PM7385

FREEDM™-84A672

FRAME ENGINE AND DATALINK MANAGER 84A672

PROGRAMMER'S GUIDE

PROPRIETARY AND CONFIDENTIAL

ADVANCE

ISSUE 1: JUNE 1999

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 Scope

The FREEDM-84A672 Programmer's Guide describes the configurable features and operation of a FREEDM-84A672 from a programmer's perspective. This document may not cover all applications of the FREEDM-84A672. Please contact a PMC-Sierra Applications Engineer for specific uses not covered in this document.

This document is a supplement to the FREEDM-84A672 Longform Datasheet[1]. Both documents should be studied together to interface the FREEDM-84A672 to an embedded processor. In case of a discrepancy between the Programmer's Guide and the Longform Datasheet, the Longform Datasheet shall always be considered correct.

1.2 Target Audience

The FREEDM-84A672 Programmer's Guide describes the configuration and initialisation necessary for programming the FREEDM-84A672 from a programmer's perspective. This document has been prepared for readers with prior knowledge of the HDLC protocol.

1.3 Numbering Conventions

The following numbering conventions are used throughout this document:

binary	011 1010B, 011
decimal	129, 6, 12
hexadecimal	0x1FE2, 09FH

1.4 Register Description

Unless specified, FREEDM-84A672 registers are described using the convention **REGISTER_NAME** (address in FREEDM-84A672). There is only one register space that can be addressed on a FREEDM-84A672, and it consists of the normal mode microprocessor accessible registers.

1.4.1 Normal Mode Registers

Normal mode registers are used to configure, monitor and control the operation of the FREEDM-84A672. Registers must be accessed as 16-bit values with a dword aligned address. For all register descriptions, the hexademical register number indicates the address in the FREEDM-84A672 when accesses are made using the external microprocessor. A register value is accessed through an external microprocessor and has the following characteristics:

- Writing values into unused register bits has no effect. However, to ensure software compatibility with future, feature-enhanced versions of the product, unused register bits should be written with logic zero. Reading back unused bits can produce either a logic one or a logic zero; hence, unused register bits should be masked off by software during a register read access.

- Except where noted, all configuration bits that can be written into can also be read back. This allows the processor controlling the FREEDM-84A672 to determine the programming state of the block.

- Writable normal mode registers are cleared to logic zero upon reset unless otherwise noted.

- Writing into read-only normal mode register bit locations does not affect FREEDM-84A672 operation unless otherwise noted.

- Certain register bits are reserved. These bits are associated with megacell functions that are unused in this application. To ensure that the FREEDM-84A672 operates as intended, reserved register bits must only be written with their default values. Similarly, writing to reserved registers should be avoided.

2 REFERENCES

1. PMC-990114, PMC-Sierra, Inc., "Frame Engine and Data Link Manager 84A672" Longform Datasheet, January 1999, Issue 1.

2. PMC-960758, PMC-Sierra, Inc., "Frame Engine and Data Link Manager" Longform Datasheet, May 1998, Issue 5.

3. PMC-980577, PMC-Sierra, Inc., "Saturn Compatible Scaleable Bandwidth Interconnect (SBI) Specification", October 1998, Issue 3.

4. PMC-990263, PMC-Sierra, Inc., "Frame Engine and Data Link Manager 32A672" Longform Datasheet, May 1999, Issue 2.

3 FREEDM-84A672 OVERVIEW

3.1 FREEDM-84A672 Summary

The PM7384 FREEDM-84A672 Frame Engine and Datalink Manager is an advanced data link layer processor that is ideal for applications such as IETF PPP interfaces for routers, Frame Relay switches and multiplexors, ATM switches and multiplexors, Internet/Intranet access equipment, packet-based DSLAM equipment, Packet over SONET, and PPP over SONET. The FREEDM-84A672 implements HDLC processing for a maximum of 672 bi-directional channels. The functional blocks of the FREEDM-84A672 are illustrated in Figure 1.

The FREEDM-84A672 may be configured to support channelised T1/J1/E1 or unchannelised DS-3 traffic on up to 84 links conveyed via a Scaleable Bandwidth Interconnect (SBI) interface. The SBI interface transports data in three Synchronous Payload Envelopes (SPEs), each of which may be configured independently to carry either 28 T1/J1 links, 21 E1 links or a single DS-3 link.

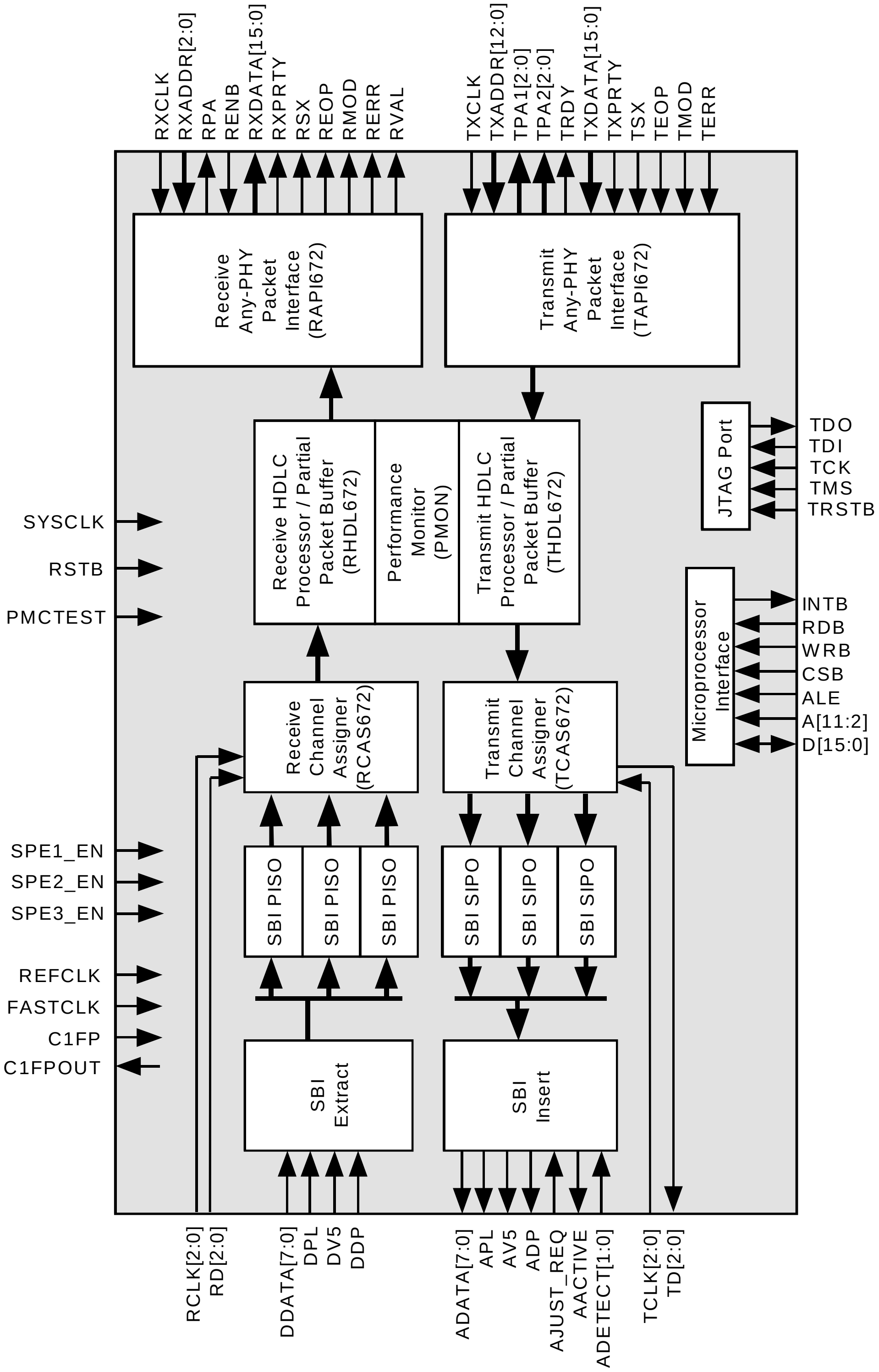
For channelised T1/J1/E1 links, the FREEDM-84A672 allows up to 672 bi-directional HDLC channels to be assigned to individual time-slots within each independently timed T1/J1 or E1 link. These links are processed by the Receive Channel Assigner (RCAS672) and the Transmit Channel Assigner (TCAS672). The channel assignment supports the concatenation of time-slots (N x DS0) up to a maximum of 24 concatenated time-slots for a T1/J1 link and 31 concatenated time-slots for an E1 link. Time-slots assigned to any particular channel need not be contiguous within a T1/J1 or E1 link. Unchannelised DS-3 links are assigned to a single HDLC channel.

Additionally, links may be configured independently to operate in an unframed or "clear channel" mode, in which the bit periods which are normally reserved for framing information in fact carry HDLC data. In unframed mode, links operate as unchannelised (i.e. the entire link is assigned to a single HDLC channel) regardless of link rate.

The FREEDM-84A672 supports mixing of channelised T1/J1/E1 and unchannelised or unframed links. The total number of channels in each direction is limited to 672. The maximum possible data rate over all links is 134.208 Mbps (which occurs with three DS-3 links running in unframed mode).

The FREEDM-84A672 supports three independently timed bidirectional clock/ data links, each carrying a single unchannelised HDLC stream. The links can be of arbitrary frame format and can operate at up to 52 MHz provided SYSCLK is running at 40 MHz. When activated, each link replaces one of the SPEs

Figure 1 – FREEDM-84A672 Block Diagram



conveyed on the SBI interface. (The maximum possible data rate when all three clock/data links are activated is 156 Mbps.)

Each data stream can be HDLC processed on a channelised basis within the Receive HDLC Processor / Partial Packet Buffer (RHDL672) and Transmit HDLC Processor / Partial Packet Buffer (THDL672). There is a 32 Kbyte buffer in the RHDL672 and another 32 Kbyte buffer in the THDL672 that must be assigned to FREEDM-84A672 channels to serve as channel FIFOs. Each buffer is a group of 2048 blocks with 16 bytes per block, and a minimum of 3 blocks must be assigned to a channel during provisioning. This allows for flexible assignment of a channel FIFO based on the expected data rate for the channel.

Alternatively, the RHDL672 and THDL672 can provision a channel as transparent, in which case, the raw data stream is transferred without HDLC processing.

The Receive Any-PHY Interface (RAPI672) provides a low latency path for transferring data out of the partial packet buffer in the RHDL672 and onto the Receive Any-PHY Packet Interface (Rx APPI). The RAPI672 contains a FIFO block for latency control as well as to segregate the APPI timing domain from the SYSCLK timing domain. The RAPI672 contains the necessary logic to manage and respond to device polling from an upper layer device. The RAPI672 also provides the upper layer device with status information on a per packet basis.

The Transmit Any-PHY Interface (TAPI672) provides a low latency path for transferring data from the Transmit Any-PHY Packet Interface (Tx APPI) into the partial packet buffer in the THDL672. The TAPI672 contains a FIFO block for latency control as well as to segregate the APPI timing domain from the SYSCLK timing domain. The TAPI672 contains the necessary logic to manage and respond to channel polling from an upper layer device.

The PMON block provides performance monitor counts for a number of events. These counters can be read via the microprocessor interface and provides a means for the host software to accumulate performance statistics.

Links can be individually placed in line loopback. There is also an internal diagnostic loopback configuration for each channel which can be used to diagnose FREEDM-84A672 operation on a per channel basis.

3.2 Any-PHY Packet Interface

The FREEDM-84A672 provides a low latency "Any-PHY" Packet Interface (APPI) to allow an external controller direct access into the 32 Kbyte partial packet buffers. Up to seven FREEDM-84A672 devices may share a single APPI.

For each of the transmit and receive APPI, the external controller is the master of each FREEDM-84A672 device sharing the APPI from the point of view of device selection. The external controller is also the master for channel selection in the transmit direction. In the receive direction, however, each FREEDM-84A672 device retains control over selection of its respective channels. The transmit and receive APPI is made up of three groups of functional signals – polling, selection and data transfer. The polling signals are used by the external controller to interrogate the status of the transmit and receive 32 Kbyte partial packet buffers. The selection signals are used by the external controller to select a FREEDM-84A672 device, or a channel within a FREEDM-84A672 device, for data transfer. The data transfer signals provide a means of transferring data across the APPI between the external controller and a FREEDM-84A672 device.

In the receive direction, polling and selection are done at the device level. Polling is not decoupled from selection, as the receive address pins serve as both a device poll address and to select a FREEDM-84A672 device. In response to a positive poll, the external controller may select that FREEDM-84A672 device for data transfer. Once selected, the FREEDM-84A672 prepends an in-band channel address to each partial packet transfer across the receive APPI to associate the data with a channel. A FREEDM-84A672 must not be selected after a negative poll response.

In the transmit direction, polling is done at the channel level. Polling is completely decoupled from selection. To increase the polling bandwidth, up to two channels may be polled simultaneously. The polling engine in the external controller runs independently of other activity on the transmit APPI. In response to a positive poll, the external controller may commence partial packet data transfer across the transmit APPI for the successfully polled channel of a FREEDM-84A672 device. The external controller must prepend an in-band channel address to each partial packet transfer across the transmit APPI to associate the data with a channel.

Detailed information on configuring the RAPI672 and the TAPI672 can be found in section 8 of this document.

4 INTERRUPT ARCHITECTURE

This section provides an overview of the FREEDM-84A672 interrupt architecture. Detailed information on the individual interrupts is available in the Longform Datasheet[1].

4.1 Non-SBI Interrupts

The FREEDM-84A672 provides a number of individual interrupts which are identified as 'I' bits within the **FREEDM-84A672 Master Interrupt Status** (0x008) register. When an interrupt source becomes active, the 'I' bit is set and remains set until the **FREEDM-84A672 Master Interrupt Status** (0x008) register is read.

The FREEDM-84A672 provides interrupts to the microprocessor bus via the INTB pin of the FREEDM-84A672. The INTB pin is gated by the **FREEDM-84A672 Master Interrupt Enable** (0x004) register. This register contains 'E' bits which can mask the 'I' bit from causing an interrupt on the INTB pin of the FREEDM-84A672. When the 'E' and 'I' bits of an interrupt source are both high, then the INTB pin is active. When the 'E' bit is low, the interrupt source will not activate the INTB pin regardless of the 'I' bit status. However, the 'I' bit remains valid when interrupts are disabled and may be polled to detect the various events.

The complete list of 'I' bits and 'E' bits for non-SBI interrupts is shown below:

'E' Bit	'I' Bit	Description
RFCSEE	RFCSEI	Receive FCS Error
RABRTE	RABRTI	Receive Abort
RPFEE	RPFEI	Receive Packet Format Error
RFOVRE	RFOVRI	Receive FIFO Overrun Error
TPRTYE	TPRTYI	Transmit Parity Error
TUNPVE	TUNPVI	Transmit Unprovisioned Error
TFOVRE	TFOVRI	Transmit FIFO Overflow Error
TFUDRE	TFUDRI	Transmit FIFO Underflow Error

4.2 SBI Interrupts

In addition to the interrupts described in section 4.1, interrupts can be provided to the microprocessor by the SBI Extract block of the FREEDM-84A672.

The SBI Extracter interrupt status bit (SBIEXTI) of the **FREEDM-84A672 Master SBI Interrupt Status** (0x02C) register reports an error condition from the SBI

Extract block to the microprocessor. Reading this register acknowledges and clears the interrupt.

The SBI Extracter interrupt enable bit (SBIEXTE) of the **FREEDM-84A672 Master SBI Interrupt Enable** (0x028) register can mask the SBIEXTI bit from causing an interrupt on the INTB pin. When the SBIEXTE and SBIEXTI pins are both high, then the INTB pin is active. When the SBIEXTE is low, the interrupt source will not activate the INTB pin, regardless of the SBIEXTI status. However, SBIEXTI remains valid when interrupts are disabled and may be polled to detect SBI Extract block error conditions.

SBI Extract Parity Error Interrupt

In the FREEDM-84A672, the only error condition which the SBI Extract block reports is a parity error on the SBI DROP BUS.

The PERRI bit of the **SBI EXTRACT Parity Error Interrupt Reason** (0x5DC) register indicates that an SBI parity error has been detected. Reading this register clears this bit. The TRIB[4:0] and SPE[1:0] fields of this register specify the SBI tributary for which a parity error was detected, and are only valid when PERRI is set.

The SBI_PERR_EN bit of the **SBI EXTRACT Control** (0x5C0) register enables or disables SBI Parity Error Interrupts. When SBI_PERR_EN is low, SBI Parity Error Interrupts are disabled. When SBI_PERR_EN is high, SBI Parity Error Interrupts are enabled. In both cases, the **SBI EXTRACT Parity Error Interrupt Reason** (0x5DC) register is updated when a parity error occurs.

Note: Even if SBI_PERR_EN and PERRI are both high (causing SBIEXTI to report an error condition), SBIEXTE must also be high for the SBI Extract block to cause an interrupt on the INTB pin.

5 CONFIGURING THE SBI INTERFACE

The Scaleable Bandwidth Interconnect (SBI) is a synchronous, time-division multiplexed bus designed to transfer, in a pin-efficient manner, data belonging to a number of independently timed links of varying bandwidth. The bus is timed to a reference 19.44MHz clock and a 2kHz or 166.7Hz frame pulse. All sources and sinks of data on the bus are timed to the reference clock and frame pulse.

