PM7385

FREEDM™-84A672

FRAME ENGINE AND DATALINK MANAGER 84A672

DATA SHEET

PROPRIETARY AND CONFIDENTIAL

ISSUE 6: AUGUST 2001

REVISION HISTORY

Issue No.	Issue Date	Details of Change
Issue 1	Jan 7, 1999	Creation of Document.
Issue 2	July 8, 1999	Update as per issue 3 of the engineering document (PMC-981263).
Issue 3	January, 2000	Update as per issue 4 of the engineering document (PMC-981263), for GCA release.
Issue 4	June 2000	Re-issue to coincide with production release of Rev C of device. Minor corrections and changes to some DC and AC timing parameters.
Issue 5	October 2000	Re-issue to coincide with Issue 6 of the Eng Doc. DEFAULT_DRV register bit changed to PERM_DRV and description changed. (See PREP #4938.) Change bars have been kept to show both Issue 4 and Issue 5 changes.
Issue 6	August 2001	Patent information included. Change bars apply to previous issue.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 FEATURES

- Single-chip multi-channel HDLC controller with a 50 MHz, 16 bit “Any-PHY” Packet Interface (APPI) for transfer of packet data using an external controller.
- Supports up to 672 bi-directional HDLC channels assigned to a maximum of 84 channelised or unchannelised links conveyed via a Scaleable Bandwidth Interconnect (SBI) interface.
- Data on the SBI interface is divided into 3 Synchronous Payload Envelopes (SPEs). Each SPE can be configured independently to carry data for either 28 T1/J1 links, 21 E1 links, or 1 unchannelised DS-3 link.
- Links in an SPE can be configured individually to operate in clear channel mode, in which case, all framing bit locations are assumed to be carrying HDLC data.
- Links in an SPE can be configured individually to operate in channelised mode, in which case, the number of time-slots assigned to an HDLC channel is programmable from 1 to 24 (for T1/J1 links) and from 1 to 31 (for E1 links).
- Supports three bi-directional HDLC channels each assigned to an unchannelised link with arbitrary rate link of up to 51.84 MHz when SYSCLK is running at 45 MHz. Each link may be configured individually to replace one of the SPEs conveyed on the SBI interface.
- For each channel, the HDLC receiver supports programmable flag sequence detection, bit de-stuffing and frame check sequence validation. The receiver supports the validation of both CRC-CCITT and CRC-32 frame check sequences.
- For each channel, the receiver checks for packet abort sequences, octet aligned packet length and for minimum and maximum packet length. The receiver supports filtering of packets that are larger than a user specified maximum value.
- Alternatively, for each channel, the receiver supports a transparent mode where each octet is transferred transparently on the receive APPI. For channelised links, the octets are aligned with the receive time-slots.
- For each channel, time-slots are selectable to be in 56 kbits/s format or 64 kbits/s clear channel format.

- For each channel, the HDLC transmitter supports programmable flag sequence generation, bit stuffing and frame check sequence generation. The transmitter supports the generation of both CRC-CCITT and CRC-32 frame check sequences. The transmitter also aborts packets under the direction of the external controller or automatically when the channel underflows.

- Alternatively, for each channel, the transmitter supports a transparent mode where each octet is inserted transparently from the transmit APPI. For channelised links, the octets are aligned with the transmit time-slots.

- Supports per-channel configurable APPI burst sizes of up to 256 bytes for transfers of packet data.

- The FREEDM maintains packet level performance metrics such as number of received packets, number of received packets with frame check sequence errors, number of transmitted packets, number of receive aborted packets, and number of transmit aborted packets.

- Provides 32 Kbytes of on-chip memory for partial packet buffering in both the transmit and the receive directions. This memory may be configured to support a variety of different channel configurations from a single channel with 32 Kbytes of buffering to 672 channels, each with a minimum of 48 bytes of buffering.

- Provides a 16 bit microprocessor interface for configuration and status monitoring.

- Provides a standard 5 signal P1149.1 JTAG test port for boundary scan board test purposes.

- Supports 3.3 Volt tolerant I/O.

- Low power 2.5 Volt 0.25 μm CMOS technology.

- 352 pin enhanced ball grid array (SBGA) package.

2 APPLICATIONS

- IETF PPP interfaces for routers
- Frame Relay interfaces for ATM or Frame Relay switches and multiplexors
- FUNI or Frame Relay service inter-working interfaces for ATM switches and multiplexors.
- Internet/Intranet access equipment.
- Packet-based DSLAM equipment.
- Packet over SONET.
- PPP over SONET.

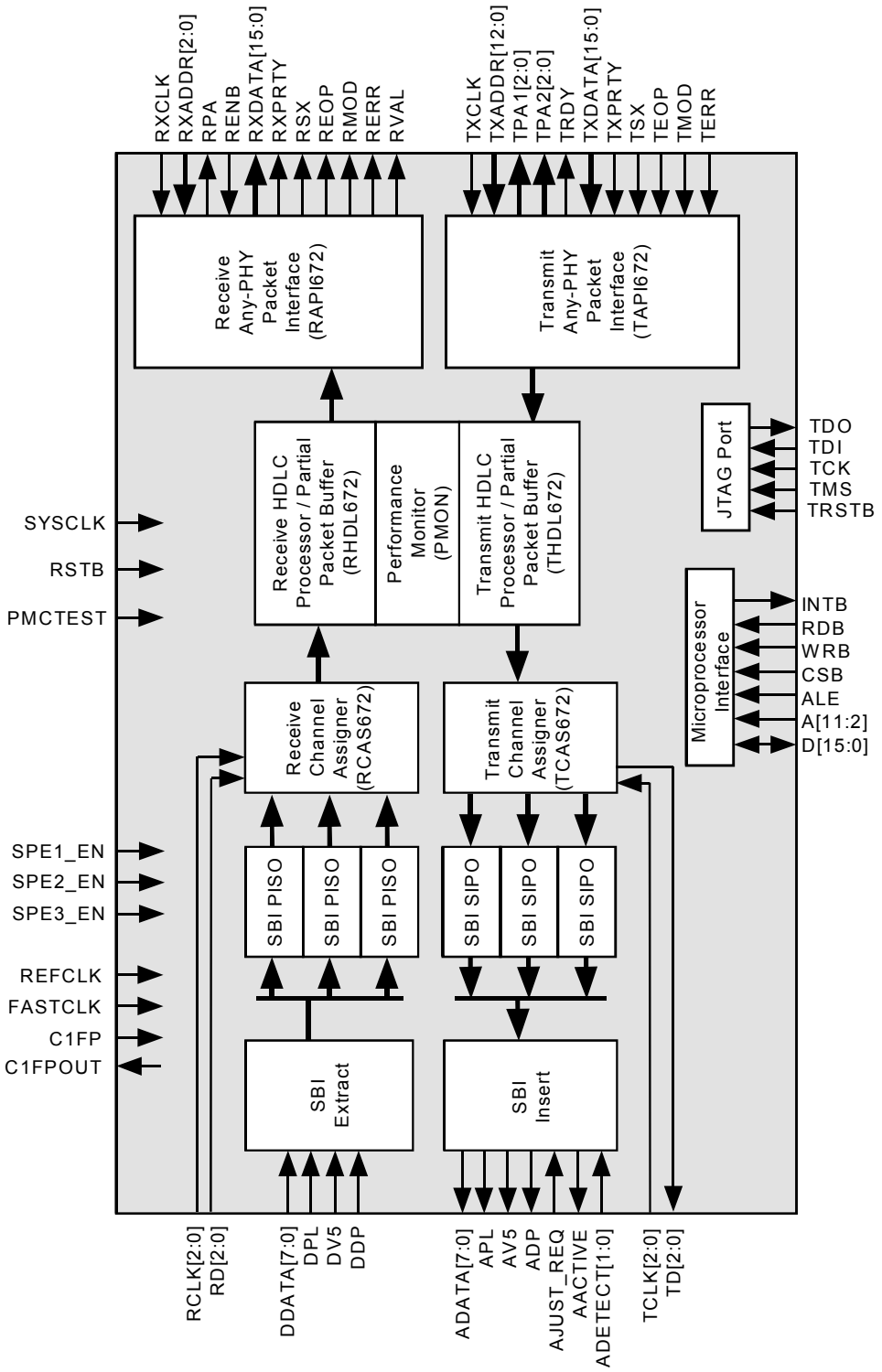
3 REFERENCES

1. International Organization for Standardization, ISO Standard 3309-1993, "Information Technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures – Frame structure", December 1993.

2. RFC-1662 – "PPP in HDLC-like Framing" Internet Engineering Task Force, July 1994.

3. PMC-1981125 – "High Density T1/E1 Framer with Integrated VT/TU Mapper and M13 Multiplexer (TEMUX) Data Sheet", PMC-Sierra Inc.

4 BLOCK DIAGRAM



5 DESCRIPTION

The PM7385 FREEDM-84A672 Frame Engine and Datalink Manager device is a monolithic integrated circuit that implements HDLC processing for a maximum of 672 bi-directional channels.

The FREEDM-84A672 may be configured to support channelised T1/J1/E1 or unchannelised 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. 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 unchannellised HDLC stream. The links can be of arbitrary frame format and can operate at up to 51.84 MHz provided SYSCLK is running at 45 MHz. When activated, each link replaces one of the SPEs conveyed on the SBI interface. (The maximum possible data rate when all three clock/data links are activated is 155.52 Mbps.)

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 the 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.

In the receive direction, the FREEDM-84A672 performs channel assignment and packet extraction and validation. For each provisioned HDLC channel, the FREEDM-84A672 delineates the packet boundaries using flag sequence detection, and performs bit de-stuffing. Sharing of opening and closing flags, as well as sharing of zeros between flags are supported. The resulting packet data is placed into the internal 32 Kbyte partial packet buffer RAM. The partial packet buffer acts as a logical FIFO for each of the assigned channels. An external controller transfers partial packets out of the RAM, across the receive APPI bus, into host packet memory. The FREEDM-84A672 validates the frame check sequence for each packet, and verifies that the packet is an integral number of octets in length and is within a programmable minimum and maximum lengths. Receive APPI bus latency may cause one or more channels to overflow, in which case, the packets are aborted. The FREEDM-84A672 reports the status of each packet on the receive APPI at the end of each packet transfer.

Alternatively, in the receive direction, the FREEDM-84A672 supports a transparent operating mode. For each provisioned transparent channel, the FREEDM-84A672 directly transfers the received octets onto the receive APPI verbatim. If the transparent channel is assigned to a channelised link, then the octets are aligned to the received time-slots.

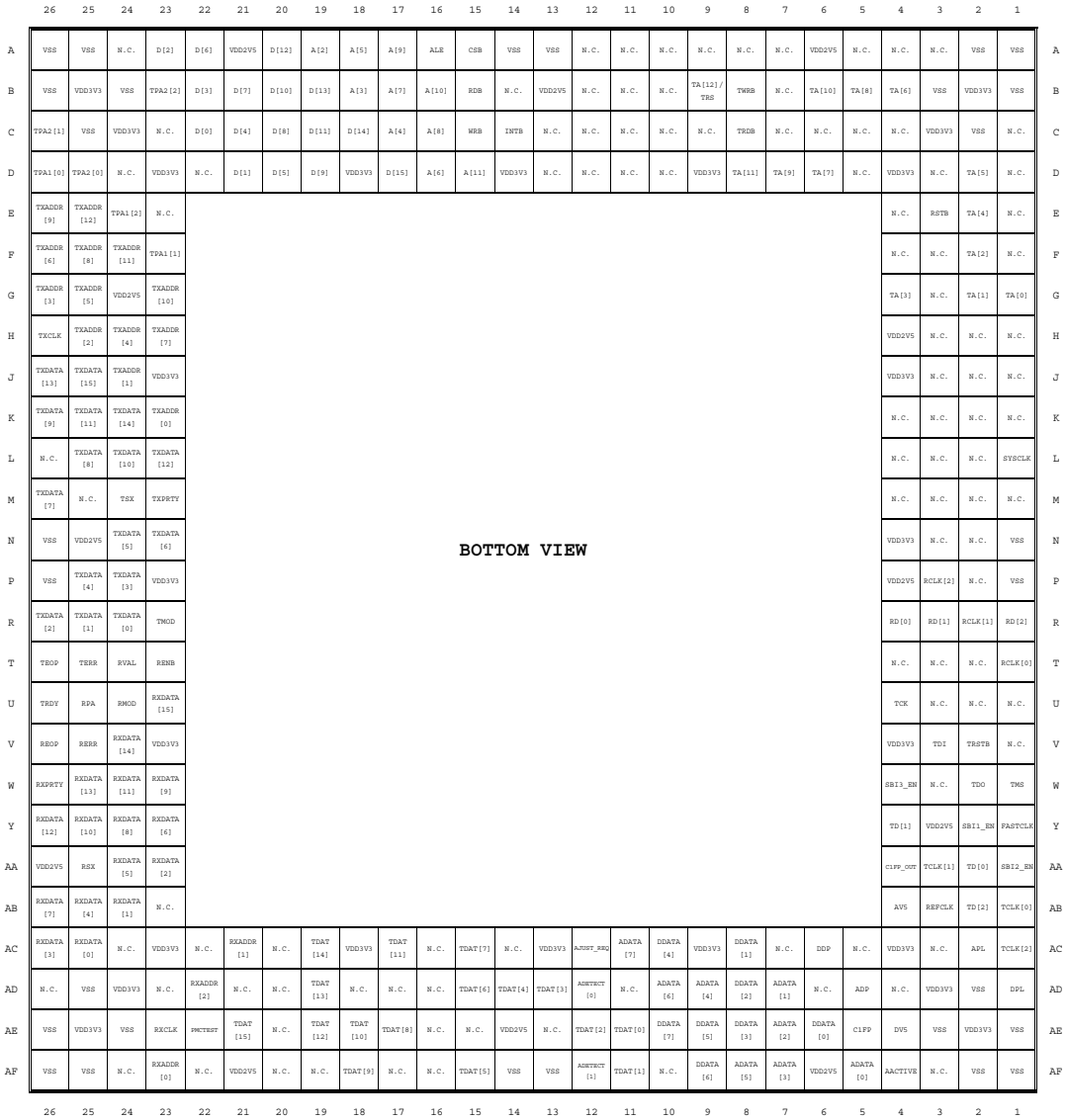
In the transmit direction, an external controller provides packets to transmit using the transmit APPI. For each provisioned HDLC channel, an external controller transfers partial packets, across the transmit APPI, into the internal 32 Kbyte transmit partial packet buffer. The partial packets are read out of the partial packet buffer by the FREEDM-84A672 and a frame check sequence is optionally calculated and inserted at the end of each packet. Bit stuffing is performed before being assigned to a particular link. The flag or idle sequence is automatically inserted when there is no packet data for a particular channel. Sequential packets are optionally separated by a single flag (combined opening and closing flag) or up to 128 flags. Zeros between flags are not shared in the transmit direction although, as stated previously, they are accepted in the receive direction. Transmit APPI bus latency may cause one or more channels to underflow, in which case, the packets are aborted. The FREEDM-84A672 generates an interrupt to notify the host of aborted packets. For normal traffic, an abort sequence is generated, followed by inter-frame time fill characters (flags or all-ones bytes) until a new packet is sourced on the transmit APPI. The FREEDM-84A672 will not attempt to re-transmit aborted packets.

Alternatively, in the transmit direction, the FREEDM-84A672 supports a transparent operating mode. For each provisioned transparent channel, the FREEDM-84A672 directly inserts the transmitted octets provided on the transmit APPI. If the transparent channel is assigned to a channelised link, then the octets are aligned to the transmitted time-slots. If a channel underflows due to excessive transmit APPI bus latency, an abort sequence is generated, followed by inter-frame time fill characters (flags or all-ones bytes) to indicate idle channel. Data resumes immediately when the FREEDM-84A672 receives new data on the transmit APPI.

The FREEDM-84A672 is configured, controlled and monitored using the microprocessor interface. The FREEDM-84A672 is implemented in low power 2.5 Volt 0.25 μm CMOS technology. All FREEDM-84A672 I/O are 3.3 volt tolerant. The FREEDM-84A672 is packaged in a 352 pin enhanced ball grid array (SBGA) package.

6 PIN DIAGRAM

The FREEDM-84A672 is manufactured in a 352 pin enhanced ball grid array (SBGA) package.



