TNETA1585
Programmer's Reference Guide

SDNU016A
December 1996

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Contents

Title *Page*

Contents (continued)

Title *Page*

List of Tables

Appendices

Appendices (continued)

List of Tables

1. INTRODUCTION

The TNETA1585 Traffic Management Scheduler is programmable, giving it the capability to perform different applications when provided with the appropriate application software. TNETA1585 data structures are initialized and configured according to the requirements of the particular application software that it has been programmed with. Thus, a unique programmer's reference guide is associated with each application software available for the TNETA1585.

This programmer's reference guide provides the guidelines for initialization and configuration of the TNETA1585 as it is programmed to implement four service types:

- VC-Level ABR
- VP-Level ABR
- VBR-nrt
- PSR Drain

The VC and VP-Level ABR and VBR-nrt service types are implemented by the TNETA1585 as specified in the ATM Forum Traffic Management 4.0 Specification. PSR Drain is described in the appropriate appendix, in addition to some optional features. The service types are allowed to intermix with each other and have equal priority in the implementation described by this document.

2. APPLICATIONS

The TNETA1585 provides available bit rate (ABR) scheduling for the TNETA1575 HyperSAR-PLUS. ABR support is required in classical LAN-to-ATM translation products, such as networking hubs. LAN emulation depends on the establishment of a large number of VCCs; ABR is the only service category that enables a single end system to establish a large number of connections (1000s) by setting the minimum cell rate (MCR) to zero.

The major features of the TNETA1585 are summarized in the following:

- A receive cell interface (UTOPIA) that is capable of rates beyond 155 Mbps. This increases the number of front-end classical LAN interfaces that are supported by one instance of the TNETA1575/TNETA1585 chip set. The TNETA1585 supports higher rate UTOPIA interfaces.
- Two-tiered scheduling of constant-bit-rate (CBR) traffic. The TNETA1585 is mainly focused on the scheduling of ARB and VBR-nrt VCCs. The HyperSAR-PLUS offers features for higher-tiered scheduling of CBR traffic.
- Support for the full range of virtual paths and virtual channels is a requirement for the collapsed backbones that are a part of virtual LANs, giving greater flexibility in VCC selection. The TNETA1585 is VPI and VCI independent, as it only processes channel identifiers assigned by the host. The TNETA1585 supports the same range of VCI and VPI values that the SAR supports, which, in the case of the TNETA1575, is the full range.
- Support for VP-Level ABR is particularly important for ATM uplinks in networking hubs, which are connected to the wide area network (WAN). WAN connections typically support VP-Level ABR, to flow control a large number of VCs.
- An option to program the receive UTOPIA cell interface as either a PHY- or ATM-layer system provides the TNETA1585 with the capability to function as a PHY where the cell switch is distributed; or as an ATM-layer interface for adapter cards or ATM uplinks.

The TNETA1585 also may be used as a scheduler in ATM switches to schedule the transmission of cells on ABR VCCs. This is particularly useful for switches that wish to implement virtual source (VS) and virtual destination (VD) capabilities on selected ports.

2. APPLICATIONS (continued)

In this case, the TNETA1585 would provide the ABR scheduling required for a switch port to provide VS/VD support. As ABR is deployed in classical LAN-to-ATM translation products and across ATM networks, there will be the need for switches to bridge the gap between existing non-ABR supporting networks and networks that support ABR. VS/VD support is one such method.

Features that the TNETA1585 provides that are attractive to this class of product are summarized in the following.

- Support for the full range of virtual paths and virtual channels is an attractive feature for switches. VPI/VCI range support by switches themselves differs greatly; covering the full range ensures the TNETA1585 is suited for all switches. The TNETA1585 is VPI and VCI independent as it only processes channel identifiers assigned by the switch. The TNETA1585 supports the same range of VCI and VPI values that the switch supports.
- Support for VP-Level ABR is particularly important for switch ports that are connected to the WAN. WAN connections typically support VP-Level ABR, to flow control a large number of VCs.
- An option to program the receive UTOPIA cell interface as either a PHY- or ATM-layer system provides the TNETA1585 with the capability to function as a PHY- or as an ATM-layer interface. The location of the TNETA1585 in a switch application may vary greatly. This flexibility ensures that the TNETA1585 fits in all configurations.

3. DATA STRUCTURES

The TNETA1585 uses data structures that reside in internal registers, on-chip RAMs, or in external Parameter Memory. All internal registers are accessed directly by applying their address directly to the TNETA1585 Slave Interface address bus while using the associated data bus for data transfers. All internal RAM and all of Parameter Memory are accessed indirectly through the use of intermediate registers (Read Address, Read Data, Write Address, and Write Data Registers). For read operations, the host must provide the desired RAM or Parameter Memory address to the Read Address Register and read the associated data from the Read Data Register after the TNETA1585 has fetched it internally. Write operations are done similarly.

NOTE:

The data structures support 2048 simultaneous connections, although the TNETA1585 scheduler actually supports 2047 simultaneous connections.

3.1 Direct-Access Data Structures

All host-visible registers of the TNETA1585 can be directly accessed by the host. When used with the TNETA1575 SAR device, the TNETA1585 internal registers are mapped into the SAR control memory space, starting at address 10000 and extending to address 1FFFF. However, since the Slave Interface address bus is six bits wide, only the first 64 addresses are used to address TNETA1585. Table 3–1 lists these registers and their addresses with respect to TNETA1575 control memory address map and PCI offset address map.

Table 3–1. Register (Direct) Address Map

DESCRIPTION	READ/WRITE	POWER UP RESET VALUE	CONTROL MEMORY ADDRESS FOR TNETA1575	PCI OFFSET ADDRESS FOR TNETA1575
Reserved	—	—	h10000–7	h40000–1C
Configuration Register	R/W	h00000000	h10008	h40020
Status Register	R	h0000001E	h10009	h40024
Interrupt Mask Register	R/W	h00000000	h1000A	h40028
Schedule On Register	W	—	h1000B	h4002C
Schedule Off Register	W	—	h1000C	h40030
Read Address Register	R/W	h00000000	h1000D	h40034
Read Data Register	R	h00000000	h1000E	h40038
Write Address Register	R/W	h00000000	h1000F	h4003C
Write Data Register	W	—	h10010	h40040
Clock Frequency Register	R/W	h0018FCA1	h10011	h40044
Revision Number Register	R	h00000000	h10012	h40048
ACR Low Register	R	h80000XXX†	h10013	h4004C
ACR OK Register	R	h80000XXX†	h10014	h40050
Reserved	—	—	h10015–3F	h40054–FC

† The channel field is not reset to a known value.

The following sections provide a detailed description of each register.

3.1.1 Configuration Register

Reserved (31:10)	Configuration (9:0)

Bit(s) (31:10) **Reserved**

Bit(s) (9) **Enable_RX_UTOPIA**
A configuration field used by the host to configure the device to turn on/off the receive UTOPIA interface. Writing a 1 to this field turns on the interface.
After power up/reset, the receive UTOPIA interface is disabled.

Bit(s) (8:7) **Perform_CRC**
A configuration field used by the host to invoke CRC checking on the code loaded into the Instruction RAM. The following encoding applies:

BIT 8	BIT 7	INDICATION
1	0	Reserved
0	x	Reserved
1	1	Perform CRC checking on the code loaded in Instruction RAM

After power up/reset, CRC checking is disabled.

3.1.1 Configuration Register (continued)

Bit(s) (6) **Global Software Reset**
A configuration field used by the host to reset the entire TNETA1585 device

Bit(s) (5) **NNI_UNI**
A configuration field used by the host to indicate NNI or UNI mode of operation. It is used by the TNETA1585 to determine how to interpret the received ATM cell header when extracting the received VPI and VCI fields for comparison with the Expected_VPCI parameter in Parameter Memory. A clear bit indicates NNI mode of operation, i.e., the TNETA1585 uses the first four bits in the received header as part of the VPI value. A set bit indicates UNI mode of operation, i.e., the TNETA1585 does not use the first four bits in the received header as part of the VPI value. The upper four bits of the Expected_VPCI field in Parameter Memory also are set to zero when in UNI mode.

The host must initialize this at power up or reset to the appropriate value.

Bit(s) (4) **Reset_Destination_Behavior_Processor**
A configuration field used by the host to reset the destination processor

Bit(s) (3) **Reset_Source_Behavior_Processor**
A configuration field used by the host to reset the source processor

Bit(s) (2) **Enable_Destination_Behavior_Processor**
A configuration field used by the host to enable or pause (suspend execution of) the destination processor. By writing a 1 to this field, the host enables the destination processor. By writing a 0 to this field, the host pauses the destination processor.

After power up/reset, the destination processor is disabled.

Bit(s) (1) **Enable_Source_Behavior_Processor**
A configuration field used by the host to enable or pause the source processor. By writing a 1 to this field, the host enables the source processor. By writing a 0 to this field the host pauses the source processor.

After power up/reset, the source processor is disabled.

Bit(s) (0) **Enable_Scheduler**
A configuration field used by the host to configure the device to turn on/off the ABR scheduler. Writing a 1 to this field turns on the ABR scheduler.

After power up/reset, the scheduler is disabled.

3.1.2 Status Register

This register provides current status information used by the host either to determine the cause of an interrupt received by the host from the TNETA1585 device or to determine the readiness of certain functions. Some fields of this register are “clear on read”, i.e., the contents are cleared when the register is read. The entire register is cleared when a hardware or software reset occurs, except where otherwise noted.

Behavior Status (31:16)	Reserved (15:8)	Status (7:0)

Bit(s) (31:16) **Behavior_Status**
These are source and destination processor-defined status bits. Each bit may be set by either the source or destination processor in the TNETA1585 to generate an outgoing interrupt.

Clear on read

3.1.2 Status Register (continued)

Bit(s) (31) **Synchronization Error**
This bit is used to support the Receive Synchronization Check feature (see Appendix F). The TNETA1585 sets this bit when it detects a mismatch of received VPI/VCI with the expected VPI/VCI, indicating that the TNETA1575 and TNETA1585 receive FIFOs have become misaligned.

Clear on read

Bit(s) (30) **PSR Drain**
This bit is used to support PSR Drain operation (see Appendix C). A set bit indicates completion of PSR Drain operation on a channel.

Clear on read

Bit(s) (15:8) **Reserved**

Bit(s) (7) **CRC_Check_in_Progress**
This bit, when set to 1, indicates that a previously requested CRC check operation is in progress.

Not clear on read

Bit(s) (6) **ACR_is_Low**
This bit is used to support Low ACR Detection (see Appendix D). A set bit indicates that a channel has had its ACR reduced below the programmed threshold. The host should check the ACR Low Register to determine which channels are affected.

Clear on read

Bit(s) (5) **ACR_is_OK**
This bit is used to support Low ACR Detection (see Appendix D). A set bit indicates that a channel has had its ACR increase above the programmed hysteresis threshold. The host should check the ACR OK Register to determine which channels are affected.

Clear on read

Bit(s) (4) **Scheduler_On_Ready**
This status bit may be polled by the host whenever it wishes to turn scheduling on for a channel. A set bit indicates that a previously requested Schedule On operation has completed and an additional Schedule On request can be made using the Schedule On Register.

Reset value of one (Scheduler ready to add scheduling entry).

Not clear on read

Bit(s) (3) **Scheduler_Off_Ready**
This status bit may be polled by the host whenever it wishes to turn scheduling off for a channel. A set bit indicates that a previously requested Schedule Off operation has completed and an additional Schedule Off request can be made using the Schedule Off Register.

Reset value of one (Scheduler ready to mark channel for deletion).

Not clear on read

3.1.2 Status Register (continued)

Bit(s) (2) **Write_Ready**
A set bit indicates that the Write Address and Write Data Registers are ready for the next write operation. After the Write Data Register is written, this bit is cleared until the write operation has completed.

Reset value of one (ready to perform write operation).

Not clear on read

Bit(s) (1) **Read_Data_Ready**
A set bit indicates that data is valid in the Read Data Register. This bit is polled by the host after it writes an address to the Read Address Register to determine if the Read Data Register contains the requested data yet. This bit will remain set after a read operation until the next read operation, when it will be cleared until data is ready.