The SBI multiplexing structure is modeled on the SONET/SDH standards. The SONET/SDH virtual tributary structure is used to carry T1/J1 and E1 links. Unchannelised DS3 payloads follow a byte synchronous structure modeled on the SONET/SDH format.

The multiplexed links are separated into three Synchronous Payload Envelopes (SPEs). Each envelope may be configured independently to carry up to 28 T1/J1s, 21 E1s or a DS3.

Full details of the operation of the SBI interface are provided in the SBI Compatibility Specification [3].

5.1 Configuring the SBI DROP BUS

The SBI DROP BUS is a byte wide serial bus which drops SBI tributaries from multiple PHY devices to multiple link layer devices such as the FREEDM-84A672.

The SBI DROP BUS is configured by programming bits within the **FREEDM-84A672 SBI DROP BUS Master Configuration** (0x048) register. The default configuration is as follows:

Bit	Register	Value
SPE1_TYP[1:0]	**FREEDM-84A672 SBI DROP BUS Master Configuration** (0x048)	00
SPE2_TYP[1:0]	**FREEDM-84A672 SBI DROP BUS Master Configuration** (0x048)	00
SPE3_TYP[1:0]	**FREEDM-84A672 SBI DROP BUS Master Configuration** (0x048)	00
FCLK_FREQ[1:0]	**FREEDM-84A672 SBI DROP BUS Master Configuration** (0x048)	00
Reserved[1:0]	**FREEDM-84A672 SBI DROP BUS Master Configuration** (0x048)	00

The default indicates that all three Synchronous Payload Envelopes conveyed on the SBI DROP BUS are configured for 28 T1/J1 links and the FASTCLK input operates at a frequency of 51.84 MHz.

SPE Type on the SBI DROP BUS

The SPE type bits (SPEn_TYP[1:0]) determine the configuration of each of the three Synchronous Payload Envelopes conveyed on the SBI DROP BUS, according to the following table.

SPEn_TYP[1:0]	Link Configuration
00	28 T1/J1 links
01	21 E1 links
10	Single DS-3 link
11	Reserved

FASTCLK Frequency

The high-speed reference clock signal (FASTCLK) is used by the FREEDM-84A672 to generate an internal clock for use when processing DS-3 links. The FASTCLK frequency selector bits (FCLK_FREQ[1:0]) must be set according to the following table, depending on the frequency chosen for the FASTCLK input.

FCLK_FREQ[1:0]	FASTCLK Frequency
00	51.84 MHz
01	44.928 MHz
10	Reserved
11	66 MHz

5.2 Configuring the SBI ADD BUS

The SBI ADD BUS is a byte wide serial bus which aggregates TDM tributaries from multiple link layer devices such as the FREEDM-84A672 to multiple PHY devices.

The SBI ADD BUS is configured by programming bits within the **FREEDM-84A672 SBI ADD BUS Master Configuration** (0x04C) register. The default configuration is as follows:

Bit	Register	Value
SPE1_TYP[1:0]	**FREEDM-84A672 SBI ADD BUS Master Configuration** (0x04C)	00
SPE2_TYP[1:0]	**FREEDM-84A672 SBI ADD BUS Master Configuration** (0x04C)	00
SPE3_TYP[1:0]	**FREEDM-84A672 SBI ADD BUS Master Configuration** (0x04C)	00
FCLK_FREQ[1:0]	**FREEDM-84A672 SBI ADD BUS Master Configuration** (0x04C)	00
Reserved[4:0]	**FREEDM-84A672 SBI ADD BUS Master Configuration** (0x04C)	0x00
DEFAULT_DRV	**FREEDM-84A672 SBI ADD BUS Master Configuration** (0x04C)	0

The default indicates that all three Synchronous Payload Envelopes conveyed on the SBI ADD BUS are configured for 28 T1/J1 links, the FASTCLK input operates at a frequency of 51.84 MHz, and the FREEDM-84A672 will only drive the bus when it has data to send.

SPE Type on the SBI ADD BUS

The SPE type bits (SPEn_TYP[1:0]) determine the configuration of each of the three Synchronous Payload Envelopes conveyed on the SBI ADD BUS, according to the following table.

SPEn_TYP[1:0]	Link Configuration
00	28 T1/J1 links
01	21 E1 links
10	Single DS-3 link
11	Reserved

FASTCLK Frequency

The high-speed reference clock signal (FASTCLK) is used by the FREEDM-84A672 to generate an internal clock for use when processing DS-3 links. The FASTCLK frequency selector bits (FCLK_FREQ[1:0]) must be set according to the following table, depending on the frequency chosen for the FASTCLK input.

FCLK_FREQ[1:0]	FASTCLK Frequency
00	51.84 MHz
01	44.928 MHz
10	Reserved
11	66 MHz

Default Bus Driver

The Default Bus Driver selector bit (DEFAULT_DRV) enables the FREEDM-84A672 device to drive the SBI ADD BUS when no other device is doing so. It is recommended that one device connected to an SBI Bus be nominated as a default driver and configured to drive the bus when no other device is doing so (when the ADETECT[1:0] inputs are both 0). This feature is configured as follows:

DEFAULT_DRV	Function
0	The FREEDM-84A672 will only drive the bus when it has data to send (and when ADETECT[1:0] are both 0).
1	The FREEDM-84A672 will drive the bus whenever the ADETECT[1:0] inputs are both 0.

6 CONFIGURING THE SBI EXTRACTER AND INSERTER

6.1 Configuring the SBI Extracter

The SBI receive circuitry consists of an SBI Extract block and three SBI Parallel to Serial Converter (SBI PISO) blocks. The SBI Extract block receives data from the SBI DROP BUS and converts it to an internal parallel bus format. The received data is then converted to serial bit streams by the PISO blocks. Each PISO block processes one of the three Synchronous Payload Envelopes (SPEs) conveyed on the SBI DROP BUS.

The SBI Extract block may be configured to enable or disable reception of individual tributaries within the SBI DROP bus. Individual tributaries may also be configured to operate in framed or unframed mode.

Each PISO block inputs data related to one SPE from the internal parallel bus and generates either 28 serial data streams at T1/J1 rate, 21 streams at E1 rate or a single stream at DS-3 rate. These serial streams are then processed by the Receive Channel Assigner block.

6.1.1 SBI EXTRACT Control

The SBI Extract block is controlled by programming bits within the **SBI EXTRACT Control** (0x5C0) register. The default configuration is as follows:

Bit	Register	Value
SBI_PAR_CTL	**SBI EXTRACT Control** (0x5C0)	1
SBI_PERR_EN	**SBI EXTRACT Control** (0x5C0)	0
Reserved[2:0]	**SBI EXTRACT Control** (0x5C0)	000
Reserved[3]	**SBI EXTRACT Control** (0x5C0)	0

The default indicates that odd parity mode is used for checking the SBI parity signal, and that the SBI Parity Error interrupts are disabled.

SBI Parity Mode

The SBI_PAR_CTL bit is used to configure the Parity mode for checking of the SBI parity signal, DDP as follows:

SBI_PAR_CTL	Function
0	Even parity checking.
1	Odd parity checking.

SBI Parity Error Interrupt Enable

The SBI_PERR_EN bit is used to enable SBI parity error interrupt generation and is decoded in the following table. Please see section 4.2 for more information on the SBI parity error interrupt.

SBI_PERR_EN	Function
0	SBI parity error interrupts are disabled.
1	SBI parity error interrupts are enabled.

6.1.2 SBI EXTRACT Tributary Configuration

SBI EXTRACT tributary configuration information is read from and written to the SBI EXTRACT tributary control configuration RAM. An SBI tributary in the receive direction is configured using the following procedure:

1. Poll the BUSY bit of the **SBI EXTRACT Tributary RAM Indirect Access Control** (0x5D0) register until it is zero. This ensures that a previous indirect RAM access has completed and a new indirect RAM access can be started.

2. The TRIB[4:0] and SPE[1:0] fields of the **SBI EXTRACT Tributary RAM Indirect Access Address** (0x5CC) register are used to specify which SBI tributary the control configuration RAM write or read operation will apply to. Legal values for TRIB[4:0] are b'00001' through b'11100'. Legal values for SPE[1:0] are b'01' through b'11'. Write this register as follows:

Bit	Register	Value
TRIB[4:0]	**SBI EXTRACT Tributary RAM Indirect Access Address** (0x5CC)	See above
SPE[1:0]	**SBI EXTRACT Tributary RAM Indirect Access Address** (0x5CC)	See above
Reserved	**SBI EXTRACT Tributary RAM Indirect Access Address** (0x5CC)	0

3. The ENBL bit of the **SBI EXTRACT Tributary RAM Indirect Access Data** (0x5D8) register is used to enable the tributary. Writing to the tributary control configuration RAM with the ENBL bit set enables the SBI EXTRACT block to take tributary data from an SBI tributary and output that data to the SBI PISO blocks.

   The TRIB_TYP[1:0] field of the **SBI EXTRACT Tributary RAM Indirect Access Data** (0x5D8) register is used to configure the tributary to operate in framed or unframed mode as follows:

TRIB_TYP[1:0]	Tributary type
00	Reserved
01	Framed
10	Unframed
11	Reserved

   Specify the configuration data to be written to the tributary control configuration RAM by writing the following register:

Bit	Register	Value
ENBL	**SBI EXTRACT Tributary RAM Indirect Access Data** (0x5D8)	See above
Reserved[0]	**SBI EXTRACT Tributary RAM Indirect Access Data** (0x5D8)	0
TRIB_TYP[1:0]	**SBI EXTRACT Tributary RAM Indirect Access Data** (0x5D8)	See above
Reserved[3:1]	**SBI EXTRACT Tributary RAM Indirect Access Data** (0x5D8)	000

4. Trigger an indirect write operation on the tributary control configuration RAM by writing the following register:

Bit	Register	Value
Reserved	**SBI EXTRACT Tributary RAM Indirect Access Control** (0x5D0)	0
RWB	**SBI EXTRACT Tributary RAM Indirect Access Control** (0x5D0)	0
BUSY	**SBI EXTRACT Tributary RAM Indirect Access Control** (0x5D0)	X

6.2 Configuring the SBI Inserter

The SBI transmit circuitry consists of an SBI Insert block and three SBI Serial to Parallel Converter (SBI SIPO) blocks. Each SIPO block processes data for one of the three Synchronous Payload Envelopes (SPEs) conveyed on the SBI ADD BUS. It receives serial data on either 28 links running at T1/J1 rate, 21 links at E1 rate or a single link at DS-3 rate and converts it to an internal parallel bus format. The SBI Insert block receives data from the SIPO blocks in the internal format and transmits it on the SBI ADD BUS.

The SIPO blocks generate the serial clocks for the TCAS672 and thus are able to control the rate at which data is transmitted on to the SBI. The SBI Insert block can command the SIPO blocks to speed up or slow down these clocks in response to justification requests received on the SBI interface. This feature is controlled by the CLK_MSTR bit which is explained in section 6.2.2. The SBI Insert block also contains FIFO circuitry to compensate for short term variations in the rate at which data is output by the TCAS672 and the rate at which it is transmitted on the SBI ADD BUS.

The SBI Insert block may be configured to enable or disable transmission of individual tributaries on to the SBI ADD bus. Individual tributaries may also be configured to operate in framed or unframed mode.

6.2.1 SBI INSERT Control

The SBI Insert block is controlled by programming bits within the **SBI INSERT Control** (0x680) register. The default configuration is as follows:

Bit	Register	Value
SBI_PAR_CTL	**SBI INSERT Control** (0x680)	1
Reserved[2:0]	**SBI INSERT Control** (0x680)	000
Reserved[3]	**SBI INSERT Control** (0x680)	0

The default indicates that the odd parity mode is used for generating the SBI parity signal.

SBI Parity Mode

The SBI_PAR_CTL bit is used to configure the Parity mode for generation of the SBI parity signal, ADP as follows:

SBI_PAR_CTL	Function
0	Even parity generation.
1	Odd parity generation.

6.2.2 SBI INSERT Tributary Configuration

SBI INSERT tributary configuration information is read from and written to the SBI INSERT tributary control configuration RAM. An SBI tributary in the transmit direction is configured using the following procedure:

1. Poll the BUSY bit of the **SBI INSERT Tributary RAM Indirect Access Control** (0x690) register until it is zero. This ensures that a previous indirect RAM access has completed and a new indirect RAM access can be started.

2. The TRIB[4:0] and SPE[1:0] fields of the **SBI INSERT Tributary RAM Indirect Access Address** (0x68C) register are used to specify which SBI tributary the control configuration RAM write or read operation will apply to. Legal values for TRIB[4:0] are b'00001' through b'11100'. Legal values for SPE[1:0] are b'01' through b'11'. Write this register as follows:

Bit	Register	Value
TRIB[4:0]	**SBI INSERT Tributary RAM Indirect Access Address** (0x68C)	See above
SPE[1:0]	**SBI INSERT Tributary RAM Indirect Access Address** (0x68C)	See above
Reserved	**SBI INSERT Tributary RAM Indirect Access Address** (0x68C)	0

3. The ENBL bit of the **SBI INSERT Tributary RAM Indirect Access Data** (0x698) register is used to enable the tributary. Writing to the tributary control configuration RAM with the ENBL bit set enables the SBI INSERT block to output tributary data on an SBI tributary.

   The TRIB_TYP[1:0] field of the **SBI INSERT Tributary RAM Indirect Access Data** (0x698) register is used to configure the tributary to operate in framed or unframed mode as follows:

TRIB_TYP[1:0]	Tributary type
00	Reserved
01	Framed
10	Unframed
11	Reserved

The CLK_MSTR bit of the **SBI INSERT Tributary RAM Indirect Access Data** (0x698) register configures the SBI tributary to operate as a timing master or slave. Setting CLK_MSTR to 1 configures the tributary as a timing master (AJUST_REQ input ignored). Setting CLK_MSTR to 0 configures the tributary as a timing slave (requests on AJUST_REQ honoured).

Specify the configuration data to be written to the tributary control configuration RAM by writing the following register:

Bit	Register	Value
ENBL	**SBI INSERT Tributary RAM Indirect Access Data** (0x698)	See above
Reserved	**SBI INSERT Tributary RAM Indirect Access Data** (0x698)	0
TRIB_TYP[1:0]	**SBI INSERT Tributary RAM Indirect Access Data** (0x698)	See above
CLK_MSTR	**SBI INSERT Tributary RAM Indirect Access Data** (0x698)	See above

4. Trigger an indirect write operation on the tributary control configuration RAM by writing the following register:

Bit	Register	Value
Reserved	**SBI INSERT Tributary RAM Indirect Access Control** (0x690)	0
RWB	**SBI INSERT Tributary RAM Indirect Access Control** (0x690)	0
BUSY	**SBI INSERT Tributary RAM Indirect Access Control** (0x690)	X

7 CONFIGURING THE SERIAL LINKS

Each of the 84 bi-directional links is controlled via the RCAS672 and the TCAS672 blocks of the FREEDM-84A672. The RCAS672 controls the receive data stream while the TCAS672 controls the transmit data stream.

The Receive Channel Assigner (RCAS672)

The Receive Channel Assigner block processes up to 84 serial links. When receiving data from the SBI PISO blocks, links may be configured to support channelised T1/J1/E1 traffic, unchannelised DS-3 traffic or unframed traffic at T1/J1, E1 or DS-3 rates. When receiving data from the RCLK/RD inputs, links 0, 1 and 2 support unchannelised data at arbitary rates up to 52 Mbps.

Each link is independent and has its own associated clock. For each link, the RCAS672 performs a serial to parallel conversion to form data bytes. The data bytes are multiplexed, in byte serial format, for delivery to the Receive HDLC Processor / Partial Packet Buffer block (RHDL672) at SYSCLK rate. In the event where multiple streams have accumulated a byte of data, multiplexing is performed on a fixed priority basis with link #0 having the highest priority and link #83 the lowest.

The 84 RCAS links have a fixed relationship to the SPE and tributary numbers on the SBI DROP BUS as shown in the following table.

SBI SPE No.	SBI Trib. No.	RCAS Link No.	SBI SPE No.	SBI Trib. No.	RCAS Link No.	SBI SPE No.	SBI Trib. No.	RCAS Link No.
1	1	0	2	1	1	3	1	2
1	2	3	2	2	4	3	2	5
1	3	6	2	3	7	3	3	8
1	4	9	2	4	10	3	4	11
1	5	12	2	5	13	3	5	14
1	6	15	2	6	16	3	6	17
1	7	18	2	7	19	3	7	20
1	8	21	2	8	22	3	8	23
1	9	24	2	9	25	3	9	26
1	10	27	2	10	28	3	10	29

SBI SPE No.	SBI Trib. No.	RCAS Link No.	SBI SPE No.	SBI Trib. No.	RCAS Link No.	SBI SPE No.	SBI Trib. No.	RCAS Link No.
1	11	30	2	11	31	3	11	32
1	12	33	2	12	34	3	12	35
1	13	36	2	13	37	3	13	38
1	14	39	2	14	40	3	14	41
1	15	42	2	15	43	3	15	44
1	16	45	2	16	46	3	16	47
1	17	48	2	17	49	3	17	50
1	18	51	2	18	52	3	18	53
1	19	54	2	19	55	3	19	56
1	20	57	2	20	58	3	20	59
1	21	60	2	21	61	3	21	62
1	22	63	2	22	64	3	22	65
1	23	66	2	23	67	3	23	68
1	24	69	2	24	70	3	24	71
1	25	72	2	25	73	3	25	74
1	26	75	2	26	76	3	26	77
1	27	78	2	27	79	3	27	80
1	28	81	2	28	82	3	28	83

Links containing a T1/J1 or an E1 stream may be channelised. Data at each time-slot may be independently assigned to a different channel. The RCAS672 performs a table lookup to associate the link and time-slot identity with a channel. The position of T1/J1 and E1 framing bits/bytes is identified by frame pulse signals generated by the SBI PISO blocks. Links containing a DS-3 stream are unchannelised, i.e. all data on the link belongs to one channel. The RCAS672 performs a table lookup using only the link number to determine the associated channel, as time-slots are non-existent in unchannelised links. Links may additionally be configured to operate in an unframed "clear channel" mode, in which all bit positions, including those normally reserved for framing information, are assumed to be carrying HDLC data. Links configured in unframed mode operate as unchannelised regardless of link rate and the

RCAS672 performs a table lookup using only the link number to determine the associated channel.