	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
A	VSS	VSS	N.C.	D[2]	D[6]	VDD2V5	D[12]	A[2]	A[5]	A[9]	ALE	CSB	VSS	VSS	N.C.	N.C.	N.C.	N.C.	N.C.	N.C.	VDD2V5	N.C.	N.C.	N.C.	VSS	VSS	A
B	VSS	VDD3V3	VSS	TPA2[2]	D[3]	D[7]	D[10]	D[13]	A[3]	A[7]	A[10]	RDB	N.C.	VDD2V5	N.C.	N.C.	N.C.	TA[12] / TRS	TWRB	N.C.	TA[10]	TA[8]	TA[6]	VSS	VDD3V3	VSS	B
C	TPA2[1]	VSS	VDD3V3	N.C.	D[0]	D[4]	D[8]	D[11]	D[14]	A[4]	A[8]	WRB	INTB	N.C.	N.C.	N.C.	N.C.	N.C.	TRDB	N.C.	N.C.	N.C.	N.C.	VDD3V3	VSS	N.C.	C
D	TPA1[0]	TPA2[0]	N.C.	VDD3V3	N.C.	D[1]	D[5]	D[9]	VDD3V3	D[15]	A[6]	A[11]	VDD3V3	N.C.	N.C.	N.C.	N.C.	VDD3V3	TA[11]	TA[9]	TA[7]	N.C.	VDD3V3	N.C.	TA[5]	N.C.	D
E	TXADDR[9]	TXADDR[12]	TPA1[2]	N.C.																			N.C.	RSTB	TA[4]	N.C.	E
F	TXADDR[6]	TXADDR[8]	TXADDR[11]	TPA1[1]																			N.C.	N.C.	TA[2]	N.C.	F
G	TXADDR[3]	TXADDR[5]	VDD2V5	TXADDR[10]																			TA[3]	N.C.	TA[1]	TA[0]	G
H	TXCLK	TXADDR[2]	TXADDR[4]	TXADDR[7]																			VDD2V5	N.C.	N.C.	N.C.	H
J	TXDATA[13]	TXDATA[15]	TXADDR[1]	VDD3V3																			VDD3V3	N.C.	N.C.	N.C.	J
K	TXDATA[9]	TXDATA[11]	TXDATA[14]	TXADDR[0]																			N.C.	N.C.	N.C.	N.C.	K
L	N.C.	TXDATA[8]	TXDATA[10]	TXDATA[12]																			N.C.	N.C.	N.C.	SYSCLK	L
M	TXDATA[7]	N.C.	TSX	TXPRTY																			N.C.	N.C.	N.C.	N.C.	M
N	VSS	VDD2V5	TXDATA[5]	TXDATA[6]																			VDD3V3	N.C.	N.C.	VSS	N
P	VSS	TXDATA[4]	TXDATA[3]	VDD3V3																			VDD2V5	RCLK[2]	N.C.	VSS	P
R	TXDATA[2]	TXDATA[1]	TXDATA[0]	TMOD																			RD[0]	RD[1]	RCLK[1]	RD[2]	R
T	TEOP	TERR	RVAL	RENB																			N.C.	N.C.	N.C.	RCLK[0]	T
U	TRDY	RPA	RMOD	RXDATA[15]																			TCK	N.C.	N.C.	N.C.	U
V	REOP	RERR	RXDATA[14]	VDD3V3																			VDD3V3	TDI	TRSTB	N.C.	V
W	RXPRTY	RXDATA[13]	RXDATA[11]	RXDATA[9]																			SBI3_EN	N.C.	TDO	TMS	W
Y	RXDATA[12]	RXDATA[10]	RXDATA[8]	RXDATA[6]																			TD[1]	VDD2V5	SBI1_EN	FASTCLK	Y
AA	VDD2V5	RSX	RXDATA[5]	RXDATA[2]																			C1PP_OUT	TCLK[1]	TD[0]	SBI2_EN	AA
AB	RXDATA[7]	RXDATA[4]	RXDATA[1]	N.C.																			AV5	REFCLK	TD[2]	TCLK[0]	AB
AC	RXDATA[3]	RXDATA[0]	N.C.	VDD3V3	N.C.	RXADDR[1]	N.C.	TDAT[14]	VDD3V3	TDAT[11]	N.C.	TDAT[7]	N.C.	VDD3V3	ADJUST_REQ	ADATA[7]	DDATA[4]	VDD3V3	DDATA[1]	N.C.	DDP	N.C.	VDD3V3	N.C.	APL	TCLK[2]	AC
AD	N.C.	VSS	VDD3V3	N.C.	RXADDR[2]	N.C.	N.C.	TDAT[13]	N.C.	N.C.	N.C.	TDAT[6]	TDAT[4]	TDAT[3]	ADETECT[0]	N.C.	ADATA[6]	ADATA[4]	DDATA[2]	ADATA[1]	N.C.	ADP	N.C.	VDD3V3	VSS	DPL	AD
AE	VSS	VDD3V3	VSS	RXCLK	PMCTEST	TDAT[15]	N.C.	TDAT[12]	TDAT[10]	TDAT[8]	N.C.	N.C.	VDD2V5	N.C.	TDAT[2]	TDAT[0]	DDATA[7]	DDATA[5]	DDATA[3]	ADATA[2]	DDATA[0]	C1FP	DV5	VSS	VDD3V3	VSS	AE
AF	VSS	VSS	N.C.	RXADDR[0]	N.C.	VDD2V5	N.C.	N.C.	TDAT[9]	N.C.	N.C.	TDAT[5]	VSS	VSS	ADETECT[1]	TDAT[1]	N.C.	DDATA[6]	ADATA[5]	ADATA[3]	VDD2V5	ADATA[0]	AACTIVE	N.C.	VSS	VSS	AF
	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

BOTTOM VIEW

7 PIN DESCRIPTION

Table 1 – SBI Interface Signals (30)

Pin Name	Type	Pin No.	Function
REFCLK	Input	AB3	The SBI reference clock signal (REFCLK) provides reference timing for the SBI ADD and DROP busses. REFCLK is nominally a 50% duty cycle clock of frequency 19.44 MHz ±50ppm.
FASTCLK	Input	Y1	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. FASTCLK is nominally a 50% duty cycle, ±50ppm clock having one of the following frequencies: 51.84 MHz, 44.928 MHz or 66 MHz.
C1FP	Input	AE5	The C1 octet frame pulse signal (C1FP) provides frame synchronisation for devices connected via an SBI interface. C1FP must be asserted for 1 REFCLK cycle every 500 µs or multiples thereof (i.e. every 9720 n REFCLK cycles, where n is a positive integer). All devices interconnected via an SBI interface must be synchronised to a C1FP signal from a single source. C1FP is sampled on the rising edge of REFCLK. Note – If the SBI bus is being operated in synchronous mode [Ref. 3], C1FP must be asserted for 1 REFCLK cycle every 6 ms or multiples thereof.

Pin Name	Type	Pin No.	Function
C1FPOUT	Output	AA4	The C1 octet frame pulse output signal (C1FPOUT) may be used to provide frame synchronisation for devices interconnected via an SBI interface. C1FPOUT is asserted for 1 REFCLK cycle every 500 µs (i.e. every 9720 REFCLK cycles). If C1FPOUT is used for synchronisation, it must be connected to the C1FP inputs of all the devices connected to the SBI interface. C1FPOUT is updated on the rising edge of REFCLK. Note – The C1FPOUT pulse generated by FREEDM-84A672 is not suitable for use in systems in which the SBI bus is operated in synchronous mode [Ref. 3].
DDATA[0] DDATA[1] DDATA[2] DDATA[3] DDATA[4] DDATA[5] DDATA[6] DDATA[7]	Input	AE6 AC8 AD8 AE8 AC10 AE9 AF9 AE10	The SBI DROP bus data signals (DDATA[7:0]) contain the time division multiplexed receive data from the up to 84 independently timed links. Data from each link is transported as a tributary within the SBI TDM bus structure. Multiple PHY devices can drive the SBI DROP bus at uniquely assigned tributary column positions. DDATA[7:0] are sampled on the rising edge of REFCLK.
DDP	Input	AC6	The SBI DROP bus parity signal (DDP) carries the even or odd parity for the DROP bus signals. The parity calculation encompasses the DDATA[7:0], DPL and DV5 signals. Multiple PHY devices can drive DDP at uniquely assigned tributary column positions. This parity signal is intended to detect accidental PHY source clashes in the column assignment. DDP is sampled on the rising edge of REFCLK.

Pin Name	Type	Pin No.	Function
DPL	Input	AD1	The SBI DROP bus payload signal (DPL) indicates valid data within the SBI TDM bus structure. This signal is asserted during all octets making up a tributary. This signal may be asserted during the V3 or H3 octet within a tributary to accommodate negative timing adjustments between the tributary rate and the fixed TDM bus structure. This signal may be deasserted during the octet following the V3 or H3 octet within a tributary to accommodate positive timing adjustments between the tributary rate and the fixed TDM bus structure. Multiple PHY devices can drive DPL at uniquely assigned tributary column positions. DPL is sampled on the rising edge of REFCLK.
DV5	Input	AE4	The SBI DROP bus payload indicator signal (DV5) locates the position of the floating payloads for each tributary within the SBI TDM bus structure. Timing differences between the port timing and the TDM bus timing are indicated by adjustments of this payload indicator relative to the fixed TDM bus structure. Multiple PHY devices can drive DV5 at uniquely assigned tributary column positions. All movements indicated by this signal must be accompanied by appropriate adjustments in the DPL signal. DV5 is sampled on the rising edge of REFCLK.

Pin Name	Type	Pin No.	Function
ADATA[0] ADATA[1] ADATA[2] ADATA[3] ADATA[4] ADATA[5] ADATA[6] ADATA[7]	Tristate Output	AF5 AD7 AE7 AF7 AD9 AF8 AD10 AC11	The SBI ADD bus data signals (ADATA[7:0]) contain the time division multiplexed transmit data from the up to 84 independently timed links. Data from each link is transported as a tributary within the SBI TDM bus structure. Multiple link layer devices can drive the SBI ADD bus at uniquely assigned tributary column positions. ADATA[7:0] are tristated when the FREEDM-84A672 is not outputting data on a particular tributary column. ADATA[7:0] are updated on the rising edge of REFCLK.
ADP	Tristate Output	AD5	The SBI ADD bus parity signal (ADP) carries the even or odd parity for the ADD bus signals. The parity calculation encompasses the ADATA[7:0], APL and AV5 signals. Multiple link layer devices can drive this signal at uniquely assigned tributary column positions. ADP is tristated when the FREEDM-84A672 is not outputting data on a particular tributary column. This parity signal is intended to detect accidental link layer source clashes in the column assignment. ADP is updated on the rising edge of REFCLK.

Pin Name	Type	Pin No.	Function
APL	Tristate Output	AC2	The SBI ADD bus payload signal (APL) indicates valid data within the SBI TDM bus structure. This signal is asserted during all octets making up a tributary. This signal may be asserted during the V3 or H3 octet within a tributary to accommodate negative timing adjustments between the tributary rate and the fixed TDM bus structure. This signal may be deasserted during the octet following the V3 or H3 octet within a tributary to accommodate positive timing adjustments between the tributary rate and the fixed TDM bus structure. Multiple link layer devices can drive this signal at uniquely assigned tributary column positions. APL is tristated when the FREEDM-84A672 is not outputting data on a particular tributary column. APL is updated on the rising edge of REFCLK.
AV5	Tristate output	AB4	The SBI ADD bus payload indicator signal (AV5) locates the position of the floating payloads for each tributary within the SBI TDM bus structure. Timing differences between the port timing and the TDM bus timing are indicated by adjustments of this payload indicator relative to the fixed TDM bus structure. Multiple link layer devices can drive this signal at uniquely assigned tributary column positions. AV5 is tristated when the FREEDM-84A672 is not outputting data on a particular tributary column. AV5 is updated on the rising edge of REFCLK.

Pin Name	Type	Pin No.	Function
AJUST_REQ	Input	AC12	The SBI ADD bus justification request signal (AJUST_REQ) is used to speed up or slow down the output data rate of the FREEDM-84A672. Negative timing adjustments are requested by asserting AJUST_REQ during the V3 or H3 octet, depending on the tributary type. In response to this the FREEDM-84A672 will send an extra byte in the V3 or H3 octet of the next frame along with a valid APL indicating a negative justification. Positive timing adjustments are requested by asserting AJUST_REQ during the octet following the V3 or H3 octet, depending on the tributary type. FREEDM-84A672 will respond to this by not sending an octet during the octet following the V3 or H3 octet of the next frame and deasserting APL to indicate a positive justification. AJUST_REQ is sampled on the rising edge of REFCLK.
AACTIVE	Output	AF4	The SBI ADD bus active indicator signal (AACTIVE) is asserted whenever FREEDM-84A672 is driving the SBI ADD bus signals, ADATA[7:0], ADP, APL and AV5. All other Link Layer devices driving the SBI ADD bus should monitor this signal (to detect multiple sources accidentaly driving the bus) and should cease driving the bus whenever a conflict is detected. AACTIVE is updated on the rising edge of REFCLK.

Pin Name	Type	Pin No.	Function
ADETECT[0] ADETECT[1]	Input	AD12 AF12	The SBI ADD bus conflict detection signals (ADETECT[1:0]) may be connected to the AACTIVE outputs of other link layer devices sharing the SBI ADD bus. FREEDM-84A672 will immediately tristate the SBI ADD bus signals ADATA[7:0], ADP, APL and AV5 if either of ADETECT[1] and ADETECT[0] is asserted. ADETECT[1:0] are asynchronous inputs.

Table 2 – Clock/Data Interface Signals (15)

Pin Name	Type	Pin No.	Function
RCLK[0] RCLK[1] RCLK[2]	Input	T1 R2 P3	The receive line clock signals (RCLK[2:0]) contain the recovered line clock for the 3 independently timed links. RCLK[n] must be externally gapped during the bits or time-slots that are not part of the transmission format payload (i.e. not part of the HDLC packet). RCLK[2:0] is nominally a 50% duty cycle clock between 0 and 51.84 MHz. The RCLK[n] inputs are invalid and should be tied low when their associated link is not configured for operation (i.e. SPEn_EN input is high).
RD[0] RD[1] RD[2]	Input	R4 R3 R1	The receive data signals (RD[2:0]) contain the recovered line data for the 3 independently timed links. RD[2:0] contain HDLC packet data. For certain transmission formats, RD[2:0] may contain place holder bits or time-slots. RCLK[n] must be externally gapped during the place holder positions in the RD[n] stream. The FREEDM-84A672 supports a maximum data rate of 51.84 Mbit/s on each link. RD[2:0] are sampled on the rising edge of the corresponding RCLK[2:0].

Pin Name	Type	Pin No.	Function
TCLK[0] TCLK[1] TCLK[2]	Input	AB1 AA3 AC1	The transmit line clock signals (TCLK[2:0]) contain the transmit clocks for the 3 independently timed links. TCLK[n] must be externally gapped during the bits or time-slots that are not part of the transmission format payload (i.e. not part of the HDLC packet). TCLK[2:0] is nominally a 50% duty cycle clock between 0 and 51.84 MHz. The TCLK[n] inputs are invalid and should be tied low when their associated link is not configured for operation (i.e. SPEn_EN input is high).
TD[0] TD[1] TD[2]	Output	AA2 Y4 AB2	The transmit data signals (TD[2:0]) contain the transmit data for the 3 independently timed links. TD[2:0] contain HDLC packet data. For certain transmission formats, TD[2:0] may contain place holder bits or time-slots. TCLK[n] must be externally gapped during the place holder positions in the TD[n] stream. The FREEDM-84A672 supports a maximum data rate of 51.84 Mbit/s on each link. TD[2:0] are updated on the falling edge of the corresponding TCLK[2:0] clock.
SPE1_EN SPE2_EN SPE3_EN	Input	Y2 AA1 W4	The Synchronous Payload Envelope Enable signals (SPEn_EN) configure the operation of the clock/data inputs and the SBI Interface. When SPEn_EN 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. When SPEn_EN is high, the corresponding Synchronous Payload Envelope conveyed on the SBI interface is enabled and the corresponding independently timed link is disabled. SPEn_EN are asynchronous inputs.

Table 3 – Any-PHY Packet Interface Signals (70)

Pin Name	Type	Pin No.	Function
TXCLK	Input	H26	The transmit clock signal (TXCLK) provides timing for the transmit Any-PHY packet interface. TXCLK is a nominally 50% duty cycle, 25 to 50 MHz clock.
TXADDR[0] TXADDR[1] TXADDR[2] TXADDR[3] TXADDR[4] TXADDR[5] TXADDR[6] TXADDR[7] TXADDR[8] TXADDR[9] TXADDR[10] TXADDR[11] TXADDR[12]	Input	K23 J24 H25 G26 H24 G25 F26 H23 F25 E26 G23 F24 E25	The transmit address signals (TXADDR[12:0]) provide a channel address for polling a transmit channel FIFO. The 10 least significant bits provide the channel number (0 to 671) while the 3 most significant bits select one of seven possible FREEDM-84A672 devices sharing a single external controller. (One address is reserved as a null address.) The Tx APPI of each FREEDM-84A672 device is identified by the base address in the TAPI672 Control register. The TXADDR[12:0] signals are sampled on the rising edge of TXCLK.

Pin Name	Type	Pin No.	Function
TPA1[0] TPA1[1] TPA1[2] TPA2[0] TPA2[1] TPA2[2]	Tristate Output	D26 F23 E24 D25 C26 B23	The transmit packet available signals (TPA1[2:0] and TPA2[2:0]) reflects the status of a poll of two transmit channel FIFOs. TPA1[2:0] returns the polled results for channel address 'n' provided on TXADDR[12:0] and TPA2[2:0] returns the polled results for channel address 'n+1'. TPAn[2] reports packet underrun events and TPAn[1:0] report the fill state of the transmit channel FIFO. TPAn[2] is set high when one or more packets has underrun on the channel and a further data transfer has occurred since it was last polled. When TPAn[2] is set low, no packet has underrun on the channel since the last poll. TPAn[1:0] are coded as follows: TPAn[1:0] = "11" => Starving TPAn[1:0] = "10" => (Reserved) TPAn[1:0] = "01" => Space TPAn[1:0] = "00" => Full A "Starving" polled response indicates that the polled transmit channel FIFO is at risk of underflowing and should be supplied with data as soon as possible. A "Space" polled response indicates that the polled transmit channel FIFO can accept XFER[3:0] plus one blocks (16 bytes per block) of data. A "Full" polled response indicates that the polled transmit channel FIFO cannot accept XFER[3:0] plus one blocks of data. (XFER[3:0] is a per-channel programmable value – see description of register 0x38C.) It is the responsibility of the external controller to prevent channel underflow conditions by adequately polling each channel before data transfer. TPAn[2:0] are tristate during reset and when a device address other than the FREEDM-84A672's base address is provided on TXADDR[12:10].

Pin Name	Type	Pin No.	Function
			TPAn[2:0] are updated on the rising edge of TXCLK.
TRDY	Tristate Output	U26	The transmit ready signal (TRDY) indicates the ability of the transmit Any-PHY packet interface (APPI) to accept data. When TRDY is set low, the transmit APPI is unable to accept further data. When TRDY is set high, data provided on the transmit APPI will be accepted by the FREEDM-84A672 device. TRDY is asserted one TXCLK cycle after TSX is sampled high. TRDY is asserted by the FREEDM-84A672 device which was selected by the in-band channel address on TXDATA[15:0] when TSX was sampled high. If TRDY is driven low, the external controller must hold the data on TXDATA[15:0] until TRDY is driven high. TRDY may be driven low for 0 or more TXCLK cycles before it is driven high. TRDY is always driven tristate one TXCLK cycle after it is driven high. TRDY is tristate during reset. TRDY is updated on the rising edge of TXCLK. It is recommended that TRDY be connected externally to a weak pull-up, e.g. 10 kΩ.