Reset value of one (ready to perform read operation)

Not clear on read

Bit(s) (0) **Code_Invalid**
A set bit indicates that the CRC check performed on the code loaded into the Instruction RAM resulted in error. This bit is initialized to zero whenever CRC checking is invoked by writing to the Perform_CRC field of the Configuration Register.

Not clear on read

3.1.3 Interrupt Mask Register

Interrupt Mask (31:16)	Reserved (15:8)	Interrupt Mask (7:0)

Bit(s) (31:0) **Interrupt Mask**
The host can use this register to mask interrupts caused by the conditions given in the Status Register. This register has a bit-to-bit correspondence to the Status Register. When a bit in the Status Register is set, an interrupt is generated only if the corresponding bit in the Interrupt Mask Register is also set.

This register is cleared whenever a hardware or software reset occurs.

3.1.4 Schedule On Register

Reserved (31:11)	Channel ID (10:0)

This register is used by the host to turn on scheduling for a connection. To turn on scheduling, the host shall write the channel number, in the range of 0..2047, of the connection to be activated to this register. The host must first poll the Scheduler_On_Ready bit in the Status Register to verify that the scheduler is ready to add a new connection before writing to this register.

Bit(s) (31:11) **Reserved**

Bit(s) (10:0) **Channel ID**

3.1.5 Schedule Off Register

Reserved (31:11)	Channel ID (10:0)

This register is used by the host to turn off scheduling of a connection. To turn off scheduling, the host shall write the channel number, in the range of 0..2047, of the connection to be turned off to this register. The host must first poll the Scheduler_Off_Ready bit in the Status Register to verify that the scheduler is ready to remove a connection before writing to this register.

Bit(s) (31:11) **Reserved**

Bit(s) (10:0) **Channel ID**

3.1.6 Read Address Register

Reserved (31:21)	Read Address (20:0)

This register is used by the host, in conjunction with the Read Data Register, to read from indirectly mapped data structures. The indirect address of the word that is to be read is written to this register by the host, after which, the data is read from the Read Data Register by the host. After this register is written, the Read_Data_Ready bit in the Status Register is cleared until the request is serviced, at which time the Read_Data_Ready bit is set and the read data is placed in the Read Data Register. The host must poll the Read_Data_Ready bit of the Status Register until it is set to verify that the appropriate word has been fetched and loaded into the Read Data Register before it attempts to retrieve the word from the Read Data Register.

Bit(s) (31:21) **Reserved**

Bit(s) (20:0) **Read Address**

3.1.7 Read Data Register

Data (31:0)

This register is used by the host, in conjunction with the Read Address Register, to read from indirectly mapped data structures. The host shall read the data from this register after it writes the indirect address of the desired word into the Read Address Register. The host must poll the Read_Data_Ready bit of the Status Register to verify that it is set before it attempts to retrieve the word from this register.

Bit(s) (31:0) **Read Data**

3.1.8 Write Address Register

Reserved (31:21)	Write Address (20:0)

This register is used by the host, in conjunction with the Write Data Register, to write to indirectly mapped data structures. The indirect address of the location is written to this register by the host, after which, the data to be written is written to the Write Data Register by the host. The host must poll the Write_Ready bit of the Status Register to verify that any previous write operations has been completed before it attempts to initiate a new write operation.

When the write operation has completed, the value in this register is automatically incremented. Since the write operation is initiated by a write to the Write Data Register, and not by a write to this register, writing to a block of sequential locations requires writing to this register only once, at the beginning of the sequence. This function facilitates the initialization of large contiguous address ranges such as Instruction RAM or Parameter Memory.

Bit(s) (31:21) **Reserved**

Bit(s) (20:0) **Write Address**

3.1.9 Write Data Register

Data (31:0)

This register is used by the host, in conjunction with the Write Address Register, to write to indirectly mapped data structures. The host shall write the data to be written to this register after it writes the indirect address of the location to be written to to the Write Address Register. The write operation is initiated by a write to this register. After this register is written, the Write_Ready bit in the Status Register is cleared until the write operation is completed, at which time the Write_Ready bit is set. The host must poll the Write_Ready bit of the Status Register to verify that any previous write operations have been completed before it attempts to initiate a new write operation.

Bit(s) (31:0) **Write Data**

3.1.10 Clock Frequency Register

Reserved (31:21)	Clock Frequency Exponent (20:16)	Clock Frequency Mantissa (15:0)

This register is used by the host to specify the frequency of operation of the TNETA1585. When used with the TNETA1575, it is the same as the frequency of the PCI Bus Interface on the TNETA1575. This register contains a 21-bit floating point value that is a representation of the clock frequency in cycles per second.

The relation between the integer clock frequency and floating-point frequency is as follows:

integer clock frequency = $2^{\text{clock frequency exponent}}$ (1 + (clock frequency mantissa / 65536))

e.g., for default 33-MHz operation, this register is configured by hardware to decimal 33,333,333 after power up/reset.

Bit(s) (31:21) **Reserved**

Bit(s) (20:16) **Clock Frequency Exponent**
This register contains the default value of 18H at power up.

Bit(s) (15:0) **Clock Frequency Mantissa**
This register contains the default value of FCA1H at power up.

3.1.11 Revision Number Register

Reserved (31:8)	Revision ID (7:0)

Bit(s) (31:8) **Reserved**

Bit(s) (7:0) **Revision_ID**
Identifies the revision of the TNETA1585.

3.1.12 ACR Low Register

Empty (31)	Ready (30)	Reserved (29:11)	Channel Number (10:0)

This register is provided to support the Low ACR Detection function (see Appendix D). It is used by the host to determine the channel number of a connection whose ACR has dropped below a certain threshold. The host can either poll this register or read it in response to an interrupt. (The host would need to configure the ACR_is_Low interrupt bit in the Interrupt Mask Register appropriately.) Register contents are valid only when the Empty bit is clear and the Ready bit is set. The contents of this register are updated each time it is read.

Bit(s) (31) **Empty**
Since more than one channel can have their ACR below the threshold by the time the host reads this register, a list of channel numbers is maintained by the TNETA1585. A set Empty bit indicates that that list is empty, i.e., there are currently no channels that have their ACR below the threshold, and thus the register contents are invalid. It may be possible for the host to be interrupted only to read that the Empty bit is set, since the channel's ACR could have risen above the threshold by the time the host read this register.

Bit(s) (30) **Ready**
A set bit indicates that the register contents are valid if the Empty bit is clear. This bit is cleared for several clock cycles after a valid channel number has been read from this register while the next channel in the list is fetched, if there is more than one channel in the list. The host should always check this bit to determine if the channel number is valid, as it is possible for the list to be non-empty while the channel number is invalid.

Bit(s) (10:0) **Channel Number**
Channel number of the connection whose ACR is low.

3.1.13 ACR OK Register

Empty (31)	Ready (30)	Reserved (29:11)	Channel Number (10:0)

This register is provided to support the Low ACR Detection function (see Appendix D). It is used by the host to determine the channel number of a connection whose ACR has risen above a certain threshold after having been below the threshold. The host can either poll this register or read it in response to an interrupt. (The host would need to configure the ACR_is_OK interrupt bit in the Interrupt Mask Register appropriately.) Register contents are valid only when the Empty bit is clear and the Ready bit is set. The contents of this register are updated each time it is read.

Bit(s) (31) **Empty**
Since more than one channel could have had their ACR increase above the threshold by the time the host reads this register, a list of channel numbers is maintained by the TNETA1585. A set Empty bit indicates that that list is empty, i.e., there are currently no channels that have had an ACR increase above the threshold, and thus the register contents are invalid. It may be possible for the host to be interrupted only to read that the Empty bit is set, since the channel's ACR could have dropped back below the threshold by the time the host read this register.

Bit(s) (30) **Ready**
A set bit indicates that the register contents are valid if the Empty bit is clear. This bit is cleared for several clock cycles after a valid channel number has been read from this register while the next channel in the list is fetched, if there is more than one channel in the list. The host should always check this bit to determine if the channel number is valid, as it is possible for the list to be non-empty while the channel number is invalid.

Bit(s) (10:0) **Channel Number**
Channel number of the connection whose ACR is OK.

3.2 Indirect Access Data Structures

All RAM data structures, internal and external, are indirectly accessed by the host. This includes Parameter Memory and all the on-chip RAMs. To perform an indirect access, the host must write to the Read Address or Write Address Register the indirect address of the location. It must then read or write the data from or to the Read Data Register or the Write Data Register.

To perform an indirect read, the following steps must be performed:

1. Write the address of the location to be read to the Read Address Register.
2. Poll the Read_Data_Ready bit in the Status Register to determine when the data has been loaded into the Read Data Register, i.e., poll until Read_Data_Ready bit is set.
3. Read the Read Data Register to obtain the read word.

To perform an indirect write, the following steps must be performed:

1. Poll the Write_Ready bit in the Status Register until it indicates that the previous write has been completed, i.e., poll until Write_Ready is set.
2. Write the address of the location to be written to the Write Address Register
3. Write the data to be written to the Write Data Register.
4. Repeat steps 1 and 3 for each sequential word that is to be written. Step 2 can be skipped since the contents of the Write Address Register are incremented automatically at the end of each write operation.

For both the read and write operations, the host must use one of the addresses given in Table 3–2, which lists the addresses of all the indirect access data structures along with their size.

3.2 Indirect Access Data Structures (continued)

Table 3–2. Parameter Memory and Internal RAM (Indirect) Address Map

DESCRIPTION	ADDRESS	SIZE (IN 32 BIT WORDS)
Parameter Memory	h000000–00FFFF	65536 (64K)
Reserved	h010000–0FFFFF	983040 (960K)
Zero RAM 1	h100000–10007F	128 (.125K)
Reserved	h100080–1100FF	65664 (64.125K)
Zero RAM 2	h110100–11013F	64 (.0625K)
Reserved	h110140–11FFFF	65216 (63.6875K)
Instruction RAM	h120000 – 1203FF	1024 (1K)
Reserved	h120400 – 13FFFF	130048 (127K)
Real-Time Variables RAM	h140000 – 1407FF	2048 (2K)
Reserved	h140800 – 1FFFFF	784384 (766K)

3.2.1 Parameter Memory

Parameter Memory is used to store all the parameters required to maintain a connection. The set of parameters for a particular connection define the state of that connection at any given time and is thus referred to as the connection state. A maximum of 2K connections can be supported by Parameter Memory.

There are a total of 64K $\times$ 1 word locations in Parameter Memory. The first 32K is equally partitioned into 2K $\times$ 16 (for 2K connections) word blocks, where each 16-word block is associated with a connection and its channel number. Each block is thus located in the order of the channel number associated with it. The second 32K of Parameter Memory is reserved for later expansion. When the host configures Parameter Memory for a new connection, it configures the connection state associated with the channel number of the new connection. Table 3–3 shows the address range for each block associated with a connection and its channel number in Parameter Memory.

Table 3–3. Parameter Memory Channel Number Map

PARAMETER MEMORY ADDRESS	CONNECTION CHANNEL NUMBER
h00000 – 0000F	0
h00010 – 0001F	1
h00020 – 0002F	2
h00030 – 0003F	3
h00040 – 0004F	4
h00050 – 0005F	5
h00060 – 0006F	6
—	—
—	—
h07FD0 – 07FDF	2045
h07FE0 – 07FEF	2046
h07FF0 – 07FFF	2047

3.2.1 Parameter Memory (continued)

The definition and configuration of each 16-word connection state depends on the service type of the connection, i.e., same word locations within the 16-word block have different meanings, depending on the service type. There are four service types for this application: VC-Level ABR, VP-Level ABR, VBR-nrt, and PSR Drain. Tables 3–4 through 3–7 summarize the configuration of Parameter Memory for each of the service types.

