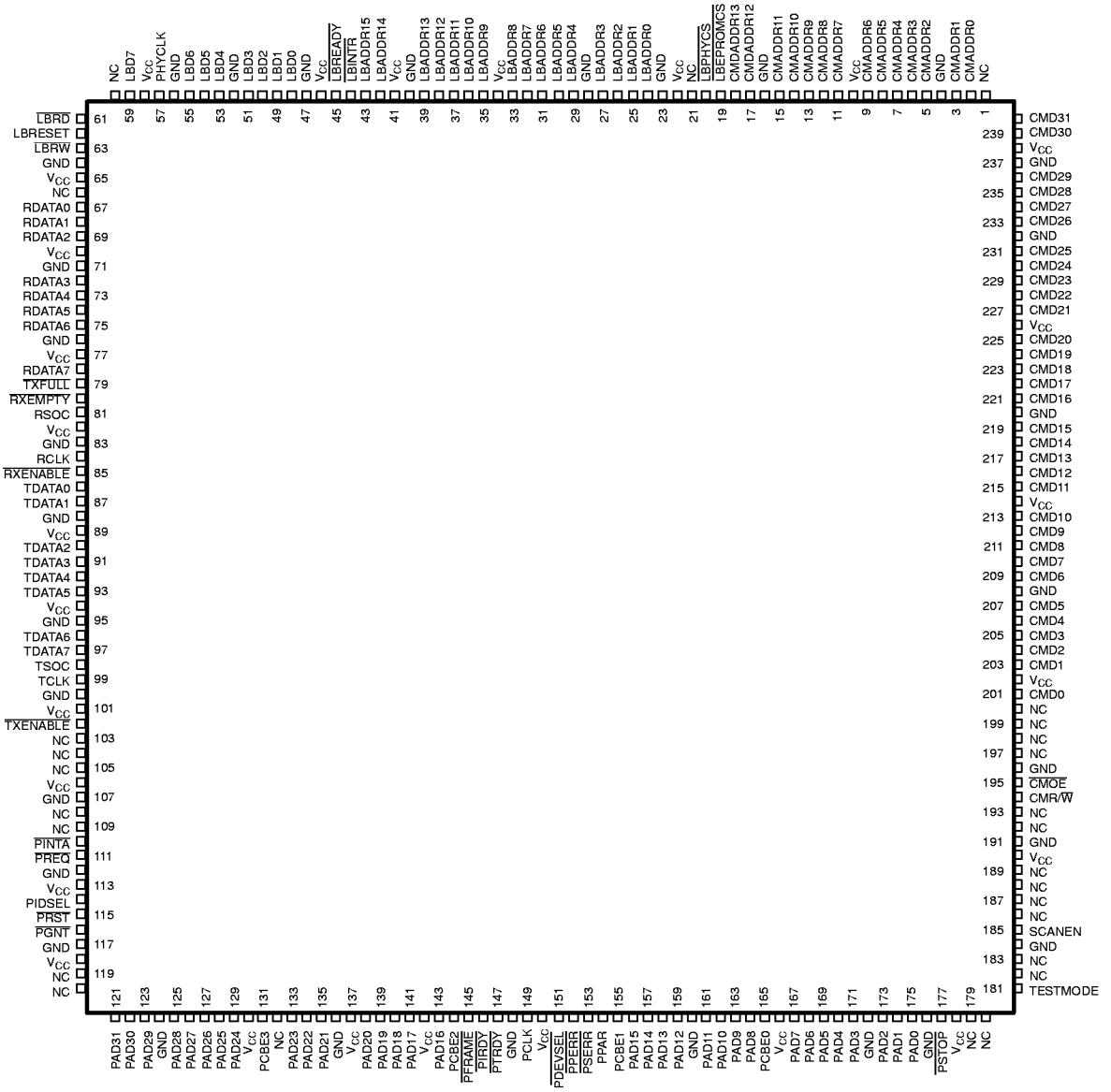
The applications for the TNETA1561 include NICs for client workstations and servers, embedded applications like LAN emulation, and multiprotocol systems like video servers. The TNETA1561 provides complete AAL5 encapsulation and termination in hardware. In addition, limited support is provided for AAL 3/4 and a null AAL is provided to facilitate real-time data transfer. The TNETA1561 recognizes ATM-layer operation and maintenance (OAM) cells.

In the transmit direction, the TNETA1561 generates data via a special bit-rate control table that provides explicit cell-level interleaving between groups of virtual circuits (VCs). This mechanism brings a higher degree of flexibility when specifying peak rates for each group (up to 155.52 Mbit/s at a resolution greater than 32 kbit/s). The VCs within a group are serviced via a first-in, first-out (FIFO) discipline on a per-packet basis.

In the receive direction, the TNETA1561 allows multiple virtual paths (VPs) with the condition that each VC is unique. The device is primarily intended for AAL5 encapsulation and termination that is supported in hardware.

The TNETA1561 has four interfaces that include the following: the PCI-bus interface with a 32-bit-wide data bus, the cell interface based on the universal test and operations interface for ATM (UTOPIA specification), a control-memory interface to access the local SRAM, and the local-bus interface to access the PHY-layer registers. The UTOPIA interface to the PHY layer consists of an 8-bit-wide datapath and associated control signals in both the transmit and receive directions. The 53-byte ATM cells pass between the ATM and PHY layers through the UTOPIA interface.

PGC PACKAGE
(TOP VIEW)



NC – No internal connection

description (continued)

The native clock for the TNETA1561 is the PCI clock, which operates up to 33 MHz. The native-word size for the device is 32 bits, corresponding to the data width for the PCI bus. The TNETA1561 host interface implements the bus protocol defined in the PCI local-bus specification (revision 2.0). The control-memory interface is 32 bits wide. This interface allows the device to access the local memory to obtain the control information on the packets being segmented and reassembled and to obtain their locations in host memory. Each packet queued for transmission can be distributed across multiple buffers in host memory, with each starting at any byte boundary. This is supported in hardware by the device. Every received package is placed in a single buffer in the host memory and is aligned to a 16-byte boundary. The TNETA1561 operation is explained in detail in the *Principles of Operation* section.

Terminal Functions

PCI-bus interface

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
$\overline{\text{PINTA}}$	110	O (open drain)	PCI interrupt. $\overline{\text{PINTA}}$ is an interrupt request from PCI SAR.
PAD31–PAD0	121–123, 125–129, 133–135, 138–141, 143, 156–159, 161–164, 167–170, 171, 173–175	I/O (3 state)	PCI address bus and data bus. PAD31–PAD0 are multiplexed on the same PCI terminals. During the first phase of a transaction (address phase), PAD31–PAD0 contains a 32-bit physical address. This phase is the clock cycle when $\overline{\text{PFRAME}}$ is asserted. During the data phase, PAD7–PAD0 contain the least significant byte and PAD31–PAD24 contain the most significant byte. Write data is stable when $\overline{\text{PIRDY}}$ is asserted. Read data is stable when $\overline{\text{PTRDY}}$ is asserted. Data is transferred during those clock cycles when both $\overline{\text{PIRDY}}$ and $\overline{\text{PTRDY}}$ are asserted.
PCBE3–PCBE0	131, 144, 155, 165	I/O (3 state)	PCI-bus command and byte enable. PCBE3–PCBE0 lines are multiplexed on the same PCI terminals. During the address phase of a transaction, PCBE3–PCBE0 lines define the bus command. During the data phase, PCBE3–PCBE0 lines define which bytes are valid.
PCLK	149	I	PCI clock. PCLK provides timing for all transactions on PCI.
$\overline{\text{PDEVSEL}}$	151	I/O (3 state)	PCI device select. $\overline{\text{PDEVSEL}}$, when actively driven, indicates that the driving device has decoded its address as the target of the current access. As an input, $\overline{\text{PDEVSEL}}$ indicates whether any device on the bus is selected.
$\overline{\text{PFRAME}}$	145	I/O (3 state)	PCI frame. $\overline{\text{PFRAME}}$ is driven by the current master to indicate the beginning and duration of an access. $\overline{\text{PFRAME}}$ is asserted at the beginning of the bus transaction and remains asserted during data transfer. When $\overline{\text{PFRAME}}$ is deasserted, the transaction is in the final data phase.
$\overline{\text{PGNT}}$	116	I	PCI bus grant. $\overline{\text{PGNT}}$ indicates to the agent that the arbiter has granted access to the bus. $\overline{\text{PGNT}}$ is a point-to-point signal and every master has its own.
PIDSEL	114	I	PCI initialization and device select. PIDSEL is used as a chip select during configuration read and write transactions.
$\overline{\text{PIRDY}}$	146	I/O (3 state)	PCI initiator ready. $\overline{\text{PIRDY}}$ indicates the initiating agent's (bus master) ability to complete the current data phase of the transaction. During a write, $\overline{\text{PIRDY}}$ indicates valid data on PAD31–PAD0. During a read, $\overline{\text{PIRDY}}$ indicates that the master is prepared to accept the data. $\overline{\text{PIRDY}}$ is used with $\overline{\text{PTRDY}}$ when wait cycles are inserted until both $\overline{\text{PIRDY}}$ and $\overline{\text{PTRDY}}$ are asserted.
PPAR	154	I/O (3 state)	PCI parity. PPAR is even across PAD31–PAD0 and PCBE3–PCBE0. For data phases, PPAR is valid one clock after either $\overline{\text{PIRDY}}$ is asserted on a write or $\overline{\text{PTRDY}}$ is asserted on a read. Once asserted, PPAR remains valid until one clock after the completion of the current data phase. The master drives the PPAR for the address and write-data phases. The target drives PPAR for the read-data phase.
$\overline{\text{PPERR}}$	152	I/O (3 state)	PCI parity error. $\overline{\text{PPERR}}$ reports a data-parity error on all commands except special cycle. An agent cannot report a $\overline{\text{PPERR}}$ until it has claimed the access by $\overline{\text{PDEVSEL}}$ and completed a data phase.
$\overline{\text{PREQ}}$	111	O	PCI request. $\overline{\text{PREQ}}$ indicates to the arbiter that this agent desires use of the bus. Every master has its own $\overline{\text{PREQ}}$.
$\overline{\text{PRST}}$	115	I	PCI reset. $\overline{\text{PRST}}$ forces the PCI sequence of each device to a known state.
$\overline{\text{PSERR}}$	153	I/O (open drain)	PCI system error. $\overline{\text{PSERR}}$ reports address-parity errors and data-parity errors on special-cycle commands.
$\overline{\text{PSTOP}}$	177	I/O (3 state)	PCI stop. $\overline{\text{PSTOP}}$ indicates the current target is requesting the master to stop the current transaction.
$\overline{\text{PTRDY}}$	147	I/O (3 state)	PCI target ready. $\overline{\text{PTRDY}}$ indicates the target agent's (selected device) ability to complete the current data phase of the transaction. During a read, $\overline{\text{PTRDY}}$ indicates that valid data is present on PAD31–PAD0. During a write, $\overline{\text{PTRDY}}$ indicates that the target is prepared to accept data.

Terminal Functions (Continued)

PCI SAR and local-bus interface

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
LBRESET	62	O	Local-bus reset. LBRESET is an active-high signal that is driven by the PCI SAR.
$\overline{\text{LBPHYCS}}$	20	O	Local-bus PHY-layer chip select. $\overline{\text{LBPHYCS}}$ is used to interface with PHY-layer devices and is driven by PCI SAR.
$\overline{\text{LBEPROMCS}}$	19	O	Local-bus EPROM chip select. $\overline{\text{LBEPROMCS}}$ is an active-low signal that is driven by PCI SAR.
$\overline{\text{LBRW}}$	63	O	Local-bus write. $\overline{\text{LBRW}}$ is an active-low write signal that indicates a write operation and is driven by PCI SAR.
$\overline{\text{LBRD}}$	61	O	Local-bus read. $\overline{\text{LBRD}}$ is an active-low read signal that indicates a read operation and is driven by PCI SAR.
$\overline{\text{LBINTR}}$	44	I	Local-bus interrupt. $\overline{\text{LBINTR}}$ is generated and driven by a local-bus device.
$\overline{\text{LBREADY}}$	45	I	Local-bus ready. $\overline{\text{LBREADY}}$ is driven by local-slave devices. The bus transaction is completed after eight PCI bus cycles, regardless of $\overline{\text{LBREADY}}$. $\overline{\text{LBREADY}}$ is accepted by the SAR as a handshake from the devices on the bus.
LBD7–LBD0	59, 55–53, 51–48	I/O	Local-bus data. LBD7–LBD0 are used to transfer data to and from local-slave devices and are driven by PCI SAR or local-slave devices.
LBADDR15–LBADDR0	43–42, 39–35, 33–29, 27–24	O	Local-bus address. LBADDR15–LBADDR0 are driven by the PCI SAR and are used to address the PHY-layer registers and the EPROM. LBADDR15–LBADDR2 are received from the PCI bus and LBADDR1–LBADDR0 are generated by the PCI SAR.

PHY-layer receive interface

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
RCLK	84	O	Receive clock. RCLK is equivalent to the internal clock at 19.44 MHz. RCLK is sent to the PHY layer.
RDATA7–RDATA0	78, 75–72, 69–67	I	Receive data. RDATA7–RDATA0 are connected to the PHY-layer receive interface and are driven by the PHY-layer device.
RSOC	81	I	Receive start of cell. RSOC is a signal from the PHY layer indicating that the first byte of an ATM cell was sent to the TNETA1561.
$\overline{\text{RXEMPTY}}$	80	I	Receive buffer empty in the PHY layer. $\overline{\text{RXEMPTY}}$ is a signal that acts as an inverted enable signal on this interface and is driven by the PHY layer.
$\overline{\text{RXENABLE}}$	85	O	Receive enable. $\overline{\text{RXENABLE}}$ is active low and is driven by the TNETA1561.

Terminal Functions (Continued)

PHY-layer transmit interface

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
TCLK	99	O	Transmit clock. The TNETA1561 generates TCLK at the PCI-clock frequency and sends it to the PHY layer. TCLK is an inverted version of the internal clock.
TDATA7–TDATA0	97–96, 93–90, 87–86	O	Transmit data. TDATA7–TDATA0 are sent at the rate of the PCI clock and are driven by the TNETA1561.
TSOC	98	O	Transmit start of cell. TSOC is sent by the PCI SAR to the PHY layer and indicates that the first byte of an ATM cell was transmitted to the PHY layer.
$\overline{\text{TXENABLE}}$	102	O	Transmit enable. The SAR turns off $\overline{\text{TXENABLE}}$ when the PHY layer sends the $\overline{\text{TXFULL}}$ signal.
$\overline{\text{TXFULL}}$	79	I	Transmit buffer full in the PHY layer. The PHY layer asserts $\overline{\text{TXFULL}}$ at least four cycles before any internal buffers are full. This makes the TNETA1561 stop the data transmission to the PHY layer.

PCI SAR and control-memory interface

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
CMADDR13–CMADDR0	18–17, 15–11, 9–5, 3–2	O	Control-memory address. CMADDR13–CMADDR0 is a 14-bit address bus and is driven by the PCI SAR.
CMD31–CMD0	240–239, 236–233, 231–227, 225–221, 219–215, 213–209, 207–203, 201	I/O	Control-memory data. The control-memory interface has a 32-bit data bus. CMD31–CMD0 are designed for 20-ns asynchronous SRAMs. The TNETA1561 uses this interface to access the data structures and pointers in the control memory.
$\overline{\text{CMOE}}$	195	O	Control-memory output enable. $\overline{\text{CMOE}}$ is an active-low signal and is driven by the PCI SAR.
CMR/$\overline{\text{W}}$	194	O	Control-memory read/write. CMR/$\overline{\text{W}}$ determines a read or write operation. If the output is low, it is a write operation. If the output is high, it is a read operation. CMR/$\overline{\text{W}}$ is driven by the PCI SAR.

Terminal Functions (Continued)

PCI SAR and test/control interface

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
NC	1, 21, 60, 66, 103–105, 108, 109, 119, 120, 132, 179, 180, 182, 183, 186–189, 192, 193, 197–200	O	No connection. Leave open.
PHYCLK	57	I	PHY-layer clock. PHYCLK is a 19.44-MHz clock signal driven by a PHY-layer clock crystal.
SCANEN	185	I	SCAN enable. Connect to ground for normal operation.
TESTMODE	181	I	Test mode. TESTMODE is driven by the test system. Leave grounded for normal operation.

power and ground

TERMINAL NAME	TERMINAL NO.	DESCRIPTION
GND	4, 16, 23, 28, 40, 47, 52, 56, 64, 71, 76, 83, 88, 95, 100, 107, 112, 117, 124, 136, 148, 160, 172, 176, 184, 191, 196, 208, 220, 232, 237	Ground
V_{CC}	10, 22, 34, 41, 46, 58, 65, 70, 77, 82, 89, 94, 101, 106, 113, 118, 130, 137, 142, 150, 166, 178, 190, 202, 214, 226, 238	Supply voltage

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, V_{CC} (see Note 1) . . . . . –0.5 V to 6 V
Input voltage range, V_I . . . . . –0.5 V to V_{CC} + 0.5 V
Output voltage range, V_O . . . . . –0.5 V to V_{CC} + 0.5 V
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{CC}) (see Note 2) . . . . . ±20 mA
Output clamp current, I_{OK} (V_O < 0 or V_O > V_{CC}) (see Note 3) . . . . . ±20 mA
Operating free-air temperature range, T_A . . . . . 0°C to 70°C
Storage temperature range, T_{stg} . . . . . –65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values are with respect to the GND terminals.
2. Applies for external input and bidirectional buffers
3. Applies for external output and bidirectional buffers

recommended operating conditions

				MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage			4.75	5	5.25	V
V_{IH}	High-level input voltage	CMOS	V_{CC} = 4.75 V	3.325			V
			V_{CC} = 5.25 V	3.675			
		TTL		2			
V_{IL}	Low-level input voltage	CMOS	V_{CC} = 4.75 V			0.95	V
			V_{CC} = 5.25 V			1.05	
		TTL				0.8	
T_A	Operating free-air temperature			0		70	°C

electrical characteristics over recommended operating conditions (unless otherwise noted), T_A = 25°C

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	I_{OH} = 8 mA		V_{CC}–0.8			V
		I_{OL} = 4 mA		V_{CC}–0.8			
V_{OL}	Low-level output voltage	I_{OH} = 8 mA				0.5	V
		I_{OL} = 4 mA				0.5	
I_{OZ}	High-impedance-state output current	V_I = V_{CC} or GND				±10	µA
I_{IL}	Low-level input current	V_I = GND				–1	µA
I_{IH}	High-level input current	V_I = V_{CC}				1	µA
I_{CC}‡	Supply current, dynamic	V_{CC} = 5 V,	f_{clock} = 33 MHz		300		mA

‡ The supply-current (I_{CC}) typical value is given for a full-duplex operation with f_{clock} equal to 33 MHz.

timing requirements (see Note 4 and Figure 1)

NO.			MIN	MAX	UNIT
1	$t_{w(RCLKH)}$	Pulse duration, RCLK high	12		ns
2	$t_{w(RCLKL)}$	Pulse duration, RCLK low	12		ns
3	$t_{su(RSOC)}$	Setup time, RSOC high before RCLK↑	10		ns
4	$t_{su(RXEMPTY)}$	Setup time, $\overline{\text{RXEMPTY}}$ low before RCLK↑	10		ns
5	$t_{su(RDATA)}$	Setup time, RDATA7–RDATA0 valid before RCLK↑	10		ns
6	$t_{h(RSOC)}$	Hold time, RSOC high after RCLK↑	1		ns
7	$t_{h(RXEMPTY)}$	Hold time, $\overline{\text{RXEMPTY}}$ low after RCLK↑	1		ns

NOTE 4: All output signals are generated on the rising edge of RCLK.

operating characteristics (see Note 4 and Figure 1)

NO.			MIN	MAX	UNIT
8	$t_{d(RXENABLE)}$	Delay time, RCLK ↑ to $\overline{\text{RXENABLE}}$↑	1	20	ns

NOTE 4: All output signals are generated on the rising edge of RCLK.

