TOSHIBA

155 Mbps PCI-ATM

SAR Chip

1998

TC35856CF

VERSION 1.1

DATA BOOK

TOSHIBA AMERICA ELECTRONIC COMPONENTS, INC.

FLOWmaster™ is a trademark of Digital Equipment Corporation.

Table of Contents:

Table of Figures:

Table of Tables:

1. Introduction

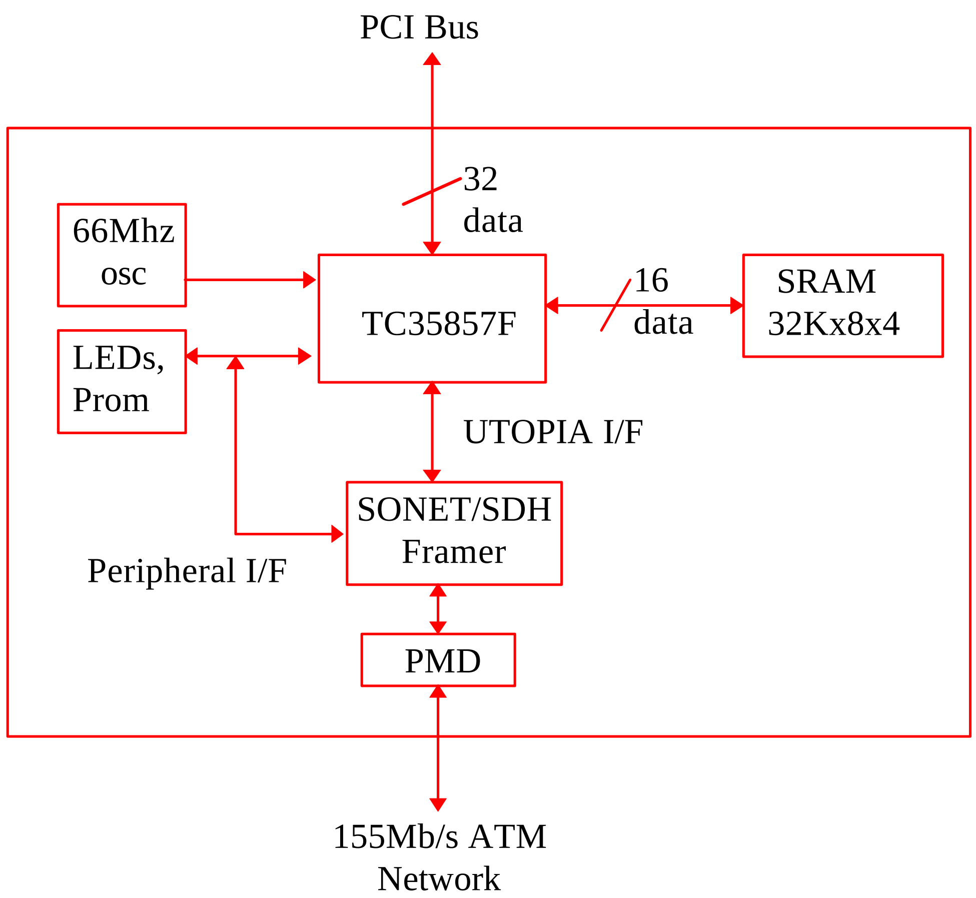
TC35856BF is a single-chip Segmentation and Reassembly controller aimed at the high-end client through low-end server markets, and edge switch application. Its main goals are:

- Low cost
- Fast time to market
- High performance
- Full ATM Forum UNI Version 4.0 compliance
- Both ATM Forum ABR and Digital FLOWmaster™ congestion control support

TC35856BF interfaces directly to the PCI bus (per the PCI bus specification Version 2.1) and to a 155.52 Mb/s UTOPIA-compliant PHY interface (STS-3c/STM-1 rate). TC35856BF can interface as not only master device (ATM) but also slave device (PHY) to UTOPIA interface. In addition, a small amount of on-board SRAM provides storage for the data structures used by TC35856BF.

Figure 1.-1 shows a typical implementation of an adapter card using TC35856BF.

Figure 1.-1 - Typical NIC Implementation Using TC35856BF



2. TC35856BF Features

The full feature set supported by TC35856BF is described in the following sections.

2.1 Low Cost

TC35856BF is implemented in a 208-pin PQFP package using 3.3V internal technology with 5V I/O. It is one of the lowest cost 155 Mb/s SAR chips in the industry.

2.2 PCI Bus Interface

TC35856BF interfaces directly to the PCI bus, per the *PCI Local Bus Specification, Revision 2.1*. 32-bit PCI with Universal I/O is implemented. Multiple longword accesses (burst accesses) are supported both in bus master and in slave modes.

2.3 Flexible UTOPIA operation

TC35856BF has following flexible enhanced features on UTOPIA interface.

(a) Direction of Clock for UTOPIA interface

The line-side of TC35856BF implements the industry-standard UTOPIA interface specification. Specifically, the UTOPIA level-2, 8-bit interface running at 33.333Mhz is supported. For that purpose TC35856BF outputs 33.333MHz clock. However, several framer device doesn't support 33MHz operation, but supports 25MHz operation as level-1 interface. In order to work with level-1 device, TC35856BF is also configured to input minimum 25MHz UTOPIA interface clock from outside.

(b) Reversible UTOPIA-Compliant PHY interface

TC35856BF may also be connected directly to devices implementing the ATM layer of the UTOPIA specification (as opposed to the PHY layer) through its **Reversible UTOPIA** mode of operation. In Reversed UTOPIA mode, TC35856BF acts as the UTOPIA PHY layer at the line-side interface. It works not only Single-PHY mode but also Multi-PHY interface except for address decode function. This is extremely useful in ATM switch applications, where TC35856BF may be connected to the switch fabric through the same mechanism as all other ports using PHY devices in the switch.

(c) Cell level and Octet level interface

TC35856BF supports both of cell level and octet level handshaking for either normal UTOPIA mode and reverse UTOPIA mode.

(d) Idle/unassigned cell generation

TC35856BF is programmed to generate following cells.

i) unassigned cell and data cell
ii) idle cell and data cell
iii) data cell only.

When TC35856BF works as SAR device, it is desired to generate "unassigned cell (VPI=0, VCI=0, PT=0, CLP=0)" when it doesn't output data cell. However, when it works as PHY device (reverse UTOPIA mode), it is desired to generate "idle cell (VPI=0, VCI=0, PT=0, CLP=1") instead of unassigned cell. And when MPHY mode is used, no idle/unassigned cell is required. These features are available for both of normal UTOPIA and reverse UTOPIA mode.

(a) Transmit Cell Interval measurement

TC35856BF has two method to control cell generation rate. One method uses cell transmission rate on UTOPIA interface. Total cell generation rate including data cell and idle/unassigned cell is exactly the same as cell rate on SONET/SDH or other line rate. It depends upon a PHY devices. The other method uses an internal cell interval timer to measure cell interval. The timer covers 155.52Mbps to 25Mbps line rate, or slower. The timer is useful when TC35856BF is programmed not to generate idle/unassigned cell.

2.4 4K Simultaneous Transmit and Receive VCs

TC35856BF supports up to 4K Virtual Circuits simultaneously, in both the Receive and Transmit directions. This means that up to 4K AAL5 PDUs may be segmented and transmitted simultaneously (4K-way interleaving of AAL5 PDUs on the link), while 4K AAL5 PDUs are being simultaneously received and reassembled.

2.5 AAL5 Segmentation and Reassembly

In the transmit direction, the driver provides TC35856BF either with complete AAL5 CPCS-PDUs including a 4-byte AAL5 CRC placeholder or with AAL5 CPCS-PDU only. The Pad, Control and Length fields provided by the driver must be correct, and are not checked by TC35856BF. The AAL5 CRC need not be correct. TC35856BF segments AAL5 PDUs, appends the cell headers to each ATM cell created by the segmentation process, computes AAL5 CRCs, and overwrites the placeholder CRCs with the computed CRCs in the transmitted PDUs. Enhanced feature of TC35856BF is automatic generation of AAL5's PAD field. TC35856BF calculates the length of PAD fields and generates complete AAL5 trailer from PAD field to CRC-32 field.

Segmentation and transmission of up to 4K AAL5 PDUs **simultaneously** is supported. Multiple PDUs can be queued on the desired VC. Those PDUs are segmented in sequential. In the receive direction, TC35856BF may reassemble up to 4K AAL5 PDUs **simultaneously**. In addition, it calculates and verifies the AAL5 CRC and length fields of received AAL5 CPCS-PDUs.

2.6 AAL5 received packet size limitation

TC35856BF can limit the maximum received length of AAL5 packet in the unit of 48 bytes. It allows TC35856BF to use host memory effectively.

2.7 Segmentation and Reassembly on DMA

Segmentation and Reassembly of AAL5 PDUs is done as the PDUs are DMA'd to and from host memory. This eliminates the need for large amounts of memory on the adapter.

2.8 Raw Cell Support

In addition to extensive support for the AAL5 adaptation layer, TC35856BF provides limited support for other adaptation layers via “raw” 52-byte cells.

2.9 OAM and RM CRC-10 Support

TC35856BF can selectively (per cell) generate and append the CRC-10 field to transmitted raw cells under driver control. This function may be used for F4 and F5 OAM cells, or when transmitting AAL3/4 frames using the raw cell functionality. TC35856BF will always generate and append the CRC-10 field to transmitted RM cells.

TC35856BF will also selectively check the CRC-10 field of all received OAM and RM cells, and may selectively (per-VC) check the CRC-10 field of other cells. All cells which are found to have CRC-10 errors should be discarded by the driver upon error reported from TC35856BF.

2.10 Full Performance at STS-3c/STM-1 Line Rates

Although the STS-3c/STM-1 rate is specified to be 155.52 Mb/s, SONET overhead reduces the theoretical achievable data transfer rate to 149.76 Mb/s, ATM overhead to 135.632 Mb/s, AAL5 overhead to 135.220 Mb/s (with 9,180 Byte Classical IP SDUs), and Classical IP overhead to 135.102 Mb/s.

2.11 CBR Support

TC35856BF supports the ATM Forum Constant Bit Rate (CBR) service class fully, with per-VC rates from ~3 Kb/s to 149.76 Mb/s (at STS-3c/STM-1 line rates).

2.12 ABR and UBR Support

TC35856BF provides full support for the ATM Forum UNI 4.0 Traffic Management Specification's ABR and UBR service classes. The ABR service class specifies a complex form of end-to-end rate-based flow control (with intermediate switches optionally acting as virtual sources and virtual destinations) using special cells known as *RM cells* to carry the rate control information. TC35856BF is completely ABR-compliant, generating and processing ABR RM cells and providing per-VC rate control without driver intervention. The UBR service class specifies a peak cell rate per VC which is driver specified and generally not time-variant.

2.13 FLOWmaster™ Support

In addition to the end-to-end flow control provided by the ABR service class, TC35856BF implements FLOWmaster credit-based flow control. This is a simple, robust, method of hop-by-hop flow control already shown to work well and implemented by several products in the market. FLOWmaster and ABR may operate either together or independently. In general, FLOWmaster is enabled on a per-link basis, whereas ABR is enabled on a per-VC basis.

2.14 Generic Flow Control (GFC) Support

TC35856BF implements Generic Flow Control as specified in ITU-T SG 13 revised version documents I.361-TD41 and I.150-TD42, November 1994, Geneva. Group A controlled and uncontrolled VCs are supported. TC35856BF maintains transmit bandwidth guarantees for uncontrolled VCs. GFC support is required by low cost ATM switches that are optimized for the desktop market.

2.15 MPEG support

TC35856BF provides MPEG support through 2 separate mechanisms:

- Segmentation of MPEG "super-packets", where each super-packet is segmented by TC35856BF into multiple 8-cell AAL5 packets.

- Host-controlled insertion of "idle packets" in order to achieve exact rates and reduce the amount of buffering needed at the set-top.

MPEG super-packet segmentation is selected on a per-VC basis, and idle packet insertion is done on a per-packet basis, where idle packets may consist of any number of cells.

2.16 Peripheral Interface

TC35856BF provides a general-purpose Peripheral Interface through which the host may access various on-board devices. These may include, but are not limited to:

- PHY chip CSRs
- MAC address ROM
- LEDs
- Jumpers
- Error-Logging EPROM
- PCI Subsystem and Subsystem Vendor ID Registers

The Peripheral interface directly supports a 64KB address space and an 8-bit wide datapath.

2.17 Small On-board Memory Requirements

Because segmentation and reassembly is done on DMA, rather than in on-board memory, TC35856BF requires 512KB of on-board, 15ns SRAM in order to support 4K transmit and receive VCs, 128KB for 1K VC and 64KB for 256 VCs. These low memory requirements contribute to an extremely cost-effective PCI-ATM adapter.

TC35856BF also supports higher density SRAM such as 4Mbit SRAM. It uses one piece of 256Kw x 16bit SRAM for 4K VCs or one piece of 64Kw x 16 bit SRAM for 1K VCs.

2.18 Scaleable Architecture

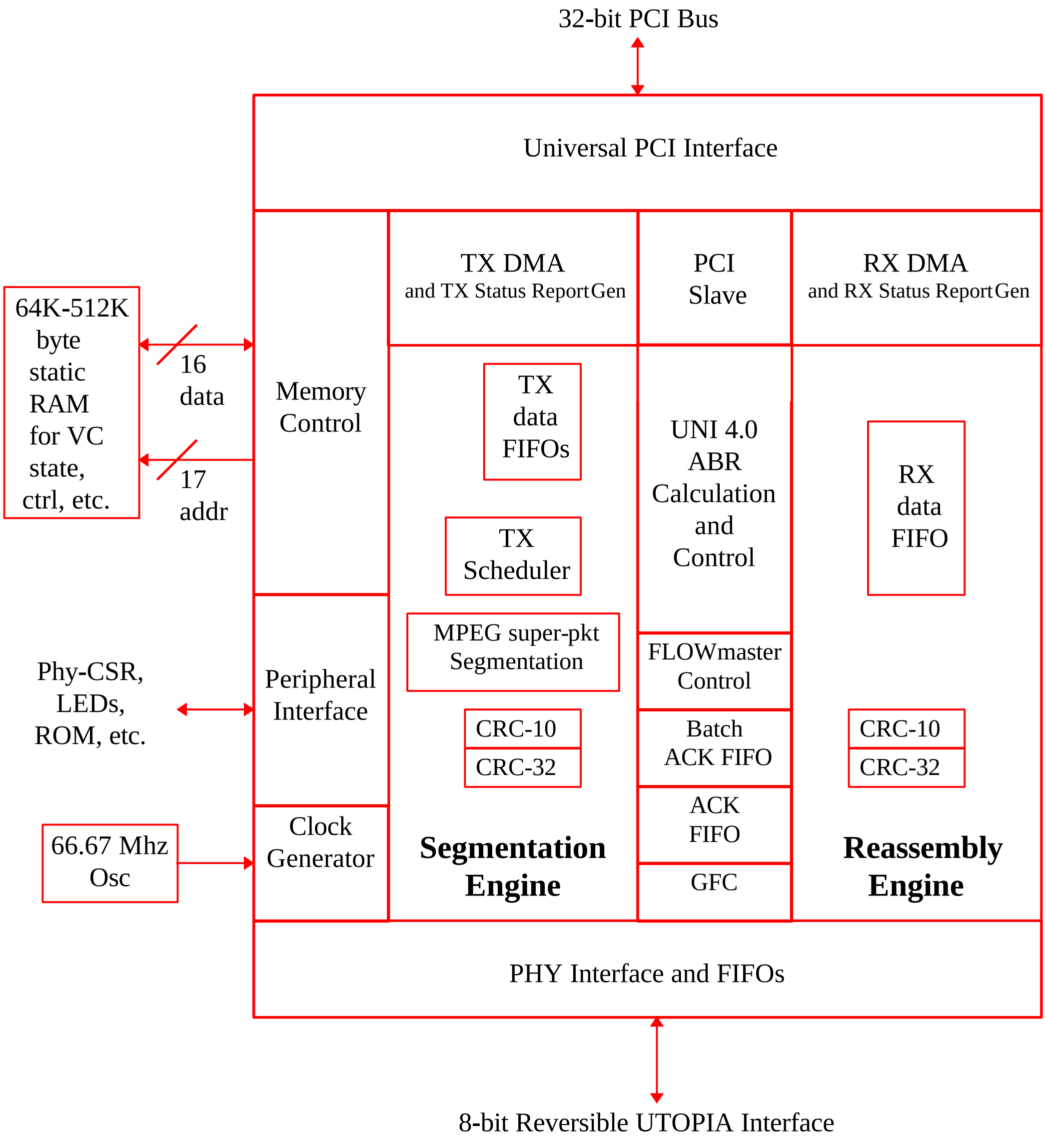
The TC35856BF architecture is designed to efficiently scale from low-cost desktop applications to high-end servers. For example, TC35856BF supports from 256 VCs for the desktop to 4K VCs for high-end servers.

3. Functional Description

3.1 Introduction

In this chapter, a detailed overview of the operation of TC35856BF is presented. The general hardware architecture and the driver operations needed to transmit and receive data within the framework of this architecture are both discussed in some detail. Figure 3.-1 is a top-level diagram illustrating the general architecture of TC35856BF. Since ATM is fundamentally a full-duplex technology, the transmit and receive operations of TC35856BF are largely (though not completely) independent. Therefore, each operation is discussed separately in the text which follows, with the exception of the PCI, PHY and Peripheral interfaces, which are shared between the transmit and receive processes.

Figure 3.-1 - Top Level Block Diagram of TC35856BF



3.2 Conventions

The term longword (LW) refers to a 4-byte unit of data. The term Word (W) refers to a 2-byte unit of data. The *Transmit* (or *tx*) direction of data flow refers to data flowing from host memory, into TC35856BF, and onto the link. The *Receive* (or *rx*) direction of data flow refers to data flowing from the link, into TC35856BF, and then into host memory. The word *packet* used in this document refers to any of the following units of data:

- AAL5 CPCS-PDUs
- Raw cells. These are cells which are not part of an AAL5 CPCS-PDU. They may be F4 or F5 OAM cells, ABR RM cells, or cells for an AAL other than AAL5. Note that transmit RM cells are generated by TC35856BF; the driver doesn't need to generate RM cells (although there is nothing in TC35856BF to prevent the driver from doing so).

There are many FIFOs and linked lists employed in the TC35856BF architecture. For both types of data structures, entries are written to the ***Tail*** of the structure, and read from the ***Head***.

The multiplier ***K*** used after numbers (e.g., "1K", "2K") denotes the quantity 1024, **not** 1000.

3.3 Transmit Functional Description

3.3.1 Tx Slots

In order to begin the transmission of a packet, the driver must place the data contained in the packet into one or more physically contiguous areas of host memory. Each of these blocks of physically contiguous host memory is hereafter referred to as a *slot*. A single AAL5 packet may be contained in a single slot, or may span many slots. Each raw cell must be contained in a single slot. Slots containing portions of an AAL5 PDU may vary in size from 1 byte (0 bytes is not allowed) to 16 Kbytes, with 1-byte granularity, while slots containing raw cells must be exactly 52 bytes long.

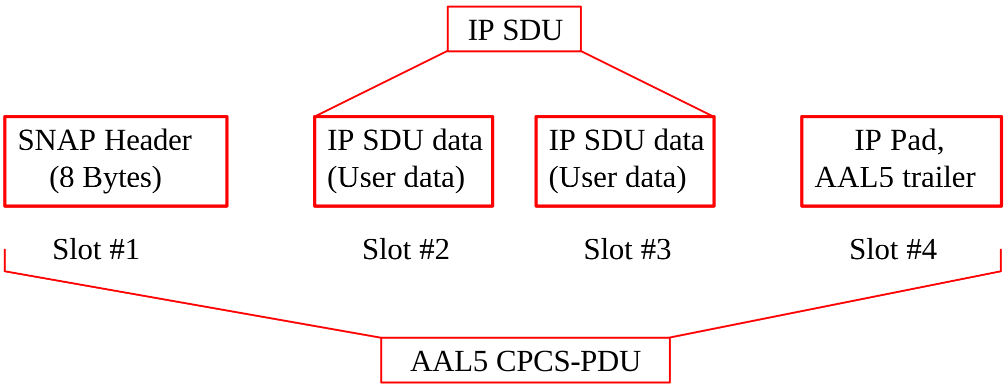
The first byte of a raw cell or AAL5 packet occupies the lowest byte address of a slot. A single slot can contain at most one AAL5 packet or raw cell.

Tx slots may be located at any byte address in host memory, although for best performance the starting address of each slot should be longword-aligned, if possible. Similarly, for best performance, slots should generally be as large as possible, with the optimal case being a one-to-one correspondence between packets and slots.

Each AAL5 packet posted for transmission must contain the AAL5 pad (0 to 47 bytes), Control (2 bytes), Length (2 bytes) and CRC placeholder (4 bytes) fields. TC35856BF will compute the correct AAL5 CRC and overwrite the placeholder CRC when transmitting the packet.

Figure 3.-2 shows how a classical IP packet (always carried over AAL5) to be transmitted on the link may appear in host memory when it occupies 4 tx slots. Note that the two slots containing the IP SDU may be different sizes.

Figure 3.-2 - Classical IP Packet Spread Out Across 4 tx Slots



The driver may allocate as much memory as it wishes for tx slots, and any number of tx slots may be in use by the driver at any point in time. Architecturally, TC35856BF only limits the total number of slots from which it may be reading data for transmission at any one point in time to the number of VCs which it supports.

3.3.2 Posting a tx slot and a segmentation engine control word

Once the driver has placed data for one or more packets across one or more VCs in tx slots in host memory, it must indicate to TC35856BF the location of the slots and the VCs with which each slot is associated. In order to post a single slot for transmission, the driver writes a 2 LW ***tx slot***

descriptor to a 16-LW deep address space in TC35856BF. The write must be 2-LW (quadword) aligned. This address space is hereafter referred to as the ***tx pending slot*** address space. When the driver writes 2 consecutive LWs describing a single tx slot to this address space, the operation is referred to as ***posting a tx slot***. And also the driver may post a control word for segmentation engine through tx pending slot address space. Note that the 2 LW writes are by no means guaranteed to occur in a single PCI burst; TC35856BF is designed to handle an arbitrarily long delay between the write of the first and the second LWs corresponding to the posting of a single tx slot. The tx pending slot address space is 16-LW deep in order to allow the posting of up to 8 tx slots in a single 16-LW PCI burst. The most significant bit of the first LW is used to distinguish the tx pending slot from the control word. When the bit is ‘0’, the posted words contain information for tx slot descriptor. When the bit is ‘1’, the posted words contain a control word for segmentation engine.

3.3.3 Tx Slot Descriptor

The first LW in a tx slot descriptor contains the 14-bit size of the slot (in bytes), the 12-bit VC of the data contained in the slot, and some additional bits describing the type of slot being posted. The second LW contains the 32-bit starting address of the slot being posted. A ***tx slot descriptor*** is shown in Figure 3.-3, and a description of each of the fields in the descriptor follows.

Figure 3.-3 - Tx Slot Descriptor

	1 bit	1 bit	1 bit	1 bit	12 bits	1 bit	1 bit	14 bits
LW0:	Cntl=0	RAW	IDLE	MGMT	VC	CRC10_EN	EOP	SLOT_SIZE
LW1:	BASE_ADDRESS							

- **Cntl**

  This 1-bit field should be ‘0’ in order to indicate tx pending slot.

- **RAW**

  This 1-bit field indicates whether the transmit slot contains data for an AAL5 or idle packet (0) or for a single 52-byte raw cell (1). The functionality of this bit is described more fully in Section 3.3.3.1.

- **IDLE**

  This 1-bit field indicates whether the data in the slot should be DMA’d and transmitted normally (0), or whether TC35856BF should schedule an idle cell for transmission for every cell contained in the slot (1). This bit is used for timing MPEG streams. When this bit is set, no data from the slot will be DMA’d or transmitted on the link.

- **MGMT**

  This 1 bit field indicates if the data in the slot constitutes an F4 or F5 OAM cell. When this bit is set, it indicates to TC35856BF that the data in the slot should be transmitted whether or not the VC of the data has FLOWmaster credits available (even if the VC is under FLOWmaster flow control).

- **VC**

  This 12-bit field indicates the VC of the data contained in the transmit slot in host memory.

- **CRC10_EN**

  This 1-bit field is of significance only when the **raw** bit is set, indicating that the transmit slot contains a single 52-byte cell. This bit is used to select whether or not a 10-bit CRC should be calculated and appended to the raw cell being transmitted. **CRC10_En** should be set to 0 for AAL5 slots. It will generally be set to 1 for both F4 and F5 OAM cells.

- **EOP**

  This 1-bit field is used for 2 purposes: to indicate that a transmit slot contains the last byte of data of an AAL5 or idle packet (**raw** = 0), or to indicate that a status report and interrupt should be generated on completion of the DMA of a raw cell slot (**raw** = 1). Section 3.3.3.2 describes the functionality of this bit in more detail.

- **SLOT_SIZE**

  This 14-bit field indicates the number of bytes contained in the transmit slot. If the slot is a raw cell slot (**raw** = 1), then this field should always be set to 52 by the host. For AAL5 or idle slots (**raw** = 0), this field must be in the range 1 - 16383 (the value 0 is not allowed).

- **BASE _ADDRESS**

  This 32-bit field indicates the physical base address of a transmit slot in host memory. It may contain any value for idle slots, since no data is DMA'd from idle slots.

3.3.3.1 Slot Raw and CRC10_En Bit Usage

When a VC is opened, it is set-up as an AAL5 VC, an AAL3/4 VC, or some other type of flow. However, TC35856BF recognizes only 3 types of flow: AAL5, Raw or MPEG. 2 bits in VC state identify the type of flow to which a VC has been assigned. In this document, any reference to AAL5 streams includes MPEG streams (since MPEG streams are assumed to be carried over AAL5). An MPEG stream is different from an AAL5 stream only in that a single MPEG packet ("super-packet") may be segmented into several AAL5 packets. When packets are transmitted on a VC which is an AAL5 VC, the AAL5 CRC will be computed and appended to the packets.

Recall, however, that OAM-F5 cells are identified through the **PT** field of the cell, rather than by VC, and may be carried on the same VC as an AAL5 flow. Whenever the host wishes to transmit an F4 or F5 OAM cell, it should set the **raw** and **crc10_en** bits in the slot descriptor for the cell. This indicates to TC35856BF that the slot contains a single raw cell for which the CRC-10 must be computed and appended. It also indicates (in the case of F5 cells carried over an AAL5 VC) that the CRC-32 intermediate result in VC state should **not** be updated.

Note that, aside from the fact that they do not affect the CRC-32 calculation, OAM-F5 cells are treated no differently than data cells of the VC on which they are being sent. In terms of DMA and transmit scheduling and for purposes of ABR rate-based flow control calculations they are considered as a part of the normal data flow of the VC.

OAM-F4 cells are identified by their VC (VC = 3 or 4). However, the driver must also set the **raw** and **crc10_en** bits in the slot descriptor for slots containing OAM-F4 cells in order to be interpreted correctly by TC35856BF.

Note that TC35856BF will generate the CRC-10 field for all transmitted cells for which the **crc-10_en** bit is set. It will then overwrite the CRC-10 bits in the transmitted cell with the calculated CRC-10 value. This function may be used to aid in the transmission of AALs other than AAL5 which use the CRC-10, such as AAL3/4.

3.3.3.2 Slot EOP Bit Usage

The EOP bit in tx slot descriptors is used for two different purposes:

- Termination of AAL5 packets:

  An AAL5 packet to be transmitted by TC35856BF may be contained in one or more tx slots in host memory. The tx slot descriptor associated with the slot containing the last byte of the packet **must** have its **EOP** bit set. TC35856BF will use the fact that this bit is set to overwrite the driver-provided CRC placeholder with the computed AAL5 CRC, set the **EOM** bit in the **PT** field of the last cell of the packet, initiate a status report to the driver to indicate that the packet has been completely read from host memory, and generate an interrupt (see Section 3.3.9).

- Mitigation of Tx Status Reports

  When all the data in a tx slot has been DMA'd across the PCI bus from host memory to TC35856BF, it is said that the tx slot has been consumed. TC35856BF will report that a slot in host memory has been consumed through a ***tx status report***. These are described fully in Section 3.3.9. In general, TC35856BF will only report consumption of slots for which the **EOP** bit is set. The driver may also choose to report and interrupt **per slot** on a per-VC basis (AAL5 streaming mode).

  For AAL5 packets, this means that only the consumption of the last slot of a packet will result in a status report. For raw cells, the driver may choose to set the **EOP** bit for every slot and thus every cell (since only one raw cell may be contained in a tx slot), or it may choose to mitigate the frequency with which raw cell status reports take place by setting the **EOP** bit every **N** slots, where **N** is chosen so as to maximize the performance of the driver.

  For idle slots, the driver should set the **EOP** bit only if it desires to be informed of the consumption of the slot. Note that although no data is DMA'd from idle slots, it may still be useful for the driver to learn when an idle slot has been consumed (in the sense that it has been processed by TC35856BF), since idle slots use up tx slot descriptors in the same way that other slots do, and the number of outstanding tx slot descriptors is limited to the size of the tx free slot list.

3.3.3.3 *Restrictions When Posting OAM-F5 and Idle Slots*

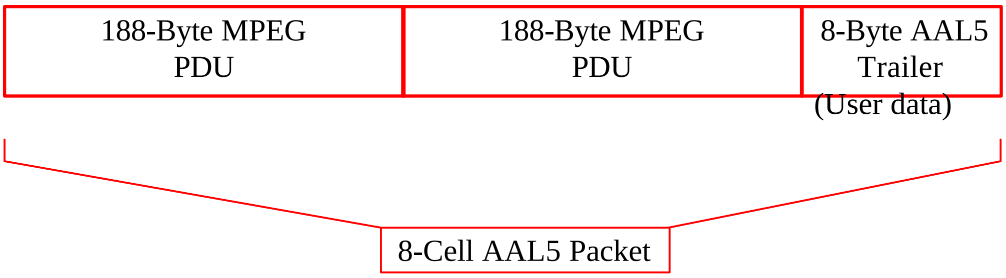
Note that slots containing OAM-F5 cells or idle cells for a given VC may be posted in the middle of AAL5 or MPEG packets (rather than between packets) for the same VC, but they must be posted at 48-byte boundaries within the packets. If TC35856BF detects an idle slot or an OAM-F5 cell for a given VC incorrectly posted (within an AAL5 or MPEG packet for the same VC, but not at a 48-byte boundary), it will generate a Tx_Fatal_Err interrupt and halt transmission of data. A reset of TC35856BF will then be required to re-initiate transmission.

This restriction imposes somewhat of a burden on the driver when posting these types of slots within AAL5 or MPEG packets. It is therefore recommended that OAM-F5 cells and idle cells of a given VC be posted only *between* packets of the same VC whenever possible.

3.3.4 MPEG Super-packets

TC35856BF provides support for transmission of data encoded according to the MPEG standard through the use of ***MPEG Super-Packets.*** MPEG PDUs are exactly 188 bytes in length. This means that 2 MPEG PDUs (376 bytes) may be packed together into a single 8-cell AAL5 packet with no AAL5 padding, as illustrated in Figure 3.-4.

Figure 3.-4 - AAL5 Packet Carrying 2 MPEG PDUs



An MPEG super-packet is a packet provided by the driver which may contain any **even** number of MPEG PDUs on a VC which has been specified as an MPEG VC (in the tx VC state table). TC35856BF will automatically segment the packet into 1 or more AAL5 packets, placing 2 MPEG PDUs in each AAL5 packet.

The driver should **not** provide any AAL5 information in MPEG super-packets (as opposed to AAL5 packets, which must include the AAL5 pad, control and length fields and a 4-byte CRC placeholder). For example, if the driver wishes to transmit 10 MPEG PDUs in a single MPEG super-packet, it must post an MPEG super-packet exactly 1880 bytes in length (188 x 10). TC35856BF will segment the super-packet into 5 AAL5 packets and append the appropriate AAL5 length (376), control (0) and CRC fields to each AAL5 packet.

As with other AAL5 packets, MPEG super-packets may be posted across any number of slots.

3.3.5 Segmentation Engine Control word

Segmentation engine control word is a new feature of TC35856BF. The word is used to specify detail behavior of segmentation engine. Currently, control word is defined for generation of AAL5 trailer. In the future, the control may used for other purpose. Therefore the fields have to be specified value as described below.

(a) AAL5 trailer generator

This word is used for generate AAL5 trailer such as PAD, UU, CP, Length and CRC-32 fields. The control word has to be post just after posting tx slot for user information of AAL5 packet. Namely, tx slot for user information should not have PAD and AAL5 trailer information at all. When TC35856BF detects AAL5 trailer generator after use information, it puts appropriate size of PAD field, and copy UU, CPI and Length filed which are specified, and calculates CRC-32.

Length field should be proper value for the AAL5 packet. TC35856BF doesn't evaluate that length filed matches user information of AAL5 packet.

Note that EOP bit of tx slot for user information should be '0' when AAL5 trailer generator follows it. Whenever EOP bit is "1", the tx slot is treated as the end of AAL5 packet including AAL5 trailer instead of the end of user information.

An ***AAL5 trailer generator*** is shown in Figure 3.-5, and a description of each of the fields in the descriptor follows.

Figure 3.-5 - AAL5 trailer generator

	1 bits	3 bits	12 bits	16 bits
LW0:	Cntl=1	Cmd	VC	Subcmd=0x4000

	UU	CPI	LENGTH
LW1:	UU	CPI	LENGTH
	8 bits	8 bits	16 bits

- **Cntl**

  This 1-bit field should be '1' in order to indicate segmentation engine control word.

- **Cmd**

  This 3-bit field should be '000' in order to indicate AAL5 trailer generator.

- **VC**

  This 12-bit field indicates the VC of the data contained in the transmit slot in host memory.

- **SubCmd**

  This 16-bit field should be 0x4000 in order to indicate AAL5 trailer generator.

- **UU**

  This 8-bit field is UU field of AAL5 packet trailer

- **CPI**

  This 8-bit field is CPI field of AAL5 packet trailer

- **LENGTH**

  This 16-bit field is LENGTH field of AAL5 packet trailer

3.3.6 TX Free and Active Slot Lists

As tx slot descriptors are written to TC35856BF, they are placed in off-chip memory in ***tx active slot lists.*** The tx active slot lists are **per-VC** FIFO lists of tx slot descriptors, where each descriptor is associated with a single tx slot in host memory.

The maximum size of tx active slot lists depends on VC_mode.

256VC: 256 entries

1KVC : 1K entries

4KVC : 4K entries

The head and tail pointers for each of the lists are 12-bits wide, and are kept off-chip. Storage for each entry in the active slot lists is taken from a structure known as the ***tx free slot list***, also located in off-chip SRAM.

The maximum size of tx free slot list depends on VC_mode.

256VC: 256 entries

1KVC : 1K entries

4KVC : 4K entries

Entries are taken from the head of the free slot list, and placed on the tail of the list.

Figure 3.-6 illustrates the tx free slot list, while Figure 3.-7 shows a single entry from the list. The **link** field in the tx free slot list entry is simply a 12-bit pointer to the next slot descriptor in the linked list in which the descriptor is resident.

Figure 3.-6 - Tx Free Slot List

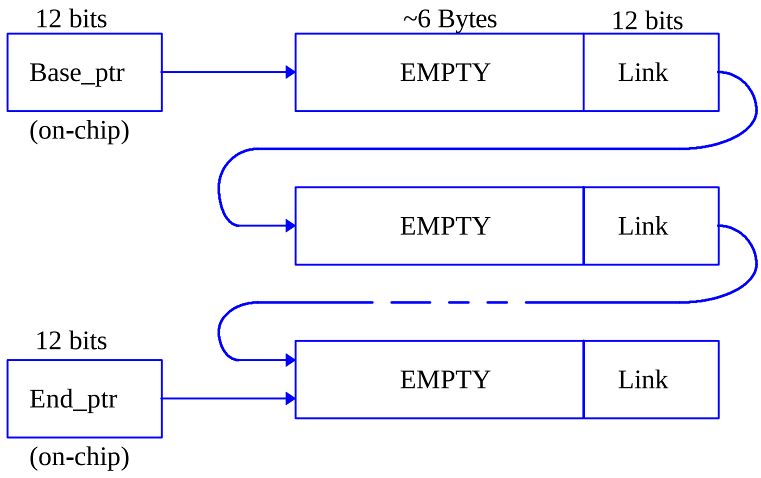


Figure 3.-7 - Tx Free Slot List Entry

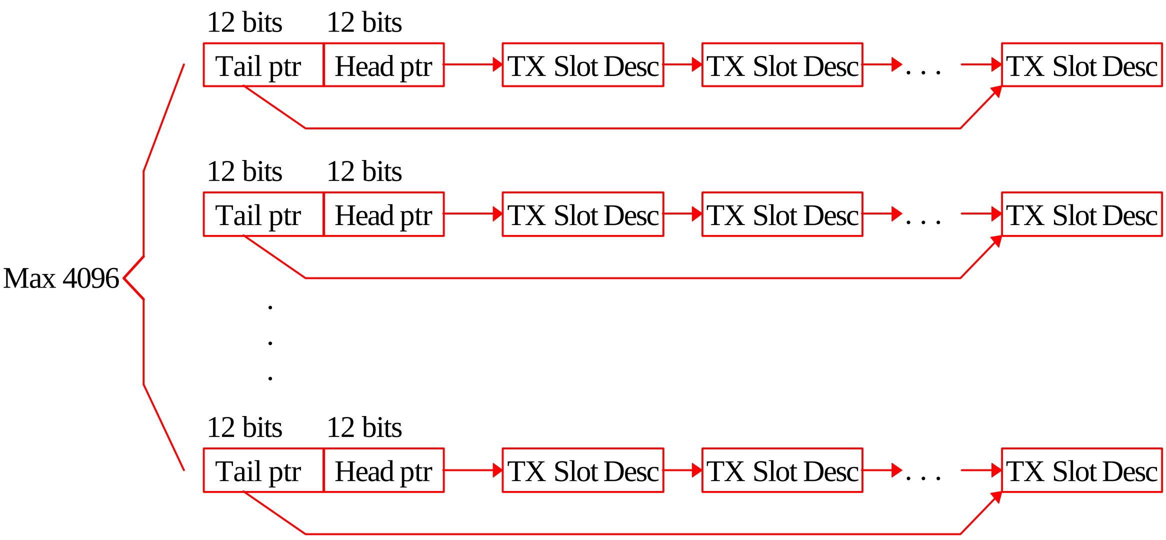
	1 bit	1 bit	1 bit	1 bit	12 bits
Word 0	Cntl	RAW	IDLE	MGMT	LINK
Word 1	CRC10_EN	EOP	SLOT_SIZE		
Word 2	BASE_ADDRESS (Low Word)				
Word 3	BASE_ADDRESS (High Word)				

When TC35856BF queues a tx slot descriptor to an active slot list, it uses the **VC** provided by the driver to determine the active slot list on which the descriptor should be placed. Figure 3.-6 illustrates the tx active slot lists.