Pin Name	Type	Pin No.	Function
TXDATA[0] TXDATA[1] TXDATA[2] TXDATA[3] TXDATA[4] TXDATA[5] TXDATA[6] TXDATA[7] TXDATA[8] TXDATA[9] TXDATA[10] TXDATA[11] TXDATA[12] TXDATA[13] TXDATA[14] TXDATA[15]	Input	R24 R25 R26 P24 P25 N24 N23 M26 L25 K26 L24 K25 L23 J26 K24 J25	The transmit data signals (TXDATA[15:0]) contain the transmit Any-PHY packet interface (APPI) data provided by the external controller. Data must be presented in big endian order, i.e. the byte in TXDATA[15:8] is transmitted by the FREEDM-84A672 before the byte in TXDATA[7:0]. The first word of each data transfer contains an address to identify the device and channel associated with the data being transferred. This prepended address must be qualified with the TSX signal. The 10 least significant bits provide the channel number (0 to 671) while the 3 most significant bits select one of seven possible FREEDM-84A672 devices sharing a single external controller. (One address is reserved as a null address.) The FREEDM-84A672 will not respond to channel addresses outside the range 0 to 671, nor to device addresses other than the base address stored in the TAPI672 Control register. The second and any subsequent words of each data transfer contain packet data. The TXDATA[15:0] signals are sampled on the rising edge of TXCLK.
TXPRTY	Input	M23	The transmit parity signal (TXPRTY) reflects the odd parity calculated over the TXDATA[15:0] signals. TXPRTY is only valid when TXDATA[15:0] are valid. TXPRTY is sampled on the rising edge of TXCLK.

Pin Name	Type	Pin No.	Function
TSX	Input	M24	The transmit start of transfer signal (TSX) denotes the start of data transfer on the transmit APPI. When the TSX signal is sampled high, the sampled word on the TXDATA[15:0] signals contain the device and channel address associated with the data to follow. When the TSX signal is sampled low, the sampled word on the TXDATA[15:0] signals do not contain a device/channel address. The TSX signal is sampled on the rising edge of TXCLK.
TEOP	Input	T26	The transmit end of packet signal (TEOP) denotes the end of a packet. TEOP is only valid during data transfer. When TEOP is sampled high, the data on TXDATA[15:0] is the last word of a packet. When TEOP is sampled low, the data on TXDATA[15:0] is not the last word of a packet. TEOP is sampled on the rising edge of TXCLK.
TMOD	Input	R23	The transmit word modulo signal (TMOD) indicates the size of the current word on TXDATA[15:0]. TMOD is only valid when TEOP is sampled high. When TMOD is sampled high and TEOP is sampled high, only the TXDATA[15:8] signals contain valid data and the TXDATA[7:0] signals are invalid. When TMOD is sampled low and TEOP is sampled high, the complete word on TXDATA[15:0] contains valid data. TMOD must be set low when TEOP is set low. TMOD is sampled on the rising edge of TXCLK.

Pin Name	Type	Pin No.	Function
TERR	Input	T25	The transmit error signal (TERR) indicates that the current packet is errored and should be aborted. TERR is only valid when TEOP is sampled high. When TERR is sampled high and TEOP is sampled high, the current packet is errored and the FREEDM-84A672 will respond accordingly. When TERR is sampled low and TEOP is sampled high, the current packet is not errored. TERR must be set low when TEOP is set low. TERR is sampled on the rising edge of TXCLK.
RXCLK	Input	AE23	The receive clock signal (RXCLK) provides timing for the receive Any-PHY packet interface (APPI). RXCLK is a nominally 50% duty cycle, 25 to 50 MHz clock.
RXADDR[0] RXADDR[1] RXADDR[2]	Input	AF23 AC21 AD22	The receive address signals (RXADDR[2:0]) serve two functions – device polling and device selection. When polling, the RXADDR[2:0] signals provide an address for polling a FREEDM-84A672 device for receive data available in any one of its 672 channels. Polling results are returned on the RPA tristate output. During selection, the address on the RXADDR[2:0] signals is qualified with the RENB signal to select a FREEDM-84A672 device enabling it to output data on the receive APPI. Note that up to seven FREEDM-84A672 devices may share a single external controller (one address is reserved as a null address). The Rx APPI of each FREEDM-84A672 device is identified by the base address in the RAPI672 Control register. The RXADDR[2:0] signals are sampled on the rising edge of RXCLK.

Pin Name	Type	Pin No.	Function
RPA	Tristate Output	U25	The receive packet available signal (RPA) reflects the status of a poll on the receive APPI of a FREEDM-84A672 device. When RPA is set high, the polled FREEDM-84A672 device has XFER[3:0] plus one blocks (16 bytes per block) of data to transfer, or alternatively, a smaller amount of data which includes an end of packet. When RPA is set low, the polled FREEDM-84A672 device does not have data ready to transfer. (XFER[3:0] is a per-channel programmable value – see description of register 0x208.) A FREEDM-84A672 device must not be selected for receive data transfer unless it has been polled and responded that it has data ready to transfer. When the RXADDR[2:0] inputs match the base address in the RAPI672 Control register, that FREEDM-84A672 device drives RPA one RXCLK cycle after sampling RXADDR[2:0]. RPA is tristate during reset and when a device address other than the FREEDM-84A672's base address is provided on RXADDR[2:0]. RPA is updated on the rising edge of RXCLK.

Pin Name	Type	Pin No.	Function
RENB	Input	T23	The receive enable signal (RENB) qualifies the RXADDR[2:0] signals for selection of a FREEDM-84A672 device. When RENB is sampled high and then low in consecutive RXCLK cycles, the address on RXADDR[2:0] during the cycle when RENB is sampled high selects a FREEDM-84A672 device enabling it to output data on the receive APPI. The Rx APPI of each FREEDM-84A672 device is identified by the base address in the RAPI672 Control register. The polling function of the RXADDR[2:0] and RPA signals operates regardless of the state of RENB. RENB may also be used to throttle the FREEDM-84A672 during data transfer on the Rx APPI. When the FREEDM-84A672 samples RENB high during data transfer, the FREEDM-84A672 will pause the data transfer and tri-state the receive APPI outputs (except RPA) until RENB is returned low. Since the Any-PHY bus specification does not support deselection during data transfers, the address on the RXADDR[2:0] inputs during the cycle before RENB is returned low must either re-select the same FREEDM-84A672 device or be a null address. To commence data transfer, RENB must be sampled low following device selection. It is the responsibility of the external controller to prevent overflow by providing each FREEDM-84A672 device on an Any-PHY point to multi-point bus sufficient bandwidth through selection. RENB is sampled on the rising edge of RXCLK.

Pin Name	Type	Pin No.	Function
RXDATA[0] RXDATA[1] RXDATA[2] RXDATA[3] RXDATA[4] RXDATA[5] RXDATA[6] RXDATA[7] RXDATA[8] RXDATA[9] RXDATA[10] RXDATA[11] RXDATA[12] RXDATA[13] RXDATA[14] RXDATA[15]	Tristate Output	AC25 AB24 AA23 AC26 AB25 AA24 Y23 AB26 Y24 W23 Y25 W24 Y26 W25 V24 U23	The receive data signals (RXDATA[15:0]) contain the receive Any-PHY packet interface (APPI) data output by the FREEDM-84A672 when selected. Data is presented in big endian format, i.e. the byte in RXDATA[15:8] was received by the FREEDM-84A672 before the byte in RXDATA[7:0]. The first word of each data transfer (when RSX is high) contains an address to identify the device and channel associated with the data being transferred. The 10 least significant bits (RXDATA[9:0]) contain the channel number (0 to 671) and the 3 most significant bits (RXDATA[15:13]) contain the device base address. The second and any subsequent words of each data transfer contain valid data. The FREEDM-84A672 may be programmed to overwrite RXDATA[7:0] of the final word of each packet transfer (REOP is high) with the status of packet reception when that packet is errored (RERR is high). This status information is bit mapped as follows: RXDATA[0]=’1’ => channel FIFO overrun. RXDATA[1]=’1’ => max. packet length violation. RXDATA[2]=’1’ => FCS error. RXDATA[3]=’1’ => non-octet aligned. RXDATA[4]=’1’ => HDLC packet abort. RXDATA[7:5]=”Xh” => Reserved. The RXDATA[15:0] signals are tristated when the FREEDM-84A672 device is not selected via the RENB signal. The RXDATA[15:0] signals are updated on the rising edge of RXCLK.

Pin Name	Type	Pin No.	Function
RXPRTY	Tristate Output	W26	The receive parity signal (RXPRTY) reflects the odd parity calculated over the RXDATA[15:0] signals. RXPRTY is driven/tristated at the same time as RXDATA[15:0]. RXPRTY is updated on the rising edge of RXCLK.
RSX	Tristate Output	AA25	The receive start of transfer signal (RSX) denotes the start of data transfer on the receive APPI. When the RSX signal is set high, the 3 most significant bits on the RXDATA[15:0] signals contain the FREEDM-84A672 device address and the 10 least significant bits on the RXDATA[15:0] signals contain the channel address associated with the data to follow. Valid device addresses are in the range 0 through 7 (with one address reserved as a null address) and valid channel addresses are in the range 0 through 671. When the RSX signal is sampled low, the word on the RXDATA[15:0] signals does not contain a device and channel address. RSX is tristated when the FREEDM-84A672 device is not selected via the RENB signal. RSX is updated on the rising edge of RXCLK. It is recommended that RSX be connected externally to a weak pull-down, e.g. 10 kΩ.
REOP	Tristate Output	V26	The receive end of packet signal (REOP) denotes the end of a packet. REOP is only valid during data transfer. When REOP is set high, RXDATA[15:0] contains the last data byte of a packet. When REOP is set low, RXDATA[15:0] does not contain the last data byte of a packet. REOP is tristated when the FREEDM-84A672 device is not selected via the RENB signal. REOP is updated on the rising edge of RXCLK.

Pin Name	Type	Pin No.	Function
RMOD	Tristate Output	U24	The receive word modulo signal (RMOD) indicates the size of the current word on RXDATA[15:0]. When RDAT[15:0] does not contain the last byte of a packet (REOP set low), RMOD is set low. When RMOD is set high and REOP is set high, RXDATA[15:8] contains the last data byte of a packet. When RMOD is set low and REOP is set high, RXDATA[7:0] contains the last byte of the packet, or optionally, the error status byte. The behavior of RMOD relates only to packet data and is unaffected when the FREEDM-84A672 device is programmed to overwrite RXDATA[7:0] with status information when errored packets are received. RMOD is tristated when the FREEDM-84A672 device is not selected via the RENB signal. RMOD is updated on the rising edge of RXCLK.
RERR	Tristate Output	V25	The receive error signal (RERR) indicates that the current packet is errored and should be discarded. When RDAT[15:0] does not contain the last byte of a packet (REOP set low), RERR is set low. When RERR is set high and REOP is set high, the current packet is errored. When RERR is set low and REOP is set high, the current packet is not errored. The FREEDM-84A672 may be programmed to overwrite RXDATA[7:0] of the final word of each packet transfer (REOP set high) with the status of packet reception when that packet is errored (RERR is high). RERR is tristated when the FREEDM-84A672 device is not selected via the RENB signal. RERR is updated on the rising edge of RXCLK.

Pin Name	Type	Pin No.	Function
RVAL	Tristate Output	T24	The receive data valid (RVAL) is asserted when packet data is being output on RXDATA[15:0]. It is deasserted whenever the FREEDM-84A672 device is selected, but not outputting packet data on RXDATA[15:0]. (E.g., when RSX is high and address/channel prepend is being output on RXDATA[15:0], RVAL is deasserted.) RVAL is tristated when the FREEDM-84A672 device is not selected via the RENB signal. RVAL is updated on the rising edge of RXCLK.

Table 4 – Microprocessor Interface Signals (31)

Pin Name	Type	Pin No.	Function
D[0] D[1] D[2] D[3] D[4] D[5] D[6] D[7] D[8] D[9] D[10] D[11] D[12] D[13] D[14] D[15]	I/O	C22 D21 A23 B22 C21 D20 A22 B21 C20 D19 B20 C19 A20 B19 C18 D17	The bi-directional data signals (D[15:0]) provide a data bus to allow the FREEDM-84A672 device to interface to an external micro-processor. Both read and write transactions are supported. The microprocessor interface is used to configure and monitor the FREEDM-84A672 device.
A[2] A[3] A[4] A[5] A[6] A[7] A[8] A[9] A[10] A[11]	Input	A19 B18 C17 A18 D16 B17 C16 A17 B16 D15	The address signals (A[11:2]) provide an address bus to allow the FREEDM-84A672 device to interface to an external micro-processor. All microprocessor accessible registers are dword aligned.
ALE	Input	A16	The address latch enable signal (ALE) latches the A[11:2] signals during the address phase of a bus transaction. When ALE is set high, the address latches are transparent. When ALE is set low, the address latches hold the address provided on A[11:2]. ALE has an integral pull-up resistor.

Pin Name	Type	Pin No.	Function
WRB	Input	C15	The write strobe signal (WRB) qualifies write accesses to the FREEDM-84A672 device. When CSB is set low, the D[15:0] bus contents are clocked into the addressed register on the rising edge of WRB.
RDB	Input	B15	The read strobe signal (RDB) qualifies read accesses to the FREEDM-84A672 device. When CSB is set low, the FREEDM-84A672 device drives the D[15:0] bus with the contents of the addressed register on the falling edge of RDB.
CSB	Input	A15	The chip select signal (CSB) qualifies read/write accesses to the FREEDM-84A672 device. The CSB signal must be set low during read and write accesses. When CSB is set high, the microprocessor interface signals are ignored by the FREEDM-84A672 device. If CSB is not required (register accesses controlled only by WRB and RDB) then CSB should be connected to an inverted version of the RSTB signal.
INTB	Open-Drain Output	C14	The interrupt signal (INTB) indicates that an interrupt source is active and unmasked. When INTB is set low, the FREEDM-84A672 device has an active interrupt that is unmasked. When INTB is tristate, no interrupts are active, or an active interrupt is masked. Please refer to the register description section of this document for possible interrupt sources and masking. It is the responsibility of the external microprocessor to read the status registers in the FREEDM-84A672 device to determine the exact cause of the interrupt. INTB is an open drain output.

Table 5 – Miscellaneous Interface Signals (111)

Pin Name	Type	Pin No.	Function
SYSCLK	Input	L1	The system clock (SYSCLK) provides timing for the core logic. SYSCLK is nominally a 50% duty cycle clock of frequency 45 MHz ±50ppm.
RSTB	Input	E3	The active low reset signal (RSTB) signal provides an asynchronous FREEDM-84A672 reset. RSTB is an asynchronous input. When RSTB is set low, all FREEDM-84A672 registers are forced to their default states. In addition, all SBI, APPI and µP interface output pins are forced tristate and will remain tristated until RSTB is set high.
PMCTEST	Input	AE22	The PMC production test enable signal (PMCTEST) places the FREEDM-84A672 is test mode. When PMCTEST is set high, production test vectors can be executed to verify manufacturing via the test mode interface signals TA[11:0], TA[12]/TRS, TRDB, TWRB and TDAT[15:0]. PMCTEST must be tied low for normal operation.
TCK	Input	U4	The test clock signal (TCK) provides timing for test operations that can be carried out using the IEEE P1149.1 test access port. TMS and TDI are sampled on the rising edge of TCK. TDO is updated on the falling edge of TCK.
TMS	Input	W1	The test mode select signal (TMS) controls the test operations that can be carried out using the IEEE P1149.1 test access port. TMS is sampled on the rising edge of TCK. TMS has an integral pull up resistor.
TDI	Input	V3	The test data input signal (TDI) carries test data into the FREEDM-84A672 via the IEEE P1149.1 test access port. TDI is sampled on the rising edge of TCK. TDI has an integral pull up resistor.

Pin Name	Type	Pin No.	Function
TDO	Tristate Output	W2	The test data output signal (TDO) carries test data out of the FREEDM-84A672 via the IEEE P1149.1 test access port. TDO is updated on the falling edge of TCK. TDO is a tristate output which is inactive except when scanning of data is in progress.
TRSTB	Input	V2	The active low test reset signal (TRSTB) provides an asynchronous FREEDM-84A672 test access port reset via the IEEE P1149.1 test access port. TRSTB is an asynchronous input with an integral pull up resistor. Note that when TRSTB is not being used, it must be connected to the RSTB input.
NC1-103	Open		These pins must be left unconnected.

Table 6 – Production Test Interface Signals (31)

Pin Name	Type	Pin No.	Function
TA[0] TA[1] TA[2] TA[3] TA[4] TA[5] TA[6] TA[7] TA[8] TA[9] TA[10] TA[11]	Input	G1 G2 F2 G4 E2 D2 B4 D6 B5 D7 B6 D8	The test mode address bus (TA[11:0]) selects specific registers during production test (PMCTEST set high) read and write accesses. In normal operation (PMCTEST set low), these signals should be grounded.
TA[12]/TRS	Input	B9	The test register select signal (TA[12]/TRS) selects between normal and test mode register accesses during production test (PMCTEST set high). TRS is set high to select test registers and is set low to select normal registers. In normal operation (PMCTEST set low), this signal should be grounded.
TRDB	Input	C8	The test mode read enable signal (TRDB) is set low during FREEDM-84A672 register read accesses during production test (PMCTEST set high). The FREEDM-84A672 drives the test data bus (TDAT[15:0]) with the contents of the addressed register while TRDB is low. In normal operation (PMCTEST set low), this signal should be tied to logic 1.
TWRB	Input	B8	The test mode write enable signal (TWRB) is set low during FREEDM-84A672 register write accesses during production test (PMCTEST set high). The contents of the test data bus (TDAT[15:0]) are clocked into the addressed register on the rising edge of TWRB. In normal operation (PMCTEST set low), this signal should be tied to logic 1.

Pin Name	Type	Pin No.	Function
TDAT[0] TDAT[1] TDAT[2] TDAT[3] TDAT[4] TDAT[5] TDAT[6] TDAT[7] TDAT[8] TDAT[9] TDAT[10] TDAT[11] TDAT[12] TDAT[13] TDAT[14] TDAT[15]	I/O	AE11 AF11 AE12 AD13 AD14 AF15 AD15 AC15 AE17 AF18 AE18 AC17 AE19 AD19 AC19 AE21	The bi-directional test mode data bus (TDAT[15:0]) carries data read from or written to FREEDM-84A672 registers during production test (PMCTEST set high). In normal operation (PMCTEST set low), these signals should be left unconnected.