The Transmit Channel Assigner (TCAS672)

The Transmit Channel Assigner block processes up to 672 channels. Data for all channels is sourced from a single byte-serial stream from the Transmit HDLC Controller / Partial Packet Buffer block (THDL672). The TCAS672 demultiplexes the data and assigns each byte to any one of 84 links. When sending data to the SBI SIPO blocks, each link may be configured to support channelised T1/J1/E1 traffic, unchannelised DS-3 traffic or unframed traffic at T1/J1, E1 or DS-3 rates. When sending data to the TD outputs, links 0, 1 and 2 support unchannelised data at arbitary rates up to 52 Mbps. Each link is independent and has its own associated clock.

The 84 TCAS links have a fixed relationship to the SPE and tributary numbers on the SBI ADD BUS as shown in the following table.

SBI SPE No.	SBI Trib. No.	TCAS Link No.	SBI SPE No.	SBI Trib. No.	TCAS Link No.	SBI SPE No.	SBI Trib. No.	TCAS Link No.
1	1	0	2	1	1	3	1	2
1	2	3	2	2	4	3	2	5
1	3	6	2	3	7	3	3	8
1	4	9	2	4	10	3	4	11
1	5	12	2	5	13	3	5	14
1	6	15	2	6	16	3	6	17
1	7	18	2	7	19	3	7	20
1	8	21	2	8	22	3	8	23
1	9	24	2	9	25	3	9	26
1	10	27	2	10	28	3	10	29
1	11	30	2	11	31	3	11	32
1	12	33	2	12	34	3	12	35
1	13	36	2	13	37	3	13	38
1	14	39	2	14	40	3	14	41
1	15	42	2	15	43	3	15	44
1	16	45	2	16	46	3	16	47

SBI SPE No.	SBI Trib. No.	TCAS Link No.	SBI SPE No.	SBI Trib. No.	TCAS Link No.	SBI SPE No.	SBI Trib. No.	TCAS Link No.
1	17	48	2	17	49	3	17	50
1	18	51	2	18	52	3	18	53
1	19	54	2	19	55	3	19	56
1	20	57	2	20	58	3	20	59
1	21	60	2	21	61	3	21	62
1	22	63	2	22	64	3	22	65
1	23	66	2	23	67	3	23	68
1	24	69	2	24	70	3	24	71
1	25	72	2	25	73	3	25	74
1	26	75	2	26	76	3	26	77
1	27	78	2	27	79	3	27	80
1	28	81	2	28	82	3	28	83

As shown in the table above, TCAS links 0, 1, and 2 are mapped to tributary 1 of SPEs 1, 2 and 3 respectively. These links may be configured to operate at DS-3 rate. (They may also be configured to output data to the TD outputs at rates up to 52 Mbps.) For each of these high-speed links, the TCAS672 provides a six byte FIFO. For the remaining links (TCAS links 3 to 83, mapped to links 2 to 28 of each SPE), the TCAS672 provides a single byte holding register. The TCAS672 performs parallel to serial conversion to form bit-serial streams which are passed to the SBI SIPO blocks. In the event where multiple links are in need of data, TCAS672 requests data from upstream blocks on a fixed priority basis with link 0 having the highest priority and link 83 the lowest.

Links containing a T1/J1 or an E1 stream may be channelised. Data at each time-slot may be independently assigned to be sourced from a different channel. The position of T1/J1 and E1 framing bits/bytes is identified by frame pulse signals generated by the SBI SIPO blocks. With knowledge of the transmit link and time-slot identity, the TCAS672 performs a table look-up to identify the channel from which a data byte is to be sourced.

Links containing a DS-3 stream are unchannelised, in which case, all data bytes on the link belong to one channel. The TCAS672 performs a table look-up to identify the channel to which a data byte belongs using only the outgoing link identity, as no time-slots are associated with unchannelised links. Links may additionally be configured to operate in an unframed “clear channel” mode, in which case the FREEDM-84A672 will output HDLC data in all bit positions, including those normally reserved for framing information. Links configured in unframed mode operate as unchannelised regardless of link rate and the TCAS672 performs a table lookup using only the link number to determine the associated channel.

7.1 SBI SPE/Tributary Links

When the SPEn_EN input pin is high, the corresponding Synchronous Payload Envelope conveyed on the SBI interface is enabled and the corresponding independently timed link is disabled. This section describes the configuration of the operational and framing modes of those links mapped to SPEs on the SBI DROP and ADD buses.

SBI Mode for SPEn Links

The SBI mode select bits (SBI_MODE[2:0]) in the following registers configure the receive and transmit links of SPEn, where *1 ≤ n ≤ 3*:

Bit	SPE No.	Register
SBI_MODE[2:0]	1	**RCAS SBI SPE1 Configuration Register #1** (0x140)
SBI_MODE[2:0]	2	**RCAS SBI SPE2 Configuration Register #1** (0x148)
SBI_MODE[2:0]	3	**RCAS SBI SPE3 Configuration Register #1** (0x150)
SBI_MODE[2:0]	1	**TCAS SBI SPE1 Configuration Register #1** (0x440)
SBI_MODE[2:0]	2	**TCAS SBI SPE2 Configuration Register #1** (0x448)
SBI_MODE[2:0]	3	**TCAS SBI SPE3 Configuration Register #1** (0x450)

The encoding of the SBI_MODE[2:0] bits is shown in the following table, where *1 ≤ n ≤ 3*:

SBI_MODE [2:0]	SPEn Configuration
000	Single unchannelised DS-3 on link n–1
001	28 T1/J1 links
010	21 E1 links (links corresponding to SPEn tributaries 22-28 are unused)
011	Reserved
100	Reserved
101	Reserved
110	Reserved
111	Reserved

Framing Mode for SPEn Links

The framing mode of those links mapped to SPE 1 of the SBI DROP BUS is configured using the FEN[11:0] bits of the **RCAS SBI SPE1 Configuration Register #1** (0x140) and the FEN[27:12] bits of the **RCAS SBI SPE1 Configuration Register #2** (0x144). Each FEN bit, FEN[n], configures link 3n for framed operation. In unframed operation (FEN[n] = 0), all framing bit locations are treated as containing data. In framed mode (FEN[n] = 1), the contents of framing bit locations are ignored.

The framing mode of those links mapped to SPE 2 of the SBI DROP BUS is configured using the FEN[11:0] bits of the **RCAS SBI SPE2 Configuration Register #1** (0x148) and the FEN[27:12] bits of the **RCAS SBI SPE2 Configuration Register #2** (0x14C). Each FEN bit, FEN[n], configures link 3n+1 for framed operation. In unframed operation (FEN[n] = 0), all framing bit locations are treated as containing data. In framed mode (FEN[n] = 1), the contents of framing bit locations are ignored.

The framing mode of those links mapped to SPE 3 of the SBI DROP BUS is configured using the FEN[11:0] bits of the **RCAS SBI SPE3 Configuration Register #1** (0x150) and the FEN[27:12] bits of the **RCAS SBI SPE3 Configuration Register #2** (0x154). Each FEN bit, FEN[n], configures link 3n+2 for framed operation. In unframed operation (FEN[n] = 0), all framing bit locations are treated as containing data. In framed mode (FEN[n] = 1), the contents of framing bit locations are ignored.

The framing mode of those links mapped to SPE 1 of the SBI ADD BUS is configured using the FEN[11:0] bits of the **TCAS SBI SPE1 Configuration Register #1** (0x440) and the FEN[27:12] bits of the **TCAS SBI SPE1 Configuration Register #2** (0x444). Each FEN bit, FEN[n], configures link 3n for framed operation. In unframed operation (FEN[n] = 0), HDLC data is transmitted in all framing bit locations. In framed mode (FEN[n] = 1), the framing bit locations are unused.

The framing mode of those links mapped to SPE 2 of the SBI ADD BUS is configured using the FEN[11:0] bits of the **TCAS SBI SPE2 Configuration Register #1** (0x448) and the FEN[27:12] bits of the **TCAS SBI SPE2 Configuration Register #2** (0x44C). Each FEN bit, FEN[n], configures link 3n+1 for framed operation. In unframed operation (FEN[n] = 0), HDLC data is transmitted in all framing bit locations. In framed mode (FEN[n] = 1), the framing bit locations are unused.

The framing mode of those links mapped to SPE 3 of the SBI ADD BUS is configured using the FEN[11:0] bits of the **TCAS SBI SPE3 Configuration Register #1** (0x450) and the FEN[27:12] bits of the **TCAS SBI SPE3 Configuration Register #2** (0x454). Each FEN bit, FEN[n], configures link 3n+2 for framed operation. In unframed operation (FEN[n] = 0), HDLC data is transmitted in all framing bit locations. In framed mode (FEN[n] = 1), the framing bit locations are unused.

Idle Time-Slot Fill Data

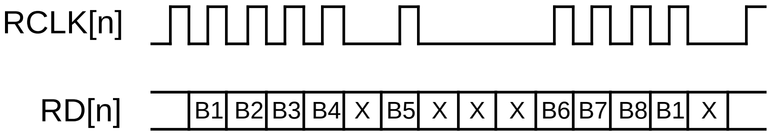
The fill data bits (FDATA[7:0]) of the **TCAS Idle Time-slot Fill Data** (0x40C) register are transmitted during disabled time-slots of a channelised link (when the PROV bit of the **TCAS Indirect Channel Data** (0x404) register is low). The default value of FDATA[7:0] is 0xFF.

7.2 Clock/Data Links

When the SPEn_EN input pin is low, the corresponding Synchronous Payload Envelope conveyed on the SBI interface is unused and the corresponding independently timed link (signals RCLK[n-1], RD[n-1], TCLK[n-1] and TD[n-1]) is enabled, where $1 \le n \le 3$.

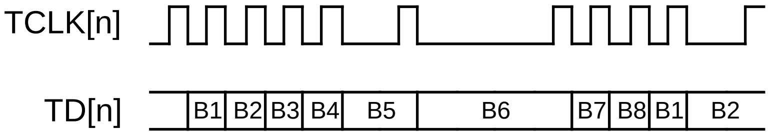
The timing relationship of the receive clock (RCLK[n]) and data (RD[n]) signals is shown in Figure 2, where $0 \le n \le 2$. The receive data is viewed as a contiguous serial stream. There is no concept of time-slots or framing. Every eight bits are grouped together into a byte with arbitrary alignment. The first bit received (B1 in Figure 2) is deemed the most significant bit of an octet. The last bit received (B8) is deemed the least significant bit. Bits that are to be processed by the FREEDM-84A672 are clocked in on the rising edge of RCLK[n]. Bits that should be ignored (X in Figure 2) are squelched by holding RCLK[n] quiescent. In Figure 2, the quiescent period is shown to be a low level on RCLK[n]. A high level, effected by extending the high phase of the previous valid bit, is also acceptable. Selection of bits for processing is arbitrary and is not subject to any byte alignment nor frame boundary considerations.

Figure 2 – Receive Link Timing



The timing relationship of the transmit clock (TCLK[n]) and data (TD[n]) signals is shown in Figure 3, where $0 \le n \le 2$. The transmit data is viewed as a contiguous serial stream. There is no concept of time-slots or framing. Every eight bits are grouped together into a byte with arbitrary byte alignment. Octet data is transmitted from most significant bit (B1 in Figure 3) and ending with the least significant bit (B8 in Figure 3). Bits are updated on the falling edge of TCLK[n]. A transmit link may be stalled by holding the corresponding TCLK[n] quiescent. In Figure 3, bits B5 and B2 are shown to be stalled for one cycle while bit B6 is shown to be stalled for three cycles. In Figure 3, the quiescent period is shown to be a low level on TCLK[n]. A high level, effected by extending the high phase of the previous valid bit, is also acceptable. Gapping of TCLK[n] can occur arbitrarily without regard to byte nor frame boundaries.

Figure 3 – Transmit Link Timing



The following registers control the operation of receive links #0 to #2 when they are configured to receive data from the RD[2:0] inputs (i.e. SPEn_EN input pin is low). Since the only mode of operation of the clock/data links is unchannelised mode, no additional configuration is necessary. However, the programmer must ensure that the reserved bits in the following RCAS672 and TCAS672 registers are set low for correct operation of the FREEDM-84A672.

Bit	Register	Value
Reserved[2:0]	**RCAS Links #0 to #2 Configuration** (0x180 – 0x188)	000
Reserved[3]	**RCAS Links #0 to #2 Configuration** (0x180 – 0x188)	0

Bit	Register	Value
Reserved[2:0]	**TCAS Links #0 to #2 Configuration** (0x480 – 0x488)	000
Reserved[3]	**TCAS Links #0 to #2 Configuration** (0x480 – 0x488)	0

8 CONFIGURING THE ANY-PHY PACKET INTERFACE

The RAPI672 and TAPI672 blocks must be configured via the normal mode registers in order to enable the transferring of data between the partial packet buffers and the receive and transmit APPI.

8.1 Configuring the Receive Any-PHY Packet Interface (RAPI672)

The RAPI672 is configured by programming bits within the **RAPI Control** (0x580) register. The values programmed affect all receive channels. The default configuration is as follows:

Bit	Register	Value
BADDR[2:0]	**RAPI Control** (0x580)	111
ALL1ENB	**RAPI Control** (0x580)	1
Reserved	**RAPI Control** (0x580)	0
STATEN	**RAPI Control** (0x580)	0
ENABLE	**RAPI Control** (0x580)	0

The default indicates that the RAPI672 is disabled from responding to device selection.

Activation of the RAPI672

By default, the RAPI672 is disabled from responding to device selection. The ENABLE bit must be set to enable normal operation of the RAPI672. The encoding of this bit is:

ENABLE	Function
0	The RAPI672 will not respond to device selection.
1	The RAPI672 operates normally, and will respond to device selection.

Base Address of the Receive APPI

The base address bits (BADDR[2:0]) configure the address space occupied by the FREEDM-84A672 device for purposes of responding to receive polling and receive device selection. During polling, the BADDR[2:0] bits are used to respond to polling via the RXADDR[2:0] pins. During device selection, the

BADDR[2:0] are used to select a FREEDM-84A672 device, enabling it to accept data on the receive APPI. During data transfer, the RXDATA[15:13] pins of the prepended channel address reflect the BADDR[2:0] bits.

All Ones Enable

The All Ones Enable bit (ALL1ENB) permits the FREEDM-84A672 to respond to receive polling and device selection when BADDR[2:0] = '111'. The encoding of this bit is:

ALL1ENB	Function
0	The FREEDM-84A672 responds to receive polling and device selection when BADDR[2:0] = RXADDR[2:0] = '111'.
1	The FREEDM-84A672 regards the all-ones address as a null address and does not respond to receive polling and device selection when BADDR[2:0] = '111', regardless of the value of RXADDR[2:0].

Status Enable

The FREEDM-84A672 can be programmed to overwrite the receive data signal pins, RXDATA[7:0], of the final word of each packet transfer (REOP pin is high) with the status of packet reception when that packet is errored (RERR pin is high). The RAPI672 Status Enable bit enables this feature as follows:

STATEN	Function
0	The RAPI672 does not report detailed status information for an errored packet.
1	The RAPI672 overwrites RXDATA[7:0] of the final word of an errored packet with status information for that packet.

When STATEN = 1 and the REOP and RERR pins are both high, the status information is bit mapped on RXDATA[7:0] as follows:

Bit	Status
RXDATA[0] = 1	Channel FIFO overrun
RXDATA[1] = 1	Maximum packet length violation
RXDATA[2] = 1	FCS error
RXDATA[3] = 1	Non-octet aligned

Bit	Status
RXDATA[4] = 1	HDLC packet abort
RXDATA[7:5] = XH	Reserved

8.2 Configuring the Transmit Any-PHY Packet Interface (TAPI672)

The TAPI672 is configured by programming bits within the **TAPI Control** (0x600) register. The values programmed affect all transmit channels. The default configuration is as follows:

Bit	Register	Value
BADDR[2:0]	**TAPI Control** (0x600)	111
ALL1ENB	**TAPI Control** (0x600)	1
Reserved[1:0]	**TAPI Control** (0x600)	00
ENABLE	**TAPI Control** (0x600)	0
BLEN[7:0]	**TAPI Indirect Channel Data Register** (0x608)	0x00

The default indicates that data provided to the TAPI672 by the transmit APPI will be ignored.

Activation of the TAPI672

By default, data provided to the TAPI672 by the transmit APPI is ignored. The ENABLE bit must be set to enable normal operation of the TAPI672. The encoding of this bit is:

ENABLE	Function
0	The state machines in the TAPI672 are held in their idle state. The TAPI672 will complete the current data transfer and will respond to any further transactions on the transmit APPI normally (by setting TRDY high), but data provided will be ignored.
1	The TAPI672 operates normally. Data can be transferred from the transmit APPI to the partial packet buffer in the THDL672.