The configuration of Parameter Memory also depends on the direction of the connection, where parameters associated with the TX connection are configured using the TX channel number, while parameters associated with the RX connection are configured using the RX channel number. This does not apply to VC-Level ABR connections, however, since ABR connections are always full-duplex, i.e., the connection is set up in both directions, resulting in identical TX and RX channel numbers. Thus, all the parameters for a VC-Level ABR connection, whether they are associated with the TX or RX side of the connection, are configured per the one TX/RX channel number. For VP-Level ABR, one TX channel number is always associated with one or more RX channel numbers. Thus, parameters associated with the TX side of the connection are configured using the TX channel number while parameters associated with the RX side of the connection are configured for each of the RX channel numbers associated with the connection. VBR-nrt are set up as half-duplex connections. The TX connections require Parameter Memory configuration using the TX channel number. The RX connections do not require any Parameter Memory configuration. Service-type PSR Drain is used as part of connection teardown for flushing out non-empty Packet Segmentation Rings and is thus set up for TX connections. Parameters for PSR Drain are configured per the TX channel number.

The address of a particular parameter in Parameter Memory can be derived using the channel number of the connection using that parameter. Specifically, the location within the connection state block of the word that contains that parameter is added to a base address as follows:

PM address = word location within connection state block + base address

Where:

base address = connection channel number $\times$ 16, and
word location is as specified in Tables 3–4 through 3–7.

The connection channel number shall either be a TX or RX channel number, depending on whether the parameter is associated with a TX or RX connection. The following example is provided for clarity (refer to Table 3–4):

To obtain the Nrm parameter for TX channel number 4 for a VC-Level ABR connection:
Channel number = 4
Word location = 7
PM address = 7 + (4 $\times$ 16) = 71

Where:

address 71 would need to be written to the Read Address or Write Address Register to access this parameter.

NOTE:

The appropriate bit fields would need to be read/written; in this example, bits 23–16 contain the Nrm parameter.

3.2.1.1 VC-/VP-Level ABR Service-Type Parameters

Tables 3–4 and 3–5 list all the parameters that define the state of a VC or VP-Level ABR service-type connection, along with their word locations within the connection state block, and their bit locations. These parameters include ABR parameters defined in the ATM Forum TM 4.0 Specification. Table 3–4 lists all parameters associated with the TX side of the connection, while Table 3–5 lists all parameters associated with the RX side of the connection.

NOTE:

For VC-Level ABR, the channel number is the same for all parameters, regardless of whether that parameter is TX or RX connection associated. For VP-Level ABR, all the parameters associated with the TX side of the connection must be configured using the TX channel number. Likewise, all the parameters associated with the RX side of the connection must be configured for each of the RX channel numbers associated with the connection.

Table 3–4. ABR Service-Type TX Connection State Parameters

WORD	BIT LOCATION(S)	DESCRIPTION
0	(23:0)	PCR
	(31:24)	Reserved
1	(23:0)	MCR
	(31:24)	Reserved
2	(23:0)	ICR
	(31:24)	Reserved
3	(23:0)	ACR
	(31:24)	Reserved
4	(31:0)	ADTF_Ticks
5	(7:0)	RDF
	(15:8)	RIF
	(23:16)	CDF
	(28:24)	ACR_Increase_Delta
	(31:29)	Reserved
6	(31:0)	Trm_Ticks
7	(7:0)	Service Type
	(8)	Calling_Party
	(15:9)	Reserved
	(23:16)	Nrm
	(31:24)	Crm
8	(31:0)	Reserved
9	(7:0)	Polling_Rate_Coeff
	(31:8)	Reserved
10–14	(31:0)	Reserved
15	(15:0)	Reserved
	(31:16)	ACR_Low_Threshold

Table 3–5. ABR Service-Type RX Connection State Parameters

WORD	BIT LOCATION(S)	DESCRIPTION
0–7	(31:0)	Reserved
8	(0)	Receive_On
	(3:1)	Reserved
	(19:4)	Expected VCI
	(31:20)	Expected VPI
9–14	(31:0)	Reserved
15	(10:0)	RX/TX Channel Cross Reference
	(31:11)	Reserved

The following provides a detailed description for each of the fields listed in Tables 3–4 and 3–5:

3.2.1.1.1 Parameter Memory Word 0

Reserved (31:24)	PCR (23:0)

Bit(s) (31:24) **Reserved**

Bit(s) (23:0) **PCR (Peak Cell Rate)**
This field contains the 24-bit integer representation of the PCR parameter (h000001 represents one cell per second), which is negotiated during call setup.

This field should be initialized by the host with the negotiated parameter value at connection setup.

3.2.1.1.2 Parameter Memory Word 1

Reserved (31:24)	MCR (23:0)

Bit(s) (31:24) **Reserved**

Bit(s) (23:0) **MCR (Minimum Cell Rate)**
This field contains the 24-bit integer representation of the MCR parameter (h000001 represents one cell per second), which is negotiated during call setup.

This field should be initialized by the host with the negotiated parameter value at connection setup.

3.2.1.1.3 Parameter Memory Word 2

Reserved (31:24)	ICR (23:0)

Bit(s) (31:24) **Reserved**

Bit(s) (23:0) **ICR (Initial Cell Rate)**
This field contains the 24-bit integer representation of the ICR parameter (h000001 represents one cell per second), which is negotiated during call setup. The host, after call setup, compares the negotiated value of ICR to TBE/FRTT. The minimum value then needs to be programmed by the host as ICR. TBE is a parameter negotiated during call setup and FRTT is a parameter accumulated during call setup.

This field should be initialized by the host, as described previously, at connection setup.

3.2.1.1.4 Parameter Memory Word 3

Reserved (31:24)	ACR (23:0)

Bit(s) (31:24) **Reserved**

Bit(s) (23:0) **ACR (Allowed Cell Rate)**
This field contains the 24-bit integer representation of the ACR parameter (h000001 represents one cell per second). This field is used by scheduler to schedule transmission of cells for a particular connection and can be adjusted by either the Source Processor or the Destination Processor, based on various different criteria. This field should never be written by the host except for initialization at connection setup.

This field should be initialized by the host to the negotiated value for ICR at connection setup.

3.2.1.1.5 Parameter Memory Word 4

ADTF_Ticks (31:0)

Bit(s) (31:0) **ADTF_Ticks**
This field contains the representation of the ADTF parameter, which is optionally negotiated during call setup. ADTF_Ticks = ADTF $\times$ PCI bus frequency, rounded up. The host may either use the value negotiated after call setup or the default value of 0.5 second for ADTF in doing the calculation. The current PCI bus frequency is 33 MHz.

This field should be initialized by the host (as described in the preceding paragraph) at power up/reset initialization time if the default ADTF value is used, or at connection setup if ADTF is negotiated.

3.2.1.1.6 Parameter Memory Word 5

Reserved (31:29)	ACR_Increase_Delta (28:24)	CDF (23:16)	RIF (15:8)	RDF (7:0)

Bit(s) (31:29) **Reserved**

Bit(s) (28:24) **ACR_Increase_Delta**
This field contains the 5-bit number used to calculate the Δ that is added to the ACR_Low_Threshold parameter used for determining if a connection's ACR has improved to an OK status (see Appendix D *Low ACR Detection*), i.e., ACR $\geq$ ACR_Low_Threshold + Δ, where $\Delta = 2^{ACR\_Increase\_Delta}$. For example, if ACR_Increase_Delta is 4, then Δ = 16 cells/seconds.

This field should be initialized by the host to the appropriate value at connection setup.

Bit(s) (23:16) **CDF**
This field contains the representation of CDF parameter. CDF = log2 (1/CDF_neg), where CDF_neg is optionally negotiated during call setup. The host may either use the value negotiated at call setup or the default value of 1/16 for CDF_neg.

This field should be initialized by the host (as described previously) at power up/reset initialization time if the default CDF_neg value is used, or at connection setup if CDF_neg is negotiated.

Bit(s) (15:8) **RIF**
This field contains the representation of the RIF parameter. RIF = log2 (1/RIF_neg) where RIF_neg is negotiated during call setup.

This field should be initialized by the host (as described in the preceding paragraph) at connection setup.

3.2.1.1.6 Parameter Memory Word 5 (continued)

Bit(s) (7:0) **RDF**

This field contains the representation of the RDF parameter. RDF = log2 (1/RDF_neg) where RDF_neg is negotiated during call setup.

This field should be initialized by the host (as described in the preceding paragraph) at connection setup.

3.2.1.1.7 Parameter Memory Word 6

Trm_Ticks (31:0)

Bit(s) (31:0) **Trm_Ticks**

This field contains the 32-bit representation of the Trm parameter. Trm_Ticks = Trm x PCI bus frequency rounded up, where Trm is equal to the default value of 100 ms. The current PCI bus frequency is 33 MHz.

This field should be initialized by the host (as described in the preceding paragraph) at power up/reset initialization time.

3.2.1.1.8 Parameter Memory Word 7

Crm (31:24)	Nrm (23:16)	Reserved (15:9)	Calling Party (8)	Service Type (7:0)

Bit(s) (31:24) **Crm**

This field contains the 8-bit representation of the Crm parameter. Crm = TBE/Nrm where TBE is the TBE parameter that is negotiated during call setup and Nrm is a constant (see the following Nrm definition).

This field should be initialized by the host (as described in the preceding paragraph) at connection setup.

Bit(s) (23:16) **Nrm**

This field contains the representation of the Nrm parameter, which is assigned the default value of 32.

This field should be initialized to 32 by the host at power up/reset initialization time.

Bit(s) (15:9) **Reserved**

Bit(s) (8) **Calling_Party**

This field is used by the host to indicate to the TNETA1585 which end system initiated the call. This supports the Calling-Party Distinction feature (see Appendix E). The host should set this bit if this end system initiated the call, otherwise, the host should clear this bit.

This field should be initialized (as described in the preceding paragraph) at connection setup.

Bit(s) (7:0) **Service Type**

This field is used by the host to indicate the service type configured for this connection, which determines the type of scheduling to be performed on the TX connection and the course of action to be performed on the RX connection. The following table lists the available options.

This field should be initialized appropriately at connection setup.

SERVICE TYPE	DESCRIPTION
0	VC-Level ABR
1	VP-Level ABR
2	VBR-nrt
3	PSR Drain
4–255	Reserved

3.2.1.1.9 Parameter Memory Word 8

Expected_VPI (31:20)	Expected_VCI (19:4)	Reserved (3:1)	Receive_On (0)

Bit(s) (31:20) **Expected_VPI**

This field contains the expected Virtual Path Identifier for cells that are being received on this channel, i.e., the VPI associated with the RX channel number. This parameter is used to support the Receive Synchronization Check feature described in Appendix F. If operating in UNI mode, i.e., the NNI_UNI bit in the TNETA1585 Configuration Register is set by the host, bits 31:28 of this field should be set to 0.

This field should be initialized by the host at connection setup.

Bit(s) (19:4) **Expected_VCI**

This field contains the expected Virtual Channel Identifier for cells that are being received on this channel, i.e., the VCI associated with the RX channel number. This parameter is used to support the Receive Synchronization Check feature described in Appendix F.

This field should be initialized by the host at connection setup.

Bit(s) (3:1) **Reserved**

Bit(s) (0) **Receive_On**

This bit is used to indicate that a particular channel number is configured in the device for receiving data. A set bit means that any cell received for that channel is to be processed by the TNETA1585 device. A clear bit means that any cells received for that channel is to be dropped by the TNETA1585 device.

This bit should be set by the host at connection setup and cleared by the host at connection teardown.

3.2.1.1.10 Parameter Memory Word 9

Reserved (31:8)	Polling_Rate_Coeff (7:0)

Bit(s) (31:8) **Reserved**

Bit(s) (7:0) **Polling_Rate_Coeff**

This field contains a byte-wide coefficient used to calculate the polling rate. The polling rate specifies the rate at which to poll for data availability when data becomes unavailable. The host would actually specify a one-byte shift coefficient to PCR for calculating the polling rate, such that:

$$\text{polling rate} = \text{PCR}\left(\frac{1}{2^{\text{Polling\_Rate\_Coeff}}}\right)$$

This field should be initialized by the host at connection setup.

3.2.1.1.11 Parameter Memory Word 15

ACR_Low_Threshold (31:16)	Reserved (15:12)	TX Channel Number (11:0)

Bit(s) (31:16) **ACR_Low_Threshold**

This field contains the parameter used to compare ACR to to determine if ACR has changed to a low status (see Appendix D Low ACR Detection). h000001 represents one cell per second. ACR_Low_Threshold must be programmed with a value greater than 10 cells per second.