In summary, for every slot posted for transmission by the driver, a free slot list entry is taken from the free slot list and posted to one of the active slot lists. Once the data in a slot has been transmitted, the slot descriptor associated with the slot will be returned to the free slot list by TC35856BF. The driver must keep a count of how many TOTAL slots (across all VCs) have been queued but not yet transmitted, and ensure that this number never exceeds the total number of slot descriptors supported by TC35856BF. This number is obviously determined by the size of the free slot list, which depends on VC_mode.

If the tx free slot list is completely emptied, TC35856BF will stop processing slot descriptors written by the host. The host will then be held off until the free slot list is no longer empty. This will result in wasted PCI bus bandwidth, and reduced performance.

Figure 3.-8 - Tx Active Slot Lists



Recall that a tx slot is a block of physically contiguous host memory. Strictly speaking, a tx slot contains only data. The bits describing the tx slot and contained in the tx slot descriptor do not appear in the tx slot itself. However, for simplicity, the remainder of this document may refer to the bits in the tx slot descriptor associated with a tx slot as part of the tx slot itself. For example, the document may refer to “the EOP bit of the slot”, in order to avoid specifying “the EOP bit of the slot descriptor associated with the slot”.

Both the tx free slot list and the tx active slot lists are initialized by the driver at initialization time. Initialization of the tx free slot list consists of writing every link in the tx free slot list entries (4K - 1 links) to create a linked list, and writing the on-chip head and tail pointers of the list. Initialization of the tx active slot lists consists of writing the head and tail pointers of all the lists (8K pointers) with the value 0, since all tx active slot lists are initially empty.

3.3.7 Tx Data FIFO

TC35856BF implements an on-chip transmit data FIFO which provides buffering for up to approximately 19 AAL5 cells to be transmitted on the link. At STS-3c/STM-1 line rates, this equates to ~54 μs of PCI latency buffering.

3.3.8 Tx DMA Scheduling

TC35856BF uses an approach to DMA scheduling in which flows are represented by their VCs in a table (the schedule table). Each entry in the schedule table represents a given point in time, and may contain a single CBR VC. If no CBR VC is contained at a given schedule table entry, an ABR or UBR VC will be transmitted at the point in time represented by the entry.

TC35856BF scans the schedule table at a rate exceeding 1 entry per cell time (at STS-3c/STM-1 line rates), processing each entry in the table sequentially by reading the CBR, ABR and UBR VCs to be transmitted. As each entry is read TC35856BF issues PCI DMA cycles, reschedules ABR and UBR flows, updates VC state, and moves tx slot descriptors to the tx free slot list as necessary.

3.3.8.1 Prescale_val and Prescale_count

The schedule table supports CBR rates, ABR allowed cell rates or UBR peak rates in the range of approximately 36.6 Kb/s to 149.76 Mb/s. However, it may be desirable to achieve CBR rates or UBR peak rates down to approximately 8 Kb/s, and it is essential to support ABR rates down to 4.2 Kb/s (10 cells/s). In order to achieve these low rates without expanding the schedule table to an unreasonable size, 2 per-VC variables, ***Prescale_val*** and ***Prescale_count***, in VC state are used.

Each of the 2 variables are 4 bits wide. Prescale_val specifies a scaling-down of the basic schedule-table implemented rate according to the following equation:

Rate = Schedule_table_rate / (Prescale_val + 1)

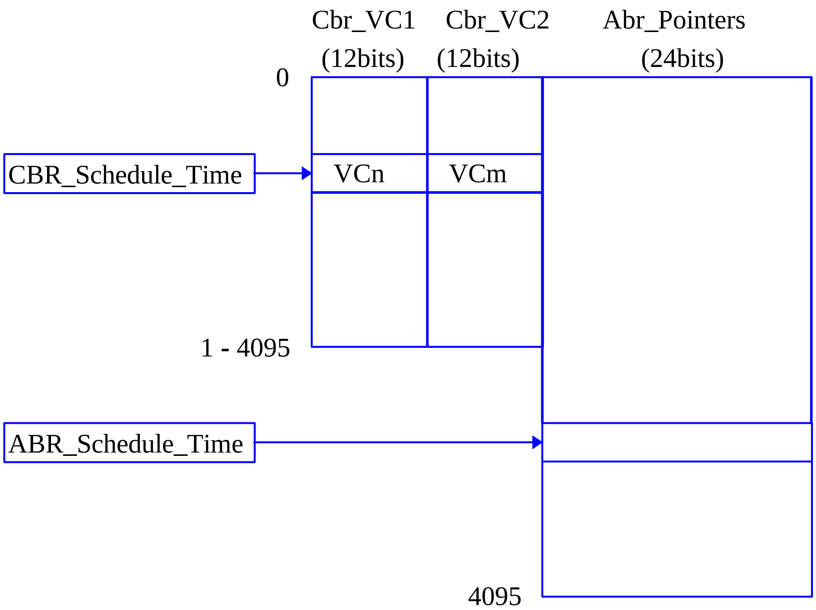
For example, if prescale_val for a VC is programmed with the value 0, then the rate specified on the VC will be equal to that implemented by the schedule table (described in the following sections). If prescale_val is programmed with the value 1, the rate on the VC will be half that implemented by the schedule table. Prescale_count is used by TC35856BF to perform the scaling-down operation.

For CBR flows, prescale_val is programmed by the driver and is never modified by TC35856BF. For ABR and UBR flows, prescale_val should be initialized to 0 by the driver. For all three types of flows, prescale_count should be initialized to 0 by the driver.

3.3.8.2 Schedule Table

The ***schedule table*** is a data structure located in off-chip SRAM. It contains from 2 to 4096 entries used to schedule CBR flows, and exactly 4K entries used to schedule ABR and UBR flows. Each entry is ~2 words (~4 bytes) wide. The schedule table is shown in Figure 3.-9.

Figure 3.-9 - Schedule Table



Each row in the schedule table represents a point in time exactly 1 cell time (at whatever rate the physical interface is running) from the adjoining rows. For example, at STS-3c/STM-1 line rates, the first row in the schedule table is associated with a time **t = 0**, the next row with a time **t » 2.83ms**, and the next row with a time **t » 5.66ms**. Conceptually, the schedule table should be thought of as a circular structure, such that the entry at row (4095) adjoins the entry at row (4094) **and** the entry at row 0 in the ABR table.

There are 3 separate columns in the schedule table, each of which is a word wide. Each of the 3 columns are described below:

3.3.8.2.1 CBR_VC1 and CBR_VC2

Each of these two fields contains a 12-bit **VC** value, along with an **active** bit and an **EOT** bit. Refer to Section 8.3.11 for an illustration of the relative locations of these fields within a schedule table entry. These fields indicate the CBR flow to be scheduled at the time at which the entry appears. When **active** is set, the VC in the entry is valid. Otherwise, there is no CBR VC to be scheduled at the time represented by the entry. The **EOT** bit is used to indicate the last entry in the table.

Only one **cbr_vc** column is used at one time. The column in use is selected by the driver through the **cbr_st_sel** bit in the **TC35856BF_Cntrl1** CSR. The reason that two **cbr_vc** columns are needed in the schedule table is to support modification of CBR rates in real-time by the driver. At initialization time, the driver will select the **cbr_vc1** column by setting the **cbr_st_sel** bit to 0, and initializing all entries in the column. When the driver wishes to modify the CBR schedule, it will write the entries in the **cbr_vc2** column with the new schedule, then toggle the **cbr_st_sel** bit to 1. TC35856BF will then start using the **cbr_vc2** column to schedule the transmission of CBR flows. If the driver wishes to modify the schedule once again, it will modify the **cbr_vc1** column, then toggle the **cbr_st_sel** bit back to 0.

The manner in which a given CBR rate maps into a given pattern of entries in the schedule table is quite straightforward. For every entry for a given VC in the schedule table, the entry will receive (1/(**st_size** * (prescale_val + 1)))th of the link bandwidth, where **st_size** is the size of the CBR schedule table in units of entries. Thus, if a VC appears only once in the schedule table, prescale_val for the VC is set to 0, and the schedule table contains 2119 entries, the VC will receive (1/2119) * 149.76 Mb/s = 70.67 Kb/s of bandwidth at STS-3c/STM-1 line rates. If the driver wishes to allocate one half of the link bandwidth to CBR VC J, it must fill half of the **cbr_vc** entries in the **cbr_vc** column being used with the value J, and program prescale_val for the VC with 0.

3.3.8.2.2 Abr_Pointers

This field contains pointers used by Meteor's internal schedule in scheduling ABR and UBR flows, in addition to an **active** bit for each entry. Refer to Section 8.3.10 for an illustration of the location of the **active** bit (which must be initialized by the driver) within a schedule table entry.

The **active** bit indicates whether or not the **abr_pointers** field in a given entry is valid. When the **active** bit is set, the field is valid. Otherwise there are no ABR or UBR VCs to be scheduled at the time represented by the entry, and the entry is ignored.

3.3.8.3 CBR, ABR, UBR and FLOWmaster Flows

It is important to differentiate between CBR flows, ABR flows, UBR flows, and flows subject to FLOWmaster flow control.

When a VC is opened for transmission and reception of data, it is defined to be in one of 3 classes:

- CBR
- ABR
- UBR

In addition, the driver must specify whether FLOWmaster flow control is to be applied to the VC. FLOWmaster flow control may be used on ABR or UBR VCs; it will never be used on CBR VCs.

Whether or not FLOWmaster flow control is used over an ABR circuit, the circuit is subject to the rate-based flow control mechanisms specified by the Traffic Management specification of the ATM Forum. The allowed cell transmission rate for such a circuit will therefore vary over time.

CBR circuits, however, have a fixed cell transmission rate which is specified at signalling time and does not change thereafter.

UBR circuits have a fixed peak cell rate which is also specified at signalling time and does not change thereafter. TC35856BF guarantees that this peak cell rate will not be exceeded under any condition, but it does not guarantee that this rate will be met.

Thus, the **cbr_vc** columns of the schedule table are never modified by TC35856BF except when they are being set-up by the host. The **abr_pointers** columns, however, are modified constantly by

TC35856BF, as ABR and UBR flows are scheduled and rescheduled according to their individual transmission rates.

Only one CBR VC may appear in a given schedule table entry (recall that each schedule table entry is associated with a “cell time”). Also, note that a given CBR VC may appear many times in a **cbr_vc** column.

The effect of FLOWmaster on a circuit from the DMA and transmit point of view is straightforward; if a circuit has a non-zero credit balance, it may DMA and transmit cells. If the credit balance for a circuit is 0, no cells for the circuit may be DMA’d or transmitted.

3.3.9 Tx Status Reports and the Tx Report Queue

As TC35856BF DMA’s data from tx slots in host memory, it must report to the driver the slots which have been consumed, so that the driver may recover and re-use the physical memory used by the slots. This reporting mechanism is implemented by means of a ***tx report queue*** in host memory. The tx report queue is a fixed-size structure located at a fixed location in host memory.

The size of tx report queue depends on VC_mode.

256VC and 1KVC : 4Kbyte(1K entries)

4KVC : 16Kbyte(4K entries)

It consists of 4KB or 16KB of contiguous physical host memory, and its location may be specified by the driver through the **Tx_Report_Base** CSR, but it must be 64-Byte aligned. It contains 1K or 4K entries, and each entry is 4 bytes wide. A tx report queue entry is illustrated in Figure 3.-10.

Note that the size of the tx report queue is chosen to be equal to the total number of tx slot descriptors supported by TC35856BF.

Figure 3.-10 - Tx Report Queue Entry

16 bits	12 bits	3 bits	1 bit
Reserved	VC	Reserved	Own

The various fields of a tx report queue entry are described below.

- **VC**

  This 12-bit field specifies the VC of the AAL5 packet or raw cell which was transmitted.

- **Own**

  Own bits are associated with each entry in the tx report queue to indicate whether the entry is owned by the adapter or by the host. The polarity of the own bits is initially set to 1 (owned by TC35856BF) after reset, and thereafter toggles every time TC35856BF fills the tx Report Queue. This scheme is chosen to allow the driver to determine ownership of a given entry in the queue without requiring driver writes to the queue

After the driver initializes the tx report queue by writing the **own** bits in all the entries in the queue to 1, it will never write the queue. Similarly, TC35856BF never reads the queue.

When TC35856BF is creating a DMA request for the last byte(s) of data in a slot, it will check if the slot is an end-of-packet (EOP) slot, or, for AAL5 packets, if the **Tx_AAL5_SM** bit in VC state is set. If either is true (the slot is an EOP slot or the slot is an AAL5 slot and the **Tx_AAL5_SM** bit is set), TC35856BF will DMA a status report entry into the tx report queue and generate an interrupt after the data DMA request has been serviced. This DMA operation is referred to as a ***tx status report***. For an AAL5 packet, the **EOP** bit will be set only for the slot containing the last byte of payload of the packet. For raw cells, the driver may choose to set the bit for every slot, or for every N slots per VC if it wishes to mitigate the frequency with which tx status reports take place. The per-VC **Tx_AAL5_SM** bit allows per-slot status report and interrupt generation for AAL5 cells. For raw cells, status report and interrupt generation may be controlled through use of the **EOP** bit in the slot descriptor.

After each tx status report, TC35856BF will generate a Tx_IOC interrupt. Tx_IOC interrupts are mitigated by a programmable holdoff timer and a programmable event counter, described in detail in Section 3.5 and Table 5.-9.

The driver will use the **VC** field in the tx Report Queue along with the FIFO order of packets within a VC to determine which packets have completed transmission. For example, the driver could maintain a per-VC count of packets which have been queued for transmission but have not yet been transmitted.

3.4 Receive Functional Description

3.4.1 Rx Slots

In order to receive packets, the host must allocate physically contiguous areas of host memory into which the packets should be DMA'd by TC35856BF. Each of these blocks of physically contiguous host memory is hereafter referred to as an ***rx slot***. A single AAL5 packet may be DMA'd into a single slot, or may span many slots. Each received raw cell will be contained in a single slot.

There are 3 types of rx slots:

- Small slots

  These slots are intended to receive AAL5 packets on VCs on which the maximum packet size is expected to be small. All small slots must be the same size, and the size is programmable in LW through the **Meteor_Cntrl2** CSR in the range 64B - 2KB.

- Big slots

  These slots are intended to receive AAL5 packets on VCs on which the maximum packet size is expected to be larger than will fit in small slots. All big slots must be the same size. The size is programmable through the **Meteor_Cntrl2** CSR to be 1K, 2K, 4K, 8K, 10K, or 16K bytes.

- Raw slots

  These slots are intended to receive 52-Byte raw cells and an accompanying status longword per cell. All raw cell slots must therefore be 56-bytes in length (they may be longer, but only the first 56 bytes of each slot will be used).

All rx slots must be longword-aligned in host memory. TC35856BF will completely fill rx slots, with the exception of raw cell slots over 56 bytes in length, and EOP AAL5 slots (since the last portion of an AAL5 packet may not exactly fill the last slot used by the packet).

When the driver opens a VC for reception of data, it specifies which type of slot the VC should use. Each VC will then receive data only into the selected slot type. The only exception to this rule are F5 OAM cells, which will be placed in raw slots, regardless of the slot type assigned to the VC over which they are being carried.

The driver may allocate as much memory as it wishes for rx slots, and any number of rx slots may be in use by the driver at any point in time. Architecturally, TC35856BF only limits the total number of slots into which data may be DMA'd at any one point in time to the number or VCs which it supports.

Figure 3.-11 illustrates an AAL5 packet which has been DMA'd into 3 rx slots in host memory. The rx slots are big slots, each of which is 2KB in length. The packet is exactly 4128 bytes in length, including the AAL5 pad, control, length and CRC fields.

Figure 3.-11 - AAL5 packet in 3 rx slots

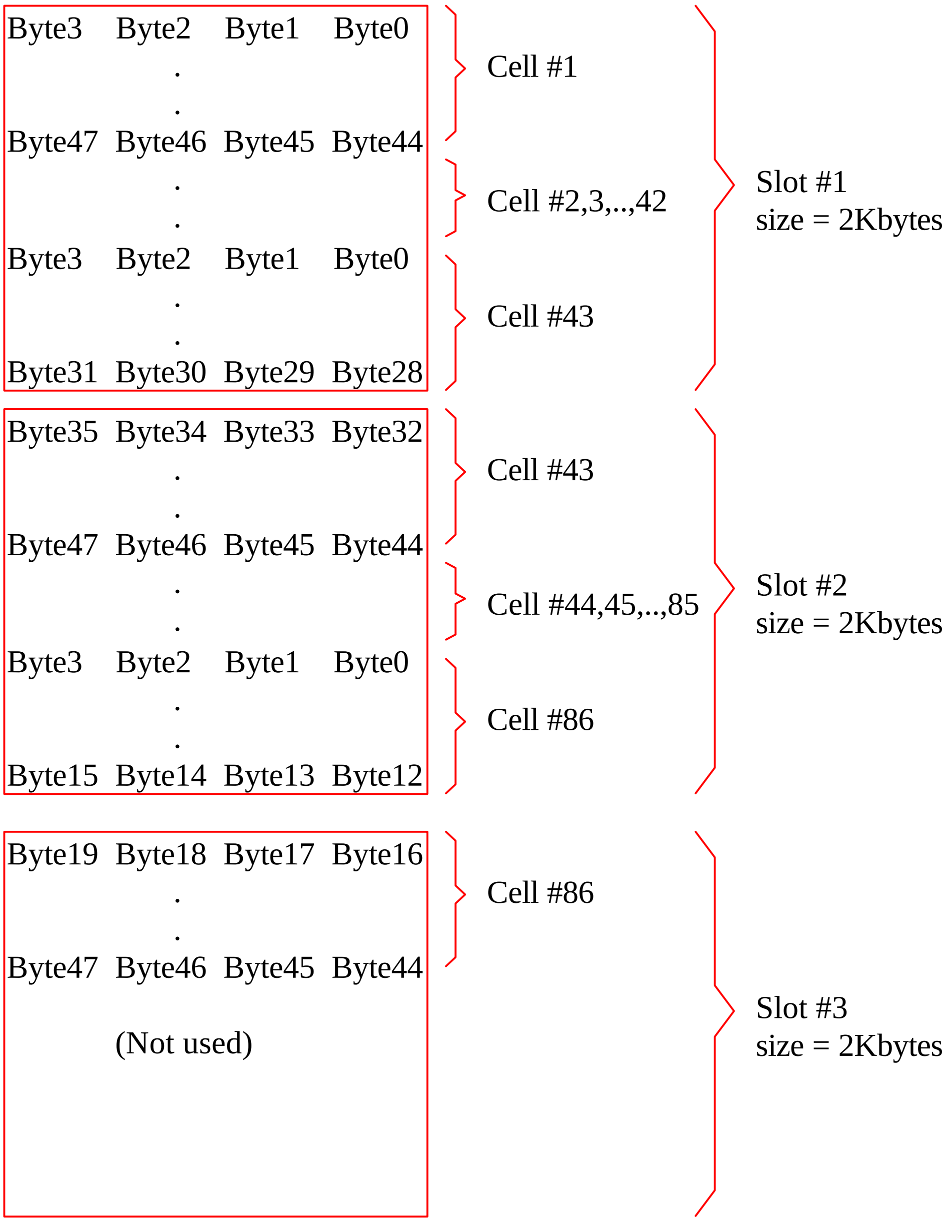


Figure 3.-12 illustrates the structure of raw cell which has been DMA'd in host memory. The raw slot has ATM header without the Header Error Correction byte, 48 Bytes data and the Rx Raw Cell Status Longword. The VC field in the Rx Raw Cell Status Longword is VC 0. The driver can check the VC number by reading VPI/VCI field of the ATM header in the raw slot. The driver can check PT in the ATM header.

Figure 3.-12 - Structure of RAW slot

H4	H3	H2	H1
Byte3	Byte2	Byte1	Byte0
Byte47	Byte46	Byte45	Byte44
Rx Raw Cell Status Longword			

Where, H1 to H4 are the header part of ATM header. Namely,

H1[7:0] = GFC[3:0], VPI[7:4]

H2[7:0] = VPI[3:0], VCI[15:12]

H3[7:0] = VCI[11:4]

H4[7:0] = VCI[3:0], PT[3:0]

The fifth byte of ATM header, HEC, is discarded by TC35856BF.

Payload part of ATM cell, 48 bytes, are stored after ATM header. And then Rx Raw Cell Status Longword is written. Please refer to section 3.4.5 Rx Raw Cell Status Longword.

3.4.1.1 Rx Slot_Tags

When the driver allocates a slot to receive data from TC35856BF, it must indicate to TC35856BF the *physical* address of the slot. Because the 2 types of AAL5 slots (big and small) are shared among VCs, and because both types of slots may hold several cells of data, the slots may be consumed (filled with data) in an order different to that in which they were posted. TC35856BF must thus report the consumption of *every* AAL5 receive slot in a manner which unambiguously identifies the slot being reported. Although the physical address of a slot uniquely identifies it, it is cumbersome in many operating systems to translate the physical address to the virtual address needed by the driver. Therefore, TC35856BF uses 14-bit ***rx Slot_Tags*** to simplify the driver's task in processing receive slots.

The driver must associate each receive slot in use with a unique 14-bit slot_tag (note that only 12 bits are necessary for this purpose, but 14 bits are provided for architectural scaleability). This slot_tag is then provided to TC35856BF along with the physical address of the receive slot, and when TC35856BF has placed data to be processed by the driver in the slot, it reports the identifying slot_tag to the driver. The driver may then use a simple table lookup to obtain the virtual address of the slot (along with any other necessary slot-specific information).

Slot_tags are not strictly necessary for rx raw slots, since these types of slots can hold only one cell, and are thus consumed in the order in which they are provided to TC35856BF. However, for simplicity, slot_tags are used for all types of rx slots used by TC35856BF, including raw slots.

3.4.2 Posting Rx Slots

Once the driver has allocated slots in host memory to receive data from TC35856BF, or when the driver wishes to re-use slots from which it has extracted received data, it must indicate to TC35856BF the location of the slots. In order to allocate a single slot for reception of data, the driver writes 2 LW to a 16-LW deep address space in TC35856BF. The write must be 2-LW (quadword) aligned. This address space is hereafter referred to as the ***rx pending slot*** address space. When the driver writes 2 consecutive LWs describing a single rx slot to this address space, the operation is referred to as ***posting an rx slot***. Note that the 2 LW writes are by no means guaranteed to occur in a single PCI burst; TC35856BF is designed to handle an arbitrarily long delay between the write of the first and the second LWs corresponding to the posting of a single rx slot. The rx pending slot address space is 16-LW deep in order to allow the posting of up to 8 rx slots in a single 16-LW PCI burst.

When the driver posts an rx slot for transmission, the first LW written to the rx pending slot address space contains a 12-bit VC, 2 bits indicating the type of slot being posted (small, big or raw), and a 14-bit slot_tag uniquely identifying the slot. The second LW written contains the 32-bit starting address of the slot being posted.

Rx free slots are not associated with particular VCs when they are queued by the host. The 12-bit VC which is written along with the slot type of each free slot is used solely for the purposes of handling rx congestion, as described below.

Whenever writes occur to the rx pending slot address space, the data written by the host is placed in an ***rx slot descriptor***. An rx slot descriptor is illustrated in Figure 3.-13, and a description of each of the fields in the descriptor follows.

Figure 3.-13 - Rx Slot Descriptor

	14 bits	2 bits	4 bits	12 bits
LW0:	SLOT_TAG	SLOT_TYPE	RESERVED	VC
LW1:	BASE_ADDRESS			

- **VC**

  This 12-bit field indicates the VC associated with the slot being posted. It is used for congestion-control purposes.

- **SLOT_TYPE**

  This 2-bit field indicates the type of slot being posted. It is decoded as follows:

  00 - Small
  01 - Big
  10 - Raw
  11 - Reserved

- **SLOT_TAG**

  This 14-bit field uniquely identifies the receive slot being posted. It is used by the driver to obtain the virtual address of a receive slot once data has been placed in the slot by TC35856BF.

- **BASE _ADDRESS**

  This 32-bit field indicates the physical base address of an rx slot in host memory. The least-significant 2 bits of this field should always be 0, since all rx slots must be longword-aligned.

3.4.3 RX Free Slot FIFOs

When TC35856BF processes rx slots posted by the host, it places the address associated with each slot into one of 3 off-chip FIFOs termed the ***rx free slot FIFOs.*** Each FIFO contains addresses for 1 slot type (small, big or raw). Each FIFO can hold up to 4K entries in 4K-VC mode, 1K entries in 1K-VC mode, or 512 entries in 256-VC mode. The head and tail pointers for each of the FIFOs are kept on-chip so as to conserve SRAM bandwidth. Entries are taken from the head of the FIFOs and written to the tail of the FIFOs.

Whenever a new rx slot is needed for reception of data, two bits in the VC state of the incoming cell indicate which rx Free Slot FIFO should be used to allocate an rx slot. However, all F5 OAM cells use raw Cell slots, regardless of the slot type which has been assigned to the VC over which they are being carried. The slot is then popped off of the correct FIFO, and its address is written to VC State in on-board SRAM.

An incoming cell will cause a new slot to be taken from one of the rx Free Slot FIFOs whenever it starts a new packet or when the previous slot allocated to the cell's VC has no more space left in it. Every raw or F5 OAM cell is considered a single-cell packet.

In summary, for every slot posted to receive data by the driver, a free slot FIFO entry is taken from one of the 3 free slot FIFOs. Once a slot has been filled with receive data, a driver-provided 14-bit slot_tag which uniquely identifies the slot will be reported to the driver (with the exception of raw cell slots, which may not all be reported; see Section 3.4.6). For each slot type, the driver must keep a count of how many TOTAL slots (across all VCs) have been posted to TC35856BF but not yet reported, and ensure that this number never exceeds the maximum size of the free slot FIFOs (4Kentries in 4K-VC mode, 1K entries in 1K-VC mode, or 512 entries in 256-VC mode).

If any of the rx free slot FIFOs is completely filled, TC35856BF will stop processing posted rx slots. Any further posts of rx slots to TC35856BF will be retried until the free slot FIFO is no longer full. This will result in wasted PCI bus bandwidth, and reduced performance. Since the head and tail pointers of the 3 free slot FIFOs are kept on-chip, they are initialized by TC35856BF at power-up or reset.

3.4.4 RX Data FIFO

TC35856BF implements an on-chip receive data FIFO which provides buffering for up to 4KB of receive data. At STS-3c/STM-1 line rates, this equates to approximately 200µs worth of PCI latency buffering.

3.4.5 Rx Raw Cell Status Longwords

Each 52-byte raw cell is DMA'd into host memory with an accompanying Raw Cell Status longword. The driver may then use the raw cell status longword to gain information about the raw cell itself and about the VC on which the raw cell has been received. A raw cell status longword appears as shown in Figure 3.-14. A description of each of the fields in the longword follows the figure.

Figure 3.-14 - Rx Raw Cell Status Longword

3 bits	14 bits	1 bit	1 bit	1 bit	12 bits
Reserved	**Slot_Tag**	**Cell_Loss**	**Slot_Cong**	**CRC10_Err**	**VC**

- **VC**

  This 12-bit field identifies the mapped VC of the raw cell.
  The bit is invalid when the RAW slot is used for over-ranged VC on 64K VC mode.

- **CRC10_Err**

  This 1-bit field when set indicates that a CRC10 check was done on the raw cell and an error was found. The driver should check the cell contains CRC10 field.
  The bit is invalid when the RAW slot is used for over-ranged VC when 64K VC mode is '1' and CRC10en_ovrrng_cell is '0'.

- **Slot_Cong**

  This 1-bit field when set indicates that the VC of the raw cell has experienced slot congestion. Cell loss may or may not have occurred as a result of the congestion.
  The bit is invalid when the RAW slot is used for over-ranged VC on 64K VC mode.

- **Cell_Loss**

  This 1-bit field when set indicates that the VC of the raw cell has experienced cell loss (of 1 or more cells). This cell loss may have occurred for 1 or more of the following reasons:

  - Rx free slot exhaustion
  - Rx data FIFO overflow
  - Slot congestion on the VC (for data cells in non-FLOWmaster mode)

The bit is invalid when the RAW slot is used for over-ranged VC on 64K VC mode.

- **Slot_Tag**

This 14-bit field contains the slot_tag of the receive slot into which the raw cell has been DMA'd. When the driver re-posts the slot which is used for over-ranged VC on 64K VC mode, the driver has to post to VC 0.

3.4.6 Rx Status Reports and the Rx Report Queue

As TC35856BF DMA's data into rx slots in host memory, it must report to the driver the slots which have been filled, so that the driver may read the data from the slots and re-post the slots to the appropriate free slot FIFOs. This reporting mechanism is implemented by means of an ***rx report queue*** in host memory.

The rx report queue is a fixed-size structure located at a fixed location in host memory.

The size of queue depends on **VC_mode**;

256 and 1KVC mode: 24K byte (3K entries)

4KVC mode : 96K byte (12K entries)

It consists of 24KB or 96KB of contiguous physical host memory, and its location may be specified by the driver through the **Rx_Report_Base** CSR, but it must be 64-Byte aligned. It contains 3K or 12K entries, and each entry is 2 longwords wide. A bit in the first longword of an entry identifies the entry as either an AAL5 report or a raw cell report. For an AAL5 report, the second longword contains useful information. For a raw cell report, the second longword is reserved . The two types of entries are illustrated in Figure 3.-15 and Figure 3.-16.

Note that the size of the rx report queue is chosen to be equal to the total number of receive slots supported by TC35856BF across all three free slot FIFOs.

Figure 3.-15 - Rx AAL5 Report Queue Entry

	16 bits	14 bits	1 bit	1 bit
Word0:	**Reserved**	**Slot_Tag**	**Raw = 0**	**Own**

Word1:	**EOP**	**SOP**	**Bad**	**Status**	**CLP**	**EFCI**	**Size**	**VC**
	1 bit	1 bit	1 bit	3 bits	1 bit	1 bit	12 bits	12 bits

Figure 3.-16 - Rx Raw Cell Report Queue Entry

	16 bits	14 bits	1 bit	1 bit
Word0:	**Reserved**	**Slot_Tag**	**Raw = 1**	**Own**
Word1:	**Reserved**			

- **Raw**

This 1-bit field indicates if the status report is for a raw cell slot (1), or for a slot containing AAL5 data (0). The AAL5 slot may be either a small slot or a big slot. When this bit is set, indicating that the slot is a raw cell slot, the second longword of the entry has no significance.

- **Own**

Own bits are associated with each entry in the rx report queue to indicate whether the entry is owned by the adapter or by the host. The polarity of the own bits written by TC35856BF is initially set to 0 after reset, and thereafter toggles every time TC35856BF fills up the rx Report queue. The driver initializes the rx report queue by writing “1” in the own bit of all the entries of the queue (1 = owned by TC35856BF). After that, the driver never write the queue. This scheme is chosen to allow the driver to determine ownership of a given entry in the queue without requiring driver writes to the queue

- **Slot_Tag**

This 14-bit field contains the driver-generated slot_tag which uniquely identifies the receive slot being reported. The driver uses this field to find the virtual address of the receive slot.

- **VC**

This 12-bit field indicates the VC of the data which has been placed in an AAL5 slot. The driver uses this field to reconstruct received AAL5 packets which are spread out among multiple slots.

- **Size**

This 12-bit field indicates the number of longwords which are valid in the slot being reported. This field will usually be equal to the size in longwords of the slot being reported, except for end-of-packet slots (slots containing the last byte of an AAL5 packet), which will probably not be completely filled. Also, the value 0x000 is used to represent exactly 16KB of valid data. The driver uses this field to determine packet boundaries in end-of-packet slots. This field is valid only if the Bad bit (described below) is **not** set.

- **EFCI**

This 1-bit field is either the logical OR of all the EFCI bits in the cells which make up a received AAL5 packet, or the EFCI bit in the last cell received in the packet. The selection of which of the two modes is to be used is made through a bit in VC state. This bit is of significance only for end-of-packet slots.

- **CLP**

This 1-bit field is the logical OR of all the CLP bits received in the cells which make up a received AAL5 packet. This bit is of significance only for end-of-packet slots.

- **Bad**

  This 1-bit field indicates if the packet being reported should be processed normally by the driver (0) or if an error condition was experienced while receiving the packet which should result in the discarding of the packet (1). When this bit is set, the Status field indicates the reason why the packet should be discarded. When this bit is cleared, the Status field gives information about the VC on which the packet was received. Table 3.-1 summarizes the use of the **Bad** and **Status** bits.

- **Status**

  When the **Bad** bit is set, this 3-bit field indicates the status of the packet in the slot being reported. When the **Bad** bit is cleared, this field gives information about the VC on which the packet was received. Table 3.-1 summarizes the use of the **Bad** and **Status** bits. Refer to 3.4.7 for detail description.

*Table 3.-1 - **Bad** and **Status** Bit Decoding in Rx AAL5 Status Reports*

BAD	STATUS	MEANING
0	XXX	Status<0> = Slot congestion experienced Status<1> = Packet loss experienced Status<2> = Reserved
1	000	CRC-32 error, discard packet
1	001	Length error, discard packet
1	010	Aborted packet, discard packet
1	011	Slot congestion cell loss, discard packet
1	100	Other cell loss, discard packet (Rx data FIFO overflow or Rx free slot exhaustion)
1	101	Reassembly timer time-out, discard packet
1	110	Packet too long error, discard packet
1	111	Reserved

- **SOP**

  This 1-bit field indicates if the slot being reported contains the first byte of an AAL5 packet. It is used by the driver to detect error conditions which may not have been reported due to lack of slots (receive congestion).

- **EOP**

  This 1-bit field indicates if the slot being reported is an end-of-packet slot. It is used in normal operation by the driver to find AAL5 packet boundaries.

After the driver initializes the rx report queue by writing the **own** bits in all the entries in the queue to 1, it will never write the queue. Similarly, TC35856BF never reads the queue.

When an AAL5 slot has been filled with data, TC35856BF creates an AAL5 status report and places it in the rx data FIFO, whether or not the slot was an end-of-packet slot. It is necessary to report **all** consumed AAL5 slots on receive (as compared to reporting only EOP slots, as may be done in the transmit direction), because slots are shared among all open VCs. The driver must therefore be explicitly informed to which VC each slot has been allocated.

When a raw cell slot has been filled with data (a single 52-byte raw cell and its associated status longword), TC35856BF **will not** necessarily create a status report. Rather, when TC35856BF detects that raw cell data is to be DMA'd into a raw cell slot, it will copy the information necessary to create a raw cell status report into a set of registers. This set of registers will be over-written with new data every time a new raw cell is seen by TC35856BF. Whenever a programmable holdoff timer or event counter expires, the information in the registers is DMA'd to the rx report queue in the form of a raw cell status report if the contents of the registers has changed since the last raw cell status report.

Reporting raw cell slot consumption only on expiration of the holdoff timer or event counter mitigates the frequency with which raw cell status reports take place and improves performance. Note, however, that doing so means that not all consumed raw cell slots will be reported to the host. However, since raw cell slots are consumed (filled with data) in the order in which they were posted by the driver, the driver can easily deduce which raw cell slots have been consumed whenever a new raw cell status report appears in the rx report queue.

This implies that the driver must keep track of the slot_tags and virtual addresses of raw cell slots and the order in which the slots were pushed onto the rx raw cell free slot FIFO. This also implies that the VC of the cell with which each free slot was filled must be extracted from the cell header itself. The mapped VC and other status information for the cell may be extracted from the raw cell status longword (described in Section 3.4.5) DMA'd with the cell.

Rx_IOC interrupts are generated whenever one of the following occurs:

- One of the AAL5 entries written to the rx Report Queue contains an end-of-packet indication or is for a streaming mode VC.
- A raw cell entry is written to the rx Report Queue in host memory.

Whenever the driver is interrupted, it processes the rx Report Queue entries by grouping all AAL5 and raw cell slot_tags together by VC. Any AAL5 End-of-packet indications signify that a complete AAL5 packet has been received. Once the complete packet has been copied to user space or otherwise processed, all slots in which the packet resided may be reused.

3.4.7 Rx AAL5 Status

- CRC32 error

  When TC35856BF detects CRC32 error at the end of a packet, it sets CRC32 error in STATUS field in Rx AAL5 Status Reports. The last cell of the end of packet is DMA'd to the host even if CRC32 error is detected. The Rx AAL5 Status Reports are outputted after the DMA of the cell.

CRC32 error, Length error and/or Abort packet may be detected at the same time when TC35856BF receives a cell as an end of packet. In this case, CRC32 error is reported prior to Length error and Abort packet.

- Abort packet

When TC35856BF detects Abort by Length at the end of a packet, it sets Abort packet in STATUS field in Rx AAL5 Status Reports. TC35856BF regards the packet should be aborted when the Length in AAL5 trailer is 0. The last cell of the end of packet is DMA'd to the host even if Abort packet is detected. The Rx AAL5 Status Reports are outputted after the DMA of the last cell.