Base Address of the Receive APPI

The base address bits (BADDR[2:0]) configure the address space occupied by the FREEDM-84A672 device for purposes of responding to transmit polling and transmit data transfers. During polling, the TXADDR[12:10] pins are compared with the BADDR[2:0] bits to determine if the poll address identified by TXADDR[9:0] is intended for a channel in this FREEDM-84A672 device. During data transmission, the TXDATA[15:13] pins of the prepended channel address are compared with the BADDR[2:0] bits to determine if the data to follow is intended for this FREEDM-84A672 device.

All Ones Enable

The All Ones Enable bit (ALL1ENB) permits the FREEDM-84A672 to respond to transmit polling and device selection when BADDR[2:0] = ‘111’. The encoding of this bit is:

ALL1ENB	Function
0	The FREEDM-84A672 responds to transmit polling when BADDR[2:0] = TXADDR[12:10] = ‘111’ and device selection when BADDR[2:0] = TXDATA[15:13] = ‘111’.
1	The FREEDM-84A672 regards the all-ones address as a null address and does not respond to transmit polling and device selection when BADDR[2:0] = ‘111’, regardless of the values of TXADDR[12:10] and TXDATA[15:13].

Channel Burst Length

The channel burst length (BLEN[7:0]) bits report the data transfer burst length read from the TAPI672 channel provision RAM after an indirect read operation has completed. The data transfer burst length specifies the length (in bytes, less one) of burst data transfers on the transmit APPI which are not terminated by the assertion of TEOP. The data transfer burst length can be specified on a per-channel basis with burst lengths of up to 256 bytes. The data transfer burst length to be written to the channel provision RAM in an indirect write operation must be set up in this register before triggering the write. BLEN[7:0] reflects the value written until the completion of a subsequent indirect read operation.

The BLEN[7:0] value must be set according to the indirect channel transfer size of the THDL672 block (XFER[3:0] in the **THDL Indirect Channel Data #3** (0x38C) register) using the following equation:

$$BLEN[7:0] = (XFER[3:0] + 1) \times 16 - 1.$$

A description of the XFER[3:0] bits can be found in section 9.5. The relationship between XFER[3:0] and BLEN[7:0] is shown in the following table.

XFER[3:0]	**BLEN[7:0]**
0000	0x0F
0001	0x1F
0010	0x2F
0011	0x3F
0100	0x4F
0101	0x5F
0110	0x6F
0111	0x7F
1000	0x8F
1001	0x9F
1010	0xAF
1011	0xBF
1100	0xCF
1101	0xDF
1110	0xEF
1111	0xFF

9 HDLC AND CHANNEL FIFO CONFIGURATION

The FREEDM-84A672 processes the data stream in the receive direction via the RHDL672 block and it processes the data stream in the transmit direction via the THDL672 block. Each of these blocks must be configured via the Normal Mode Register Space.

9.1 Configuring the RHDL672

The RHDL672 is configured by programming bits within the **RHDL Configuration** (0x220) register and the **RHDL Maximum Packet Length** (0x224) register. The values programmed affect all receive channels. The default configuration is as follows:

Bit	Register	Value
LENCHK	**RHDL Configuration** (0x220)	0
TSTD	**RHDL Configuration** (0x220)	0
MAX[15:0]	**RHDL Maximum Packet Length** (0x224)	0xFFFF

The default indicates no maximum packet length checking and datacom bit ordering.

Maximum Packet Length

The RHDL672 may be configured to abort packets which exceed the maximum length of *n* where $0 \leq n \leq 0xFFFF$. The following bits are written to enable or disable this feature:

LENCHK	MAX[15:0]	Function
0	0xFFFF	Receive packets are not checked for maximum size and MAX[15:0] must be set to 0xFFFF.
1	n	Receive packets with total length, including address, control, information and FCS fields, greater than MAX[15:0] bytes are aborted and the remainder of the frame discarded.

Datacom/Telecom Bit Order

The RHDL672 may be configured to reverse the order of bits in the HDLC data transferred across the receive APPI. The following bit is written to specify the order of bits:

TSTD	Function
0	Datacom standard: least significant bit of each byte on the receive APPI bus (AD[0], AD[8], AD[16], AD[24]) is the first HDLC bit received. Normally, when HDLC processing is enabled, the TSTD bit must be set to zero.
1	Telecom standard: most significant bit of each byte on the receive APPI bus (AD[7], AD[15], AD[23], AD[31]) is the first HDLC bit received.

9.2 Configuring the THDL672

The THDL672 is configured by programming bits within the **THDL Configuration** (0x3B0) register. The values programmed affect all transmit channels. The default configuration is as follows:

Bit	Register	Value
Reserved[3:0]	**THDL Configuration** (0x3B0)	0x0
Reserved[4]	**THDL Configuration** (0x3B0)	0
TSTD	**THDL Configuration** (0x3B0)	0
BIT8	**THDL Configuration** (0x3B0)	0

The default indicates that data is formatted in datacom bit ordering.

Datacom/Telecom Bit Order

The THDL672 may be configured to reverse the order of bits in the HDLC data transferred on the transmit APPI. The following bit is written to specify the order of bits:

TSTD	Function
0	Datacom standard: least significant bit of each byte on the transmit APPI bus (AD[0], AD[8], AD[16], AD[24]) is the first HDLC bit transmitted. Normally, when HDLC processing is enabled, the TSTD bit must be set to zero.

TSTD	Function
1	Telecom standard: most significant bit of each byte on the transmit APPI bus (AD[7], AD[15], AD[23], AD[31]) is the first HDLC bit transmitted.

BIT8

The BIT8 field affects channels of the THDL672 that are configured with 7BIT set. The BIT8 value specifies the data bit transmitted on the least significant bit of each octet.

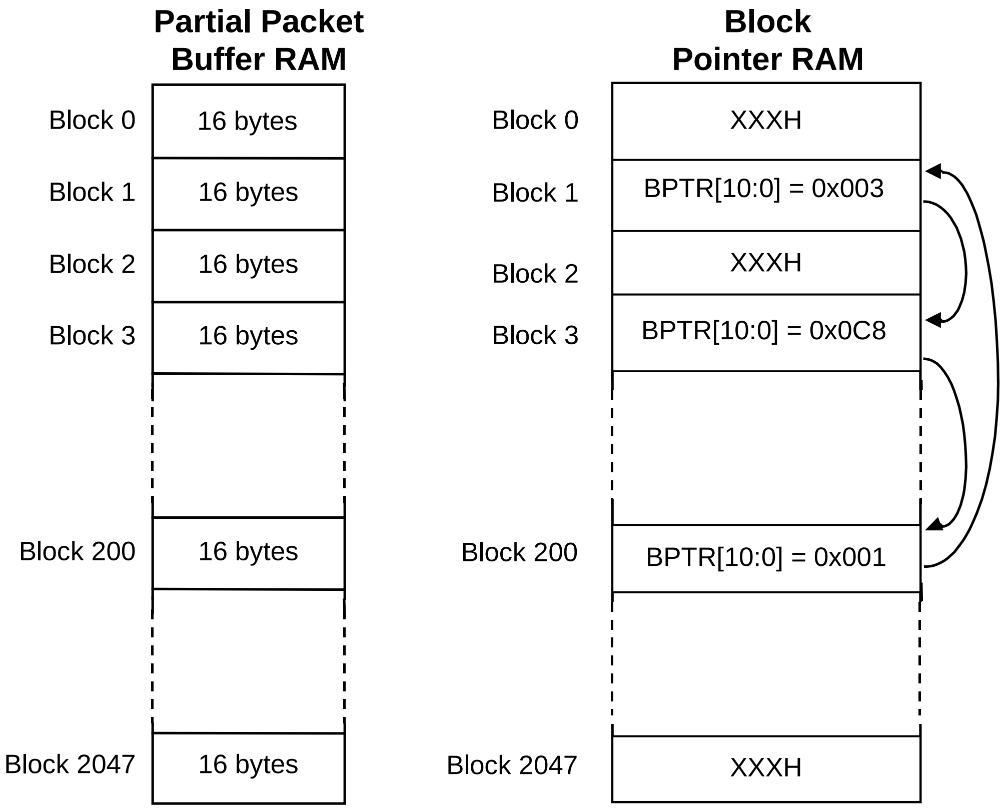
BIT8	Function
0	Channels configured for 7BIT will transmit a zero on the least significant bit of each octet.
1	Channels configured for 7BIT will transmit a one on the least significant bit of each octet.

9.3 Programming a Channel FIFO

A Channel FIFO is created from 3 or more blocks of internal RAM, and each block holds 16 bytes of packet data. There is a total of 2048 blocks (32 Kbytes) available to assign among the receive channels, and another 2048 blocks (32 Kbytes) available to assign among the transmit channels.

A FIFO is created by assigning a circular linked list of blocks as shown in Figure 4. This shows a channel FIFO consisting of 3 blocks. The quantity of buffers and the arrangement of links is chosen by the programmer, and the selection of blocks can be arbitrary. The programmer must ensure that a block is not assigned to more than one circularly linked list.

Figure 4 – Specifying a Channel FIFO



9.3.1 Receive Channel FIFO

A receive channel FIFO is programmed by repeating the following procedure for each block within the circularly linked list:

1. Poll the BUSY bit of the **RHDL Indirect Block Select** (0x210) register until it is zero. This ensures that a previous indirect RAM access has completed and that a new indirect RAM access can be started.

2. Write the following register with the next block in the circular linked list, or exit if all links have been programmed:

Bit	Register	Value
BPTR[10:0]	**RHDL Indirect Block Data** (0x214)	0 through 0x7FF are valid
Reserved	**RHDL Indirect Block Data** (0x214)	0

3. Specify the block and update the internal block pointer RAM by writing the following register. Proceed to step 1.

Bit	Register	Value
BLOCK[10:0]	**RHDL Indirect Block Select** (0x210)	0 through 0x7FF are valid
Reserved	**RHDL Indirect Block Select** (0x210)	0
BRWB	**RHDL Indirect Block Select** (0x210)	0
BUSY	**RHDL Indirect Block Select** (0x210)	X

9.3.2 Transmit Channel FIFO

A transmit channel FIFO is programmed by repeating the following procedure for each block within the circularly linked list:

1. Poll the BUSY bit of the **THDL Indirect Block Select** (0x3A0) register until it is zero. This ensures that a previous indirect RAM access has completed and that a new indirect RAM access can be started.

2. Write the following register with the next block in the circular linked list, or exit if all links have been programmed:

Bit	Register	Value
BPTR[10:0]	**THDL Indirect Block Data** (0x3A4)	0 through 0x7FF are valid
Reserved[0]	**THDL Indirect Block Data** (0x3A4)	0
Reserved[1]	**THDL Indirect Block Data** (0x3A4)	0

3. Specify the block and update the internal block pointer RAM by writing the following register. Proceed to step 1.

Bit	Register	Value
BLOCK[10:0]	**THDL Indirect Block Select** (0x3A0)	0 through 0x7FF are valid
Reserved	**THDL Indirect Block Select** (0x3A0)	0
BRWB	**THDL Indirect Block Select** (0x3A0)	0

Bit	Register	Value
BUSY	**THDL Indirect Block Select** (0x3A0)	X

9.4 RHDL672 Channel Configuration

The RHDL672 provides configurable options for each receive channel as identified in the following register fields:

Bit	Register
DELIN	**RHDL Indirect Channel Data Register #1** (0x204)
STRIP	**RHDL Indirect Channel Data Register #1** (0x204)
XFER[3:0]	**RHDL Indirect Channel Data Register #2** (0x208)
OFFSET[1:0]	**RHDL Indirect Channel Data Register #2** (0x208)
CRC[1:0]	**RHDL Indirect Channel Data Register #2** (0x208)
INVERT	**RHDL Indirect Channel Data Register #2** (0x208)
PRIORITY	**RHDL Indirect Channel Data Register #2** (0x208)
7BIT	**RHDL Indirect Channel Data Register #2** (0x208)

Note: When writing to **RHDL Indirect Channel Data Register #1** (0x204), the reserved bit (bit 11) must be set low for correct operation of the FREEDM-84A672.

Delineation

The data bits from the RCAS672 can be written directly to the Partial Packet Buffer or processed for flag sequence delineation, bit de-stuffing and CRC verification. The following bit enables or disables this feature:

DELIN	Function
0	Data is written to the Partial Packet Buffer without any HDLC processing (no flag sequence delineation, bit de-stuffing nor CRC verification) on the incoming stream.
1	Data is processed for flag sequence delineation, bit de-stuffing and optionally, CRC verification (CRC verification depends on CRC[1:0] value).

Strip FCS Bit

The indirect frame check sequence discard bit (STRIP) enables the RHDL672 to remove the FCS data before writing to the channel FIFO. STRIP is ignored when DELIN is low or when CRC[1:0] = 00B. This feature is configured as follows:

STRIP	Function
0	Includes FCS data with the data stream written to the channel FIFO.
1	Removes the FCS data from the data stream written to the channel FIFO.

DMA Transfer Size

The indirect channel transfer size configures the amount of data transferred in each transaction. When the channel FIFO depth reaches the depth specified by XFER[3:0] or when an end-of-packet exists in the FIFO, a poll of this FREEDM-84A672 device will indicate that data exists and is ready to be transferred across the receive APPI. Specifying a large transfer size may affect APPI bus access latencies for other channels. The following bits specify the channel transfer size:

XFER[3:0]	Function
0 through 15 are valid	Specifies the data transfer size in blocks: Blocks = XFER[3:0] +1, and there are 16 bytes per block.

Note: XFER[3:0] should be set such that the number of blocks transferred is at least two fewer than the total allocated to the associated channel.

Insertion of Offset Bytes

The RHDL672 can be configured to insert offset bytes into the data stream before writing the data stream to the channel FIFO. The offset bytes are placed before each packet and their value is undefined. The following configuration options are available:

OFFSET[1:0]	Function
00	RHDL672 does not insert offset bytes
01	RHDL672 inserts 1 offset byte per packet
10	RHDL672 inserts 2 offset bytes per packet

OFFSET[1:0]	Function
11	RHDL672 inserts 3 offset bytes per packet

CRC Algorithm

The RHDL672 can perform CRC verification of the incoming data stream. The available options are as follows:

CRC[1:0]	DELIN	Function
X	0	No CRC verification
00	1	No CRC verification
01	1	CRC-CCITT verification
10	1	CRC-32 verification
11	1	Reserved

HDLC Data Inversion

The INVERT bit configures the RHDL672 to logically invert the incoming HDLC stream from the RCAS672 before processing it. The bit is specified as follows:

INVERT	Function
0	HDLC stream is not inverted.
1	HDLC stream is inverted.

Specifying Receive Channel Priority

All receive channels that must transfer data from their channel FIFO to packet memory contend for access to the receive APPI bus. The PRIORITY bit allows specified channels to have priority access to the receive APPI bus. The bit encoding is as follows:

PRIORITY	Function
0	This channel is serviced after channels with PRIORITY=1.
1	This channel is serviced before channels with PRIORITY=0.

Handling of Robbed bit Signaling

The 7BIT enable bit configures the RHDL672 to ignore the least significant bit of each octet (last bit of each octet received) in the incoming channel stream. This bit is encoded as follows:

7BIT	Function
0	The entire receive data stream is processed.
1	The least significant bit (last bit of each octet received) is ignored.

9.5 THDL672 Channel Configuration

The THDL672 provides configurable options for each transmit channel as identified in the following register fields:

Bit	Register
DELIN	**THDL Indirect Channel Data Register #1** (0x384)
CRC[1:0]	**THDL Indirect Channel Data Register #1** (0x384)
FLEN[10:0]	**THDL Indirect Channel Data Register #2** (0x388)
DFCS	**THDL Indirect Channel Data Register #2** (0x388)
INVERT	**THDL Indirect Channel Data Register #2** (0x388)
7BIT	**THDL Indirect Channel Data Register #2** (0x388)
XFER[3:0]	**THDL Indirect Channel Data Register #3** (0x38C)
FLAG[2:0]	**THDL Indirect Channel Data Register #3** (0x38C)
LEVEL[3:0]	**THDL Indirect Channel Data Register #3** (0x38C)
IDLE	**THDL Indirect Channel Data Register #3** (0x38C)
TRANS	**THDL Indirect Channel Data Register #3** (0x38C)

Note: When writing to **THDL Indirect Channel Data Register #1** (0x384), the reserved bit (bit 11) must be set low for correct operation of the FREEDM-84A672. When writing to **THDL Indirect Channel Data Register #2** (0x388), the reserved bit (bit 11) must be set low for correct operation of the FREEDM-84A672.

Frame Delineation

The transmit packet data from packet memory can be written directly to the outgoing data stream or processed for flag sequence insertion, bit stuffing and CRC generation. The following bit enables or disables this feature:

DELIN	Function
0	Data is written directly to the outgoing data stream without any HDLC processing (no flag sequence insertion, bit stuffing nor CRC generation).
1	Data is processed for flag sequence insertion, bit stuffing and optionally, CRC generation (CRC generation depends on CRC[1:0] value).

CRC Algorithm

The THDL672 can perform CRC generation on the outgoing data stream. The available options are as follows:

CRC[1:0]	DELIN	Function
X	0	No CRC generation
00	1	No CRC generation
01	1	CRC-CCITT generation
10	1	CRC-32 generation
11	1	Reserved

Channel FIFO Length

The indirect FIFO length (FLEN[10:0]) is the number of blocks, less one, that is provisioned to the circular channel FIFO specified by the FPTR[10:0] block pointer.