This field should be initialized by the host to the appropriate value at connection setup.

Bit(s) (15:11) **Reserved**

3.2.1.1.11 Parameter Memory Word 15 (continued)

Bit(s) (10:0) **RX/TX Channel Cross Reference**

Service Type = VC-Level ABR:
This field is used to specify the TX Channel Number associated with a connection that a cell was received on. Since TX and RX channel numbers are identical for VC-Level ABR service type, this parameter serves no function for VC-Level ABR service type but is needed to support co-existence of this service type with VP-Level ABR service-type connections.

This field should be initialized by the host to the appropriate TX Channel Number at connection setup.

Service Type = VP-Level ABR:
This field is used to specify the TX Channel Number associated with a VP connection that a cell was received on, i.e., the RX Channel Number of a received cell is used to derive the base address used to locate in Parameter Memory the word (this word) that contains the TX Channel Number associated with the connection that a cell was received on.

This field should be initialized by the host to the appropriate TX Channel Number value at connection setup.

3.2.1.1.12 VC-Level ABR Parameter Memory Configuration Example

The following is an example of all parameters configured for a particular VC-Level ABR connection with a given channel number:

TX and RX Channel Number = 4

Expected VPI = 0

Expected VCI = 75

Base address = 4 $\times$ 16 = 64

PM ADDRESS	BITS	CONTENTS
64	23:0	PCR
65	23:0	MCR
66	23:0	ICR
67	23:0	ACR
68	31:0	ADTF_Ticks
69	28:0	ACR_Increase_Delta, CDF, RIF, RDF
70	31:0	Trm_Ticks
71	31:16, 8:0	Crm, Nrm, Calling_Party, Service_Type
72	31:20	0 (Expected VPI)
	19:4	75 (Expected VCI)
	0	1 (Receive_On)
73	7:0	Polling_Rate_Coeff
79	31:16	ACR_Low_Threshold
	10:0	4 (RX/TX Channel Cross Ref)

3.2.1.1.13 VP-Level ABR Parameter Memory Configuration Example

For VP-Level ABR connections, there is more than one RX channel number for one TX channel number associated with the connection. The following is an example of all parameters configured for one particular VP-Level ABR connection with the given channel numbers.

NOTE:

Each RX channel number is associated with a different VCI number, while the connection as a whole is associated with one VPI number.

TX Channel Number = 5 (TX channel used for VPI = 0:VCI = 75, VPI = 0:VCI = 76)

RX Channel Number = 5

Expected VPI = 0

Expected VCI = 75

RX Channel Number = 6

Expected VPI = 0

Expected VCI = 76

Parameters configured using the TX channel number:

TX base address = 5 $\times$ 16 = 80

PM ADDRESS	BITS	CONTENTS
80	23:0	PCR
81	23:0	MCR
82	23:0	ICR
83	23:0	ACR
84	31:0	ADTF_Ticks
85	28:0	ACR_Increase_Delta, CDF, RIF, RDF
86	31:0	Trm_Ticks
87	31:16, 8:0	Crm, Nrm, Calling_Party, Service_Type
89	7:0	Polling_Rate_Coeff
95	31:16	ACR_Low_Threshold

Parameters configured using the RX channel number 5:

RX base address = 5 $\times$ 16 = 80

Expected_VPI word location = 8 $\rightarrow$ 80 + 8 = 88

Expected_VCI word location = 8 $\rightarrow$ 80 + 8 = 88

Receive_On location = 8 $\rightarrow$ 80 + 8 = 88

RX/TX Channel Cross Reference word location = 15 $\rightarrow$ 80 + 15 = 95

PM ADDRESS	BITS	CONTENTS
88	31:20	0 (Expected VPI)
88	19:4	75 (Expected VCI)
88	0	1 (Receive_On)
95	10:0	5 (RX/TX Channel Cross Ref)

3.2.1.1.13 VP-Level ABR Parameter Memory Configuration Example (continued)

Parameters configured using the RX channel number 6:

RX base address = 6 × 16 = 96

Expected_VPI word location = 8 → 96 + 8 = 104

Expected_VCI word location = 8 → 96 + 8 = 104

Receive_On location = 8 → 96 + 8 = 104

RX/TX Channel Cross Reference word location = 15 → 96 + 15 = 111

PM ADDRESS	BITS	CONTENTS
104	31:20	0 (Expected VPI)
104	19:4	76 (Expected VCI)
104	0	1 (Receive_On)
111	10:0	5 (RX/TX Channel Cross Ref)

3.2.1.2 VBR-nrt Service-Type TX Connection Parameters

Table 3–6 lists all the parameters that define the state of a VBR-nrt service-type TX connection, along with their word locations within the connection state block and their bit locations. These parameters include VBR-nrt parameters defined in the ATM Forum TM 4.0 Specification.

Table 3–6. VBR-nrt Service-Type TX Connection State Parameters

WORD	BIT LOCATION(S)	DESCRIPTION
0	(23:0)	PCR
	(31:24)	Reserved
1	(23:0)	SCR
	(31:24)	Reserved
2	(23:0)	MBS
	(31:24)	Reserved
3–6	(31:0)	Reserved
7	(7:0)	Service Type
	(31:8)	Reserved
8–15	(31:0)	Reserved

The following provides a detailed description for each of the fields shown in Table 3–6.

3.2.1.2.1 Parameter Memory Word 0

Reserved (31:24)	PCR (23:0)

Bit(s) (31:24) **Reserved**

Bit(s) (23:0) **PCR**

This field contains the 24-bit integer representation of the PCR parameter (h000001 represents one cell per second) that is negotiated during call setup.

This field should be initialized by the host at connection setup.

3.2.1.2.2 Parameter Memory Word 1

Reserved (31:24)	SCR (23:0)

Bit(s) (31:24) **Reserved**

Bit(s) (23:0) **SCR**
This field contains the 24-bit integer representation of the SCR parameter (h000001 represents one cell per second) that is negotiated during call setup.

This field should be initialized by the host at connection setup.

3.2.1.2.3 Parameter Memory Word 2

Reserved (31:24)	MBS (23:0)

Bit(s) (31:24) **Reserved**

Bit(s) (23:0) **MBS**
This field contains the 24-bit integer representation of the MBS parameter (h000001 represents one cell) that is negotiated during call setup.

This field should be initialized by the host at connection setup.

3.2.1.2.4 Parameter Memory Word 7

Reserved (31:8)	Service Type (7:0)

Bit(s) (31:8) **Reserved**

Bit(s) (7:0) **Service Type**
This field is used by the host to indicate the service type configured for this connection that determines the type of scheduling to be performed. The following table lists the available options.

This field should be initialized appropriately at connection setup.

SERVICE TYPE	DESCRIPTION
0	VC-Level ABR
1	VP-Level ABR
2	VBR-nrt
3	PSR Drain
4–255	Reserved

3.2.1.2.5 VBR-nrt TX Connection Parameter Memory Configuration Example

The following is an example of all parameters that need to be configured for one VBR-nrt TX connection with a given channel number:

TX Channel Number = 4

Parameters configured using TX channel number 4:

TX base address = 4 × 16 = 64

PM ADDRESS	BITS	CONTENTS
64	23:0	PCR
65	23:0	SCR
66	23:0	MBS
71	8:0	2

3.2.1.3 PSR Drain Service-Type Parameters

Table 3–7 lists all the parameters that define the state of a PSR Drain service-type connection, along with their word locations within the connection state block and their bit locations. PSR Drain is done on the TX side only, and thus all parameters are configured using the associated TX channel number.

Table 3–7. PSR Drain Service-Type Connection State Parameters

WORD	BIT LOCATION(S)	DESCRIPTION
0	(23:0)	PSR Drain Rate
	(31:24)	Reserved
1–6	(31:0)	Reserved
7	(7:0)	Service Type
	(31:8)	Reserved
8–15	(31:0)	Reserved

The following provides a detailed description for each of the fields shown in Table 3–7.

3.2.1.3.1 Parameter Memory Word 0

Reserved (31:24)	PSR Drain Rate (23:0)

Bit(s) (31:24) **Reserved**

Bit(s) (23:0) **PSR Drain Rate**

This field contains the 24-bit integer representation of the PSR Drain rate (h000001 represents one cell per second) as specified by the host.

This field should be initialized by the host at connection setup.

3.2.1.3.2 Parameter Memory Word 7

Reserved (31:8)	Service Type (7:0)

Bit(s) (31:8) **Reserved**

Bit(s) (7:0) **Service Type**

This field is used by the host to indicate the service type configured for this connection that determines the type of scheduling to be performed. The following table lists the available options.

This field should be initialized appropriately at connection setup.

SERVICE TYPE	DESCRIPTION
0	VC-Level ABR
1	VP-Level ABR
2	VBR-nrt
3	PSR Drain
4–255	Reserved

3.2.1.3.3 PSR Drain Connection Parameter Memory Configuration Example

The following is an example of all parameters that need to be configured for one PSR Drain connection with a given channel number:

TX Channel Number = 4

Parameters configured using TX channel number 4:

TX base address = $4 \times 16 = 64$

PSR Drain Rate word location = $0 \rightarrow 64 + 0 = 64$

Service_Type word location = $7 \rightarrow 64 + 7 = 71$

PM ADDRESS	BITS	CONTENTS
64	23:0	PSR Drain Rate
71	8:0	3

3.2.2 Real-Time Variables (RTV) RAM

The internal Real-Time Variables (RTV) RAM is used to store real-time or processing intensive variables used by the TNETA1585. There are four such variables per channel stored in RTV RAM. Each variable is one bit long, for a total RTV RAM size of 2K × 4 bit locations. The RTV RAM mapping is shown in Table 3–8.

Table 3–8. RTV RAM Individual Address Map

ADDRESS	CHANNEL NUMBER
h140000	0
h140001	1
h140002	2
h140003	3
—	—
—	—
h1407FD	2045
h1407FE	2046
h1407FF	2047

Each word in Table 3–8 contains four variables. Table 3–9 shows the RTV RAM word as seen by the host.

Table 3–9. RTV RAM Individual Word Fields

RESERVED	BRM_Flag	RESERVED	Close_ Channel_ Pending	RESERVED	Scheduler_ Paused	RESERVED	Data_ Available
(31:25)	(24)	(23:17)	(16)	(15:9)	(8)	(7:1)	(0)

The following provides a description for each of the fields given in Table 3–9.

Bit(s) (31:25) **Reserved**

Bit(s) (24) **BRM_Flag**
This bit indicates if there is a received forward RM cell waiting to be turned around. A set bit indicates there is a received forward RM cell waiting.

This field should be initialized to zero by the host after connection setup.

Bit(s) (23:17) **Reserved**

Bit(s) (16) **Close_Channel_Pending**
This bit indicates if a channel has been requested by the host to be torn down. A set bit indicates that the connection is waiting to be torn down.

This field should be initialized to zero by the host after connection setup.

Bit(s) (15:9) **Reserved**

Bit(s) (8) **Scheduler_Paused**
This bit indicates that the channel has been placed in sleep mode. A set bit indicates that the channel is in sleep mode.

This field should be initialized to zero by the host after connection setup.

Bit(s) (7:1) **Reserved**

Bit(s) (0) **Data_Available**
This bit indicates if there is data available for the channel to transmit. A set bit indicates that there is data available.

This field should be initialized to zero by the host after connection setup.

3.2.3 Instruction RAM

The Instruction RAM is used to store processor code (provided by Texas Instruments) used by the TNETA1585 on-chip processors to implement this particular application. The code is downloaded by the host after power up/reset (see Section 4.1). Table 3–10 lists the address range of the Instruction RAM.

Table 3–10. Instruction RAM Address Map

INSTRUCTION RAM ADDRESS RANGE	INSTRUCTION RAM CONTENTS
h120000–1203FF	Downloaded Code

Once the code is downloaded, the locations listed in Table 3–11 of the Instruction RAM contains information regarding the particular revision of code that has been downloaded.