When TC35856BF detects Abort packet and Length error at the same time, Abort packet is reported prior to Length error. When Abort packet is reported, CRC32 of the packet is correct.

- Length error

When TC35856BF detects Length error at the end of a packet, it sets Length error in STATUS field in Rx AAL5 Status Reports. TC35856BF regards the packet has Length error when the PAD length is more than 47 Bytes. The last cell of the end of packet is DMA'd to the host even if Length error is detected. The Rx AAL5 Status Reports are outputted after the DMA of the last cell. When Length error is reported, CRC32 of the packet is correct and the packet is not Abort packet.

- Slot congestion cell loss and Slot congestion experienced

When a cell is dropped by slot congestion, TC35856BF sets Slot congestion cell loss in STATUS field in Rx AAL5 Status Reports. TC35856BF doesn't set Slot congestion cell loss to the VCs which are set to FLOWmaster mode because TC35856BF shouldn't drop cells by slot congestion in FLOWmaster mode.

Slot congestion is checked when TC35856BF receives a cell. When slot congestion is detected, the cell is not DMA'd to the host. If a congested VC has a slot in Rx VC state, TC35856BF outputs Rx Status Report. If the VC doesn't have a slot in Rx VC state, Slot congestion cell loss status is stored in Rx VC state.

Slot congestion cell loss in Rx VC state is reported after the VC is recovered from slot congestion. The report is outputted when a cell is received to the VC whose cell(s) were dropped. When slot congestion is fixed before the end of the packet which is being dropped by slot congestion, Slot congestion cell loss is reported.

On the other hands, after all the cells of the packet which had been dropped, Slot congestion experienced is reported. The Slot congestion experienced is reported when a cell which belongs to the other packets is received,

- Other cell loss and Packet loss experienced

When a cell is dropped by Rx data FIFO overflow or Rx free slot exhaustion, TC35856BF sets Other cell loss in STATUS field in Rx AAL5 Status Reports.

This cell loss information is stored in Rx VC state.

Other cell loss in Rx VC state is reported after the VC is recovered from Rx data FIFO overflow and Rx free slot exhaustion. The report is outputted when a cell is received to the VC whose cell(s) were dropped. When Rx data FIFO overflow and Rx free slot exhaustion are fixed before the end of the packet which is being dropped, Other cell loss is reported.

On the other hands, after all the cells of the packet which had been dropped, when a cell which belongs to the other packets is received, then Packet loss experienced is reported.

- Reassembly timer time-out

  When TC35856BF detects reassembly timer time-out by polling VCs, it sets reassembly timer time-out in STATUS field in Rx AAL5 Status Reports. Reassembly time out is checked if the function is enabled by Rx_RATO_Time in Meteor_Cntrl2 CSR.

  When TC35856BF finds that it cannot output Rx AAL5 Status Reports by slot congestion, Rx data FIFO overflow, or Rx free slot exhaustion even if the VC is reassembly time-out situation, reassembly time-out is not reported.

  When TC35856BF finds reassembly time-out with slot congestion experienced or packet loss experienced by polling a VC, it reports Reassembly timer time-out only.

  There is a special case to report Reassembly timer time-out. When a whole packet is dropped by Rx data FIFO overflow and TC35856BF finds that the VC has a slot which is not returned to the host, reassembly timer time-out may be reported in order to recover the trapped slot. When the reassembly timer time-out is reported in this case, Rx_AAL5_DropCellCnt is increment never the less this packet loss is reported.

- Packet too long error

  When TC35856BF detects a packet is too long by comparing to the specified AAL5 maximum length, it sets Packet too long error in STATUS field in Rx AAL5 Status Reports. Default maximum packet length is 65535 Bytes. The maximum length can be changed by AAL5_Max_Length_Enb CSR and Rx_Buff_Cntrl CSR.

  If TC35856BF receives a cell which is not the end of packet and the packet length reaches to the specified limit, TC35856BF regards the packet is too long. When TC35856BF regards the packet is too long, the received cell which is not the end of packet is dropped. And TC35856BF outputs a Rx AAL5 Status Report with packet too long.

3.4.8 RX Congestion Handling

As TC35856BF receives cells, it writes the data in the cells into slots taken from one of the three rx free slot FIFOs, depending on the VC of the cells, and on whether the cells are OAM F5 cells or not (recall that OAM F5 cells are always placed in raw cell slots, regardless of the VC on which they are being transmitted).

The maximum size of each of the three rx free slot FIFOs depends on VC_mode.

256VC: 512 entries

1KVC : 1K entries

4KVC : 4K entries

As the slots are taken from the free slot FIFOs, filled with data, and reported to the host through the rx report queue in host memory, the host will copy data from the slots and re-post the slots to the free slot FIFOs. However, the host will usually empty slots and re-post them to the free slot FIFOs only on packet boundaries. For raw cell slots, a "packet boundary" occurs at every slot. For AAL5 slots, however, packet boundaries occur every N slots, where N may be arbitrarily large, depending on the size of the incoming packets and the size of the slots.

Thus, if many VCs are active simultaneously, and there is a high degree of interleaving on the physical link, and the packets being received are large in comparison to the size of the rx slots (such that a single packet occupies many slots), then one of the rx AAL5 free slot FIFOs may be depleted of slots. It is assumed that this situation would never occur with raw cell slots, which may be reclaimed by the host and re-posted to the free slot FIFO on a slot-by-slot basis.

The situation where one or both of the rx AAL5 free slot FIFOs are becoming empty (less than 10% full, for example), is termed rx congestion. If either of the FIFOs becomes completely empty, slot exhaustion is said to have occurred, and if all the slots from the FIFO are occupied with partially assembled packets, then reassembly deadlock has occurred. At this point, one or more packets must be dropped as quickly and efficiently as possible, in order to reuse the slots which the packets are occupying to complete reassembly on other packets.

TC35856BF implements a simple algorithm in order to avoid rx congestion and to detect and recover from reassembly deadlock.

An 8-bit counter, termed the ***rx_slots_consumed*** counter is kept per VC in VC state. The counters are initialized to 0 by the host at initialization time. Whenever a new slot is taken from one of the free slot FIFOs, the rx_slots_consumed counter corresponding to the VC of the data to be placed in the slot is incremented by 1. Whenever a slot is returned to one of the free slot FIFOs by the host, the rx_slots_consumed counter corresponding to the VC indicated by the host is decremented by 1. (Recall that the host queues a slot to one of the free slot FIFOs by writing the 32-bit address of the slot followed by a 12-bit VC). Note that the rx_slots_consumed counters stick at 0 during decrementing, so any value may be written to the VC field of a write to the rx pending slot address spaces at initialization time (when all the rx_slots_consumed counters will be 0).

The **Rx_Slot_Cong_Th** register holds 3 ***rx_congestion_threshold*** values, each value associated with one of the 3 free slot FIFOs. These threshold values are used across all VCs. Whenever TC35856BF tries to allocate a new rx slot to a VC to receive a new cell for the VC, it will compare the rx_slots_consumed counter for the VC to the appropriate rx_congestion_threshold.

If the VC is at threshold at this time, the new cell will be dropped if the VC is non-FLOWmaster, or, if the VC is a FLOWmaster VC, a credit for the VC will not be returned, but instead will be accumulated in an 8-bit ***ack_count*** per VC. The credits in ack_count will not be returned until the VC is no longer congested (rx_slots_consumed drops below the congestion threshold).

When TC35856BF drops a cell for an AAL5 VC, it will continue to do so (even if the VC becomes un-congested) until the next AAL5 packet boundary. When congestion is detected on a VC, it is reported as soon as possible (at the next status report time) through the **status** field of the status report.

When the host sees that congestion on a VC has occurred, it knows that the VC has too many slots outstanding. At this point the host must discard the packet in which the bit was set if the VC is an AAL5 VC, and re-post the outstanding slots to the appropriate free queue as soon as possible. Re-posting the slots will cause the rx_slots_consumed counter to decrement below the congestion_threshold, and enable the VC to start receiving data again.

The mechanism described above ensures that a single or a few misbehaving VCs (which are either being serviced too slowly by the host or which are carrying large packets compared to the slot sizes) do not impact reassembly of more well-behaved VCs by consuming more than their fair share of free slots.

Even with the above mechanism, it is still possible that TC35856BF will encounter slot exhaustion and/or reassembly deadlock. For example, even if rx_congestion_threshold for a given free slot FIFO is set to a small value, such as 8, for 1K slots this means that as little as 128 VCs may consume all available slots. A mechanism is therefore provided to deal with this condition.

When reception of a cell requires taking a new slot from one of the free slot FIFOs and the free slot FIFO is empty, the cell will be discarded and an interrupt will be generated. In the case of AAL5 slots, the rest of the packet to which the cell belonged will be discarded. No attempt is made to report the VC which caused the exhaustion of slots, since no VC is directly at fault (unless that VC is above the congestion_threshold for the free slot FIFO which it is using, in which case a status report for the VC will eventually indicate the condition).

Note that the above scheme does not attempt to differentiate between "important" VCs and "less important" VCs when dropping cells due to congestion. Instead, it reports to the host all VCs on which data was lost due to congestion whenever possible. It is then the host's responsibility to respond to the situation through either reducing the number of VCs active at one time, increasing the rate at which it services receive traffic (the latter will help only when congestion is a result of slow host response to receive traffic), increasing the size of the rx slots, etc. Lastly, note that there is no way for the host to determine which VCs have lost data due to slot exhaustion.

3.4.9 AAL5 Reassembly Time-out

It is not guaranteed that every received AAL5 packet for which TC35856BF begins reassembly will complete. For example, if the cell containing the EOM bit for a packet on a given VC is lost due to data corruption by the network, reassembly of the packet will not complete until a subsequent packet is received on the VC. If no other packet is received on the VC for a long period of time, the rx slots being used to reassemble the first packet will go unrecovered for this time period, wasting host memory.

TC35856BF provides the capability to detect when no cells for an AAL5 packet being reassembled have been received for a programmable period of time. When this occurs, the packet will be “timed-out”, the remainder of it (if received) will be received as a new packet. Consequently, the packet will be reported with error.

A single global (across all VCs) reassembly time-out time may be specified through the **Rx_RATO_Time** field of the **Meteor_Cntrl2** register. This field is 4-bits wide, and provides for time-out times in the range of 386ms - 2.895s ± 193ms. If the field is written with a 0, the time-out function is disabled, such that AAL5 packets will never be timed-out.

Note that OAM F5 cells received on a given VC **are not** considered part of an AAL5 packet being reassembled on the VC for purposes of reassembly time-out.

Note that when this timer expires , TC35856BF sets at least EOP bits in the receive status report to 1. SOP in time-out report may not be set.

3.4.10 AAL5 Maximum Length Limitation

Several applications may specify maximum length of AAL5 packet. When a longer packet than specified length is received, the driver discards the packet. Therefore it is preferred to discard extra cells without using extra receive slot.

When host driver specifies the AAL5 maximum length by Rx_Buff_Cntrl CSR and AAL5_Max_Length_Enb CSR, TC35856BF discards the part of AAL5 packet which exceeds the specified length. By the function, TC35856BF can effectively use host memory.

3.4.11 64K VC mode

TC35856BF can receive a cell of which VPI/VCI is over ranged from VPI_VCI_mapping. This mode is called as “64K VC mode”, and enabled by setting 64K_Mode_Enb bit in Meteor_Cntrl_1 CSR.

For example, assume VPI_VCI_mapping is ‘000’(No VPI bits are used for identifying VC number) and VC_Mode_sel is ‘11’ (4K VC mode). The received cell is classified by VPI and VCI value as follows;

Cell_type_a) VPI is 0x00 and VCI is in the range of 0x0000 to 0x0FFF

Cell_type_b) VPI is 0x00 and VCI is bigger than 0x0FFF

Cell_type_c) VPI is not 0x00 and VCI is any value

When 64K_Mode_Enb is “0”, Cell_type_a is received as a normal cell and associated with Rx VC state table. Cell_type_b and cell_type_c are dropped and cause Rx_Unknown_VC interrupt.

However, 64K_Mode_en is set to “1”, Cell_type_a is received as normal cell, Cell_type_b and Cell_type_c are received as RAW cells. They don’t cause Rx_Unknown_VC interrupt. RAW cell format is described in 3.4.1. VC field of RAW status longword is set to “0x000” for indicating the cell is received as a over ranged cell. The driver may assemble those RAW cells in any type of AAL packet. When the driver returns the RAW slot into TC35856BF, VC value of Rx slot descriptor should be “0x000”.

CRC10en_ovrrng_cell bit in Meteor_Cntrl_1 CSR is used for controlling CRC10_Err bit in RAW status longword. When CRC10en_ovrrng_cell is set to “0”, CRC10_Err is always cleared. When CRC10en_ovrrng_cell is set to “1”, CRC10_Err reflects CRC10 checking. The driver software may check if the cell is a part of AAL3/4 packet. If so, CRC10_Err has proper value for AAL3/4 packet. If the cell is a part of AAL-NULL packet, CRC10_Err has no meaning. The driver decides how to use CRC10_Err bit.

3.5 Interrupts

3.5.1 Overview

TC35856BF supports PCI interrupts through an **Intr_Status** register and an **Intr_Enb** register. Each bit in the **Intr_Status** register represents a single interrupt source. When an interrupt event occurs, its corresponding bit in the **Intr_Status** register will be set. A driver write of the value 1 to the bit will clear it. A driver write of 0 to the bit will not modify its state.

Each bit in the **Intr_Status** register has a corresponding bit in the **Intr_Enb** register. When any bit in the **Intr_Status** register is set, an interrupt will be generated on the PCI bus if the corresponding **Intr_Enb** bit is set. If the corresponding **Intr_Enb** bit is cleared, no interrupt will be generated, although the **Intr_Status** bit will be set. This means that the driver may determine that an interrupt event has occurred by reading the **Intr_Status** register even if the **Intr_Enb** bit for the event is cleared, disabling the interrupt. The various interrupt events generated by TC35856BF and the structure of the **Intr_Status** and **Intr_Enb** registers are described in Table5.-7 and Table 5.-8.

3.5.2 IOC Interrupts

There is one class of interrupts which is treated slightly differently from all other interrupt sources. This is the class comprising the ***IOC*** (Interrupt on Completion) interrupts. There are 2 different interrupts which fall into this class, the Tx_IOC interrupt and the Rx_IOC interrupt. Each of these interrupts has corresponding bits in the **Intr_Status** and **Intr_Enb** registers, like all other interrupt sources. However, each of these interrupts has associated with it a ***holdoff timer*** and an ***event counter***, which serve to mitigate the rates at which the interrupts occur.

Whenever TC35856BF DMAs a status report for a Tx slot it sets the Tx_IOC bit in the **Intr_Status** register. Whenever TC35856BF DMAs a status report for an Rx AAL5 EOP slot, for an Rx raw slot, or for any Rx AAL5 slot for a streaming mode VC, it sets the Rx_IOC bit in the **Intr_Status** register. However, each of the 2 **Intr_Status** bits will generate an interrupt on the PCI bus only if the corresponding **Intr_Enb** bit is set **and either** the holdoff timer or the event counter for the interrupt source has expired.

The **Intr_Hldoff** register is used to specify the values of the Tx_IOC and Rx_IOC holdoff timers and event counters. The Tx_IOC holdoff time is specified through the **Tx_IOC_Hldoff_Time** field, and may be in the range 0 - 128ms, in units of 503.04µs. The Rx_IOC holdoff time is specified through the **Rx_IOC_Hldoff_Time** field and may be in the range 0 - 1ms, in units of 3.93µs**.**

The Tx_IOC event counter value is specified through the **Tx_IOC_Hldoff_Cnt** field, and may be in the range 0 - 127. A Tx_IOC “event” is the DMA of any Tx status report. Every time a Tx_IOC event occurs, the contents of the **Tx_IOC_Hldoff_Cnt** field is decremented. When the field reaches 0, the event counter has expired, and a Tx_IOC interrupt may be generated to the host.

The Rx_IOC event counter value is specified through the **Rx_IOC_Hldoff_Cnt** field, and may be in the range 0 - 127. An Rx_IOC “event” is the DMA of an Rx status report for an AAL5 EOP slot, a raw slot, or any AAL5 slot for a streaming mode VC. Every time an Rx_IOC event occurs, the contents of the **Rx_IOC_Hldoff_Cnt** field is decremented. When the field reaches 0, the event counter has expired, and an Rx_IOC interrupt may be generated to the host.

Once a holdoff timer or event counter has expired, it must be re-loaded by the driver if it is to be used to hold off the next interrupt.

In order to reduce the amount of PCI bandwidth consumed by raw slots, the driver may use the **Rx_Raw_Report_Holdoff** field of the **Meteor_Cntrl2** register to specify a value *N* by which raw cell reports should be prescaled. TC35856BF will perform a single raw cell status report for every *N* raw cells received. *N* may be either 1, 4, 16 or 32. Note, however, that received raw cells are counted by the **Rx_IOC_Hldoff_Cnt** counter whether or not they result in status reports. **This means that Rx_IOC_Hldoff_Cnt must be programmed with a value *greater than or equal to* the value N (N = 1, 4, 16, or 32) specified by Rx_Raw_Report_Holdoff in order to minimize spurious interrupts.**

3.6 Peephole Interface Functional Description

3.6.1 Overview

The Peephole interface in TC35856BF is used to provide the host with access to 2 classes of devices external to TC35856BF. These are:

- On-board SRAM. This is accessed through the ***SRAM peephole space***.
- Other on-board devices. This class will typically include, but is not limited to:
  - PHY device CSRs
  - LEDs
  - MAC address ROM
  - Configuration jumpers
  - PCI Subsystem Vendor ID and Subsystem ID

  These devices are accessed through the ***peripheral peephole space***.

When the host uses the peephole interface to access the SRAM peephole space, a single 16-bit read or write to SRAM is initiated through Meteor's SRAM interface. When the peephole interface is used to access the peripheral peephole space, Meteor's general-purpose peripheral interface is used.

2 registers constitute the host's interface to the peephole mechanism. These are the Peephole_Cmd register and the Peephole_Data register, described below.

Peephole_Cmd:	<31>	- Ph_Read	LW address: 0x2D
(RW)	<30>	- Ph_Prf_Access	
	<29:16>	- Reserved	
	<15:0>	- Ph_Addr	
Peephole_Data:	<31>	- Ph_Done (RO)	LW address: 0x2E
(RW)	<30:16>	- Reserved	
	<15:0>	- Ph_Data	

- **Ph_Read**

  This 1-bit field is used to indicate whether the host wishes to perform a read (1) or a write (0) through the peephole interface.

- **Ph_Prf_Access**

This 1-bit field is used to indicate whether the host wishes to perform an access to the on-board SRAM (0) through the SRAM interface or to a peripheral device (1) through the Peripheral interface.

- **Ph_Addr**

This 16-bit field is used to select a particular entry in the device being read or written through the peephole interface. When **Ph_Prf_Access** = 0, **Ph_Addr[15:0]** selects a **word** in external SRAM. When **Ph_Prf_Access** = 1, **Ph_Addr[15:0]** is output through the Peripheral Interface to select a peripheral device external to TC35856BF (e.g., PHY chip, LEDs) and, when necessary, an address within the device.

No error checking is performed on **Ph_Addr** to ensure that accesses are not out-of-range. For example, TC35856BF supports between 64KB and 512KB of on-board SRAM only. If only 64KB is implemented, accesses to the upper 64KB of SRAM peephole space (recall that **Ph_Addr** is a word address for SRAM accesses) will yield invalid data on a read and will perform no useful operation on a write.

- **Ph_Data**

This 16-bit field contains the address to be written to the device and location specified by the **Peephole_Cmd** CSR in the case of a write, and the data read from the device in the case of a read.

When accessing the on-board SRAM peephole space, all 16 bits of data specified in Ph_Data will be written to SRAM in the case of a peephole write, and a read will return 16 valid bits of data in the Ph_Data register. When accessing the peripheral peephole space, only the least significant 8 bits of data in the Ph_Data register may be written to a peripheral device, and a read will return only 8 valid bits of data in the Ph_Data register.

- **Ph_Done**

This 1-bit field indicates the status of the peephole interface. When the bit is cleared, it indicates that the interface is busy performing a peephole operation. When the bit is 1, it indicates that the interface is idle and a peephole operation may be initiated by the host. The bit is cleared by the writing of the **Peephole_Cmd** register, which will initiate a peephole command. It is set by TC35856BF when the command completes, at which time a **Ph_Done_Intr** interrupt is generated. If either the **Peephole_Data** or **Peephole_Cmd** register is written while the **Ph_Done** bit is cleared, TC35856BF will inhibit the write and generate a **Ph_Access_Err** interrupt, unless the **Ph_Test** bit of the **Test_Cntrl** CSR is set, as described below. Note that the value of the **Ph_Done** bit may not be modified by a host write to the **Peephole_Cmd** CSR, unless the **Ph_Test** bit is set, as described below (this is why the bit is shown as read-only).

The **Ph_Test** bit of the **Test_Cntrl** register allow the host to write the **Ph_Done** bit, and inhibits the starting of peephole cycles and the generation of **Ph_Access_Err** interrupts when the **Peephole_Cmd** register is written. This enables test code to fully test the 2 peephole registers without undesired side effects.

When the host performs a peephole operation, any access to on-board SRAM space results in a single access taking place across Meteor's SRAM interface. Any access to peripheral space results in a single access taking place across Meteor's general purpose peripheral interface. It is up to the board designer using TC35856BF to partition the 64KB of address space available through the peripheral interface to the various devices on the board.

Table 3.-2 illustrates a sample Peephole address decoding for a typical NIC (Network Interface Card) implemented using TC35856BF. In the sample decoding shown, the upper 3 bits of the address output on the Peripheral Interface are used to select a device on the board, and the least significant 7 bits are used to select a location within the device when necessary. This means that each device is mapped multiple times into its address space. The column labeled "Ph_Addr Bits Used to Address Device" indicate which bits in **Ph_Addr** are used to address different locations within a device. It assumes that only 127 bytes of address space are available in each of the PHY device and the MAC Address ROM, and the LEDs and Configuration Jumpers spaces each contain a single 4-bit wide register.

Table 3.-2 - Sample Peephole Address Decoding in Peripheral Space

Ph_Addr	Device Selected	Access Type	Ph_Addr Bits Used to Address Device	Ph_Data Bits Used
0x0000 - 0x1FFF	PCI Subsystem ID Registers	RW	Ph_Addr<1:0>	Ph_Data<7:0>
0x2000 - 0x3FFF	LEDs	RW	None	Ph_Data<3:0>
0x4000 - 0x5FFF	MAC Address ROM	RO	Ph_Addr<6:0>	Ph_Data<7:0>
0x6000 - 0x7FFF	Configuration Jumpers	RO	None	Ph_Data<3:0>
0x8000 - 0xFFFF	PHY Device	RW	Ph_Addr<6:0>	Ph_Data<7:0>

3.6.2 PCI Subsystem ID Registers

TC35856BF assumes that the first 4 locations of the peripheral peephole space are mapped to 8-bit wide registers containing the PCI Subsystem ID and Subsystem Vendor IDs as described in Section 3.7.4. Thus, these registers are visible through the peripheral peephole space at addresses 0 - 3.

For purposes of future expansion, addresses 4 - 31 of the peripheral peephole space should be considered reserved when mapping the space in a board design.

3.6.3 Host Utilization of Peephole Interface

When the host wishes to access the on-board SRAM or any other on-board device external to TC35856BF, it must use the peephole interface. This section describes in detail the manner in which these accesses take place.

3.6.3.1 Read Cycles

When the host wishes to read a location in on-board SRAM or an on-board peripheral device, it writes the **Peephole_Cmd** register with the address of the device and location within the device to be read, setting the **Ph_Read** bit to 1 and setting the **Ph_Prf_Access** bit to indicate whether SRAM or a peripheral device is to be accessed. It then polls the **Peephole_Data** register waiting for the **Ph_Done** bit to be set. Alternately, the **Ph_Done_Intr** interrupt may be used to interrupt the host when the peephole operation has completed.

Once the peephole operation has terminated, as indicated by the setting of the **Ph_Done** bit and the generation of the **Ph_Done_Intr** interrupt, the data returned by the operation may be read from the **Peephole_Data** register. In the case of an access to on-board SRAM, all 16 bits of data will be valid. In the case of an access to a peripheral device, only the least significant 8 bits will be valid, and the most significant 8 bits will be 0.

3.6.3.2 Write Cycles

When the host wishes to write a location in on-board SRAM or an on-board peripheral device, it must first write the **Peephole_Data** register with the data to be written. All 16 bits of data should be specified for writes to SRAM, while only the lower 8 bits need be specified for writes to a peripheral device. The **Ph_Done** bit in the **Peephole_Data** register may be written with any value, as it is not modified by host writes to the register in normal operation.

The host must then write the **Peephole_Cmd** register with the address of the device and location within the device to be written, setting the **Ph_Read** bit to 0 and setting the **Ph_Prf_Access** bit to indicate whether SRAM or a peripheral device is to be accessed. It must then poll the **Peephole_Data** register waiting for the **Ph_Done** bit to be set. Alternately, the **Ph_Done_Intr** interrupt may be used to interrupt the host when the peephole operation has completed.

Once the peephole operation has terminated, as indicated by the setting of the **Ph_Done** bit and the generation of the **Ph_Done_Intr** interrupt, the write operation has completed, and the peephole interface may be used for other accesses.

3.7 Peripheral Interface Functional Description

3.7.1 General Description

The Peripheral Interface (PRF) Block performs the following functions:

- Interfaces to the Peripheral devices, allowing write/read operations to a 64KB address space (16-bits of address, 8-bits of data).

This general-purpose Peripheral Interface allows the host to access various on-board devices. These may include, but are not limited to:

- PHY chip CSRs
- MAC address ROM
- LEDs
- Jumpers
- PCI Subsystem Vendor ID and Subsystem ID

3.7.2 PH to PRF Interface

The Peephole interface in TC35856BF is used to provide the host with access to the PRF block within TC35856BF. The PRF block in turn provides access to a variety of devices external to TC35856BF.

All 16 address bits provided by the peephole interface through the **Peephole_Cmd** register are transferred to the PRF and from the PRF to the Chip I/O. Only 8 of the data bits from the **Ph_Data** portion of the **Peephole_Data** Register are used by the PRF. Refer to Section 3.6 for a description of how the peephole interface is used to access off-chip devices through the PRF.

3.7.3 PRF Interface to Chip I/O Interface

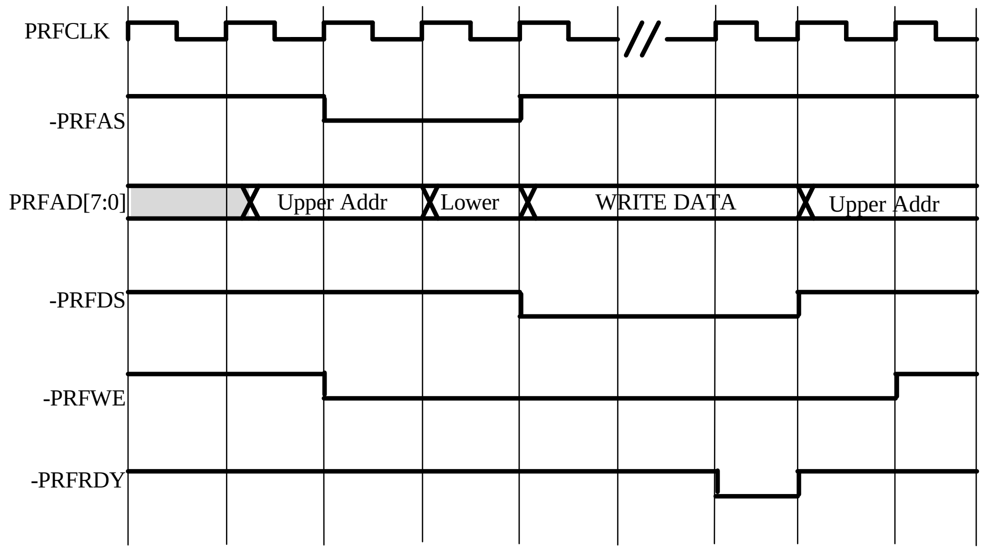
Two types of cycles are supported:

- 64KB read
- 64KB write.

The *PRFCLK* output is generated from Meteor's internal clock, which is a divided-down version of the 66.666MHz oscillator output required by TC35856BF. All PRF I/O timing is relative to *PRFCLK.*

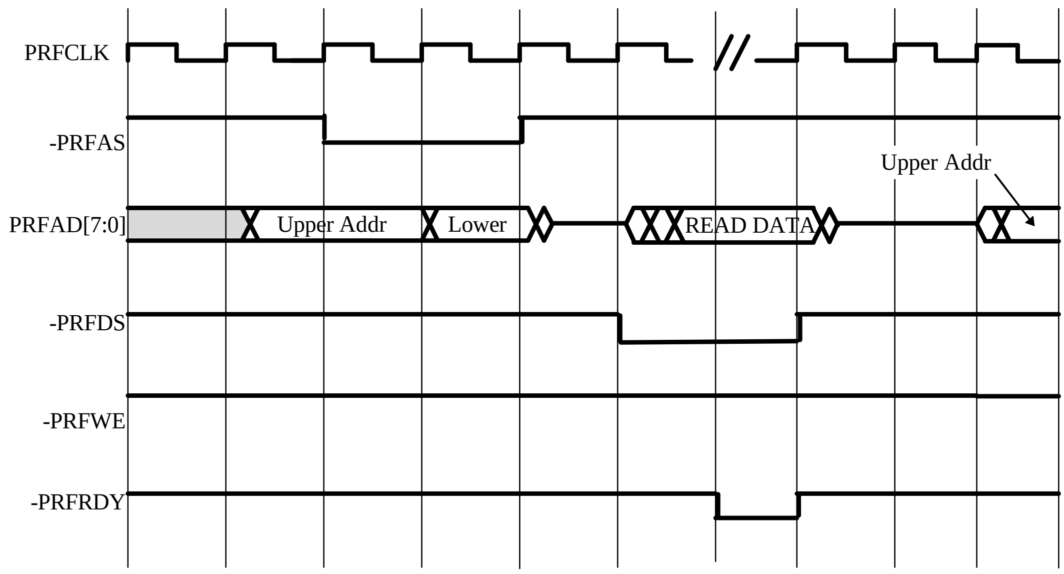
3.7.3.1 PRF Write Timing

Figure 3.-17 - PRF Write Cycle



3.7.3.2 PRF Read Timing

Figure 3.-18 - PRF Read Cycle



3.7.4 Peripheral Interface Access to PCI Subsystem ID Registers

In order for TC35856BF to support the PCI Subsystem ID and Subsystem Vendor ID fields of the PCI Configuration Space Header, the first 4 locations of the peripheral interface address space **must** be mapped to 8-bit wide registers containing the PCI Subsystem ID and Subsystem Vendor IDs as follows:

Table 3.-3 - PCI Subsystem ID Mapping into Peripheral Address Space

Address	Contents
0x0000	Subsystem_ID[7:0]
0x0001	Subsystem_ID[15:8]
0x0002	Subsystem_Vendor_ID[7:0]
0x0003	Subsystem_Vendor_ID[15:8]

Immediately following the deassertion of the PCI reset signal (hard reset), TC35856BF will automatically upload the PCI Subsystem and Subsystem Vendor IDs into its configuration block, assuming the above mapping and using normal Peripheral Interface cycles. Subsequent host reads to the Subsystem ID and Subsystem Vendor ID fields of the PCI Configuration Space Header (in PCI Configuration Space) will then yield the correct values. If a host access to Configuration Space occurs before the IDs have been uploaded into TC35856BF, the access will be retried. Note that the registers containing the IDs may also be read through the Peephole Interface at addresses 0 - 3 of the Peripheral Interface Space, as shown above.

For purposes of future expansion, addresses 4 - 31 of the peripheral address space should be considered reserved when mapping the space in a board design.

3.7.5 Sample TC35856BF Module Interface

In this example, the upper 8 address bits are ignored on the module. The lower 256 byte addresses are then partitioned into two sub-blocks:

- The upper 128 bytes are used to access the PHY Chip CSRs.
- The lower 128 bytes are partitioned among the following devices, each of which needs only 1 to 4 bytes of address space:
    - PCI subsystem ID registers (Subsystem Vendor ID and Subsystem ID), mapped into addresses 0 - 3 as required.
    - LEDs
    - MAC address ROM

Also, addresses 4 - 31 are reserved (actually, the PCI subsystem ID registers are shadowed into these addresses) as required.

This allows for a simple module implementation. The upper address is not loaded into a staging register on the module, and the module ignores this cycle. The lower address is decoded according to the above description as shown in Table 3.-4. Note that this table differs somewhat from Table 3.-2, in order to illustrate the variety of address mappings possible through the PRF.

Table 3.-4 - Sample Module Address Map

Ph_Addr	**Device Selected**	**Access Type**	**Ph_Addr Bits Used to Address Device**	**Ph_Data Bits Used**
0x0000 - 0x001F	PCI Subsystem ID Registers	RO	Ph_Addr<1:0>	Ph_Data<7:0>
0x0020- 0x003F	LEDs	RW	None	Ph_Data<7:0>
0x0040 - 0x007F	MAC Address ROM	RW	None	Ph_Data<3:0>
0x0080 - 0x00FF	PHY Device	RW	Ph_Addr<6:0>	Ph_Data<7:0>

3.7.5.1 Mac Address ROM Access

Ph_Data<3> is used as the read return data from the EEROM. Only 1 bit may be read from the EEROM (a serial ROM) at a time.

Ph_Data<2> is used as the CS input to the EEROM. Ph_Data<1> is used as the DI input to the EEROM, driving the cycle type and address inputs to the EEROM through some combinatorial module logic which continues to drive this DI input to the EEROM when read cycles occur to this EEROM Address.

Ph_Data<0> is used as the SK input to the EEROM, driving high and low to force the EEROM through its states - it also goes through some combinatorial logic which continues to drive this SK input to the EEROM when a read cycle to this EEROM Address occurs.

The module logic causes a read to be initiated to the EEROM, even though software will write and read this PeepHole Address to read the EEROM. The software goes through a long sequence of reads and writes in order to cause the serial EEROM to update the output data bit with 16-bits of read return data for a particular address. Software can only read one data bit at this particular address at a time.

3.8 PCI Interface Functional Description

3.8.1 Cycle Code Support and Aliasing

Table 3.-5 - PCI Commands

-CBE[3:0]	Command Type	Comments
0000	Interrupt Acknowledge	Not supported.
0001	Special Cycle	Not supported.
0010	I/O Read	Supported as target.
0011	I/O Write	Supported as target.
0100	Reserved	
0101	Reserved	
0110	Memory Read	Supported as both master and target.
0111	Memory Write	Supported as both master and target.
1000	Reserved	
1001	Reserved	
1010	Configuration Read	Supported as target.
1011	Configuration Write	Supported as target.
1100	Memory Read Multiple	Aliased to Memory Read as target, supported as master.
1101	Dual Address Cycle	Not supported.
1110	Memory Read Line	Aliased to Memory Read as target, not supported as master.
1111	Memory Write and Invalidate	Aliased to Memory Write as target, not supported as master.

3.8.2 Configuration Mode

PCI compliant devices are required to provide 256 bytes of space for initialization and configuration. The first 64 bytes of which are in a predefined header region where fields are uniquely identified. The remaining 192 bytes are device-specific and only implemented when necessary.

Not all registers (or all bits in a register) in the predefined region are required to be implemented, but access to those bytes (or bits) is allowed. In TC35856BF, writing to and reading from a non-implemented register will be completed normally on the bus. In the first case the data is discarded and in the latter case a value of '0' will be returned.

All multi-byte numeric fields follow little-endian ordering. All registers can be accessed anytime the Host wishes, via Configuration Commands.

3.8.2.1 Predefined Configuration Registers

A layout of the first 64-byte predefined header portion of the configuration space is shown in Figure 3.-19.

Figure 3.-19 - Configuration Space Header

31	16	15	0	
Device ID		Vendor ID		00h
Status		Command		04h
Class Code			Revision ID	08h
BIST	Header Type	Latency Timer	Cacheline Size	0Ch
Base Address Registers				10h
				14h
				18h
				1Ch
				20h
				24h
Cardbus CIS Pointer				28h
Subsystem ID		Subsystem Vendor ID		2C
Expansion ROM Base Address				30h
Reserved				34h
Reserved				38h
Max_Lat	Min_Gnt	Interrupt Pin	Interrtupt Line	3Ch

3.8.2.1.1 Vendor ID

This field identifies the manufacturer of the device. 0x102F is used here to represent a division of Toshiba.

3.8.2.1.2 Device ID

This field identifies the particular device. 0x0020 is used here to represent TC35856BF.

3.8.2.1.3 Command

This register provides coarse control over a device's ability to generate and respond to PCI cycles. The register bits are defined as follows:

Bits	Read/Write	Descriptions
<15:10>		Reserved.
<09>	Read/Write	Fast back-to-back Enable. This function is not implemented. Writing to this bit has no effect and reading it will return a '0'.
<08>	Read/Write	-SERR Enable. Writing a '1' to this bit enables TC35856BF to assert SERR#. Writing a '0' turns the function off.
<07>	Read/Write	Wait Cycle Control. This function is not implemented. Write to this bit has no effect and reading it will return a ‘0’.
<06>	Read/Write	Parity Error Response. Write an ‘1’ to this bit enables TC35856BF to take normal actions when a parity error is detected. Writing a ‘0’ turns this function off.
<05>	Read/Write	VGA Palette snoop. This function is not implemented. Write to this bit has no effect and reading it will return a ‘0’.
<04>	Read/Write	Memory Write and Invalidate Enable. This function is not implemented. Writing to this bit has no effect and reading it will return a ‘0’.
<03>	Read/Write	Special Cycle. This function is not implemented. Writing to this bit has no effect and reading it will return a ‘0’.
<02>	Read/Write	Bus Master. Bus master bit is used for enabling PCI bus master operation. The bit is read as “0” after reset. Writing a “1” to this bit enables TC35856BF to output REQ#.
<01>	Read/Write	Memory Space. Writing an ‘1’ to this bit enables TC35856BF to response to memory space accesses. Writing a ‘0’ to turn this function off.