Table 7 – Power and Ground Signals (64)

Pin Name	Type	Pin No.	Function
VDD3V3[1] VDD3V3[2] VDD3V3[3] VDD3V3[4] VDD3V3[5] VDD3V3[6] VDD3V3[7] VDD3V3[8] VDD3V3[9] VDD3V3[10] VDD3V3[11] VDD3V3[12] VDD3V3[13] VDD3V3[14] VDD3V3[15] VDD3V3[16] VDD3V3[17] VDD3V3[18] VDD3V3[19] VDD3V3[20] VDD3V3[21] VDD3V3[22] VDD3V3[23] VDD3V3[24]	Power	B25 C3 C24 D4 D9 D14 D18 D23 J4 N4 P23 J23 V4 V23 AC4 AC9 AC13 AC18 AC23 AD3 AE2 AE25 B2 AD24	The VDD3V3[24:1] DC power pins should be connected to a well decoupled +3.3 V DC supply. These power pins provide DC current to the I/O pads.
VDD2V5[1] VDD2V5[2] VDD2V5[3] VDD2V5[4] VDD2V5[5] VDD2V5[6] VDD2V5[7] VDD2V5[8] VDD2V5[9] VDD2V5[10] VDD2V5[11] VDD2V5[12]	Power	H4 P4 Y3 AF6 AE14 AF21 AA26 N25 G24 A21 B13 A6	The VDD2V5[12:1] DC power pins should be connected to a well decoupled +2.5 V DC supply. These power pins provide DC current to the digital core.

Pin Name	Type	Pin No.	Function
VSS[1] VSS[2] VSS[3] VSS[4] VSS[5] VSS[6] VSS[7] VSS[8] VSS[9] VSS[10] VSS[11] VSS[12] VSS[13] VSS[14] VSS[15] VSS[16] VSS[17] VSS[18] VSS[19] VSS[20] VSS[21] VSS[22] VSS[23] VSS[24] VSS[25] VSS[26] VSS[27] VSS[28]	Ground	A1 A2 A13 A14 A25 A26 B1 B3 B24 B26 C2 C25 N1 N26 P1 P26 AD2 AD25 AE1 AE3 AE24 AE26 AF1 AF2 AF13 AF14 AF25 AF26	The VSS[28:1] DC ground pins should be connected to ground. They provide a ground reference for the 3.3 V rail. They also provide a ground reference for the 2.5 V rail.

Notes on Pin Description:

1. All FREEDM-84A672 inputs and bi-directionals present minimum capacitive loading and are 3.3V tolerant.

2. All FREEDM-84A672 outputs and bi-directionals have 8 mA drive capability except TDO, which has 4 mA drive capability.

3. All FREEDM-84A672 outputs can be tristated under control of the IEEE P1149.1 test access port, even those which do not tristate under normal operation. All outputs and bi-directionals are 3.3 V tolerant when tristated.

4. All inputs with the exception of the Any-PHY interface are Schmitt triggered. Inputs ALE, TMS, TDI and TRSTB have internal pull-up resistors.

5. Power to the VDD3V3 pins should be applied *before* power to the VDD2V5 pins is applied. Similarly, power to the VDD2V5 pins should be removed *before* power to the VDD3V3 pins is removed.

8 FUNCTIONAL DESCRIPTION

8.1 Scaleable Bandwidth Interconnect (SBI) Interface

The Scaleable Bandwidth Interconnect 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 2 kHz or 166.7Hz frame pulse. All sources and sinks of data on the bus are timed to the reference clock and frame pulse.

Timing is communicated across the Scaleable Bandwidth Interconnect by floating data structures. Payload indicator signals in the SBI control the position of the floating data structure and therefore the timing. When sources are running faster than the SBI the floating payload structure is advanced by an octet by passing an extra octet in the V3 octet locations (H3 octet for DS3 mappings). When the source is slower than the SBI the floating payload is retarded by leaving the octet after the V3 or H3 octet unused. Both these rate adjustments are indicated by the SBI control signals.

An SBI interface consists of a DROP BUS and an ADD BUS. On the DROP BUS all timing is sourced from the PHY and is passed onto the FREEDM-84A672 by the arrival rate of data over the SBI. On the ADD BUS timing can be controlled by either the PHY or the FREEDM-84A672. When the FREEDM-84A672 is the timing master, the PHY device determines its transmit timing information from the arrival rate of data across the SBI. When the PHY device is the timing master, it signals the FREEDM-84A672 to speed up or slow down with justification request signals. The PHY timing master indicates a speedup request to the Link Layer by asserting the justification request signal high during the V3 or H3 octet. When this is detected by the FREEDM-84A672 it will advance the channel by inserting data in the next V3 or H3 octet as described above. The PHY timing master indicates a slowdown request to the FREEDM-84A672 by asserting the justification request signal high during the octet after the V3 or H3 octet. The FREEDM-84A672 responds by leaving the octet following the next V3 or H3 octet unused. Both advance and retard rate adjustments take place in the frame or multi-frame following the justification request.

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. Unchannelized DS3 payloads follow a byte synchronous structure modeled on the SONET/SDH format.

The SBI structure uses a locked SONET/SDH structure fixing the position of the TUG-3/TU-3 relative to the STS-3/STM-1 transport frame. The SBI is also of fixed

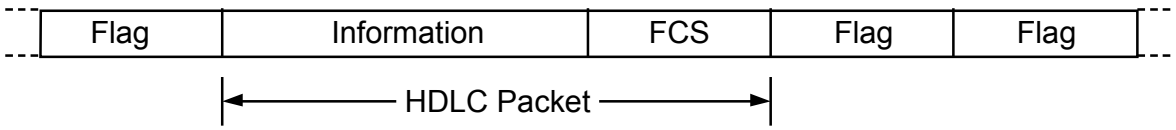
frequency and alignment as determined by the reference clock (REFCLK) and frame indicator signal (C1FP). Frequency deviations are compensated by adjusting the location of the T1/J1/E1/DS3 channels using floating tributaries as determined by the V5 indicator and payload signals (DV5, AV5, DPL and APL).

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

8.2 High-Level Data Link Control (HDLC) Protocol

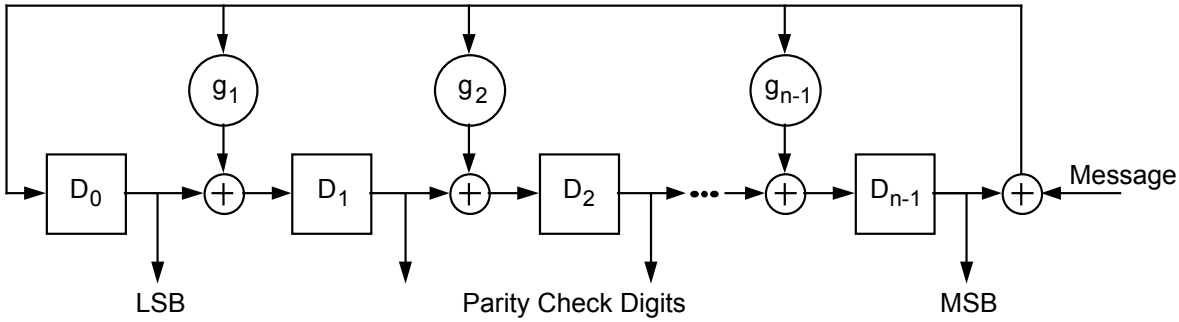
Figure 1 shows a diagram of the synchronous HDLC protocol supported by the FREEDM-84A672 device. The incoming stream is examined for flag bytes (01111110 bit pattern) which delineate the opening and closing of the HDLC packet. The packet is bit de-stuffed which discards a "0" bit which directly follows five contiguous "1" bits. The resulting HDLC packet size must be a multiple of an octet (8 bits) and within the expected minimum and maximum packet length limits. The minimum packet length is that of a packet containing two information bytes (address and control) and FCS bytes. For packets with CRC-CCITT as FCS, the minimum packet length is four bytes while those with CRC-32 as FCS, the minimum length is six bytes. An HDLC packet is aborted when seven contiguous "1" bits (with no inserted "0" bits) are received. At least one flag byte must exist between HDLC packets for delineation. Contiguous flag bytes, or all ones bytes between packets are used as an "inter-frame time fill". Adjacent flag bytes may share zeros.

Figure 1 – HDLC Frame



The CRC algorithm for the frame checking sequence (FCS) field is either a CRC-CCITT or CRC-32 function. Figure 2 shows a CRC encoder block diagram using the generating polynomial $g(X) = 1 + g_1X + g_2X^2 + \ldots + g_{n-1}X^{n-1} + X^n$. The CRC-CCITT FCS is two bytes in size and has a generating polynomial $g(X) = 1 + X^5 + X^{12} + X^{16}$. The CRC-32 FCS is four bytes in size and has a generating polynomial $g(X) = 1 + X + X^2 + X^4 + X^5 + X^7 + X^8 + X^{10} + X^{11} + X^{12} + X^{16} + X^{22} + X^{23} + X^{26} + X^{32}$. The first FCS bit received is the residue of the highest term.

Figure 2 – CRC Generator



8.3 SBI Extracter and PISO

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 processes data from one SPE on 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.

8.4 Receive Channel Assigner

The Receive Channel Assigner block (RCAS672) 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 51.84 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.

Table 8 – SBI SPE/Tributary to RCAS Link Mapping

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
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

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	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 so configured operate as unchannelised regardless of link rate and the RCAS672 performs a table lookup using only the link number to determine the associated channel.

8.4.1 Line Interface

There are 84 line interface blocks in the RCAS672. Each line interface block contains a bit counter, an 8-bit shift register and a holding register that, together, perform serial to parallel conversion. Whenever the holding register is updated, a request for service is sent to the priority encoder block. When acknowledged by the priority encoder, the line interface responds with the data residing in the holding register.

To support channelised links, each line interface block contains a time-slot counter. The time-slot counter is incremented each time the holding register is updated and is reset on detection of a frame pulse from the SBI PISO blocks. For unchannelised or unframed links, the time-slot counter is held reset.

8.4.2 Priority Encoder

The priority encoder monitors the line interfaces for requests and synchronises them to the SYSCLK timing domain. Requests are serviced on a fixed priority scheme where highest to lowest priority is assigned from the line interface attached to link 0 to that attached to link 83. Thus, simultaneous requests from link 'm' will be serviced ahead of link 'n', if m < n. When there are no pending requests, the priority encoder generates an idle cycle. In addition, once every fourth SYSCLK cycle, the priority encoder inserts a null cycle where no requests are serviced. This cycle is used by the channel assigner downstream for host microprocessor accesses to the provisioning RAMs.

8.4.3 Channel Assigner

The channel assigner block determines the channel number of the data byte currently being processed. The block contains a 2688 word channel provision RAM. The address of the RAM is constructed from concatenating the link number and the time-slot number of the current data byte. The fields of each RAM word include the channel number and a time-slot enable flag. The time-slot enable flag labels the current time-slot as belonging to the channel indicated by the channel number field.

8.4.4 Loopback Controller

The loopback controller block implements the channel based diagnostic loopback function. Every valid data byte belonging to a channel with diagnostic loopback enabled from the Transmit HDLC Processor / Partial Packet Buffer block (THDL672) is written into a 256 word FIFO. The loopback controller monitors for an idle time-slot or a time-slot carrying a channel with diagnostic loopback enabled. If either conditions hold, the current data byte is replaced by data retrieved from the loopback data FIFO.

8.5 Receive HDLC Processor / Partial Packet Buffer

The Receive HDLC Processor / Partial Packet Buffer block (RHDL672) processes up to 672 synchronous transmission HDLC data streams. Each channel can be individually configured to perform flag sequence detection, bit de-stuffing and CRC-CCITT or CRC-32 verification. The packet data is written into the partial packet buffer. At the end of a frame, packet status including CRC error, octet alignment error and maximum length violation are also loaded into the partial packet buffer. Alternatively, a channel can be provisioned as transparent, in which case, the HDLC data stream is passed to the partial packet buffer processor verbatim.

There is a natural precedence in the alarms detectable on a receive packet. Once a packet exceeds the programmable maximum packet length, no further processing is performed on it. Thus, octet alignment detection, FCS verification and abort recognition are squelched on packets with a maximum length violation. An abort indication squelches octet alignment detection, minimum packet length violations, and FCS verification. In addition, FCS verification is only performed on packets that do not have octet alignment errors, in order to allow the RHDL672 to perform CRC calculations on a byte-basis.

The partial packet buffer is a 32 Kbyte RAM that is divided into 16-byte blocks. Each block has an associated pointer which points to another block. A logical FIFO is created for each provisioned channel by programming the block pointers to form a circular linked list. A channel FIFO can be assigned a minimum of 3 blocks (48 bytes) and a maximum of 2048 blocks (32 Kbytes). The depth of the channel FIFOs are monitored in a round-robin fashion. Requests are made to the Receive Any-PHY Interface block (RAPI672) to transfer, on the Rx APPI, data in channel FIFOs with depths exceeding their associated threshold.

8.5.1 HDLC Processor

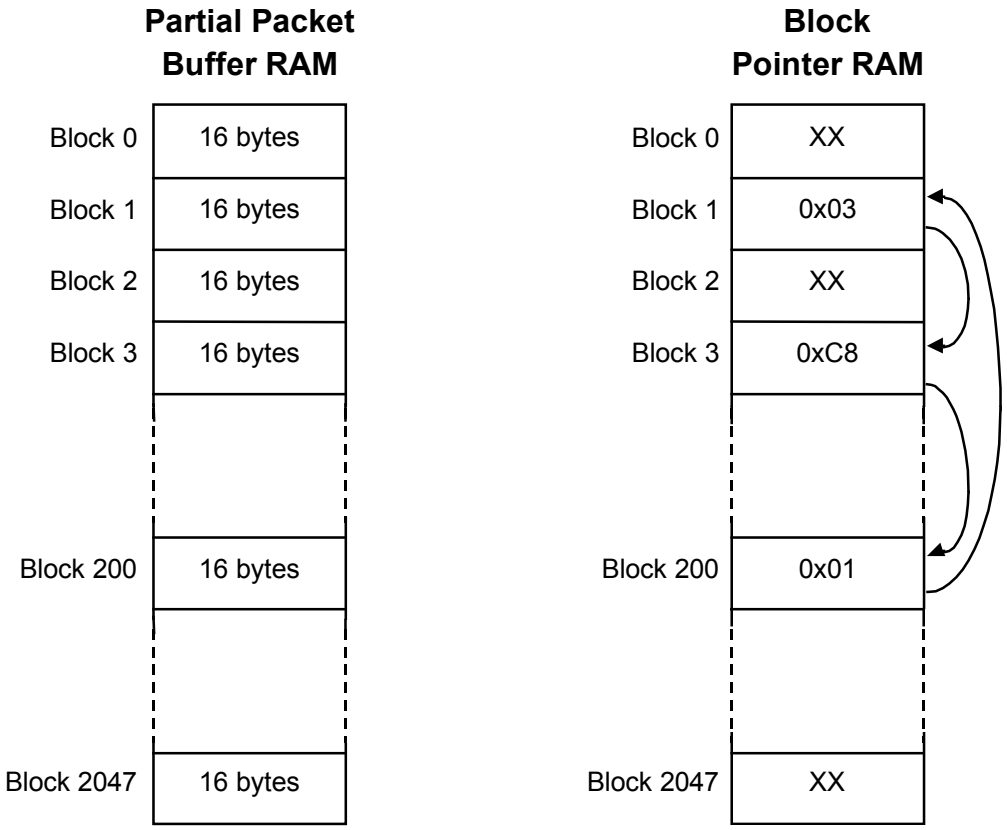
The HDLC processor is a time-slice state machine which can process up to 672 independent channels. The state vector and provisioning information for each channel is stored in a RAM. Whenever new channel data arrives, the appropriate state vector is read from the RAM, processed and written back to the RAM. The HDLC state-machine can be configured to perform flag delineation, bit de-stuffing, CRC verification and length monitoring. The resulting HDLC data and status information is passed to the partial packet buffer processor to be stored in the appropriate channel FIFO buffer.

The configuration of the HDLC processor is accessed using indirect channel read and write operations. When an indirect operation is performed, the information is accessed from RAM during a null clock cycle generated by the upstream Receive Channel Assigner block (RCAS672). Writing new provisioning data to a channel resets the channel's entire state vector.

8.5.2 Partial Packet Buffer Processor

The partial packet buffer processor controls the 32 Kbyte partial packet RAM which is divided into 16 byte blocks. A block pointer RAM is used to chain the partial packet blocks into circular channel FIFO buffers. Thus, non-contiguous sections of the RAM can be allocated in the partial packet buffer RAM to create a channel FIFO. System software is responsible for the assignment of blocks to individual channel FIFOs. Figure 3 shows an example of three blocks (blocks 1, 3, and 200) linked together to form a 48 byte channel FIFO.

Figure 3 – Partial Packet Buffer Structure



The partial packet buffer processor is divided into three sections: writer, reader and roamer. The writer is a time-sliced state machine which writes the HDLC data and status information from the HDLC processor into a channel FIFO in the packet buffer RAM. The reader transfers channel FIFO data from the packet buffer RAM to the downstream Receive Any-PHY Interface block (RAPI672). The roamer is a time-sliced state machine which tracks channel FIFO buffer depths and signals the reader to service a particular channel. If a buffer over-run occurs, the writer ends the current packet from the HDLC processor in the channel FIFO with an overrun flag and ignores the rest of the packet.

The FIFO algorithm of the partial packet buffer processor is based on a programmable per-channel transfer size. Instead of tracking the number of full blocks in a channel FIFO, the processor tracks the number of transactions. Whenever the partial packet writer fills a transfer-sized number of blocks or writes an end-of-packet flag to the channel FIFO, a transaction is created. Whenever the partial packet reader transmits a transfer-size number of blocks or an end-of-packet flag to the RAPI672 block, a transaction is deleted. Thus, small packets

less than the transfer size will be naturally transferred to the RAPI672 block without having to precisely track the number of full blocks in the channel FIFO.