Table 3–11. Instruction Code Revision Information

INSTRUCTION RAM ADDRESS	INSTRUCTION RAM CONTENTS	VALUE
h120002	Software Revision I.D.	X†
h120003	Traffic Management Implementation	1‡

† This number reflects the latest revision of the TNETA1585 device.

‡ This value represents traffic management implementation as described in the ATM Forum TM 4.0 Specification, document number af-tm-00056.0000 April 1996.

3.2.4 Zero RAM 1 and Zero RAM 2

The two internal Zero RAMs store real-time or processing-intensive variables used by the TNETA1585. The Zero RAMs are accessible by the host for initialization ONLY. The host is required to initialize all 32-bit words in these RAMs at power up/reset time.

4. CONFIGURATION PROCEDURES

This section details the sequence of actions that the host must execute within each one of three configuration procedures:

1. Power Up/Reset Initialization
2. Connection Setup
3. Connection Teardown

Power Up/Reset Initialization is required and is the same, regardless of service type and options implemented. Connection Setup and Teardown details the required configuration for VC-Level ABR service-type connections, while the attached appendices provide the details for configuration required by other service types and optional features.

NOTE:

The TNETA1575 SAR, if used, must be configured such that it is aware that the TNETA1585 is being used with it. The COP_PRES bit must be set in the SAR configuration register indicating that the TNETA1585 is present. This enables the interface between the TNETA1575 and the TNETA1585. This should be done before or during TNETA1585 initialization.

4.1 Power Up or Reset Initialization

At power up/reset, the host shall perform the following steps to initialize the TNETA1585 device. The TNETA1585 cannot make any connections until initialization is complete. Initialization is complete after all the initialization steps are performed and the host has verified that the operating code was loaded correctly.

The initialization steps are summarized in the following, including paragraphs describing the details of each step.

1. Reset Device
2. Download Code to Instruction RAM
3. Enable CRC Checking
4. Clear Parameter Memory
5. Clear Zero RAMs
6. Verify Operating Code Loaded Correctly
7. Initialize Clock Frequency Register
8. Initialize NNI_UNI Bit
9. Enable Destination Processor
10. Enable Source Processor
11. Enable Scheduler
12. Enable Receive UTOPIA Interface
13. Configure Interrupt Mask Register

4.1.1 Reset Device

The host should perform a global software reset to the device prior to initialization.

Configuration Register

Bit(s) (6) 1 – Reset the entire TNETA1585 device

4.1.2 Download Code to Instruction RAM

The host must load the operating code (provided by Texas Instruments) into the Instruction RAM at initialization. The Source and Destination Processors must not be enabled when this is done.

NOTE:

The host has indirect access to the Instruction RAM and must use the Write Data and Write Address Registers for downloading code.

The loading of the code is performed by first setting the Write Address Register to point to the start of the Instruction RAM (see Table 3–2), after verifying that the Write_Ready bit in the Status Register is set. Then the first word of code should be written to the Write Data Register. After writing the first word of code to the Write Data Register, the host should wait until the Write_Ready bit in the Status register is set, after which it should write the second word of code into the Write Data Register. That procedure should be repeated for all words that need to be written to the Instruction RAM. The contents of the Write Address Register is automatically incremented after each write operation so that an address need not be written to the Write Address Register for each data written to the Write Data Register except for the first word of data (see the indirect write procedure in Section 3.2).

NOTE:

Since the Source and Destination Processors are disabled, write accesses to the Instruction RAM are more deterministic. Thus, checking of the Write_Ready bit may be omitted (except for the first word of data written) if the latencies of Instruction RAM writes are long enough to allow the Write_Ready bit to become valid by the time the next write is initiated. Thus, after the first address is loaded to the Write Address Register, the Write Data Register can be sequentially written until the entire code is loaded without having to check the Write_Ready bit.

4.1.3 Enable CRC Checking

The host must invoke CRC checking once the code has been downloaded to the Instruction RAMs.

Configuration Register

Bit(s) (8:7) 1,1 – Invoke CRC checking

4.1.4 Clear Parameter Memory

All locations in Parameter Memory need to be cleared (set to zero) at initialization.

NOTE:

The host has indirect access to Parameter Memory and must use the Write Data and Write Address Registers for initialization.

To configure the Parameter Memory, the address pointing to the start of Parameter Memory (see Table 3–2) should be written to the Write Address Register, after verifying that the Write_Ready bit in the Status Register is set. Then the first word of data should be written to the Write Data Register. After writing the first word of data to the Write Data Register, the host should wait until the Write_Ready bit in the Status register is set, after which it should write the second word of data into the Write Data Register. This procedure should be repeated until the entire Parameter Memory is cleared. The contents of the Write Address Register is automatically incremented after each write operation so that an address need not be written to the Write Address Register for each data written to the Write Data Register except for the first word of data (see the indirect write procedure in Section 3.2).

NOTE:

Since the Source and Destination Processors and the Scheduler are disabled, write accesses to Parameter Memory are more deterministic. Thus, checking of the Write_Ready bit may be omitted (except for the first word of data written) if the latencies of Parameter Memory writes are long enough to allow the Write_Ready bit to become valid by the time the next write is initiated. Thus, after the first address is loaded to the Write Address Register, the Write Data Register can be sequentially written until the entire Parameter Memory is configured without having to check the Write_Ready bit.

4.1.5 Clear Zero RAMs

All locations of both Zero RAMs should be cleared (set to zero) at initialization.

NOTE:

The host has indirect access to the Zero RAMs and must use the Write Data and Write Address Registers for initialization.

To clear all locations, the address pointing to the start of a Zero RAM (see Table 3–2) should be written to the Write Address Register, after verifying that the Write_Ready bit in the Status Register is set. Then the first word of data should be written to the Write Data Register. After writing the first word of data to the Write Data Register, the host should wait until the Write_Ready bit in the Status register is set, after which it should write the second word of data into the Write Data Register. This procedure should be repeated until all of Zero RAM is cleared. The contents of the Write Address Register is automatically incremented after each write operation so that an address need not be written to the Write Address Register for each data written to the Write Data Register except for the first word of data (see the indirect write procedure in Section 3.2).

NOTE:

Since the Source and Destination Processors are disabled, write accesses to on-chip RAMs are more deterministic. Thus, checking of the Write_Ready bit may be omitted (except for the first word of data written) if the latencies of Zero RAM writes are long enough to allow the Write_Ready bit to become valid by the time the next write is initiated. Thus, after the first address is loaded to the Write Address Register, the Write Data Register can be sequentially written until the entire Zero RAM is configured without having to check the Write_Ready bit.

4.1.6 Verify Operating Code Loaded Correctly

To verify that the operating code was downloaded correctly to the instruction RAMs, the host must first verify that CRC checking has completed. If CRC checking has not yet completed, the host must continuously poll the Status Register until it indicates that CRC checking has completed. Then the host can proceed to verify the code was downloaded correctly.

Status Register

Bit(s) 7 1 – CRC checking is in progress

0 – CRC checking has completed

Status Register

Bit(s) 0 1 – Code was loaded incorrectly

0 – Code was loaded correctly

4.1.7 Initialize Clock-Frequency Register

The Clock-Frequency Register must be configured by the host at initialization if not operating at 33 MHz. The register is configured by hardware to indicate 33 MHz operation after power up/reset. Refer to Section 3.1.10 to calculate appropriate configuration values for other frequency operations.

Clock-Frequency Register Default Value

Bit(s) (31:21) Set to all zeros

Bit(s) (20:16) 18H

Bit(s) (15:0) FCA1H

4.1.8 Initialize NNI_UNI Bit

The host must initialize the NNI_UNI bit in the Configuration Register to the appropriate value at initialization.

Configuration Register

Bit(s) (5) 1 – UNI mode of operation

0 – NNI mode of operation

4.1.9 Enable Destination Processor

The host shall then enable the Destination Processor after completing the previous steps. It shall do this by setting bit 2 of the Configuration Register. The host may also enable the Source Processor and the Scheduler simultaneously with the Destination Processor.

Configuration Register

Bit(s) (2) 1 – Enable Destination Processor

0 – Disable Destination Processor

4.1.10 Enable Source Processor

The host shall then enable the Source Processor after completing the previous steps. It shall do this by setting bit 1 of the Configuration Register. The host may also enable the Destination Processor and the Scheduler simultaneously with the Source Processor.

Configuration Register

Bit(s) (1) 1 – Enable Source Processor

0 – Disable Source Processor

4.1.11 Enable Scheduler

The host shall then enable the Scheduler after completing the previous steps. It shall do this by setting bit 0 of the Configuration Register. The host may also enable the Source Processor and the Destination Processor simultaneously with the Scheduler.

Configuration Register

Bit(s) (0) 1 – Enable Scheduler

0 – Disable Scheduler

4.1.12 Enable Receive UTOPIA Interface

The host shall then enable the receive UTOPIA interface after completing the previous steps. It shall do this by setting bit 9 of the Configuration Register.

Configuration Register

Bit(s) (9) 1 – Enable Receive UTOPIA Interface

0 – Disable Receive UTOPIA Interface

4.1.13 Configure Interrupt Mask Register

The host should configure this register to enable desired interrupts caused by the conditions given in the Status Register. This register has a bit-to-bit correspondence to the Status Register. To enable an interrupt, the host should set the bit in the Interrupt Mask Register that corresponds to the bit in the Status Register for that interrupt.

Bit(s) (31:0) Interrupt Mask

4.2 Power Up/Reset Initialization Summary

Table 4–1 summarizes the register accesses required to perform initialization.

Table 4–1. Initialization Register Accesses

DEFINITION	CONTROL MEMORY ADDRESS	PCI OFFSET ADDRESS	WRITE	READ
Reset Device	h10008	h40020	h00000040	—
Download code to Instruction RAM			See Section 4.1.2	See Section 4.1.2
Enable CRC Checking	h10008	h40020	h00000180	—
Clear Parameter Memory			See Section 4.1.4	See Section 4.1.4
Clear Zero RAMs			See Section 4.1.5	See Section 4.1.5
Verify Operating Code loaded correctly	h10009	hh40024	See Section 4.1.6	—
Initialize Clock Frequency Register	h10011	h40044	See Section 4.1.7	—
Initialize NNI_UNI Bit	h10008	h40020	See Section 4.1.8	—
Enable Destination Processor†	h10008	h40020	h00000004	—
Enable Source Processor†	h10008	h40020	h00000002	—
Enable Scheduler†	h10008	h40020	h00000001	—
Enable Receive UTOPIA Interface‡	h10008	h40020	h00000200	—
Configure Interrupt Mask Register	h1000A	h40028	See Section 4.1.13	—

† The Destination and Source Processors and the Scheduler can be enabled simultaneously by writing a h00000007 to control memory address h10008 (PCI offset address h40020). If enabling individually, the values must be ORed with current register value.

‡ Enable value must be ORed with current configuration register value.

4.3 Connection Setup

Connection setup of the TNETA1585 must be done as shown in the following sequence, with respect to the TNETA1575 connection setup procedure. As shown, the whole TNETA1585 connection setup procedure is performed in the middle of the TNETA1575 connection setup procedure. This sequence applies, regardless of whether the connection is a Permanent Virtual Connection or a Switched Virtual Connection. For Switched Virtual Connections, TNETA1575 and TNETA1585 configuration is initiated by the calling party as soon as the calling party receives the CONNECT message from the network. However, configuration of the TNETA1575 and TNETA1585 must be completed by the called party before the called party can send the CONNECT message to the network back to the calling party.

TNETA1575 Configure TX DMA State Table // see *TNETA1575 Programmer's Reference Guide*

TNETA1585 Connection Setup **// configure TNETA1585**

TNETA1575 Configure RX DMA State Table // see *TNETA1575 Programmer's Reference Guide*

TNETA1575 Write to TX Queue Register // see *TNETA1575 Programmer's Reference Guide*

NOTE:

The host may queue data in memory before performing the TNETA1585 Connection Setup. However, the host must not write to the TNETA1575 TX Queue Register until after completing the TNETA1585 Connection Setup procedures.

4.3.1 VC-Level ABR Service Type

TNETA1585 connection setup for VC-Level ABR service type consists of the following steps. The details of each step are described in the following sections. The procedure for the other service types (VP-Level ABR, VBR-nrt, and PSR Drain) connection setups are outlined in the appropriate appendices.