<00>	Read/Write	IO Space. Writing an ‘1’ to this bit enables TC35856BF to response to IO space accesses. Writing a ‘0’ to turn this function off.

3.8.2.1.4 Status

The Status register is used to record status information for PCI bus related events. Reading from this register returns the current state of events. Writing '1' to a bit (except read-only bits <26:25>) will reset its value.

The definitions of the register bits are defined as follows:

Bits	Read/Write	Descriptions
<31>	Read/Write	Detected Parity Error. This bit is set to indicate that a parity error was detected regardless the state of Parity Error Response bit.
<30>	Read/Write	Signaled System Error. This bit is set to indicate that a system error was detected.
<29>	Read/Write	Received Master Abort. This bit is set to indicate that a TC35856BF initiated transaction was terminated with Master-abort.
<28>	Read/Write	Received Target Abort. This bit is set to indicate that a TC35856BF initiated transaction was terminated with Target-abort.
<27>	Read/Write	Signaled Target Abort. This bit is set to indicate TC35856BF terminated a Host access with Target-abort.
<26:25>	Read/Write	DEVSEL Timing. These bits encode the timing for signal -DEVSEL assertion. TC35856BF returns the value ‘01’ when these bits are read. Writing to those bits will have no effect.

<24>	Read/Write	Data Parity Detected. This bit is set when all three conditions are met: 1. TC35856BF asserted signal -PERR itself or observed -PERR asserted. 2. TC35856BF was the bus master for the operation in which the error occurred. 3. The Parity Error Response bit is set.
<23>	Read/Write	Fast Back-to-Back Capable. This bit is always set to indicate that TC35856BF as a target is capable of supporting fast back-to-back transactions. This is a read-only bit. Writing to this bit will have no effect.
<22>	Read/Write	UDF Supported. This bit is always cleared to indicate that TC35856BF will not support User Definable Features. Writing to this bit will have no effect.
<21>	Read/Write	66 MHz Capable. This bit is always cleared to indicate that TC35856BF is not capable of running at 66.667Mhz of PCI clock. Writing to this bit will have no effect.
<20:16>	Read/Write	Reserved. Reading these bits will always return the value ‘0’. Writing to them will have no effect.

3.8.2.1.5 Revision ID

This register specifies the revision identifier. This value is 0x2 for TC35856BF.

3.8.2.1.6 Class Code

This read-only register is for identifying the function of the device. TC35856BF selects the value '020300h' for Class Code (bits<31:8>), which represents "ATM network controllers".

3.8.2.1.7 Cache line Size

Host access-only register that is to specify the system cache line size in units of 32-bit words. This is to help TC35856BF to determined whether to use Memory Read or Memory Read Multiple commands to read from the Host memory. When read data size is bigger or equal to cache line size, TC35856BF uses Memory Read Multiple or Memory Read Line.

3.8.2.1.8 Latency Timer

Host access-only register that is to specify the time (in terms of number of PCI bus clocks) that TC35856BF is allowed to be the PCI bus master in the presence of other bus requests.

3.8.2.1.9 Header Type

TC35856BF returns '00h' when this read-only register is read.

3.8.2.1.10 BIST

This register is not supported. Reading from this register will return '0'. Writing to this register has no effect.

3.8.2.1.11 Base Address Registers

These registers serve two purposes:

1. They allow the Host to determine how much address space in either memory space and/or IO space a device requests using the mechanism described in Section 6.2.5.1 of the PCI specification.

2. They allow the Host to assign physical address to each of the address spaces a device uses.

TC35856BF hardwires bit<7:0> of the first base address (offset 0x10) with the value 0x00 for 256 bytes of contiguous memory space, and prefetchable bit is set to “0” because TC35856BF doesn’t support burst read operation in salve mode. Bit<7:0> of the second base address (offset 0x14) with the value 0x01 for 256 bytes of contiguous IO space for its internal data structures that are accessible by the Host.

3.8.2.1.12 Cardbus CIS Pointer

This register is not supported. Reading from this register will return a ‘0’. Writing to it will have no effect.

3.8.2.1.13 Subsystem Vendor ID

Reading from this register will return the value of the subsystem vendor ID provided by the subsystem (NIC) vendor. Writing to it will have no effect.

3.8.2.1.14 Subsystem ID

Reading from this register will return the value of the subsystem ID provided by the subsystem (NIC) vendor. Writing to it will have no effect.

3.8.2.1.15 Expansion ROM Base Address

This register is not supported. Reading from this register will return a '0'. Writing to this register has no effect.

3.8.2.1.16 Interrupt Line

This read/write register is used by the Host to indicate which of the input pins on the system’s interrupt controller chip is connecting to Meteor’s interrupt line.

3.8.2.1.17 Interrupt Pins

This register indicates to the Host which of the interrupt pins a device is using. TC35856BF returns '01h' when this register is read to indicate that it interrupts the Host via the signal -INTA. Writing to this register will have no effect.

3.8.2.1.18 Min_Gnt

This read-only register allows a device to specify how long a burst period (in terms of 0.25 microseconds) it needs, assuming a clock rate of 33.333MHz.

TC35856BF returns 0x08 when this register is read.

Writing to this register will have no effect.

3.8.2.1.19 Max_Lat

This read-only register allows a device to specify how often the device needs to gain access to the PCI bus.

TC35856BF returns 0x10 when this register is read.

Writing to this register will have no effect.

3.8.2.2 Device Dependent Configuration Registers

The device dependent configuration region of the Configuration Space is not implemented.

3.8.3 Slave Mode

TC35856BF allows the Host to access its internal registers and data structures whose addresses are defined in Base Address 0 and/or Base Address 1.

The PCI Interface Block serves as the interface between the Host and TC35856BF's internal functional blocks. Its major function in this mode is to constantly monitor the activities on the PCI bus. Whenever there is an access targeted at the address described by Base Address Register 1 and/or 0, it will respond to the request by asserting -DEVSEL. The PCI Interface Block will then pass all 32 bits of address to the internal blocks and signal the direction of data transfer. It is the internal function blocks' responsibility to decode the address and decide whether to accept data on the internal data bus which is driven by the PCI Interface Block, or to present data to the PCI Interface Block.

The PCI Interface Block is capable to accept either a single data phase, or multiple data phases per access. It is also capable to issue Retry to the Host when it is not ready to accept any data transfer, or to issue Disconnect to the Host when it cannot accept all data the Host intend to transfer in a single access. Note that a PCI bus master is required to resume the data transfer in a separate access after it is being disconnected from the bus, or forced to retry the access.

The PCI Interface Block generates parity whenever it drives the PCI bus and checks both address and data parity when it is receiving. Upon detection of bad parity on a data phase it will assert the signal -PERR if the Parity Error Response bit in the Command Register is enabled. The PCI Interface Block will not assert -DEVSEL if there is bad parity detected during the address phase of a transfer, and if the -SERR Enable bit in the Command Register is enabled, it will assert -SERR.

3.8.4 Master Mode

In master mode the PCI Interface Block is capable of requesting the PCI bus and initiating transactions requested by Meteor's internal DMA engines. It will also handle abnormal situations such as parity error detection and handling, disruption of a transaction caused by target abort, target issuing retry, target failing to respond and itself being forced off the bus due to Latency Timer expiration.

To initiate an DMA access to the Host memory the TC35856BF internal DMA engines need to first obtain the ownership of the PCI Interface Block, then to present the base address, data transfer length, and type of operation to the PCI Interface Block. The PCI Interface Block will do its job to start the DMA transaction and notify internal DMA engines to present or accept data to or from the PCI Interface Block. If TC35856BF is forced off the bus before the intend transfer is completed, the PCI Interface Block will re-arbitrate for the bus and resume the operation, without even notifying the internal DMA engines.

3.9 PHY Interface Functional Description

3.9.1 Functional Overview

3.9.1.1 General Description

- Support Reversible UTOPIA
- Transmit ATM cells to the PHY chip over the UTOPIA interface (cell-level and octet-level support)
- Receive ATM cells from the PHY chip over the UTOPIA interface (cell-level and octet-level support)
- Perform internal Loopback
- Handle GFC Halt and Control-A functions to regulate transmit traffic flow
- Synchronize transmit to the link rate or internal timebase
- Idle / unassigned cell generation and suppression

3.9.2 PHY Clock

An external pin (-REVERSE) is used to select whether PHYCLK pin is input or output. PHYCLK of TC35856BF is used for both of transmission and reception of UTOPIA interface.

When -REVERSE is set to “H”, PHYCLK outputs a half frequency of CLK66 clock input. When -REVERSE is set to “L”, input clock frequency of PHYCLK should be in the range of 25.00 MHz to 33.33 MHz.

3.9.3 Normal / Reverse UTOPIA Mode

TC35856BF supports both of Normal UTOPIA mode and Reverse UTOPIA mode, which is controlled by Reverse_UTOPIA mode in Meteor_cntrl_1 CSR.

Reverse UTOPIA mode has following features:

- Support both of octet-mode and cell-mode
- Support single-PHY operation
- Support various multi-PHY operation
  - Polling operation (Cell-level handshaking with one TxClav/RxClav signal)
  - Direct status operation
  - Multiplexed status polling operation
    (Behavior of TC35856BF is the same as polling operation).
  - Two bit select signals for polling and selection in multi-PHY mode, for transmit and receive respectively. Multi-PHY address should be decoded externally.
  - Polarity of select signal input pin is programmable.
- Common Transmit/Receive Clock

Detail features of Reverse UTOPIA are described in the following section.

3.9.3.1 Connection to ATM layer device

The mapping of signals for reverse UTOPIA mode is shown in Table 3.-6.

Table 3.-6 - UTOPIA interface: Pin function on reverse UTOPIA mode

Signal name of ATM layer device	Direction	Pin name of TC35856BF
RxData	←	**TXDATA**
RxSOC	←	**TXSOC**
RxClav/RxEmpty*	←	**-TXENB**
RxEnb*	→	**TXCLAV**
RxAddr(decoded)	→	**PHYRXA**
TxData	→	**RXDATA**
TxSOC	→	**RXSOC**
TxEnb*	→	**RXCLAV**
TxClav/Txfull*	←	**-RXENB**
TxAddr(decoded)	→	**PHYTXA**

TC35856BF in reverse UTOPIA mode acts as if it is a PHY layer device.
The signals of RxClav/RxEmpty* and RxEnb* in UTOPIA specification are used for handshaking between ATM layer device and PHY layer device (TC35856BF in the reverse UTOPIA operation). -TXENB and TXCLAV pins of TC35856BF should be respectively connected to RxClav/RxEmpty* and RxEnb* pins of ATM layer device. -TXENB pin is high active in reverse UTOPIA mode. TXCLAV is low active in reverse UTOPIA mode. TXDATA and TXSOC pins of TC35856BF should be respectively connected to RxData and RxSOC of ATM layer device.

The signals of TxEnb* and TxClav/Txfull* in UTOPIA specification are used for handshaking between ATM layer device and PHY layer device (TC35856BF in the reverse UTOPIA operation). RXCLAV and -RXENB pins of TC35856BF should be respectively connected to TxEnb* and TxClav/Txfull* of ATM layer device. RXCLAV pin is low active in reverse UTOPIA mode. -RXENB pin is high active in reverse UTOPIA mode. RXDATA and RXSOC pins of TC35856BF should be respectively connected to TxData and TxSOC of ATM layer device.

PHYCLK pin of TC35856BF should be connected to both of TxClk and RxClk of ATM layer device.

3.9.3.2 Single-PHY operation

When both of MPHY_POLLING and MPHY_DIRECT in MPHY_Cntrl CSR are cleared, TC35856BF works as UTOPIA Level-1 single-PHY device. Either of cell-level handshake and Octet-level handshake is selected via Phy_Cell_Mode_Enb in Meteor_Cntrl_1 CSR. PHYTXA and PHTRXA inputs are ignored.

3.9.3.3 Multi-PHY operation

Multi-PHY operation is activated when MPHY_POLLING or MPHY_DIRECT bit in MPHY_Cntrl CSR are set. TC35856BF supports three type of MPHY configuration, which are shown in Figure 3.-20. However TC35856BF needs to decode TxAddr[4:0] and RxAddr[4:0] externally. Output of decoder is connected to PHYTXA and PHYRXA. Polarity of PHYTXA and

PHYRXA is controlled by MPHY_ADDR bit in MPHY_Cntrl CSR.

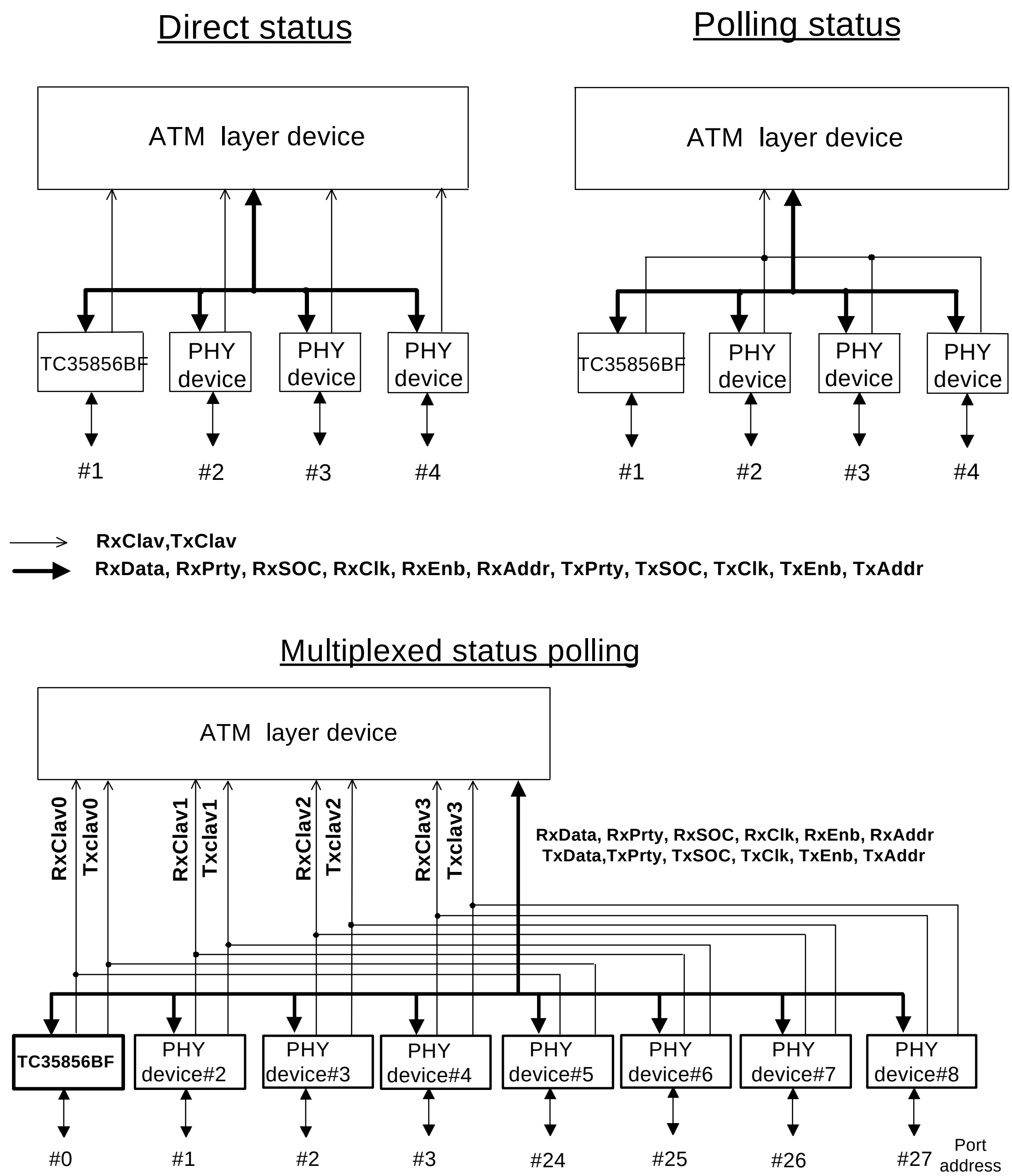


Figure 3.-20 - Multi-PHY configuration

Direct status operation is enabled when MPHY_DIRECT is set and MPHY_POLLING bit is cleared. TXCLAV/RXCLAV signal are permanently available for indicating the cell available status and they are not shared by the other device. External address decoder may decode the most significant 3 bits of TxAddr/RxAddr.

Polling status operation is enabled when MPHY_DIRECT is cleared and MPHY_POLLING bit is set. TXCLAV/RXCLAV signals are driven only during each cycle followed by active PHYTXA/PHYRXA . Those signals are tri-stated during a cycle followed by inactive PHYTXA/PHYRXA. External address decoder may decode all bits of TxAddr/RxAddr.

Multiplexed status polling is exactly same operation as polling status operation for TC35856BF. External address decoder may decode the most significant 3 bits of TxAddr/RxAddr. As the least significant 2 bit are not decoded, up to 4 PHY device including TC35856BF are selected. The devices which are selected at the same time are connected with separate TxClav/RxClav signals. But the devices which are not selected at the same time share the same TxClav/RxClav signals. For example, TC35856BF can share TxClav/RxClav signals with PHY device #5 (port address #24).

Cell transfer phase is same among above three operation mode. Cell transfer is started after selection phase. Selection phase is executed when TxEnb*/RxEnb* is deasserted during the current PHYCLK cycle and is asserted during the next PHYCLK cycle. The MPHY port is selected in a cycle starting from the cycle after PHYTXA/PHYRXA is asserted and TxEnb*/RxEnb* is deasserted; and ending in the cycle when a new MPHY port is addressed for selection and TxEnb*/RxEnb* is deasserted.

3.9.3.4 Setting Reverse UTOPIA mode

In order to act as a PHY device on UTOPIA bus, host should set not only Reverse_UTOPIA bit in Meteor_cntrl_1 CSR register but also another fields by the following order.

Step 1: Reset TC35856BF by hardware reset or software reset
Step 2: Set Phy_I/F_reset (bit8) of Meteor_Cntrl_1 to "1"
Step 3: **Set Reverse_UTOPIA bit in Meteor_Cntrl_1 to "1".**
Step 4: Reset Phy_I/F_reset (bit8) of Meteor_Cntrl_1 to "0"
Do not change Reverse_UTOPIA bit any more.
Step 5: Set other CSR registers as necessary.

3.9.4 Idle/unassigned cell selection

When there is no valid data cell for outputting from TC35856BF, TC35856BF generates either IDLE cell or UNASSIGNED cell according to Idle_cell_mode bit in Meteor_cntrl_1 CSR. When the bit is set, IDLE cell is generated. As TC35856BF acts as PHY device, Idle cell is preferable.

3.9.5 Suppressing IDLE cell generation

In Multi-PHY environment, idle/unassigned cell from TC35856BF causes waste of UTOPIA bandwidth. Suppress_Idle_cell bit in MPHY_Cntrl CSR is used for suppressing idle/unassigned cell generation.

When Suppress_Idle_cell is cleared, idle/unassigned cell is used for keeping data cell output rate. PHY device accepts cells at exact line rate. Transmit scheduler uses time tick from cell transfer to PHY. Therefore the time tick synchronizes to line rate.

However, when idle/unassigned cell is suppressed, there is no reference to synchronized to line rate. In order to keep cell output rate, TC35856BF uses "cell interval timer" which counts internal clock deriving from CLK66 input pin. Internal clock period is 30ns when 66.66 MHz is supplied to

CLK66. Cell interval of STS-3c/STM-1 (155.52Mbps) is 2.83119us. It is 94.373 time-tick in units of 30ns. In order to make cell interval timer close to 94.373, TC35856BF uses dual timer scheme. Cell_timer_1 and Cell_timer_2 in MPHY_Cntrl CSR are used alternatively. Namely, after cell_timer_1 is expired, then cell_timer_2 is stared. And after cell_timer_2 is expired, cell_timer_1 is started again. Average cell interval becomes (cell_timer_1 + cell_timer_2)/2.

It is recommended that cell_timer_1 and cell_timer_2 are set to 94 and 95, respectively. Average cell interval is 94.5. It is 0.135% slower than ideal cell rate.

Because the maximum count by cell_teimr_1 or cell_timer_2 is 255, those counters are not enough for slower line rate of which cell interval becomes longer than 255 clock ticks. Cell_timer_1 and cell_timer_2 are concatenated to give longer interval. Maximum cell interval is 65535 clock ticks, which is equal to about 1966us.

3.9.6 GFC Control

The GFC flow control functionality of TC35856BF is used to generate the GFC signals that are sent out with all transmit cells including idle cells, and to decode SET_A and HALT functions on cell receive to throttle transmit cell flow.

The PHY chip must be programmed to not filter out unassigned cells with GFC != 0.

3.9.6.1 GFC Overview

Two types of traffic are supported:

- controlled
- uncontrolled

Controlled traffic connections are controlled by the GFC fields - there are two groups, Group A and B, with TC35856BF only implementing the default Group A.

Uncontrolled traffic connections are not controlled by the GFC signals except for the HALT signals. Controlled traffic connections in TC35856BF are controlled by the GFC signals - HALT and SET_A. **In TC35856BF, ABR and UBR VCs are considered controlled traffic, whereas CBR VCs are uncontrolled traffic.**

Table 3.-7 - TC35856BF GFC Functionality Controlled by CSR Bits

phy_gfc_enb	phy_gfc_a_enb	Tx GFC	Rx GFC
0	1	Illegal	NA
0	0	0000	Ignore GFC
1	0	Uncontrolled = 0001	Decode GFC HALT
1	1	Controlled, Uncontrolled = 0X01	Decode GFC HALT and GFC SET_A

With *phy_gfc_enb* deasserted, TC35856BF ignores all GFC codes on receive and sets the transmit GFC to 0000..

Table 3.-8 - Receive GFC Field Decoding

GFC[3]	**GFC[2]**	**GFC[1]**	**GFC[0]**
HALT	SET_A	SET_B	0

Table 3.-9 - Transmit GFC Field Encoding

GFC[3]	**GFC[2]**	**GFC[1]**	**GFC[0]**
0	Controlled-A	0	Uncontrolled - GFC enabled

3.9.6.2 GFC Flow (normal, loopback and reverse)

The controlling equipment starts sending one of the HALT or SET_A/SET_B signals to determine the GFC capability of the controlled equipment for some period of time.

On start up, the connections start in uncontrolled mode, and scan the receive stream for GFC codes.

3.9.7 Internal loop-back mode

TC35856BF supports internal loop-back for testing transmit and receive function without PHY device. In the internal loop-back mode, transmit cell is bypassed to receive block just before outputting to UTOPIA bus. Therefore most of logic in TC35856BF is exercised. GFC control is also tested in this mode by Phy_GFC_Halt and Phy_GFC_Set_A bits in Test_Cntrl CSR.

In order to set loop-back, following setting sequence is recommended.

1. **Normal UTOPIA mode**

    Step 1: Reset TC35856BF by hardware or software reset
    Step 2: Set **Phy_Loopback_Enb (bit3), Phy_Chip_Reset (bit 6) and Phy_I/F_reset (bit8)** of Meteor_Cntrl_1 to “1”
    Step 4: Reset Phy_I/F_reset (bit8) of Meteor_Cntrl_1 to “0”
    Keep Phy_Loopback_Enb (bit3) and Phy_Chip_Reset (bit 6) as “1”.
    Step 5: Set other CSR register as necessary.
    Don’t change the value of Phy_Loopback_Enb (bit3) and Phy_I/F_reset (bit8).

2. **Reverse UTOPIA mode**

    In order to act as a PHY device on UTOPIA bus, host should set not only Reverse_UTOPIA of Meteor_cntrl_1 CSR register but also another fields by the following order.
    Step 1: Reset TC35856BF by hardware or software reset
    Step 2: Set **Phy_Loopback_Enb (bit3), Phy_Chip_Reset (bit 6),**

Reverse_UTOPIA (bit25) and Phy_I/F_reset (bit8)
of Meteor_Cntrl_1 to “1”

Step 3: Reset Phy_I/F_reset (bit8) of Meteor_Cntrl_1 to “0”
Keep Phy_Loopback_Enb (bit3), Phy_Chip_Reset (bit 6),
Reverse_UTOPIA (bit25) as “1”.

Step 5: Set other CSR registers as necessary.
Don’t change the value of Phy_Loopback_Enb (bit3),
Phy_Chip_Reset (bit 6) and Reverse_UTOPIA (bit25) .

3.9.8 UTOPIA operation timing

3.9.8.1 Transmit operation

(A) Octet level handshake

UTOPIA specification uses TxEnb* and TxFull* signals for handshaking. -TXENB and TXCLAV pins of TC35856BF should be connected to TxEnb* and TxFull* pins of the PHY device, respectively. PHYCLK output pin of TC35856BF should be connected to TxClk pin of the PHY device. TC35856BF indicates that it wants to output data by asserting -TXENB to low. The PHY indicates it can accepting at least 4 bytes valid data by deasserting TxFull* (asserting TXCLAV) to high. During a time period termed the transmit window, the PHY device stores data from TXDATA on the low-to-high transition of PHYCLK, if -TXENB is asserted. The transmit window exists from the time that the PHY device indicates it can accept data by asserting TXCLAV, until 4 valid write cycles after the PHY device deasserts TXCLAV. The PHY device may deassert TXCLAV at any time. After being deasserted this indicates that the ATM layer may transfer no more than 4 data words on TXDATA until TXCLAV us asserted again.

TC35856BF is required to deassert -TXENB within 4 data writes of TXCLAV deassertion and not to reassert -TXENB until TXCLAV is detected asserted. Inside the transmit window TC35856BF may assert and deassert -TXENB as required.

Handshaking timings for data cell and idle cell are different. Figure 3.-21 and Figure 3.-22 show the difference. Figure 3.-21 shows an example of data cell transfer using octet-level handshaking.

(1) In the middle of cell, the PHY may deassert TXCLAV. TC35856BF detects TXCLAV deasserted on clock edge 2 and deassert -TXENB 2 clock later. Therefore, 2 octets, P45 and P46, should be accepted by the PHY.

(2) On clock edge 6, TC35856BF recognizes that TXCLAV was asserted by the PHY and, as a consequence of this, it asserts -TXENB 2 clocks later and re-start to transmit data.

(3) After TC35856BF has output the last data of the cell, -TXENB is deasserted on clock edge 10.

(4) 1 clock later, TC35856BF re-samples TXCLAV to determine if the next cell can be output.

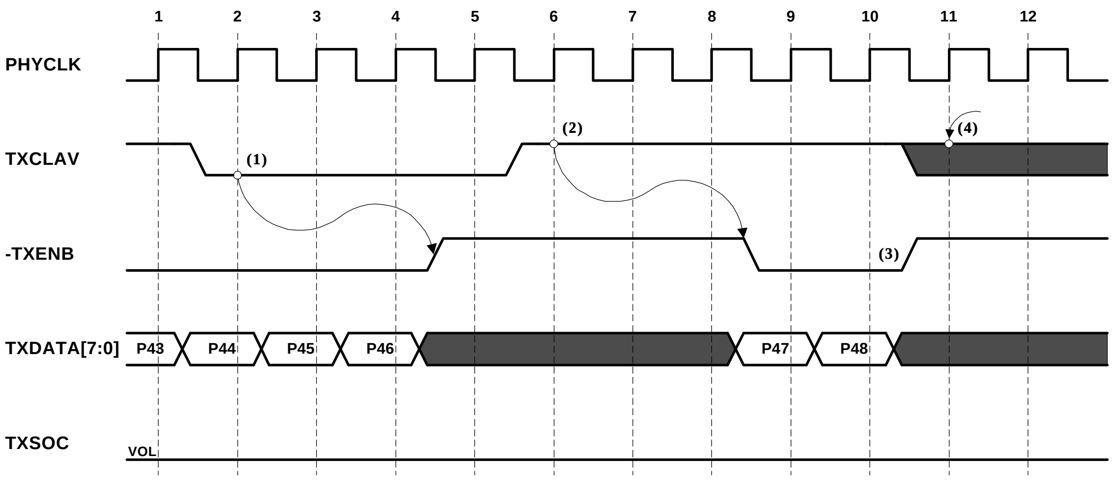


Figure 3.-21 - Octet level handshake -data cell-

Figure 3.-22 shows an example of unassigned cell transfer using octet-level handshaking.

(1) In the middle of cell, the PHY may deassert TXCLAV. TC35856BF detects TXCLAV deasserted on clock edge 2 and deassert -TXENB 1 clock later. Therefore, 1 octet, P45, should be accepted by the PHY.

(2) On clock edge 6, TC35856BF recognizes that TXCLAV was asserted by the PHY and, as a consequence of this, it asserts -TXENB 2 clocks later and re-start to transmit data.

(3) After TC35856BF has output the last data of the cell, -TXENB is deasserted on clock edge 11.

(4) 1 clock later, TC35856BF re-samples TXCLAV to determine if the next cell can be output.

Note that interrupt point on P44 is an example. Interrupt may occur on any octet position.

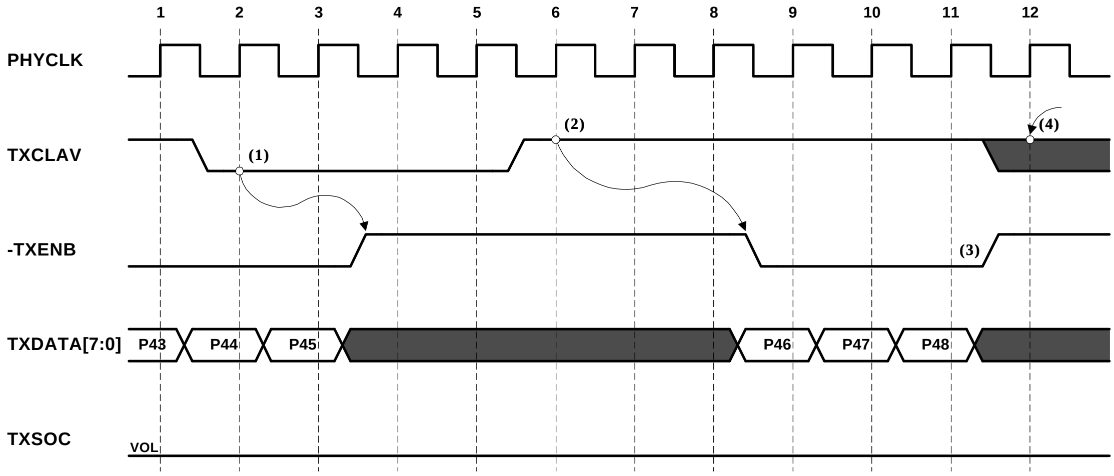


Figure 3.-22 - Octet level handshake -unassigned cell-

(B) Cell level handshake

UTOPIA specification uses TxEnb* and TxClav signals for handshaking. -TXENB and TXCLAV pins of TC35856BF should be connected to TxEnb* and TxClav pins of the PHY device, respectively. PHYCLK output pin of TC35856BF should be connected to TxClk pin of the PHY device. TC35856BF indicates that it wants to output data by asserting -TXENB to low. The PHY indicates it can accepting the transfer of a whole cell has valid cell data by asserting TxClav to high. The PHY device must be capable of accepting the transfer of a whole cell when the PHY device assert TXCLAV.

Figure 3.-23 shows an example of cell-level handshaking between the PHY device and TC35856BF.

(1) On clock edge 2, TC35856BF recognizes that TXCLAV was asserted by the PHY. TC35856BF asserts -TXENB and start to transmit a whole cell. -TXENB asserting-delay from TXCLAV has a range of 2 to 22 clocks, which depends on TC35856BF internal operation. TC35856BF ignores TXCLAV until the end of cell.

(2) After the last octet of cell (P48) has been outputted, TC35856BF deasserts -TXENB on clock edge 57.

(3) On the clock edge when the last octet of cell is valid, TC35856BF checks TXCLAV. If TXCLAV is deasserted, TC35856BF waits for TXCLAV.

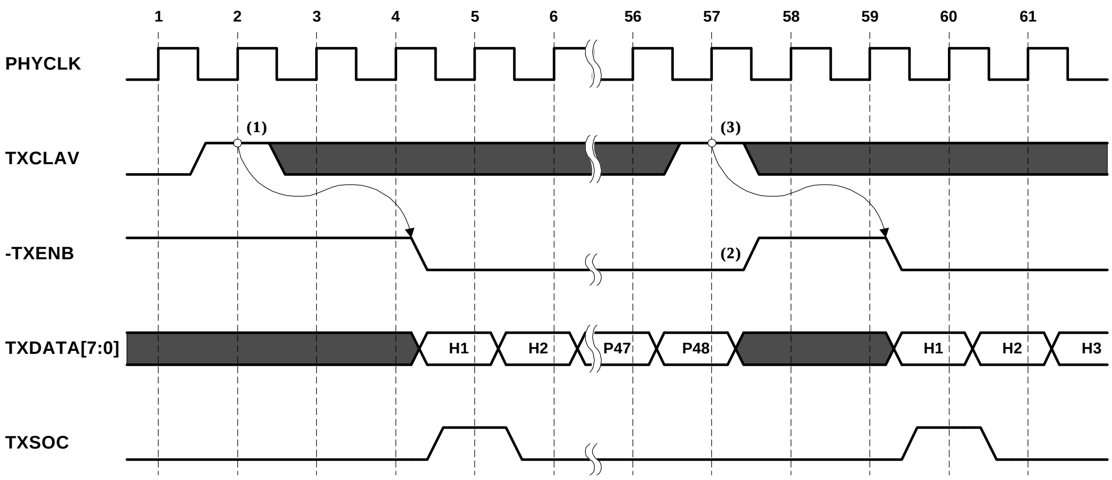


Figure 3.-23 - Cell level handshake

3.9.8.2 Receive operation

(A) Octet level handshake

UTOPIA specification defines RxEnb* and RxEmpty* signals for handshaking. -RXENB and RXCLAV pins of TC35856BF should be connected to RxEnb* and RxEmpty* pins of the PHY device, respectively. PHYCLK output pin of TC35856BF should be connected to RxClk pin of the PHY device. TC35856BF indicates that it wants to read received data by asserting -RXENB to low. The PHY indicates it has valid data by deasserting RxEmpty* (asserting RXCLAV) to high. "Deasserting RxEmpty*" is the same polarity of "asserting RXCLAV". The cycle during which -RXENB is asserted constitute a read window. During a read window the PHY device reads data from its internal FIFO and presents it on RXDATA/RXSOC on each low-to-high transition of PHYCLK. Asserting -RXENB while RXCLAV is low is not an error but the value of RXDATA is ignored.

Figure 3.-24 shows the octet-level handshaking between the PHY device and TC35856BF.

(1) On clock edge 2, TC35856BF recognizes that RXCLAV was asserted by the PHY and, as a consequence of this, it asserts -RXENB.

(2) The PHY detects -RXENB asserted on clock edge 3 and drives the first cell octet on RXDATA and RXSOC.

(3) Between clock edge 4 and 5, the PHY runs out of data and indicates invalid data on RXDATA by deasserting RXCLAV.

(4) On clock edge 9, TC35856BF cannot accept additional data and therefore it drives -RXENB deasserted in the previous cycle.

(5) TC35856BF continues transfer on clock edge 11 by asserting -RXENB on clock edge 10.

The meaning of RXCLAV depends on -RXENB. While -RXENB is deasserted, RXCLAV indicates the availability of an octet to transfer. While -RXENB is asserted, RXCLAV indicates valid/invalid data on RXDATA.

Note that interrupt point on H1 is an example. Interrupt may occur on any octet position.

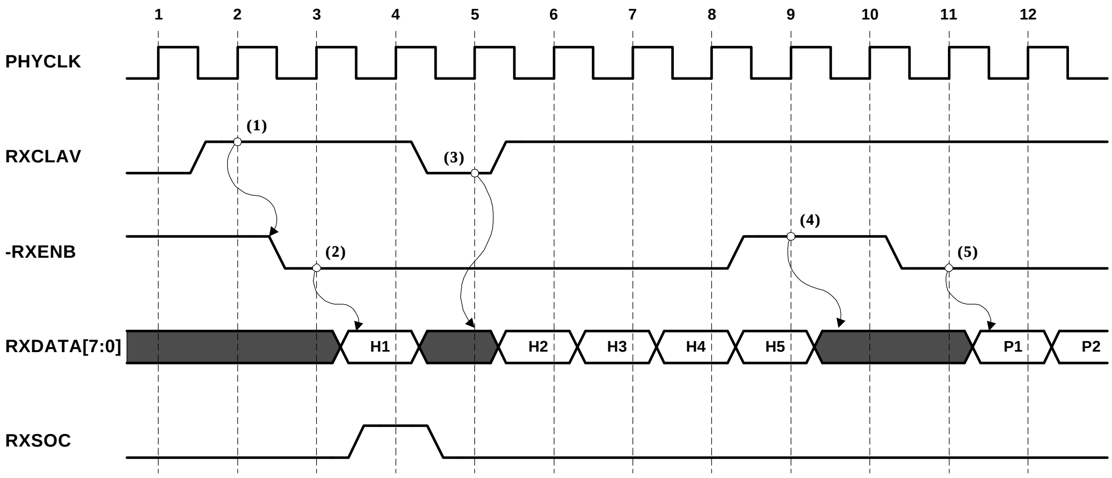


Figure 3.-24 - Octet level handshake

(B) Cell level handshake

UTOPIA specification uses RxEnb* and RxClav signals for handshaking. -RXENB and RXCLAV pins of TC35856BF should be connected to RxEnb* and RxClav pins of the PHY device, respectively. PHYCLK output pin of TC35856BF should be connected to RxClk pin of the PHY device. TC35856BF indicates that it wants to read received data by asserting -RXENB to low. The PHY indicates it has valid cell data by asserting RxClav to high only if the PHY device is ready to transfer a whole cell. Therefore, in the cycle following the one with the final octet of the cell, RXCLAV asserted indicates there is a new cell to transfer, while RXCLAV deasserted indicates there is no new cell to transfer.