The partial packet roamer performs the transaction accounting for all channel FIFOs. The roamer increments the transaction count when the writer signals a new transaction and sets a per-channel flag to indicate a non-zero transaction count. The roamer searches the flags in a round-robin fashion to decide for which channel FIFO to request transfer by the RAPI672 block. The roamer informs the partial packet reader of the channel to process. The reader transfers the data to the RAPI672 until the channel transfer size is reached or an end of packet is detected. The reader then informs the roamer that a transaction is consumed. The roamer updates its transaction count and clears the non-zero transaction count flag if required. The roamer then services the next channel with its transaction flag set high.

The writer and reader determine empty and full FIFO conditions using flags. Each block in the partial packet buffer has an associated flag. The writer sets the flag after the block is written and the reader clears the flag after the block is read. The flags are initialized (cleared) when the block pointers are written using indirect block writes. The writer declares a channel FIFO overrun whenever the writer tries to store data to a block with a set flag. In order to support optional removal of the FCS from the packet data, the writer does not declare a block as filled (set the block flag nor increment the transaction count) until the first double word of the next block in channel FIFO is filled. If the end of a packet resides in the first double word, the writer declares both blocks as full at the same time. When the reader finishes processing a transaction, it examines the first double word of the next block for the end-of-packet flag. If the first double word of the next block contains only FCS bytes, the reader would, optionally, process next transaction (end-of-packet) and consume the block, as it contains information not transferred to the RAPI672 block.

8.6 Receive Any-PHY Interface

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.

8.6.1 FIFO Storage and Control

The FIFO block temporarily stores channel data during transfer across the Rx APPI. RAPI672 burst data transfers are transaction based – a write burst data transfer must be complete before any data will be read, and all data must be completely read from the FIFO before any further data will be written into the FIFO. To support full Rx APPI bus rate, a double buffer scheme is used. While data is being read from one FIFO onto the Rx APPI, data can be written into the other FIFO. Because the bandwidth on the writer side of the FIFOs is higher than that on the reader side, the RAPI672 can maintain continuous full bandwidth transfer over the Rx APPI.

A maximum of 256 bytes can be stored in each of the two FIFOs for any given burst transfer. A separate storage element samples the 10 bit channel ID to associate the data in that FIFO with a specific HDLC channel. This channel ID is prepended in-band as the first word of every burst data transfer across the Rx APPI. (The maximum length of a burst data transfer on the Rx APPI is therefore 129 words, including prepend.) The 3 most significant bits of the prepended word of each burst data transfer across the Rx APPI identify the FREEDM-84A672 device associated with the transfer and reflect the value of the base address programmed in the RAPI672 Control register.

The writer controller provides a means for writing data into the FIFOs. The writer controller indicates that it can accept data when there is at least one completely empty FIFO. In response, a burst transfer of data, up to a maximum of 256 bytes, is written into that empty FIFO. (The transfer is sourced by the upstream RHDL672 block which selects from those channels with data available using its round-robin algorithm.) The writer controller then informs the reader controller that data is available in that FIFO. The writer controller now switches to the other FIFO and repeats the process. When both FIFOs are full, the writer throttles the upstream RHDL672 block to prevent of any further data writes into the FIFOs.

The reader controller provides a means of reading data out of the FIFOs onto the Rx APPI. When selected to do so, and the writer controller has indicated that at least one FIFO is full, the reader controller will read the data out of the FIFOs in the order in which they were filled. To prevent from overloading the Rx APPI with several small bursts of data, the RAPI672 automatically deselects after every burst transfer. This provides time for the upper layer device to detect an end of packet indication and possibly reselect a different FREEDM-84A672 device without having to store the extra word or two which may have been output onto the Rx APPI during the time it took for deselection.

The RAPI672 provides packet status information on the Rx APPI at the end of every packet transfer. The RAPI672 asserts RERR at the end of packet

reception (REOP high) to indicate that the packet is in error. The RAPI672 may optionally be programmed to overwrite RXDATA[7:0] of the final word of each packet transfer (REOP is high) with the status of packet reception when that packet is errored (RERR is high). Overwriting of status information is enabled by setting the STATEN bit in the RAPI Control register.

8.6.2 Polling Control and Management

The RAPI672 only responds to device polls which match the base address programmed in the RAPI672 Control register. A positive poll response indicates that at least one of the two FIFOs has a complete XFER[3:0] plus one blocks of data, or an end of packet, and is ready to be selected to transfer this data across the Rx APPI.

8.7 Transmit Any-PHY Interface

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.

8.7.1 FIFO Storage and Control

The FIFO block temporarily stores channel data during transfer across the Tx APPI. TAPI672 burst data transfers are transaction based on the writer side of the FIFO – all data must be completely read from the FIFO before any further data will be written into the FIFO. To support as close as possible to full Tx APPI bus rate, a double buffer is used. While data is being read from the one FIFO, data can be written into the other FIFO. Because the bandwidth on the reader side of the FIFOs is higher than that on the writer side, the TAPI672 will not incur any bandwidth reduction to maximum burst data transfers through its FIFOs.

The upper layer device cannot interrupt data transfers on the Tx APPI. However, the FREEDM-84A672 may throttle the upper layer device if both FIFOs in the TAPI672 are full. When the FIFOs in the TAPI672 cannot accept data, the TAPI672 deasserts the TRDY output to the upper layer device connected to the Tx APPI. In this instance, the upper layer device must halt data transfer until the TRDY output is returned high. The upper layer device connected to the Tx APPI must sample the TRDY output high before continuing to burst data across the Tx APPI.

A maximum of 256 bytes may be stored in each of the two FIFOs for any given burst transfer. The first word of each burst transfer contains a prepended address field. (The maximum length of a burst transfer on the Tx APPI is therefore 129 words, including prepend.) A separate storage element samples the 10 least significant bits of the prepended channel address to associate the data with a specific channel. The 3 most significant bits must match the base address programmed into the TAPI672 Control register for the TAPI672 to respond to the data transaction on the Tx APPI.

The writer controller provides a means for writing data from the Tx APPI into the FIFOs. The writer controller can accept data when there is at least one completely empty FIFO. When a data transfer begins and there are no empty FIFOs, the writer controller catches the data provided on the Tx APPI and throttles the upper layer device. The writer controller will continue to throttle the upper layer device until at least one FIFO is completely empty and can accept a maximum burst transfer of data.

The whisper controller provides the channel address of the data being written into the FIFO. As soon as the first word of data has been written into the FIFO, the whisper controller provides the channel information for that data to the downstream THDL672 block. The whisper controller will wait for acknowledgement and the reader controller is then requested to read the data from the FIFO. Once the reader controller has commenced the data transfer, the whisper controller will provide the channel information for the other FIFO. The whisper controller alternates between the two FIFOs in the order in which data is written into them.

The reader controller provides a means of reading data out of the FIFOs. When the writer controller indicates that data has been completely written into one of the two FIFOs, the reader controller is permitted to read that data. The reader controller will then wait for a request for data from the THDL672 block. When requested to transfer data, the reader controller will completely read all the data out of the FIFO before indicating to the writer controller that more data may be written into that FIFO. Because the reader controller reads data out of the FIFOs in the order in which they were filled, the THDL672 block will request data for channels in the order in which they were whispered. The reader controller manages the read and write FIFO pointers to allow simultaneous reading and writing of data to/from the double buffer FIFO.

8.7.2 Polling Control and Management

The TAPI672 only responds to poll addresses which are in the range programmed in the base address field in the TAPI672 Control register. The TAPI672 uses the 3 most significant bits of the poll address for device recognition

and the 10 least significant bits of the poll address for identification of a channel. The TAPI672 provides three poll results for every poll address according to Table 9. The TPAn[0] bit indicates whether or not space exists in the channel FIFO for data and the TPAn[1] bit indicates whether or not that polled channel FIFO is at risk of underflowing and should be provided data soon. The TPAn[2] bit indicates that an underflow event has occurred on that channel FIFO.

Table 9 – Transmit Polling

Poll Address	TPA1[0] (Full/Space)	TPA1[1] (Space/Starving)	TPA1[2] (Underflow)	TPA2[0] (Full/Space)	TPA2[1] (Space/Starving)	TPA2[2] (Underflow)
Channel 0	Channel 0	Channel 0	Channel 0	Channel 1	Channel 1	Channel 1
Channel 1	Channel 1	Channel 1	Channel 1	Channel 2	Channel 2	Channel 2
Channel 2	Channel 2	Channel 2	Channel 2	Channel 3	Channel 3	Channel 3
Channel 3	Channel 3	Channel 3	Channel 3	Channel 4	Channel 4	Channel 4
Channel 4	Channel 4	Channel 4	Channel 4	Channel 5	Channel 5	Channel 5
Channel 5	Channel 5	Channel 5	Channel 5	Channel 6	Channel 6	Channel 6
Channel 6	Channel 6	Channel 6	Channel 6	Channel 7	Channel 7	Channel 7
Channel 7	Channel 7	Channel 7	Channel 7	Channel 8	Channel 8	Channel 8
Channel 8	Channel 8	Channel 8	Channel 8	Channel 9	Channel 9	Channel 9
• • •	• • •	• • •	• • •	• • •	• • •	• • •
Channel 671	Channel 671	Channel 671	Channel 671	Channel 0	Channel 0	Channel 0

The TAPI672 maintains a mirror image of the status of each channel FIFO in the partial packet buffer. The THDL672 continuously reports the status of the 672 channel FIFOs to the TAPI672 and the TAPI672 updates the mirror image accordingly. The THDL672 also signals to the TAPI672 whenever an underflow event has occurred on a channel FIFO. At the beginning of every data transfer across the Tx APPI, the TAPI672 sets the mirror image status of the channel to “full”. Only the TAPI672 can cause the status to be set to “full” and only the THDL672 can cause the status to be set to “space” or “starving”. Only the THDL672 can cause the status to be set to “underflow” and only the TAPI672 can clear the “underflow” status when that channel FIFO is polled. In the event that both the TAPI672 and the THDL672 try to change the mirror image status of a particular channel simultaneously, the TAPI672 takes precedence, except for the “underflow” status, where the THDL672 takes precedence.

8.8 Transmit HDLC Controller / Partial Packet Buffer

The Transmit HDLC Controller / Partial Packet Buffer block (THDL672) contains a partial packet buffer for Tx APPI latency control and a transmit HDLC controller. The THDL672 also contains logic to monitor the full/empty status of each channel FIFO and push this status onto the polling interface signals.

The THDL672 requests data from the TAPI672 in response to control information from the TAPI672 indicating the channel for which data is available and ready to be transferred. Packet data received from the TAPI672 is stored in channel specific FIFOs residing in the partial packet buffer. When the amount of data in a FIFO reaches a programmable threshold, the HDLC controller is enabled to initiate transmission. The HDLC controller performs flag generation, bit stuffing and, optionally, frame check sequence (FCS) insertion. The FCS is software selectable to be CRC-CCITT or CRC-32. The minimum packet size, excluding FCS, is two bytes. A single byte payload is illegal. The HDLC controller delivers data to the Transmit Channel Assigner block (TCAS672) on demand. A packet in progress is aborted if an under-run occurs. The THDL672 is programmable to operate in transparent mode where packet data retrieved from the TAPI672 is transmitted verbatim.

8.8.1 Transmit HDLC Processor

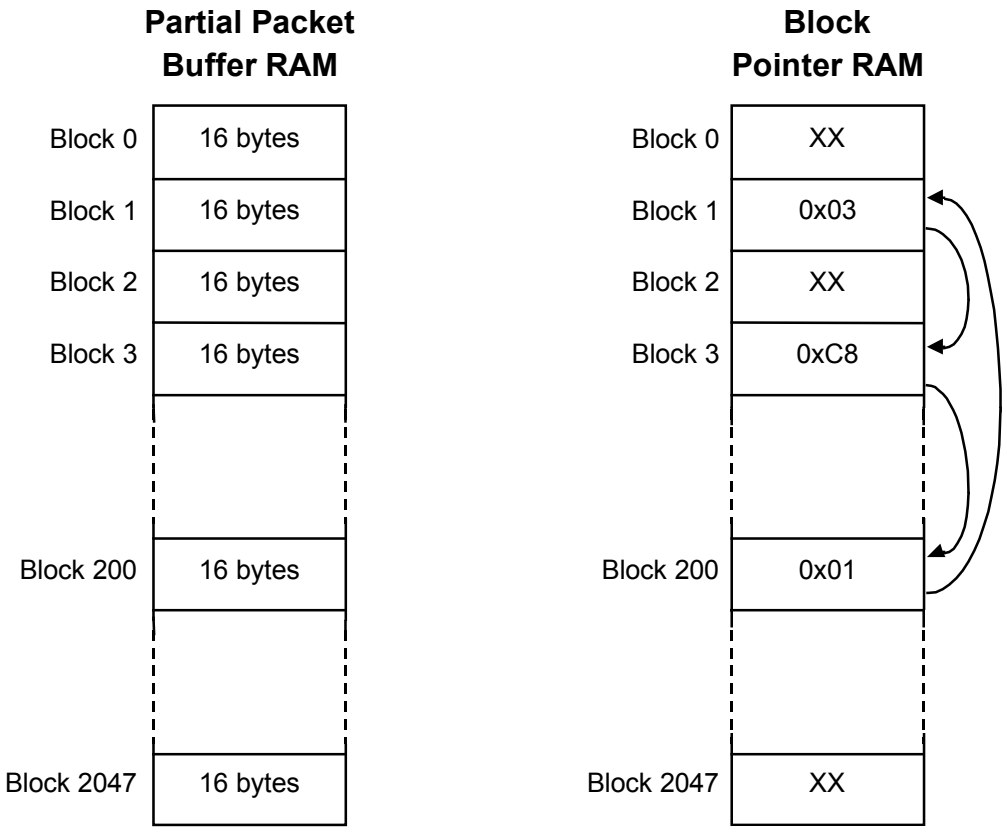
The HDLC processor is a time-slice state machine which can process up to 672 independent channels. The state vector and provisioning information for each channel is stored in a RAM. Whenever the TCAS672 requests data, the appropriate state vector is read from the RAM, processed and finally written back to the RAM. The HDLC state-machine can be configured to perform flag insertion, bit stuffing and CRC generation. The HDLC processor requests data from the partial packet processor whenever a request for channel data arrives. However, the HDLC processor does not start transmitting a packet until the entire packet is stored in the channel FIFO or until the FIFO free space is less than the software programmable limit. If a channel FIFO under-runs, the HDLC processor aborts the packet, generates a microprocessor interrupt and signals the underflow to the transmit Any-PHY interface.

The configuration of the HDLC processor is accessed using indirect channel read and write operations. When an indirect operation is performed, the information is accessed from RAM during a null clock cycle inserted by the TCAS672 block. Writing new provisioning data to a channel resets the channel's entire state vector.

8.8.2 Transmit Partial Packet Buffer Processor

The partial packet buffer processor controls the 32 Kbyte partial packet RAM which is divided into 16 byte blocks. A block pointer RAM is used to chain the partial packet blocks into circular channel FIFO buffers. Thus, non-contiguous sections of RAM can be allocated in the partial packet buffer RAM to create a channel FIFO. Figure 4 shows an example of three blocks (blocks 1, 3, and 200) linked together to form a 48 byte channel FIFO. The three pointer values would be written sequentially using indirect block write accesses. When a channel is provisioned within this FIFO, the state machine can be initialized to point to any one of the three blocks.

Figure 4 – Partial Packet Buffer Structure



The partial packet buffer processor is divided into three sections: reader, writer and roamer. The roamer is a time-sliced state machine which tracks each channel's FIFO buffer free space and signals the writer to service a particular channel. The writer requests data from the TAPI672 block and transfers packet data from the TAPI672 to the associated channel FIFO. The reader is a time-

sliced state machine which transfers the HDLC information from a channel FIFO to the HDLC processor in response to a request from the HDLC processor. If a buffer under-run occurs for a channel, the reader informs the HDLC processor and purges the rest of the packet. If a buffer overflow occurs for a channel (this can only happen if an external device disregards or mis-interprets poll results on the Tx APPI and transfers data to a channel which does not have space in its FIFO), the THDL672 overwrites the FIFO contents resulting in data corruption on that particular channel. When either an underflow or an overflow occurs, an interrupt is generated and the cause of the interrupt may be read via the interrupt status register using the microprocessor interface.

The writer and reader determine empty and full FIFO conditions using flags. Each block in the partial packet buffer has an associated flag. The writer sets the flag after the block is written and the reader clears the flag after the block is read. The flags are initialized (cleared) when the block pointers are written using indirect block writes. The reader declares a channel FIFO under-run whenever it tries to read data from a block without a set flag.

The FIFO algorithm of the partial packet buffer processor is based on per-channel software programmable transfer size and free space trigger level. Instead of tracking the number of full blocks in a channel FIFO, the processor tracks the number of empty blocks, called free space, as well as the number of end of packets stored in the FIFO. Recording the number of empty blocks instead of the number of full blocks reduces the amount of information the roamer must store in its state RAM.

The partial packet roamer records the FIFO free space and end-of-packet count for all channel FIFOs. When the reader signals that a block has been read, the roamer increments the FIFO free space and sets a per-channel request flag if the free space is greater than the limit set by XFER[3:0]. The roamer pushes this status information to the TAPI672 to indicate that it can accept at least XFER[3:0] blocks of data. The roamer also decrements the end-of-packet count when the reader signals that it has passed an end of a packet to the HDLC processor. If the HDLC processor is transmitting a packet and the FIFO free space is greater than the starving trigger level and there are no complete packets within the FIFO (end-of-packet count equal to zero), a per-channel starving flag is set. The roamer searches the starving flags in a round-robin fashion to decide which channel FIFOs are at risk of underflowing and pushes this status information to the TAPI672. The roamer listens to control information from the TAPI672 to decide which channel FIFO requests data from the TAPI672 block. The roamer informs the partial packet writer of the channel FIFO to process and the FIFO free space. The writer sends a request for data to the TAPI672 block, writes the response data to the channel FIFO, and sets the block full flags. The writer reports back to the roamer the number of blocks and end-of-packets transferred.