Before either the TNETA1575 or TNETA1585 are configured, the host must assign channel numbers to the connection's VPI/VCI pair on the transmit and receive sides of the connection. For VC-Level ABR service-type connections, the host must assign the same number to both the TX channel number and the RX channel number. The one TX/RX channel number is then used to configure the TNETA1575 and TNETA1585 data structures.

Step 1. Configure Parameter Memory
Step 2. Configure RTV RAM
Step 3. Enable Scheduling for Channel

4.3.1.1 Configure Parameter Memory

Table 4–2 lists the minimum set of parameters that require configuration for a new VC-Level ABR connection. They do not include parameters used for Low ACR Detection and Receive Synchronization Check features. Those are described in Appendices D and F. Each parameter must be configured according to the procedure outlined in the referenced sections or as shown in the table. Table 4–2 also lists 32-bit reserved words that require initialization at connection setup. These words are used for temporary variables storage. Those 32-bit reserved words not listed in Table 4–2 must not be modified.

Any reserved bits that are part of a word containing a valid parameter, e.g., bits 31:24 of word 0 (PCR) of the VC-Level ABR connection state, must be cleared when that word is written in Parameter Memory. For example, if PCR = 125 cells/second, then word 0 must be written as 0000007Dh.

Table 4–2. VC-Level ABR Parameter Memory Connection Setup

DESCRIPTION	WORD	SECTION NUMBER
PCR	0	3.2.1.1.1
MCR	1	3.2.1.1.2
ICR	2	3.2.1.1.3
ACR	3	3.2.1.1.4
ADTF_Ticks	4	3.2.1.1.5
RDF	5	3.2.1.1.6
RIF	5	3.2.1.1.6
CDF	5	3.2.1.1.6
Trm_Ticks	6	3.2.1.1.7
Service Type	7	3.2.1.1.8
Calling Party	7	3.2.1.1.8
Nrm	7	3.2.1.1.8
Crm	7	3.2.1.1.8
Receive_On	8	Receive_On = 1
Polling_Rate_Coeff	9	3.2.1.1.10
Reserved	10	Clear (All 0s)
RX/TX Channel Cross Reference	15	3.2.1.1.11

NOTE:
The host has indirect access to Parameter Memory and must use the Write Data and Write Address Registers for initialization accesses.

4.3.1.2 Configure RTV RAM

The RTV RAM word associated with the channel number of the new connection must be cleared at connection setup time. The host should use the following RTV RAM address map.

ADDRESS	CHANNEL NUMBER
h140000	0
h140001	1
h140002	2
h140003	3
—	—
—	—
h1407FD	2045
h1407FE	2046
h1407FF	2047

NOTE:

The host has indirect access to RTV RAM and must use the Write Data and Write Address Registers for initialization accesses.

4.3.1.3 Enable Scheduling for Connection

After all the previous steps have been executed, the host must turn on scheduling for the connection.

Part 1

The host must poll the Scheduler_On_Ready bit in the Status Register to verify that the scheduler is ready to add a new connection.

Status Register

Bit(s) (4) 1 – Scheduler is ready to accept new connection request

0 – Scheduler is not ready to accept new connection request

Part 2

The host shall then write the channel number of the connection to be added to the Scheduler On Register.

Scheduler On Register

Bit(s) (10:0) Channel number

4.4 Connection Teardown

Connection teardown of the TNETA1585 must be done as shown in the following sequence, with respect to the TNETA1575 connection teardown procedure.

TNETA1575 Write to Add/Delete Register // see *TNETA1575 Programmer's Reference Guide*

TNETA1585 Connection Teardown **// configure TNETA1585 for teardown**

If the associated TNETA1575 Packet Segmentation Rings (PSR) are not empty by the time the host wants to teardown a TX connection, the host may go into PSR Drain mode to flush the ring. However, before going into PSR Drain mode, the host must teardown the connection through the network and in the TNETA1575 first.

4.4.1 VC-Level ABR Service Type

The TNETA1585 connection teardown procedure for VC-Level ABR service type varies depending on whether PSR Drain mode is used after connection teardown. The procedure for each is described in the following steps. For both, Steps 1 and 2 must be done immediately after the host writes to the TNETA1575 Add/Delete Register. The procedure for the other service types (VP-Level ABR, VBR-nrt, and PSR Drain) connection teardown are outlined in the appropriate appendices.

Without PSR Drain:
Step 1. Disable scheduling for connection
Step 2. Clear Receive_On bit in Parameter Memory
Step 3. Verify channel number ready to use

Step 3 can be done any time between Step 2 to the time a new connection is setup using that same channel number.

With PSR Drain:
Step 1. Disable scheduling for connection
Step 2. Clear Receive_On bit in Parameter Memory
Step 3. Flush PSR (see Appendix C)

4.4.1.1 Disable Scheduling for Connection

To disable scheduling for a connection, the host must perform the following steps:

Part 1

The host must poll the Scheduler_Off_Ready bit in the Status Register to verify that the scheduler is ready to remove a connection.

Status Register

Bit(s) (3) 1 – Scheduler is ready to remove a connection

0 – Scheduler is not ready to remove a connection

Part 2

The host shall then write the channel number of the connection to be torn down to the Scheduler Off Register.

Scheduler Off Register

Bit(s) (10:0) Channel number

4.4.1.2 Clear Receive_On Bit in Parameter Memory

The Receive_On bit must be cleared for this channel in Parameter Memory. The Receive_On bit is located in word 8 bit 0 of Parameter Memory.

4.4.1.3 Verify Channel Number Ready to Use for Next Connection

The host must verify that all data structures associated with a connection (which in turn is associated with a channel number) that have been torn down are freed up and are ready to be used for a new connection.

NOTE:

This step can be done any time between the time the connection of a given channel number is torn down to the time a new connection is set up using that same channel number.

The host must check the Close_Channel_Pending bit in the RTV RAM to verify that a particular channel number can be used for a new connection. However, there is period of time after issuing the schedule off request before the Close_Channel_Pending bit becomes valid. Thus, the host must wait a minimum of 1 second before checking the Close_Channel_Pending bit.

RTV RAM (Close_Channel_Pending bit)

Bit(s) (16) 1 – Channel number is not ready to be used for new connection

0 – Channel number is ready to be used for new connection

4.4.1.4 Flush PSR

If the associated PSRs are not empty by the time the host wants to tear down a TX connection, the host can go into PSR Drain mode to flush the ring, but only after tearing down the connection first through the network and in the TNETA1575. The details of PSR Drain mode are provided in Appendix C.

5. INTERRUPT HANDLING

Interrupts are generated when a bit is set in the TNETA1585 Status Register and the corresponding bit in the Interrupt Mask Register is set. When an interrupt occurs, the host should read the Status Register to determine the cause of the interrupt before proceeding to handle the interrupt.

Several of the bits in the Status and Interrupt Mask Registers pertain to error conditions, while others pertain to readiness of certain data structures or completion of certain operations. Table 5–1 summarizes all the possible interrupts and their bit position in the Status and Interrupt Mask Registers, followed by sections giving detailed descriptions.

Table 5–1. Summary of Interrupts

STATUS REGISTER BIT LOCATION	STATUS NAME
31	Synchronization Error
30	PSR Drain Complete
6	ACR_is_Low
5	ACR_is_OK
4	Scheduler_On_Ready
3	Scheduler_Off_Ready
2	Write_Ready
1	Read_Data_Ready
0	Code_Invalid

5.1 Synchronization Error (bit 31)

This interrupt supports the Receive Synchronization Check feature described in Appendix F. The host should set this bit in the Interrupt Mask Register if it wants to be notified if the receive circuitry of the TNETA1585 and SAR become misaligned with each other. This should never occur. If this occurs, the host must tear down all connections and reset the device.

5.2 PSR Drain Complete (bit 30)

This interrupt supports the PSR Drain service type described in Appendix C. The host should set this bit in the Interrupt Mask Register if it wants to be notified when PSR Drain operation on a channel has completed and that all the data for the connection has been transmitted. This also means that the connection is completely torn down and the host can use the channel number associated with that connection for a new connection. It is recommended that the host perform PSR Drain operation on only one channel at a time and that it maintain information on that channel, since no information is provided to determine which channel has completed PSR Drain operation and caused this interrupt.

5.3 ACR_is_Low (bit 6)

This bit supports the Low ACR Detection feature (see Appendix D). The host should set this bit in the Interrupt Mask Register if it wants to be notified via interrupt that a channel has had its ACR reduced below a threshold that the host has previously specified. Upon receiving this interrupt, the host should check the ACR Low Register to determine which channel(s) is(are) affected. Alternatively, the host may mask this interrupt at any time if it instead plans to poll this status bit periodically.

5.4 ACR_is_OK (bit 5)

This bit supports the Low ACR Detection feature (see Appendix D). The host should set this bit in the Interrupt Mask Register if it wants to be notified via interrupt that a channel has had its ACR increase above a threshold that the host has previously specified indicating a previously low connection has returned to OK status. Upon receiving this interrupt, the host should check the ACR OK Register to determine which channel(s) is(are) affected. Alternatively, the host may mask this interrupt at any time if instead it plans to poll this status bit periodically.

5.5 Scheduler_On_Ready (bit 4)

The host should set this bit in the Interrupt Mask Register if it wants to be notified via interrupt that a previously requested Scheduler On operation has completed. The host can then proceed to make additional requests to turn on scheduling for a channel by writing the channel number to the Schedule On Register. Alternatively, the host may mask this interrupt at any time if, instead, it plans to poll this status bit periodically or just prior to making a scheduler on request.

5.6 Scheduler_Off_Ready (bit 3)

The host should set this bit in the Interrupt Mask Register if it wants to be notified via interrupt that a previously requested Scheduler Off operation has been completed. The host can then make additional requests to turn off scheduling for a channel by writing the channel number to the Schedule Off Register. Alternatively, the host may mask this interrupt at any time if, instead, it plans to poll this status bit periodically or just prior to making a scheduler off request.

5.7 Write_Ready (bit 2)

This bit supports writing to the indirectly accessible data structures of the TNETA1585 device. The host should set this bit in the Interrupt Mask Register if it wants to be notified via interrupt that the Write Address and Write Data Registers are ready for the next write operation. The host can then proceed to initiate additional write operations using the Write Address and Write Data Registers. Alternatively, the host may mask this interrupt at any time if instead it plans to poll this status bit periodically or just prior to initiating a write operation.

5.8 Read_Data_Ready (bit 1)

This bit supports reading from the indirectly accessible data structures of the TNETA1585 device. The host should set this bit in the Interrupt Mask Register if it wants to be notified via interrupt that requested data is valid in the Read Data Register. The host can then proceed to read the data from the Read Data Register and initiate additional read operations using the Read Address Register. Alternatively, the host may mask this interrupt at any time if it instead plans to poll this status bit periodically.

5.9 Code_Invalid (bit 0)

The host may set this bit in the Interrupt Mask Register to enable interrupts due to an error in CRC checks performed by both the Source and Destination Processors on code that has been downloaded by the host.

Appendix A: VP-Level ABR Service Type

VP-Level ABR is different from VC-Level ABR in that multiple channels (VCs) within one VP are grouped together and treated as one channel with respect to flow control. One RM-cell represents the group as a whole and regulates the aggregate rate for the group. Since the collection of channels is treated as one pool for transmit data, the group is identified by one TX channel number. However, since cells received are associated with unique VCI numbers, the corresponding RX channel numbers remain unique. Thus, there is a one-to-many relationship between TX channel numbers and RX channel numbers for a VP-Level ABR service-type connection.

A. Connection Setup

When the host selects the TX channel number for the TX side of the connection and the RX channel numbers for each of the VCIs in the connection, the channel number assigned to the TX channel number by the host should be assigned to one of the RX channel numbers. For example, if the TX channel number = 15, and there are three VCIs, one of the VCIs should be assigned an RX channel number of 15. If the TNETA1585 is not configured for any VBR-nrt service-type connections, then this restriction may not apply.