FLEN[10:0]	Function
0 through 2047 are valid	Specifies the Channel FIFO size in blocks, where Blocks = FLEN[10:0] + 1, and each block is 16 bytes.

Inverting the FCS

The diagnose frame check sequence bit (DFCS) specifies whether the FCS field inserted into the transmit data stream is inverted. This is provided for diagnostic purposes and is programmed as follows:

DFCS	Function
0	FCS field in the outgoing HDLC stream is not inverted.
1	FCS field in the outgoing HDLC stream is logically inverted.

Robbed Bit Signaling

The least significant stuff enable bit (7BIT) configures the THDL672 to stuff the least significant bit of each octet assigned to the transmit channel in the outgoing channel stream.

7BIT	Function
0	The entire octet contains valid data and BIT8 is ignored.
1	The least significant bit (last bit of each octet transmitted) does not contain channel data and is forced to the value configured by the BIT8 register bit.

DMA Transfer Size

The indirect channel transfer size specifies the amount of data that the partial packet processor requests from the TAPI672 block. When the channel FIFO free space reaches or exceeds the limit specified by XFER[3:0], the partial packet processor will inform the TAPI672 so that a poll on that channel reflects that the channel FIFO is able to accept XFER[3:0] + 1 blocks of data. Specifying a large transfer size may affect APPI bus access latencies for other channels. The following bits specify the channel transfer size:

XFER[3:0]	Function
0 through 15 are valid	Specifies the data transfer size in blocks, where Blocks = XFER[3:0] + 1, and each block is 16 bytes.

Note: To prevent lockup, the channel transfer size (XFER[3:0]) can be configured to be less than or equal to the start transmission level set by LEVEL[3:0] and TRANS. Alternatively, the channel transfer size can be set such that the total number of blocks in the logical channel FIFO minus the start transmission level is an integer multiple of the channel transfer size.

Specifying The Number of Flag or Idle Bytes Inserted Between Frames

The THDL672 can be configured to insert either flag or idle bytes (8 bits of one's) into the data stream between HDLC packets. The number of these is programmed as follows:

FLAG[2:0]	Minimum Number of Flag/Idle Bytes
000	1 flag / 0 Idle byte
001	2 flags / 0 idle byte
010	4 flags / 2 idle bytes
011	8 flags / 6 idle bytes
100	16 flags / 14 idle bytes
101	32 flags / 30 idle bytes
110	64 flags / 62 idle bytes
111	128 flags / 126 idle bytes

Interframe Time Fill

The IDLE bit specifies the byte pattern inserted in the data stream between HDLC packets.

IDLE	Function
0	Flag bytes are inserted between HDLC packets.
1	HDLC idle (all one's bit with no bit-stuffing) is inserted between HDLC packets.

Specifying the Channel FIFO's Starving Level and Start Transmit Level

The HDLC processor starts transmitting a packet when the channel FIFO free space is less than or equal to the level specified in the appropriate Start Transmission Level column of the following table or when an end of a packet is stored in the channel FIFO.

When the channel FIFO free space is less than or equal to than the level specified in the Starving Trigger Level column of the following table and the HDLC processor is transmitting a packet and an end of a packet is not stored in the channel FIFO, the partial packet buffer makes expedite requests to the TAPI672 to retrieve XFER[3:0] + 1 blocks of data.

The starving trigger level and start transmission level are programmed via the LEVEL[3:0] and the TRANS field as follows:

LEVEL[3:0]	Starving Trigger Level	Start Transmission Level (TRANS=0)	Start Transmission Level (TRANS=1)
0000	2 Blocks (32 bytes free)	1 Block (16 bytes free)	1 Block (16 bytes free)
0001	3 Blocks (48 bytes free)	2 Blocks (32 bytes free)	1 Block (16 bytes free)
0010	4 Blocks (64 bytes free)	3 Blocks (48 bytes free)	2 Blocks (32 bytes free)
0011	6 Blocks (96 bytes free)	4 Blocks (64 bytes free)	3 Blocks (48 bytes free)
0100	8 Blocks (128 bytes free)	6 Blocks (96 bytes free)	4 Blocks (64 bytes free)
0101	12 Blocks (192 bytes free)	8 Blocks (128 bytes free)	6 Blocks (96 bytes free)
0110	16 Blocks (256 bytes free)	12 Blocks (192 bytes free)	8 Blocks (128 bytes free)
0111	24 Blocks (384 bytes free)	16 Blocks (256 bytes free)	12 Blocks (192 bytes free)
1000	32 Blocks (512 bytes free)	24 Blocks (384 bytes free)	16 Blocks (256 bytes free)
1001	48 Blocks (768 bytes free)	32 Blocks (512 bytes free)	24 Blocks (384 bytes free)
1010	64 Blocks (1 Kbytes free)	48 Blocks (768 bytes free)	32 Blocks (512 bytes free)
1011	96 Blocks (1.5 Kbytes free)	64 Blocks (1 Kbytes free)	48 Blocks (768 bytes free)
1100	192 Blocks (3 Kbytes free)	128 Blocks (2 Kbytes free)	96 Blocks (1.5 Kbytes free)
1101	384 Blocks (6 Kbytes free)	256 Blocks (4 Kbytes free)	192 Blocks (2 Kbytes free)
1110	768 Blocks (12 Kbytes free)	512 Blocks (8 Kbytes free)	384 Blocks (4 Kbytes free)

LEVEL[3:0]	Starving Trigger Level	Start Transmission Level (TRANS=0)	Start Transmission Level (TRANS=1)
1111	1536 Blocks (24 Kbytes free)	1024 Blocks (16 Kbytes free)	768 Blocks (8 Kbytes free)

10 FREEDM-84A672 OPERATIONAL PROCEDURES

10.1 Device Identification, Location and System Resource Assignment

This section describes the software interaction required to identify a FREEDM-84A672 device on the APPI bus, and to map the Normal Mode Registers in the microprocessor memory map.

Identifying and Locating a FREEDM-84A672

The software can identify a FREEDM-84A672 attached to an APPI bus by reading the TYPE[3:0] bits in the **FREEDM-84A672 Master Reset** (0x000) register. The default value of TYPE[3:0] = 0101B indicates that the device is the FREEDM-84A672 member of the FREEDM family of products.

For purposes of responding to receive polling and receive device selection, the address space occupied by each FREEDM-84A672 on the receive APPI needs to be configured using the base address bits (BADDR[2:0]) in the **RAPI Control** (0x580) register. Similarly, for purposes of responding to transmit polling and transmit data transfers, the address space occupied by each FREEDM-84A672 on the transmit APPI needs to be configured using the base address bits (BADDR[2:0]) in the **TAPI Control** (0x600) register. Note that up to seven FREEDM-84A672 devices may share a single APPI bus (one address is reserved as a null address), with an external controller acting as bus master.

In addition, the software can identify the version level of the FREEDM-84A672 with the ID[7:0] bits in the **FREEDM-84A672 Master Reset** (0x000) register. This may be useful to distinguish between future versions of the FREEDM-84A672.

Memory Mapping the Register Space

During power-up, the Normal Mode Register space needs to be mapped to the microprocessor. This register space is located in the FREEDM-84A672 and is accessed through the microprocessor interface. All registers are 16 bits wide but are dword aligned in the microprocessor memory map.

10.2 Reset

This section describes the procedure to reset the FREEDM-84A672 via software. The FREEDM-84A672 is powered on in an inactive state and should be reset via software following a hardware reset, or as required by the embedded processor. The reset procedure is normally followed by the initialization procedure.

The steps to reset a FREEDM-84A672 are:

1. If the FREEDM-84A672 was active before the reset procedure then the deactivation procedure must be done (see section 10.5).

2. The RESET bit in the **FREEDM-84A672 Master Reset** (0x000) register must be written high and then written low.

This reset procedure has the following effects:

- The RESET bit allows the FREEDM-84A672 to be reset under software control. If the RESET bit is a logic one, the entire FREEDM-84A672 except the microprocessor interface is held in reset. This bit is not self-clearing. Therefore, a logic zero must be written to bring the FREEDM-84A672 out of reset. Holding the FREEDM-84A672 in a reset state places it into a low power, stand-by mode. A hardware reset clears the RESET bit, thus negating the software reset.

- All Normal Mode registers are set to their default values.

- None of the channel provisioning, or the Channel FIFO configuration is preserved under software reset.

10.3 Initialization

This section describes the procedure to initialize the FREEDM-84A672. The initialization procedure normally follows the software reset procedure and is followed by the activation procedure.

The steps to initialize a FREEDM-84A672 are:

1. Configure the SBI interface, and the SBI Extracter and Inserter for the SPEs conveyed on the SBI interface. The register accesses are described in sections 5 and 6.

2. Configure the RCAS672 and TCAS672 serial links. The register accesses are described in section 7.

3. Assign base addresses for the receive and the transmit APPI. The register accesses are described in sections 8.1 and 8.2.

4. Configure HDLC processing of the RHDL672 and the THDL672 blocks. The register accesses are described in sections 9.1 and 9.2.

10.4 Activation Procedure

The activation procedure is required to place the FREEDM-84A672 in a state after which the software may service FREEDM-84A672 interrupts, provision/unprovision channels, and monitor the status of the FREEDM-84A672.

The activation procedure normally follows the initialization procedure.

The steps to activate a FREEDM-84A672 are:

1. Enable interrupt 'E' bits, SBIEXTE and SBI_PERR_EN as described in section 4.

2. Enable data transfer across the receive and transmit APPI by setting the ENABLE bits to one in the RAPI672 and TAPI672 registers as described in sections 8.1 and 8.2.

3. The SYSCLKA, REFCLKA, FASTCLKA, C1FPA, RXCLKA, and TXCLKA bits in the **FREEDM-84A672 Master Clock/Frame Pulse Activity Monitor and Accumulation Trigger** (0x00C) register should be read periodically to detect for stuck at conditions. The SYSCLKA bit must be read high for proper operation of the FREEDM-84A672. A low value indicates a failure in clocking that is provided at the SYSCLK input pin of the FREEDM-84A672. Similarly, a low value in the other register bits indicates a failure in clocking that is provided by the corresponding input pin.

10.5 Deactivation Procedure

The deactivation procedure is required to place the FREEDM-84A672 in a state in which it will not interrupt the embedded processor, or transfer data across the APPI. This procedure should occur after the FREEDM-84A672 actively transfers packets, or to gracefully shut down the FREEDM-84A672.

The steps to deactivate a FREEDM-84A672 are:

1. Disable interrupt 'E' bits, SBIEXTE and SBI_PERR_EN as described in section 4.

2. Disable data transfer across the receive and transfer APPI by programming the ENABLE bits to zero in the RAPI672 and TAPI672 registers as described in sections 8.1 and 8.2.

3. Continue by performing the software reset procedure.

10.6 Provisioning a Channel

The provisioning procedure normally follows the activation procedure and enables the FREEDM-84A672 to receive and/or transmit packets.

10.6.1 Receive Channel Provisioning

The steps to provision a receive channel *RCC*, where *0 ≤ RCC ≤ 671* are:

1. Disable FREEDM-84A672 processing of the channel's data stream to allow for graceful provisioning. Write the following bits:

Bit	Register	Value
DCHAN[9:0]	**RCAS Channel Disable** (0x10C)	*RCC*
CHDIS	**RCAS Channel Disable** (0x10C)	1

2. Program the Channel FIFO as described in section 9.3.1.

3. Poll the BUSY bit of the **RHDL Indirect Channel Select** (0x200) register until it is zero. This ensures that a previous indirect RAM access has completed and that a new indirect RAM access can be started.

4. Specify the HDLC configuration for this channel by writing appropriate bits in the **RHDL Indirect Channel Data Register #1** (0x204) and the **RHDL Indirect Channel Data Register #2** (0x208) as described in section 9.4. When writing the **RHDL Indirect Channel Data Register #1** (0x204), ensure that the PROV bit is set, and ensure that the FPTR[10:0] bits identify a block within the circular linked list of buffers of step 2.

5. Specify the RHDL672 channel to provision by writing the following register. Then poll the BUSY bit to ensure that it is low before proceeding to step 6.

Bit	Register	Value
CHAN[9:0]	**RHDL Indirect Channel Select** (0x200)	*RCC*
CRWB	**RHDL Indirect Channel Select** (0x200)	0
BUSY	**RHDL Indirect Channel Select** (0x200)	X

6. Poll the BUSY bit of the **RCAS Indirect Link and Time-slot Select** (0x100) register until it is zero. This ensures that a previous indirect RAM access has completed and that a new indirect RAM access can be started.

7. Specify the RCAS672 channel that is provisioned. Write the following register:

Bit	Register	Value
CHAN[9:0]	**RCAS Indirect Channel Data** (0x104)	*RCC*
PROV	**RCAS Indirect Channel Data** (0x104)	1
CDLBEN	**RCAS Indirect Channel Data** (0x104)	0

8. For a **channelised** link, specify the time-slots which are assigned for processing on this channel by writing the following register once for each time-slot that is assigned to the channel. Valid values for TSLOT[4:0] are 1 through 24 for a T1/J1 link, and 1 through 31 for an E1 link. For an **unchannelised** or **unframed** link, TSLOT[4:0] must only have the value 0, and this register is written just once. Each write must be followed by a read to determine whether the BUSY bit (bit15) is low, and to ensure that the indirect RAM has been updated.

Bit	Register	Value
TSLOT[4:0]	**RCAS Indirect Link and Time-slot Select** (0x100)	see above
LINK[6:0]	**RCAS Indirect Link and Time-slot Select** (0x100)	0 through 83 are valid
RWB	**RCAS Indirect Link and Time-slot Select** (0x100)	0
BUSY	**RCAS Indirect Link and Time-slot Select** (0x100)	X

9. Enable FREEDM-84A672 processing of the channel data stream to allow for graceful provisioning. Write the following bits:

Bit	Register	Value
DCHAN[9:0]	**RCAS Channel Disable** (0x10C)	*RCC*
CHDIS	**RCAS Channel Disable** (0x10C)	0

Warning:

- The RCAS Channel Disable bit (CHDIS) is only applicable to one channel at a time. In other words, the receive channel provisioning procedure needs to be run once for each channel.

- The programmer must ensure that the channel has not been provisioned, or has been unprovisioned before doing the provisioning procedure. The reset procedure has the effect of unprovisioning all channels of the FREEDM-84A672.

- Continuous polling of a register in a tight loop involves multiple microprocessor memory read transactions and may have an adverse effect on the microprocessor bus bandwidth available for other activities. The recommended method of polling the BUSY bit is to read the register on expiration of a system timer, or after a number of CPU clock ticks. Recommended time intervals are in the range 0.1 msec through 1 msec.

- A Channel is not provisioned until the BUSY bit toggles low.

10.6.2 Transmit Channel Provisioning

The steps to provision a transmit channel *TCC*, where *0 ≤ TCC ≤ 671* are:

1. Disable FREEDM-84A672 processing of the channel's data stream to allow for graceful provisioning. Write the following bits:

Bit	Register	Value
DCHAN[9:0]	**TCAS Channel Disable** (0x410)	*TCC*
CHDIS	**TCAS Channel Disable** (0x410)	1

2. Program the Channel FIFO as described in section 9.3.2 for a transmit channel.

3. Poll the BUSY bit of the **THDL Indirect Channel Select** (0x380) register until it is zero. This ensures that a previous indirect RAM access has completed and that a new indirect RAM access can be started.

4. Specify the HDLC configuration for this transmit channel by writing the **THDL Indirect Channel Data Register #1** (0x384), **THDL Indirect Channel Data Register #2** (0x388) and the **THDL Indirect Channel Data Register #3** (0x38C) as described in section 9.5. In writing the **THDL Indirect Channel Data Register #1** (0x384), ensure the PROV bit is set, and ensure the FPTR[10:0] bits identify a block within the circular linked list of buffers of step 2.

5. Specify the THDL672 channel that is provisioned by writing the following register. Then poll the BUSY bit to ensure it is low before proceeding with step 6.

Bit	Register	Value
CHAN[9:0]	**THDL Indirect Channel Select** (0x380)	*TCC*
CRWB	**THDL Indirect Channel Select** (0x380)	0
BUSY	**THDL Indirect Channel Select** (0x380)	X

6. Poll the BUSY bit of the **TCAS Indirect Link and Time-slot Select** (0x400) register until it is zero. This ensures that a previous indirect RAM access has completed and that a new indirect RAM access can be started.

7. Specify the TCAS672 channel that is provisioned. Write the following register:

Bit	Register	Value
CHAN[9:0]	**TCAS Indirect Channel Data** (0x404)	*TCC*
PROV	**TCAS Indirect Channel Data** (0x404)	1

8. For a **channelised** link, specify the time-slots which are assigned for processing on this channel by writing the following register once for each time-slot that is assigned to the channel. Valid values for TSLOT[4:0] are 1 through 24 for a T1/J1 link, and 1 through 31 for an E1 link. For an **unchannelised** or **unframed** link, TSLOT[4:0] must only have the value 0, and this register is written just once. Each write must be followed by a read to determine whether the BUSY bit (bit15) is low, and to ensure that the indirect RAM has been updated.

Bit	Register	Value
TSLOT[4:0]	**TCAS Indirect Link and Time-slot Select** (0x400)	see above
LINK[6:0]	**TCAS Indirect Link and Time-slot Select** (0x400)	0 through 83 are valid
RWB	**TCAS Indirect Link and Time-slot Select** (0x400)	0
BUSY	**TCAS Indirect Link and Time-slot Select** (0x400)	X

9. Poll the BUSY bit of the **TAPI Indirect Channel Provisioning** (0x604) register until it is zero. This ensures that a previous indirect RAM access has completed and that a new indirect RAM access can be started.