The maximum amount of data transferred during one request is limited by a software programmable limit (XFER[3:0]).

The roamer round-robins between all channel FIFOs and pushes the status to the TAPI672 block. The status consists of two pieces of information: (1) is there space in the channel FIFO for at least one XFER[3:0] of data, and (2) is this channel FIFO at risk of underflowing. Where a channel FIFO is at risk of underflowing, the THDL672 pushes a starving status for that channel FIFO to the TAPI672 at the earliest possible opportunity.

The configuration of the HDLC processor is accessed using indirect channel read and write operations as well as indirect block read and write operations. When an indirect operation is performed, the information is accessed from RAM during a null clock cycle identified by the TCAS672 block. Writing new provisioning data to a channel resets the entire state vector.

8.9 Transmit Channel Assigner

The Transmit Channel Assigner block (TCAS672) 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 51.84 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.

Table 10 – SBI SPE/Tributary to TCAS Link Mapping

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

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	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
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 51.84 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 tributaries 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 so configured operate as unchannelised regardless of link rate and the TCAS672 performs a table lookup using only the link number to determine the associated channel.

8.9.1 Line Interface

There are 84 line interface blocks in the TCAS672. Each line interface block contains a bit counter, an 8-bit shift register and a holding register that, together, perform parallel to serial conversion. Whenever the shift register is updated, a request for service is sent to the priority encoder block. When acknowledged by the priority encoder, the line interface responds by writing the data into the holding register.

To support channelised links, each line interface block contains a time-slot counter. The time-slot counter is incremented each time the shift register is updated and is reset on detection of a frame pulse from the SBI SIPO blocks. For unchannelised or unframed links, the time-slot counter is held reset.

8.9.2 Priority Encoder

The priority encoder monitors the line interfaces for requests and synchronises them to the SYSCLK timing domain. Requests are serviced on a fixed priority scheme where highest to lowest priority is assigned from the line interface attached to link 0 to that attached to link 83. Thus, simultaneous requests from link ‘m’ will be serviced ahead of link ‘n’, if m < n. The priority encoder selects the

request from the link with the highest priority for service. When there are no pending requests, the priority encoder generates an idle cycle. In addition, once every fourth SYSCLK cycle, the priority encoder inserts a null cycle where no requests are serviced. This cycle is used by the channel assigner upstream for CBI accesses to the channel provision RAM.

8.9.3 Channel Assigner

The channel assigner block determines the channel number of the request currently being processed. The block contains a 2688 word channel provision RAM. The address of the RAM is constructed from concatenating the link number and the time-slot number of the highest priority requester. The fields of each RAM word include the channel number and a time-slot enable flag. The time-slot enable flag labels the current time-slot as belonging to the channel indicted by the channel number field. For time-slots that are enabled, the channel assigner issues a request to the THDL672 block which responds with packet data within one byte period of the transmit stream.

8.10 SBI Inserter and SIPO

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 justfication requests received on the SBI interface. 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.

8.11 Performance Monitor

The Performance Monitor block (PMON) contains four counters. The first two accumulate receive partial packet buffer FIFO overrun events and transmit partial packet buffer FIFO underflow events, respectively. The remaining two counters are software programmable to accumulate a variety of events, such as receive packet count, FCS error counts, etc. All counters saturate upon reaching maximum value. The accumulation logic consists of a counter and holding register pair. The counter is incremented when the associated event is detected. Writing to the FREEDM-84A672 Master Clock / Frame Pulse Activity Monitor and Accumulation Trigger register transfer the count to the corresponding holding register and clear the counter. The contents of the holding register is accessible via the microprocessor interface.

8.12 JTAG Test Access Port Interface

The JTAG Test Access Port block provides JTAG support for boundary scan. The standard JTAG EXTEST, SAMPLE, BYPASS, IDCODE and STCTEST instructions are supported. The FREEDM-84A672 identification code is 073850CD hexadecimal.

8.13 Microprocessor Interface

The FREEDM-84A672 supports microprocessor access to an internal register space for configuring and monitoring the device. All registers are 16 bits wide but are DWORD aligned in the microprocessor memory map. The registers are described below:

Table 11 – Normal Mode Microprocessor Accessible Registers

Address	Register
0x000	FREEDM-84A672 Master Reset
0x004	FREEDM-84A672 Master Interrupt Enable
0x008	FREEDM-84A672 Master Interrupt Status
0x00C	FREEDM-84A672 Master Clock / Frame Pulse Activity Monitor and Accumulation Trigger
0x010	FREEDM-84A672 Reserved
0x014	FREEDM-84A672 Master Line Loopback
0x018 – 0x020	FREEDM-84A672 Reserved

Address	Register
0x024	FREEDM-84A672 Master Performance Monitor Control
0x028	FREEDM-84A672 Master SBI Interrupt Enable
0x02C	FREEDM-84A672 Master SBI Interrupt Status
0x030	FREEDM-84A672 Master Tributary Loopback #1
0x034	FREEDM-84A672 Master Tributary Loopback #2
0x038	FREEDM-84A672 Master Tributary Loopback #3
0x03C	FREEDM-84A672 Master Tributary Loopback #4
0x040	FREEDM-84A672 Master Tributary Loopback #5
0x044	FREEDM-84A672 Master Tributary Loopback #6
0x048	FREEDM-84A672 SBI DROP BUS Master Configuration
0x04C	FREEDM-84A672 SBI ADD BUS Master Configuration
0x050 – 0x0FC	Reserved
0x100	RCAS Indirect Channel and Time-slot Select
0x104	RCAS Indirect Channel Data
0x108	RCAS Reserved
0x10C	RCAS Channel Disable
0x110 – 0x13C	RCAS Reserved
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
0x158 – 0x17C	RCAS Reserved
0x180 – 0x188	RCAS Link #0 to #2 Configuration
0x18C - 0x1FC	RCAS Reserved
0x200	RHDL Indirect Channel Select
0x204	RHDL Indirect Channel Data Register #1

Address	Register
0x208	RHDL Indirect Channel Data Register #2
0x20C	RHDL Reserved
0x210	RHDL Indirect Block Select
0x214	RHDL Indirect Block Data Register
0x218 – 0x21C	RHDL Reserved
0x220	RHDL Configuration
0x224	RHDL Maximum Packet Length
0x228 – 0x23C	RHDL Reserved
0x240 – 0x37C	Reserved
0x380	THDL Indirect Channel Select
0x384	THDL Indirect Channel Data #1
0x388	THDL Indirect Channel Data #2
0x38C	THDL Indirect Channel Data #3
0x390 – 0x39C	THDL Reserved
0x3A0	THDL Indirect Block Select
0x3A4	THDL Indirect Block Data
0x3A8 – 0x3AC	THDL Reserved
0x3B0	THDL Configuration
0x3B4 – 0x3BC	THDL Reserved
0x3C0 – 0x3FC	Reserved
0x400	TCAS Indirect Channel and Time-slot Select
0x404	TCAS Indirect Channel Data
0x408	TCAS Reserved
0x40C	TCAS Idle Time-slot Fill Data
0x410	TCAS Channel Disable
0x414 – 0x43C	TCAS Reserved
0x440	TCAS SBI SPE1 Configuration Register #1
0x444	TCAS SBI SPE1 Configuration Register #2

Address	Register
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
0x458 - 0x47C	TCAS Reserved
0x480 - 0x488	TCAS Link #0 to #2 Configuration
0x48C - 0x4FC	TCAS Reserved
0x500	PMON Status
0x504	PMON Receive FIFO Overflow Count
0x508	PMON Transmit FIFO Underflow Count
0x50C	PMON Configurable Count #1
0x510	PMON Configurable Count #2
0x514 – 0x51C	PMON Reserved
0x520 – 0x57C	Reserved
0x580	RAPI Control
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 Channel Data Register
0x60C – 0x63C	TAPI Reserved

Address	Register
0x640 – 0x67C	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
0x700 – 0x7FC	Reserved

9 NORMAL MODE REGISTER DESCRIPTION

Normal mode registers are used to configure and monitor the operation of the FREEDM-84A672.

Notes on Normal Mode Register Bits:

1. 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 must 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 when read.

2. 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.

3. Writable normal mode register bits are cleared to logic zero upon reset unless otherwise noted.

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

5. 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.

9.1 Microprocessor Accessible Registers

Microprocessor accessible registers can be accessed by the external microprocessor. For each register description below, the hexadecimal register number indicates the address in the FREEDM-84A672 when accesses are made using the external microprocessor.

Note

These registers are not byte addressable. Writing to any one of these registers modifies all 16 bits in the register.

Register 0x000 : FREEDM-84A672 Master Reset

Bit	Type	Function	Default
Bit 15	R/W	Reset	0
Bit 14 to Bit 12		Unused	XH
Bit 11	R	TYPE[3]	0
Bit 10	R	TYPE[2]	1
Bit 9	R	TYPE[1]	0
Bit 8	R	TYPE[0]	1
Bit 7	R	ID[7]	0
Bit 6	R	ID[7]	0
Bit 5	R	ID[5]	0
Bit 4	R	ID[4]	0
Bit 3	R	ID[3]	0
Bit 2	R	ID[2]	0
Bit 1	R	ID[1]	1
Bit 0	R	ID[0]	0

This register provides software reset capability and device ID information.

RESET:

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. In addition, all registers are reset to their default values. 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.

Note

Unlike the hardware reset input (RSTB), RESET does not force the FREEDM-84A672's microprocessor interface pins tristate. RESET causes all registers to be set to their default values and forces the APPI outputs tristate.

TYPE[3:0]:

The Device Type bits (TYPE[3:0]) allow software to identify the device as the FREEDM-84A672 member of the FREEDM family of products.

ID[7:0]:

The Device ID bits (ID[7:0]) allow software to identify the version level of the FREEDM-84A672.

Register 0x004 : FREEDM-84A672 Master Interrupt Enable

Bit	Type	Function	Default
Bit 15	R/W	TFUDRE	0
Bit 14	R/W	TFOVRE	0
Bit 13	R/W	TUNPVE	0
Bit 12	R/W	TPRTYE	0
Bit 11 to Bit 6		Unused	XXH
Bit 5	R/W	RFOVRE	0
Bit 4	R/W	RPFEE	0
Bit 3	R/W	RABRTE	0
Bit 2	R/W	RFCSEE	0
Bit 1		Unused	X
Bit 0		Unused	X

This register provides interrupt enables for various events detected or initiated by the FREEDM-84A672.

RFCSEE:

The receive frame check sequence error interrupt enable bit (RFCSEE) enables receive FCS error interrupts to the microprocessor. When RFCSEE is set high, a mismatch between the received FCS code and the computed CRC residue will cause an interrupt to be generated on the INTB output. Interrupts are masked when RFCSEE is set low. However, the RFCSEI bit remains valid when interrupts are disabled and may be polled to detect receive FCS error events.

RABRTE:

The receive abort interrupt enable bit (RABRTE) enables receive HDLC abort interrupts to the microprocessor. When RABRTE is set high, receipt of an abort code (at least 7 contiguous 1's) will cause an interrupt to be generated on the INTB output. Interrupts are masked when RABRTE is set low. However, the RABRTI bit remains valid when interrupts are disabled and may be polled to detect receive abort events.

RPFEE:

The receive packet format error interrupt enable bit (RPFEE) enables receive packet format error interrupts to the microprocessor. When RPFEE is set high, receipt of a packet that is longer than the maximum specified in the RHDL Maximum Packet Length register, or a packet that is shorter than 32 bits (CRC-CCITT) or 48 bits (CRC-32), or a packet that is not octet aligned will cause an interrupt to be generated on the INTB output. Interrupts are masked when RPFEE is set low. However, the RPFEI bit remains valid when interrupts are disabled and may be polled to detect receive packet format error events.

RFOVRE:

The receive FIFO overrun error interrupt enable bit (RFOVRE) enables receive FIFO overrun error interrupts to the microprocessor. When RFOVRE is set high, attempts to write data into the logical FIFO of a channel when it is already full will cause an interrupt to be generated on the INTB output. Interrupts are masked when RFOVRE is set low. However, the RFOVRI bit remains valid when interrupts are disabled and may be polled to detect receive FIFO overrun events.

TPRTYE:

The transmit parity error interrupt enable bit (TPRTYE) enables parity errors on the transmit APPI to generate interrupts to the microprocessor. When TPRTYE is set high, detection of a parity error on the transmit APPI will cause an interrupt to be generated on the INTB output. Interrupts are masked when TPRTYE is set low. However, the TPRTYI bit remains valid when interrupts are disabled and may be polled to detect parity error events.

TUNPVE:

The transmit unprovisioned error interrupt enable bit (TUNPVE) enables attempted transmissions to unprovisioned channels to generate interrupts to the microprocessor. When TUNPVE is set high, attempts to write data to an unprovisioned channel will cause an interrupt to be generated on the INTB output. Interrupts are masked when TUNPVE is set low. However, the TUNPVI bit remains valid when interrupts are disabled and may be polled to detect attempted transmissions to unprovisioned channel events.

TFOVRE:

The transmit FIFO overflow error interrupt enable bit (TFOVRE) enables transmit FIFO overflow error interrupts to the microprocessor. When TFOVRE is set high, attempts to write data to the logical FIFO when it is already full will cause an interrupt to be generated on the INTB output.

Interrupts are masked when TFOVRE is set low. However, the TFOVRI bit remains valid when interrupts are disabled and may be polled to detect transmit FIFO overflow events.

TFUDRE:

The transmit FIFO underflow error interrupt enable bit (TFUDRE) enables transmit FIFO underflow error interrupts to the microprocessor. When TFUDRE is set high, attempts to read data from the logical FIFO when it is already empty will cause an interrupt to be generated on the INTB output. Interrupts are masked when TFUDRE is set low. However, the TFUDRI bit remains valid when interrupts are disabled and may be polled to detect transmit FIFO underflow events.

Register 0x008 : FREEDM-84A672 Master Interrupt Status

Bit	Type	Function	Default
Bit 15	R	TFUDRI	X
Bit 14	R	TFOVRI	X
Bit 13	R	TUNPVI	X
Bit 12	R	TPRTYI	X
Bit 11 to Bit 6		Unused	XXH
Bit 5	R	RFOVRI	X
Bit 4	R	RPFEI	X
Bit 3	R	RABRTI	X
Bit 2	R	RFCSEI	X
Bit 1		Unused	X
Bit 0		Unused	X

This register reports the interrupt status for various events detected or initiated by the FREEDM-84A672. Reading this registers acknowledges and clears the interrupts.

RFCSEI:

The receive frame check sequence error interrupt status bit (RFCSEI) reports receive FCS error interrupts to the microprocessor. RFCSEI is set high when a mismatch between the received FCS code and the computed CRC residue is detected. RFCSEI remains valid when interrupts are disabled and may be polled to detect receive FCS error events.

RABRTI:

The receive abort interrupt status bit (RABRTI) reports receive HDLC abort interrupts to the microprocessor. RABRTI is set high upon receipt of an abort code (at least 7 contiguous 1's). RABRTI remains valid when interrupts are disabled and may be polled to detect receive abort events.

RPFEI:

The receive packet format error interrupt status bit (RPFEI) reports receive packet format error interrupts to the microprocessor. RPFEI is set high upon receipt of a packet that is longer than the maximum programmed length, of a packet that is shorter than 32 bits (CRC-CCITT) or 48 bits (CRC-32), or of a packet that is not octet aligned. RPFEI remains valid when interrupts are disabled and may be polled to detect receive packet format error events.

RFOVRI:

The receive FIFO overrun error interrupt status bit (RFOVRI) reports receive FIFO overrun error interrupts to the microprocessor. RFOVRI is set high on attempts to write data into the logical FIFO of a channel when it is already full. RFOVRI remains valid when interrupts are disabled and may be polled to detect receive FIFO overrun events.

TPRTYI:

The transmit parity error interrupt status bit (TPRTYI) reports the detection of a parity on the transmit APPI. TPRTYI is set high upon detection of a parity error. TPRTYI remains valid when interrupts are disabled and may be polled to detect parity errors.

TUNPVI:

The transmit unprovisioned error interrupt status bit (TUNPVI) reports an attempted data transmission to an unprovisioned channel FIFO. TUNPVI is set high upon attempts to write data to an unprovisioned channel FIFO. TUNPVI remains valid when interrupts are disabled and may be polled to detect an attempt to write data to an unprovisioned channel FIFO.

TFOVRI:

The transmit FIFO overflow error interrupt status bit (TFOVRI) reports transmit FIFO overflow error interrupts to the microprocessor. TFOVRI is set high upon attempts to write data to the logical FIFO when it is already full. TFOVRI remains valid when interrupts are disabled and may be polled to detect transmit FIFO overflow events. (Note – Transmit FIFO overflows will not occur if channels are properly polled on the Transmit APPI before transferring data.)

TFUDRI:

The transmit FIFO underflow error interrupt status bit (TFUDRI) reports transmit FIFO underflow error interrupts to the microprocessor. TFUDRI is set high upon attempts to read data from the logical FIFO when it is already

empty. TFUDRI remains valid when interrupts are disabled and may be polled to detect transmit FIFO underflow events.

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

Bit	Type	Function	Default
Bit 15 to Bit 14		Unused	XH
Bit 13	R	TXCLKA	X
Bit 12	R	RXCLKA	X
Bit 11 to Bit 4		Unused	XH
Bit 3	R	C1FPA	X
Bit 2	R	FASTCLKA	X
Bit 1	R	REFCLKA	X
Bit 0	R	SYSCLKA	X

This register provides activity monitoring on the FREEDM-84A672 clock and SBI frame pulse inputs. When a monitored input makes a transition, the corresponding register bit is set high. The bit will remain high until this register is read, at which point, all the bits in this register are cleared. A lack of transitions is indicated by the corresponding register bit reading low. This register should be read periodically to detect for stuck at conditions.

Writing to this register delimits the accumulation intervals in the PMON accumulation registers. Counts accumulated in those registers are transferred to holding registers where they can be read. The counters themselves are then cleared to begin accumulating events for a new accumulation interval. The bits in this register are not affected by write accesses.