Refer to Section 4.3.1 for the connection setup procedures. Section 4.3.1 describes connection setup for VC-Level ABR connections and are similar to what is done for VP-Level ABR connections.

1. **Step 1: Configure Parameter Memory**

   The host will need to make a distinction between parameters associated with the TX side of the connection and parameters associated with the RX side of the connection when configuring Parameter Memory.

   - **TX Parameters**

   The host must use Table 3–4 *ABR Service-Type TX Connection State Parameters* to configure parameters associated with the TX side of the connection. The host must use the TX channel number associated with the connection to configure the appropriate connection state. The Table A–1 below is a subset of Table 3–4 and lists the minimum set of the parameters that require configuration for a new VP-Level ABR TX side connection. They do not include parameters used for the Low ACR feature. Those are described in Appendix D. Each parameter must be configured according to the procedure outlined in the referenced sections.

1. Step 1: Configure Parameter Memory (continued)

Table A–1. VP-Level ABR TX Connection Parameter Memory Setup

DESCRIPTION	WORD	SECTION NUMBER
PCR	0	3.2.1.1.1
MCR	1	3.2.1.1.2
ICR	2	3.2.1.1.3
ACR	3	3.2.1.1.4
ADTF_Ticks	4	3.2.1.1.5
RDF	5	3.2.1.1.6
RIF	5	3.2.1.1.6
CDF	5	3.2.1.1.6
Trm_Ticks	6	3.2.1.1.7
Service Type	7	3.2.1.1.8
Calling Party	7	3.2.1.1.8
Nrm	7	3.2.1.1.8
Crm	7	3.2.1.1.8
Polling_Rate_Coeff	9	3.2.1.1.10
Reserved	10	Clear (All 0s)

- **RX Parameters**

The host must use Table 3–5 *ABR Service-Type RX Connection State Parameters* to configure parameters associated with the RX side of the connection. This must be done for each RX channel number associated with the connection, since there may be more than one RX channel associated with a VP-Level ABR connection. Table A–2 is a subset of Table 3–5 and lists the minimum set of parameters that require configuration for a new VP-Level ABR RX connection. They do not include parameters used for the Receive Synchronization Check feature. Those are described in the Appendix F. Each parameter must be configured according to the procedure outlined in the referenced sections.

Table A–2. VP-Level ABR RX Connection Parameter Memory Setup

DESCRIPTION	WORD	SECTION NUMBER
Receive_On	8	Receive_On = 1
RX/TX Channel Cross Reference	15	3.2.1.1.11

Table A–1 also lists 32-bit reserved words that require initialization at connection setup. These words are used for temporary variables storage. These 32-bit reserved words not listed in Table A–1 must not be modified.

Any reserved bits that are part of a word containing a valid parameter, e.g., bits 31:24 of word 0 of the VP-Level ABR connection state, must be cleared when that word is written in Parameter Memory. For example, if PCR = 125 cells/second, then word 0 must be written as 0000007Dh.

2. Step 2: Configure RTV RAM

The host must use the TX channel number to configure RTV RAM (see Section 4.3.1.2).

3. Step 3: Enable Scheduling for Channel

The host must use the TX channel number when enabling scheduling for the channel (see Section 4.3.1.3).

B. Connection Teardown

Teardown procedure for a VP-Level connection is similar to that of VC-Level ABR connections. However, with VP-Level ABR, the connection is torn down only when the TX side of the connection is torn down. An RX connection that is a member of the VP group can be removed from the VP group without having to teardown the connection but must still follow a procedure for removing it out of the VP group. Both VP-Level teardown and RX deletion procedures are described below.

VP-Level Connection Teardown Procedures

1. Disable scheduling for connection
   The host must specify the TX channel number when disabling scheduling for the connection (see Section 4.4.1.1).
2. Clear Receive_On bit in Parameter Memory
   The host must use the RX channel number to clear the Receive_On bit in Parameter Memory for each RX channel that is a membler of the VP connection (see Section 4.4.1.2).
3. Verify channel number ready to use
   If the host does not need to flush its PSR, then it can proceed to verify that the TX channel number is ready to use for the next connection (see Section 4.4.1.3). This step can be done anytime between Step 2 above to the time a new connection is set up using the same channel number.
4. Flush PSR
   If the host needs to flush its PSR, then this step is taken after Step 2 above (see Section 4.4.1.4).

VP-Level RX Deletion Procedures

1. Clear Receive_On bit in Parameter Memory
   The host must use the RX channel number to clear the Receive_On bit in Parameter Memory for each RX channel of the VP connection being deleted from the VP group (see Section 4.4.1.2).

Appendix B: VBR-nrt Service Type

The TNETA1585 implements the service-type VBR-nrt as specified in the ATM Forum Traffic Management 4.0 Specification. This implementation will always meet the VBR-nrt conformance definition.

During VBR-nrt operation, scheduling will be performed at SCR as long as data is available. When data is no longer available, credits will be accumulated, up to the Maximum Burst Size (MBS). When data subsequently becomes available, scheduling will be performed at the Peak Cell Rate, for up to MBS cells, after which the scheduling rate will be reduced back to SCR.

A. Connection Setup

VBR-nrt connections are set up as half-duplex connections, i.e., TX and RX connections are independent of each other. Both TX and RX connections need to be set up in the SAR, however, connection setup and teardown procedures are required only for the TX connections and are not required for RX connections in the TNETA1585 device.

B. Connection Setup for TX Connections

Refer to Section 4.3.1 for the connection setup procedures. Section 4.3.1 describes connection setup for VC-Level ABR connections and are similar to what is done for VBR-nrt TX connections.

1. **Step 1: Configure Parameter Memory**

   The host must use Table 3–6 *VBR-nrt Service-Type TX Connection State Parameters* to configure parameters associated with a VBR-nrt TX connection. The host must use the TX channel number associated with the connection to configure the appropriate connection state. Table B–1 is a subset of Table 3–6 and lists the parameters that require configuration for a new VBR-nrt TX connection. Table B–1 also lists 32-bit reserved words that require initialization at connection setup. These words are used for temporary variables storage. Those 32-bit reserved words not listed in Table B–1 must not be modified.

   Any reserved bits that are part of a word containing a valid parameter, e.g., bits 31:24 of word 0 of the VBR-nrt TX connection state, must be cleared when that word is written in Parameter Memory. For example, if PCR = 125 cells/second, then word 0 must be written as 0000007Dh.

1. Step 1: Configure Parameter Memory (continued)

Table B–1. VBR-nrt TX Connection Parameter Memory Setup

DESCRIPTION	WORD	SECTION NUMBER
PCR	0	3.2.1.2.1
SCR	1	3.2.1.2.2
MBS	2	3.2.1.2.3
Reserved	3–4	Clear (All 0s)
Service Type	7	3.2.1.2.4

2. Step 2: Configure RTV RAM

The host must use the TX channel number to configure RTV RAM (see Section 4.3.1.2).

3. Step 3: Enable Scheduling for Channel

The host must use the TX channel number when enabling scheduling for the channel (see Section 4.3.1.3).

C. Connection Teardown for TX Connections

Connection teardown applies only to VBR-nrt TX connection in the TNETA1585 device. Connection teardown for VBR-nrt TX connections is similar to that of VC-Level ABR connections.

1. Disable scheduling for connection
   The host must specify the TX channel number when disabling scheduling for the connection (see Section 4.4.1.1).
2. Verify channel number ready to use
   If the host does not need to flush its PSR, then it can proceed to verify that the channel number is ready to use for the next connection (see Section 4.4.1.3). This step can be done anytime between Step 1 above to the time a new connection is set up using the same channel number.
3. Flush PSR
   If the host needs to flush its PSR, then this step is taken after Step 1 above and Step 2 above is skipped (see Section 4.4.1.4).

Appendix C: PSR Drain Service Type

The PSR Drain service-type connection is used to flush TX buffers used by a previous TX connection that has been torn down while there was still data in its TX buffers. During PSR Drain operation, the rate of the connection is specified and adjustable by the host and is not subject to adjustment by the Source and Destination processors in the TNETA1585. In this configuration, the connection is scheduled/rescheduled at the host-specified rate until no more data is available, at which time an interrupt is issued by the TNETA1585 to the host, indicating the end of PSR Drain processing. All cells sent during PSR Drain will have their Cell Loss Priority (CLP) bit set, giving it a lower priority through the network than data cells of other service types. Since the interrupt provides no information on which channel finished PSR Drain processing, it is highly recommended that the host process PSR Drain on one channel at a time and that it keep track of which channel is being processed.

The host must complete connection teardown procedures through the network and with the TNETA1575 for the ABR connection before initiating PSR Drain operation. The host must also use the TX channel number of the previous connection for the PSR Drain connection TX channel number. PSR Drain only requires a TX channel number since it is a TX-only connection.

A. Enable PSR Drain Interrupt

Before setting up the connection for PSR Drain, the host must configure the Interrupt Mask Register to enable an interrupt due to completion of PSR Drain operation. This step can be done instead after power up/reset if the host intends to use the PSR Drain feature all the time, in which case, it need not be done here again.

Interrupt Mask Register

Bit(s) (30) 1 – Enable PSR Drain Complete interrupt

0 – Disable PSR Drain Complete interrupt

B. Configure Parameter Memory

The host must use Table 3–7 *PSR Drain Service-Type Connection State Parameters* to configure parameters for PSR Drain. The host must use the TX channel number associated with the connection to configure the appropriate connection state.

All 32-bit reserved words must not be modified, e.g., words 1 through 6 of the PSR Drain connection state must not be modified. However, any reserved bits that are part of a word containing a valid parameter, e.g., bits 31:24 of word 0 of the PSR Drain connection state, must be cleared when that word is written in Parameter Memory. For example, if PSR Drain Rate = 20 cells/second, then word 0 must be written as 00000014h.

C. Enable Scheduling for Connection

The host must turn on scheduling for the connection.

Part 1

The host must poll the Scheduler_On_Ready bit in the Status Register to verify that the scheduler is ready to add a new connection.

Status Register

Bit(s) (4) 1 – Scheduler is ready to accept new connection request

0 – Scheduler is not ready to accept new connection request

Part 2

The host shall then write the channel number of the connection to be added to the Scheduler On Register. The channel number of the original connection must be used for the PSR Drain connection channel number.

Scheduler On Register

Bit(s) (10:0) Channel number

D. Wait for PSR Drain Complete Interrupt

When the host receives an interrupt from the TNETA1585 device indicating PSR Drain operation is complete, then connection teardown is complete for the connection in the TNETA1585 device.

E. Disable PSR Drain Interrupt (Optional)

After tearing down the connection for PSR Drain, the host may configure the Interrupt Mask Register to disable an interrupt due to completion of PSR Drain operation if it no longer plans to use the PSR Drain feature.

Interrupt Mask Register

Bit(s) (30) 1 – Enable PSR Drain Complete interrupt

0 – Disable PSR Drain Complete interrupt

Appendix D: Low ACR Detection

The Low ACR Detection feature is provided to aid efficient host buffer management schemes to be implemented for ABR service-type connections. If a channel is experiencing congestion, its ACR will be driven low and transmission of data may be halted. When this occurs, data that has already been queued by the host into its transmit buffers may occupy those buffers indefinitely until congestion ceases. Thus, the pool of available transmit buffers may be severely limited when several channels become congested, each with data occupying a number of transmit buffers. To minimize this effect, the host can slow the queuing of data to the transmit buffers if it knows a particular channel is congested.

The host can assume that a channel is congested when its ACR falls below a certain threshold. With Low ACR Detection, the host can set the threshold at which it wants to be notified of low ACR on a connection. This corresponds to the ACR_Low_Threshold parameter in Parameter Memory. The host can also specify the OK threshold by specifying the amount above the ACR_Low_Threshold value at which it wants to be notified of a recovered ACR on a connection, i.e., when the ACR that was previously below the ACR_Low_Threshold value is now above the ACR_Low_Threshold by a delta amount. This delta value corresponds to the ACR_Increase_Delta parameter in Parameter Memory. This delta value is used for hysteresis.