10. Specify the channel burst length and enable channel provisioning by writing the following register. The BLEN[7:0] bits need to be set according to the XFER[3:0] value of the THDL672 as described in section 8.2.

Bit	Register	Value
BLEN[7:0]	**TAPI Indirect Channel Data Register** (0x608)	see above
PROV	**TAPI Indirect Channel Data Register** (0x608)	1

11. Specify the TAPI672 channel to provision. Write the following register fields, then poll the BUSY bit to ensure that the provisioning process has completed.

Bit	Register	Value
CHAN[9:0]	**TAPI Indirect Channel Provisioning** (0x604)	*TCC*
RWB	**TAPI Indirect Channel Provisioning** (0x604)	0
BUSY	**TAPI Indirect Channel Provisioning** (0x604)	X

12. Enable FREEDM-84A672 processing of the channel data stream to allow for graceful provisioning. Write the following bits:

Bit	Register	Value
DCHAN[9:0]	**TCAS Channel Disable** (0x410)	*TCC*
CHDIS	**TCAS Channel Disable** (0x410)	0

Warning:

- The TCAS Channel Disable bit (CHDIS) is only applicable to one channel at a time. In other words, the transmit channel provisioning procedure needs to be run once for each channel.

- The programmer must ensure that the channel has not been provisioned, or has been unprovisioned before doing the provisioning procedure. The reset procedure has the affect of unprovisioning all channels of the FREEDM-84A672.

- Continuous polling of a register in a tight loop involves multiple microprocessor memory read transactions and may have an adverse effect on the microprocessor bus bandwidth available for other activities. The recommended method of polling the BUSY bit is to read the register on expiration of a system timer, or after a number of CPU clock ticks. Recommended time intervals are in the range 0.1 msec through 1 msec.

- A Channel is not provisioned until the BUSY bit toggles low.

10.7 Unprovisioning a Channel

The unprovisioning procedure is normally applied to channels that are provisioned.

10.7.1 Receive Channel Unprovisioning

The steps to unprovision a receive channel *RCC*, where *0 ≤ RCC ≤ 671* are:

1. Disable FREEDM-84A672 processing of the channel's data stream to allow for graceful unprovisioning. Write the following bits:

Bit	Register	Value
DCHAN[9:0]	**RCAS Channel Disable** (0x10C)	*RCC*
CHDIS	**RCAS Channel Disable** (0x10C)	1

2. Poll the BUSY bit of the **RCAS Indirect Link and Time-slot Select** (0x100) register until it is zero. This ensures that a previous indirect RAM access has completed and that a new indirect RAM access can be started.

3. Specify the RCAS672 channel to unprovision by writing the following register:

Bit	Register	Value
CHAN[9:0]	**RCAS Indirect Channel Data** (0x104)	*RCC*
PROV	**RCAS Indirect Channel Data** (0x104)	0
CDLBEN	**RCAS Indirect Channel Data** (0x104)	X

4. For a **channelised** link, specify the time-slots which are unassigned on this channel by writing the following register once for each time-slot that is unassigned. Valid values for TSLOT[4:0] are 1 through 24 for a T1/J1 link, and 1 through 31 for an E1 link. For an **unchannelised** or **unframed** link, TSLOT[4:0] must only have the value 0, and this register is written just once. Each write must be followed by a read to determine whether the BUSY bit (bit15) is low, and to ensure that the indirect RAM has been updated.

Bit	Register	Value
TSLOT[4:0]	**RCAS Indirect Link and Time-slot Select** (0x100)	see above

Bit	Register	Value
LINK[6:0]	**RCAS Indirect Link and Time-slot Select** (0x100)	0 through 83 are valid
RWB	**RCAS Indirect Link and Time-slot Select** (0x100)	0
BUSY	**RCAS Indirect Link and Time-slot Select** (0x100)	X

5. Poll the BUSY bit of the **RHDL Indirect Channel Select** (0x200) register until it is zero. This ensures that a previous indirect RAM access has completed and that a new indirect RAM access can be started.

6. Read the RHDL672 channel data by writing the following register. Then poll the BUSY bit to ensure it is low before proceeding with step 7.

Bit	Register	Value
CHAN[9:0]	**RHDL Indirect Channel Select** (0x200)	*RCC*
CRWB	**RHDL Indirect Channel Select** (0x200)	1
BUSY	**RHDL Indirect Channel Select** (0x200)	X

7. Read the RHDL672 indirect channel data and check that the TAVAIL bit of the **RHDL Indirect Channel Data #1** (0x204) register is zero. This ensures that the last DMA transfer request for this channel has completed. If the TAVAIL bit is zero, proceed to step 8, otherwise, return to step 6.

8. Write the **RHDL Indirect Channel Data #1** (0x204) register with PROV modified to zero, while keeping the same FPTR[10:0] bits.

9. Specify the RHDL672 channel to unprovision by writing the following register. Then poll the BUSY bit to ensure that it is low before proceeding with step 10.

Bit	Register	Value
CHAN[9:0]	**RHDL Indirect Channel Select** (0x200)	*RCC*
CRWB	**RHDL Indirect Channel Select** (0x200)	0
BUSY	**RHDL Indirect Channel Select** (0x200)	X

10. Enable FREEDM-84A672 processing of the unprovisioned channel. Write the following bits:

Bit	Register	Value
DCHAN[9:0]	**RCAS Channel Disable** (0x10C)	*RCC*
CHDIS	**RCAS Channel Disable** (0x10C)	0

Warning:

- The RCAS Channel Disable bit (CHDIS) is only applicable to one channel at a time. In other words, the receive channel unprovisioning procedure needs to be run once for each channel.
- Continuous polling of a register in a tight loop involves multiple microprocessor memory read transactions and may have an adverse effect on the microprocessor bus bandwidth available for other activities. The recommended method of polling the BUSY bit is to read the register on expiration of a system timer, or after a number of CPU clock ticks. Recommended time intervals are in the range 0.1 msec through 1 msec.
- A Channel is not unprovisioned until the BUSY bit toggles low.

10.7.2 Transmit Channel Unprovisioning

The steps to unprovision a transmit channel *TCC*, where $0 \le TCC \le 671$ are:

1. Poll the BUSY bit of the **TAPI Indirect Channel Provisioning** (0x604) register until it is zero. This ensures that a previous indirect RAM access has completed and that a new indirect RAM access can be started.
2. Enable channel unprovisioning by writing the following register:

Bit	Register	Value
PROV	**TAPI Indirect Channel Data Register** (0x608)	0

3. Specify the TAPI672 channel to unprovision. Write the following register fields, then poll the BUSY bit to ensure that the unprovisioning process has completed.

Bit	Register	Value
CHAN[9:0]	**TAPI Indirect Channel Provisioning** (0x604)	*TCC*
RWB	**TAPI Indirect Channel Provisioning** (0x604)	0
BUSY	**TAPI Indirect Channel Provisioning** (0x604)	X

4. Disable FREEDM-84A672 processing of the channel's data stream to allow for graceful unprovisioning. Write the following bits:

Bit	Register	Value
DCHAN[9:0]	**TCAS Channel Disable** (0x410)	*TCC*
CHDIS	**TCAS Channel Disable** (0x410)	1

5. Poll the BUSY bit of the **TCAS Indirect Link and Time-slot Select** (0x400) register until it is zero. This ensures that a previous indirect RAM access has completed and that a new indirect RAM access can be started.

6. Specify the TCAS672 channel to be unprovisioned. Write the following register:

Bit	Register	Value
CHAN[9:0]	**TCAS Indirect Channel Data** (0x404)	*TCC*
PROV	**TCAS Indirect Channel Data** (0x404)	0

7. For a **channelised** link, specify the time-slots which are unassigned for processing on this channel by writing the following register once for each time-slot that is unassigned. Valid values for TSLOT[4:0] are 1 through 24 for a T1/J1 link, and 1 through 31 for an E1 link. For an **unchannelised** or **unframed** link, TSLOT[4:0] must only have the value 0, and this register is written just once. Each write must be followed by a read to determine whether the BUSY bit (bit15) is low, and to ensure that the indirect RAM has been updated.

Bit	Register	Value
TSLOT[4:0]	**TCAS Indirect Link and Time-slot Select** (0x400)	see above
LINK[6:0]	**TCAS Indirect Link and Time-slot Select** (0x400)	0 through 83 are valid
RWB	**TCAS Indirect Link and Time-slot Select** (0x400)	0
BUSY	**TCAS Indirect Link and Time-slot Select** (0x400)	X

8. Poll the BUSY bit of the **THDL Indirect Channel Select** (0x380) register until it is zero. This ensures that a previous indirect RAM access has completed and that a new indirect RAM access can be started.

9. Read the THDL672 channel data by writing the following register. Then poll the BUSY bit to ensure that it is low before proceeding with step 9.

Bit	Register	Value
CHAN[9:0]	**THDL Indirect Channel Select** (0x380)	*TCC*
CRWB	**THDL Indirect Channel Select** (0x380)	1
BUSY	**THDL Indirect Channel Select** (0x380)	X

10. Read the **THDL Indirect Channel Data #1** (0x384) register. Then write this register with PROV modified to zero, while keeping the same FPTR[10:0] bits.

11. Specify the THDL672 channel to unprovision by writing the following register. Then poll the BUSY bit to ensure that it is low before proceeding with step 11.

Bit	Register	Value
CHAN[9:0]	**THDL Indirect Channel Select** (0x380)	*TCC*
CRWB	**THDL Indirect Channel Select** (0x380)	0
BUSY	**THDL Indirect Channel Select** (0x380)	X

12. Enable FREEDM-84A672 processing of the channel. Write the following bits:

Bit	Register	Value
DCHAN[9:0]	**TCAS Channel Disable** (0x410)	*TCC*
CHDIS	**TCAS Channel Disable** (0x410)	0

Warning:

- The TCAS Channel Disable bit (CHDIS) is only applicable to one channel at a time. In other words, the transmit channel unprovisioning procedure needs to be run once for each channel.

- Continuous polling of a register in a tight loop involves multiple microprocessor memory read transactions and may have an adverse effect on the microprocessor bus bandwidth available for other activities. The recommended method of polling the BUSY bit is to read the register on expiration of a system timer, or after a number of CPU clock ticks. Recommended time intervals are in the range 0.1 msec through 1 msec.

- A Channel is not unprovisioned until the BUSY bit toggles low.

10.8 Receive Sequence

The software is not required to receive packets when interfacing to the RAPI672. Data transfer functions for the FREEDM-84A672 are performed by an external controller.

In the receive direction, the external controller transfers partial packets out of the internal 32 Kbyte partial packet buffer RAM in the RHDL672, across the receive APPI bus, and into host packet memory. Please refer to the Longform Datasheet[1] for detailed information on the operation and timing of the receive APPI.

10.9 Transmit Sequence

The software is not required to transmit packets when interfacing to the TAPI672. Data transfer functions for the FREEDM-84A672 are performed by an external controller.

In the transmit direction, the external controller provides packets to transmit using the transmit APPI. For each provisioned HDLC channel, an external controller transfers partial packets across the transfer APPI, and into the internal 32 Kbyte partial packet buffer RAM in the THDL672. Please refer to the Longform Datasheet[1] for detailed information on the operation and timing of the transmit APPI.

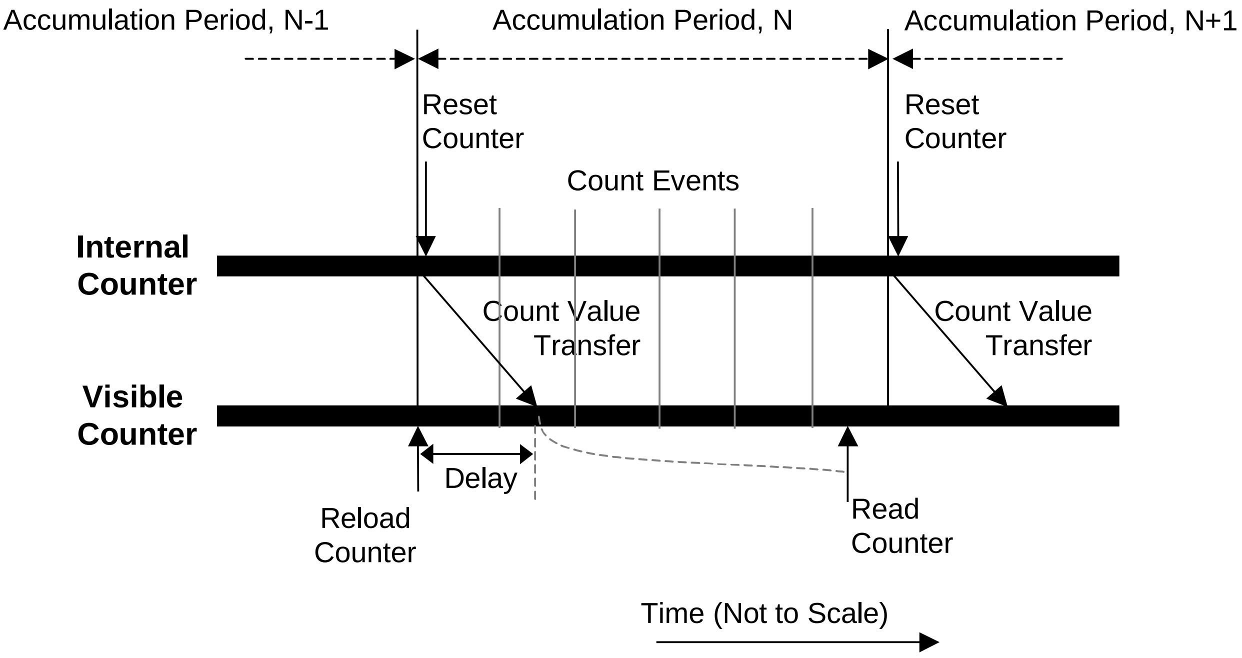
10.10 Performance Counters

The FREEDM-84A672 provides four count registers within the Normal Mode Register Space. These are as follows:

Bit	Register
OF[15:0]	**PMON Receive FIFO Overflow Count** (0x504)
UF[15:0]	**PMON Transmit FIFO Underflow Count** (0x508)
C1[15:0]	**PMON Configurable Count #1** (0x50C)
C2[15:0]	**PMON Configurable Count #2** (0x510)

The software must poll these counters to prevent overflow. Figure 5 illustrates the sequence of events when the counters are polled. The **PMON Status** (0x500) register provides status bits which indicate whether any of the four internal holding counters has overflowed.

Figure 5 – Event Sequence for Polling of Counters



The software initiates a counter reload by writing to the **FREEDM-84A672 Master Clock/Frame Pulse Activity Monitor and Accumulation Trigger** (0x00C) register. There is a small delay to transfer data from internal counters to the visible counters. The recommended polling strategy is to read the counters first before initiating a reload. Using this strategy, the transfer latency can be ignored.

Counters are normally configured during initialization. The first configurable count register is assigned by setting one of the following register bits, while setting all other bits to zero:

Bit	Register
RSPE1EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)

Bit	Register
RFCSE1EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)
RABRT1EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)
RLENE1EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)
RP1EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)
TABRT1EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)
TP1EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)

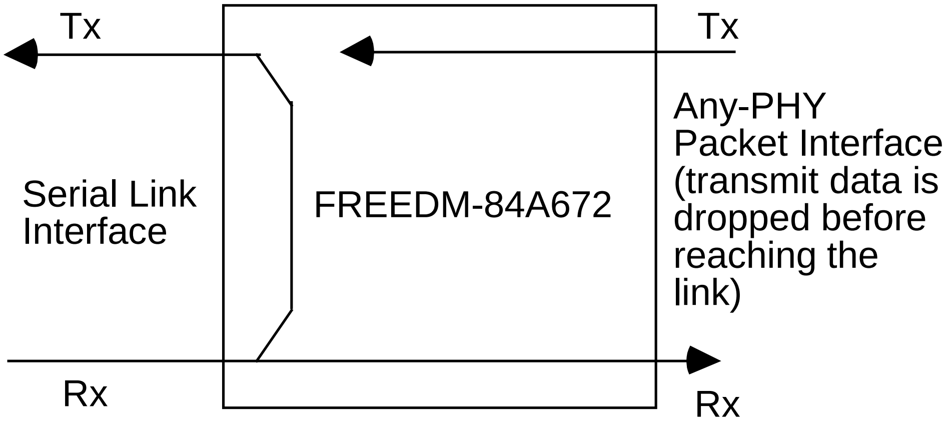
The second configurable count register is assigned by setting one of the following register bits, while setting all other bits to zero:

Bit	Register
RSPE2EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)
RFCSE2EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)
RABRT2EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)
RLENE2EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)
RP2EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)
TABRT2EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)
TP2EN	**FREEDM-84A672 Master Performance Monitor Control** (0x024)

10.11 Line Loopback

Serial links of the RCAS672/TCAS672 can be placed in line loopback. In this configuration, the data on the receive link output by the SBI PISO blocks is looped back to the transmit link input of the SBI SIPO blocks as illustrated in Figure 6.

Figure 6 – Line Loopback



Serial links can be placed in line loopback by setting the appropriate bit within one of the following registers. There are 84 bits corresponding to the 84 serial links.