SYSCLKA:

The system clock active bit (SYSCLKA) monitors for low to high transitions on the SYSCLK input. SYSCLKA is set high on a rising edge of SYSCLK, and is set low when this register is read.

REFCLKA:

The SBI reference clock active bit (REFCLKA) monitors for low to high transitions on the REFCLK input. REFCLKA is set high on a rising edge of REFCLK, and is set low when this register is read.

FASTCLKA:

The SBI fast clock active bit (FASTCLKA) monitors for low to high transitions on the FASTCLK input. FASTCLKA is set high on a rising edge of FASTCLK, and is set low when this register is read.

C1FPA:

The SBI frame pulse active bit (C1FPA) monitors for low to high transitions on the C1FP input. C1FPA is set high on a rising edge of C1FP, and is set low when this register is read.

RXCLKA:

The Any-PHY receive clock active bit (RXCLKA) monitors for low to high transitions on the RXCLK input. RXCLKA is set high on a rising edge of RXCLK, and is set low when this register is read.

TXCLKA:

The Any-PHY transmit clock active bit (TXCLKA) monitors for low to high transitions on the TXCLK input. TXCLKA is set high on a rising edge of TXCLK, and is set low when this register is read.

Register 0x014 : FREEDM-84A672 Master Line Loopback

Bit	Type	Function	Default
Bit 15 to Bit 3	R/W	Reserved	0000H
Bit 2	R/W	LLBEN[2]	0
Bit 1	R/W	LLBEN[1]	0
Bit 0	R/W	LLBEN[0]	0

This register controls line loopback for the three serial data links (enabled when SPEn_EN is low).

LLBEN[2:0]:

The line loopback enable bits (LLBEN[2:0]) control line loopback for links #2 to #0. When LLBEN[n] is set high, the data on RD[n] is passed verbatim to TD[n] which is then updated on the falling edge of RCLK[n]. TCLK[n] is ignored. When LLBEN[n] is set low, TD[n] is processed normally.

Register 0x024 : FREEDM-84A672 Master Performance Monitor Control

Bit	Type	Function	Default
Bit 15		Unused	X
Bit 14	R/W	TP2EN	0
Bit 13	R/W	TABRT2EN	0
Bit 12	R/W	RP2EN	0
Bit 11	R/W	RLENE2EN	0
Bit 10	R/W	RABRT2EN	0
Bit 9	R/W	RFCSE2EN	0
Bit 8	R/W	RSPE2EN	0
Bit 7		Unused	X
Bit 6	R/W	TP1EN	0
Bit 5	R/W	TABRT1EN	0
Bit 4	R/W	RP1EN	0
Bit 3	R/W	RLENE1EN	0
Bit 2	R/W	RABRT1EN	0
Bit 1	R/W	RFCSE1EN	0
Bit 0	R/W	RSPE1EN	0

This register configures the events that are accumulated in the two configurable performance monitor counters in the PMON block.

RSPE1EN:

The receive small packet error accumulate enable bit (RSPE1EN) enables counting of minimum packet size violation events. When RSPE1EN is set high, receipt of a packet that is shorter than 32 bits (CRC-CCITT, Unspecified CRC or no CRC) or 48 bits (CRC-32) will cause the PMON Configurable Accumulator #1 register to increment. Small packet errors are ignored when RSPE1EN is set low.

RFCSE1EN:

The receive frame check sequence error accumulate enable bit (RFCSE1EN) enables counting of receive FCS error events. When RFCSE1EN is set high,

a mismatch between the received FCS code and the computed CRC residue will cause the PMON Configurable Accumulator #1 register to increment. Receive frame check sequence errors are ignored when RFCSE1EN is set low.

RABRT1EN:

The receive abort accumulate enable bit (RABRT1EN) enables counting of receive HDLC abort events. When RABRT1EN is set high, receipt of an abort code (at least 7 contiguous 1's) will cause the PMON Configurable Accumulator #1 register to increment. Receive aborts are ignored when RABRT1EN is set low.

RLENE1EN:

The receive packet length error accumulate enable bit (RLENE1EN) enables counting of receive packet length error events. When RLENE1EN is set high, receipt of a packet that is longer than the programmable maximum or of a packet that in not octet aligned will cause the PMON Configurable Accumulator #1 register to increment. (Receipt of a packet that is both too long and not octet aligned results in only one increment.) Receive packet length errors are ignored when RLENE1EN is set low.

RP1EN:

The receive packet enable bit (RP1EN) enables counting of receive error-free packets. When RP1EN is set high, receipt of an error-free packet will cause the PMON Configurable Accumulator #1 register to increment. Receive error-free packets are ignored when RP1EN is set low.

TABRT1EN:

The transmit abort accumulate enable bit (TABRT1EN) enables counting of transmit HDLC abort events. When TABRT1EN is set high, insertion of an abort in the outgoing stream will cause the PMON Configurable Accumulator #1 register to increment. Transmit aborts are ignored when TABRT1EN is set low.

TP1EN:

The transmit packet enable bit (TP1EN) enables counting of transmit error-free packets. When TP1EN is set high, transmission of an error-free packet will cause the PMON Configurable Accumulator #1 register to increment. Transmit error-free packets are ignored when TP1EN is set low.

RSPE2EN:

The receive small packet error accumulate enable bit (RSPE2EN) enables counting of minimum packet size violation events. When RSPE2EN is set high, receipt of a packet that is shorter than 32 bits (CRC-CCITT, Unspecified CRC or no CRC) or 48 bits (CRC-32) will cause the PMON Configurable Accumulator #2 register to increment. Small packet errors are ignored when RSPE2EN is set low.

RFCSE2EN:

The receive frame check sequence error accumulate enable bit (RFCSE2EN) enables counting of receive FCS error events. When RFCSE2EN is set high, a mismatch between the received FCS code and the computed CRC residue will cause the PMON Configurable Accumulator #2 register to increment. Receive frame check sequence errors are ignored when RFCSE2EN is set low.

RABRT2EN:

The receive abort accumulate enable bit (RABRT2EN) enables counting of receive HDLC abort events. When RABRT2EN is set high, receipt of an abort code (at least 7 contiguous 2's) will cause the PMON Configurable Accumulator #2 register to increment. Receive aborts are ignored when RABRT2EN is set low.

RLENE2EN:

The receive packet length error accumulate enable bit (RLENE2EN) enables counting of receive packet length error events. When RLENE2EN is set high, receipt of a packet that is longer than the programmable maximum or of a packet that in not octet aligned will cause the PMON Configurable Accumulator #2 register to increment. (Receipt of a packet that is both too long and not octet aligned results in only one increment.) Receive packet length errors are ignored when RLENE2EN is set low.

RP2EN:

The receive packet enable bit (RP2EN) enables counting of receive error-free packets. When RP2EN is set high, receipt of an error-free packet will cause the PMON Configurable Accumulator #2 register to increment. Receive error-free packets are ignored when RP2EN is set low.

TABRT2EN:

The transmit abort accumulate enable bit (TABRT2EN) enables counting of transmit HDLC abort events. When TABRT2EN is set high, insertion of an abort in the outgoing stream will cause the PMON Configurable Accumulator

#2 register to increment. Transmit aborts are ignored when TABRT2EN is set low.

TP2EN:

The transmit packet enable bit (TP2EN) enables counting of transmit error-free packets. When TP2EN is set high, transmission of an error-free packet will cause the PMON Configurable Accumulator #2 register to increment. Transmit error-free packets are ignored when TP2EN is set low.

Register 0x028 : FREEDM-84A672 Master SBI Interrupt Enable

Bit	Type	Function	Default
Bit 15 to Bit 1		Unused	XXH
Bit 0	R/W	SBIEXTE	0

This register provides interrupt enables for various events detected or initiated by the SBI circuitry within the FREEDM-84A672.

SBIEXTE:

The SBI Extracter interrupt enable bit (SBIEXTE) enables interrupts from the SBI Extract block to the microprocessor. When SBIEXTE is set high, an interrupt from the SBI Extract block will cause an interrupt to be generated on the INTB output. Interrupts are masked when SBIEXTE is set low. However, the SBIEXTI bit remains valid when interrupts are disabled and may be polled to detect interrupts from the SBI Extract Block.

Register 0x02C : FREEDM-84A672 Master SBI Interrupt Status

Bit	Type	Function	Default
Bit 15 to Bit 1		Unused	XXH
Bit 0	R	SBIEXTI	X

This register reports the interrupt status for various events detected or initiated by the SBI circuitry within the FREEDM-84A672. Reading this register acknowledges and clears the interrupts.

SBIEXTI:

The SBI Extracter interrupt status bit (SBIEXTI) reports an error condition from the SBI Extract block to the microprocessor. SBIEXTI remains valid when interrupts are disabled and may be polled to detect SBI Extract block error conditions.

Note

The only error condition which the SBI Extract block reports is a parity error on the SBI DROP BUS. If parity errors occur, the SBI EXTRACT Parity Error Interrupt Reason register (0x5DC) may be read to obtain more detailed information concerning the error.

Register 0x030 : FREEDM-84A672 Master Tributary Loopback #1

Bit	Type	Function	Default
Bit 15	R/W	SPE1_LBEN[16]	0
Bit 14	R/W	SPE1_LBEN[15]	0
Bit 13	R/W	SPE1_LBEN[14]	0
Bit 12	R/W	SPE1_LBEN[13]	0
Bit 11	R/W	SPE1_LBEN[12]	0
Bit 10	R/W	SPE1_LBEN[11]	0
Bit 9	R/W	SPE1_LBEN[10]	0
Bit 8	R/W	SPE1_LBEN[9]	0
Bit 7	R/W	SPE1_LBEN[8]	0
Bit 6	R/W	SPE1_LBEN[7]	0
Bit 5	R/W	SPE1_LBEN[6]	0
Bit 4	R/W	SPE1_LBEN[5]	0
Bit 3	R/W	SPE1_LBEN[4]	0
Bit 2	R/W	SPE1_LBEN[3]	0
Bit 1	R/W	SPE1_LBEN[2]	0
Bit 0	R/W	SPE1_LBEN[1]	0

This register controls line loopback for tributaries #1 to #16 of SPE #1.

SPE1_LBEN[16:1]:

The SPE #1 loopback enable bits (SPE1_LBEN[16:1]) control line loopback for tributaries #16 to #1 of SPE #1 of the SBI Interface. When SPE1_LBEN[n] is set high, the data on tributary #n output by the SBI PISO block is looped back to the tributary #n input of the SBI SIPO block. When SPE1_LBEN[n] is set low, transmit data for tributary #n is provided by the TCAS block (i.e. processed normally).

Register 0x034 : FREEDM-84A672 Master Tributary Loopback #2

Bit	Type	Function	Default
Bit 15	R/W	SPE2_LBEN[4]	0
Bit 14	R/W	SPE2_LBEN[3]	0
Bit 13	R/W	SPE2_LBEN[2]	0
Bit 12	R/W	SPE2_LBEN[1]	0
Bit 11	R/W	SPE1_LBEN[28]	0
Bit 10	R/W	SPE1_LBEN[27]	0
Bit 9	R/W	SPE1_LBEN[26]	0
Bit 8	R/W	SPE1_LBEN[25]	0
Bit 7	R/W	SPE1_LBEN[24]	0
Bit 6	R/W	SPE1_LBEN[23]	0
Bit 5	R/W	SPE1_LBEN[22]	0
Bit 4	R/W	SPE1_LBEN[21]	0
Bit 3	R/W	SPE1_LBEN[20]	0
Bit 2	R/W	SPE1_LBEN[19]	0
Bit 1	R/W	SPE1_LBEN[18]	0
Bit 0	R/W	SPE1_LBEN[17]	0

This register controls line loopback for tributaries #17 to #28 of SPE #1 and tributaries #1 to #4 of SPE #2.

SPE1_LBEN[28:17]:

The SPE #1 loopback enable bits (SPE1_LBEN[28:17]) control line loopback for tributaries #28 to #17 of SPE #1 of the SBI Interface. When SPE1_LBEN[n] is set high, the data on tributary #n output by the SBI PISO block is looped back to the tributary #n input of the SBI SIPO block. When SPE1_LBEN[n] is set low, transmit data for tributary #n is provided by the TCAS block (i.e. processed normally).

SPE2_LBEN[4:1]:

The SPE #2 loopback enable bits (SPE2_LBEN[4:1]) control line loopback for tributaries #4 to #1 of SPE #2 of the SBI Interface. When SPE2_LBEN[n] is

set high, the data on tributary #n output by the SBI PISO block is looped back to the tributary #n input of the SBI SIPO block. When SPE2_LBEN[n] is set low, transmit data for tributary #n is provided by the TCAS block (i.e. processed normally).

Register 0x038 : FREEDM-84A672 Master Tributary Loopback #3

Bit	Type	Function	Default
Bit 15	R/W	SPE2_LBEN[20]	0
Bit 14	R/W	SPE2_LBEN[19]	0
Bit 13	R/W	SPE2_LBEN[18]	0
Bit 12	R/W	SPE2_LBEN[17]	0
Bit 11	R/W	SPE2_LBEN[16]	0
Bit 10	R/W	SPE2_LBEN[15]	0
Bit 9	R/W	SPE2_LBEN[14]	0
Bit 8	R/W	SPE2_LBEN[13]	0
Bit 7	R/W	SPE2_LBEN[12]	0
Bit 6	R/W	SPE2_LBEN[11]	0
Bit 5	R/W	SPE2_LBEN[10]	0
Bit 4	R/W	SPE2_LBEN[9]	0
Bit 3	R/W	SPE2_LBEN[8]	0
Bit 2	R/W	SPE2_LBEN[7]	0
Bit 1	R/W	SPE2_LBEN[6]	0
Bit 0	R/W	SPE2_LBEN[5]	0

This register controls line loopback for tributaries #5 to #20 of SPE #2.

SPE2_LBEN[20:5]:

The SPE #2 loopback enable bits (SPE2_LBEN[20:5]) control line loopback for tributaries #20 to #5 of SPE #2 of the SBI Interface. When SPE2_LBEN[n] is set high, the data on tributary #n output by the SBI PISO block is looped back to the tributary #n input of the SBI SIPO block. When SPE2_LBEN[n] is set low, transmit data for tributary #n is provided by the TCAS block (i.e. processed normally).

Register 0x03C : FREEDM-84A672 Master Tributary Loopback #4

Bit	Type	Function	Default
Bit 15	R/W	SPE3_LBEN[8]	0
Bit 14	R/W	SPE3_LBEN[7]	0
Bit 13	R/W	SPE3_LBEN[6]	0
Bit 12	R/W	SPE3_LBEN[5]	0
Bit 11	R/W	SPE3_LBEN[4]	0
Bit 10	R/W	SPE3_LBEN[3]	0
Bit 9	R/W	SPE3_LBEN[2]	0
Bit 8	R/W	SPE3_LBEN[1]	0
Bit 7	R/W	SPE2_LBEN[28]	0
Bit 6	R/W	SPE2_LBEN[27]	0
Bit 5	R/W	SPE2_LBEN[26]	0
Bit 4	R/W	SPE2_LBEN[25]	0
Bit 3	R/W	SPE2_LBEN[24]	0
Bit 2	R/W	SPE2_LBEN[23]	0
Bit 1	R/W	SPE2_LBEN[22]	0
Bit 0	R/W	SPE2_LBEN[21]	0

This register controls line loopback for tributaries #21 to #28 of SPE #2 and tributaries #1 to #8 of SPE #3.

SPE3_LBEN[28:21]:

The SPE #2 loopback enable bits (SPE2_LBEN[28:21]) control line loopback for tributaries #28 to #21 of SPE #2 of the SBI Interface. When SPE2_LBEN[n] is set high, the data on tributary #n output by the SBI PISO block is looped back to the tributary #n input of the SBI SIPO block. When SPE2_LBEN[n] is set low, transmit data for tributary #n is provided by the TCAS block (i.e. processed normally).

SPE3_LBEN[8:1]:

The SPE #3 loopback enable bits (SPE3_LBEN[8:1]) control line loopback for tributaries #8 to #1 of SPE #3 of the SBI Interface. When SPE3_LBEN[n] is

set high, the data on tributary #n output by the SBI PISO block is looped back to the tributary #n input of the SBI SIPO block. When SPE3_LBEN[n] is set low, transmit data for tributary #n is provided by the TCAS block (i.e. processed normally).

Register 0x040 : FREEDM-84A672 Master Tributary Loopback #5

Bit	Type	Function	Default
Bit 15	R/W	SPE3_LBEN[24]	0
Bit 14	R/W	SPE3_LBEN[23]	0
Bit 13	R/W	SPE3_LBEN[22]	0
Bit 12	R/W	SPE3_LBEN[21]	0
Bit 11	R/W	SPE3_LBEN[20]	0
Bit 10	R/W	SPE3_LBEN[19]	0
Bit 9	R/W	SPE3_LBEN[18]	0
Bit 8	R/W	SPE3_LBEN[17]	0
Bit 7	R/W	SPE3_LBEN[16]	0
Bit 6	R/W	SPE3_LBEN[15]	0
Bit 5	R/W	SPE3_LBEN[14]	0
Bit 4	R/W	SPE3_LBEN[13]	0
Bit 3	R/W	SPE3_LBEN[12]	0
Bit 2	R/W	SPE3_LBEN[11]	0
Bit 1	R/W	SPE3_LBEN[10]	0
Bit 0	R/W	SPE3_LBEN[9]	0

This register controls line loopback for tributaries #9 to #24 of SPE #3.

SPE3_LBEN[24:9]:

The SPE #3 loopback enable bits (SPE3_LBEN[24:9]) control line loopback for tributaries #24 to #9 of SPE #3 of the SBI Interface. When SPE3_LBEN[n] is set high, the data on tributary #n output by the SBI PISO block is looped back to the tributary #n input of the SBI SIPO block. When SPE3_LBEN[n] is set low, transmit data for tributary #n is provided by the TCAS block (i.e. processed normally).