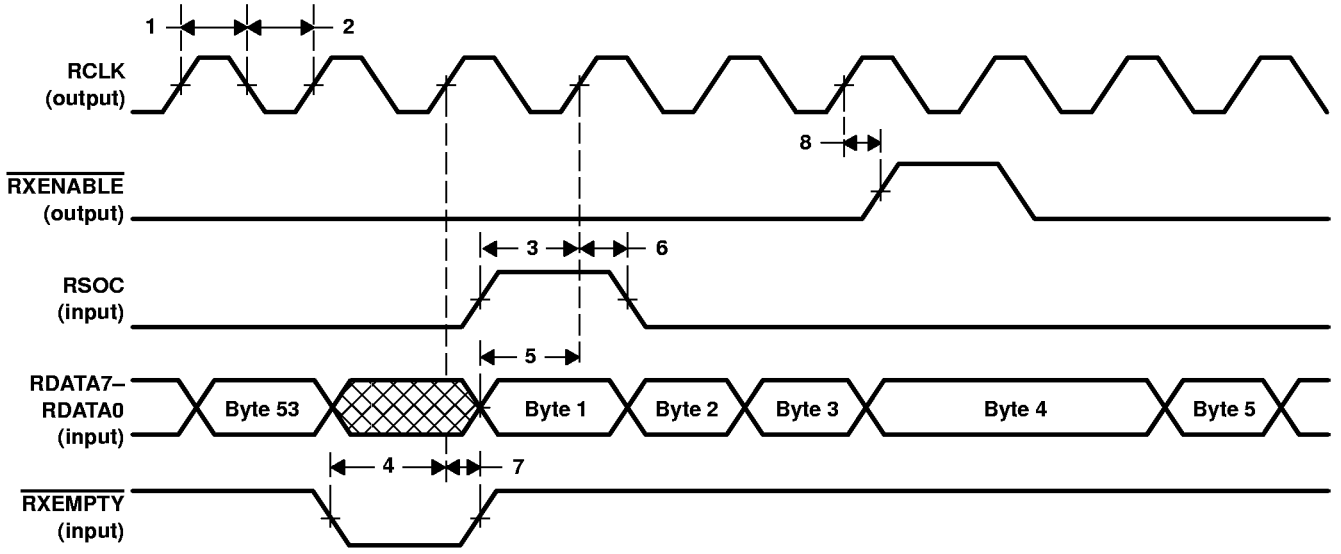


Figure 1. Receive-Cell Interface

timing requirements (see Note 5 and Figure 2)

NO.			MIN	MAX	UNIT
1	$t_{w(TCLKH)}$	Pulse duration, TCLK high	12	18	ns
2	$t_{w(TCLKL)}$	Pulse duration, TCLK low	12	18	ns
3	$t_{su(TXFULL)}$	Setup time, $\overline{\text{TXFULL}}$ low before TCLK↑	10		ns
4	$t_{h(TXFULL)}$	Hold time, $\overline{\text{TXFULL}}$ low after TCLK↑	1		ns

NOTE 5: All output signals are generated on the rising edge of RCLK. All inputs are sampled on the rising edge of TCLK.

operating characteristics (see Note 5 and Figure 2)

NO.			MIN	MAX	UNIT
5	$t_{d(TXENABLE)}$	Delay time, TCLK↑ to $\overline{\text{TXENABLE}}$↓	1	20	ns
6	$t_{d(TSOC)}$	Delay time, TCLK↑ to TSOC↑	1	20	ns
7	$t_{d(TDATA)}$	Delay time, TCLK↑ to TDATA7–TDATA0 valid	1	20	ns

NOTE 5: All output signals are generated on the rising edge of RCLK. All inputs are sampled on the rising edge of TCLK.

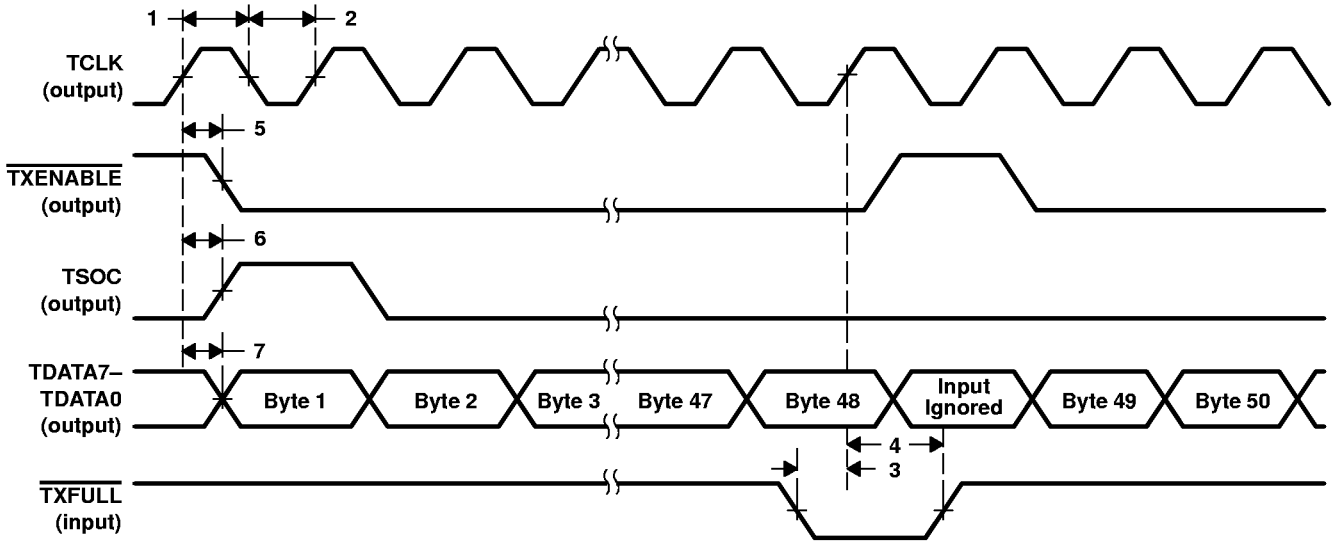


Figure 2. Transmit-Cell Interface

timing requirements (see Figure 3)

NO.			MIN	MAX	UNIT
1	$t_{su(PGNT)}$	Setup time, $\overline{PGNT}$ low before PCLK↑	10		ns
2	$t_{su(PTRDY)}$	Setup time, $\overline{PTRDY}$ low before PCLK↑	7		ns
3	$t_{su(PDEVSEL)}$	Setup time, $\overline{PDEVSEL}$ low before PCLK↑	7		ns
4	$t_{h(PTRDY)}$	Hold time, $\overline{PTRDY}$ low after PCLK↑	3		ns
5	$t_{h(PDEVSEL)}$	Hold time, $\overline{PDEVSEL}$ low after PCLK↑	3		ns

operating characteristics (see Figure 3)

NO.			MIN	MAX	UNIT
6	$t_{d(PREQ)}$	Delay time, PCLK↑ to $\overline{PREQ}$↓	2	12	ns
7	$t_{d(PFRAME)}$	Delay time, PCLK↑ to $\overline{PFRAME}$↓	2	11	ns
8	$t_{d(PCBE)}$	Delay time, PCLK↑ to PCBE3–PCBE0 valid	2	11	ns
9	$t_{d(PIRDY)}$	Delay time, PCLK↑ to $\overline{PIRDY}$↓	2	11	ns
10	$t_{d(PAD)}$	Delay time, PCLK↑ to PAD31–PAD0 valid	2	11	ns

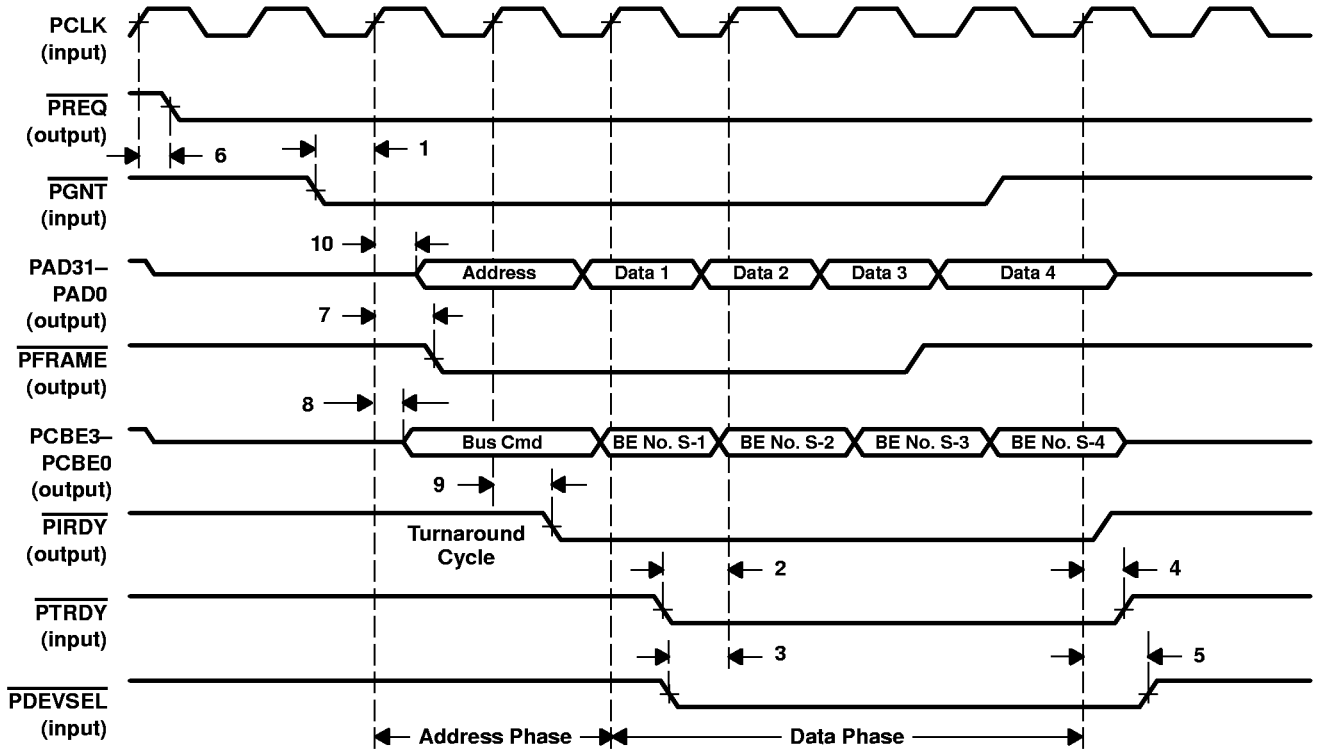


Figure 3. TNETA1561 Write Operation (PCI SAR as Master)

timing requirements (see Figure 4)

NO.			MIN	MAX	UNIT
1	$t_{w(PCLKH)}$	Pulse duration, PCLK high	12		ns
2	$t_{w(PCLKL)}$	Pulse duration, PCLK low	13		ns
3	$t_{su(PGNT)}$	Setup time, $\overline{PGNT}$ low before PCLK↑	10		ns
4	$t_{su(PAD)}$	Setup time, PAD31–PAD0 valid before PCLK↑	7		ns
5	$t_{su(PTRDY)}$	Setup time, $\overline{PTRDY}$ low before PCLK↑	7		ns
6	$t_{su(PDEVSEL)}$	Setup time, $\overline{PDEVSEL}$ low before PCLK↑	7		ns
7	$t_{h(PAD)}$	Hold time, PAD31–PAD0 valid after PCLK↑	2		ns
8	$t_{h(PTRDY)}$	Hold time, $\overline{PTRDY}$ low after PCLK↑	3		ns
9	$t_{h(PDEVSEL)}$	Hold time, $\overline{PDEVSEL}$ low after PCLK↑	3		ns

operating characteristics (see Figure 4)

NO.			MIN	MAX	UNIT
10	$t_{d(PFRAME)}$	Delay time, PCLK↑ to $\overline{PFRAME}$↓	2	11	ns
11	$t_{d(PAD)}$	Delay time, PCLK↑ to PAD31–PAD0 valid	2	11	ns
12	$t_{d(PCBE)}$	Delay time, PCLK↑ to PCBE3–PCBE0 valid	2	11	ns
13	$t_{d(PIRDY)}$	Delay time, PCLK↑ to $\overline{PIRDY}$↓	2	11	ns
14	$t_{d(PREQ)}$	Delay time, PCLK↑ to $\overline{PREQ}$↓	2	12	ns

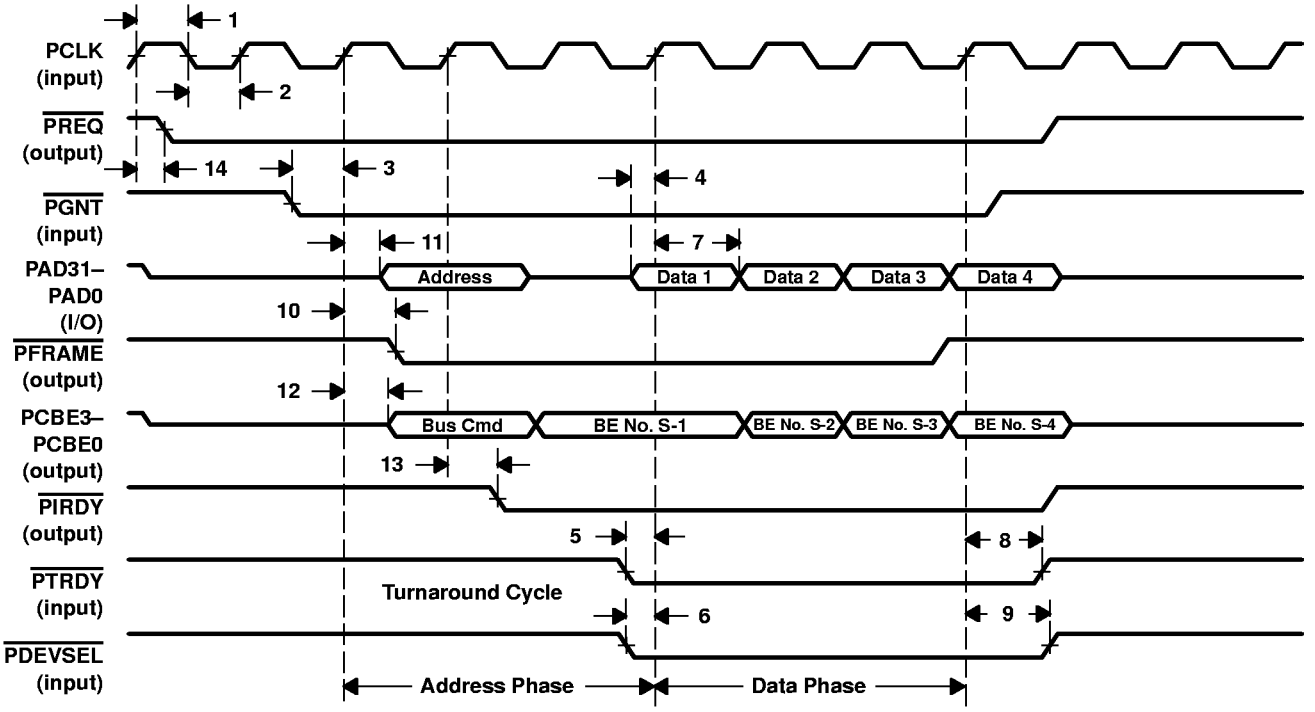


Figure 4. TNETA1561 Read Operation (PCI SAR as Master)

timing requirements (see Figure 5)

NO.			MIN	MAX	UNIT
1	$t_{su(PIDSEL)}$	Setup time, PIDSEL high before PCLK↑	7		ns
2	$t_{su(PAD)}$	Setup time, PAD31–PAD0 valid before PCLK↑	7		ns
3	$t_{su(PCBE)}$	Setup time, PCBE3–PCBE0 valid before PCLK↑	7		ns
4	$t_{su(PIRDY)}$	Setup time, $\overline{\text{PIRDY}}$ low before PCLK↑	7		ns
5	$t_{h(PIDSEL)}$	Hold time, PIDSEL high after PCLK↑	1		ns
6	$t_{h(PAD)}$	Hold time, PAD31–PAD0 valid after PCLK↑	0		ns
7	$t_{h(PCBE)}$	Hold time, PCBE3–PCBE0 valid after PCLK↑	1		ns
8	$t_{h(PIRDY)}$	Hold time, $\overline{\text{PIRDY}}$ low after PCLK↑	1		ns

operating characteristics (see Figure 5)