Figure 3.-25 shows the cell-level handshaking between the PHY device and TC35856BF.

(1) On clock edge 1, the PHY indicates the availability of a new cell and, as a consequence of this, TC35856BF asserts -RXENB. After that TC35856BF doesn’t check RXCLAV until end of the cell. Because, UTOPIA specification specifies the PHY must be able to transfer a whole cell after once the PHY asserts RXCLAV.

(2) The PHY detects -RXENB asserted on clock edge 3, and starts the cell transfer.

(3) On clock edge 55, TC35856BF deasserts -RXENB while P48 is on RXDATA because TC35856BF knows that P48 is the last octet of the cell.

(4) The PHY may assert or deassert RXCLAV indicating whether a new cell to transfer is ready on clock edge 56. TC35856BF asserts -RXENB when TC35856BF detects RXCLAV asserted. It means that 2 clock cycles (clock edge 56 and 57) are the minimum time of -RXENB deasserted for back-to-back cell reception. Otherwise, TC35856BF waits for RXCLAV asserted. When the PHY device deasserts RXCLAV on clock edge 56, TC35856BF detects RXCLAV deasserted on clock edge 57, and keeps -RXENB deasserted. TC35856BF waits for RXCLAV asserted.

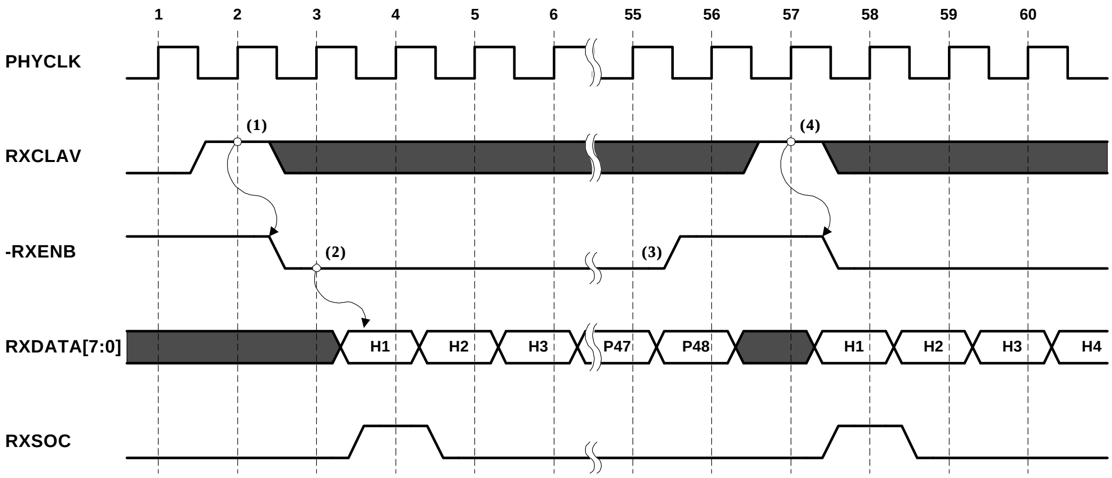


Figure 3.-25 - Cell level handshake -back to back transfer-

Figure 3.-26 shows an example where TC35856BF interrupts the cell transfer by deasserting -RXENB on cell-level handshaking.

(1) On clock edge 5, TC35856BF deasserts -RXENB while H3 is on RXDATA. The PHY detects -RXENB deasserted on clock edge 6, and stops the cell transfer.

(2) On clock edge 9, TC35856BF re-assert -RXENB to re-start receiving. The PHY device recognizes -RXENB asserted on clock edge 10, and outputs valid data (H4) on RXDATA.

Note that interrupt point on H3 is an example. Interrupt may occur on any octet position.

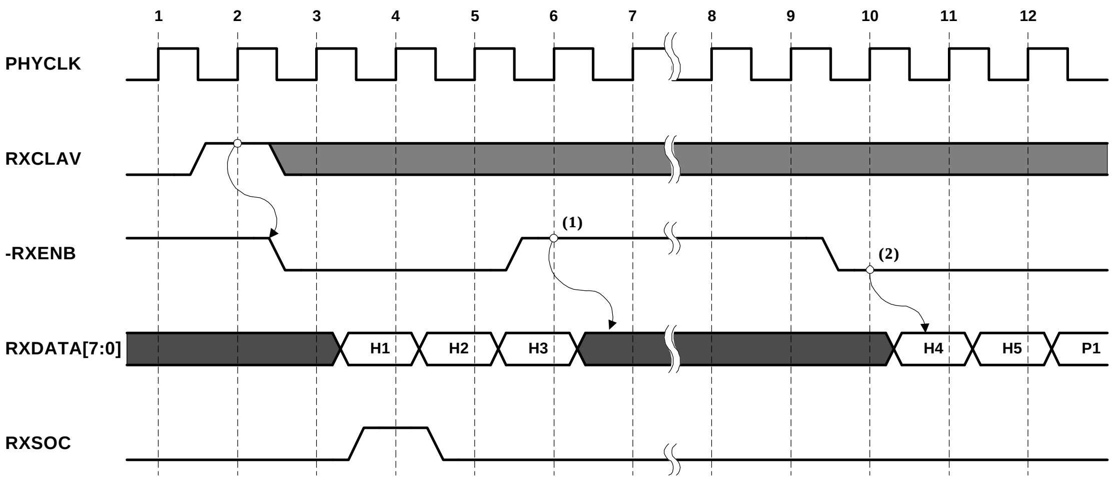


Figure 3.-26 - Cell level handshake -suspending reception-

3.9.9 Reverse UTOPIA operating timing

Following sections describe functional timing of reverse UTOPIA mode. The term "receive" and "transmit" refer "PHY->ATM" and "ATM->PHY", respectively. However UTOPIA mode, TC35856BF acts as PHY device in reverse, "receive" is "output operation" using segmentation engine, and "transmit" is "input operation" using reassembly engine.

3.9.9.1 Receive operation (Output operation)

TC35856BF indicates that data is available by asserting -TXENB to high when TXCLAV is asserted to high at the previous cycle. TC35856BF indicates that output data is valid by asserting -TXENB to high when TXCLAV is asserted to low at the previous cycle.

TC35856BF always deasserts -TXENB to low after it outputs the last byte of cell. After the last byte of the cells, TC35856BF doesn't output valid data on TXDATA for 3 clocks at least.

Figure 3.-27 shows the octet level handshaking between TC35856BF and ATM layer device.

(1) On clock edge 2, TC35856BF indicates data is available by asserting -TXENB to high while ATM layer device indicates that it cannot accept data at the next clock edge by asserting TXCLAV to low.

(2) On clock edge 3, TC35856BF continuously indicates that data is available by asserting -TXENB to high. ATM layer device indicates that it can accept data at the next clock edge by asserting TXCLAV to low.

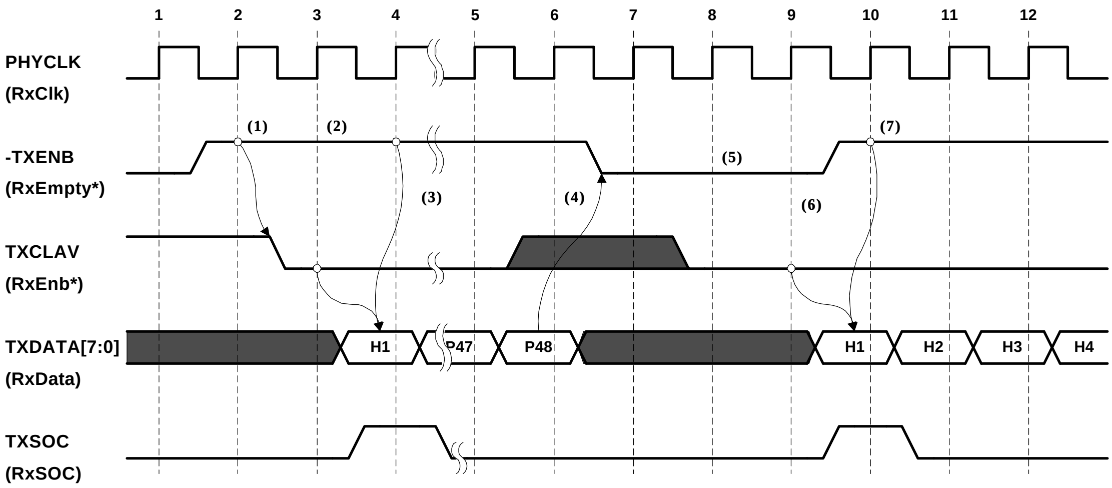
(3) Between clock edge 4 and 5, TC35856BF indicates that the data on TXDATA is valid by asserting -TXENB to high. ATM layer device continuously indicates that it can accept data at the next clock edge by asserting TXCLAV to low.

(4) On clock edge 6, TC35856BF indicates that the data on TXDATA is valid by asserting -TXENB to high. On clock edge 7, TC35856BF indicates data is not available by deasserting -TXENB to low because TC35856BF outputted the last byte of cell : P48 on clock edge 6. On clock edge 6 and 7, TXCLAV from ATM layer device is don't care.

(5) On clock edge 8, TC35856BF indicates data is not available by deasserting -TXENB to low. ATM layer device indicates that it can accept data at the next clock edge by asserting TXCLAV to low.

(6) On clock edge 9, TC35856BF indicates data is not valid by deasserting -TXENB to low. ATM device still asserts TXCLAV to low in order to indicate it can accept data.

(7) On clock edge 10, TC35856BF indicates data is valid by asserting -TXENB to high. ATM layer device continuously indicates that it can accept data at the next clock edge by asserting TXCLAV to low.



Note. Signal names in parenthesis are the names defined in UTOPIA specification.

Figure 3.-27 - Octet level handshake in Receive operation 1 in Reverse UTOPIA

Figure 3.-28 also shows the octet level handshaking between TC35856BF and ATM layer device. This figure indicates ATM layer device can deasserts TXCLAV to high at any time.

(1) On clock edge 3, TC35856BF indicates that data is available by asserting -TXENB to high while ATM layer device indicates that it can accept data at the next clock edge by asserting TXCLAV to low.

(2) On clock edge 4, TC35856BF indicates that data is valid by asserting -TXENB to high while ATM layer device indicates that it can accept data at the next clock edge by asserting TXCLAV to low.

(3) On clock edge 5, TC35856BF indicates that data is valid by asserting -TXENB to high while ATM layer device indicates that it cannot accept data at the next clock edge by deasserting TXCLAV to high.

(4) On clock edge 6, TC35856BF indicates that data is available by asserting -TXENB to high while ATM layer device indicates that it can accept data at the next clock edge by asserting TXCLAV to low.

(5) Between clock edge 7 and 8, TC35856BF indicates that data is valid by asserting -TXENB to high while ATM layer device indicates that it can accept data at the next clock edge by asserting TXCLAV to low.

(6) On clock edge 9, TC35856BF indicates that data is valid by asserting -TXENB to high while ATM layer device indicates that it cannot accept data at the next clock edge by deasserting TXCLAV to high.

(7) On clock edge 10, TC35856BF indicates that data is available by asserting -TXENB to high while ATM layer device indicates that it can accept data at the next clock edge by asserting TXCLAV to low.

(8) On clock edge 11, TC35856BF indicates that the data on TXDATA is valid by asserting -TXENB to high. On clock edge 12, TC35856BF indicates data is not available by deasserting -TXENB to low because TC35856BF received 53rd Bytes data : P48 on clock edge 11. On clock edge 11 and 12, TXCLAV from ATM layer device is don’t care.

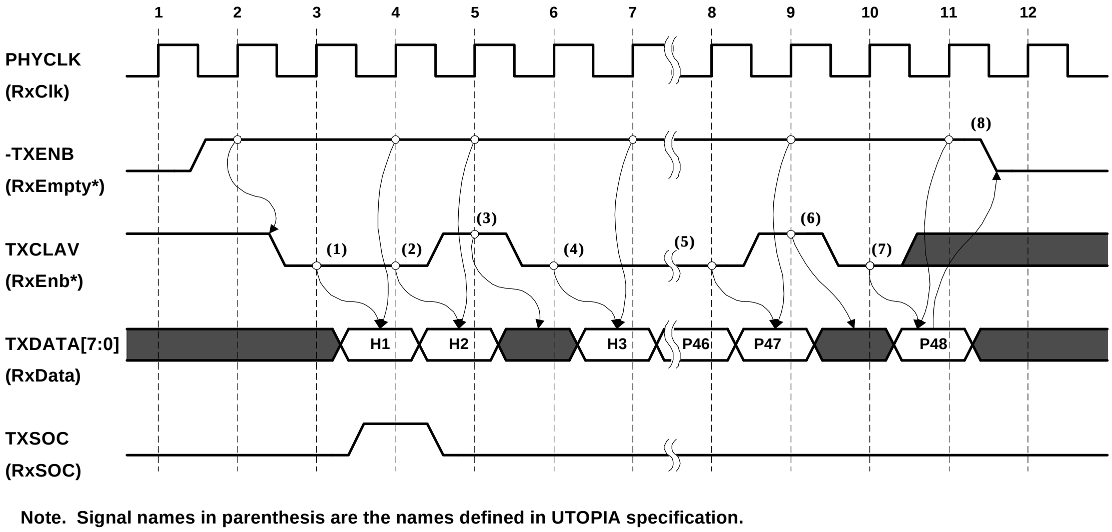


Figure 3.-28 - Octet level handshake in Receive operation 2 in Reverse UTOPIA

3.9.9.2 Transmit operation (Input operation)

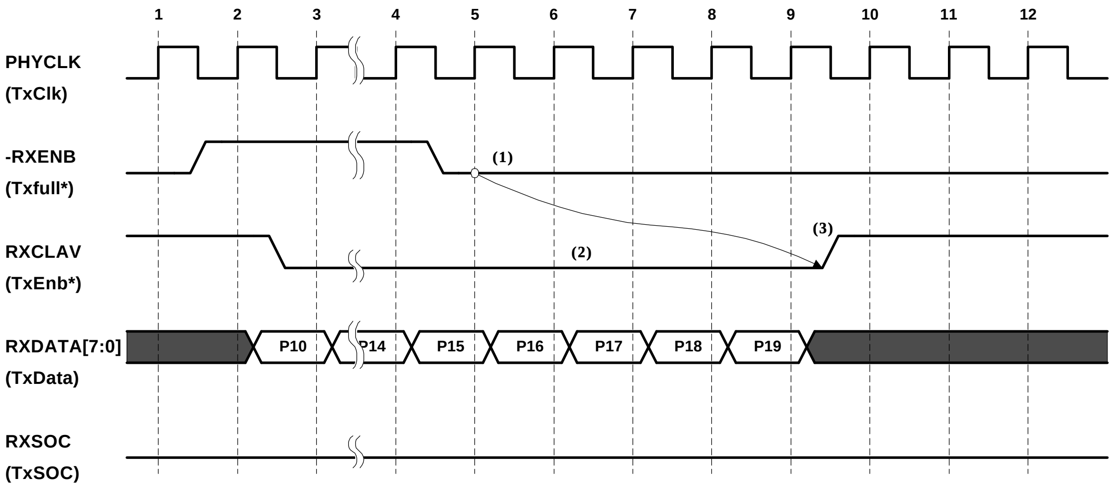
TC35856BF indicates that it can accept at least 4 bytes valid data by asserting -RXENB to high. During a time period termed the transmit window, TC35856BF stores data from RXDATA on the low-to-high transition of PHYCLK, if RXCLAV is asserted to low. The window exists from the time that TC35856BF indicates it can accept data by asserting -RXENB to high and the time until 4 valid input cycle after TC35856BF deasserts -RXENB to low. ATM layer device can deassert RXCLAV to high at any time. After -RXENB is deasserted to low, this indicates that ATM layer device may transfer no more than 4 data bytes on RXDATA until -RXENB is asserted to high again. After 4 valid input cycle, ATM layer device must deassert RXCLAV to high.

Figure 3.-29 shows the octet level handshaking between TC35856BF and ATM layer device.

(1) On clock edge 5, TC35856BF indicates it can accept data at most 4 bytes by deasserting -RXENB to low.

(2) Between clock 6 and 9, ATM layer device still continues outputting data by asserting RXCLAV to low.

(3) On clock edge 10, ATM layer device stop outputting data by deasserting RXCLAV to high.



Note. Signal names in parenthesis are the names defined in UTOPIA specification.

Figure 3.-29 - Octet level handshake in Transmit operation 1 in Reverse UTOPIA

Figure 3.-30 also shows the octet level handshaking between TC35856BF and ATM layer device.

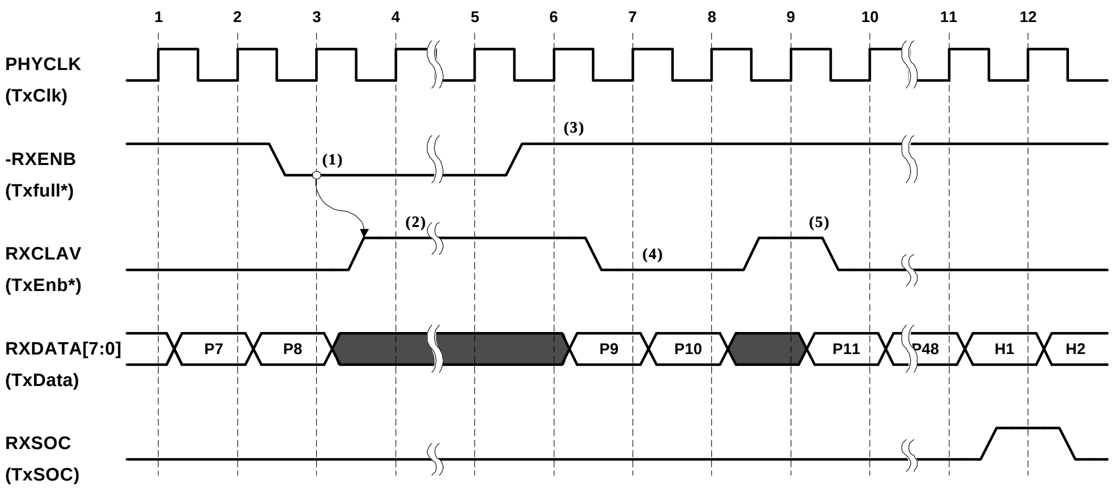
(1) On clock edge 3, TC35856BF indicates it can accept data at most 4 bytes by deasserting -RXENB to low.

(2) On clock edge 4, ATM layer device immediately stops outputting data by deasserting RXCLAV to high.

(3) On clock edge 6, TC35856BF indicates it can accept data by asserting -RXENB to high again.

(4) On clock edge 7, ATM layer device indicates it starts outputting data by asserting RXCLAV to low.

(5) On clock edge 9, ATM layer device indicates it stall outputting data by deasserting RXCLAV to high. TC35856BF doesn't input the data on RXDATA.



Note. Signal names in parenthesis are the names defined in UTOPIA specification.

Figure 3.-30 - Octet level handshake in Transmit operation 2 in Reverse UTOPIA

4. TC35856BF Address Map and CSR Bit Assignments

4.1 PCI Memory and I/O Space

CSR	Bit Assignments	LW Address	Default
Tx_Pending_Slots:	<31> - Cntrl \	0x0 - 0xF	N.A.
(WO)	<30> - Raw \		
(16LW range)	<29> - Idle \		
	<28> - MGMT \ Even		
	<27:16> - VC / Address		
	<15> - CRC10_Enb /		
	<14> - EOP /		
	<13:0> - Slot_Size /		
	<31:0> - Base_Addr > Odd addresses		
Rx_Pending_slots:	<31:18> - Slot_Tag \	0x10 - 0x1F	N.A.
(16LW range)	<17:16> - Slot_Type \ Even		
(WO)	<15:12> - Reserved / Addresses		
	<11:0> - VC /		
	<31:2> - Base_Addr \ Odd addresses		
	<1:0> - Reserved /		
Meteor_Cntrl1:	<31:29> - Reserved	0x20	00000040
(RW)	<28> - CRC10en_ovrrng_cell		
	<27> - SRAM_mode		
	<26> - Idle_cell_mode		
	<25> - Reverse_UTOPIA		
	<24> - PCI_MRM_Sel		
	<23> - PCI_MRM_MRL_Dis		
	<22> - SW_ABR_Enb		
	<21> - CBR_ST_Sel		
	<20> - FM_RM_Enb		
	<19> - FM_Resynch (WO)		
	<18> - FM_Enb		
	<17> - DMA_BSwap_Data		
	<16> - DMA_BSwap_Literals		
	<15> - 64K_Mode_Enb		
	<14:13> - VC_Mode_Sel		
	<12:10> - VPI_VCI_Mapping		
	<9> - Tx_Enb (Enables TX scheduler)		
	<8> - Phy_IF_reset		
	<7> - Phy_Cell_Mode_Enb		
	<6> - Phy_Chip_reset		
	<5> - GFC_Signal_Detect (RW1C)		
	<4> - Phy_Int_High		
	<3> - Phy_Loopback_Enb		
	<2> - Phy_GFC_Enb		
	<1> - Phy_GFC_A_Enb		
	<0> - Rx_Enb		

Meteor_Cntrl2:	<31:30>	- Rx_Raw_Report_Hldoff	0x21	00001001
(RW)	<29>	- Slot_Cong_CI_Enb		
	<28>	- Big_CI_Enb		
	<27>	- Small_CI_Enb		
	<26>	- Raw_CI_Enb		
	<25:24>	- Big_Slot_Low_Th (8, 16, 32, 64 slots)		
	<23:22>	- Small_Slot_Low_Th (8, 16, 32, 64 slots)		
	<21:20>	- Raw_Slot_Low_Th (8, 16, 32, 64 slots)		
	<19:17>	- Big_Slot_Size (1, 2, 4, 8, 10, 16 KB)		
	<16:8>	- Small_Slot_Size (16LW - 512LW)		
	<7:4>	- Rx_RATO_Time		
	<3>	- Rx_Bad_RM_DMA_Enb		
	<2>	- AAL5_Max_Length_Enb		
	<1>	- EFCI_Or_Enb		
	<0>	- Soft_Reset		
Intr_Status:	<31:21>	- Reserved	0x22	00000000
(RW1C)	<20>	- Tx_Fatal_Err		
	<19>	- PCI_Fatal_Err		
	<18>	- Phy_Intr		
	<17>	- Ph_Done_Intr		
	<16>	- Ph_Access_Err		
	<15>	- Host_Access_Err		
	<14>	- Tx_Free_Slot_Unfl		
	<13>	- Tx_IOC		
	<12>	- FM_Resynch_Done		
	<11>	- Rx_Free_Slot_Ovfl		
	<10>	- Rx_Data_Fifo_Ovfl		
	<9>	- Rx_No_Big_Slots		
	<8>	- Rx_No_Small_Slots		
	<7>	- Rx_No_Raw_Slots		
	<6>	- Rx_Big_Slots_Low		
	<5>	- Rx_Small_Slots_Low		
	<4>	- Rx_Raw_Slots_Low		
	<3>	- Rx_Unopened_VC		
	<2>	- Rx_Unknown_VC		
	<1>	- Rx_Unknown_Ack		
	<0>	- Rx_IOC		
Intr_Enb:	<31:21>	- Reserved	0x23	00000000
(RW)	<20:0>	- Intr_Enb		
Intr_Hldoff:	<31>	- Tx_IOC_Hldoff_Wr (WO)	0x24	00000000
(RW)	<30:24>	- Tx_IOC_Hldoff_Cnt		
	<23:16>	- Tx_IOC_Hldoff_Time		
	<15>	- Rx_IOC_Hldoff_Wr (WO)		
	<14:8>	- Rx_IOC_Hldoff_Cnt		
	<7:0>	- Rx_IOC_Hldoff_Time		

Tx_FS_List_Ptrs:	<31>	- Tx_FS_List_Valid	0x25	00000000
(RW)	<30:28>	- Reserved		
	<27:16>	- Tx_FS_List_Head		
	<15:12>	- Reserved		
	<11:0>	- Tx_FS_List_Tail		
Tx_Report_Base:	<31:6>	- Tx_Report_Base	0x26	00000000
(RW)		(64B aligned)		
	<5:0>	- Reserved		
Rx_Report_Base:	<31:6>	- Rx_Report_Base	0x27	00000000
(RW)		(64B aligned)		
	<5:0>	- Reserved		
Tx_Report_Ptr:	<31:2>	- Tx_Report_Ptr	0x28	00000000
(RO)	<1>	- Reserved		
	<0>	- Tx_Own_Wr_Val		
Rx_Report_Ptr:	<31:2>	- Rx_Report_Ptr	0x29	00000000
(RO)	<1>	- Reserved		
	<0>	- Rx_Own_Wr_Val		
Rx_Slot_Cong_Th:	<31:24>	- Reserved	0x2A	00000000
(RW)	<23:16>	- Raw_Slot_Cong_Thld		
	<15:8>	- Big_Slot_Cong_Thld		
	<7:0>	- Small_Slot_Cong_Thld		
Tx_ABR_ADTF:	<31:16>	- ADTF_2	0x2B	00000000
(RW)	<15:0>	- ADTF_1		
Tx_ABR_Nrm_Trm:	<31:30>	- Reserved	0x2C	00000000
(RW)	<29>	- OFRM_suppress		
	<28:24>	- Line_Rate_Exp		
	<23:21>	- Nrm_2		
	<20:12>	- Trm_2		
	<11:9>	- Nrm_1		
	<8:0>	- Trm_1		
Peephole_Cmd:	<31>	- Ph_Read	0x2D	40000000
(RW)	<30>	- Ph_Prf_Access		
	<29:18>	- Reserved		
	<17:0>	- Ph_Addr		
Peephole_Data:	<31>	- Ph_Done (RO)	0x2E	80000000
(RW)	<30:16>	- Reserved		
	<15:0>	- Ph_Data		

Rx_Raw_Slot_Tag:	<31:16>	- Reserved	0x2F	00000000
(RO)	<15:2>	- Rx_Raw_Slot_Tag		
	<1>	- Reserved		
	<0>	- Rx_Raw_Slot_Valid		
Rx_CellCnt:	<31:20>	- Reserved	0x30	00000000
(RO)	<19:0>	- Rx_Cell_Cnt		
Rx_AAL5_DropPktCnt:	<31:20>	- Reserved	0x31	00000000
(RO)	<19:0>	- Rx_AAL5_Drop_PktCnt		
Rx_Raw_DropCellCnt:	<31:20>	- Reserved	0x32	00000000
(RO)	<19:0>	- Rx_Raw_Drop_CellCnt		
Tx_CellCnt:	<31:20>	- Reserved	0x33	00000000
(RO)	<19:0>	- Tx_Cell_Cnt		
Real_Time:	<31:30>	- Reserved	0x34	00000000
(RO)	<29:8>	- Real_Time_3p93us		
	<7:0>	- Real_Time_30ns		
Current_Time:	<31:16>	- Reserved	0x35	00000000
(RO)	<15:8>	- ABR_Current_Time		
	<7:0>	- CBR_Current_Time		
Test_Cntrl:	<31:16>	- Reserved	0x36	00000000
(RW)	<15>	- ACK_FIFO_Read_Disable		
	<14>	- Test_Batch_FIFO		
	<13>	- Inc_PCI_FIFO_Addr		
	<12>	- Report_Segment		
	<11>	- Tx_Partial_Cell_Wait (RO)		
	<10>	- Tac_Timer_Speedup		
	<9>	- LastRM_Poll_Time (RO)		
	<8>	- RATO_Poll_Time (RO)		
	<7>	- Phy_GFC_Halt		
	<6>	- Phy_GFC_Set_A		
	<5>	- Ph_Test		
	<4>	- Force_All_Intr (WO)		
	<3>	- Counter_Force_Count (WO)		
	<2>	- Counter_Segment		
	<1>	- Tx_DMA_Disable		
	<0>	- Rx_DMA_Disable		

MPHY_Cntrl:	<31:21>	- Reserved	0x37	00000000
(RW)	<20>	- MPHY_ADDR		
	<19>	- MPHY_POLLING		
	<18>	- MPHY_DIRECT		
	<17>	- Suppress_idle_cell		
	<16>	- Dual_timer_mode		
	<15:8>	- cell_timer_2		
	<7:0>	- cell_timer_1		
Rx_Buff_Cntrl:	<31:16>	- Reserved	0x38	00004020
(RW)	<15:0>	- AAL5_Max_Length		

Notes:

- RW = Read/Write
  RW1C = Read/Write-one-to-clear
  RO = Read-only
  WO = Write-only
- Where bit-accessability within a register differs from the accessability of the register (for example, a WO bit within a R/W register) this is indicated next to the bit name definition.
- All "Reserved" fields should be written with 0's and yield undefined value when read.
- All RO counters are segmentable for testability, and are wrapping counters.
- Interrupt-enable bits control whether or not specific interrupts affect the PCI interrupt line. They do not affect the state of the bits in the interrupt registers.
- The Rx_IOC_Hldoff_Wr and Tx_IOC_Hldoff_Wr bits in the Intr_Hldoff register control whether the Rx_IOC and/or Tx_IOC Holdoff values are written, respectively. The Rx_IOC_Hldoff_Wr and Tx_IOC_Hldoff_Wr bits must be written with a 1 for the corresponding holdoff values to be written.

4.2 PCI Configuration Space

TC35856BF supports only the predefined Configuration Space header portion of the PCI Configuration Space. This header is 64-bytes in size, and is described in detail in Section 3.8.2 and illustrated in Figure 3.-19. The structure of the header is illustrated again in Figure 4.-1 with the default values of the various fields included for convenience. All values shown are in hex.

Figure 4.-1 - Configuration Space Header Default Values

31	16	15	0	
Device ID = 0020		Vendor ID = 102F		00h
Status = 0080		Command = 0000		04h
Class Code = 020300			Revision ID = 02	08h
BIST = 00	Header Type = 00	Latency Timer = 00	Cacheline Size = 00	0Ch
Base Adress Register 0 = 00000000				10h
Base Adress Register 1 = 00000001				14h
Base Address Registers 2-5 = 00000000				18h 1Ch 20h 24h
Cardbus CIS Pointer = 00000000				28h
Subsystem ID		Subsystem Vendor ID		2Ch
Expansion ROM Base Address = 00000000				30h
Reserved = 00000000				34h
Reserved = 00000000				38h
Max_Lat = 0x10	Min_Gnt = 0x08	Interrupt Pin = 0x01	Interrupt Line = 0x00	3Ch

5. CSR Descriptions

This section contains descriptions of all CSRs in TC35856BF located in PCI memory space. **All reserved fields will yield undefined values when read and should be written with 0s.**

Table 5.-1 - Tx_pending_Slot Registers when cntrl=0, 0x0 - 0xF (Even Addresses)

Bits	Name	RW	Description
31	Cntrl	WO	When set, indicates that the words are segmentation engine control word. When cleared, indicates that the words are tx pending slot.
30	Raw	WO	When set, indicates that the transmit slot contains a single 52-byte raw cell (all bytes in cell except HEC). When cleared, indicates that slot contains AAL5 or MPEG data.
29	Idle	WO	When set, indicates that the transmit slot contains 1 or more idle cells which will not be transmitted. Used for CBR rating control purposes only.
28	Mgmt	WO	When set, indicates that the transmit slot contains a single OAM F4 or F5 cell. If this bit is set, **Raw** should also be set.
27:16	VC	WO	VC of the data in the transmit slot.
15	CRC10_Enb	WO	When set, CRC10 will be calculated and inserted in every cell transmitted with data from this transmit slot. This bit may be used to generate CRC10 on OAM F4 or F5 cells, or on any other type of raw cell. If this bit is set, **Raw** should also be set.
14	EOP	WO	When set, indicates that this transmit slot contains either: • the last byte of an AAL5 packet. • a raw cell whose consumption should be reported, and counted for interrupt purposes.
13:0	Slot_Size	WO	Size of transmit slot in bytes. For raw slots, this field must always be 52. For non-raw slots, this field must be non-zero.

Table 5.-2 - Tx_pending_Slot Registers when cntrl=0, 0x0 - 0xF (Odd Addresses)

Bits	Name	RW	Description
31:0	Base_Addr	WO	Physical base address of the tx slot being posted (byte address). May be any value for slots for which the **Idle** bit is set, since no data is DMA'd from these slots.

Table 5.-3 - Rx_pending_Slot Registers, 0x10 - 0x1F (Even Addresses)

Bits	**Name**	**RW**	**Description**
31:18	Slot_Tag	WO	These bits indicate the slot_tag of the rx slot being posted. Each rx slot posted by the driver must be identified by a unique 14-bit slot_tag. When TC35856BF reports the consumption of the rx slot, it will include the slot_tag of the slot in the receive report. This allows the driver to obtain the virtual address of the slot through a simple table lookup.
17:16	Slot_Type	WO	These bits indicate the type of rx slot being posted. They are decoded as follows: 00 - Small 01 - Big 10 - Raw 11 - Reserved
15:12	Reserved	WO	
11:0	VC	WO	Indicates the VC associated with the rx slot being posted. This field selects the VC whose ***slots_consumed*** counter should be decremented by 1 for FLOWmaster congestion control purposes. Since the ***slots_consumed*** counters stick at 0, this field may contain any value for slots which are posted at initialization time. A value of 0 may be used if no ***slots_consumed*** counter is to be decremented.

Table 5.-4 - Rx_pending_Slot Registers, 0x10 - 0x1F (Odd Addresses)

Bits	Name	RW	Description
31:2	Base_Addr	WO	Physical base address of the rx slot being posted (longword address).
1:0	Reserved		

Table 5.-5 - Meteor_Cntrl1, 0x20

Bits	Name	RW	Description
31:29	Reserved		
28	CRC10en_ovrrng_cell	RW	Enable CRC-10 checking for over ranged cell on 64K_mode.
27	SRAM_mode	RW	Selects SRAM bank mode. When one bank SRAM is used for 1K or 4K VC mode, this bit should be set to “1”.
26	Idle_cell_mode	RW	Selects whether TC35856BF outputs “unassigned cell (VPI=VCI=PT=CLP=0)” or “Idle cell (VPI=VCI=PT=0, CLP=1)”. When this field set to “1”, TC35856BF outputs Idle cell. This field should be set before Phy_IF_reset bit is clear to “0”.
25	Reverse_UTOPIA	W	Selects whether UTOPIA interface acts as SAR device (0) or PHY device (1). **This field should be set immediately after TC35856BF has been reset and before Tx_Enb and Rx_Enb bits have been set. This field should not be changed during operation.**
24	PCI_MRM_Sel	RW	Selects whether TC35856BF will generate a Memory-read-line (0) or a Memory-read-multiple (1) cycle when a burst read to host Memory exceeds one cache line in length. Note that both types of cycles are inhibited when the **PCI_MRM_MRL_Dis** bit is set.
23	PCI_MRM_MRL_Dis	RW	When cleared, PCI Memory-read-line or Memory-read-multiple cycles (as selected by the **PCI_MRM_Sel** bit) will be generated by TC35856BF when necessary. When set, TC35856BF will generate only Memory-read cycles.
22	SW_ABR_Enb	RW	When set, enables software implementation of ABR flow control. When cleared, TC35856BF will implement ABR flow control without software intervention.
21	CBR_ST_Sel	RW	Selects which of the two CBR schedule tables should be used for scheduling of CBR flows. When cleared, CBR Schedule Table #1 will be used. When set, CBR Schedule Table #2 will be used.
20	FM_RM_Enb	RW	When set, the transmission of ABR RM cells will be subject to FLOWmaster flow control, and will require a positive credit balance to be transmitted. When cleared, ABR RM cells for a given VC may be transmitted even when the credit balance for the VC is 0.