Bit	Register
LLBEN[15:0]	**FREEDM-84A672 Master Line Loopback #1** (0x030)
LLBEN[31:16]	**FREEDM-84A672 Master Line Loopback #2** (0x034)
LLBEN[47:32]	**FREEDM-84A672 Master Line Loopback #3** (0x038)
LLBEN[63:48]	**FREEDM-84A672 Master Line Loopback #4** (0x03C)
LLBEN[79:64]	**FREEDM-84A672 Master Line Loopback #5** (0x040)
LLBEN[83:80]	**FREEDM-84A672 Master Line Loopback #6** (0x044)

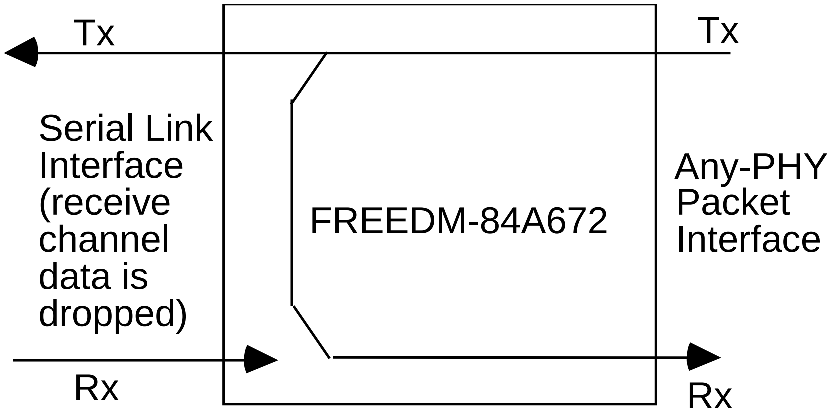
Note: The software should unprovision channels associated with the link that is placed in line loopback mode before placing the link in line loopback. This will prevent the data stream at the serial link from passing through the FREEDM-84A672 to the receive APPI.

10.12 Diagnostic Loopback

Each channel of the FREEDM-84A672 can be placed in a diagnostic loopback mode. In this configuration, the transmit data stream is looped back to the receive data stream as illustrated in Figure 7. The pair of transmit/receive channels is configured in diagnostic loopback mode by provisioning both the transmit and the receive channels as specified in section 10.6, except with the CDLBEN bit set high within the **RCAS Indirect Channel Data** (0x104) register.

In diagnostic loopback mode, the transmit channel data is looped back as well as driven onto the transmit serial link. The channel data from the receive serial link is dropped. The bit timing for the diagnostic loopback mode is generated internally. This clock is derived from REFCLK, C1FP and FASTCLK (if the SPE is configured to support DS-3 links) so these inputs should be active.

Figure 7 – Diagnostic Loopback



APPENDIX A – REGISTER LEVEL CHANGES

The following table is a comparison of the normal mode registers at the register level among the FREEDM-32, the FREEDM-32A672 and the FREEDM-84A672. Registers in bold indicate differences at the register level among the members of the FREEDM family listed in the table. Table entries that are "N/A" indicate that the register is not applicable in the corresponding FREEDM device. Please see Appendix E for differences at the bit level for the normal mode registers.

Register	FREEDM-32 PCI Offset	FREEDM-32A672 Address	FREEDM-84A672 Address
FREEDM-x Master Reset	0x000	0x000	0x000
FREEDM-x Master Interrupt Enable	0x004	0x004	0x004
FREEDM-x Master Interrupt Status	0x008	0x008	0x008
FREEDM-x Master Clock/Frame Pulse/BERT Activity Monitor and Accumulation Trigger	0x00C	0x00C	0x00C
FREEDM-x Master Link Activity Monitor	**0x010**	**0x010**	**N/A**
FREEDM-x Master Line Loopback #1	**0x014**	**0x014**	**0x030**
FREEDM-x Master Line Loopback #2	**0x018**	**0x018**	**0x034**
Reserved	**0x01C**	**N/A**	**0x010 – 0x020**
FREEDM-x Reserved	**N/A**	**0x01C**	**N/A**
FREEDM-x Master BERT Control	**0x020**	**0x020**	**N/A**
FREEDM-x Master Performance Monitor Control	0x024	0x024	0x024
FREEDM-x Master SBI Interrupt Enable	**N/A**	**N/A**	**0x028**

Register	FREEDM-32 PCI Offset	FREEDM-32A672 Address	FREEDM-84A672 Address
FREEDM-x Master SBI Interrupt Status	**N/A**	**N/A**	**0x02C**
FREEDM-x Master Line Loopback #3	**N/A**	**N/A**	**0x038**
FREEDM-x Master Line Loopback #4	**N/A**	**N/A**	**0x03C**
FREEDM-x Master Line Loopback #5	**N/A**	**N/A**	**0x040**
FREEDM-x Master Line Loopback #6	**N/A**	**N/A**	**0x044**
FREEDM-x SBI DROP BUS Master Configuration	**N/A**	**N/A**	**0x048**
FREEDM-x SBI ADD BUS Master Configuration	**N/A**	**N/A**	**0x04C**
Reserved	**0x028 – 0x03C**	**0x028 – 0x0FC**	**0x050 – 0x0FC**
GPIC Control	**0x040**	**N/A**	**N/A**
GPIC Reserved	**0x044 – 0x07C**	**N/A**	**N/A**
Reserved	**0x080 – 0x0FC**	**N/A**	**N/A**
RCAS Indirect Channel and Time-slot Select	0x100	0x100	0x100
RCAS Indirect Channel Data	0x104	0x104	0x104
RCAS Framing Bit Threshold	**0x108**	**0x108**	**N/A**
RCAS Reserved	**N/A**	**N/A**	**0x108**
RCAS Channel Disable	0x10C	0x10C	0x10C
RCAS Reserved	**0x110 – 0x17C**	**0x110 – 0x17C**	**0x110 – 0x13C**

Register	FREEDM-32 PCI Offset	FREEDM-32A672 Address	FREEDM-84A672 Address
RCAS SBI SPE1 Configuration Register #1	**N/A**	**N/A**	**0x140**
RCAS SBI SPE1 Configuration Register #2	**N/A**	**N/A**	**0x144**
RCAS SBI SPE2 Configuration Register #1	**N/A**	**N/A**	**0x148**
RCAS SBI SPE2 Configuration Register #2	**N/A**	**N/A**	**0x14C**
RCAS SBI SPE3 Configuration Register #1	**N/A**	**N/A**	**0x150**
RCAS SBI SPE3 Configuration Register #2	**N/A**	**N/A**	**0x154**
RCAS Reserved	**N/A**	**N/A**	**0x158 – 0x17C**
RCAS Links #0 through #2 Configuration	0x180 – 0x188	0x180 – 0x188	0x180 – 0x188
RCAS Links #3 through #31 Configuration	**0x18C – 0x1FC**	**0x18C – 0x1FC**	**N/A**
RCAS Reserved	**N/A**	**N/A**	**0x18C – 0x1FC**
RHDL Indirect Channel Select	0x200	0x200	0x200
RHDL Indirect Channel Data Register #1	0x204	0x204	0x204
RHDL Indirect Channel Data Register #2	0x208	0x208	0x208
RHDL Reserved	0x20C	0x20C	0x20C
RHDL Indirect Block Select	0x210	0x210	0x210
RHDL Indirect Block Data Register	0x214	0x214	0x214
RHDL Reserved	0x218 – 0x21C	0x218 – 0x21C	0x218 – 0x21C
RHDL Configuration	0x220	0x220	0x220

Register	FREEDM-32 PCI Offset	FREEDM-32A672 Address	FREEDM-84A672 Address
RHDL Maximum Packet Length	0x224	0x224	0x224
RHDL Reserved	0x228 – 0x23C	0x228 – 0x23C	0x228 – 0x23C
Reserved	**0x240 – 0x27C**	**0x240 – 0x37C**	**0x240 – 0x37C**
RMAC Control	**0x280**	**N/A**	**N/A**
RMAC Indirect Channel Provisioning	**0x284**	**N/A**	**N/A**
RMAC Packet Descriptor Table Base LSW	**0x288**	**N/A**	**N/A**
RMAC Packet Descriptor Table Base MSW	**0x28C**	**N/A**	**N/A**
RMAC Queue Base LSW	**0x290**	**N/A**	**N/A**
RMAC Queue Base MSW	**0x294**	**N/A**	**N/A**
RMAC Packet Descriptor Reference Large Buffer Free Queue Start	**0x298**	**N/A**	**N/A**
RMAC Packet Descriptor Reference Large Buffer Free Queue Write	**0x29C**	**N/A**	**N/A**
RMAC Packet Descriptor Reference Large Buffer Free Queue Read	**0x2A0**	**N/A**	**N/A**
RMAC Packet Descriptor Reference Large Buffer Free Queue End	**0x2A4**	**N/A**	**N/A**
RMAC Packet Descriptor Reference Small Buffer Free Queue Start	**0x2A8**	**N/A**	**N/A**
RMAC Packet Descriptor Reference Small Buffer Free Queue Write	**0x2AC**	**N/A**	**N/A**

Register	FREEDM-32 PCI Offset	FREEDM-32A672 Address	FREEDM-84A672 Address
RMAC Packet Descriptor Reference Small Buffer Free Queue Read	0x2B0	N/A	N/A
RMAC Packet Descriptor Reference Small Buffer Free Queue End	0x2B4	N/A	N/A
RMAC Packet Descriptor Reference Ready Queue Start	0x2B8	N/A	N/A
RMAC Packet Descriptor Reference Ready Queue Write	0x2BC	N/A	N/A
RMAC Packet Descriptor Reference Ready Queue Read	0x2C0	N/A	N/A
RMAC Packet Descriptor Reference Ready Queue End	0x2C4	N/A	N/A
RMAC Reserved	0x2C8 – 0x2FC	N/A	N/A
TMAC Control	0x300	N/A	N/A
TMAC Indirect Channel Provisioning	0x304	N/A	N/A
TMAC Descriptor Table Base LSW	0x308	N/A	N/A
TMAC Descriptor Table Base MSW	0x30C	N/A	N/A
TMAC Queue Base LSW	0x310	N/A	N/A
TMAC Queue Base MSW	0x314	N/A	N/A
TMAC Descriptor Reference Free Queue Start	0x318	N/A	N/A

Register	FREEDM-32 PCI Offset	FREEDM-32A672 Address	FREEDM-84A672 Address
TMAC Descriptor Reference Free Queue Write	**0x31C**	**N/A**	**N/A**
TMAC Descriptor Reference Free Queue Read	**0x320**	**N/A**	**N/A**
TMAC Descriptor Reference Free Queue End	**0x324**	**N/A**	**N/A**
TMAC Descriptor Reference Ready Queue Start	**0x328**	**N/A**	**N/A**
TMAC Descriptor Reference Ready Queue Write	**0x32C**	**N/A**	**N/A**
TMAC Descriptor Reference Ready Queue Read	**0x330**	**N/A**	**N/A**
TMAC Descriptor Reference Ready Queue End	**0x334**	**N/A**	**N/A**
TMAC Reserved	**0x338 – 0x37C**	**N/A**	**N/A**
THDL Indirect Channel Select	0x380	0x380	0x380
THDL Indirect Channel Data #1	0x384	0x384	0x384
THDL Indirect Channel Data #2	0x388	0x388	0x388
THDL Indirect Channel Data #3	0x38C	0x38C	0x38C
THDL Reserved	0x390 – 0x39C	0x390 – 0x39C	0x390 – 0x39C
THDL Indirect Block Select	0x3A0	0x3A0	0x3A0

Register	FREEDM-32 PCI Offset	FREEDM-32A672 Address	FREEDM-84A672 Address
THDL Indirect Block Data	0x3A4	0x3A4	0x3A4
THDL Reserved	0x3A8 – 0x3AC	0x3A8 – 0x3AC	0x3A8 – 0x3AC
THDL Configuration	0x3B0	0x3B0	0x3B0
THDL Reserved	0x3B4 – 0x3BC	0x3B4 – 0x3BC	0x3B4 – 0x3BC
Reserved	0x3C0 – 0x3FC	0x3C0 – 0x3FC	0x3C0 – 0x3FC
TCAS Indirect Channel and Time-slot Select	0x400	0x400	0x400
TCAS Indirect Channel Data	0x404	0x404	0x404
TCAS Framing Bit Threshold	**0x408**	**0x408**	**N/A**
TCAS Reserved	**N/A**	**N/A**	**0x408**
TCAS Idle Time-slot Fill Data	0x40C	0x40C	0x40C
TCAS Channel Disable	0x410	0x410	0x410
TCAS Reserved	**0x414 – 0x47C**	**0x414 – 0x47C**	**0x414 – 0x43C**
TCAS SBI SPE1 Configuration Register #1	**N/A**	**N/A**	**0x440**
TCAS SBI SPE1 Configuration Register #2	**N/A**	**N/A**	**0x444**
TCAS SBI SPE2 Configuration Register #1	**N/A**	**N/A**	**0x448**
TCAS SBI SPE2 Configuration Register #2	**N/A**	**N/A**	**0x44C**
TCAS SBI SPE3 Configuration Register #1	**N/A**	**N/A**	**0x450**
TCAS SBI SPE3 Configuration Register #2	**N/A**	**N/A**	**0x454**
TCAS Reserved	**N/A**	**N/A**	**0x458 – 0x47C**

Register	FREEDM-32 PCI Offset	FREEDM-32A672 Address	FREEDM-84A672 Address
TCAS Links #0 through #2 Configuration	0x480 – 0x488	0x480 – 0x488	0x480 – 0x488
TCAS Links #3 through #31 Configuration	**0x48C – 0x4FC**	**0x48C – 0x4FC**	**N/A**
TCAS Reserved	**N/A**	**N/A**	**0x48C – 0x4FC**
PMON Status	0x500	0x500	0x500
PMON Receive FIFO Overflow Count	0x504	0x504	0x504
PMON Transmit FIFO Underflow Count	0x508	0x508	0x508
PMON Configurable Count #1	0x50C	0x50C	0x50C
PMON Configurable Count #2	0x510	0x510	0x510
PMON Reserved	0x514 – 0x51C	0x514 – 0x51C	0x514 – 0x51C
Reserved	**0x520 – 0x7FC**	**0x520 – 0x57C**	**0x520 – 0x57C**
RAPI Control	**N/A**	**0x580**	**0x580**
RAPI Reserved	**N/A**	**0x584 – 0x5BC**	**0x584 – 0x5BC**
Reserved	**N/A**	**0x5C0 – 0x5FC**	**N/A**
SBI EXTRACT Control	**N/A**	**N/A**	**0x5C0**
SBI EXTRACT Reserved	**N/A**	**N/A**	**0x5C4 - 0x5C8**
SBI EXTRACT Tributary RAM Indirect Access Address	**N/A**	**N/A**	**0x5CC**
SBI EXTRACT Tributary RAM Indirect Access Control	**N/A**	**N/A**	**0x5D0**
SBI EXTRACT Reserved	**N/A**	**N/A**	**0x5D4**
SBI EXTRACT Tributary RAM Indirect Access Data	**N/A**	**N/A**	**0x5D8**

Register	FREEDM-32 PCI Offset	FREEDM-32A672 Address	FREEDM-84A672 Address
SBI EXTRACT Parity Error Interrupt Reason	N/A	N/A	0x5DC
SBI EXTRACT Reserved	N/A	N/A	0x5E0 – 0x5FC
TAPI Control	N/A	0x600	0x600
TAPI Indirect Channel Provisioning	N/A	0x604	0x604
TAPI Indirect Channel Data Register	N/A	0x608	0x608
TAPI Reserved	N/A	0x60C – 0x63C	0x60C – 0x63C
Reserved	N/A	0x640 – 0x7FC	0x640 – 0x67C
SBI INSERT Control	N/A	N/A	0X680
SBI INSERT Reserved	N/A	N/A	0x684 – 0x688
SBI INSERT Tributary RAM Indirect Access Address	N/A	N/A	0x68C
SBI INSERT Tributary RAM Indirect Access Control	N/A	N/A	0x690
SBI INSERT Reserved	N/A	N/A	0x694
SBI INSERT Tributary RAM Indirect Access Data	N/A	N/A	0x698
SBI INSERT Reserved	N/A	N/A	0x69C – 0x6FC
Reserved	N/A	N/A	0x700 – 0x7FC

Note: There are no PCI Configuration registers in the FREEDM-84A672 device.

APPENDIX B – NEW NORMAL MODE REGISTERS

The following registers are new for the FREEDM-84A672. The new registers are used to configure and control the SBI interface, the SBI Extracter and Inserter, the new RAPI672 and TAPI672 blocks, and line loopback for the increased number of links. Please refer to the Longform Datasheet[1] for detailed descriptions of these registers.