NO.			MIN	MAX	UNIT
9	$t_{d(PAD)}$	Delay time, PCLK↑ to PAD31–PAD0 valid	2	11	ns
10	$t_{d(PTRDY)}$	Delay time, PCLK↑ to $\overline{\text{PTRDY}}$↓	2	11	ns
11	$t_{d(PDEVSEL)}$	Delay time, PCLK↑ to $\overline{\text{PDEVSEL}}$↓	2	11	ns

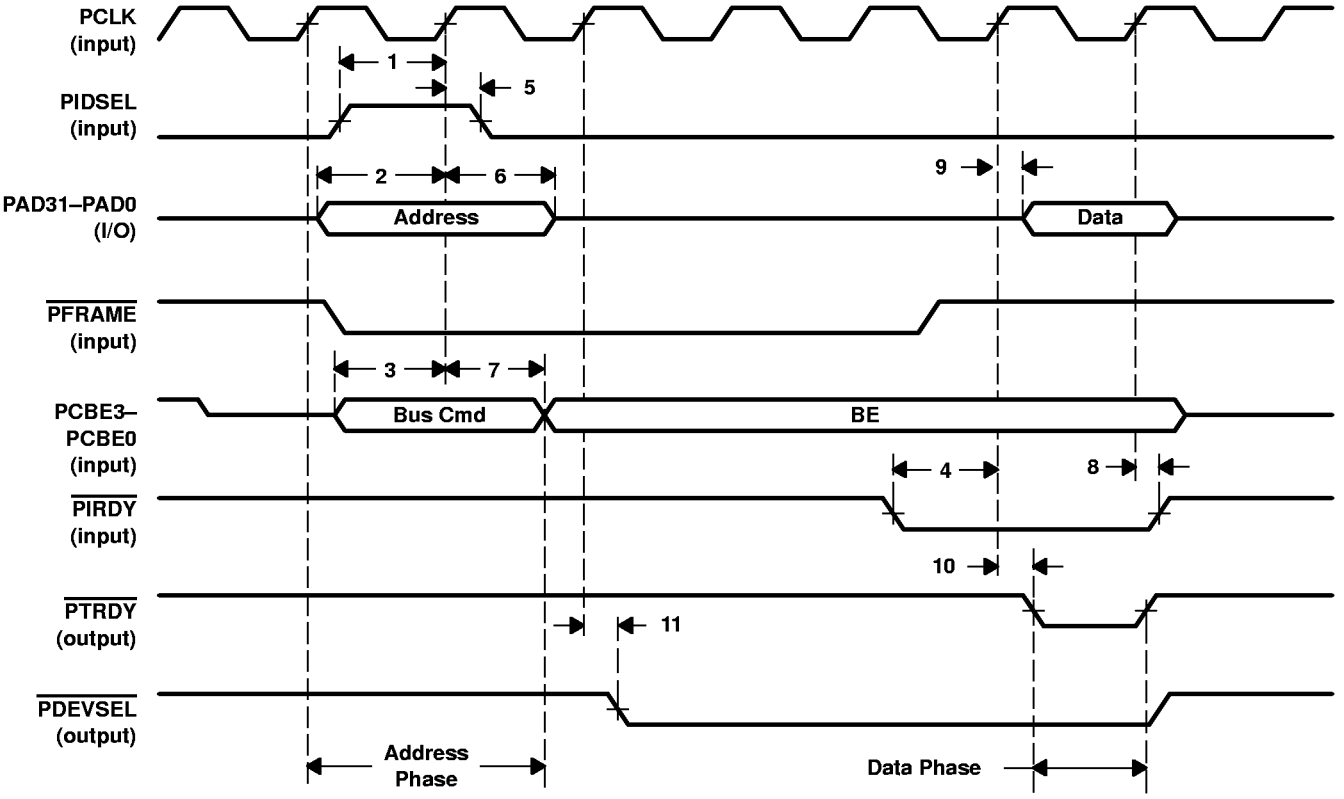


Figure 5. TNETA1561 Read Operation (PCI SAR as Slave)

timing requirements (see Figure 6)

NO.			MIN	MAX	UNIT
1	$t_{su(PIDSEL)}$	Setup time, PIDSEL high before PCLK↑	7		ns
2	$t_{su(PAD)}$	Setup time, PAD31–PAD0 valid before PCLK↑	7		ns
3	$t_{su(PFRAME)}$	Setup time, $\overline{\text{PFRAME}}$ low before PCLK↑	7		ns
4	$t_{su(PIRDY)}$	Setup time, $\overline{\text{PIRDY}}$ low before $\overline{\text{PCLK}}$↑	7		ns
5	$t_{h(PIDSEL)}$	Hold time, PIDSEL high after PCLK↑	2		ns
6	$t_{h(PIRDY)}$	Hold time, $\overline{\text{PIRDY}}$ low after PCLK↑	1		ns
7	$t_{h(PFRAME)}$	Hold time, $\overline{\text{PFRAME}}$ low after PCLK↑	3		ns

operating characteristics (see Figure 6)

NO.			MIN	MAX	UNIT
8	$t_{d(PTRDY)}$	Delay time, PCLK↑ to $\overline{\text{PTRDY}}$↓	2	11	ns

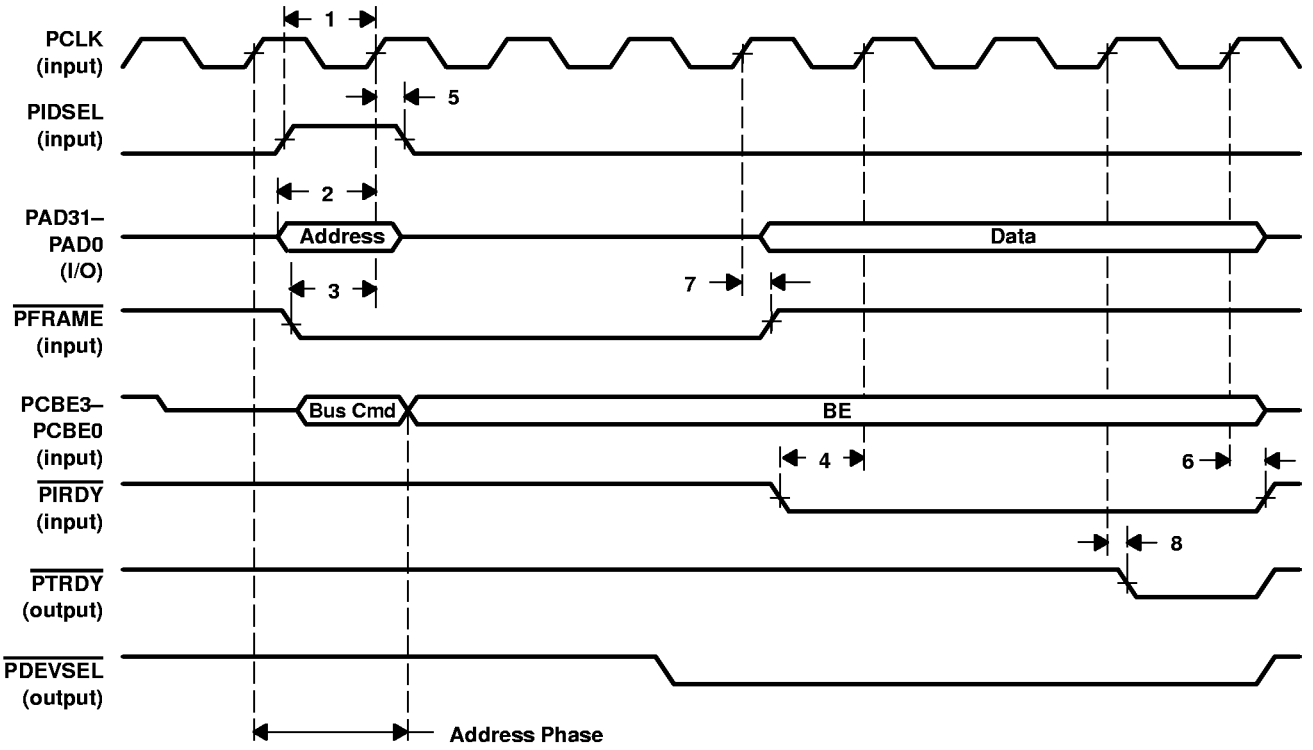


Figure 6. TNETA1561 Write Operation (PCI SAR as Slave)

operating characteristics (see Note 6 and Figure 7)

NO.			MIN	TYP	MAX	UNIT
1	$t_{d(LBPHYCS)1}$	Delay time, $\overline{LBRD}\downarrow$ to $\overline{LBPHYCS}\downarrow$		30†		ns
2	$t_{d(LBPHYCS)2}$	Delay time, $\overline{LBREADY}\downarrow$ to $\overline{LBPHYCS}\uparrow$		95†		ns

† These values are given for operation with a 33-MHz PCI clock.

NOTE 6: If $\overline{LBREADY}$ does not go active low within eight PCI clocks after $\overline{LBPHYCS}$ goes active low, the TNETA1561 latches in the data of the LBD7–LBD0 bus and terminates the read operation.

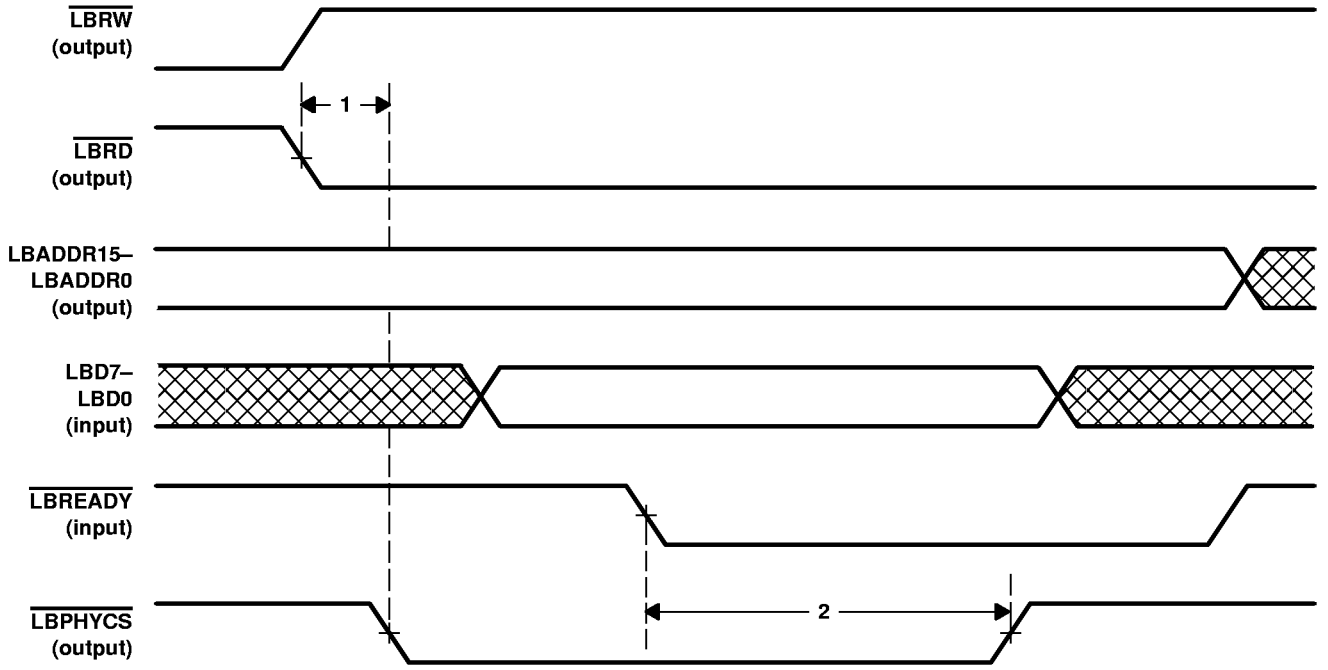


Figure 7. Local-Bus-Interface Read Operation (TNETA1561 as Slave)

operating characteristics (see Figure 8)†

NO.			MIN	TYP	MAX	UNIT
1	$t_{d(LBPHYCS)1}$	Delay time, $\overline{LBRW}\downarrow$ to $\overline{LBPHYCS}\downarrow$		30‡		ns
2	$t_{d(LBPHYCS)2}$	Delay time, LBADDR15–LBADDR0 valid to $\overline{LBPHYCS}\downarrow$		210‡		ns
3	$t_{d(LBPHYCS)3}$	Delay time, LBD7–LBD0 valid to $\overline{LBPHYCS}\downarrow$		30‡		ns
4	$t_{d(LBADDR)}$	Delay time, $\overline{LBPHYCS}\uparrow$ to LBADDR15–LBADDR0 invalid		150‡		ns

† $\overline{LBPHYCS}$ is asserted low for eight PCI-clock cycles during a write operation to allow access to slow devices.
‡ These values are given for operation with a 33-MHz PCI clock.

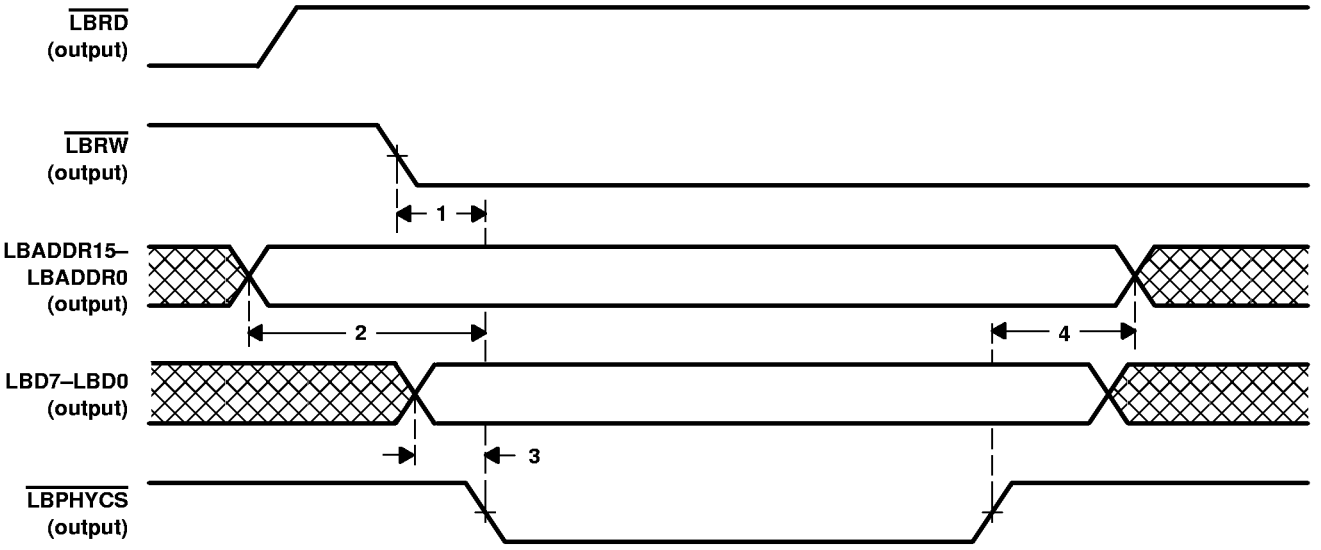


Figure 8. Local-Bus-Interface Write Operation (TNETA1561 as Slave)

operating characteristics (see Figure 9)

NO.			MIN	TYP	MAX	UNIT
1	$t_{w(CMR/WL)}$	Pulse duration, CMR/$\overline{W}$ low		29†		ns
2	$t_{d(CMR/W)1}$	Delay time, CMADDR13–CMADDR0 valid to CMR/$\overline{W}\downarrow$		16†		ns
3	$t_{d(CMR/W)2}$	Delay time, CMD31–CMD0 valid to CMR/$\overline{W}\uparrow$		42†		ns
4	$t_{d(CMD)}$	Delay time, CMR/$\overline{W}\uparrow$ to CMD31–CMD0 invalid		16†		ns

† These values are given for operation with a 33-MHz PCI clock.