19	FM_Resynch	WO	FLOWmaster credit resynchronization request. When a 1 is written to this bit, TC35856BF will record the number of entries in its internal FLOWmaster ack FIFOs. When all of the entries have been transmitted, it will generate an FM_Resynch_Done interrupt. Writing a 0 to this bit has no effect.
18	FM_Enb	RW	When set, FLOWmaster flow control is enabled.
17	DMA_Bswap_Data	RW	When set, TC35856BF will byte-swap all Tx and Rx data which it DMA's across the PCI bus. Literals will not be byte-swapped unless the **DMA_Bswap_Literals** bit is set.
16	DMA_Bswap_Literals	RW	When set, TC35856BF will byte-swap all literals which it DMA's across the PCI bus. Literals are: - Tx and Rx status reports - Rx status longwords Tx and Rx data will not be byte-swapped unless the **DMA_Bswap_Data** bit is set.
15	64K_Mode_Enb	RW	When set, TC35856BF will operate in 64K mode. This means that received cells with a VPI/VCI outside of the allowed range selected by VPI_VCI_Mapping will be mapped to VC 0. When cleared, TC35856BF will drop received out-of-range cells and generate an Rx_Unknown_VC interrupt when out-of-range cells are received.
14:13	VC_Mode_Sel	RW	These bits select the number of VCs supported, according to the amount of on-board SRAM provided to TC35856BF. They are decoded as follows: 00 - 256 VCs (64KB of SRAM required) 01 - 1K VCs (128KB of SRAM required) 10 - Reserved 11 - 4K VCs(512KB of SRAM required) **This field must be written with the correct value (depending on the amount of SRAM available) before the SRAM may be accessed.**

12:10	VPI_VCI_Mapping	RW	These bits determine the manner in which the VC provided by the driver for transmitted AAL5 packets is mapped to cell VPI/VCIs. They also determine the VC to which received cells are mapped. The bits are decoded as follows. The symbol "↔" denotes a mapping from left to right on reception, and from right to left on transmission. **VC_Mode_Sel = 00 (256 VCs):** **000** - VCI[7:0] ↔ VC[7:0] **001** - VPI[0] ↔ VC[7] VCI[6:0] ↔ VC[6:0] **010** - VPI[1:0] ↔ VC[7:6] VCI[5:0] ↔ VC[5:0] **011** - Reserved **1xx** - Reserved **VC_Mode_Sel = 01 (1K VCs):** **000** - VCI[9:0] ↔ VC[9:0] **001** - VPI[0] ↔ VC[9] VCI[8:0] ↔ VC[8:0] **010** - VPI[1:0] ↔ VC[9:8] VCI[7:0] ↔ VC[7:0] **011** - VPI[2:0] ↔ VC[9:7] VCI[6:0] ↔ VC[6:0] **1xx** - Reserved **VC_Mode_Sel = 10 (4K VCs):** **000** - VCI[11:0] ↔ VC[11:0] **001** - VPI[1] ↔ VC[11] VCI[10:0] ↔ VC[10:0] **010** - VPI[1:0] ↔ VC[11:10] VCI[9:0] ↔ VC[9:0] **011** - VPI[2:0] ↔ VC[11:9] VCI[8:0] ↔ VC[8:0] 1xx - Reserved
9	Tx_Enb	RW	When set, enables the transmission of data. Note that this bit enables and disables transmission at the **input** of Meteor's internal transmit data FIFO.
8	Phy_IF_reset	RW	When set, resets the TC35856BF PHY UTOPIA interface. This bit has to be set when phy_loopback_enb is set or cleared, and then cleared without phy_loopback_enb bit changing. **This field should not be set after Rx_Enb or Tx_Enb is set to "1".**

7	Phy_Cell_Mode_Enb	RW	When set, the TC35856BF PHY UTOPIA interface uses cell-level handshaking. When cleared, the interface uses octet-level handshaking. This bit has no effect when the UTOPIA interface is in reversed mode.
6	Phy_Chip_reset	RW	When set, resets the PHY chip connected to TC35856BF. The -PHYRST output pin is simply an inverted version of this bit.
5	GFC_Signal_Detect	RW1C	This bit is set when 3 or more Set_A, Set_B or Halt signals are received by TC35856BF on the Utopia interface (in any combination). Writing a 1 to this bit clears it. Writing a 0 has no effect.
4	Phy_Int_High	RW	When set, TC35856BF expects an active high interrupt signal from the PHY chip connected to TC35856BF.
3	Phy_Loopback_Enb	RW	When set, Meteor's PHY interface is put into loopback mode. **This bit should be modified only after TC35856BF has been reset and before the Rx_Enb or Tx_Enb bits have been set.**
2	Phy_GFC_Enb	RW	When set, GFC flow control is enabled at Meteor's UTOPIA interface. Care should be taken in modifying this bit, as outlined in TBD.
1	Phy_GFC_A_Enb	RW	When set in conjunction with Phy_GFC_Enb, controlled (non-CBR) traffic in TC35856BF will be subject to Set_A/Null_A commands from the controlling equipment. When cleared, controlled traffic will not be subject to Set_A/Null_A commands. This bit has no effect when Phy_GFC_Enb is cleared.
0	Rx_Enb	RW	When set, enables reception of either PHY chip receive data or TC35856BF transmit (loopback) data, depending on the value of Phy_Loopback_Enb.

Table 5.-6 - Meteor_Cntrl2, 0x21

Bits	Name	RW	Description
31:30	Rx_Raw_Report_Hldoff	RW	Selects the number of raw cell slots which must be filled with Rx data before a raw cell Rx status report is generated. This field is decoded as follows: 00 - Report every raw slot 01 - Report every 4th raw slot 10 - Report every 16th raw slot 11 - Report every 32nd raw slot In order to minimize spurious interrupts, this field should be programmed with a value **equal to or less than** the value specified by the **Rx_IOC_Hldoff_Cnt** field of the Intr_Hldoff register.
29	Slot_Cong_CI_Enb	RW	When set, enables setting of the CI bit in turn-around RM cells when VC-specific slot congestion occurs.
28	Big_CI_Enb	RW	When set, enables unconditional setting of the CI bit in turn-around RM cells for VCs assigned to big slots.
27	Small_CI_Enb	RW	When set, enables unconditional setting of the CI bit in turn-around RM cells for VCs assigned to small slots.
26	Raw_CI_Enb	RW	When set, enables unconditional setting of the CI bit in turn-around RM cells for VCs assigned to raw slots.
25:24	Big_Slot_Low_Th	RW	Specifies the FIFO level (number of Rx big free slots remaining) at which an Rx_Big_Slots_Low interrupt will be generated, indicating that TC35856BF is running out of free Rx big slots. These bits are decoded as follows: 00 - 8 slots 01 - 16 slots 10 - 32 slots 11 - 64 slots
23:22	Small_Slot_Low_Th	RW	Specifies the FIFO level (number of Rx small free slots remaining) at which an Rx_Small_Slots_Low interrupt will be generated, indicating that TC35856BF is running out of free Rx small slots. These bits are decoded as follows: 00 - 8 slots 01 - 16 slots 10 - 32 slots 11 - 64 slots

21:20	Raw_Slot_Low_Th	RW	Specifies the FIFO level (number of Rx raw free slots remaining) at which an Rx_Raw_Slots_Low interrupt will be generated, indicating that TC35856BF is running out of free Rx raw slots. These bits are decoded as follows: 00 - 8 slots 01 - 16 slots 10 - 32 slots 11 - 64 slots
19:17	Big_Slot_Size	RW	Specifies the size of the Rx big slots as follows: 000 - 1KB 001 - 2KB 010 - 4KB 011 - 8KB 100 - 10KB 101 - 16KB 110 - Reserved 111 - Reserved
16:8	Small_Slot_Size	RW	Specifies the size of the Rx small slots in units of longwords (1LW = 4 bytes). This value must be programmed in the range 16LW - 512LW. **512LW is specified by programming this field with the value 0.**
7:4	Rx_RATO_Time	RW	Specifies the Reassembly time-out time to be used for AAL5 reassembly time-out purposes in 193ms increments. This field must be in the range 2 - 15 if the reassembly time-out function is to be used. The value 0 disables the function, and the value 1 is **illegal**. The time-out time will be: Time = (Rx_RATO_Time * 193) ms ± 193ms
3	Rx_Bad_RM_DMA_Enb	RW	When set, bad receive RM cells will be DMA'd to host memory in raw cell slots. When cleared, bad received RM cells will be dropped. Bad RM cells are forward or backward RM cells for which the ID field is ≠ 1.
2	Rx_AAL5_16K_Lmt (AAL5_Max_Length_Enb),	RW	When set, received AAL5 packets will be limited to the length which is specified by AAL5_Max_Length. When cleared, AAL5 packets up to (64K - 1) bytes in length may be received

1	EFCI_Or_Enb	RW	When set, TC35856BF will report the logical OR of all the EFCI bits in the cells which make up a received AAL5 packet. When cleared, the EFCI bit in the last cell received in a packet will be reported.
0	Soft_Reset	RW	When set, TC35856BF is reset and all CSRs are set to their default state. This bit must be cleared for any CSR writes to take place.

Table 5.-7 - Intr_Status, 0x22

- All bits are read/write-one-to-clear (RW1C)

Bits	Name	RW	Description
31:21	Reserved		
20	Tx_Fatal_Err	RW1C	Indicates that the one of the following fatal errors has occurred in the transmission of data: - The length of an AAL5, Idle or MPEG packet was found to not be a multiple of 48 bytes. - An idle slot was posted within an AAL5 or MPEG packet at a non 48-byte boundary. - An OAM-F5 slot was posted within an AAL5, Idle or MPEG packet at a non 48-byte boundary. When this interrupt occurs, TC35856BF halts transmission of data, and must be reset for transmission to re-start.
19	PCI_Fatal_Err	RW1C	Indicates that one of the following fatal errors occurred on the PCI bus. - Target abort - Devsel time-out - Data parity error - Address parity error When this interrupt occurs, TC35856BF halts all master DMA operations (including transmission and reception of data), and must be reset for the operations to re-start.
18	Phy_Intr	RW1C	Interrupt line directly from the PHY chip connected to TC35856BF.
17	Ph_Done_Intr	RW1C	Indicates that the last Peephole Interface cycle started has completed.
16	Ph_Access_Err	RW1C	Indicates that an error has been made by the driver in trying to begin a Peephole Interface cycle.
15	Host_Access_Err	RW1C	Indicates illegal register access; - Read-only register has been written - Write-only register has been read - a non-implemented(reserved) address location within the PCI address space allocated to TC35856BF has bee written.

14	Tx_Free_Slot_Unfl	RW1C	Indicates that the driver has underflowed the Tx free slot list by posting too many Tx active slots.
13	Tx_IOC	RW1C	Tx Interrupt-On-Completion. Indicates that one or more of the following have been completely DMA'd to TC35856BF for transmission: - An AAL5 packet. - A raw cell. - All the data from an AAL5 slot for a streaming-mode VC.
12	FM_Resynch_Done	RW1C	Indicates the completion of a FLOWmaster credit resynchronization request initiated by writing FM_Resynch in the Meteor_Cntrl1 register with a 1.
11	Rx_Free_Slot_Ovfl	RW1C	Indicates that the driver has overflowed the Rx free slot FIFO by posting too many Rx free slots.
10	Rx_Data_Fifo_Ovfl	RW1C	Indicates that Meteor's internal Rx data FIFO has overflowed, causing data loss.
9	Rx_No_Big_Slots	RW1C	Indicates that the Rx big free slot FIFO is empty. This interrupt is generated on reception of a cell which causes the FIFO to become empty and on reception of every cell for a big slot while the FIFO remains empty.
8	Rx_No_Small_Slots	RW1C	Indicates that the Rx small free slot FIFO is empty. This interrupt is generated on reception of a cell which causes the FIFO to become empty and on reception of every cell for a small slot while the FIFO remains empty.
7	Rx_No_Raw_Slots	RW1C	Indicates that the Rx raw free slot FIFO is empty. This interrupt is generated on reception of a cell which causes the FIFO to become empty and on reception of every cell for a raw slot while the FIFO remains empty.
6	Rx_Big_Slots_Low	RW1C	Indicates that the Rx big free slot FIFO has less than N slots left in it, where N is specified by the value of the Big_Slot_Low_Th field of the Meteor_Cntrl2 register. This interrupt is generated on reception of a cell which causes the FIFO to fall below threshold and on reception of every cell for a big slot while the FIFO remains below threshold.

5	Rx_Small_Slots_Low	RW1C	Indicates that the Rx small free slot FIFO has less than N slots left in it, where N is specified by the value of the Small_Slot_Low_Th field of the Meteor_Cntrl2 register. This interrupt is generated on reception of a cell which causes the FIFO to fall below threshold and on reception of every cell for a small slot while the FIFO remains below threshold.
4	Rx_Raw_Slots_Low	RW1C	Indicates that the Rx raw free slot FIFO has less than N slots left in it, where N is specified by the value of the Raw_Slot_Low_Th field of the Meteor_Cntrl2 register. This interrupt is generated on reception of a cell which causes the FIFO to fall below threshold and on reception of every cell for a raw slot while the FIFO remains below threshold.
3	Rx_Unopened_VC	RW1C	Indicates that a cell was received for a VC which has not been opened for reception by the driver.
2	Rx_Unknown_VC	RW1C	Indicates that a cell was received for a VC which falls outside of the range specified by the value of the VPI_VCI_Mapping field of the Meteor_Cntrl1 register. This interrupt will only be generated when TC35856BF is not in 64K mode, as specified by the 64k_Mode_Enb field of the Meteor_Cntrl1 register.
1	Rx_Unknown_Ack	RW1C	Indicates that a cell containing a FLOWmaster ack for an unsupported VC was received. The number of VCs supported is specified by the VC_Mode_Sel field of the Meteor_Cntrl1 register. This interrupt will only be generated when TC35856BF is in FLOWmaster mode, as specified by the FM_Enb field of the Meteor_Cntrl1 register.
0	Rx_IOC	RW1C	Rx Interrupt-On-Completion. Indicates that one or more of the following have been received and DMA'd to host memory: - An AAL5 packet. - A raw cell. - All the data for an AAL5 slot for a streaming-mode VC.

Table 5.-8 - Intr_Enb, 0x23

The Intr_Enb register enables and disables the generation of PCI interrupts to the host due to specific interrupt sources. For each bit in the Intr_Status register, there exists a bit in the Intr_Enb register. Bit 0 in the Intr_Status register is paired with bit 0 in the Intr_Enb register, Bit 1 with bit 1, etc. If an Intr_Enb bit is cleared, setting of the corresponding Intr_Status bit will not generate an interrupt to the host. If an Intr_Enb bit is set, setting of the corresponding Intr_Status bit will result in an interrupt to the host. Note that the Intr_Enb bits do not affect the setting and clearing of Intr_Status bits; they only determine which Intr_Enb bits may produce interrupts to the host.

Note also that Tx_IOC interrupts and Rx_IOC interrupts may be masked off both by Intr_Enb bits and by the holdoff and event timers described in Table 5.-9.

Bits	Name	RW	Description
31:21	Reserved		
20:0	Intr_Enb	RW	For each bit in this field, when the bit is set, the corresponding bit in the Intr_Status register may generate host interrupts. When the bit is cleared, the corresponding bit in the Intr_Status register may not generate host interrupts. These bits do not affect the contents of the Intr_Status register.

Table 5.-9- Intr_Hldoff, 0x24

This register provides holdoff capability for Rx_IOC and Tx_IOC interrupts. An event counter and a holdoff timer are provided for each type of interrupt. A single Tx status report represents an event for Tx_IOC purposes. Recall that a Tx status report occurs when data from a Tx EOP slot, or from any slot for a streaming-mode VC is DMA'd. A single received raw cell, consumption of an AAL5 slot for a streaming mode VC, or reception of a complete AAL5 packet represent an event for Rx_IOC purposes.

When **either** the holdoff timer or the event counter requirement is satisfied for a given interrupt type, an interrupt may be generated to the host. Note that the holdoff timers and event counters do not affect the contents of the Intr_Status registers. For example, if a Tx status report occurs ,Tx_IOC will be set in the Intr_Status register, regardless of the state of the Tx_IOC holdoff timer and counter. When either the timer or the counter expires, an interrupt will be generated to the host.

Note that reading this register may yield values different than those which were written, since the same registers used to store the values written are used in the counting logic. This means that the event counters and timers are not self-retriggering. After an interrupt is generated, the register **must** be re-written with the appropriate values for the holdoff parameters to apply to the next interrupt event.

Bits	Name	RW	Description
31	Tx_IOC_Hldoff_Wr	WO	When written with a 1, enables writing of the Tx_IOC_Hldoff_Cnt and Tx_IOC_Hldoff_Time fields below. When written with a 0, the fields are not modified during a write to this register.
30:24	Tx_IOC_Hldoff_Cnt	RW	Specifies the number of Tx ***events*** which must occur before a Tx_IOC interrupt may be generated to the host, where a Tx ***event*** is any of the following: - Data from a Tx EOP raw slot is DMA'd to TC35856BF. - Data from a Tx AAL5 slot for a streaming-mode VC is DMA'd to TC35856BF. - Data from a Tx EOP AAL5 slot is DMA'd to TC35856BF.
23:16	Tx_IOC_Hldoff_Time	RW	Specifies the amount of time which must elapse since this field was last written before a Tx_IOC interrupt may be generated to the host. The range of holdoff times supported is 0 - 128ms, in units of 503.04µs.
15	Rx_IOC_Hldoff_Wr	WO	When written with a 1, enables writing of the Rx_IOC_Hldoff_Cnt and Rx_IOC_Hldoff_Time fields below. When written with a 0, the fields are not modified during a write to this register.
14:8	Rx_IOC_Hldoff_Cnt	RW	Specifies the number of Rx ***events*** which must occur before an Rx_IOC interrupt may be generated to the host, where an Rx ***event*** is any of the following: - An Rx raw slot is filled with data - An Rx AAL5 slot for a streaming-mode VC is filled with data. - A complete AAL5 packet is received. Note that the value of **Rx_Raw_Report_Holdoff** may affect when an in interrupt is actually seen by the driver, since at least 1 status report must take place before an Rx_IOC interrupt is interrupted. In order to minimize spurious interrupts, **Rx_IOC_Hldoff_Cnt** should be programmed with a value greater than or equal to the value (1, 2, 4, or 32) selected by the **Rx_Raw_Report_Holdoff** field of the Meteor_Cntrl2 register.

7:0	Rx_IOC_Hldoff_Time	RW	Specifies the amount of time which must elapse since this field was last written before an Rx_IOC interrupt may be generated to the host. The range of holdoff times supported is 0 - 1ms, in units of 3.93µs.

Table 5.-10 - Tx_FS_List_Ptrs, 0x25

Bits	Name	RW	Description
31	Tx_FS_List_Valid	RW	When written with a 1, indicates to TC35856BF that the Tx free slot list has been created and is ready for use. This bit must be set at initialization time after the Tx free slot list has been created, and must thereafter not be written. Reading it will yield the value 1 unless the free slot list is completely empty, in which case reading it will yield the value 0.
30:28	Reserved		
27:16	Tx_FS_List_Head	RW	Tx free slot list head (read) pointer. This field must be initialized by the driver to the value 0 and must thereafter not be written. Reading it will yield the value of Meteor's read pointer into the Tx free slot list. Read value depends on VC_Mode_Sel. 256VC mode : [27:24] = 0000, [23:16] = valid 1KVC mode : [27:26] = 0000, [25:16] = valid 4KVC mode : [27:16] = valid
15:12	Reserved		
11:0	Tx_FS_List_Tail	RW	Tx free slot list tail (write) pointer. This field must be initialized by the driver to the value 0x0FF in 256-VC mode, and to the value 0x3FF in 1K-VC mode and to the value 0xFFF in 4K-VC mode (as specified by the VC_Mode_Sel field of the Meteor_Cntrl1 register) and must thereafter not be written. Reading it will yield the value of Meteor's write pointer into the Tx free slot list. Read value depends on VC_Mode_Sel. 256VC mode : [11:8] = 0000, [7:0] = valid 1KVC mode : [11:10] = 0000, [9:0] = valid 4KVC mode : [11:0] = valid

Table 5.-11 - Tx_Report_Base, 0x26

Bits	Name	RW	Description
31:6	Tx_Report_Base	RW	Specifies the base of the Tx report queue in host memory in units of 64-bytes.
5:0	Reserved		

Table 5.-12 - Rx_Report_Base, 0x27

Bits	Name	RW	Description
31:6	Rx_Report_Base	RW	Specifies the base of the Rx report queue in host memory in units of 64-bytes.
5:0	Reserved		

Table 5.-13 - Tx_Report_Ptr, 0x28

Bits	Name	RW	Description
31:2	Tx_Report_Ptr	RO	Gives the position of TC35856BF's internal Tx report queue pointer. This pointer indicates the **longword** address in host memory to which TC35856BF will write the next Tx report.
1	Reserved		
0	Tx_Own_Wr_Val	RO	This bit indicates the value which TC35856BF is currently writing to the Own field of Tx report queue entries.

Table 5.-14 - Rx_Report_Ptr, 0x29

Bits	Name	RW	Description
31:2	Rx_Report_Ptr	RO	Gives the position of TC35856BF's internal Rx report queue pointer. This pointer indicates the **longword** address in host memory to which TC35856BF will write the next Rx report.
1	Reserved		
0	Rx_Own_Wr_Val	RO	This bit indicates the value which TC35856BF is currently writing to the Own field of Rx report queue entries.

Table 5.-15 - Rx_Slot_Cong_Th, 0x2A

The Rx_Slot_Cong_Th register is used to specify the global (across all VCs) congestion thresholds for each type of Rx slot. When the number of slots consumed by any single VC exceeds the threshold for the type of slot assigned to the VC, the VC is considered congested. At this point, credit-return for the VC will be inhibited in FLOWmaster mode, or cells for the VC will be dropped in non-FLOWmaster mode. This condition will persist until the number of slots consumed by the VC falls below the congestion threshold. A congestion threshold of 0 disables the congestion detection and reporting mechanism for the specific slot type. **The value 0xFF is illegal, and TC35856BF may behave in undetermined ways if this value is used for any of the congestion thresholds.** This register should be written at initialization time only.

Bits	Name	RW	Description
31:24	Reserved		
23:16	Raw_Slot_Cong_Th	RW	Specifies the number of raw slots which a single VC may be allowed to consume before it is considered congested. A value of 0 disables the congestion detection and reporting mechanism. **The value 0xFF is illegal.**
15:8	Big_Slot_Cong_Th	RW	Specifies the number of big slots which a single VC may be allowed to consume before it is considered congested. A value of 0 disables the congestion detection and reporting mechanism. **The value 0xFF is illegal.**
7:0	Small_Slot_Cong_Th	RW	Specifies the number of small slots which a single VC may be allowed to consume before it is considered congested. A value of 0 disables the congestion detection and reporting mechanism. **The value 0xFF is illegal.**

Table 5.-16 - Tx_ABR_ADTF, 0x2B

The Tx_ABR_ADTF register is used to store 2 ABR ADTF (ACR Decay Timer Factor) values for each of the 2 ABR parameter base sets supported by TC35856BF. Each VC may be assigned to one of the 2 parameter base sets. The units of the ADTF values are 251.52µs. This register should be written at initialization time only.

Bits	Name	RW	Description
31:16	ADTF_2	RW	ADTF for ABR parameter base set #2 in units of 251.52µs.
15:0	ADTF_1	RW	ADTF for ABR parameter base set #1 in units of 251.52µs

Table 5.-17 - Tx_ABR_Nrm_Trm, 0x2C

The Tx_ABR_Nrm_Trm register is used to store 2 sets of Nrm and Trm values for each of the 2 ABR parameter base sets supported by TC35856BF. Each VC may be assigned to one of the 2 parameter base sets. The Nrm values specified should be (log(base 2)N) - 1, where N is the desired value of Nrm in the range 2 to 256. The values of Trm are in units of 251.52µs. The driver should round the desired Trm values **up** when determining the values of Trm to be programmed. This register should be written at initialization time only.

Bits	Name	RW	Description
31:30	Reserved		
29	OFRM_suppress		This bit controls out-of-rate FRM cell generation. When it is set and Disofr is set, out-of-rate FRM cell is not generated while the VC is in ABR idle state. Refer to 8.3.7 for detail operation
28:24	Line_Rate_Exp	RW	Specifies the Exponent adder to be used for ABR calculations. This field depends on the line rate in use, and is given by the expression: Line_Rate_Exp = Int(Log(base2)Line_Rate) Where Line_Rate is in cells/sec, and Int(n) represents the integer portion of n.
23:21	Nrm_2	RW	Nrm for ABR parameter base set #2. The Nrm value specified should be (Log(base 2)N) - 1, where N is the desired value of Nrm in the range 2 to 256.
20:12	Trm_2	RW	Trm for ABR parameter base set #2 in units of 251.52µs.
11:9	Nrm_1	RW	Nrm for ABR parameter base set #1. The Nrm value specified should be (Log(base 2)N) - 1, where N is the desired value of Nrm in the range 2 to 256.
8:0	Trm_1	RW	Trm for ABR parameter base set #1 in units of 251.52µs.

Table 5.-18 - Peephole_Cmd, 0x2D

The Peephole_Cmd register is used in conjunction with the Peephole_Data register to access off-chip devices through Meteor's Peripheral Interface, or off-chip SRAM through Meteor's SRAM interface. Writing this register will clear the Ph_Done bit in the Peephole_Data register and initiate a Peephole cycle.

Bits	Name	RW	Description
31	Ph_Read	RW	When set, indicates that a Peephole read access should be executed. When cleared, indicates that a write access should be executed.
30	Ph_Prf_Access	RW	When set, indicates that an access to an off-chip device through the Peripheral interface should be executed. When cleared, indicates than an access to off-chip SRAM should be executed.
29:18	Reserved		
17:0	Ph_Addr	RW	Specifies the address in SRAM to be accessed when Ph_Prf_Access is cleared, or the off-chip device and address (where applicable) to be accessed when Ph_Prf_Access is set. Valid bits for SRAM access: **256VC or 1K VC mode : Ph_addr[15:0]** **4K VC mode : Ph_addr[17:0]** **Valid bits for off-chip device : Ph_addr[15:0]**

Table 5.-19 - Peephole_Data, 0x2E

The Peephole_Data register is used in conjunction with the Peephole_Cmd register to access off-chip devices through Meteor's Peripheral Interface, or off-chip SRAM through Meteor's SRAM interface.

Bits	Name	RW	Description
31	Ph_Done	RO	When set, indicates that the last peephole operation to be initiated has completed. When cleared, indicates that the peephole interface is busy, and the last operation initiated has not completed. Writing the Peephole_Cmd register will initiate a peephole cycle and clear this bit. When the cycle completes, TC35856BF will set the bit.
30:16	Reserved		

15:0	Ph_Data	RW	When accessing off-chip SRAM, this field provides the 16-bits of data to be written to SRAM or the 16-bits of data to be read from SRAM. When accessing other off-chip devices, only the lower 8 bits are used.

Table 5.-20 - Rx_Raw_Slot_Tag, 0x2F

Bits	Name	RW	Description
31:16	Reserved	RO	
15:2	Rx_Raw_Slot_Tag	RO	Indicates the rx slot_tag of the last raw slot to which a received raw cell was DMA'd. Since Rx raw slots are reported only immediately prior to an Rx_IOC interrupt, this field may be used by the driver to gauge the consumption of Rx raw slots between Rx_IOC interrupts.
1	Reserved		
0	Rx_Raw_Slot_Valid	RO	Indicates that the Rx_Raw_Slot_Tag field contains the rx slot_tag of a slot which has been filled with Rx data but which has not yet been reported to the host.

Table 5.-21 - Rx_CellCnt, 0x30

Bits	Name	RW	Description
31:20	Reserved		
19:0	Rx_CellCnt	RO	Contains the count of the total number of cells received by TC35856BF, not including idle or unassigned cells (whether or not they are carrying FLOWmaster acks). The count is wrapping.

Table 5.-22 - Rx_AAL5_DropPktCnt, 0x31

Bits	Name	RW	Description
31:20	Reserved		
19:0	Rx_AAL5_DropPktCnt	RO	Contains the count of the total number of AAL5 packets dropped by TC35856BF **and not reported to the host.** The count is wrapping.

Table 5.-23 - Rx_Raw_DropCellCnt, 0x32

Bits	Name	RW	Description
31:20	Reserved		
19:0	Rx_Raw_DropCellCnt	RO	Contains the count of the total number of raw cells dropped by TC35856BF **and not reported to the host.** The count is wrapping.

Table 5.-24 - Tx_CellCnt, 0x33

Bits	Name	RW	Description
31:20	Reserved		
19:0	Tx_CellCnt	RO	Contains the count of the total number of cells transmitted by TC35856BF, not including idle or unassigned cells (whether or not they are carrying FLOWmaster acks). The count is wrapping.

Table 5.-25 - Real_Time, 0x34

Bits	Name	RW	Description
31:30	Reserved		
29:8	Real_Time_3p93us	RO	Real-time timer, in units of 3.93µs.
7:0	Real_Time_30ns	RO	Real-time timer, in units of 30ns. Wraps at the value 0x83 (3.93µs), at which point the Real_Time_3p93us field increments.

Table 5.-26 - Current_Time, 0x35

Bits	Name	RW	Description
31:16	Reserved		
15:0	ABR_Current_Time	RO	ABR Current Time, in units of cell intervals at line rate. **This field is used for diagnostic purposes only.**
7:0	CBR_Current_Time	RO	CBR Current Time, in units of cell intervals at line rate. **This field is used for diagnostic purposes only.**

Table 5.-27 - Test_Cntrl, 0x36

The Test_Cntrl register is used solely for testing of TC35856BF, and should never have to be accessed in normal operation.

Bits	Name	RW	Description
31:16	Reserved		
15	ACK_FIFO_read_disable	RW	When set, causes ACK FIFO to stop reading. It is used for ACK FIFO fill up testing.
14:12	reserved		
11	Tx_Partial_Cell_Wait	RO	When set, indicates that the Tx scheduler is waiting for a Tx slot to complete the DMA of an AAL5 cell.
10:8	reserved		
7	Phy_GFC_Halt	RW	When set, forces reception of GFC HALT signals on receive. This bit only has meaning when the PHY interface is in loopback mode.
6	Phy_GFC_Set_A	RW	When set, forces reception of GFC SET_A signals on receive. This bit only has meaning when the PHY interface is in loopback mode.
5	Ph_Test	RW	When set, puts the peephole registers in test mode. In this mode, the Ph_Done bit may be written with any value, and the Peephole_Cmd register may be written without starting a peephole cycle.
4	Force_All_Intr	WO	When written with a 1, sets all interrupt bits.
3:2	reserved		
1	Tx_DMA_Disable	RW	When set, disables Tx DMA operations.
0	Rx_DMA_Disable	RW	When set, disables Rx DMA operations.

Table 5.-28 - MPHY_Cntrl1, 0x37

Bits	Name	RW	Description
31:21	Reserved		
20	MPHY_ADDR	RW	Selects polarity of PHYTXA and PHYRXA input pins. When MPHY_ADDR=0, those inputs are active low, When MPHY_ADDR=1, those inputs are active high. This bit is valid in REVERSE_UTOPIA mode.
19	MPHY_POLLING	RW	Activate MPHY address polling phase in REVERSE_UTOPIA mode. Controlling TXCLAV and RXCLAV depends on MPHY_DIRECT bit.
18	MPHY_DIRECT	RW	Activate MPHY address direct status indication in REVERSE_UTOPIA mode. When set, RXCLAV and TXCLAV are assertd high or low level. When cleared, RXCLAV and TXCLAV are active high tri-state signals from TC35856BF to ATM layer device. TXCLAV or RXCLAV is driven only during each cycle following one with PHTTXA or PHYRXA is asserted, respectively.
17	Suppress_idle_cell	RW	When cleared, Idle/unassigned cell is usually generated when no valid data cell is ready. Idle_cell_mode controls the type of the cell. When set, Idle/unassigned cell generation is suppressed. Cell outputting rate is controlled by cell interval timer.
16	Dual_timer_mode	RW	When set, cell interval timer uses cell_time_1 and cell_time_2, alternatively. When cleared, cell interval timer has 16 bit resolution by concatenating cell_time_1 and cell_time_2.
15:8	cell_timer_2	RW	cell interval timer#2, in units of 30ns. When Dual_timer_mode is off, it becomes the most significant 8 bits of cell interval timer.
7:0	cell_timer_1	RW	cell interval timer#1, in units of 30ns. When Dual_timer_mode is off, it becomes the least significant 8 bits of cell interval timer.

Table 5.-29 - Rx_Buff_Cntrl Registers, 0x38

Bits	Name	RW	Description
31:16	Reserved		
15:0	AAL5_Max_Length	RW	When specified with Rx_AAL5_16K_Lmt (AAL5_Max_Length_Enb), received AAL5 packets will be limited to the specified length in unit of byte which includes the AAL5 trailer. The length must be multiple of 48 bytes. Therefore, bit<3:0> should be always 0. The minimum length is 48 bytes and the maximum length is 65520 bytes. When Rx_AAL5_16K_Lmt (AAL5_Max_Length_Enb) is not set, the specified length is ignored. If received AAL5 packets exceed this length, the excessive bytes will be dropped and an error will be reported.

6. Pin Definitions and Descriptions

6.1 Signal Type Definition

The following signal type definitions are from the view point of TC35856BF. The symbol "-" before a signal name denotes an active-low signal.

Signal Type	Definition
TI	TTL level input pin (5V input type)
TO	TTL level output pin (3V output type)
TIO	TTL level bi-directional pin (5V input/output type)
UI	Universal PCI-bus specific input pin.
UO	Universal PCI-bus specific output pin.
UOD	Universal PCI-bus specific open drain output.
UIO	Universal PCI-bus specific bi-directional pin.
PU	I/O buffer includes pull-up resistor

6.2 Pin Description

Pin description is categorized in following 5 parts;

a) UTOPIA interface (excluding PHYTXA and PHYRXA)
b) Off-chip SRAM interface
c) Peripheral interface
d) PCI bus interface
e) Miscellaneous (including PHYTXA and PHYRXA)

6.2.1 UTOPIA Interface - 26 pins

Signal Name	Pin NO	Type	Description
TXDATA<7:0>	187,188, 189,191, 192,194, 195,196	TO	Transmit cell data bus that carries the ATM cell octets that are written into the transmit FIFO of the SONET Framing chip. This bus is valid only when -TXENB is asserted.
TXSOC	186	TO	Transmit start of cell. When asserted **high**, marks the start of a cell (first octet of an ATM cell) on the TXDATA bus.
TXCLAV	198	TI	Transmit cell available. In cell mode, asserted **high** to indicate that the PHYchip can accept a complete cell. In octet mode, asserted **low** to indicate that the PHY chip can accept no more than 4 octets of data.
-TXENB	202	TO	Transmit write strobe. When asserted **low**, indicates that the TXDATA bus contains valid ATM cell octets.
RXDATA<7:0>	204,206, 207,2,3, 4, 6, 7	TI	Receive cell data bus that carries the ATM cell octets read from the receive FIFO in the PHY chip.
RXSOC	203	TI	Receive start of cell. When asserted **high,** marks the start of cell (first octet of an ATM cell) on the RXDATA bus.
RXCLAV	8	TI	Receive cell available signal. In cell mode, asserted **high** to indicate that a full cell may be read from the PHY chip. In octet mode, asserted **low** to indicate that a single octet may be read from the PHY chip.
-RXENB	9	TO	Receive read strobe. When asserted **low**, indicates that TC35856BF is ready to accept data read from the PHY chip.
PHYCLK	199	TIO (PU)	PHY chip clock that should be used as **both** the transmit write clock and receive read clock for the PHYchip in normal mode. When -REVERSE=L, this pin is the input PHY clock.
PHYINT	28	TI	PHY chip interrupt. When asserted, indicates an unmasked interrupt source is active on the PHY chip. The polarity with which TC35856BF interprets this signal is programmable. This signal is asynchronous to PHYCLK.
-PHYRST	30	TO	**Active-low** reset signal for PHY chip. This signal is asynchronous to PHYCLK.
-REVERSE	29	TI	PHYCLK input enable. When asserted low, PHYCLK pin becomes input mode. When asserted high, PHYCLK pin outputs a half frequency clock of CLK66. It also controls meaning of TEST[2:0] pins.

6.2.2 Off-chip SRAM Interface - 39 pins

Signal Name	Pin NO	Type	Description
SDAT<15:0>	*1	TIO	On board memory data bus that is used to interface with 15ns on board SRAMs that are used to hold VC states, transmit schedule table, VC list, receive free slot lists and other control information.
SADR<16:0>	*2	TO	On board memory address bus that is used to access the data structures in the on board SRAMs.
-SCS<1:0>	180,176	TO	On board memory chip select lines are used to select between two different word-wide banks of SRAM. These signals are **active-low** signals. -SCS<0> controls lower address bank, -SCS<1> controls higher address bank. When SRAM_mode is “1”, -SCS<1> is used as SADR<17>, -SCS<0> is used to indicate SRAM access cycle.
-SWE<1:0>	181,177	TO	On board memory write enable lines when asserted **low** simultaneously with -SCS indicate that TC35856BF is writing one of the word-wide banks of on board SRAMs. -SWE<0> controls lower address bank, -SWE<1> controls higher address bank. When SRAM_mode is “1”, -SWE<0> is used to indicate write enable. -SWE<1> outputs “1”
-SOE<1:0>	179,175	TO	On board memory output enables when asserted **low** simultaneously with -SCS indicate TC35856BF is expecting a given bank of on-board SRAMs to drive the SDAT bus lines. -SOE<0> controls lower address bank, -SOE<1> controls higher address bank. When SRAM_mode is “1”, -SOE<0> is used to indicate write enable. -SOE<1> outputs “1”

NOTE:

*1: SDAT<15:0> 145, 146, 149, 151, 152, 154, 155, 158, 159, 161, 162, 166, 167, 170, 171, 172.

*2: SADR<16:0> 121, 122, 124, 125, 126, 128, 129, 131, 132, 134, 135, 137, 138, 140, 141, 143, 144.

6.2.3 Peripheral Interface - 14 pins

Signal Name	Pin NO	Type	Description
PRFAD<7:0>	16,17, 19, 20, 21,23, 24,25	TIO (PU)	In normal operation, a multiplexed bus that contains address and data for accessing on board control and status registers, MAC address ROMs, PHY device CSRs, etc. This bus drives address bits when the signal -PRFAS is asserted (2 address cycles are supported, so that a 16-bit address space is directly accessible) and drives write data when the signal -PRFDS is asserted. Read data is input through this bus when the -PRFRDY line is asserted.
-PRFWE	11	TIO (PU)	Peripheral Interface write enable output. When asserted **low,** it indicates that TC35856BF is initiating a Peripheral Interface write access. TEST mode control-C input for testing the chip when -REVERSE=L and TEST2=L.
-PRFAS	14	TIO (PU)	Peripheral Interface address strobe. When asserted **low,** it indicates that the PRFAD bus contains the address of a TC35856BF initiated peripheral device access cycle. Asserted low for 2 cycles of PRFCLK; the most significant address bits of an access are driven during the first cycle, and the least significant during the second cycle. TEST mode control-B input for testing the chip when -REVERSE=L and TEST2=L
-PRFDS	26	TIO (PU)	Peripheral Interface data strobe. When asserted **low,** it indicates that TC35856BF is driving data on the PRFAD bus during a write access, or that a peripheral device may drive data on the PRFAD bus during a read access. TEST mode control-A input for testing the chip when -REVERSE=L and TEST2=L
-PRFRDY	13	TI	Peripheral Interface ready signal. When asserted **low** during a write**,** it indicates that the addressed peripheral device has latched the write data. When asserted **low** during a read, it indicates that the peripheral device is driving valid data on the PRFAD bus.