The host can be notified of any significant changes to a connection's ACR via interrupt. The ACR_is_Low and ACR_is_OK bits in the Status Register will generate interrupts (if the corresponding bits in the Interrupt Mask Register are set) when a connection's ACR crosses the low and OK thresholds, respectively. The host can then adjust queuing of data accordingly. Note that an ACR OK interrupt will be generated only if the ACR was previously low and is now OK, i.e., new connections start out with ACR OK status but do not invoke ACR_is_OK interrupts.

To implement the Low ACR Detection feature, the host must do the following:

A. Configure Parameter Memory

In addition to the parameters required for an ABR service-type connection, the parameters listed in Table D–1 need to be configured in Parameter Memory to support the Low ACR Detection feature for a particular connection. Both Low ACR Detection parameters are with respect to the TX side of the connection, i.e., the TX channel number must be used to derive the Parameter Memory address and configure the appropriate connection state.

Table D–1. Low ACR Detection Parameter Memory Setup

DESCRIPTION	WORD	BIT LOCATION(S)	VALUE	SECTION NUMBER
ACR_Increase_Delta	5	(28:24)	Delta value	3.2.1.1.6
ACR_Low_Threshold	15	(31:16)	ACR low threshold	3.2.1.1.11

Note that the host has indirect access to Parameter Memory and must use the Write Data and Write Address Registers for updates. Also note that any reserved bits in words 5 and 15 must be cleared when words 5 and 15 are configured.

B. Enable Low ACR Detection Interrupts

The host must configure the Interrupt Mask Register to enable an interrupt due to ACR crossing either the low or OK thresholds. This step can be done instead after power up/reset if the host intends to use the Low ACR Detection feature all the time, in which case, it need not be done here again.

Interrupt Mask Register

Bit(s) (6) 1 – Enable ACR Low interrupt

0 – Disable ACR Low interrupt

Bit(s) (5) 1 – Enable ACR OK interrupt

0 – Disable ACR OK interrupt

C. Read ACR Low/OK Registers

When the host receives an ACR_is_Low or an ACR_is_OK interrupt, it needs to read the ACR Low Register or the ACR OK Register, respectively, to determine what channel(s) have crossed the specified thresholds. Before doing so however, the host should mask the interrupt it just received to avoid redundant interrupts while the interrupt is being serviced. Once the host is through reading the respective register, it should enable the interrupt again.

ACR Low Register

Empty (31)	Ready (30)	Reserved (29:11)	Channel Number (10:0)

Bit(s) (31) 1 – There is no connections with low ACRs

0 – There is at least one connection with low ACR

Bit(s) (30) 1 – Register content is valid if Empty is clear

0 – Register content is not valid

ACR OK Register

Empty (31)	Ready (30)	Reserved (29:11)	Channel Number (10:0)

Bit(s) (31) 1 – There are no connections with OK ACRs

0 – There is at least one connection with OK ACR

Bit(s) (30) 1 – Register content is valid if Empty is clear

0 – Register content is not valid

NOTE:

The Empty bit must be clear and the Ready bit must be set simultaneously for the specified channel number to be valid.

The host may need to read the respective register multiple times during one interrupt service routine since it is possible that several connections have had their ACRs cross the threshold by the time the host services the interrupt. This is indicated by the Empty bit. As long as the Empty bit is clear, there are other connections that satisfy the condition. The TNETA1585 maintains two lists to support this, one containing the list of connections with their ACR low, the other containing the list of connections whose ACR has become OK.

C. Read ACR Low/OK Registers (continued)

Due to the asynchronous nature of passing ACR Low/OK information to the host, it is possible for a connection's ACR to cross one of the thresholds and then cross back to the original state before the host has read the appropriate register (ACR Low or ACR OK Register). The host would receive an interrupt for each time ACR crosses a threshold. However, a channel can only be stored in one list at a time, and every time the ACR crosses the threshold, both lists are updated, i.e., clear the channel's entry in one and set it in the other. This may result in the example scenario below:

1. The host receives an ACR_is_Low interrupt and reads the channel from the ACR Low Register.
2. The host receives an ACR_is_OK interrupt for the same channel only to find the ACR OK Register empty by the time it reads the register. This is because the channel's ACR has gone low again.
3. Shortly after the ACR_is_OK interrupt, it receives an ACR_is_Low interrupt for the same channel. Thus, the host receives two consecutive ACR_is_Low interrupts for the same channel.

D. Notes on Queuing Data for a Connection With an Artificial ACR OK Status

The TNETA1585 may issue an ACR_is_OK interrupt to the host due to data not being available on a connection, although the connection may still have low ACR. This is done to prevent the situation where a low ACR connection goes to sleep because there is no more data, but the host does not queue more data because ACR is low. This situation could cause the connection to remain in sleep mode indefinitely. Thus, the TNETA1585 will issue an ACR_is_OK interrupt to the host when data is not available on the connection so that the host can queue additional data. Thus, upon receiving an ACR_is_OK interrupt, the host should check to see if the associated PSR is empty for that connection. If the PSR is empty, there is a high likelihood that the connection is in an artificial ACR OK mode. The host can verify the actual ACR rate of the connection by reading the connection state in Parameter Memory. If ACR is indeed still low, the host should only queue a few more buffers and wait for subsequent interrupts.

E. ACR Low Detection Pseudocode Summary

The following pseudocode summarizes the procedure for ACR Low Detection.

```
// Initialization
    PM_Word5                // Configure ACR_Increase_Delta parameter
    PM_Word15               // Configure ACR_Low_Threshold parameter
    Set_Intr_Mask_Bit6      // Enable ACR Low interrupt
    Set_Intr_Mask_Bit5      // Enable ACR OK interrupt
// Interrupt Handling
    Receive_ACR_Low         // Receive ACR Low interrupt
    Receive_ACR_OK          // Receive ACR OK interrupt
                            // Note: ACR Low interrupts should always be serviced
                               before ACR OK interrupts
// Service ACR Low interrupt
Clr_Intr_Mask_Bit6          // Disable ACR Low interrupt while it is being serviced
while (!ACR_Low_Reg_Bit31) { // Continue reading ACR Low Register while there is an
                               entry in the ACR Low list
    if (ACR_Low_Reg_Bit30) {// Verify that the channel number entry is a valid entry
     ACR_Low_Channel        // Read the channel number of the connection with ACR low
    }
}
Set_Intr_Mask_Bit6          // Enable ACR Low interrupt after servicing
// Service ACR OK interrupt
Clr_Intr_Mask_Bit5          // Disable ACR OK interrupt while it is being serviced
while (!ACR_OK_Reg_Bit31) {  // Continue reading ACR OK Register while there is an
                               entry in the ACR OK list
    if (ACR_OK_Reg_Bit30) {  // Verify that the channel number entry is a valid entry
    ACR_OK_Channel          // Read the channel number of the connection with ACR OK
    }
}
Set_Intr_Mask_Bit5          // Enable ACR OK interrupt after servicing
```

Appendix E: Calling Party Distinction

The Calling Party Distinction feature is used to support handshaking between two ABR connection end points to prevent transmission of data until both parties are ready to receive, and thus prevent loss of data. Specifically, it will allow the party initiating the call (calling party) to transmit an RM cell immediately after connection setup. The non-calling party, on the other hand, must wait until it receives a cell of any type sent by the calling party before transmitting its first RM cell.

For those systems that do not wish to take advantage of this feature, they must configure this bit to calling party, i.e., set this bit. Otherwise, the only configuration required is that the host configure the Calling Party bit in Parameter Memory to indicate whether or not the connection is the calling party.

Appendix F: Receive Synchronization Check

With the Receive Synchronization Check feature, the host can assume that the TNETA1585 receive circuitry is synchronized with the SAR receive circuitry unless it is notified via interrupt by the TNETA1585 device. To verify receive synchronization, the TNETA1585 compares the VPI/VCI field of all received cells to the expected VPI/VCI of that received channel. The expected VPI/VCI of a channel is configured in Parameter Memory by the host.

Thus, to utilize the Receive Synchronization Check feature, the host must do the following:

A. Enable-Synchronization Error Interrupt

The host must configure the Interrupt Mask Register at power up/reset to enable an interrupt due to a receive synchronization error.

Interrupt Mask Register

Bit(s) (31) 1 – Enable receive synchronization error interrupt

0 – Disable receive synchronization error interrupt

B. Configure Parameter Memory

In addition to the parameters configured for a particular connection, the host must configure the following parameters at connection setup to support the Receive Sync check. Both Receive Sync parameters are with respect to the RX side of the connection, i.e., the RX channel number must be used to derive the Parameter Memory address and configure the appropriate connection state. For VC-Level ABR connections, TX and RX Channel Numbers are identical.

Table F–1. Receive Sync-Check Parameter Setup

DESCRIPTION	WORD	BIT LOCATION(S)	VALUE
Expected VPI	8	(31:20)	Expected VPI
Expected VCI	8	(19:4)	Expected VCI

Note that the host has indirect access to Parameter Memory and must use the Read Data and Read Address Registers or the Write Data and Write Address Registers for all accesses. Also note that bits 3:1 of word 8 must be cleared when word 8 is configured.

Appendix G: Code Update Sequence

When new code is available, it shall replace the old code already downloaded into the system. The following outlines the procedure for downloading new code to a system that is already up and running.

A. Tear Down Connections

All connections must be torn down in the TNETA1585.

B. Turn Off Processors

Both the source and destination processors in the TNETA1585 must be disabled.

C. Reset Device

The TNETA1585 device must be reset.

D. Proceed as Usual

Proceed with normal power up/reset initialization as outlined in Section 4.1.

Appendix H: Fixed-Rate Connections

The TNETA1585 can also be used for fixed-rate connections, such that the host specifies a rate that is not adjustable by the TNETA1585, for scheduling. Data cells are simply scheduled at the host specified rate. Provided that there is no scheduler congestion and negligible bus latencies, data will be transmitted at the fixed rate. When data is not available, sleep mode is applied.

To configure the TNETA1585 for a fixed-rate connection, the host must configure the TNETA1585 for service-type VBR-nrt. Configuration for fixed-rate connections are very similar to VBR-nrt configuration, with the exceptions outlined below. The differences in configuration allow the TNETA1585 to bypass all VBR-nrt specific procedures and to instead schedule data cells at the host specified rate.

To configure the TNETA1585 for fixed-rate connections, refer to Section 3.2.1.2 and Appendix B, which are the VBR-nrt specific configuration procedures. Parameter Memory parameters are modified from the VBR-nrt Parameter Memory configuration as follows for fixed-rate connections:

A. Set PCR to host specified fixed rate

B. Set SCR to host specified fixed rate

C. Set MBS to zero

Glossary

A

ABR
Available Bit Rate

ACR
Allowed Cell Rate

ADTF
Allowed Cell Rate Decrease Time Factor

ATM
Asynchronous-Transfer Mode

B

BRM
Backward Resource Management Cell

C

CDF
Cutoff Decrease Factor

CLP
Cell Loss Priority

CRC
Cyclic Redundancy Check

Crm
Count Resource Management

D

DMA
Direct-Memory Access

E

EFCI
Explicit Forward Congestion Indication

F

FIFO
First In, First Out

57

I

ICR
Initial Cell Rate

M

MBS
Maximum Burst Size

MCR
Minimum Cell Rate

N

NNI
Network-to-Network Interface

Nrm
Counter n (resource management)

P

PCR
Peak Cell Rate

PSR
Packet Segmentation Ring

R

RAM
Random-Access Memory

RDF
Rate Decrease Factor

RIF
Rate Increase Factor

RX, Rx
Receive

S

SAR
Segmentation and Reassembly

SCR
Sustainable Cell Rate

T

TM
Traffic Management

TM4.0
ATM Forum Traffic Management Specification Version 4.0

TNETAxxxx
TI Networking ATM device xxxx

Trm
Timer (resource management)

TX, Tx
Transmit

U

UNI
User-to-Network Interface

UTOPIA
Universal Test and Operations PHY Interface for ATM

V

VBR-nrt
Variable Bit Rate – nonreal time

VC
Virtual Channel (virtual circuit)

VCC
Virtual Channel Connection

VCI
Virtual Channel Identifier

VPCI
Virtual Path and Channel Identifier

VPI
Virtual Path Identifier