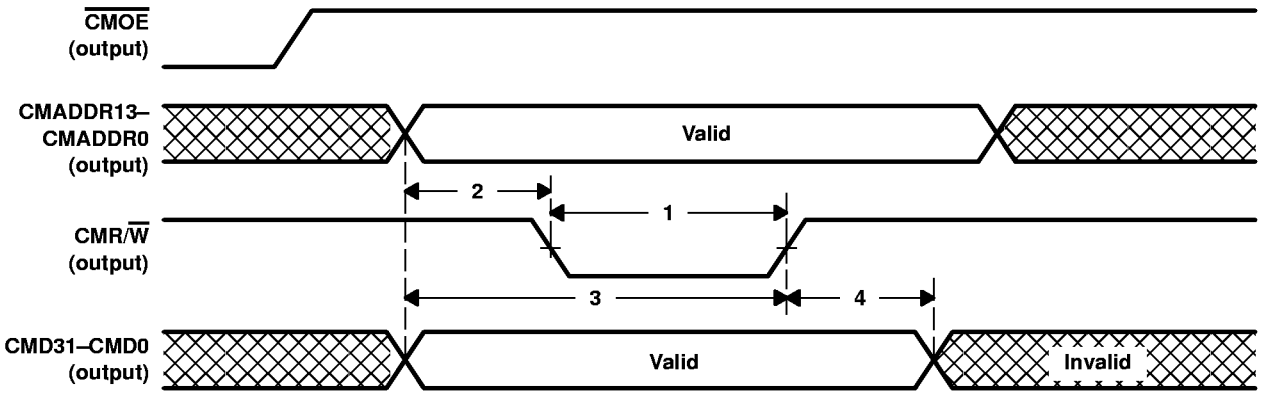


Figure 9. Control-Memory-Interface Write Operation

timing requirements (see Figure 10)

NO.			MIN	MAX	UNIT
1	$t_{su(CMD)}$	Setup time, CMD31–CMD0 valid before $\overline{CMOE}\uparrow$	7†		ns
2	$t_{h(CMD)}$	Hold time, CMD31–CMD0 valid after $\overline{CMOE}\uparrow$	0†		ns

† These values are given for operation with a 33-MHz PCI clock.

operating characteristics (see Figure 10)

NO.			MIN	TYP	MAX	UNIT
3	$t_{d(CMOE)}$	Delay time, CMADDR13–CMADDR0 valid to $\overline{CMOE}\downarrow$		1		ns

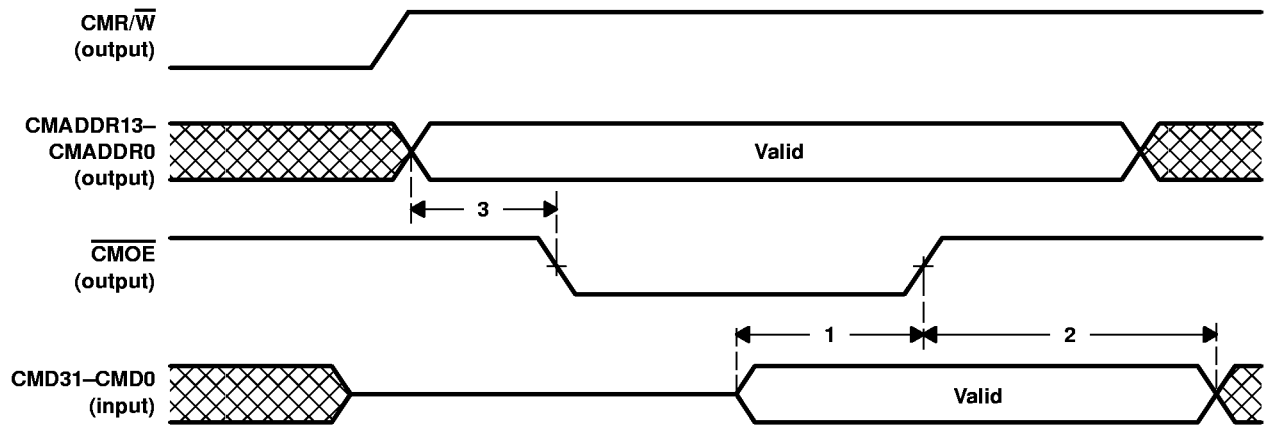


Figure 10. Control-Memory-Interface Read Operation

PRINCIPLES OF OPERATION

Table of Contents

functional overview (see Figure 11)

The TNETA1561 (PCI SAR) implements the PCI-bus protocols for connecting a peripheral device to a PCI-bus host system (mapped in memory space) and is designed for the PCI plug-in card concept. The central-resource functions, such as PCI-bus arbitration, are implemented by the host processor using the PCI SAR adapter.

The PCI SAR provides the PCI-configuration space to support its configuration and initialization. This configuration space specifies data for initialization software and error-handling software. The PCI SAR supports the mechanism to implement an external-EPROM interface for device-specific initialization and other booting mechanisms.

The PCI bus uses bursts as the basic mechanism to transfer data. The TNETA1561 supports data-burst sizes up to 52 bytes for a PCI-bus access, requiring a total of 13 data-phase transfers. A 13-word data transfer occurs when the payload data is not 32-bit aligned. The typical latency for PCI-bus access is 2 μs, but this PCI SAR provides adequate data buffering for a worst-case latency of up to 30 μs.

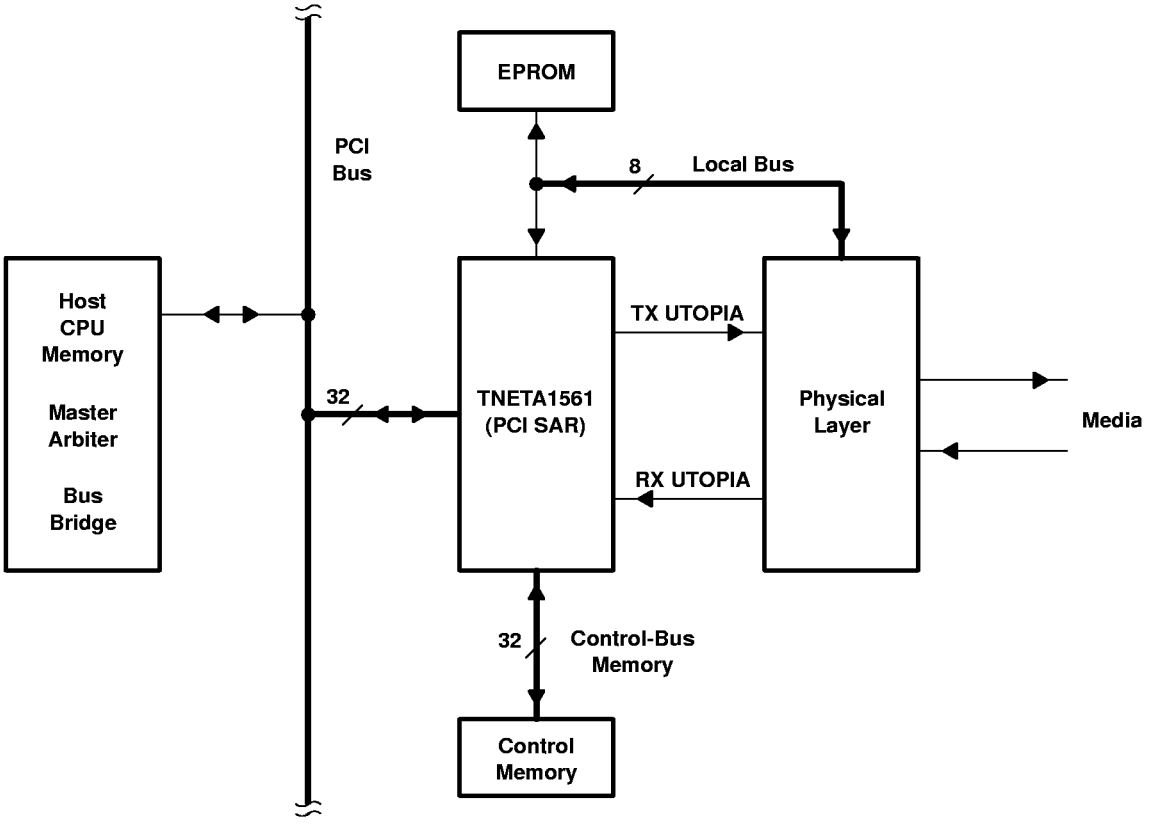


Figure 11. TNETA1561 Architecture

PRINCIPLES OF OPERATION

functional overview (see Figure 11) (continued)

The TNETA1561 provides a packet interface that is managed by descriptor rings to make the 53-byte ATM framing format transparent to the user. The PCI SAR passes the 48-byte payload of each cell across the PCI bus. All packets are stored in host memory and accessed by PCI SAR by the descriptor-ring mechanism.

The PCI SAR generates data in the transmit direction via a special bit-rate control table that provides explicit cell-level interleaving between groups of VCs. This mechanism provides a high degree of flexibility in specifying peak rates for each group of up to 155 Mbit/s at a resolution greater than 32 kbit/s. The VCs within a group are serviced via a FIFO discipline on a per-packet basis.

The PCI SAR supports 1023 unique VCs, typically all associated with virtual path identifier (VPI) 0. VPI 0 allows multiple VPs, with the reminder that each VC is unique. Limited support is provided to recognize ATM-layer OAM cells. The PCI SAR is intended primarily for ATM AAL5 encapsulation and termination that is fully supported in hardware. Limited support is provided for AAL3/4 with 48-byte transfers across the PCI-bus interface and hardware recognition of the EOM indicator on the receive side. In addition, a null AAL is supported to facilitate real-time data transfer. The interface to the PHY layer consists of an 8-bit-wide datapath and associated control signals in both the transmit and receive directions. The 53-byte cells pass between the ATM and PHY layers. The native clock for PCI SAR is the PCI-bus clock frequency of 33 MHz. The 8-bit-wide datapath on the receive ATM-PHY interface requires a clock rate of at least 19.44 MHz when interacting with a 155.52-Mbit/s physical layer. The receive interface uses the PHY-layer clock. The native-word size for PCI SAR is 32 bits, corresponding to the data-bus width for the PCI bus.

functional description

The PCI SAR implements the functions of the transmit and receive modules. The implementation of these modules is described in terms of their functional blocks. The PCI SAR has the following basic blocks: PCIMAC, PMIF, LBIN, USR REG, transmit block (XBTP, CA, XALP, XMB FIFO, XPIN), and receive block (RBTP, RALP, RMB FIFO, RPIN) (see Figure 12).

transmit modules

The transmit host and buffer transaction processor (XBTP) is responsible for all host-related functions on the transmit side. It requests 48-byte transfers from the PCI bus-interface block, PCIMAC. The cell actuator (CA) accesses the BWG table and determines the next VC to be serviced. The transmit adaptation-layer processor (XALP) processes all AAL-related functions and adds the four bytes of the ATM header to each cell. The AAL5 cyclic-redundancy check (CRC) is generated by the XBTP module and appended to the packet. The transmit buffer (XMB), a FIFO, is an 8-cell buffer that receives 13 words per cell. Idle cells also are placed in this buffer. The transmit PHY interface (XPIN) does word-to-byte unpacking and interacts with the PHY layer using the PCI-bus clock.

receive modules

The receive PHY interface (RPIN) performs byte-to-word packing, filters idle cells, and interacts with the PHY layer using the system PHY-layer clock crystal. The receive buffer (RMB) performs rate synchronization from the PHY-layer clock to the PCI-bus clock and buffers up to 32 cells. The receive ATM processor (RAT) and the receive ATM adaptation-layer processor (RALP) operate in parallel and are part of the same module. The RALP terminates the AAL5 CRC and processes various EOP indicators. The RAT function is responsible for deleting the ATM header and accessing the correct receive direct-memory access (DMA) entry. Finally, the receive host and buffer transaction processor (RBTP) performs all host-specific functions on the receive side.

PRINCIPLES OF OPERATION

PCIMAC

The PCIMAC block is an interface unit between a PCI-based host system and PMIF block of PCI SAR. It masters the PCI bus in its master mode and allows the host to access the PCI SAR in its slave-mode operation. This block provides the PCI-bus host interface with all the interface signals for master and slave operations. This block implements a 32-bit data buffer to provide a datapath to and from the host. This data buffer has an interface with the XMB FIFO for transmit data and with RMB FIFO for receive data. This module also has a PCI-configuration space implemented as 32-bit registers.

The PMIF has another interface with PMIF (internal to PCI SAR) that provides all the necessary control signals enabling PCI SAR to operate in master mode. The operation in slave mode is controlled by the host system.

PMIF

The PMIF block provides interfaces to LBIN, CMIA, XBTP, RBTP, and USR REG blocks. The LBIN function provides access to the PHY layer and EPROM. The CMIA interface provides a datapath to access control-memory data. The XBTP and RBTP interfaces provide appropriate signals that make the PCI SAR device a PCI-bus master for the transmit or receive function.

The USR REG interface provides status and control data for PCI SAR functions. This block also has a SAR configuration register that is written by the host to enable transmit or receive operation. This block is only a carrier of data and control signals in either its master- or slave-mode operation. It does not initiate any operation except generating PCI-bus requests.

CMIA

The CMIA block provides an interface to the control memory using the local memory bus. It performs memory arbitration for all the functions that access control memory. Each access is a one (32-bit) word access. The priority mechanism to service various functions is in the following order: RALP, XALP, CA, RBTP, XBTP, and PMIF.

PHY layer

The PHY-layer interface is serviced by the XPIN and RPIN modules for either reading data from the XMB FIFO in the transmit direction or writing data to RMB FIFO in the receive direction. Figure 12 depicts the data-flow representation of the PCI SAR functional block diagram.

PRINCIPLES OF OPERATION

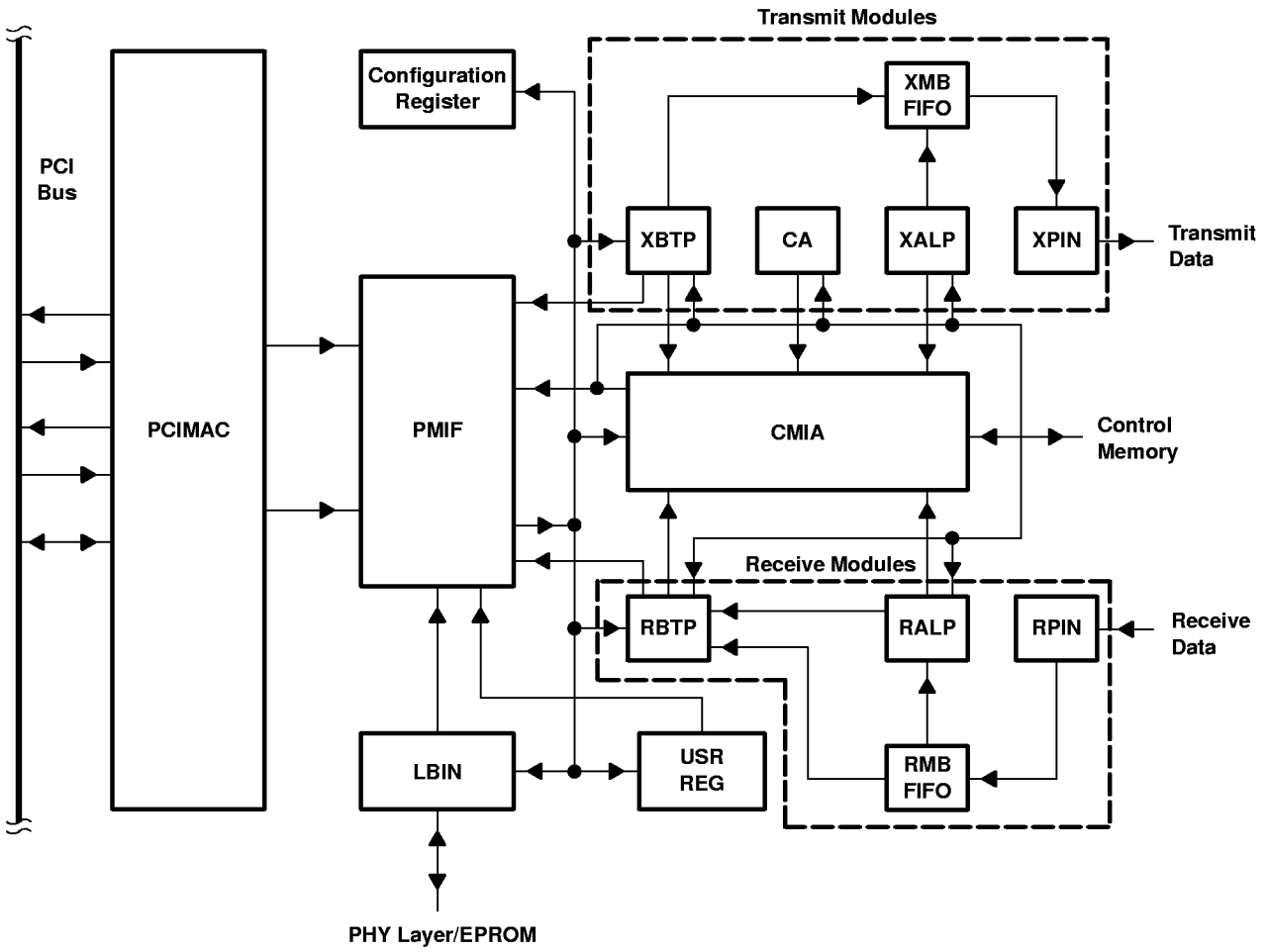


Figure 12. PCI SAR Functional Block Diagram

PRINCIPLES OF OPERATION

interfaces

The PCI SAR (TNETA1561) has the interfaces shown in Figure 13. The PCI-bus interface communicates with any other host on the PCI bus. The local-bus interface allows PCI SAR to have access to PHY-layer device registers and to an external EPROM. The control-memory interface allows the host and other internal functions of the PCI SAR to access the control memory. The PHY-layer interface allows the PCI SAR to transmit packets to a PHY-layer device or receive cells from it.