PRFCLK	12	TO	Peripheral Interface clock. 33.33MHz clock to which all Peripheral Interface outputs are synchronized, and to which all inputs must be synchronized.
-PRFRST	31	TO	In normal operation, Peripheral Interface reset. The assertion of this signal is asynchronous to PRFCLK, while the deassertion is synchronous to PRFCLK. This signal is asserted whenever -PCIRST is asserted. **It is not affected by TC35856BF soft reset (bit 0 of the Meteor_Cntrl2 register).** In NAND_Enb test mode, this signal is output of NAND chain

6.2.4 PCI Bus Interface - 50 pins

Signal Name	Pin NO	Type	Description
CLK	37	UI	PCI bus clock that provides the timing for TC35856BF for its PCI bus related operations. Note that in PCI systems the frequency for this clock can vary between nominal DC and 33.333Mhz, but TC35856BF can only guarantee work properly with frequency between 16.667Mhz to 33.333Mhz.
-RST	39	UI	PCI bus reset when asserted low it will force the TC35856BF to undergo a reset process. This signal is asynchronous to the CLK and when asserted, it must stay asserted for at least 20 CLK cycles after the clock is stable.
AD<31:0>	*3	UIO	Multiplexed bus that either contains address during the address phase, or contains data during the data phase of a PCI bus transaction.
-CBE<3:0>	59,75,91, 106	UIO	Multiplexed bus that either contains PCI bus commands during the address phase, or contains byte enables during the data phase of a PCI bus transaction.
PAR	89	UIO	Parity bit that contains the **even parity** across two buses AD<31:0> and -CBE<3:0>. TC35856BF drives this signal one CLK cycle after it was driving those buses, or verifies this signal one CLK cycle after it was receiving from the two buses.
-FRAME	77	UIO	Cycle frame when asserted **low** by a bus master it indicates the beginning and duration of an access.
-IRDY	79	UIO	Initiator ready when asserted **low** by a bus master it indicates that the bus master has placed data on the bus in a write operation, or is ready to accept data on the bus in a read operation.
-TRDY	81	UIO	Target ready when assertion **low** by a bus target agent it indicates that the target agent has placed data on the bus in a ready operation, or is ready to accept data in a write operation.
-STOP	85	UIO	Stop when asserted **low** it indicates the target agent is requesting the bus master to stop the current transaction.

NOTE:

*3: AD<31:0> 45, 47, 48, 50, 51, 54, 55, 57, 63, 65, 66, 68, 69, 71, 72, 74,
92, 94, 95, 98, 99, 100, 102, 103, 107, 109, 110, 112, 113, 115, 116, 118.

PCI Bus Interface - 50 pins - continued -

Signal Name	Pin No	Type	Description
-DEVSEL	83	UIO	Device select when asserted low it indicates the current bus access has been claimed by a target agent.
-REQ	44	UO	Request when asserted low by TC35856BF it indicates that it needs to obtain the PCI bus ownership in order to initiate Host memory access operations.
-GNT	42	UI	Grant when asserted low indicates that Meteor's request for the PCI bus ownership has been granted.
-PERR	86	UIO	Parity error when asserted low it indicates that data parity error has been detected during a PCI transaction.
-SERR	88	UOD	System error when asserted low it indicates that address parity error has been detected during a PCI transaction.
-INTA	41	UOD	Interrupt A when asserted low by TC35856BF it indicates that TC35856BF is requesting an interrupt service from the Host. Note this is a "level sensitive" signal.
IDSEL	62	UI	Initialization Device Select is used as a chip select during configuration read write transactions.

Miscellaneous - 77 pins

Signal Name	Pin NO	Type	Description
CLK66	184	TI	Clock input. Must be connected to a 66MHz oscillator with a duty cycle of 60/40 or better and a precision of 100ppm maximum.
UVDDREF	73	UI	Universal PCI interface voltage select. This pin should be connected to the UVDD pins.
-TEST2	35	TI	Test mode control-A
-TEST1/PHYRXA	34	TI	Test mode control-B when -REVERSE=H or input for decoded MPHY receive address when -REVERSE=L
-TEST0/PHYTXA	33	TI	Test mode control-C when -REVERSE=H or input for decoded MPHY transmit address when -REVERSE=L
UVDD	*4	VDD	Universal PCI I/O supply. These pins should be connected to the PCI Vi/o board contacts on a Universal board, to the +5V contacts on a +5V board, and to the +3.3V contacts on a +3.3V board.
VDD5	*5	VDD	+5V I/O supply
VDD3	*6	VDD	+3.3V input buffer and internal logic supply
VSS	*7	VSS	Common ground

NOTE:

*4: UVDD 36,46,52,53,67,76,84,90,104,105,120
*5: VDD5 1,32,43,58,70,97,111,147,153,160,169,185,201
*6: VDD3 10,56,61,93,114,127,133,139,165,178,190,197
*7: VSS 5,15,18,22,27,38,40,49,60,64,78,80,82,87,96,101,108,117,119,123,130,136,142, 148,150,156,157,163,164,168,173,174,182,183,193,200,205,208

Total: 208 pins

6.3 TC35856BF Pinout

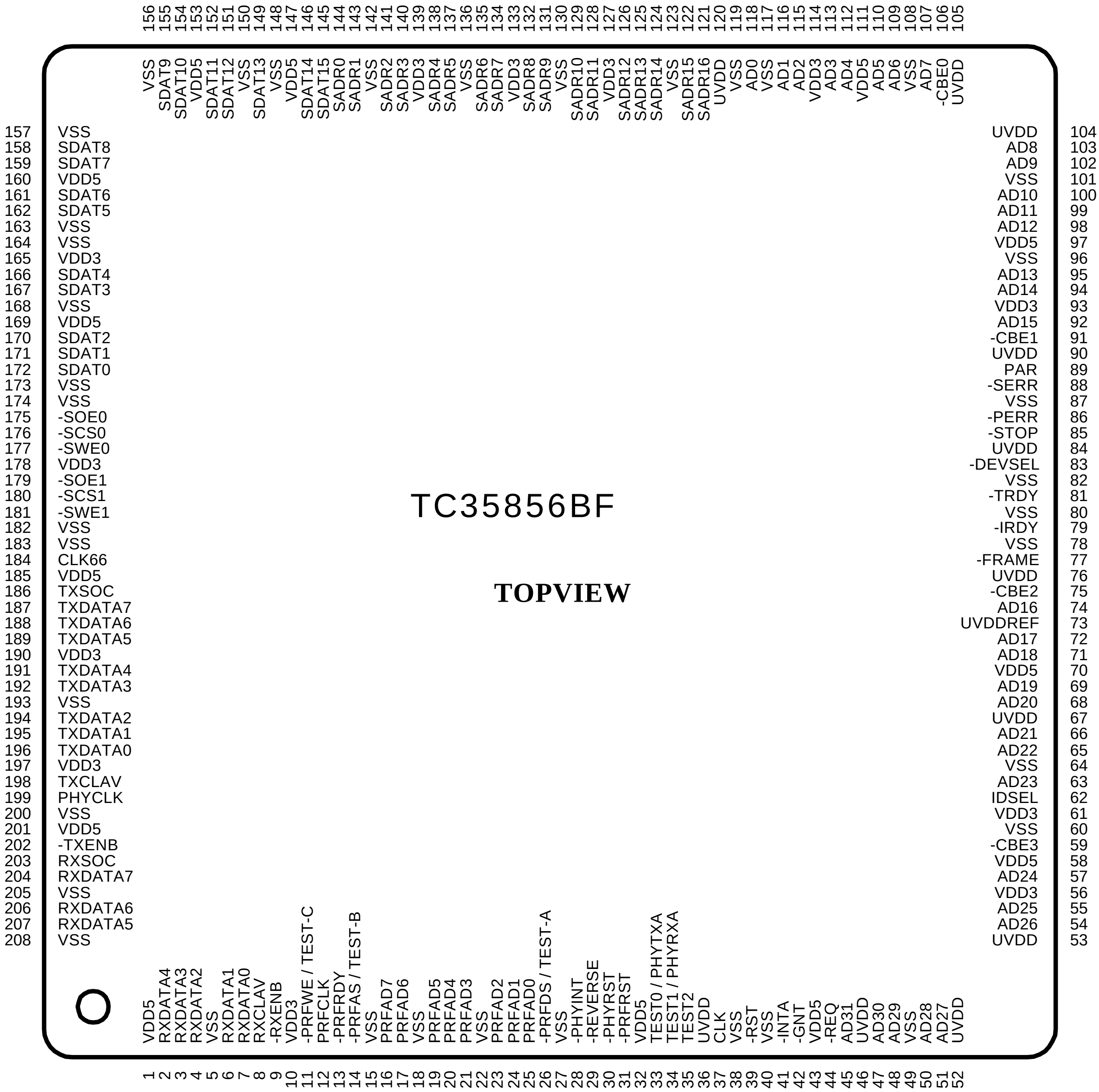


Figure 6.-1 - TC35856BF Pinout

7. Test Support

TC35856BF provides support for chip testing and module-level testing through the **-TEST2, -TEST1, -TEST0, -PRFDS, -PRFAS and -PRFWE** pins.
-REVERSE pin controls which pins are used for test control. When -REVERSE is high, the set of -TEST2, -TEST1 and -TEST0 are mapped to test mode control, namely MODE<2:0>. When -REVERSE is low, the set of -PRFDS, -PRFAS and -PRFWE are mapped to MODE<2:0>. Those mapping is summarized in Table 7.-1

When -REVERSE is high and all three of test control pins are pulled high, TC35856BF operates in normal mode. Any other value on these pins puts TC35856BF in one of several test modes, as described in Table 7.-2.

When -REVERSE is low, -TEST2 controls generic test mode. When -TEST2 is **HIGH,** TC35856BF operates in normal mode. -PRFDS, -PRFAS and -PRFWE are **output** pins and act as normal function. When -TEST2 is **LOW**. the set of -PRFDS, -PRFAS and -PRFWE becomes **input** pins, and test operations are describes in Table 7.-2

The **-PRFRST** pin is used in the NAND_Enb test mode as the output of the NAND chain which includes all of Meteor's inputs and I/Os, with the exception of the **-TEST<2:0>** pins.

The patterns specified in the Pattern1 and Pattern2 test modes are guaranteed to appear at Meteor's outputs only after the **CLK66** and **CLK** inputs have both been toggled a minimum of 4 times each while TC35856BF is in the desired test mode.

Table 7.-1 - TC35856BF Test Mode control pin mapping

	When -REVERSE is "H"	When -REVERSE is "L"
MODE<2>	-TEST2	-PRFDS
MODE<1>	-TEST1	-PRFAS
MODE<0>	-TEST0	-PRFWE

Table 7.-2 - Test Modes

MODE<2:0>	MODE	Description
000	Pattern1	-SCS<1:0> = 11, -SOE<1:0> = 00, -SWE<1:0> = 00 PHYCLK outputs clock at CLK66/2 when -REVERSE is "H". PHYCLK is **input** when -REVERSE is "L". Please refer to Table 7.-3
001	Pattern2	-SCS<1:0> = 00, -SOE<1:0> = 11, -SWE<1:0> = 11 PHYCLK outputs clock at CLK66/2 when -REVERSE is "H". PHYCLK is **input** when -REVERSE is "L". Please refer to Table 7.-3.
010	NAND_Enb	I/Os Function as inputs and are part of NAND tree. This includes PHYCLK also, which will function as an input **regardless** of the state of the -REVERSE pin. All outputs except -PRFRST are tristated. -PRFRST is output of NAND chain.
011	Hi-Z	All Outputs and I/Os are set to Hi-Z.
111	Normal	Normal operation when -REVERSE is HIGH)

Table 7.-3 - Output value for output pin testing

Signal Name	Pin NO	I/O Type	Pattern 1	Pattern 2
TXDATA<7:0>	*1	TO	0xB2	0x4D
TXSOC	186	TO	0	1
-TXENB	202	TO	1	0
-RXENB	9	TO	0	1
PHYCLK	199	TIO(PU)	Toggle or input (*7)	Toggle or input (*7)
-PHYRST	30	TO	1	0
SDAT<15:0>	*2	TIO	0xB2CA	0x4D35
SADR<16:0>	*3	TO	0x12666	0x0D999
-SCS<1:0>	180,176	TO	0x3	0x0
-SWE<1:0>	181,177	TO	0x0	0x3
-SOE<1:0>	179,175	TO	0x0	0x3
PRFAD<7:0>	*4	TIO(PU)	0x6D	0x92
-PRFWE	11	TO	0 (*9)	1 (*9)
-PRFAS	14	TO	1 (*9)	0 (*9)
-PRFDS	26	TO	1 (*9)	0 (*9)
PRFCLK	12	TO	Toggle(*8)	Toggle(*8)
-PRFRST	31	TO	1	0
AD<31:0>	*5	UIO	0xCBCC29CC	0x3433D633
-CBE<3:0>	*6	UIO	0xE	0x1
-INTA	41	UOD	1	0
-REQ	44	UO	0	1
-FRAME	77	UIO	1	0
-IRDY	79	UIO	1	0
-TRDY	81	UIO	1	0
-DEVSEL	83	UIO	1	0
-STOP	85	UIO	1	0
-PERR	86	UIO	0	1
-SERR	88	UOD	0	1
PAR	89	UIO	1	0

*1 TXDATA<7:0> : 187, 188, 189, 191, 192, 194, 195, 196
*2 SDAT<15:0> : 145, 146, 149, 151, 152, 154, 155, 158, 159, 161, 162, 166, 167, 170, 171, 172
*3 SADR<16:0> : 121, 122, 124, 125, 126, 128, 129, 131, 132, 134, 135, 137, 138, 140, 141, 143, 144
*4 PRFAD<7:0> : 16, 17, 19, 20, 21, 23, 24, 25
*5 AD<31:0> : 45, 47, 48, 50, 51, 54, 55, 57, 63, 65, 66, 68, 69 , 71, 72, 74,
92, 94, 95, 98, 99, 100, 102, 103, 107, 109, 110, 112, 113, 115, 116, 118
*6 -CBE<3:0> : 59, 75, 91, 106
*7 PHYCLK : When -REVERSE is asserted to ”H”, PHYCLK outputs a half frequency of CLK66 input. Otherwise, PHYCLK becomes clock input pin.
*8 PRFCLK : PRFCLK outputs a half frequency of CLK66 input.

*9 Those pins are input pins when -REVERSE=L and TEST2=L, and act as MODE<2:0> .

8. Off-Chip SRAM Requirements and Utilization

8.1 Supported Memory Configurations

Number of VCs	Memory size	Two bank Configuration	One bank Configuration
256	about 30.25 KW (60.5 KB)	32K x 8-bits x 2pcs	32K x 8-bits x 2pcs
1024	about 62.5 KW (125 KB)	32K x 8-bits x 4pcs	64K x 16-bit x 1pcs
4096	about 200.5KW(401KB)	128K x 8-bits x 4pcs	256K x 16-bit x 1pcs

8.2 Mapping of SRAM Structures

8.2.1 256-VC mode

One/two bank configuration : 32K * 8-bits * 2 pcs,
- Address[16:15], -SCS1, -SOE1 and -SWE1 are not used.

Item	Entry size * number of entries	Size	Location
RX AAL5 big free slot FIFO	2 W * 512	1 KW	0000 -- 03FF
RX AAL5 small free slot FIFO	2 W * 512	1 KW	0400 -- 07FF
RX raw cell free slot FIFO	2 W * 512	1 KW	0800 -- 0BFF
Reserved		1 KW	0C00 -- 0FFF
RX AAL5 big slot_tags	1 W * 512	0.5 KW	1000 -- 11FF
RX AAL5 small slot_tags	1 W * 512	0.5 KW	1200 -- 13FF
RX raw slot_tags	1 W * 512	0.5 KW	1400 -- 15FF
RX slot_tag VC state table	1W * VC(256)	0.25 KW	1600 -- 16FF
Reserved		0.25 KW	1700 -- 17FF
RX VC state table	8 W * VC(256)	2 KW	1800 -- 1FFF
TX VC state table	8 W * VC(256)	2 KW	2000 -- 27FF
ABR parameter table	8 W * VC(256)	2 KW	2800 -- 2FFF
ABR value table	4 W * VC(256)	1 KW	3000 -- 33FF
RM cell data table	4 W * VC(256)	1 KW	3400 -- 37FF
TX slot descriptors	4 W * 256	1 KW	3800 -- 3BFF
ACR lookup table	1W * 512	0.5 KW	3C00 --3DFF
Reserved		0.5KW	3E00 - 3FFF
ABR schedule table	2 W * 4096	8 KW	4000 -- 5FFF
CBR schedule table #1	up to 1 W * 4096	4 KW	6000 -- 6FFF
CBR schedule table #2	up to 1 W * 4096	4 KW	7000 -- 7FFF
Total (non-Reserved)		30.25 KW	

8.2.2 1K-VC mode

Two bank configuration : 32K * 8-bits * 4 pcs,
- Address[16:15] are not used.

One bank configuration : 64K * 16-bits * 1 pcs,
- Address[16], -SCS1, -SOE1 and -SWE1 are not used.

Item	Entry size * Number of entries	Size	Location
RX AAL5 big free slot FIFO	2 W * 1024	2 KW	0000 -- 07FF
RX AAL5 small free slot FIFO	2 W * 1024	2 KW	0800 -- 0FFF
RX raw cell free slot FIFO	2 W * 1024	2 KW	1000 -- 17FF
Reserved		1 KW	1800 -- 1BFF
RX slot_tag VC state table	1W * VC(1024)	1 KW	1C00 -- 1FFF
RX VC state table	8 W * VC(1024)	8 KW	2000 -- 3FFF
TX VC state table	8 W * VC(1024)	8 KW	4000 -- 5FFF
ABR parameter table	8 W * VC(1024)	8 KW	6000 -- 7FFF
ABR value table	4 W * VC(1024)	4 KW	8000 -- 8FFF
RM cell data table	4 W * VC(1024)	4 KW	9000 -- 9FFF
TX slot descriptors	4 W * 1024	4 KW	A000 -- AFFF
ACR lookup table	1W * 512	0.5 KW	B000 -- B1FF
Reserved		0.5 KW	B200 -- B3FF
RX AAL5 big slot_tags	1W * 1024	1 KW	B400 -- B7FF
RX AAL5 small slot_tags	1W * 1024	1 KW	B800 -- BBFF
RX raw slot_tags	1W * 1024	1 KW	BC00 -- BFFF
ABR schedule table	2 W * 4096	8 KW	C000 -- DFFF
CBR schedule table #1	up to 1 W * 4096	4 KW	E000 -- EFFF
CBR schedule table #2	up to 1 W * 4096	4 KW	F000 -- FFFF
Total (non-Reserved)		62.5 KW	

8.2.3 4K-VC mode

Two bank configuration : 128K * 8-bits * 4 pcs;
One bank configuration : 256K * 16-bits * 1 pcs,
-SOE1 and -SWE1 are not used.

Item	Entry size * Number of entries	Size	Location
RX AAL5 big free slot FIFO	2 W * 4096	8 KW	00000 -- 01FFF
RX AAL5 small free slot FIFO	2 W * 4096	8 KW	02000 -- 03FFF
RX raw cell free slot FIFO	2 W * 4096	8 KW	04000 -- 05FFF
ACR lookup table	1W * 512	0.5 KW	06000 -- 061FF
Reserved		7.5 KW	06200 -- 07FFF
RX VC state table	8 W * VC(4096)	32 KW	08000 -- 0FFFF
TX VC state table	8 W * VC(4096)	32 KW	10000 -- 17FFF
ABR parameter table	8 W * VC(4096)	32 KW	18000 -- 1FFFF
ABR value table	4 W * VC(4096)	16 KW	20000 -- 23FFF
RM cell data table	4 W * VC(4096)	16 KW	24000 -- 27FFF
TX slot descriptors	4 W * 4096	16 KW	28000 -- 2BFFF
ABR schedule table	2 W * 4096	8 KW	2C000 -- 2DFFF
CBR schedule table #1	up to 1 W * 4096	4 KW	2E000 -- 2EFFF
CBR schedule table #2	up to 1 W * 4096	4 KW	2F000 -- 2FFFF
RX AAL5 big slot_tags	1W * 4096	4 KW	30000 -- 30FFF
RX AAL5 small slot_tags	1W * 4096	4 KW	31000 -- 31FFF
RX raw slot_tags	1W * 4096	4 KW	32000 -- 32FFF
RX slot_tag VC state table	1W * VC(4096)	4 KW	33000 -- 33FFF
Total (non-Reserved)		200.5 KW	

8.3 Description of SRAM Structures

This section contains descriptions of all data structure in off-chip SRAM.
All reserved fields will yield undefined values when read and should be written with 0s.

8.3.1 Rx Free Slot FIFOs (3 FIFOs - AAL5 small, AAL5 big, and Raw)

- 2 words/entry, 512 entries in 256-VC mode, 1K entries in 1K-VC mode, 4K entries in 4K-VC mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Base_address (Low word)															
1	Base_address (High word)															

Field	Bits	Description	Driver Init. to:
Base_address	32	Base address in PCI address space of free raw, small, or big rx slots. The least significant 2-bits of this field will always be 0, since all rx slots are longword aligned.	X

8.3.2 Rx Slot_Tags (3 areas - AAL5 small, AAL5 big, and Raw)

- 1 word/entry, 512 entries in 256-VC mode, 1K entries in 1K-VC mode, 4K entries in 4K-VC mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Reserved		Slot_Tag													

Field	Bits	Description	Driver Init. to:
Slot_Tag	14	14-bit slot tag associated with a single entry in one of the 3 rx free slot FIFOs.	X

8.3.3 Rx Slot_Tag VC State Table

- 1 word/entry, 1 entry/VC

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Reserved		Slot_Tag													

Field	Bits	Description	Driver Init. to:
Slot_Tag	14	14-bit slot tag of the rx slot currently in use by the VC associated with the entry.	X

8.3.4 Rx VC State Table

- 8 words/entry, 1entry/VC

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CRC10_En	Drop	Open	Congestion	Cell_loss	Rx_AAL5_SM	Slot_type		Slots_Consumed							
1	AAL5_length															
2	AAL5_Slot_Address (Low word)															
3	AAL5_Slot_Address (High word)															
4	Rx_AAL5_CRC (Low word)															
5	Rx_AAL5_CRC (High word)															
6	Reserved			Slot_Pointer												
7	In_AAL5_pkt	Or_CI	Or_CLP	FM	Ex_RATO				Ack_count							

Field	Bits	Description	Driver Init. to:
CRC10_En	1	CRC10 checking enable bit.	0,1
Drop	1	Flag to indicate that cells received on this VC should be dropped. Set under any of 4 conditions: 1) VC is congested 2) Partial AAL5 packet has been lost due to Rx Data FIFO overflow. 3) Partial AAL5 packet has been lost due to free slot exhaustion. 4) Partial AAL5 packet has been lost due to AAL5 max packet length error.	0
Open	1	Indicates that VC has been opened by the driver.	0,1
Congestion	1	Flag to indicate that the VC is congested (slots_consumed = congestion_threshold when an attempt at new slot allocation was made).	0
Cell_loss	1	Flag to indicate that cell loss on this VC has occurred and should be reported to the host.	0
Rx_AAL5_SM	1	Rx AAL5 streaming-mode enable (report and interrupt per slot).	0,1
Slot_Type	2	Selects which type of free slots to use for data received on this VC: 00 - Small 01 - Big 10 - Raw 11 - Reserved	00, 01, 10
Slots_consumed	8	Indicates the number of slots consumed by this VC and not yet returned by the host.	0x00
AAL5_length	16	Counter to count number of received bytes in AAL5 packets. Used for error checking, to compare with received AAL5 length field and to detect AAL5 packets which exceed (64K-1) bytes in length.	0x0000
AAL5_Slot_Address	32	Base address of the AAL5 rx slot currently being used to receive data on this VC.	X
Rx_AAL5_CRC	32	Partial CRC-32 for received AAL5 packets.	0xFFFFFFFF
Slot_pointer	13	Longword write pointer into current rx slot.	0x0000
In_AAL5_pkt	1	Flag to indicate that an AAL5 packet is being reassembled on this VC.	0
Or_CI	1	Logical OR of CI bits (PT<1>) of received cells in an AAL5 packet. Cleared at initialization by the driver and at the beginning of each AAL5 packet by TC35856BF.	0
Or_CLP	1	Logical OR of CLP bits of received cells in an AAL5 packet. Cleared at initialization by the driver and at the beginning of each AAL5 packet by TC35856BF.	0
FM	1	FLOWmaster enable.	0,1
Ex_RATO	4	Expected reassembly time-out time. Used to detect reassembly time-out of AAL5 packets.	X
Ack_count	8	FLOWmaster acknowledge counter. Used to accumulate FLOWmaster acks when the ack FIFO is full or when the VC is congested.	0x00

8.3.5 Tx VC State Table

- 8 words/entry, 1entry/VC

<table>
<tr><th></th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="2">chtyp</td><td>Tx_AAL5_SM</td><td>FM</td><td>ABS</td><td colspan="2">RC</td><td>CLP</td><td colspan="4">rsv</td><td colspan="4">prescale_val</td></tr>
<tr><td>1</td><td>rsv</td><td colspan="3">MPEG_count</td><td colspan="4">prescale_count</td><td colspan="8">credit</td></tr>
<tr><td>2</td><td colspan="16">tx_AAL5_CRC (Low word)</td></tr>
<tr><td>3</td><td colspan="16">tx_AAL5_CRC (High word)</td></tr>
<tr><td>4</td><td>Active</td><td>sch</td><td>OR_BRM</td><td>Turn</td><td colspan="12">tx_A_Slot_H</td></tr>
<tr><td>5</td><td colspan="4">rsv</td><td colspan="12">tx_A_Slot_T</td></tr>
<tr><td>6</td><td>rsv</td><td>Idle</td><td colspan="14">tx_Slot_Ptr</td></tr>
<tr><td>7</td><td colspan="4">rsv</td><td colspan="12">VC_link</td></tr>
</table>

Field	Bits	Description	Driver Init. to:
Chtyp	2	Channel type: 00 - Raw 01 - Reserved 10 - AAL5 11 - MPEG	00, 10, 11
Tx_AAL5_SM	1	Tx AAL5 streaming mode enable (report and interrupt per slot).	0,1
FM	1	FLOWmaster enable.	0,1
ABS	1	ABR parameter base set selection.	0,1
RC	2	Rate Control: 00 - CBR 01 - ABR 10 - UBR 11 - Reserved	00, 01, 10
CLP	1	Cell Loss Priority transferred to ATM header of transmitted AAL5 cells.	0,1
Prescale_val	4	Prescale counter initial value for CBR rate control.	0x0 - 0xF
MPEG_count	3	Counter used for counting the number of cells in an MPEG packet.	0x0
Prescale_count	4	Down-counter used for slowing down CBR/ABR/UBR rates.	0x0
Credit	8	FLOWmaster transmit credit counter.	0x00 - 0xFF
tx_AAL5_CRC	32	Partial CRC-32 for transmitted AAL5 packets.	0xFFFFFFFF
Active	1	Flag to indicate that data exists in the active slot list for a VC.	0
Sch	1	Flag to indicate that a VC is scheduled in the ABR/UBR schedule table.	0
OR_BRM	1	Flag to indicate that an Out-of-rate backward RM cell should be transmitted.	0
Turn	1	Flag to indicate that a backward RM turn-around cell is pending.	0
tx_A_Slot_H	12	Tx active slot list head pointer.	X
tx_A_Slot_T	12	Tx active slot list tail pointer.	X
Idle	1	Flag to indicate that the current slot being processed on a VC is an idle slot.	0
tx_Slot_Ptr	14	Byte offset into the current slot from which transmit data is being DMA'd on a VC.	X
VC_link	12	ABR/UBR scheduling pointer.	X

8.3.6 ABR Parameter Table

- 8 words/entry, 1 entry/VC

<table>
<tr><th></th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td>rsv</td><td colspan="3">CDFx</td><td colspan="4">RDFx</td><td colspan="3">Crmx</td><td colspan="5">rsv</td></tr>
<tr><td>1</td><td>rsv</td><td colspan="15">AIR</td></tr>
<tr><td>2</td><td>rsv</td><td colspan="15">ICR</td></tr>
<tr><td>3</td><td>rsv</td><td colspan="15">MCR</td></tr>
<tr><td>4</td><td>rsv</td><td colspan="15">PCR</td></tr>
<tr><td>5</td><td colspan="15">rsv</td><td>CI-VC</td></tr>
<tr><td>6</td><td colspan="16">rsv</td></tr>
<tr><td>7</td><td colspan="16">rsv</td></tr>
</table>

Field	Bits	Description	Driver Init. to: (for ABR VCs)
CDFx	3	ABR Cutoff Decrease Factor (CDF) specifier. This parameter controls the decrease in ACR associated with CRM. CDF is in the range 1/64 - 1, by powers of 2. CDFx = Log(1/CDF), or 7 if CDF = 0.	Log(1/CDF), or 7 if CDF = 0
RDFx	4	ABR Rate Decrease Factor (RDF) specifier. This parameter controls the decrease in the cell transmission rate RDF is in the range 1/1024 - 1, by powers of 2. RDFx = Log(1/RDF), or 11 if RDF = 0. Valid range of RDFx is 0 - 11.	Log(1/RDF), or 11 if RDF = 0
Crmx	3	Missing RM cell count (Crm) specifier. This parameter specifies the number of FRM cells which may be sent without receiving a BRM cell. Crm is in the range 2 - 256, by powers of 2 (note that in actuality, 255 is implemented instead of 256). Crmx = Log(Crm) - 1.	Log(Crm) - 1
AIR	15	Additive Increase Rate (PCR*RIF). 15-bit floating-point.	AIR
ICR	15	Initial Cell Rate. 15-bit floating-point.	ICR
MCR	15	Minimum Cell Rate. 15-bit floating point.	MCR
PCR	15	Peak Cell Rate. 15-bit floating point.	PCR
CI-VC	1	Logical OR of 2 conditions: - The last data cell received on this VC had its EFCI bit set. - TC35856BF congestion has been detected (driver generated, unconditional slot-specific congestion or VC-specific congestion). This bit is set by the Rx block on cell receive, and cleared by the Tx block when it is copied into a turn-around backward RM cell.	0

8.3.7 ABR Value Table

- 4 words/entry, 1 entry/VC

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	rsv	ACR														
1	FirstTrn	ABR_active	ForceFRM	LastRMValid	Disofr	Adtfsel	rsv	Fraction								
2	LastRM															
3	CrmCnt							NrmCnt								

Field	Bits	Description	Driver Init. to:
ACR	15	Allowed Cell Rate. 15-bit floating point.	X for ABR/CBR, PCR for UBR
FirstTrn	1	A flag indicating the turn-around RM has priority over data cells	X
ABRactive	1	A flag indicating ABR value are valid. If it is clear, all variable should be initialized.	0
ForceFRM	1	When set, TC35856BF sends a forward RM cell instead of a data cell.	0,1
LastRMValid	1	Indicates that the LastRM field defined below is valid. Cleared when LastRM exceeds 12.3s, since LastRM is capable of counting to 16.48s only.	X
Disofr	1	Controls out-of-rate Forward RM cell generation in conjunction with OFRM_suppress bit in Meteor_Cntrl_1 CSR.	0,1 for ABR
Adtfsel	1	It selects ADTF parameter which is used with OFRM_suppress bits	0,1 for ABR
Fraction	9	Fractional component of cell interval used to schedule ABR/UBR flows into schedule table.	X for ABR/CBR, 0 for UBR
LastRM	16	LastRM is the time that the most recent RM cell was sent in units of 251.52μs.	X
CrmCnt	8	Number of forward RM cells send without an RM received	X
NrmCnt	8	Number of cells since the last FRM.	X

Out-of-rate Forward RM (FRM) cell generation is specified in the following table. Note that OFRM_suppress affects all VCs and Disofr is VC specific parameter.

OFRM_suppress	Disofr	Out-of-rate FRM
0	0	Out-of-rte FRM is generated whenever 100ms elapses after the previous FRM.
0	1	Out-of-rate FRM is never generated
1	0	Out-of-rte FRM is generated whenever 100ms elapse after the previous FRM.
1	1	Out-of-rate FRM is generated before ADTF time elapses after in-rate cell transmission. After ADTF time elapses, out-of-rate FRM is not generated until in-rate cell is transmitted.

8.3.8 RM Cell Data Table

- 4 words/entry, 1 entry/VC

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	PI								MT							
1	rsv	ER_nz	ER_e					ER_m								
2	rsv	CCR_nz	CCR_e					CCR_m								
3	rsv	MCR_nz	MCR_e					MCR_m								

Field	Bits	Description	Driver Init. to:
MT	8	Backward RM cell message type	X
PI	8	Backward RM cell Protocol Identifier (this field should always = 1)	X
XX_m	9	Backward RM cell FP mantissa used to specify Explicit Cell Rate (ER), Current Cell Rate (CCR), and Minimum Cell Rate (MCR).	X
XX_e	5	Backward RM cell FP exponent used to specify Explicit Cell Rate (ER), Current Cell Rate (CCR), and Minimum Cell Rate (MCR).	X
XX_nz	1	Backward RM cell "not-zero" bit used to specify Explicit Cell Rate (ER), Current Cell Rate (CCR), and Minimum Cell Rate (MCR).	X

8.3.9 Tx Slot Descriptors

- 4 words/entry, 1 entry/VC for TC35856BF-1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	rsv	Raw	Idle	Mgmt	Link											
1	CRC10_En	EOP	Slot_size													
2	Base_address (Low word)															
3	Base_address (High word)															

Field	Bits	Description	Driver Init. to:
Raw	1	Flag to indicate that the slot contains a raw cell.	X
Idle	1	Flag to indicate that the slot contains an idle cell.	X
Mgmt	1	Flag to indicate that the slot contains an F4 or F5 OAM cell.	X
Link	12	Linked-list pointer to the next slot descriptor in a free or active slot list.	N+1, where N is the VC of the entry.
CRC10_En	1	Enable CRC10 generation on the cells DMA'd from the slot.	X
EOP	1	End_of_Packet indicator.	X
Slot_size	14	Length of the slot in bytes.	X
Base_address	32	Base address in PCI address space of the tx slot.	X

8.3.10 ABR/UBR Schedule Table

- 2 words/entry, 4K entries

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Active	reserved			Abr_Pointer_Low											
1	reserved				Abr_Pointer_High											

Field	Bits	Description	Driver Init. to:
Active	1	Flag to indicate that the data in the ABR/UBR schedule table entry is valid.	0
ABR_Pointer_Low	12	Pointer used in scheduling ABR/UBR flows.	X
ABR_Pointer_High	12	Pointer used in scheduling ABR/UBR flows.	X

8.3.11 CBR Schedule Tables

- 1 word/entry, 2 - 4K entries/table, 2 tables

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Active	EOT	reserved		VC											

Field	Bits	Description	Driver Init. to:
Active	1	Flag to indicate that the data in the CBR schedule table entry is valid.	1 if CBR slot is used, 0 otherwise.
EOT	1	Flag to indicate that this entry is located at the end of the table.	1 for End-of-table entry, 0 otherwise.
VC	12	CBR VC number.	CBR VC number if CBR slot is used, X otherwise.

8.3.12 ACR Lookup Table

- 1 word/entry, 512 entries

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved						Cell_Interval									

Field	Bits	Description	Driver Init. to:
Cell_Interval	10	Mapping of ACR to 10-bit Cell_Interval (1/ACR in units of cell_times).	Fixed values depending on line rate (see TBD).

Preliminary

9. ELECTRICAL CHARACTERISTIC

9.1 ABSOLUTE MAXIMUM RATINGS

Nov, 1996

Vdd(5V)	Supply Voltage(5V)	-0.5V to 7.0V
Vdd(3V)	Supply Voltage(3V)	-0.5V to 5.0V
Vdd(PCI)	Supply Voltage(PCI)	T.B.D.
Vin(5V)	Input Voltage(5V)	-0.5V to Vdd(5V)+0.5V
Vin(PCI)	Input Voltage(PCI)	T.B.D.
Pd	Power Dissipation	1.5W
Tsolder	Soldering Temperature (10sec)	240°C
Tstg	Storage Temperature	-65°C to 150°C
Topr	Operating Temperature	0°C to 70°C

9.2 DC CHARACTERISTICS

Nov, 1996

		Min.	Typ.	Max.	Unit
Vdd(5V)	Supply Voltage(5V)	4.75	5.00	5.25	V
Vdd(3V)	Supply Voltage(3.3V)	3.0	3.3	3.6	V
Iddd(5V)	Operating Current(5V)	T.B.D.			
Iddd(3V)	Operating Current(3.3V)	T.B.D.			
Iih	Input High Current	T.B.D.			
Iil	Input Low Current	T.B.D.			

5V Input/Output and bi-directional

Oct., 1997

			Min.	Typ.	Max.	Unit
Vih	Input High Voltage		2.0	-	-	V
Vil	Input Low Voltage		-		0.8	V
Voh	Output High Voltage	IOH= 0 mA	-	-	Vdd (5V)	V
		IOH= -2.5mA	2.4	-	-	
Vol	Output Low Voltage	IOL= 2.5mA	-	-	0.4	V

3V Output

Oct., 1997

			Min.	Typ.	Max.	Unit
Voh	Output High Voltage	IOH= 0 mA	-	-	Vdd(3V)	V
		IOH= -2.5mA	2.4	-	-	
Vol	Output Low Voltage	IOL= 2.5mA	-	-	0.4	V

Preliminary

UTOPIA I/F Input/Output

Nov, 1996

		Min.	Typ.	Max.	Unit
Vih	Input High Voltage	2.0	-	-	V
Vil	Input Low Voltage	-		0.8	V
Voh	Output High Voltage (IOH=-4.0mA)	2.4	-	-	V
Vol	Output Low Voltage (IOL=4.0mA)	-	-	0.4	V

DC characteristics for 5V PCI

Nov, 1996

		Min.	Typ.	Max.	Unit
Vdd(PCI)	Supply Voltage(5V)	4.75	5	5.25	V
Vih	Input High Voltage	2.0	-	-	V
Vil	Input Low Voltage	-		0.8	V
Voh	Output High Voltage (IOH=-2.0mA)	2.4	-	-	V
Vol	Output Low Voltage (IOL=6mA)	-	-	0.55	V
Iih	Input High Current			70	ƒ ÊA
Iil	Input Low Current			-70	ƒ ÊA

DC characteristics for 3.3V PCI

Nov, 1996

		Min.	Typ.	Max.	Unit
Vdd(PCI)	Supply Voltage(5V)	3.0	3.3	3.6	V
Vih(PCI)	Input High Voltage	0.5Vdd(PCI)	-	-	V
Vil(PCI)	Input Low Voltage	-		0.4Vdd(PCI)	V
Voh	Output High Voltage (IOH=-2.0mA)	0.9Vdd(PCI)	-	-	V
Vol	Output Low Voltage (IOL=6mA)	-	-	0.1Vdd(PCI)	V
Iil	Input Leakage			+/- 10	ƒ ÊA

Preliminary

10. AC CHARACTERISTICS

10.1 Clock Input timing

10.1.1 CLK66 input timing

Jan, 1997

No.	Parameter	Min	Typ.	Max	Unit
1	tCYC66 CLK66 cycle time	15ns ± 50ppm			ns
2	tDTY66 CLK66 duty cycle	50±10			%

NOTE: All timing requirements are preliminary and subject to change upon production release.