FREEDM-84A672 Address	Register
0x028	FREEDM-84A672 Master SBI Interrupt Enable
0x02C	FREEDM-84A672 Master SBI Interrupt Status
0x038	FREEDM-84A672 Master Line Loopback #3
0x03C	FREEDM-84A672 Master Line Loopback #4
0x040	FREEDM-84A672 Master Line Loopback #5
0x044	FREEDM-84A672 Master Line Loopback #6
0x048	FREEDM-84A672 SBI DROP BUS Master Configuration
0x04C	FREEDM-84A672 SBI ADD BUS Master Configuration
0x140	RCAS SBI SPE1 Configuration Register #1
0x144	RCAS SBI SPE1 Configuration Register #2
0x148	RCAS SBI SPE2 Configuration Register #1
0x14C	RCAS SBI SPE2 Configuration Register #2
0x150	RCAS SBI SPE3 Configuration Register #1
0x154	RCAS SBI SPE3 Configuration Register #2
0x440	TCAS SBI SPE1 Configuration Register #1
0x444	TCAS SBI SPE1 Configuration Register #2
0x448	TCAS SBI SPE2 Configuration Register #1
0x44C	TCAS SBI SPE2 Configuration Register #2
0x450	TCAS SBI SPE3 Configuration Register #1
0x454	TCAS SBI SPE3 Configuration Register #2
0x580	RAPI Control

FREEDM-84A672 Address	Register
0x584 – 0x5BC	RAPI Reserved
0x5C0	SBI EXTRACT Control
0x5C4 - 0x5C8	SBI EXTRACT Reserved
0x5CC	SBI EXTRACT Tributary RAM Indirect Access Address
0x5D0	SBI EXTRACT Tributary RAM Indirect Access Control
0x5D4	SBI EXTRACT Reserved
0x5D8	SBI EXTRACT Tributary RAM Indirect Access Data
0x5DC	SBI EXTRACT Parity Error Interrupt Reason
0x5E0 - 0x5FC	SBI EXTRACT Reserved
0x600	TAPI Control
0x604	TAPI Indirect Channel Provisioning
0x608	TAPI Indirect Data Register
0x60C – 0x63C	TAPI Reserved
0x680	SBI INSERT Control
0x684 - 0x688	SBI INSERT Reserved
0x68C	SBI INSERT Tributary RAM Indirect Access Address
0x690	SBI INSERT Tributary RAM Indirect Access Control
0x694	SBI INSERT Reserved
0x698	SBI INSERT Tributary RAM Indirect Access Data
0x69C - 0x6FC	SBI INSERT Reserved

APPENDIX C – NON-APPLICABLE NORMAL MODE REGISTERS

The following FREEDM-32 registers are no longer applicable in the FREEDM-84A672.

FREEDM-32 PCI Offset	Register
0x010	FREEDM-32 Master Link Activity Monitor
0x020	FREEDM-32 Master BERT Control
0x040	GPIC Control
0x044 - 0x07C	GPIC Reserved
0x108	RCAS Framing Bit Threshold
0x18C – 0x1FC	RCAS Link #3 through #31 Configuration
0x280	RMAC Control
0x284	RMAC Indirect Channel Provisioning
0x288	RMAC Packet Descriptor Table Base LSW
0x28C	RMAC Packet Descriptor Table Base MSW
0x290	RMAC Queue Base LSW
0x294	RMAC Queue Base MSW
0x298	RMAC Packet Descriptor Reference Large Buffer Free Queue Start
0x29C	RMAC Packet Descriptor Reference Large Buffer Free Queue Write
0x2A0	RMAC Packet Descriptor Reference Large Buffer Free Queue Read
0x2A4	RMAC Packet Descriptor Reference Large Buffer Free Queue End
0x2A8	RMAC Packet Descriptor Reference Small Buffer Free Queue Start
0x2AC	RMAC Packet Descriptor Reference Small Buffer Free Queue Write

FREEDM-32 PCI Offset	Register
0x2B0	RMAC Packet Descriptor Reference Small Buffer Free Queue Read
0x2B4	RMAC Packet Descriptor Reference Small Buffer Free Queue End
0x2B8	RMAC Packet Descriptor Reference Ready Queue Start
0x2BC	RMAC Packet Descriptor Reference Ready Queue Write
0x2C0	RMAC Packet Descriptor Reference Ready Queue Read
0x2C4	RMAC Packet Descriptor Reference Ready Queue End
0x2C8 - 0x2FC	RMAC Reserved
0x300	TMAC Control
0x304	TMAC Indirect Channel Provisioning
0x308	TMAC Descriptor Table Base LSW
0x30C	TMAC Descriptor Table Base MSW
0x310	TMAC Queue Base LSW
0x314	TMAC Queue Base MSW
0x318	TMAC Descriptor Reference Free Queue Start
0x31C	TMAC Descriptor Reference Free Queue Write
0x320	TMAC Descriptor Reference Free Queue Read
0x324	TMAC Descriptor Reference Free Queue End
0x328	TMAC Descriptor Reference Ready Queue Start
0x32C	TMAC Descriptor Reference Ready Queue Write
0x330	TMAC Descriptor Reference Ready Queue Read
0x334	TMAC Descriptor Reference Ready Queue End
0x338 - 0x37C	TMAC Reserved
0x408	TCAS Framing Bit Threshold
0x48C – 0x4FC	TCAS Link #3 through #31 Configuration

APPENDIX D – MOVED NORMAL MODE REGISTERS

The following registers have been moved when comparing its FREEDM-32 location to its FREEDM-84A672 location.

Register	FREEDM-32 PCI Offset	FREEDM-84A672 Address
FREEDM-x Master Line Loopback #1	0x014	0x030
FREEDM-x Master Line Loopback #2	0x018	0x034
RCAS Reserved	0x110 – 0x17C	0x108 0x110-0x13C 0x158 – 0x17C 0x18C – 0x1FC
TCAS Reserved	0x414 – 0x47C	0x408 0x414 – 0x43C 0x458 – 0x47C 0x48C – 0x4FC

APPENDIX E – NORMAL MODE REGISTER BIT CHANGES

The following normal mode registers have changed at the bit level from the FREEDM-32 to the FREEDM-84A672. Unless specified, register names, locations and comments refer to FREEDM-84A672 registers.

Register 0x000 : FREEDM-84A672 Master Reset

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
15	Reset	0	Reset	0	Now also forces the APPI outputs tristate.
11	TYPE[3]	0	Unused	X	New Device Type bits allow software to identify the device as the FREEDM-84A672 member of the FREEDM family of products.
10	TYPE[2]	1	Unused	X	
9	TYPE[1]	0	Unused	X	
8	TYPE[0]	1	Unused	X	
7	ID[7]	0	Unused	X	New Device ID bits allow software to identify the version level of FREEDM-84A672.
6	ID[6]	0	Unused	X	
5	ID[5]	0	Unused	X	
4	ID[4]	0	Unused	X	
3	ID[3]	0	Unused	X	
2	ID[2]	0	Unused	X	
1	ID[1]	0	Unused	X	
0	ID[0]	0	Unused	X	

Register 0x004 : FREEDM-84A672 Master Interrupt Enable

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
14	TFOVRE	0	IOCE	0	SERRE, PERRE, IOCE, and all queue-related interrupt enable bits are not used because queues and the PCI bus are not used for the FREEDM-84A672. New interrupt enable bits are TPRTYE, TUNPVE, and TFOVRE.
13	TUNPVE	0	TDFQEE	0	
12	TPRTYE	0	TDQRDYE	0	
11	Unused	X	TDQFE	0	
10	Unused	X	RPDRQEE	0	
9	Unused	X	RPDFQEE	0	
8	Unused	X	RPQRDYE	0	
7	Unused	X	RPQLFE	0	
6	Unused	X	RPQSFE	0	
1	Unused	X	PERRE	0	
0	Unused	X	SERRE	0	

Register 0x008 : FREEDM-84A672 Master Interrupt Status

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
14	TFOVRI	0	IOCI	0	SERRI, PERRI, IOCI, and all queue-related interrupt status bits are not used because queues and the PCI bus are not used for the FREEDM-84A672. New interrupt status bits are TPRTYI, TUNPVI, and TFOVRI.
13	TUNPVI	0	TDFQEI	0	
12	TPRTYI	0	TDQRDYI	0	
11	Unused	X	TDQFI	0	
10	Unused	X	RPDRQEI	0	
9	Unused	X	RPDFQEI	0	
8	Unused	X	RPQRDYI	0	
7	Unused	X	RPQLFI	0	
6	Unused	X	RPQSFI	0	
1	Unused	X	PERRI	0	
0	Unused	X	SERRI	0	

Register 0x00C : FREEDM-84A672 Master Clock / Frame Pulse Activity Monitor and Accumulation Trigger

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
13	TXCLKA	X	Unused	X	Any-PHY transmit clock active bit.
12	RXCLKA	X	Unused	X	Any-PHY receive clock active bit.
3	C1FPA	X	Unused	X	SBI frame pulse active bit.
2	FASTCLKA	X	Unused	X	SBI fast clock active bit.
1	REFCLKA	X	TBDA	X	SBI reference clock active bit. Transmit BERT data active bit is not applicable in the FREEDM-84A672.

Register 0x100 : RCAS Indirect Link and Time-slot Select

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
12	LINK[6]	0	LINK[4]	0	Increase in the number of receive links from 32 to 84.
11	LINK[5]	0	LINK[3]	0	
10	LINK[4]	0	LINK[2]	0	
9	LINK[3]	0	LINK[1]	0	
8	LINK[2]	0	LINK[0]	0	
7	LINK[1]	0	Unused	X	
6	LINK[0]	0	Unused	X	

Register 0x104 : RCAS Indirect Channel Data

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
15	CDLBEN	0	Unused	X	
14	PROV	0	Unused	X	
9	CHAN[9]	0	CDLBEN	0	Increase in size of CHAN from 7 bits to 10 bits as the result of increase in HDLC channels from 128 to 672.
8	CHAN[8]	0	PROV	0	
7	CHAN[7]	0	Unused	X	

Register 0x10C : RCAS Channel Disable

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
9	DCHAN[9]	0	Unused	X	Increase in size of DCHAN from 7 bits to 10 bits as the result of increase in HDLC channels from 128 to 672.
8	DCHAN[8]	0	Unused	X	
7	DCHAN[7]	0	Unused	X	

Registers 0x180 – 0x188 : RCAS Links #0 to #2 Configuration

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
4	Reserved	0	Unused	X	The reserved bits must be set low for correct operation of the FREEDM-84A672.
2	Reserved	0	BSYNC	0	
1	Reserved	0	E1	0	
0	Reserved	0	CEN	0	CEN, E1 and BSYNC are not applicable in the FREEDM-84A672.

Register 0x200 : RHDL Indirect Channel Select

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
9	CHAN[9]	0	Unused	X	Increase in size of CHAN from 7 bits to 10 bits as the result of increase in HDLC channels from 128 to 672.
8	CHAN[8]	0	Unused	X	
7	CHAN[7]	0	Unused	X	

Register 0x204 : RHDL Indirect Channel Data #1

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
14	STRIP	0	CRC[1]	0	CRC[1] moved to bit 11 of Register 0x208 of the FREEDM-84A672.
13	DELIN	0	CRC[0]	0	CRC[0] moved to bit 10 of Register 0x208 of the FREEDM-84A672.
12	TAVAIL	X	STRIP	0	
11	Reserved	X	DELIN	0	Reserved bit must be set low for correct operation of the FREEDM-84A672.
10	FPTR[10]	X	TAVAIL	X	Increase in size of FPTR from 9 bits to 11 bits as the result of increase in addressable descriptors from 512 to 2048.
9	FPTR[9]	X	Unused	X	

Register 0x208 : RHDL Indirect Channel Data #2

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
11	CRC[1]	0	Unused	X	CRC[1] moved from bit 14 of Register 0x204 of the FREEDM-32.
10	CRC[0]	0	Unused	X	CRC[0] moved from bit 13 of Register 0x204 of the FREEDM-32.
3	XFER[3]	0	Unused	X	XFER increased from 3 bits to 4 bits to support larger data transfers.

Register 0x210 : RHDL Indirect Block Select

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
11	Reserved	X	Unused	X	Reserved bit must be set low for correct operation of FREEDM-84A672.
10	BLOCK[10]	X	Unused	X	BLOCK increased from 9 bits to 11 bits as the result of increase in addressable blocks from 512 to 2048.
9	BLOCK[9]	X	Unused	X	

Register 0x214 : RHDL Indirect Block Data

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
11	Reserved	X	Unused	X	Reserved bit must be set low for correct operation of FREEDM-84A672.
10	BPTR[10]	X	Unused	X	BPTR increased from 9 bits to 11 bits as the result of increase in addressable blocks from 512 to 2048.
9	BPTR[9]	X	Unused	X	

Register 0x220 : RHDL Configuration

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
2	Unused	X	Reserved[2]	1	These reserved bits are no longer used in the FREEDM-84A672.
1	Unused	X	Reserved[1]	1	
0	Unused	X	Reserved[0]	1	

Register 0x380 : THDL Indirect Channel Select

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
9	CHAN[9]	0	Unused	X	Increase in size of CHAN from 7 bits to 10 bits as the result of increase in HDLC channels from 128 to 672.
8	CHAN[8]	0	Unused	X	
7	CHAN[7]	0	Unused	X	

Register 0x384 : THDL Indirect Channel Data #1

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
12	DELIN	X	IDLE	0	IDLE moved to bit 14 of Register 0x38C of the FREEDM-84A672.
11	Reserved	X	DELIN	0	Reserved bit must be set low for correct operation of FREEDM-84A672.
10	FPTR[10]	0	Unused	X	Increase in size of FPTR from 9 bits to 11 bits as the result of increase in addressable descriptors from 512 to 2048.
9	FPTR[9]	0	Unused	X	

Register 0x388 : THDL Indirect Channel Data #2

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
14	Reserved	0	PRIORITYB	0	Reserved bit must be set low for correct operation of the FREEDM-84A672. PRIORITYB is not used in the FREEDM-84A672.
11	Reserved	0	Unused	X	
10	FLEN[10]	0	Unused	X	Increase in size of FLEN from 9 bits to 11 bits as the result of increase in addressable descriptors from 512 to 2048.
9	FLEN[9]	0	Unused	X	

Register 0x38C : THDL Indirect Channel Data #3

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
14	IDLE	0	Unused	X	IDLE moved from bit 12 of Register 0x384 of the FREEDM-32.
3	XFER[3]	0	Unused	X	XFER increased from 3 bits to 4 bits to support larger data transfers.

Register 0x3A0 : THDL Indirect Block Select

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
11	Reserved	X	Unused	X	Reserved bit must be set low for correct operation of FREEDM-84A672.
10	BLOCK[10]	0	Unused	X	BLOCK increased from 9 bits to 11 bits as the result of increase in addressable blocks from 512 to 2048.
9	BLOCK[9]	0	Unused	X	

Register 0x3A4 : THDL Indirect Block Data

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
11	Reserved	X	Unused	X	Reserved bit must be set low for correct operation of FREEDM-84A672.
10	BPTR[10]	0	Unused	X	BPTR increased from 9 bits to 11 bits as the result of increase in addressable blocks from 512 to 2048.
9	BPTR[9]	0	Unused	X	

Register 0x3B0 : THDL Configuration

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
7	Reserved	0	BURSTEN	0	Reserved bits must be set low for correct operation of FREEDM-84A672. The DMA burst length feature is not used in the FREEDM-84A672.
3	Reserved	0	Unused	X	
2	Reserved	0	BURST[2]	0	
1	Reserved	0	BURST[1]	0	
0	Reserved	0	BURST[0]	0	

Register 0x400 : TCAS Indirect Link and Time-slot Select

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
12	LINK[6]	0	LINK[4]	0	Increase in the number of transmit links from 32 to 84.
11	LINK[5]	0	LINK[3]	0	
10	LINK[4]	0	LINK[2]	0	
9	LINK[3]	0	LINK[1]	0	
8	LINK[2]	0	LINK[0]	0	
7	LINK[1]	0	Unused	X	
6	LINK[0]	0	Unused	X	

Register 0x404 : TCAS Indirect Channel Data

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
15	PROV	0	Unused	X	
9	CHAN[9]	0	Unused	X	Increase in size of CHAN from 7 bits to 10 bits as the result of increase in HDLC channels from 128 to 672.
8	CHAN[8]	0	PROV	0	
7	CHAN[7]	0	Unused	X	

Register 0x410 : TCAS Channel Disable

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
9	DCHAN[9]	0	Unused	X	Increase in size of DCHAN from 7 bits to 10 bits as the result of increase in HDLC channels from 128 to 672.
8	DCHAN[8]	0	Unused	X	
7	DCHAN[7]	0	Unused	X	

Registers 0x480 – 0x488 : TCAS Links #0 to #2 Configuration

Bit	FREEDM-84A672		FREEDM-32		Comments
	Function	Default	Function	Default	
4	Reserved	0	Unused	X	The reserved bits must be set low for correct operation of the FREEDM-84A672.
2	Reserved	0	BSYNC	0	
1	Reserved	0	E1	0	
0	Reserved	0	CEN	0	CEN, E1 and BSYNC are not applicable in the FREEDM-84A672.

CONTACTING PMC-SIERRA, INC.

PMC-Sierra, Inc.
105-8555 Baxter Place Burnaby, BC
Canada V5A 4V7

Tel: (604) 415-6000

Fax: (604) 415-6200

Document Information:	document@pmc-sierra.com
Corporate Information:	info@pmc-sierra.com
Application Information:	apps@pmc-sierra.com (604) 415-4533
Web Site:	http://www.pmc-sierra.com

None of the information contained in this document constitutes an express or implied warranty by PMC-Sierra, Inc. as to the sufficiency, fitness or suitability for a particular purpose of any such information or the fitness, or suitability for a particular purpose, merchantability, performance, compatibility with other parts or systems, of any of the products of PMC-Sierra, Inc., or any portion thereof, referred to in this document. PMC-Sierra, Inc. expressly disclaims all representations and warranties of any kind regarding the contents or use of the information, including, but not limited to, express and implied warranties of accuracy, completeness, merchantability, fitness for a particular use, or non-infringement.

In no event will PMC-Sierra, Inc. be liable for any direct, indirect, special, incidental or consequential damages, including, but not limited to, lost profits, lost business or lost data resulting from any use of or reliance upon the information, whether or not PMC-Sierra, Inc. has been advised of the possibility of such damage.

PMC-990716 (A1) date: June 1999