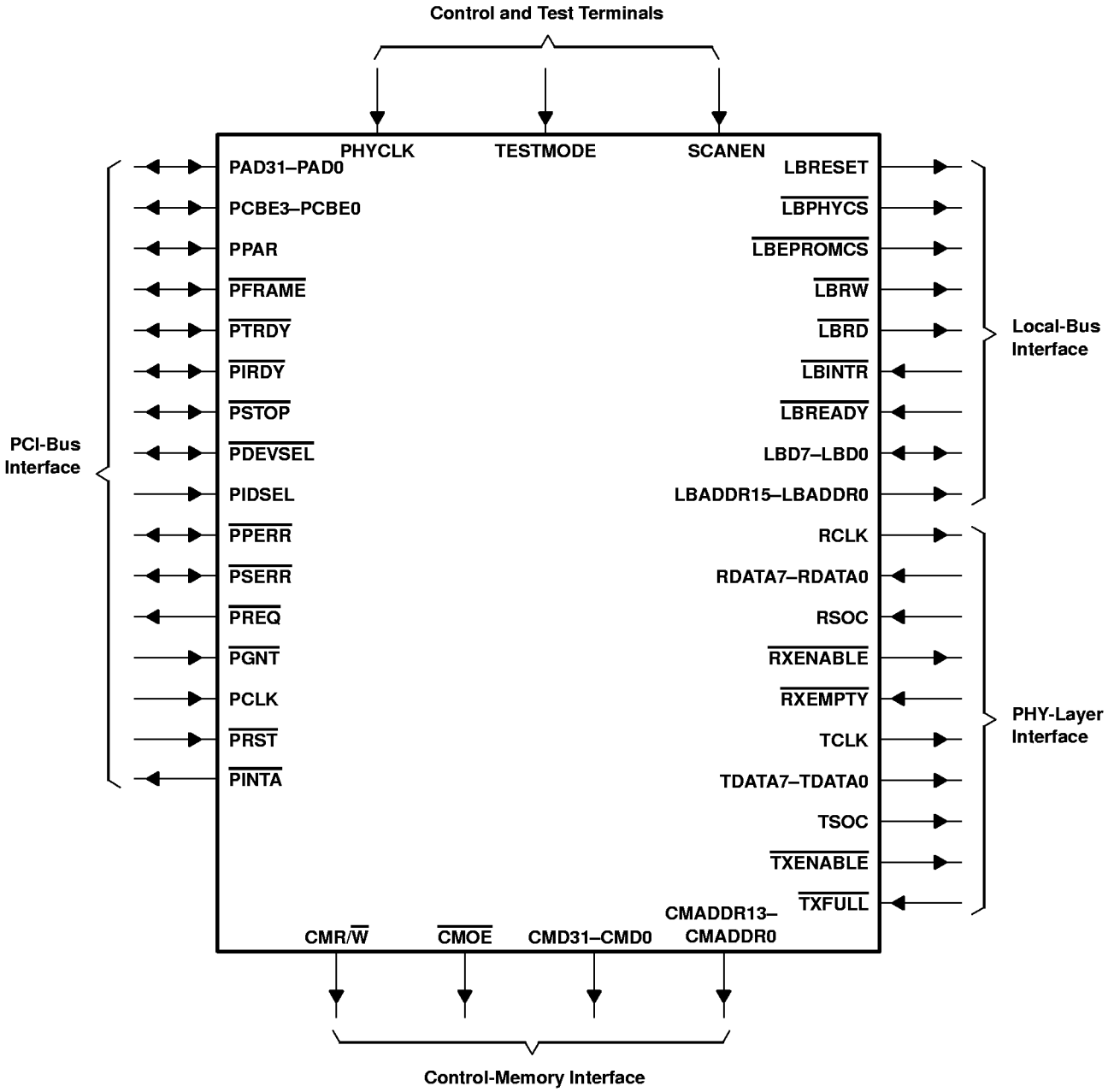


Figure 13. PCI SAR Interface to Other Hosts on the PCI Bus

PRINCIPLES OF OPERATION

PCI-bus interface

The PCI-bus interface is provided by the PCIMAC block of the PCI SAR device. The system terminals are for the PCI-clock and reset function. The address and data terminals are for a 32-bit interface with the least significant byte (LSB) being the bits 7–0 and the most significant byte (MSB) being the bits 31–24. The bus command and byte enable are used to indicate the valid byte of the data. The device fully supports all bus commands (per PCI local-bus specification, rev 2.0, 30 April 1993) except for the interrupt acknowledge, special-cycle command, and I/O commands. In the slave mode, the memory-read multiple and the memory-read lines are treated as the memory-read command. The memory-write and invalidate commands are treated as the memory-write command. In the master mode, it does not support memory-read multiple and memory-read line commands. The device also provides all the interface-control terminals; $\overline{\text{PFRAME}}$ $\overline{\text{PIRDY}}$, $\overline{\text{PTRDY}}$, $\overline{\text{PSTOP}}$, PIDSEL, and $\overline{\text{PDEVSEL}}$. The $\overline{\text{PLOCK}}$ feature is not supported. For bus-master operation, the $\overline{\text{PREQ}}$ and $\overline{\text{PGNT}}$ terminals are provided. The error reporting terminal $\overline{\text{PPERR}}$ is for reporting parity errors (except on a special cycle) and $\overline{\text{PSSER}}$ is for reporting address-and-data parity errors or any other catastrophic system error. The PCIMAC also generates $\overline{\text{PSERR}}$ when it is self selected as the target.

The PCI SAR keeps track of the number of times it has retried a PCI-master transaction. This feature is externally programmable up to a maximum of 15 retries. Once the number of retries exceeds this count value, the TNETA1561 asserts $\overline{\text{PSERR}}$ low. The interrupt $\overline{\text{PINTA}}$ is defined for the PCI SAR. The PCI SAR does not support any JTAG or boundary-scan function. The PCI SAR implements the following functions: the PCI-memory bus master for DMA transfers responds as a PCI slave for local-memory accesses and supports disconnection with retry for PCI. As a PCI-bus master, it supports burst and nonburst data accesses; however, in slave mode it supports only nonburst data transfers. The PCI SAR is designed to meet the worst-case latency of the PCI bus up to 30 μs. A minimum bus-grant value ensures the PCI-bus access for a minimum duration that is long enough to transfer a cell (48 bytes). The PCI macro terminates a transaction when the TNETA1561 is acting as a bus master and no device-select return is detected after it has initiated a transaction.

local-bus interface

The local-bus interface is between the PMIF and LBIN modules. The local bus allows access to the EPROM and the registers on the PHY-layer device. Since several devices are allowed on the local bus, the PCI SAR accepts a ready signal from devices on the bus as a handshake. This accommodates slow devices such as EPROMs and is used to relax timing constraints on the register interface for PHY-layer devices. The local bus is accessed only via PCI-bus transactions with the PCI SAR as the slave (with the exception of the local-bus interrupt signal). The PCI-bus address must remain stable while the local bus is active.

control-memory interface

The control-memory interface is between the control-memory interface and arbitration (CMIA) and all other modules that access the control memory. The control memory is set up in a 16K × 32 configuration with the cycle time given by the PCI-bus clock. The control-memory interface is designed for an asynchronous SRAM with a 32-bit data bus, a 14-bit address bus, a read or write signal, and an output-enable (CMOE) signal.

PHY-layer interface

The ATM cell-transfer rate is full-duplex 149.76 Mbit/s, but data may arrive in bursts at 155.52 Mbit/s due to the framing scheme described by the PHY layer. A clock rate of at least 19.44 MHz is essential in the receive direction to prevent cell loss due to buffer overflow in the PHY layer. The PCI SAR decouples the PCI-bus clock from the PHY-layer clock in the receive direction via an asynchronous FIFO, which holds up to 32 cells. The PCI SAR transmits data to the PHY layer at the PCI-bus clock rate.

The PCI SAR sends a transmit clock at the PCI clock frequency and a receive clock at 19.44 MHz to the PHY layer. The transmit clock sent to the PHY layer is an inverted version of the internal clock. This ensures that all setup- and hold-time restrictions are met. The PCI SAR generates output data along with a start-of-cell indicator in the transmit direction.

PRINCIPLES OF OPERATION

PHY-layer interface (continued)

This data is sent at the rate of the PCI-bus clock. The PHY layer can respond with a full signal, which is asserted at least four cycles before any internal buffers are full. The PCI SAR then turns off the $\overline{\text{TXENABLE}}$ signal until the full signal is deasserted. The PHY layer sends a start-of-cell indicator with output data. The empty signal acts as an inverted enable signal on this interface.

The PHY-layer interrupt signal is directly connected to the PCI-bus interrupt signal; therefore, PCI-bus interrupt is asserted when the PHY-layer interrupt signal is asserted.

operation

The memory mapping of the PCI SAR local-memory elements is mapped in the host-memory space. The host memory-block location, which is determined by the host, is not predefined. The host writes the starting address in the base-address register located in the configuration space. The PCI SAR during read-from or write-to host memory uses the little-endian addressing scheme. This requires byte swapping of data into big endian and writing into the XMB FIFO during the transmit operation. The received data bytes from the RMB FIFO also must be changed from big endian into little endian.

PCI-bus and data-transfer requirements

The PCI SAR behaves as a PCI-bus DMA master and as a slave. The PCI SAR supports a maximum AAL5-buffer size of 64K bytes, which corresponds to a maximum AAL5-packet length of 64K bytes. In burst mode, the data transfer between the PCI SAR and the host is cell based (48 bytes). This transfer is completed in a single access of the PCI bus, but this depends on the bus latency of the host system. This transfer always is initiated by the PCI SAR as a master. The data transfer across the PCI bus is word based (4 bytes). The PCI SAR also supports nonburst transfers as a master and as a slave for host accesses (as defined in the PCI-bus transaction).

PCI-bus interaction and transfer size

TRANSACTION	PCI SAR ROLE	TRANSFER SIZE
Host access – PCI SAR registers, PHY-layer registers and control memory	Slave	Word
Host access – PCI-configuration space	Slave	Byte/word
Host access – EPROM	Slave	One, two, three or four bytes
PCI SAR access – Transmit completion ring, transmit descriptor ring, and receive free-buffer ring transactions	Master	Word
PCI SAR access – Posting to host-receive completion-ring entry	Master	4 words
PCI SAR access – Cell-payload transfers	Master	1–13 words latency dependent

PRINCIPLES OF OPERATION

memory-map table

The following memory-map table defines the offset-address range for the various blocks of the control memory as they are mapped into host memory. The host-memory base address of the control-memory block is obtained from the base register 0. This is defined in the paragraph for the PCI SAR configuration-space registers. The host-memory base address of the EPROM-memory block is obtained from the expansion-ROM base-address register. These base addresses are defined in the PCI-SAR configuration-space register section.

control-memory block – maximum size of 64K bytes

The first 48K bytes of this block are in the control memory (external to the PCI SAR) and are divided into the first 16K bytes for the transmit-side information and the next 32K bytes for the receive-side information. The remaining 16K bytes are divided into 8K bytes each for the USR register (within the PCI SAR) and PHY-layer register (PHY-layer device external to PCI SAR).

OFFSET ADDRESS BITS	DESCRIPTION	READ/WRITE REGISTER
00000000h–000003FFh	Initialization block (256 words)	R/W
00000400h–000023FFh	Transmit DMA states (2K words)	R/W
00002400h–00003FFFh	BWG table (1.2K words)	R/W
00004000h–0000BFFFh	Receive DMA states (8K words)	R/W
0000C000h–0000DFFFh	PHY-layer register (2K words)	R/W
0000E000h–0000FFFFh	USR register (2K words)	R/W

indirect local-memory block – maximum size of 8K bytes

The indirect control-memory block includes the following registers for addressing, data, and status information:

REGISTER	SIZE	DESCRIPTION
Control-memory address register	32 bit	Contains the address of the control-memory block
Control-memory data register	32 bit	Buffer that provides data read from or written to the control-memory block
Control-memory control register 1–register 8	32 bit	Display PCI SAR register information

EPROM memory block

The TNETA1561 can access an EPROM via the local-bus interface. The maximum-size EPROM that can be accessed by the TNETA1561 is 8K bytes. There are four sizes of data transfers from the EPROM (one, two, three, or four byte). To read data from the EPROM, the host generates a 32-bit PCI address and drives the appropriate byte-enable signals according to the type of transfer being executed.

PHY-layer registers access

The TNETA1561 uses the local-bus interface to access the PHY-layer registers. The host system must use the PCI interface to address the register in the PHY layer; therefore, a 32-bit address has to be generated from the host and passed to the TNETA1561. To access a byte-wide address offset for the PHY-layer device registers, the host software uses the PCI-bus byte-enable signals to specify which data byte to access on the PHY-layer device. During a write operation to a PHY-layer register, the host uses byte-enable signals to indicate to the TNETA1561 which data byte contains the valid byte. During a read operation to a PHY-layer register, the TNETA1561 copies the byte read from the PHY-layer four times to form a 32-bit word. The 32-bit word containing the PHY-layer data in all four bytes is transferred to the host.

PRINCIPLES OF OPERATION

control-memory access

The control memory is accessed by using the offset-address bit of the PCI-bus address. This provides a 14-bit-wide address bus to the control memory. All PCI-bus accesses to control memory are one-word accesses at word boundaries.

control-memory address map

The data below specifies the PCI SAR slave-mode PCI-bus physical-address ranges for peripheral devices.

MEMORY REGIONS	CONTROL-MEMORY BASE POINTERS (HEX)
Initialization block	0000h
Transmit BWG 0–255 – DMA block	0100h
BWG table (1200 words, 4800 entries)	0900h
Receive BWG/VCI 0–1023 – DMA block	1000h

PCI-bus physical addresses for PCI SAR peripheral devices

The data below specifies the PCI SAR slave-mode PCI-bus physical-address ranges for peripheral devices.

DESCRIPTION	ADDRESS BITS	READ/WRITE REGISTER
EPROM addresses	14	R
PHY-layer register addresses	14	R/W
Control-memory addresses	14	R/W

packet-interface information

Packet interface, BWG-table mechanism, AAL5 processing, AAL3/4 processing, null-AAL processing, VPI/VCI/GFC processing, OAM processing, and details on the transmit-descriptor rings/DMA, receive free-buffer rings/DMA, and completion rings are described in this section.

The PCI SAR uses host memory to store a packet (48-byte cells) in both transmit and receive directions. The PCI SAR initiates the data transfer for the PCI bus for both transmit and receive operations. The packet does not include AAL5 encapsulation while in host memory. The PCI SAR provides this header data. The buffering of data within the PCI SAR is limited to an 8-cell FIFO for transmit and a 32-cell FIFO for receive.

Each packet queued for transmission can be distributed across multiple buffers in host memory with each starting on a one-byte boundary. Packets that are received over ATM are placed in a single buffer in host memory (either big or small) aligned to a 16-byte boundary.

PRINCIPLES OF OPERATION

bandwidth group (BWG) table mechanism

The PCI SAR generates data via a special bit-rate control table known as the BWG table (see Figure 14). The BWG table consists of up to 4800 entries and each entry consists of an 8-bit BWG index. The table is organized with 1200 words in control memory. The size is programmable via the BWG-size register. Each BWG index corresponds to a transmit DMA channel, and the TNETA1561 can support 255 (8-bit index) transmit DMA channels simultaneously. The PCI SAR cycles through the table and sends an ATM cell for the transmit DMA channel for each entry in the table. If a zero value is entered for a BWG index, an idle cell is transmited. The BWG table assigns the transmit-side bandwidth. The total available bandwidth for an OC-3c SONET link is 135.63 Mbit/s (155.52 Mbit/s less SONET and ATM overhead). The granularity is obtained by dividing the SONET-link bit rate (135.63 Mbit/s) by the entries in the BWG table (4800).

135.63 Mbit/s ÷ 4800 = 28,250 bit/s per entry

An application requiring 500 kbit/s requires 19 entries in the table.

500 kbit/s ÷ 28,250 bit/s per entry = 19 required entries in the table

If the application uses the transmit descriptor ring 5, then there will be 19 entries with 5 as the BWG index (see Figure 14).

5	5	5	5	0
5	5	5	5	1
5	5	5	5	2
5	5	5	5	3
5	5	5	0	4
		•		•
		•		•
		•		•
		•		•
		•		•
		•		•
0	0	0	0	1199

Figure 14. BWG Table

PRINCIPLES OF OPERATION

AAL-type processing

The PCI SAR supports various types of AAL processing. AAL5, AAL3/4, null-AAL, OAM, and GFC processing are described in this section.

AAL5 processing

The primary support is for AAL5 with encapsulation in the transmit direction and termination in the receive direction. AAL5 packets are converted to cells by the PCI SAR before delivery to the PHY layer. Similarly, the device recovers the 53-byte ATM cells from the PHY layer before it performs AAL5 termination.

Since 48 bytes are provided across the PCI-bus interface, all AAL3/4 packet data processing is performed by the host in software. AAL5 processing is disabled on VCIs using AAL3/4. The AAL3/4 EOM indicator, which is located in the first byte of the ATM payload (see Figure 15), is recognized in hardware, initiating an interrupt to the host. This is used by the host to retrieve successive 48-byte payload segments from the appropriate buffer.

AAL3/4 processing

The PCI SAR adds the pad, the control/length field, and the cyclic-redundancy check (CRC) for transmit packets. The PCI SAR does not interpret the field length in the AAL5 frame in the receive direction; therefore, the entire AAL5 packet is forwarded to host memory, allowing the driver to remove the correct payload. This also allows the host to examine the control field in software, necessary in a time of evolving standards in this area. The PCI SAR performs CRC checks in the receive direction and indicates EOP processing to the host, based on the EOP indication in AAL5.

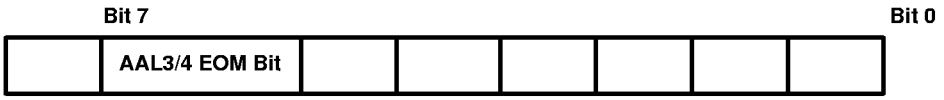


Figure 15. AAL3/4 Processing

null-AAL processing

Null-AAL processing uses the same mechanism as AAL3/4 in the transmit direction to disable AAL5 processing. The control entry associated with each BWG (VCI) in the receive direction has an entry to indicate an interval defined in units of cells received. The PCI SAR then provides an interrupt to the host when the number of cells received on the VCI is equal to that indicated by the table entry. This counter is reset after each interrupt (at the end of each interval). This interval also is referred to as a packet, although it does not encapsulate a well-defined unit of information.

OAM processing

ATM-layer OAM processing does not require real-time intervention and is processed in software. OAM cells received on the link are identified by the PCI SAR.

high-order VPI/VCI bits and GFC processing

The lower ten bits of the VCI are used to encode the 1023 possible VCIs. VCI 0 is not used since it indicates unassigned cells. The upper-order bits of the VCI and the VPI field are programmable on a per-VC basis on transmit. The generic-flow-control (GFC) field always is set to zero.

The upper-order bits of the VCI, the VPI field, and the GFC field are ignored on all cells that are received. These cells are passed only to the PCI SAR if the header-error-control (HEC) field is correct, the upper-order bits of the header are set intentionally, or the cell is misrouted. The probability of misrouting is small and such an event would be detected via the CRC check in AAL5. The advantage of this scheme is that any VPI/VCI combination is supported if the lower ten bits of the VCI are unique.