10.1.2 CLK input timing

Nov, 1996

No.	Parameter	Min	Typ.	Max	Unit
1	tCYC CLK cycle time (*1)	30		60	ns
2	tHIGH CLK high time	11		-	ns
3	tLOW CLK low time	11		-	ns

NOTE: TC35856F can only guarantee work properly with cycle time between 30ns and 60ns.
All timing requirements are preliminary and subject to change upon production release.

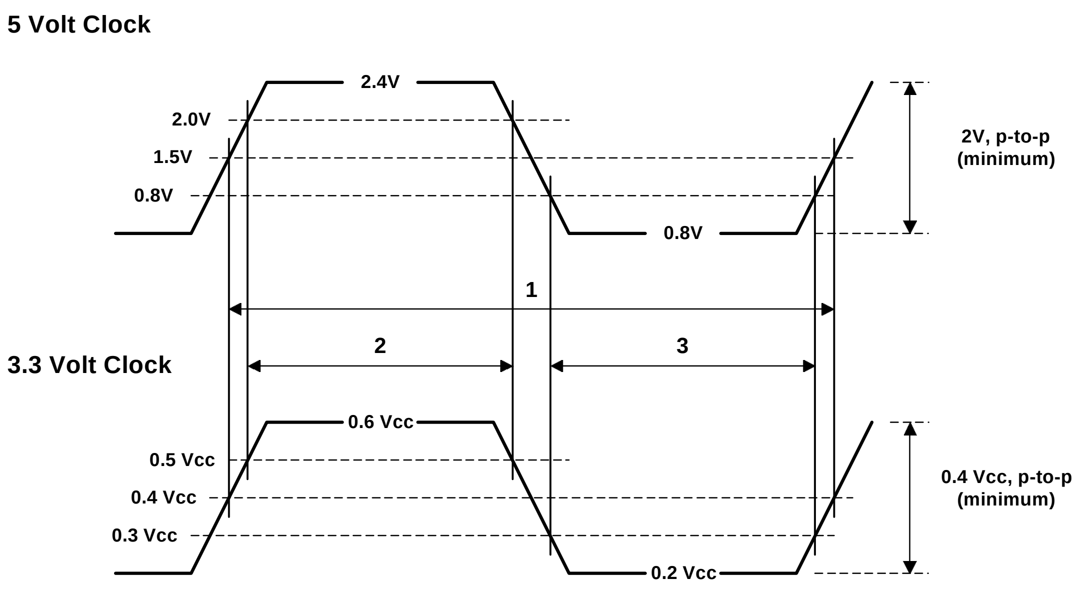


Figure 10.-1 - PCI clock input requirement

Preliminary

10.2 PCI I/F timing

10.2.1 PCI I/F timing specification

Nov, 1996

No.	Parameter		Min	Typ.	Max	Unit
1	tVAL	CLK to Signal Valid Delay -bussed signals	2		11	ns
1	tVAL(ptp)	CLK to Signal Valid Delay -point to point	2		12	ns
2	tON	Float to Active Delay	2		-	ns
3	tOFF	Active to Float Delay	-		28	ns
4	tSU	Input Set up Time to CLK -bussed signals	7		-	ns
4	tSU(ptp)	Input Set up Time to CLK (-GNT signal)	10		-	ns
5	tH	Input Hold Time from CLK	0		-	ns

NOTE: All timing requirements are preliminary and subject to change upon production release.

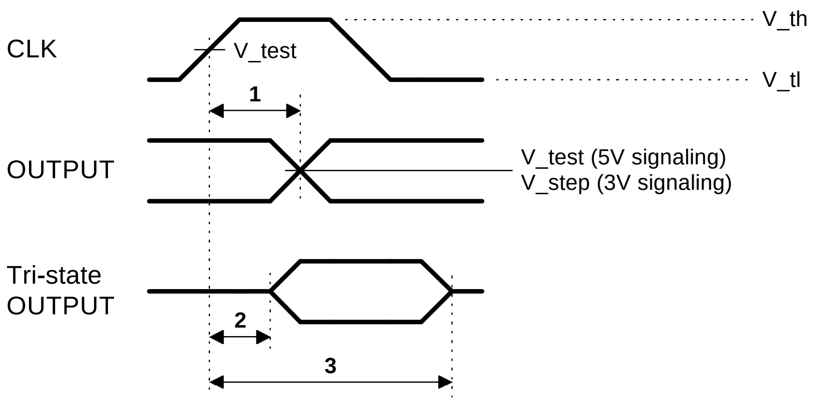


Figure 10.-2 - PCI I/F output timing

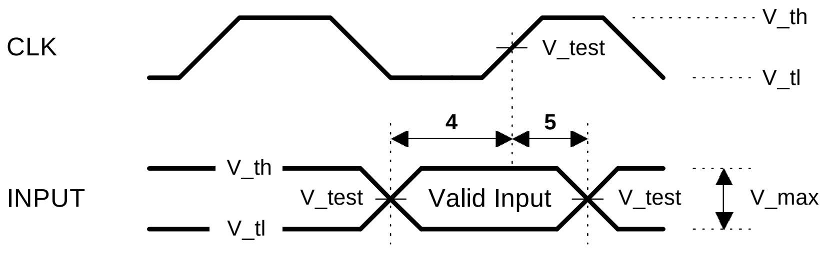


Figure 10.-3 - PCI I/F Input timing

Symbol	5V signaling	3V signaling	Unit
V_th	2.4	0.6Vcc	V
V_tl	0.4	0.2Vcc	V
V_test	1.5	0.4Vcc	V
V_step(rising)	n/a	0.285Vcc	V
V_step(falling)	n/a	0.615Vcc	V
V_max	2.0	0.4Vcc	V

Preliminary

10.3 Off chip SRAM I/F Timing

10.3.1 SRAM read timing

Nov, 1996

No.	Parameter		Min	Typ.	Max	Unit
1	tACC	Address access time	-		17.0	ns
2	tOH	Read valid data hold time from valid address	0.0		-	ns
3	tCO	-SCS↓ to Read data valid	-		17.0	ns
4	tCOV	-SCS↓ to -SOE↓ time	2.0		-	ns
5	tODO	SRAM data invalid time from -SOE↑	-		8.0	ns
6	tOCH	-SCS hold time from -SOE↑	0.0		-	ns
7	tOE	-SOE↓ to read data valid	-		12.0	ns
8	tOEE	-SOE↓ to SRAM output drive	0.0		-	ns
9	tOW	-SOE↑ to Write data drive	8.0		-	ns

NOTE: All timing requirements are preliminary and subject to change upon production release.

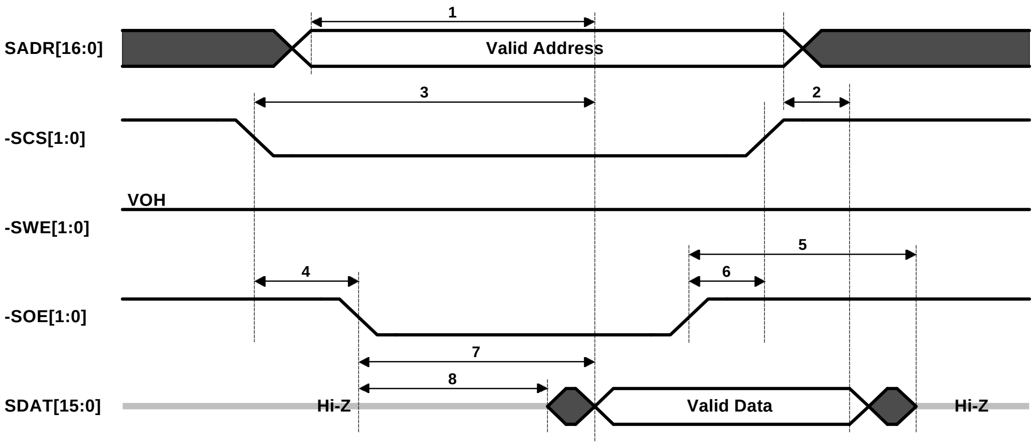
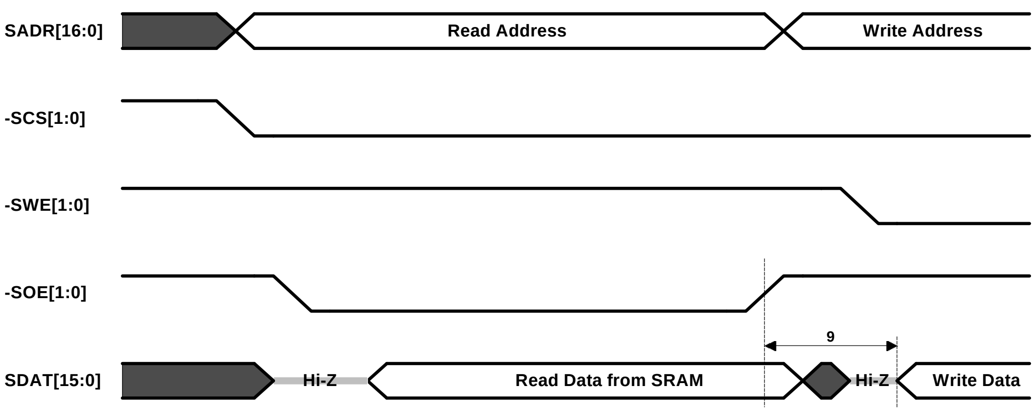


Figure 10.-4 - Single SRAM read timing



NOTE: TC35856F keeps to read SRAM at same address, if no succeeding read or write cycle follows the read cycle

Figure 10.-5 - Read cycle before Write cycle

Preliminary

10.3.2 SRAM Write timing

Nov, 1996

No.	Parameter		Min	Typ.	Max	Unit
1	tAW	Address valid to write end time	15.0		-	ns
2	tWR	Address hold time	4.0		-	ns
3	tAS	Address setup time	4.0		-	ns
4	tCW	Chip select time	15.0		-	ns
5	tWP	-SWE width time	12.0		-	ns
6	tDS	Write data setup time	10.0		-	ns
7	tDH	Write data hold time	4.0		-	ns
8	tWO	Data Hi-Z to -SOE↓	4.0		-	ns

NOTE: All timing requirements are preliminary and subject to change upon production release.

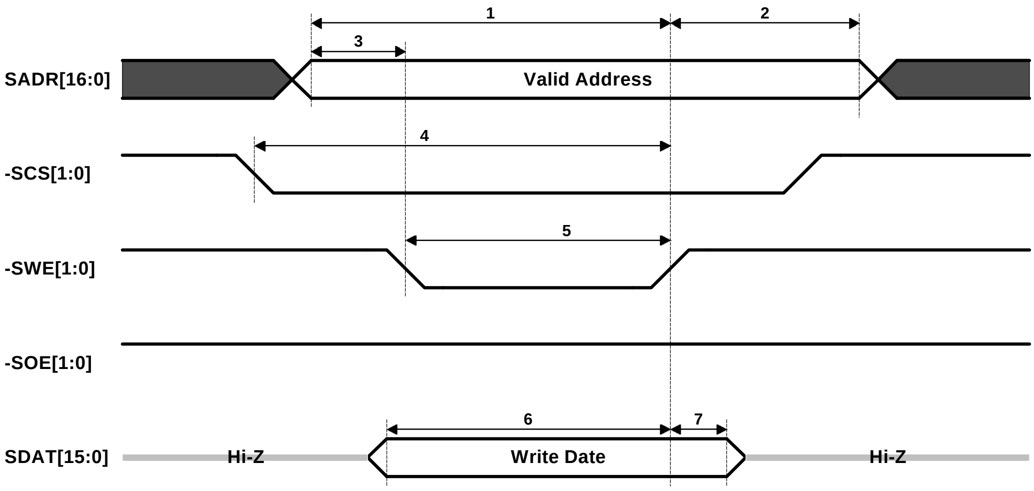
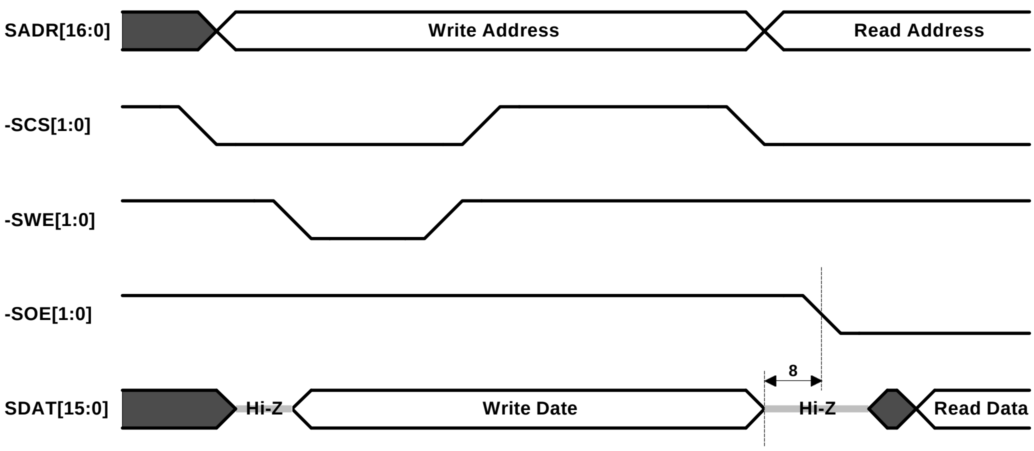


Figure 10.-6 - Single SRAM write timing



Note: TC35856F keeps outputting write data on SDAT bus, if no succeeding read or write cycle follows the write cycle.

Figure 10.-7 - Write cycle before Read cycle

10.4 UTOPIA I/F Timing

10.4.1 UTOPIA I/F Timing specification

Nov, 1996

No.	Parameter	Min	Typ.	Max	Unit
1	tUOD UTOPIA output data valid delay from PHYCLK↑	1.0		21	ns
2	tUIS UTOPIA input data setup time from PHYCLK↑	8.0		-	ns
3	tUIH UTOPIA input data hold time from PHYCLK↑	1.0		-	ns

NOTE: All timing requirements are preliminary and subject to change upon production release.

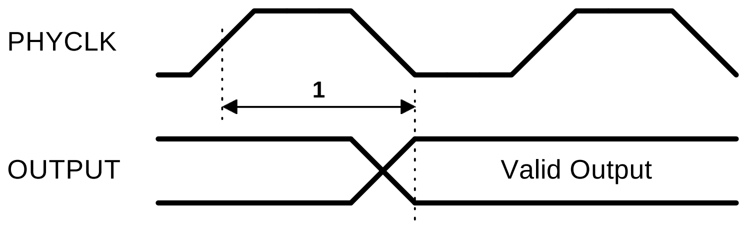


Figure 10.-8 - UTOPIA I/F output timing

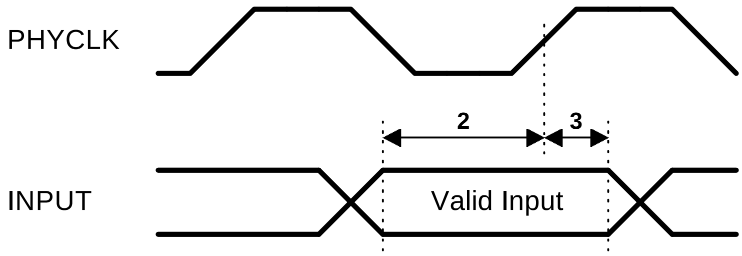


Figure 10.-9 - UTOPIA I/F input timing

Preliminary

10.5 Peripheral I/F Timing

10.5.1 Peripheral I/F Read Timing

Nov, 1996

No.	Parameter		Min	Typ.	Max	Unit
1	tCKW	PRFCLK pulse width		30		ns
2	tASD	PRFAS valid delay from PRFCLK↑	-		18.0	ns
3	tASH	PRFAS hold time from PRFCLK↑	2.0		-	ns
4	tDSD	PRFDS valid delay from PRFCLK↑	-		18.0	ns
5	tDSH	PRFDS hold time from PRFCLK↑	2.0		-	ns
6	tRDYS	PRFRDY setup time from PRFCLK↑	15.0		-	ns
7	tRDYH	PRFRDY hold time from PRFCLK↑	0.0		-	ns
8	tHAD	PRFAD high address valid delay from PRFCLK↑	-		18.0	ns
9	tHAH	PRFAD high address hold time from PRFCLK↑	2.0		-	ns
10	tLAD	PRFAD low address valid delay from PRFCLK↑	-		18.0	ns
11	tLAH	PRFAD low address hold delay from PRFCLK↑	2.0			ns
12	tADZ	PRFAD low address disable time from PRFCLK↑	-		18.0	ns
13	tRDS	Read data setup time from PRFCLK↑	15.0		-	ns
14	tRDH	Read data hold time from PRFCLK↑	0.0		-	ns
15	tZAD	PRFAD high address enable time from PRFCLK↑	-		18.0	ns

NOTE: All timing requirements are preliminary and subject to change upon production release.

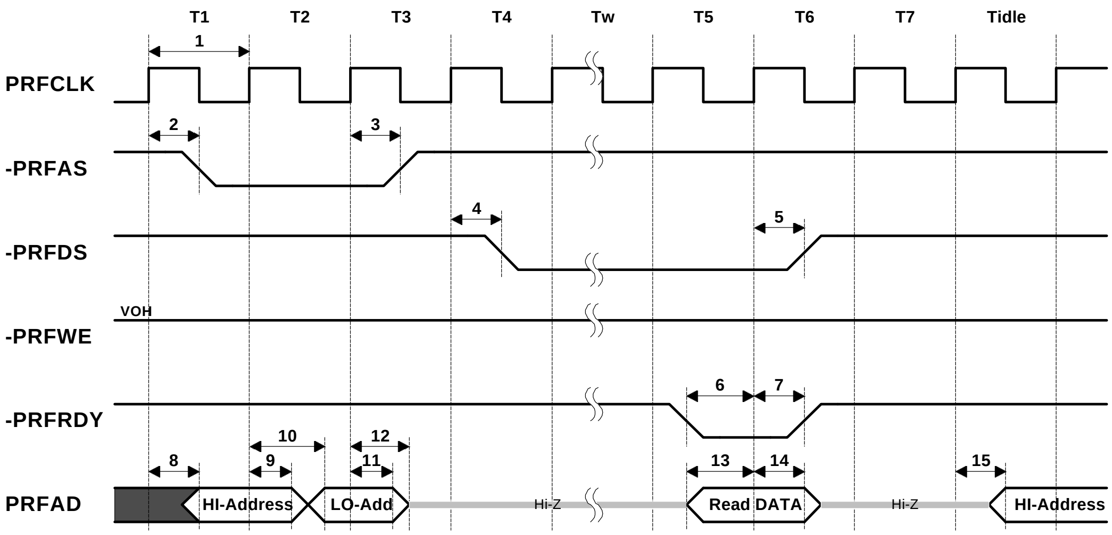


Figure 10.-10 - Peripheral I/F read timing

Preliminary

10.5.2 Peripheral I/F Write Timing

Nov, 1996

No.	Parameter	Min	Typ.	Max	Unit
1	tWED PRFWE valid delay from PRFCLK↑	-		18.0	ns
2	tWEH PRFWE hold time from PRFCLK↑	2.0		-	ns
3	tWDD PRFAD write data valid delay from PRFCLK↑	-		18.0	ns
4	tWDH PRFAD write data hold time from PRFCLK↑	2.0			ns

NOTE: All timing requirements are preliminary and subject to change upon production release.

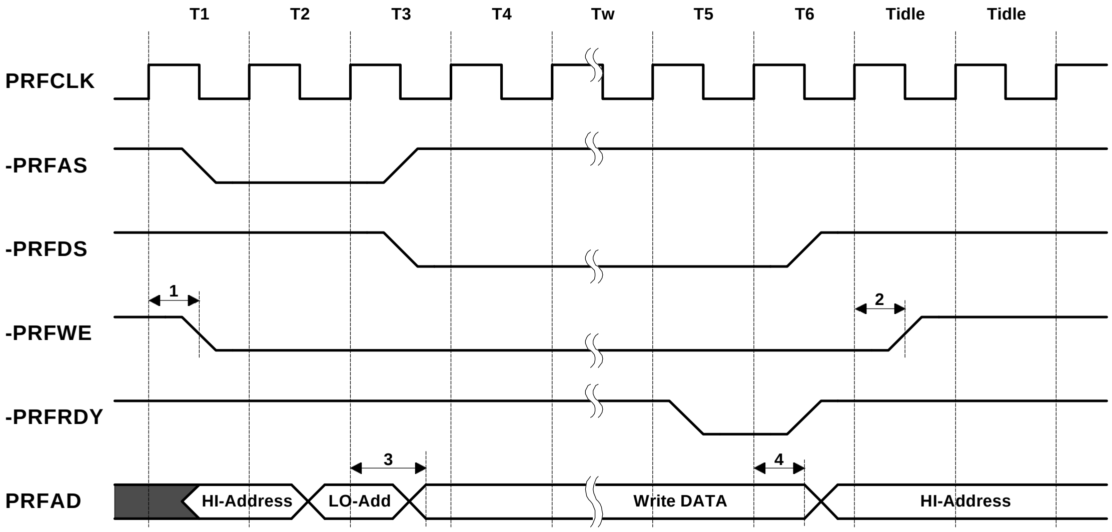


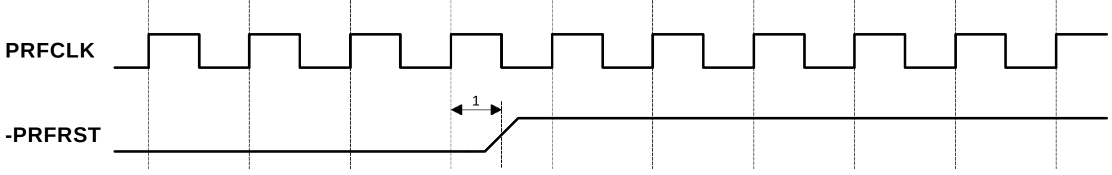
Figure 10.-11 - Peripheral I/F write timing

10.5.3 Peripheral I/F reset timing

Nov, 1996

No.	Parameter	Min	Typ.	Max	Unit
1	tRSD -PRFRST deassertion delay from PRFCLK↑	-		18.0	ns

NOTE: All timing requirements are preliminary and subject to change upon production release.

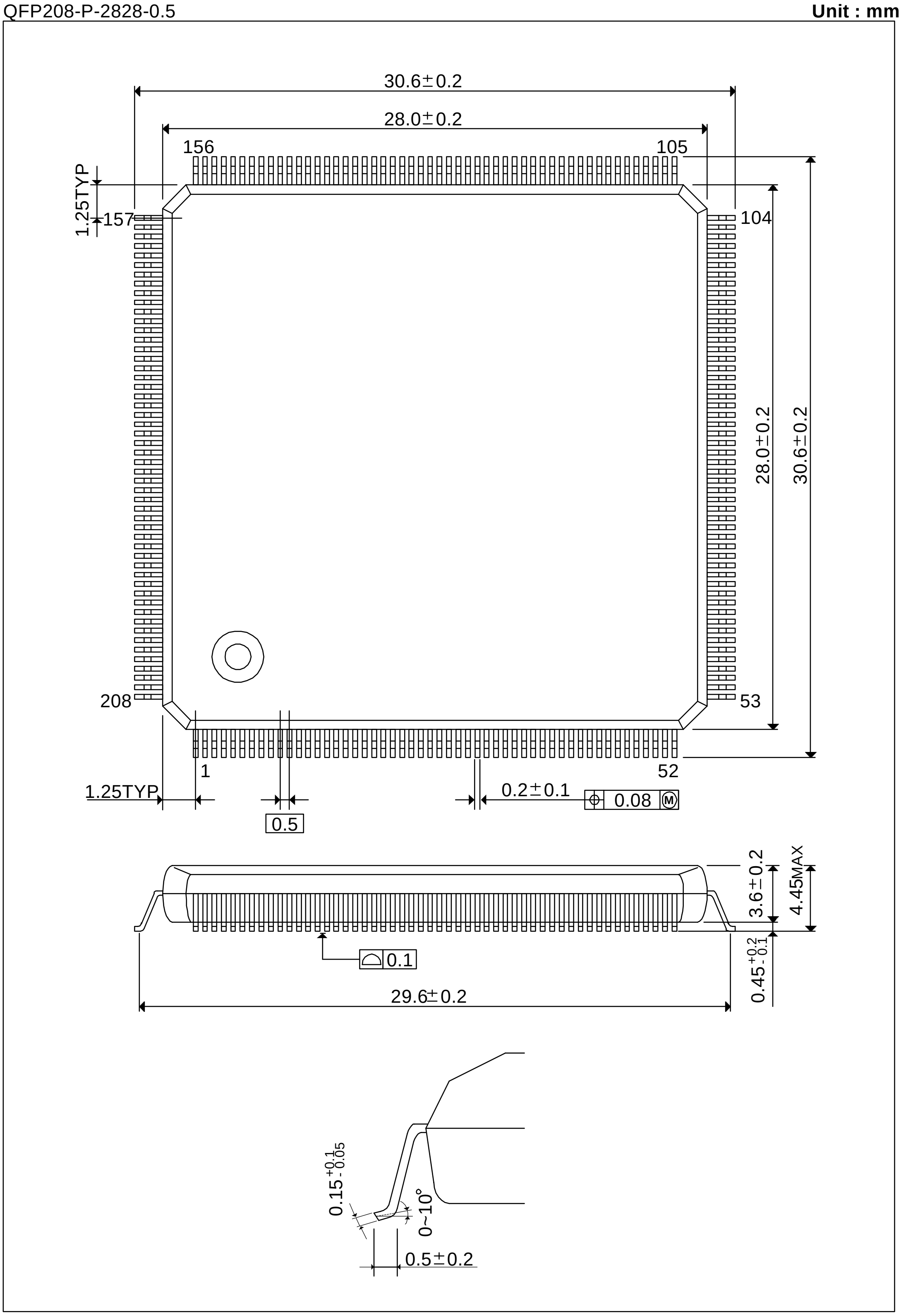


NOTE: The assertion of -PRFRST is asynchronous to PRFCLK, while the deassertion is synchronous to PRFCLK.
-PRFRST is asserted whenever -RST is asserted.

Figure 10.-12 - Peripheral I/F reset timing

11. Package outline drawing

QFP208-P-2828-0.5 **Unit : mm**



12. REFLOW

For epoxy resin molded packages, the material has a storing tendency to absorb moisture. If Surface Mount Device suffers thermal stress of solder reflow under the wet device condition, the reliability of device will be deteriorate remarkably. So dry packing is performed to TC35856BF. TC35856BF has some restraint for solder reflow.

12.1 Method of Solder Reflow

This device is permitted only Far Infrared Reflow. In the case of Medium / Near Infrared Reflow and V.P.S, the reliability of device will be deteriorate remarkably.

12.2 Condition of Solder Reflow

12.2.1 Method of Heat

We recommended Far Infrared Reflow with top/bottom heating.
The soldering should be performed within 30 seconds at more than 210^{o}C and 240^{o}C maximum.

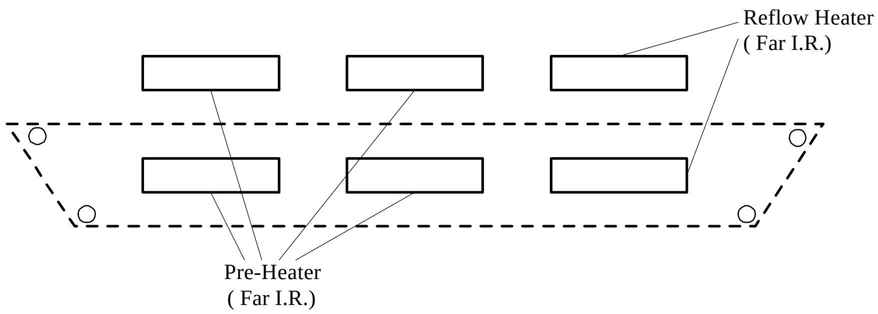


Figure 12.-1 - Method of Heat

12.2.2 Recommended Temperature Profile

The soldering should be performed with Recommended temperature profile. Recommended temperature profile is shown in Figure 12.-2.

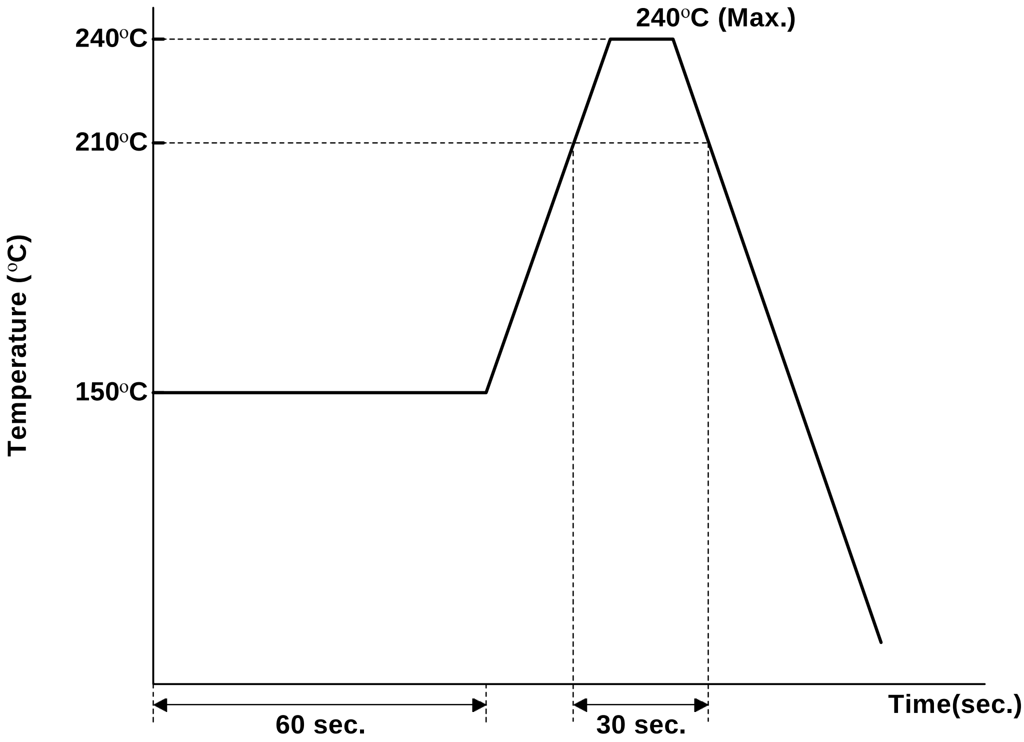


Figure 12.-2 - Recommended Temperature Profile for I.R. Soldering

12.3 Dry Package

Dry packing is applied to TC35856BF. After the customer opens the bag, the parts must be mounted or soldered within 24 hours under normal storage RH = 60%, Ta = 30°C.
If all the parts can not be soldered within the recommended period after opening the bag, pre-baking must be performed. The recommended pre-baking conditions are as follows:

Head proof tray type : 125°C/20hours baking with the original heat proof pack trays

Appendix A - Host Memory Logical Structures

This appendix contains a listing of the logical structures used by TC35856BF which are located in host memory and of their sizes in bytes.

Tx Report Queue - 4KB (256 or 1K VC mode), 16KB (4K VC mode):

- Stored in contiguous physical host memory.
- 64-Byte aligned.
- 1K entries (256 or 1K VC mode) or 4K entries (4K VC mode).
- Each entry is 4 bytes and contains:
  - 12-bit VC
  - 1-bit own
  - 19-bits reserved

Rx Report Queue - 24KB (256 or 1K VC mode), 96KB (4K VC mode)::

- Stored in contiguous physical host memory.
- 64-Byte aligned.
- 3K entries (256 or 1K VC mode) or 12K entries (4K VC mode).
- Sized to match the largest TOTAL number of entries in the 3 rx Free Slot FIFOs (3K/12K).
- Each entry is 8 bytes and contains:
  - 14-bit slot tag (may map to virtual address of the receive slot)
  - 1-bit Raw
  - 1-bit Own
  - 12-bit VC
  - 12-bit Size (in LW, valid for EOP AAL5 slots only)
  - 1-bit EFCI (valid for EOP AAL5 slots only)
  - 1-bit CLP (valid for EOP AAL5 slots only)
  - 3-bit Status (valid for EOP AAL5 slots only)
  - 1-bit Bad (valid for EOP AAL5 slots only)
  - 1-bit SOP (valid for AAL5 slots only)
  - 1-bit EOP (valid for AAL5 slots only)

RX Slot_Tag Table - 6KB in 256-VC mode, 12KB in 1K-VC mode, 48KB in 4K-VC mode (for a 32-bit machine; memory requirements are doubled for 64-bit architectures)

- Provides mapping of rx slot_tags to virtual addresses.
- 1.5K entries in 256-VC mode, 3K entries in 1K-VC mode, or 12K entries in 4K-VC mode.
- Sized to match the TOTAL number of entries in the 3 rx Free Slot FIFOs.
- Each entry consists of a single virtual address.

Appendix B - ACR lookup table

This appendix describes ACR lookup table generation. ACR lookup table is used for conversion of cell rate representation defined in UNI4.0 spec to cell interval time. It has 512 entries with 10 bit accuracy. The cell rate defines the number of cells per second. As the cell interval is measured by cell_times, ACR lookup table depends on line rate.

Following program is an example to generate ACR lookup table. After compiling the code, and assuming executable file name is “acrgen”, you shall execute;

acrgen 2.83119 0x64b1 (for 155Mbps line rate)

or

acrgen 16.5625 0x5faf (for 25.6Mbps line rate)

The program displays the difference between ideal cell rate and actual cell rate using ACR lookup table. The table itself is stored in “cellint.tbl” file.

```
<< ACR lookup table generation program : acrgen.c >>
/* $Id: acrgen.c,v 1.4 1997/03/03 02:04:36 tanaka Exp $ */
/*********************************/
/***  TC35856BF ACR lookup table**/
/***          Generator           **/
/***                              **/
/***          Output file :       **/
/***                cellint.tbl   **/
/***                              **/
/***           (C) Toshiba        **/
/*********************************/
#include     <stdio.h>
#include     <math.h>

FILE         *fdw;
double       line, unit;
float        cell_int;

main( argc, argv)
int  argc;
char         *argv[];
{
    long     bx;
    double   a, b, c;

    double   ideal_rate, actual_rate, diff, error;
    int      e;        /* exponent */
    int      m;        /* mantissa */

    int      min_m, max_m;
    double   min_r=1000.0, max_r=-1000.0;
```

```
if(argc<2){
        printf("Usage : acr cell_interval_timee(in us)\n");
        printf("cell interval time ;\n");
        printf("155.52Mbps : 2.83119 us, 25.6Mbps : 16.5625 us\n");
        exit(1);
}
sscanf(argv[1], "%f", & cell_int);

if((fdw=fopen("cellint.tbl", "w")) ==
        fprintf(stderr, "file (cellint.tbl) can not create\n");
        exit(1);
}

line = (1000000.0 / cell_int);
unit = cell_int / 1000000.0;

e = (int) log2(line);
m = (((int)line) >> (e-9));          /* mantissa of max line rate */

printf("Line rate (1 + 0x%x/0x200)*2^%d = %d cell/sec\n",
        m-0x200, e, (int)line);
fprintf(fdw, "//*** PCR = %04x ***//\n", 0x4000+(e<<9)+(m & 0x1FF));
fprintf(fdw, "//*** Line rate ( 1 + 0x%x/0x200)*2^%d = %d cell/sec ***//\n",
        m-0x200, e, (int)line);
printf("  m  lookup   ideal    actual    diff (error ratio %%)\n");
printf("     table     rate     rate\n");
printf("-------------------------------------------------------------\n");

for(m=0; m<512;
        ideal_rate = (double)((0x200+m) << (e-9));

        /* B is an ideal cell interval in cell time. */
        b = line / ideal_rate;

        /* lookup table is 10 bits */
        bx = (int)(b * 512.0);

      /* Bx adjusting in order not to exceed ideal ACR */
        bx--;
      do{
                bx++;
                /* cell interval in second */
                c = ((double)bx / 512.0) * unit;
                actual_rate = 1.0 / c;

                diff = actual_rate - ideal_rate;
        } while(diff > 0.0);

        /* Write into ACR lookup table */
```

```
		fprintf(fdw, "%04x\n", bx);

		error = (diff/ideal_rate) * 100.0;

		if(error>max_r){
			max_r = error;
			max_m = m;
		}
		if(error<min_r){
			min_r = error;
			min_m = m;
		}

	printf("%3d  %04x  %10.3lf  %10.3lf  %10.3lg ( %10.3lg %%)\n",
		 m, bx, ideal_rate, actual_rate, diff, error);
	}
	printf("Max error=%10.3g%%(m= %d), Min error=%10.3g%%(m=%d)\n",
		max_r, max_m, min_r, min_m);
	fprintf(stderr, "ACR lookup table is stored in \"cellint.tbl\"\n");
}
```

Appendix C - Restrictions of TC35856BF

Currently there is no restriction

Appendix D - Revision

This section shows the difference of technical data sheet from revision 2.0 to 2.3.

Rev.	No.	Section	Comment
1.0 Sept. 17, 1997	1	N.A.	The first edition for TC35856BF
1.1 Oct. 8, 1997	1	N.A.	Editorial change