PRINCIPLES OF OPERATION

ATM-layer OAM encoding

NO.	ITEM	VCI	PTI
1	VP level: link-associated OAM cell	3	–
2	VP level: end-to-end OAM cell	4	–
3	VC level: link-associated OAM cell	Any	4
4	VC level: end-to-end OAM cell	Any	5

Each OAM cell forms a fully encapsulated packet. ATM-layer OAM cells transcend AAL protocols and are recognized differently. The end system recognizes all four ATM-layer OAM flows. OAM cells received on VCI 3 and 4 do not interfere with the normal data stream. The only special processing necessary is to initiate EOP processing for each cell. The software driver must configure VCI 3 and 4 as null-AAL channels with a packet length equal to one cell in the receive direction. OAM cells are transmitted as null-AAL packets with length equal to one cell. VC-level OAM cells are specially interpreted. They are diverted to receive DMA channel 0 and the 4-byte ATM header is passed on to a receive completion ring in host memory during normal EOP processing.

transmit descriptor rings and DMA

Each transmit BWG is supported by a corresponding DMA channel and its own descriptor ring. The PCI SAR supports 255 BWGs, 255 descriptor rings, and 255 DMA channels in the transmit direction. This implies that the number of packets and VCs that are active simultaneously is limited to 255. BWG 0 represents null and a null cell is transmitted. This null cell is generated by the PCI SAR and no data is buffered in the FIFO memory for transmission.

Each descriptor ring holds up to 256 entries corresponding to 256 buffers that can be queued for transmission for that ring. The total number of buffers that can be queued for transmission is approximately 64K (256 buffers per descriptor ring x 255 descriptor rings). The buffers within a descriptor ring are serviced in FIFO order on a per-buffer basis.

Each descriptor-ring entry contains a control bit that indicates whether a buffer is queued up for transmission. The DMA entry for each BWG contains a pointer to the first item in the queue in the corresponding descriptor ring. An idle cell is transmitted if the control bit in the descriptor entry indicates an inactive entry. The DMA entry has a bit that allows the host to disable any BWG.

receive free-buffer rings and DMA

The PCI SAR uses buffer pointers from two free-buffer rings to place the incoming packet data in the host memory. These are called small free-buffer ring and big free-buffer ring. Each receive BWG has a control bit indicating the type of buffer it uses – small or big. These buffers are preallocated by the host application for the next packet and not by the BWG.

The PCI SAR supports 1023 receive DMA channels and 1023 VCIs. The incoming VCI indexes the receive DMA channels. BWG 0 is reserved to process information for OAM cells.

completion rings

The PCI SAR indicates completion of packet processing in either direction to the host via an interrupt and by posting entries to receive and transmit completion rings. Each completion ring accepts up to 256 entries. A control bit in each entry of the completion ring prevents the PCI SAR from overwriting an entry that has not been processed by the host.

PRINCIPLES OF OPERATION

data structure

The PCI SAR data structure and contents of various physical locations are summarized below:

CONTROL MEMORY	HOST MEMORY	INTERNAL REGISTERS
BWG table	TX descriptor rings (255)	PCI SAR operational registers
TX DMA states	TX completion ring	PCI SAR configuration registers
RX DMA states	Small free-buffer ring	PCI configuration space
Initialization block	Big free-buffer ring	
	RX completion ring	
	Data buffers	

The parameters necessary for booting the device are stored in the PCI configuration space. Some systems may use an external EPROM that contains the booting sequence.

The system has a bus width of four bytes and all transactions are conducted on 4-byte boundaries. The PCI SAR uses little-endian addressing as a PCI-bus device. Each descriptor ring has 256 entries and each descriptor-ring entry consists of four words. Each descriptor ring is aligned to a 4K-byte boundary in host memory with each entry aligned to a 16-byte boundary.

The PCI SAR has two receive free-buffer rings, one transmit completion ring, and one receive completion ring. The current pointer to each of these rings is stored in the initialization block in the control memory. An entry in each transmit DMA channel points to one of the 255 transmit descriptor rings in host memory.

Each DMA-channel entry consists of eight words and is located in control memory. The DMA entries on both transmit and receive have an OWN bit that is set when the DMA channel is active. The descriptor-ring entries, the completion-ring entries, and the free-buffer ring entries have an OWN bit that is set when the entry belongs to the PCI SAR.

PRINCIPLES OF OPERATION

initialization block

The initialization block contains exactly four entries and resides in control memory. The following data shows the configuration of the initialization block.

initialization block table

PCI-BUS ADDRESS OFFSET (HEX)	CONTROL MEMORY ADDRESS (HEX)	BITS 31–0
0000	0000	TX completion-ring offset pointer
0004	0001	RX completion-ring offset pointer
0008	0002	Small free-buffer-ring offset pointer
000C	0003	Big free-buffer-ring offset pointer

The PCI-bus address offsets for control memory have the lower-order two address bits always set to zero since accesses to control memory are permitted only on a word basis. In addition, the driver must write the pointers to the data structures in the initialization block as follows:

- The transmit completion-ring offset pointer, small free-buffer ring pointer, and big free-buffer ring pointer are set at 4-byte boundaries. The driver selects a host-memory address and writes it to control memory by shifting it two bits to the right.

  Example: The host address 4EFF0000 (hex) is written to the control-memory initialization block as 13BFC000 (hex).

- The receive completion-ring offset pointer is set at a 16-byte boundary. This pointer is written by shifting the address four bits to the right.

  Example: The host address 4EFF0000 (hex) is written to the control-memory initialization block as 04EFF0000 (hex).

transmit descriptor rings

The TNETA1561 device uses a set of 255 transmit descriptor rings in host memory to manage the 255 transmit DMA channels. Each ring corresponds to a transmit DMA channel. In addition, each ring holds 256 entries, with each entry corresponding to a buffer. At any time, the TNETA1561 supports 256 buffers per packet; however, if the software driver can recycle buffers fast enough, it is possible to support more than 256 buffers per packet. This is accomplished by using four 32-bit word entries per ring entry. These entries contain the control information for each buffer of a packet that is segmented in a given transmit descriptor ring.

The software driver places the buffer information on a transmit descriptor-ring entry for all the buffers being queued for transmission. The TNETA1561 fetches each entry to obtain the segmentation information and to start transmitting ATM cells. The transmit descriptor rings can be assigned to any VPI/VCI combination by entering the proper VPI/VCI information on the descriptor-ring entries.

The TNETA1561 polls the transmit descriptor ring in a circular fashion looking for entries queued for transmission. When the PCI SAR reaches an entry that is not queued for transmission, it stops and sends an idle cell for the given VPI/VCI supported by the transmit descriptor ring. The next time the PCI SAR returns to poll this ring, it starts at the place where the entry was not queued for segmentation the last time the ring was polled. The order and the frequency at which each descriptor ring is polled is determined by the BWG table.

The software driver configures the transmit descriptor rings in host memory at 4-byte boundaries and reserves 4K bytes per ring. Control information for the transmit descriptor rings is contained in the following sections.

PRINCIPLES OF OPERATION

transmit descriptor-ring summary

The data below shows the composition of the four-word entry.

ENTRY	DESCRIPTION
Word 0	Control field, packet length, buffer length
Word 1	Start-of-buffer pointer – 32 bits
Word 2	4-byte ATM header
Word 3	AAL5 tail – control and length fields

TX descriptor-ring word 0 – configuration

Control (bits 31–27)	Current-packet length (bits 26–16)	Current-buffer length (bits 15–0)

OWN (bit 31)

The descriptor is owned by the PCI SAR when the OWN bit is set. The descriptor is owned by the host when the OWN bit is zero. The OWN bit is set by the host when a buffer/packet is queued for transmission. When the next BWG index from the BWG table does not have an active buffer location in the transmit DMA entry, the PCI SAR attempts to recover a new-buffer-descriptor entry from the transmit data-descriptor ring. This entry is loaded into the DMA entry if the OWN bit is set. If the OWN bit for the first descriptor in the transmit data-descriptor ring is zero, no data is queued for transmission and an idle cell is transmitted.

The host places all the buffers for a packet in the descriptor ring before setting the OWN bits on the entries representing each buffer in sequence from the last buffer to the first buffer (in reverse order). The PCI SAR clears the OWN bit after it finishes transmitting/processing the bytes associated with the buffer that is pointed to by the DMA entry. When the OWN bit is cleared by the host, word 0 is not meaningful and is overwritten by the host.

start of chain (SOC) (bit 30)

The SOC bit indicates that this is the first buffer of a packet, which consists of one or more buffers. This bit also is set in packets with single buffers.

end of chain (EOC) (bit 29)

The EOC bit indicates that this is the last buffer of a packet. Single buffer packets have both the SOC and EOC bits set. Packets with multiple buffers have the SOC bit set on the first buffer and the EOC bit set on the last buffer.

interrupt-control bit (ICB) (bit 28)

The ICB bit controls interrupt posting by the PCI SAR to the host after a packet has been transmitted. The setting of ICB to active high disables posting of interrupts to the host by the PCI SAR after a posting of a packet completion is done for this transmit descriptor ring.

AAL-type – AAL5 indicator (bit 27)

The AAL-type bit indicates that the packet/buffer described in this descriptor-ring entry is an AAL5 packet. When zero, this bit indicates to the PCI SAR that AAL5 processing is being performed in the transmit direction. This includes addition of the pad, the control- and packet-length fields, and the 32-bit CRC. The total size of the AAL5 packet is a multiple of 48 bytes. The PCI SAR implements the functions related to packet length and the generation of the pad. The PCI SAR does not perform any packet-level encapsulation similar to that used in AAL5 for either AAL3/4 or the null AAL. The host provides packets correctly formatted into 48-byte cells to the PCI SAR.

PRINCIPLES OF OPERATION

packet length (bits 26–16)

The packet-length field is expressed in units of cells in the packet. The host computes the correct number of cells in the packet, including additional cells that are sometimes needed for AAL5 to accommodate the 8-byte tail. This field represents the value used by the PCI SAR in silicon to determine the number of cells in a packet and enable EOP processing. The field is programmed in two's complement. Incrementing the value by 1 each time a cell is sent results in zero when the entire packet is transmitted. The maximum size of a packet is 64K bytes; therefore,11 bits are adequate to describe the largest packet.

Since this is a packet-level field as opposed to one that applies to individual buffers, it is placed only in the first buffer descriptor of a packet in the transmit data-descriptor rings. The DMA channel updates only the packet-length field on a per-packet basis. The packet-length field is used for all three AAL modes that are supported. In each case, the PCI SAR enables EOP processing to notify the host when the EOP is detected on transmit via the packet-length field.

buffer length (bits 15–0)

The buffer-length field specifies the number of bytes in the buffer represented by this descriptor-ring entry. The maximum buffer size is 64K bytes, which is the largest packet size, and allows an entire packet in one buffer. This field is programmed in two's complement and is equal to zero when all the bytes in a buffer are retrieved by the PCI SAR.

TX descriptor-ring word 1 – start-of-buffer pointer

Byte-aligned start-of-buffer pointer (bits 31–0)

The start-of-buffer pointer is 32 bits. Each buffer is aligned on byte boundaries.

TX descriptor-ring word 2 – ATM header

PTI (bits 31–29)	CLP (bit 28)	VPI (bits 27–20)	VCI (bits 19–4)	PTI (bits 3–1)	CLP (bit 0)

Word 2 contains the 4-byte header for every cell of the packet. The upper-order four bits of the ATM header, representing the GFC at the user-to-network interface (UNI), are set to zero in every outgoing cell. Bits (3–0) in word 2 represent the payload-type indicator (PTI) and cell-loss priority (CLP) fields used in every cell of the packet except the last one (the cell that contains the EOP indication). Bits 31–28 in word 2 represent the PTI and CLP fields used in the last cell of the packet.

The PTI field in the last cell of the AAL5 packet is set either to 001 or 011. The CLP is programmable and the cell containing the EOP indication can have a different priority level from the other cells. This field is required only in the first descriptor for the packet. In AAL3/4 or null-AAL packets, the PTI and CLP fields are the same in both the upper- and lower-order bits of word 2.

BIT	PLACE IN ATM HEADER
3–0	Least-significant four bits of byte 4 of 5-byte ATM header
7–4	Most-significant four bits of byte 4 of 5-byte ATM header
15–8	Byte 3 of 5-byte ATM header
19–16	Least-significant four bits of byte 2 of 5-byte ATM header
23–20	Most-significant four bits of byte 2 of 5-byte ATM header
27–24	Least-significant four bits of byte 1 of 5-byte ATM header

PRINCIPLES OF OPERATION

TX descriptor-ring word 3 – AAL5 control/length

AAL5 control field (bits 31–16)	AAL5 length field (bits 15–0)

The AAL5 control and length fields apply to packets, not to buffers, and this entry is required only in the first descriptor for the packet. The AAL5 length field is not used to determine the length of the packet during transmit processing. Both fields are placed in the descriptor ring in an AAL5 packet in the proper position (in the four bytes preceding the AAL5 32-bit CRC). These fields are not used if the packet is in AAL3/4 or a null-AAL packet.

transmit BWG DMA block

The control memory on the PCI SAR contains 255 transmit BWG DMA entries, each containing eight words. The contents of each entry are summarized in the following table.

transmit BWG DMA entry table

ENTRY	DESCRIPTION	STATIC/DYNAMIC
Word 0	Control field, packet length, buffer length	Dynamic
Word 1	Current-buffer pointer – 32 bits	Dynamic
Word 2	4-byte ATM header	Dynamic
Word 3	Static bits – BWG ON/OFF (BWG_ON bit)	Static
Word 4	BWG data-ring pointer, descriptor pointer	Dynamic
Word 5	Reserved	Dynamic
Word 6	Partial 32-bit packet CRC	Dynamic
Word 7	AAL5 tail – control and length fields	Static

The PCI SAR initiates all transactions affecting the DMA table during normal operation based on cell-transmission opportunities from the BWG table. During initialization, the host has to configure word 0, word 3, and word 4 (shown in the transmit BWG DMA entry table) for each BWG selected for transmission in the BWG table including the BWG0. These words allow the TNETA1561 to start transmission of a new packet. After configuration, the TNETA1561 reads word 3 to check if the BWG_ON bit is set. If it is set, the device reads word 0 to determine if the OWN bit is set. When the OWN bit is not set, it indicates that this is the first buffer of a new packet. The TNETA1561 then reads word 4 to obtain a transmit descriptor-ring pointer that indicates the memory address in host memory for the transmit descriptor-ring pointer. The following sections explain each TX DMA table word in detail.

TX DMA word 0 – state/configuration

Control (bits 31–27)	Current-packet length (bits 26–16)	Current-buffer length (bits 15–0)

The contents of word 0 are copied directly from the corresponding transmit data-descriptor-ring entry at the start of each new buffer. This applies to all the fields in this status word, and the host must ensure consistency across the fields.

OWN (bit 31)

The OWN bit is set when the DMA channel for the BWG is active and all related state information in the DMA entry is current. The OWN bit indicates a packet is currently being segmented and transmitted for this BWG. This OWN bit is cleared by the PCI SAR after the entire packet is transmitted, a completion-ring entry is posted, and an interrupt is generated to the host.

The host sets the OWN bits for individual buffers in a packet in the transmit data-descriptor rings, in order, from last to first. This ensures that the DMA block is not held up while waiting to acquire the next buffer from a partially transmitted packet.

PRINCIPLES OF OPERATION

start of chain (SOC) (bit 30)

The SOC bit indicates that this is the first buffer of a packet which consists of one or more buffers. The SOC bit also is set in packets with single buffers. The SOC bit is cleared by the PCI SAR after all processing for the first buffer is complete.

end of chain (EOC) (bit 29)

The EOC bit indicates that this is the last buffer of a packet. Every packet has at least one buffer with the EOC bit set.

AAL-type – AAL5 indicator (bit 27)

The AAL-type bit is set to zero to indicate that the packet described in this descriptor-ring entry is an AAL5 packet. This bit is a configuration item rather than a bit carrying state information. This bit is set in every buffer of a packet and the software driver must ensure that all buffers in a packet use the same AAL type.

current-packet length (bits 26–16)

The PCI SAR increments this two's-complement value with every cell transmitted until the counter is equal to zero, which indicates to the PCI SAR that the entire packet has been transmitted.

current-buffer length (bits 15–0)

The buffer-length field specifies the number of remaining bytes in the buffer currently being processed in this BWG. The PCI SAR adds to the value of this two's-complement field with every transfer of payload data to the XMB until it is equal to zero, which indicates to the PCI SAR that all the bytes in this buffer are processed and queued for transmission.

TX DMA word 1 – current-buffer pointer

Byte-aligned current-buffer pointer (bits 31–0)

The current-buffer pointer is copied directly from the start-of-buffer pointer in the corresponding transmit data-descriptor-ring entry at the start of each new buffer. The field is 32 bits, which implies that the buffer is aligned to a byte boundary. The pointer is adjusted to point to the current location after each transfer of payload data from the host to the XMB.

TX DMA word 2 – ATM header

PTI (bits 31–29)	CLP (bit 28)	VPI (bits 27–20)	VCI (bits 19–4)	PTI (bits 3–1)	CLP (bit 0)

The 4-byte ATM header field is copied directly from the corresponding transmit data-descriptor entry at the start of each new packet. Bits 28–0 are concatenated to the 4-bit GFC field that is set to zero for every cell in the packet except the last one. Bits 31–28 provide the PTI and CLP fields in the last cell of each packet.

BIT	PLACE IN ATM HEADER
3–0	Least-significant four bits of byte 4 of 5-byte ATM header
7–4	Most-significant four bits of byte 4 of 5-byte ATM header
15–8	Byte 3 of 5-byte ATM header
19–16	Least-significant four bits of byte 2 of 5-byte ATM header
23–20	Most-significant four bits of byte 2 of 5-byte ATM header
27–24	Least-significant four bits of byte 1 of 5-byte ATM header

PRINCIPLES OF OPERATION

TX DMA word 3 – configuration

BWG_ON (bit 31)	Unused (bits 30–0)

This bit allows the host to enable data transmission on a per-BWG basis. The BWG_ON bit from the current BWG index is examined by the PCI SAR on each cell opportunity. BWG_ON (31) is directly set by the host to indicate that the BWG is enabled and that normal data processing is followed. If the bit is zero, no processing of transmit data on the BWG is performed and an idle cell is transmitted on the link. This idle cell is used by the host to respond to congestion indicators.

TX DMA word 4 – descriptor-ring address

TX-data descriptor-ring pointer (bits 31–12)	TX descriptor-ring entry (bits 11–4)	0000 (bits 3–0)

This pointer is a DMA address to the location of the current entry (there are 256 entries in each ring) in the corresponding transmit data-descriptor ring (one of 255 rings) for this BWG. Each descriptor ring is aligned to a 4K-byte boundary in host memory, with each entry aligned to a 16-byte boundary.

The address of the 4K-byte boundary in host memory is provided by bits 31–12. The entry number between 0 and 255 is provided by bits 11–4. The low-order four bits are set to zero and each entry is 16-byte aligned. Bits 11–0 are initialized by the host to zero to correspond with the first entry used by the host in the transmit data-descriptor ring.

TX DMA word 5 – reserved

Reserved

TX DMA word 6 – transmit CRC

Partial AAL5 transmit CRC (bits 31–0)

This field stores the 32-bit CRC calculated over the entire payload of each AAL5 packet. The CRC is placed in the last four bytes of the last cell of the corresponding packet.

TX DMA word 7 – AAL5 tail

AAL5 control field (bits 31–16)	AAL5 length field (bits 15–0)

The AAL5 control and length fields are copied directly from the corresponding transmit data-descriptor entry at the start of each new packet. The length field is not used for any control functions within the PCI SAR. Both fields are used exclusively for placement in the tail of an AAL5-protocol data unit (PDU).

transmit completion ring

This entry contains only one word. The transmit completion ring is a descriptor ring with 256 entries. The PCI SAR posts an item to the next entry in the completion ring when it completes the transmission of each packet. The transmit-completion-ring pointer maintains the value of the current entry within the PCI SAR. The host can recalibrate to this by reading the value from the initialization block in control memory.

transmit-completion-ring summary

ENTRY	DESCRIPTION		
Word 0	OWN (bit 31)	Unused (bits 30–8)	BWG index (bits 7–0)

PRINCIPLES OF OPERATION

TX completion-ring word 0

OWN (bit 31)

This completion-ring entry is owned by the PCI SAR when the OWN bit is set. The completion-ring entry is owned by the host when the OWN bit is zero. The PCI SAR uses the next completion-ring entry in the ring if the OWN bit is set. The TNETA1561 clears the OWN bit after updating the entry. The host then receives an interrupt and retrieves the next entry in the completion ring to post the completion of packet transmission for a BWG and the release of the buffer space occupied by the buffers constituting the packet. The host then sets the OWN bit to allow the PCI SAR to use the completion-ring entry when it has queued a packet for transmission. If the OWN bit is not set when the PCI SAR is ready to post a completed packet, a status bit is set in the hardware-status register and an interrupt is generated if the error condition is unmasked.

BWG index (bits 7–0)

The only item that is posted to the transmit completion ring when the PCI SAR completes transmission of a packet is the BWG index. This is adequate for the host to locate the transmit-buffer pointers to the buffer locations where data for the packet was stored and reclaim the buffer space.

receive free-buffer-ring format

There are two free-buffer rings. A receive free-buffer-ring entry consists of one word. Each of the two rings has 256 entries. The host places free-buffer pointers in the entries of each ring. The PCI SAR removes a pointer when it starts processing each new packet from the link.

receive free-buffer-ring summary

ENTRY	DESCRIPTION		
Word 0	OWN (bit 31)	Unused (bits 30–28)	Start-of-buffer pointer (bits 27–0)

RX free-buffer-ring word 0

OWN (bit 31)

Each free-buffer-ring entry is owned by the PCI SAR when the OWN bit is set and it is owned by the host when the OWN bit is zero. The host sets the OWN bit for new entries placed in the free-buffer rings. The PCI SAR uses the next free-buffer-ring entry in the respective ring if the OWN bit is set. The PCI SAR clears the OWN bit after acquiring the buffer and releasing the ring location to the host. The buffer is not freed until a packet is posted to the receive completion ring. If the OWN bit is not set when the PCI SAR polls a free-buffer ring for a new entry, a status bit is set in the hardware-status register and an interrupt to host is generated if the error condition is unmasked.

start-of-buffer pointer (bits 27–0)

A pointer to a buffer, aligned to a 16-byte boundary, is the only information placed in each free-buffer ring.

receive DMA block

The PCI SAR supports 1024 receive DMA-channel entries, with each containing eight words. Each DMA channel represents a VCI on which data is received, and DMA entries in the control memory are indexed by incoming VCIs. The PCI SAR initiates all transactions affecting the DMA table, except those required for one-time configuration of a channel in word 3, during normal operation, based on the header of cells received from the link.

Data with the PTI field equal to 10X, representing VC-level OAM cells, is diverted to DMA channel 0 that operates in the null-AAL mode with a packet length of one cell. Word 0 in each receive DMA-channel entry is copied from word 3 at the start of each new packet. A number of the fields in word 0 represent the dynamic state of the reassembly process for a cell. The fields in word 3 represent one-time configuration values for the VC entered by the host. PCI SAR accesses word 0 during normal cell-level processing to retrieve configuration items.

PRINCIPLES OF OPERATION

receive DMA virtual-channel entry summary

ENTRY	DESCRIPTION	STATIC/ DYNAMIC
Word 0	Control, status, EFCN cell count, current packet length	Dynamic
Word 1	Current-buffer pointer – 28 bits	Dynamic
Word 2	Start-of-buffer pointer – 28 bits	Static
Word 3	Control, packet length	Static
Word 4	Reserved	
Word 5	AAL5 partial CRC – 32 bits	Dynamic
Word 6	Reserved	
Word 7	Reserved	

RX DMA word 0 – VC status/configuration

Control (bits 31–23)	Unused (bit 22)	Current-congestion number (bits 21–11)	Current-packet length (bits 10–0)

OWN (bit 31)

The OWN bit is set when the DMA channel for this BWG is active and all DMA parameters such as the receive-data pointer, buffer length, and packet length are current. The OWN bit is set by the PCI SAR when word 3 is copied to word 0 at the start of each new packet. The bit is cleared by the PCI SAR when the entire packet has been posted to a buffer in host memory. The BWG is inactive when the OWN bit is zero. Then, the free-buffer ring indicated in word 3 is used to poll a new buffer on the arrival of the first cell of a new packet on the VCI used to index this BWG.

static-configuration bits from word 3

The next summary lists five static-configuration bits copied from word 3 at the start of each packet. Each is described in detail in the section on RX DMA word 3.

RX DMA word 0 static-configuration bit summary

LOCATION	FIELD
Bit 31	OWN
Bit 30	VCON
Bit 29	Buffer type: small or big
Bit 28	Null-AAL indication
Bit 25	AAL3/4 indication
Bit 24	End-of-packet wait
Bit 23	Enable end-of-packet wait

explicit-forward congestion-notification (EFCN) cell counter (bits 21–11)

The number of cells received with the EFCN indicator set in each packet is counted and the value is stored in this field. The EFCN indication is given a logic value of 01x in the PTI field of the ATM header. This value is passed to the receive completion ring at the end of each packet. Since this field is copied from word 3 at the start of each new packet, it is reset to zero at this time.

PRINCIPLES OF OPERATION

packet length (bits 10–0)

The packet-length field in word 0 is set up with the two's-complement value for the buffer size in cells used by this BWG at the start of each new packet. The counter is incremented with each new cell until the EOP signal or the value is zero. Null-AAL packets are terminated when the value of this counter reaches zero. If either the AAL5 or AAL3/4 packet fills the buffer to capacity, the counter reaches zero and the packet is terminated, with the buffer-overflow indicator set in the receive-completion-ring entry.

RX DMA word 1 – current-buffer pointer

Unused (bits 31–28)	Current-buffer pointer – 16 byte aligned (bits 27–0)

The current-buffer pointer is 28 bits, which implies that the buffer is aligned to 16-byte boundaries. This is a dynamic field that is updated with every RCB-to-PCI-bus transaction.

RX DMA word 2 – start-of-buffer pointer

Unused (bits 31–28)	Start-of-buffer pointer – 16 byte aligned (bits 27–0)

The start-of-buffer pointer is 28 bits because the buffer is aligned to 16-byte boundaries. This field is copied from the corresponding 28-bit field in word 0 of a free-buffer-ring entry.

RX DMA word 3 – configuration

Configuration (bits 31–23)	Unused (bits 22–11)	Null-AAL packet length (bits 10–0)

OWN bit position (bit 31)

The OWN bit is set high for each valid receive channel. It is copied into the corresponding OWN bit location in word 0 at the start of each new packet to indicate that the DMA channel is active. This OWN bit is automatically reset to a 0 after the end-of-packet indicator is received.

VC_ON (bit 30)

The VC_ON bit enables packet-reassembly processing. The bit is set in the default mode to indicate that the VC is enabled. The PCI SAR discards cells received on the corresponding VC when the VC_ON bit is deasserted on a per-cell basis.

buffer type – small or big (bit 29)

The PCI SAR supports only two buffer sizes on receive – small and big. The host determines the sizes of the small and big buffers. The buffer-type bit is used to select between a buffer pointer from the small free-buffer ring or the big free-buffer ring for each new packet, which allows the host to target small or big buffers for all packets on a given VC. The small free-buffer ring is used when the bit is set and the big free-buffer ring is used in the default (zero) state.

null-AAL indication (bit 28)

This field is set to indicate that null-AAL packets are received on this BWG (VC). The null-AAL packet-length field in bits (10–0) is used to determine the end of a packet. CRC errors are ignored for null-AAL packets. The CRC-error indicator in the receive completion ring is not used.

AAL3/4 indication (bit 25)

This field is set to indicate that AAL3/4 packets are received on this BWG (VC). This indicates the EOM field in byte 6 (bit 6 of an ATM cell is used as the EOP indicator). CRC errors are ignored for AAL3/4 packets. The CRC-error indicator in the receive completion ring is not used.

PRINCIPLES OF OPERATION

end-of-packet wait (bit 24)

This bit must be set to zero by the device driver during initialization. This gives the SAR the responsibility of setting it to one in DMA word 0 (when this feature is enabled). This bit is a status bit used by the TNETA1561 during operation.

enable end-of-packet wait (bit 23)

When a start of packet is detected by the TNETA1561, the TNETA1561 requests a buffer from the host memory. If the buffer is not available, the first cell of this packet is dropped. The rest of the packet is dropped after it is received. The host can set bit 23 to one to enable the TNETA1561 to drop the cells of a packet that had the first cell dropped. Once the TNETA1561 detects the end packet, it begins to receive packets in this VCI. This feature only works for AAL5 and ALL3/4. For null-AAL and OAM cells, bit 23 must be set to zero.

EFCN cell-counter place holder (bits 21–11)

This field is set to zero since it is a place holder for the EFCN cell counter in word 0 of this DMA block.

AAL packet length (bits 10–0)

The AAL packet-length field in word 3 indicates the length of the buffer in cells for each packet in this BWG. This is used in different ways, based on whether the BWG supports AAL5 or AAL3/4 packets or null-AAL packets. This field indicates the length of the buffer size allocated by entries in the free-buffer ring used by this BWG for AAL5 or AAL3/4 packets. This is used to detect buffer overflow.

When the null-AAL indicator is set, this field, programmed in two's-complement notation, represents the number of cells in each null-AL packet. Since receive DMA channel 0 operates off the null-AAL mode, with each packet size equal to one cell, this field is programmed with the value of one in two's-complement notation (7FFhex).

RX DMA word 5 – AAL5 partial CRC

Partial AAL5 receive CRC (bits 31–0)

This field stores the 32-bit CRC that is calculated over the entire payload of each received AAL5 packet. The CRC is stored in the last four bytes of the last cell in the AAL5 frame. The CRC check results in a unique polynomial.

receive completion ring

The following table shows the composition of a 4-word receive-completion-ring entry. The receive completion ring has 256 entries. The PCI SAR posts an item to the next entry in the completion ring when it completes reassembly on a packet. The receive-completion-ring pointer maintains the value of the current entry within the PCI SAR. The host can recalibrate to this by reading the value from the initialization section in control memory.

receive-completion-ring summary

ENTRY	DESCRIPTION
Word 0	Reserved
Word 1	Start-of-buffer pointer – 28 bits
Word 2	4-byte ATM header
Word 3	Control field, EFCN cells received, packet length

RX completion-ring word 0 – reserved

This word is not used or defined.

PRINCIPLES OF OPERATION

RX completion-ring word 1 – start-of-buffer pointer

Unused (bits 31–28)	Start-of-buffer pointer – 16 byte aligned (bits 27–0)

The 28-bit start-of-buffer pointer is provided to the host in the RX completion ring to enable it to locate the reassembled packet.

RX completion-ring word 2 – ATM header

ATM header byte 1	ATM header byte 2	ATM header byte 3	ATM header byte 4

The 4-byte header from the last cell in the reassembled packet is passed to the host.

RX completion-ring word 3 – control

Control (bits 31–29)	Unused (bits 28–22)	Congestion cells received (bits 21–11)	Packet length (bits 10–0)

OWN (bit 31)

This completion-ring entry is owned by the PCI SAR when the OWN bit is set and it is owned by the host when the OWN bit is zero. If the OWN bit of the next entry in the respective receive completion ring is zero when the PCI SAR polls it to post the completion-of-packet processing, an error indicator in the status register is set and an interrupt is generated. This causes the buffer that the PCI SAR attempted to post to be lost. The PCI SAR clears the OWN bit in the receive completion ring after it posts the packet. The host then owns the entry and can retrieve various pointers to the packet.

packet overflow (bit 30)

The packet-overflow bit is set if the receive buffer overflowed while processing the current packet. Every packet that ends in a buffer overflow is terminated immediately and a completion-ring entry is posted to the host.

CRC condition (bit 29)

The PCI SAR forwards AAL5 packets with a CRC error to the host. This bit is set when a packet is received with an AAL CRC error.

congestion cells received (bits 21–11)

The number of cells received in the packet with the EFCN indication set is forwarded to the host to implement associated feedback mechanisms to squelch the source.

packet length (bits 10–0)

All received data is passed to the host in units of 48 bytes. The packet length in 48-byte payload units from word 0 of the receive DMA block is passed to the host in two's-complement notation. This value always is zero for null-AAL packets. The length of an AAL5 or AAL3/4 packet in integer units is obtained by subtracting this value from the reassembly-buffer length reserved for the packet.

PRINCIPLES OF OPERATION

registers

The PCI SAR has defined two types of registers: the PCI configuration-space registers and control and status registers. The PCI-SAR internal registers have a PCI bus physical-address base value read from BASE REG 0 of the PCI configuration space. This section describes several host-accessible internal PCI-SAR registers. Host-write accesses to nonexistent registers are ignored. A null word (32 zeros) is returned to the host on a read access from a nonexistent register.

- PCI SAR configuration-space registers: These registers are initialized by the system-initialization procedure (BIOS device-initialization routine) to program the operation of the PCI SAR device with a PCI-bus interface.
- PCI SAR control and status registers: These registers provide PCI SAR device status and control information.

TNETA1561 configuration-space registers

The TNETA1561 supports the 64-byte header that is defined by the PCI specification revision 2.0. None of the device-specific registers in locations 64–255 are used. The predefined header region has a size of 64 bytes. The layout of the PCI configuration-space registers is shown below.

ADDRESS	BYTE 3	BYTE 2	BYTE 1	BYTE 0	READ/WRITE
0x00	Device ID		Vendor ID		R
0x04	Status		Command		R/W
0x08	Class code			Revision ID	R
0x0C	BIST	Header type	Latency timer	Cache line size†	R/W
0x10	Base address 0				R/W
0x14	Base address 1†				R/W
0x18	Base address 2†				R/W
0x1C	Base address 3†				R/W
0x20	Base address 4†				R/W
0x24	Base address 5†				R/W
0x28	Reserved (returns 0 when read)				
0x2C	Reserved (returns 0 when read)				
0x30	Expansion ROM base address				R/W
0x34	Reserved (returns 0 when read)				
0x38	Reserved (returns 0 when read)				
0x3C	Maximum latency	Minimum grant	Interrupt pin	Interrupt line	R/W
0x40	Reserved (returns 0 when read)				
0x44–0xFF	Reserved (returns 0 when read)				R

† Registers not implemented and return 0

The PCI configuration-space registers are accessible only by PCI-configuration cycles. All multibyte numeric fields follow little-endian byte format.

PRINCIPLES OF OPERATION

vendor-ID register (offset address 00h)

The vendor-ID register is a 16-bit register that identifies the manufacturer of the TNETA1561. The Texas Instruments (TI) vendor ID is 104Ch. The vendor ID is assigned by the PCI special-interest group. The vendor-ID register is located at offset address 00h in the PCI configuration space and is read only.

device-ID register (offset address 02h)

The device-ID register is a 16-bit register that uniquely identifies the TNETA1561 device within TI's product line. The device ID assigned by TI is A100h. The device-ID register is located at offset address 02h in the PCI configuration space and is read only.

command register (offset address 04h)

The command register provides device control over its ability to generate and respond to PCI cycles. When a zero is written to this register, the device is logically disconnected from the PCI bus for all accesses except configuration accesses. On power up and reset, the host places all zeros in this register. However, for the normal operation of the PCI SAR, the host enables the bus master (bit 2) and memory space (bit1) of the command register. The individual bits in the register and the default values are described below.

BIT	NAME	DESCRIPTION	RESET VALUE
0	I/O space	Control bit 0 controls the response of the TNETA1561 to I/O space accesses. A value of 0 disables the response of TNETA1561. A value of 1 allows TNETA1561 to respond to I/O space accesses.	0
1	Memory space	Control bit 1 controls the response of the TNETA1561 to memory space accesses. A value of 0 disables the response of TNETA1561. A value of 1 allows TNETA1561 to respond to memory space accesses.	0
2	Bus master	Control bit 2 controls the ability of the TNETA1561 to act as a master on the PCI bus. A value of 0 disables TNETA1561 from generating PCI accesses. A value of 1 allows TNETA1561 to act as a bus master.	0
3	Special cycles	Control bit 3 controls the action of the TNETA1561 on special cycle operation. A value of 0 causes TNETA1561 to ignore all special cycle operations. A value of 1 allows the TNETA1561 to monitor special cycle operations.	0
4	Memory write and invalidate enable	Control bit 4 is an enable bit for using memory write and invalidate command. When this bit is 1, the master can generate the command. When the bit is 0, memory write must be used instead.	0
5	VGA palette snoop	Control bit 5 controls how VGA-compatible and graphics devices handle accesses to VGA-palette registers.	0
6	Parity error response	Control bit 6 controls the response of the TNETA1561 to parity errors when the bit is set. The device must take its normal action when a parity error is detected. When the bit is 0, the device must ignore any parity errors that it detects and continue normal operation.	0
7	Wait cycle control	Control bit 7 controls whether or not a device does address data stepping.	0
8	$\overline{\text{PSERR}}$ enable	Control bit 8 is an enable bit for the $\overline{\text{PSERR}}$ driver. A value of 0 disables $\overline{\text{PSERR}}$ driver. After $\overline{\text{PRST}}$, control bit 8 goes to 0.	0
9	Fast back-to-back enable	Control bit 9 controls whether or not a master can do back-to-back transactions to different devices. Initialization software sets the bits if all targets are fast back-to-back capable.	0
10–15	Reserved	Reserved	0

PRINCIPLES OF OPERATION

status register (offset address 06h)

The status register is a 16-bit register that contains status information for the PCI-bus related events. The status register is located at offset address 06h in the PCI configuration space. Reads to this register behave normally. Writes are slightly different in that bits can be reset but not set. A bit is reset when the register is written and the data in the corresponding location is a one. For example, to clear bit 14 and not affect any other bits, write the value 0100_0000_0000_0000b. On power up and reset, this register is set to 0000001000000000. This register is set by the host and is not programmable. The individual bits in the status register are described in the following table.

BIT	NAME	DESCRIPTION	RESET VALUE
0–6	Reserved	Status bits 0–6 always return to 0.	0
7	Fast back-to-back enable	Status bit 7 indicates that the device is capable of performing fast back-to-back transitions.	0
8	Data parity reported	Status bit 8 is set only by master devices. It is set when three conditions are met: 1. The bus agent asserted $\overline{\text{PERR}}$ itself or observed $\overline{\text{PERR}}$ asserted. 2. The agent setting the bit acted as the bus master for the operation in which the error occurred. 3. The parity error response in the command register is set.	0
9–10	$\overline{\text{PDEVSEL}}$ timing	Status bits 9–10 encode the timing of $\overline{\text{PDEVSEL}}$. There are three allowable timings for $\overline{\text{PDEVSEL}}$. They are 00b for fast, 01b for medium, and 10b for slow.	01
11	Signaled target abort	Status bit 11 is set by the target device when it terminates a transaction with target abort.	0
12	Received target abort	Status bit 12 is set by the master device when its transaction is terminated with target abort.	0
13	Initiated master abort	Status bit 13 must be set by a master device when its transaction is terminated with master abort.	0
14	Signaled-system error	Status bit 14 must be set when the system asserts $\overline{\text{PSERR}}$.	0
15	Detected-parity error	Status bit 15 is set by the device when it detects a parity error even if the parity handling is disabled by the parity-error response bit in the command register.	0

revision-ID register (offset address 08h)

The revision-ID register is an 8-bit register that specifies a device-specific revision-identifier number. The current value of the register is 00h. The revision ID register is located at offset address 08h in the PCI configuration space and is read only.

class-code register (offset address 09h)

The class-code register is a 24-bit register that specifies the generic function of the device. The class code register is located at offset address 09h in the PCI configuration space and is read only. The current value is 028000h for a network controller.

cache line-size register (offset address 0Ch)

The PCIMAC supports write and invalidate as a master. The host writes the cache line size into this byte-wide register.

latency-timer register (offset address 0Dh)

The latency-timer register is an 8-bit register that specifies the maximum time TNETA1561 device can continue with bus-master transfers. The PCIMAC supports a burst of more than one data cycle. The host sets the latency requirements of the system in this register in PCI-bus clock units. When the current-time value (00h) stored in the latency-timer register expires, the TNETA1561 immediately releases the bus after finishing the current data phase.

PRINCIPLES OF OPERATION

header-type register (offset address 0Eh)

The header-type register is an 8-bit register that describes the format of the PCI configuration-space locations 10h to 3Ch. The header defined here is referred to as type 0. The header-type register is located at offset address 0Eh in the PCI configuration space and is read only. The current value is 001h.

built-in self-test register (BIST) (offset address 0Fh)

BIST is not supported by the PCI SAR. Reading this register returns 0.

base-address register 0 (offset address 10h)

Power-up software needs to build a consistent address map before booting the machine to the operating system. This means that questions like how much memory is in the system and how much address space the I/O controllers in the system require must be answered. After determining this information, power-up software can map the I/O controllers into reasonable locations and proceed with the system boot. To do this mapping in a device-independent manner, the base registers for this mapping are placed in a predefined header portion of configuration space.

The base-address register is updated by the host device during power up. The number of upper bits that a device implements depends on how much address space is responded to by the device. A device that uses a 64K-byte address space builds the top 16 bits and hardwires the lower 16 bits. The power-up software determines the address space required by writing a value of all ones to this register and reading the value back. The device returns zeros all in do-not-care bits. In the case of TNETA1561, it returns FFFF0000h to specify that it requires an address space of 64K bytes. The individual bits in the register are described in the following table.

BIT	NAME	DESCRIPTION	RESET VALUE
0	Memory-space indicator	Bit 0 is used to determine whether the register maps into memory or into I/O space. The base-address register that maps to memory space must return a 0. The base-address register that maps to I/O space must return a 1. Since TNETA1561 is a memory-mapped device, bit 0 defaults to 0.	0
1–2	Type	Bits 1–2 have the following encoded meaning: 00 Base register is 32 bits wide and mapping can be done anywhere in the 32-bit memory space. 01 Base register is 32 bits wide but must be mapped below 1 Mbyte. 10 Base register is 64 bits wide and can be mapped anywhere in the 64-bit address space. 11 Reserved	00
3	Prefetchable	Bit 3 is set to 1 if there are no side effects on reads. The device returns all bytes on reads regardless of the byte enables, and the host bridges can merge processor writes into this range without causing errors. Bit 3 must be set to 0 for all other conditions.	0
4–31	Base address	Bits 4–31 are set by the host device during power up.	

PRINCIPLES OF OPERATION

expansion-ROM base-address register (offset address 30h)

Some PCI devices, especially those that are defined for use on add-in modules in PC architectures, require local EPROMs for expansion ROM. The base address and the size information of the expansion ROM is handled by the expansion-ROM base-address register.

On power up and reset, the bits of this register are set to 0. However, if the host needs to access the EPROM, bit 0 is set to 1. The individual bits in the register are described below.

BIT	NAME	DESCRIPTION	RESET VALUE
0	Address-decode enable	Bit 0 is used to control whether or not the device accepts accesses to its expansion ROM. When this bit is 0, the device's expansion-ROM address space is disabled. When this bit is 1, address decoding is enabled using the parameters in the other part of the register. This allows the device to be used with or without an expansion ROM, depending on the system configuration. The memory-space bit in the command register has precedence over this bit. A device must respond to accesses to expansion ROM only when the memory-space bit is set and the expansion-ROM base-address enable bit are set. After $\overline{\text{RST}}$, the value in bit 0 goes to 0.	0
1–10	Reserved	Reserved	
11–31	Expansion-ROM base address	Bits 11–31 correspond to the upper 21 bits of the expansion-ROM base address. However, only the top 16 bits are used since the expansion ROM needs only 64K bytes.	

interrupt-line register (offset address 3Ch)

The interrupt-line register is an 8-bit register that is used to communicate the routing of the interrupt. This register is written by the HOST software during system initialization. The value in this 8-bit register indicates which input of the system-interrupt controller is connected to the PCI-SAR interrupt terminal. The typical value is between 0 and 15. The interrupt-line register is located at offset address 3Ch in the PCI configuration space and is read and written by the host.

interrupt-pin register (offset address 3Dh)

The interrupt-pin register is an 8-bit register, indicating the interrupt pin that the TNETA1561 is using. The PCI SAR is defined as a single-function device, uses only interrupt A, and has a value of 1. The interrupt-pin register is located at offset address 3Dh in the PCI configuration space and is read only.

minimum-grant register (offset address 3Eh)

The minimum-grant register is an 8-bit register that specifies the length of the data burst required by the TNETA1561 for every PCI-bus grant. This specifies the length of the burst period that the PCI-SAR device needs in 0.25-μs units. The typical value of 0.75 μs (decimal 3) is defined for PCI SAR. The minimum-grant register is located at offset address 3Eh in the PCI configuration space and is read only.

maximum-latency register (offset address 3Fh)

The maximum-latency register is an 8-bit register that defines the maximum-latency value for the PCI SAR. This specifies how often the PCI-SAR gains access to the PCI bus in 0.25-μs units. A typical value of 10 μs (decimal 40) is defined for the PCI SAR. The maximum-latency register is located at offset address 3Fh in the PCI configuration space and is read only.

PRINCIPLES OF OPERATION

PCI-SAR control and status registers

The PCI SAR has defined the following registers for status and control information.

OFFSET ADDRESS (24-BIT HEX)	DESCRIPTION	WIDTH (BITS)	READ/WRITE
00E000	Software reset	32	Write only
00E004	SAR-status register	32	Read only
00E008	Interrupt-enable mask register	32	Read/write
00E010	Reserved	32	—
00E00C	SAR-configuration register	32	Read/write
00E014	BWG-table-size register	32	Read/write
00E018	Transmit/receive FIFO maximum-depth register	32	Read/write
00E01C	Reserved	32	—
00E020	Clear-transmit-freeze command	32	Write only
00E024	Clear-receive-freeze command	32	Write only

software-reset register (offset address 00E000h)

The PCI SAR reset operation is enabled when there is a hardware reset or when the host device writes into the software-reset register. During a software-reset operation, the host reads initially the current status of the PCI command/status register. The host writes to the PCI SAR software-reset register. Any value can be written into the register by the host software. When the host writes into the software-reset register, the PCI SAR reset operation is enabled. To reactivate the PCI SAR, the host has to write appropriate values in the PCI command/status register.

PCI-SAR-status register (offset address 00E004h)

The PCI-SAR-status register is read only for the host. All the bits, except the transmit-freeze bit and the PCI-bus error flags, are cleared when the register is read. The PCI SAR generates a PCI-bus interrupt to the host if one of the bits in the register is set and if the condition represented by the bit is enabled by the interrupt-enable mask register. The PCI-bus interrupt is an asynchronous signal that is held until the system clears the condition that caused the interrupt. The bit format is shown in following table:

Reserved (bit 11)	Local-bus interrupt (bit 10)	Reserved (bit 9)	Reserved (bit 8)
Receive freeze (bit 7)	Transmit freeze (bit 6)	Transmit completion not available (bit 5)	Receive completion not available (bit 4)
Receive big-free buffer not available (bit 3)	Receive small-free buffer not available (bit 2)	Transmit completion update (bit 1)	Receive completion update (bit 0)

transmit completion update and receive completion update (bits 1–0)

The transmit- or receive-completion update bit is set when the hardware releases a transmit or receive descriptor, respectively, to the completion ring. This is initiated when the OWN bits in the respective DMA blocks are cleared by the TNETA1561.

PRINCIPLES OF OPERATION

receive big free-buffer not available and receive small free-buffer not available (bits 3–2)

The appropriate receive free-buffer not-available bit is set when the first entry in the corresponding receive free-buffer ring is not available. This is indicated when the OWN bit in the first entry of the free ring is zero. The incoming cell is deleted because there is no buffer available to hold it. This eventually causes the loss of the entire packet due to the resultant CRC error. The buffer allocation-error bit in the DMA block is set. This is indicated by a zero in the first free-buffer ring entry.

receive completion-ring not available (bit 4)

The receive completion-ring not-available bit is set when the next descriptor in the receive completion ring is not released by the host. This is indicated when the OWN bit in the entry is zero (host owns it). This packet and buffer are both lost to host memory.

transmit completion-ring not available (bit 5)

The transmit completion-ring not-available bit is set when the next descriptor in the transmit completion ring is not released by the host. This is indicated when the OWN bit in the entry is zero. The transmit-freeze bit is set when this bit is set, disabling all transmit operation until the transmit-freeze bit is cleared via an active command from the host.

transmit freeze (bit 6)

The transmit-freeze bit is set when the transmit completion-ring not-available bit is set, disabling all transmit operation until the transmit-freeze bit is cleared via an active command from the host. This has the same effect on the transmit circuitry as disabling the enable-transmit bit.

receive freeze (bit 7)

The receive-freeze bit is set when the receive completion-ring not-available bit is set, disabling all receive operation until the receive-freeze bit is cleared via an active command by the host. The buffer that could not be posted is effectively lost, and the host must find some way to recover it while the freeze is in operation. The receive-freeze indicator has the same effect on the receive path as disabling the enable-receive bit.

reserved (bit 8)

reserved (bit 9)

local-bus interrupt (bit 10)

The local-bus interrupt bit is set if an interrupt is generated on the local bus.

reserved (bit 11)

interrupt-enable mask register (offset address 00E008h)

Unused (bits 31–12)	Mask bits (bits 11–0)

An interrupt-enable mask-register bit has a bit that corresponds to every entry in the PCI-SAR status register. When a bit is set in the status register, an interrupt is generated if a corresponding bit in the interrupt-enable mask register also is set.

SAR-configuration register (offset address 00E00Ch)

The SAR-configuration register holds various values pertaining to the overall PCI-SAR configuration. The host can read the register and is allowed to program the EN receive and the EN transmit bits. In addition, two more bits are defined for posted write-buffer enable (PWBE) and software reset (SR).

Unused (bits 31–5)	SDH (bit 5)	Unused (bits 4–3)	EN receive (bit 2)	EN transmit (bit 1)	0 (bit 0)

PRINCIPLES OF OPERATION

enable-transmit operation (EN transmit) (bit 1)

The EN-transmit bit allows the host to disable packet-to-cell segmentation and any payload-data transfer from the host to the link. The EN-transmit bit is set high to enable normal transmit processing and set to zero to disable such processing. The EN-transmit bit is set to zero on reset, disabling transmit operation until various configuration registers, the BWG table, and DMA blocks are configured by the host. The transfer of the new cells from PCI bus to PCI SAR is inhibited when the enable-transmit bit is disabled. Cells already in the output buffer are forwarded to the PHY layer.

enable-receive operation (EN receive) (bit 2)

The EN-receive bit allows the host to disable packet reassembly. All cells from the PHY layer are dropped when the EN-receive bit is zero. The EN-receive bit is set high to enable normal processing and is set to zero on reset, disabling receive operation until various configuration registers and the DMA blocks are reconfigured by the host. The transfer of the new cells from ATM link to the receive buffer is inhibited when the enable-receive bit is disabled.

SDH bit (bit 5)

If the SDH bit is set to zero, the TNETA1561 transmits null cells (unassigned cells) when no valid cells are ready for transmission. If SDH bit is set to one, the device transmits idle cells as fillers.

BWG table-size register (offset address 00E014h)

Unused (bits 31–11)	BWG table size (bits 10–0)

The 11-bit BWG table-size register allows the user to configure the size of the BWG table in 4-byte words. Each word in the table consists of four 8-bit entries. The maximum table size is 1200 (decimal) allowing 4800 entries. A resolution of 32 kbit/s is achieved with 4800 entries. The number of entries in the table is one more than the number programmed in this register and there is one entry in the table when the register is set to zero.

transmit/receive FIFO maximum-depth register (offset address 00E018h)

Unused (bits 31–20)	Maximum receive FIFO depth (bits 19–10)	Maximum transmit FIFO depth (bits 9–0)

This is the only set of statistics collected by the TNETA1561 because it is useful information for queuing analysis in different platforms with varying PCI-bus clock speeds and latencies. These registers are not of the read and reset variety and must be set to zero to restart the measurement.

Clear transmit-freeze command (offset address 00E020h)

The transmit-freeze bit is set when the transmit completion-ring not-available bit is set and the transmit side is disabled. The host enables the transmit side by writing any value into this register.

Clear receive-freeze command (offset address 00E024h)

The receive-freeze bit is set when the receive completion-ring not-available bit is set and the receive side is disabled. The host enables the receive side by writing any value into this register.

TNETA1561 ERRATA

The following timing requirements deviate from PCI-Local Bus Specification (revision 2.0). They are shown in Figures 3 through 6.

FIGURE NO.			PCI-LOCAL BUS SPEC (REV 2.0)		TNETA1561		UNIT
			MIN	MAX	MIN	MAX	
3	$t_{h(PTRDY)}$	Hold time, $\overline{\text{PTRDY}}$ low after PCLK↑, write operation	0		3		ns
3	$t_{h(PDEVSEL)}$	Hold time, $\overline{\text{PDEVSEL}}$ low after PCLK↑, write operation	0		3		ns
4	$t_{w(PCLKL)}$	Pulse duration, PCLK low	12		13		ns
4	$t_{h(PAD)}$	Hold time, PAD31–PAD0 valid after PCLK↑	0		2		ns
4	$t_{h(PTRDY)}$	Hold time, $\overline{\text{PTRDY}}$ low after PCLK↑, read operation	0		2		ns
4	$t_{h(PDEVSEL)}$	Hold time, $\overline{\text{PDEVSEL}}$ low after PCLK↑, read operation	0		2		ns
5	$t_{h(PIDSEL)}$	Hold time, PIDSEL high after PCLK↑, read operation	0		1		ns
5	$t_{h(PCBE)}$	Hold time, PCBE3–PCBE0 valid after PCLK↑	0		1		ns
5	$t_{h(PIRDY)}$	Hold time, $\overline{\text{PIRDY}}$ low after PCLK↑, read operation	0		1		ns
6	$t_{h(PIDSEL)}$	Hold time, PIDSEL high after PCLK↑, write operation	0		2		ns
6	$t_{h(PIRDY)}$	Hold time, $\overline{\text{PIRDY}}$ low after PCLK↑, write operation	0		1		ns
6	$t_{h(PFRAME)}$	Hold time, $\overline{\text{PFRAME}}$ low after PCLK↑	0		3		ns