Register 0x044 : FREEDM-84A672 Master Tributary Loopback #6

Bit	Type	Function	Default
Bit 15 to Bit 4		Unused	X
Bit 3	R/W	SPE3_LBEN[28]	0
Bit 2	R/W	SPE3_LBEN[27]	0
Bit 1	R/W	SPE3_LBEN[26]	0
Bit 0	R/W	SPE3_LBEN[25]	0

This register controls line loopback for tributaries #25 to #28 of SPE #3.

SPE3_LBEN[28:25]:

The SPE #3 loopback enable bits (SPE3_LBEN[28:25]) control line loopback for tributaries #28 to #25 of SPE #3 of the SBI Interface. When SPE3_LBEN[n] is set high, the data on tributary #n output by the SBI PISO block is looped back to the tributary #n input of the SBI SIPO block. When SPE3_LBEN[n] is set low, transmit data for tributary #n is provided by the TCAS block (i.e. processed normally).

Register 0x048 : FREEDM-84A672 SBI DROP BUS Master Configuration

Bit	Type	Function	Default
Bit 15 to Bit 10		Unused	X
Bit 9	R/W	Reserved	0
Bit 8	R/W	Reserved	0
Bit 7	R/W	FCLK_FREQ[1]	0
Bit 6	R/W	FCLK_FREQ[0]	0
Bit 5	R/W	SPE3_TYP[1]	0
Bit 4	R/W	SPE3_TYP[0]	0
Bit 3	R/W	SPE2_TYP[1]	0
Bit 2	R/W	SPE2_TYP[0]	0
Bit 1	R/W	SPE1_TYP[1]	0
Bit 0	R/W	SPE1_TYP[0]	0

This register controls configures the operation of the SBI DROP BUS.

SPEn_TYP[1:0]:

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.

Table 12 – SPE Type Configuration

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

FCLK_FREQ[1:0]:

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.

Table 13 – FASTCLK Frequency Selection

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

Reserved:

The reserved bits must be set low for correct operation of the FREEDM-84A672 device.

Register 0x04C : FREEDM-84A672 SBI ADD BUS Master Configuration

Bit	Type	Function	Default
Bit 15 to Bit 14		Unused	X
Bit 13	R/W	PERM_DRV	0
Bit 12	R/W	Reserved	0
Bit 11	R/W	Reserved	0
Bit 10	R/W	Reserved	0
Bit 9	R/W	Reserved	0
Bit 8	R/W	Reserved	0
Bit 7	R/W	FCLK_FREQ[1]	0
Bit 6	R/W	FCLK_FREQ[0]	0
Bit 5	R/W	SPE3_TYP[1]	0
Bit 4	R/W	SPE3_TYP[0]	0
Bit 3	R/W	SPE2_TYP[1]	0
Bit 2	R/W	SPE2_TYP[0]	0
Bit 1	R/W	SPE1_TYP[1]	0
Bit 0	R/W	SPE1_TYP[0]	0

This register controls configures the operation of the SBI ADD BUS.

SPEn_TYP[1:0]:

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.

Table 14 – SPE Type Configuration

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

FCLK_FREQ[1:0]:

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.

Table 15 – FASTCLK Frequency Selection

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

Reserved:

The reserved bits must be set low for correct operation of the FREEDM-84A672 device.

PERM_DRV:

The Permanent Bus Driver selector bit (PERM_DRV) enables the FREEDM-84A672 device to drive the SBI ADD BUS continously. When set to 1, the FREEDM-84A672 will drive the bus and assert the AACTIVE output at all times, provided that the ADETECT[1:0] inputs are both 0. When set to 0, the FREEDM-84A672 will only drive the bus and assert AACTIVE when it has data to send (and when ADETECT[1:0] are both 0). PERM_DRV should only be set to 1 if the FREEDM-84A672 is the only device driving the SBI ADD bus and it is desired to prevent the bus tristating.

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

Bit	Type	Function	Default
Bit 15	R	BUSY	X
Bit 14	R/W	RWB	0
Bit 13		Unused	X
Bit 12	R/W	LINK[6]	0
Bit 11	R/W	LINK[5]	0
Bit 10	R/W	LINK[4]	0
Bit 9	R/W	LINK[3]	0
Bit 8	R/W	LINK[2]	0
Bit 7	R/W	LINK[1]	0
Bit 6	R/W	LINK[0]	0
Bit 5		Unused	X
Bit 4	R/W	TSLOT[4]	0
Bit 3	R/W	TSLOT[3]	0
Bit 2	R/W	TSLOT[2]	0
Bit 1	R/W	TSLOT[1]	0
Bit 0	R/W	TSLOT[0]	0

This register provides the receive link and time-slot number used to access the channel provision RAM. Writing to this register triggers an indirect register access.

TSLOT[4:0]:

The indirect time-slot number bits (TSLOT[4:0]) indicate the time-slot to be configured or interrogated in the indirect access. For a channelised T1/J1 link, time-slots 1 to 24 are valid. For a channelised E1 link, time-slots 1 to 31 are valid. For unchannelised or unframed links, only time-slot 0 is valid.

LINK[6:0]:

The indirect link number bits (LINK[6:0]) select amongst the 84 receive links to be configured or interrogated in the indirect access.

RWB:

The indirect access control bit (RWB) selects between a configure (write) or interrogate (read) access to the channel provision RAM. The address to the channel provision RAM is constructed by concatenating the TSLOT[4:0] and LINK[6:0] bits. Writing a logic zero to RWB triggers an indirect write operation. Data to be written is taken from the PROV, the CDLBEN and the CHAN[9:0] bits of the RCAS Indirect Channel Data register. Writing a logic one to RWB triggers an indirect read operation. Addressing of the RAM is the same as in an indirect write operation. The data read can be found in the PROV, the CDLBEN and the CHAN[9:0] bits of the RCAS Indirect Channel Data register.

BUSY:

The indirect access status bit (BUSY) reports the progress of an indirect access. BUSY is set high when this register is written to trigger an indirect access, and will stay high until the access is complete. At which point, BUSY will be set low. This register should be polled to determine when data from an indirect read operation is available in the RCAS Indirect Channel Data register or to determine when a new indirect write operation may commence.

Register 0x104 : RCAS Indirect Channel Data

Bit	Type	Function	Default
Bit 15	R/W	CDLBEN	0
Bit 14	R/W	PROV	0
Bit 13 to Bit 10		Unused	XH
Bit 9	R/W	CHAN[9]	0
Bit 8	R/W	CHAN[8]	0
Bit 7	R/W	CHAN[7]	0
Bit 6	R/W	CHAN[6]	0
Bit 5	R/W	CHAN[5]	0
Bit 4	R/W	CHAN[4]	0
Bit 3	R/W	CHAN[3]	0
Bit 2	R/W	CHAN[2]	0
Bit 1	R/W	CHAN[1]	0
Bit 0	R/W	CHAN[0]	0

This register contains the data read from the channel provision RAM after an indirect read operation or the data to be inserted into the channel provision RAM in an indirect write operation.

CHAN[9:0]:

The indirect data bits (CHAN[9:0]) report the channel number read from the channel provision RAM after an indirect read operation has completed. Channel number to be written to the channel provision RAM in an indirect write operation must be set up in this register before triggering the write. CHAN[9:0] reflects the value written until the completion of a subsequent indirect read operation.

PROV:

The indirect provision enable bit (PROV) reports the channel provision enable flag read from the channel provision RAM after an indirect read operation has completed. The provision enable flag to be written to the channel provision RAM, in an indirect write operation, must be set up in this register before

triggering the write. When PROV is set high, the current receive data byte is processed as part of the channel as indicated by CHAN[9:0]. When PROV is set low, the current time-slot does not belong to any channel and the receive data byte ignored. PROV reflects the value written until the completion of a subsequent indirect read operation.

CDLBEN:

The indirect channel based diagnostic loopback enable bit (CDLBEN) reports the loopback enable flag read from channel provision RAM after an indirect read operation has complete. The loopback enable flag to be written to the channel provision RAM, in an indirect write operation, must be set up in this register before triggering the write. When CDLBEN is set high, the current receive data byte is to be over-written by data retrieved from the loopback FIFO of the channel as indicated by CHAN[9:0]. When CDLBEN is set low, the current receive data byte is processed normally. CDLBEN reflects the value written until the completion of a subsequent indirect read operation.

Register 0x10C : RCAS Channel Disable

Bit	Type	Function	Default
Bit 15	R/W	CHDIS	0
Bit 14 to Bit 10		Unused	XXH
Bit 9	R/W	DCHAN[9]	0
Bit 8	R/W	DCHAN[8]	0
Bit 7	R/W	DCHAN[7]	0
Bit 6	R/W	DCHAN[6]	0
Bit 5	R/W	DCHAN[5]	0
Bit 4	R/W	DCHAN[4]	0
Bit 3	R/W	DCHAN[3]	0
Bit 2	R/W	DCHAN[2]	0
Bit 1	R/W	DCHAN[1]	0
Bit 0	R/W	DCHAN[0]	0

This register controls the disabling of one specific channel to allow orderly provisioning of time-slots associated with that channel.

DCHAN[9:0]:

The disable channel number bits (DCHAN[9:0]) selects the channel to be disabled. When CHDIS is set high, the channel specified by DCHAN[9:0] is disabled. Data in time-slots associated with the specified channel is ignored. When CHDIS is set low, the channel specified by DCHAN[9:0] operates normally.

CHDIS:

The channel disable bit (CHDIS) controls the disabling of the channels specified by DCHAN[9:0]. When CHDIS is set high, the channel selected by DCHAN[9:0] is disabled. Data in time-slots associated with the specified channel is ignored. When CHDIS is set low, the channel specified by DCHAN[9:0] operates normally.

Register 0x140 : RCAS SBI SPE1 Configuration Register #1

Bit	Type	Function	Default
Bit 15	R/W	FEN[11]	0
Bit 14	R/W	FEN[10]	0
Bit 13	R/W	FEN[9]	0
Bit 12	R/W	FEN[8]	0
Bit 11	R/W	FEN[7]	0
Bit 10	R/W	FEN[6]	0
Bit 9	R/W	FEN[5]	0
Bit 8	R/W	FEN[4]	0
Bit 7	R/W	FEN[3]	0
Bit 6	R/W	FEN[2]	0
Bit 5	R/W	FEN[1]	0
Bit 4	R/W	FEN[0]	0
Bit 3		Unused	X
Bit 2	R/W	SBI_MODE[2]	0
Bit 1	R/W	SBI_MODE[1]	0
Bit 0	R/W	SBI_MODE[0]	0

This register configures the operational mode of receive links 0, 3, 6, 9, … 33, 36, 39, …81, i.e. those links mapped to SPE 1 of the SBI DROP BUS.

SBI_MODE[2:0]:

The SBI mode select bits (SBI_MODE[2:0]) configure the receive links of SPE1, as shown in the following table:

Table 16 – SBI Mode SPE1 Configuration

SBI_MODE [2:0]	SPE1 Configuration
000	Single unchannelised DS-3 on link 0
001	28 T1/J1 links
010	21 E1 links (links 63, 66, 69, … , 81 are unused)
011	Reserved
100	Reserved
101	Reserved
110	Reserved
111	Reserved

FEN[11:0]

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.

Register 0x144 : RCAS SBI SPE1 Configuration Register #2

Bit	Type	Function	Default
Bit 15	R/W	FEN[27]	0
Bit 14	R/W	FEN[26]	0
Bit 13	R/W	FEN[25]	0
Bit 12	R/W	FEN[24]	0
Bit 11	R/W	FEN[23]	0
Bit 10	R/W	FEN[22]	0
Bit 9	R/W	FEN[21]	0
Bit 8	R/W	FEN[20]	0
Bit 7	R/W	FEN[19]	0
Bit 6	R/W	FEN[18]	0
Bit 5	R/W	FEN[17]	0
Bit 4	R/W	FEN[16]	0
Bit 3	R/W	FEN[15]	0
Bit 2	R/W	FEN[14]	0
Bit 1	R/W	FEN[13]	0
Bit 0	R/W	FEN[12]	0

The bits of this register set are used to configure the framing modes of receive links 36, 39, 42 … 81.

FEN[27:12]:

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.

Register 0x148 : RCAS SBI SPE2 Configuration Register #1

Bit	Type	Function	Default
Bit 15	R/W	FEN[11]	0
Bit 14	R/W	FEN[10]	0
Bit 13	R/W	FEN[9]	0
Bit 12	R/W	FEN[8]	0
Bit 11	R/W	FEN[7]	0
Bit 10	R/W	FEN[6]	0
Bit 9	R/W	FEN[5]	0
Bit 8	R/W	FEN[4]	0
Bit 7	R/W	FEN[3]	0
Bit 6	R/W	FEN[2]	0
Bit 5	R/W	FEN[1]	0
Bit 4	R/W	FEN[0]	0
Bit 3		Unused	X
Bit 2	R/W	SBI_MODE[2]	0
Bit 1	R/W	SBI_MODE[1]	0
Bit 0	R/W	SBI_MODE[0]	0

This register configures the operational mode of receive links 1, 4, 7, 10, … 34, 37, …82, i.e. those links mapped to SPE 2 of the SBI DROP BUS.

SBI_MODE[2:0]:

The SBI mode select bits (SBI_MODE[2:0]) configure the receive links of SPE2, as shown in the following table:

Table 17 – SBI Mode SPE2 Configuration

SBI_MODE [2:0]	SPE2 Configuration
000	Single unchannelised DS-3 on link 1
001	28 T1/J1 links
010	21 E1 links (links 64, 67, 70, … , 82 are unused)
011	Reserved
100	Reserved
101	Reserved
110	Reserved
111	Reserved

FEN[11:0]

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.

Register 0x14C : RCAS SBI SPE2 Configuration Register #2

Bit	Type	Function	Default
Bit 15	R/W	FEN[27]	0
Bit 14	R/W	FEN[26]	0
Bit 13	R/W	FEN[25]	0
Bit 12	R/W	FEN[24]	0
Bit 11	R/W	FEN[23]	0
Bit 10	R/W	FEN[22]	0
Bit 9	R/W	FEN[21]	0
Bit 8	R/W	FEN[20]	0
Bit 7	R/W	FEN[19]	0
Bit 6	R/W	FEN[18]	0
Bit 5	R/W	FEN[17]	0
Bit 4	R/W	FEN[16]	0
Bit 3	R/W	FEN[15]	0
Bit 2	R/W	FEN[14]	0
Bit 1	R/W	FEN[13]	0
Bit 0	R/W	FEN[12]	0

The bits of this register set are used to configure the framing modes of receive links 37, 40, 43 … 82.

FEN[27:12]:

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.

Register 0x150 : RCAS SBI SPE3 Configuration Register #1

Bit	Type	Function	Default
Bit 15	R/W	FEN[11]	0
Bit 14	R/W	FEN[10]	0
Bit 13	R/W	FEN[9]	0
Bit 12	R/W	FEN[8]	0
Bit 11	R/W	FEN[7]	0
Bit 10	R/W	FEN[6]	0
Bit 9	R/W	FEN[5]	0
Bit 8	R/W	FEN[4]	0
Bit 7	R/W	FEN[3]	0
Bit 6	R/W	FEN[2]	0
Bit 5	R/W	FEN[1]	0
Bit 4	R/W	FEN[0]	0
Bit 3		Unused	X
Bit 2	R/W	SBI_MODE[2]	0
Bit 1	R/W	SBI_MODE[1]	0
Bit 0	R/W	SBI_MODE[0]	0

This register configures the operational mode of receive links 2, 5, 8, 11, … 35, 38, …83, i.e. those links mapped to SPE 3 of the SBI DROP BUS.

SBI_MODE[2:0]:

The SBI mode select bits (SBI_MODE[2:0]) configure the receive links of SPE3, as shown in the following table:

Table 18 – SBI Mode SPE3 Configuration

SBI_MODE [2:0]	SPE3 Configuration
000	Single unchannelised DS-3 on link 2
001	28 T1/J1 links
010	21 E1 links (links 65, 68, 71, … , 83 are unused)
011	Reserved
100	Reserved
101	Reserved
110	Reserved
111	Reserved

FEN[11:0]

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.

Register 0x154 : RCAS SBI SPE3 Configuration Register #2

Bit	Type	Function	Default
Bit 15	R/W	FEN[27]	0
Bit 14	R/W	FEN[26]	0
Bit 13	R/W	FEN[25]	0
Bit 12	R/W	FEN[24]	0
Bit 11	R/W	FEN[23]	0
Bit 10	R/W	FEN[22]	0
Bit 9	R/W	FEN[21]	0
Bit 8	R/W	FEN[20]	0
Bit 7	R/W	FEN[19]	0
Bit 6	R/W	FEN[18]	0
Bit 5	R/W	FEN[17]	0
Bit 4	R/W	FEN[16]	0
Bit 3	R/W	FEN[15]	0
Bit 2	R/W	FEN[14]	0
Bit 1	R/W	FEN[13]	0
Bit 0	R/W	FEN[12]	0

The bits of this register set are used to configure the framing modes of receive links 38, 41, 44 … 83.

FEN[27:12]:

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.

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

Bit	Type	Function	Default
Bit 15 to Bit 5		Unused	XXXH
Bit 4	R/W	Reserved	0
Bit 3		Unused	X
Bit 2	R/W	Reserved	0
Bit 1	R/W	Reserved	0
Bit 0	R/W	Reserved	0

This register controls 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 is low).

Reserved:

The reserved bits must be set low for correct operation of the FREEDM-84A672 device.

Register 0x200 : RHDL Indirect Channel Select

Bit	Type	Function	Default
Bit 15	R	BUSY	X
Bit 14	R/W	CRWB	0
Bit 13 to Bit 10		Unused	XH
Bit 9	R/W	CHAN[9]	0
Bit 8	R/W	CHAN[8]	0
Bit 7	R/W	CHAN[7]	0
Bit 6	R/W	CHAN[6]	0
Bit 5	R/W	CHAN[5]	0
Bit 4	R/W	CHAN[4]	0
Bit 3	R/W	CHAN[3]	0
Bit 2	R/W	CHAN[2]	0
Bit 1	R/W	CHAN[1]	0
Bit 0	R/W	CHAN[0]	0

This register provides the channel number used to access the receive channel provision RAM. Writing to this register triggers an indirect channel register access.

CHAN[9:0]:

The indirect channel number bits (CHAN[9:0]) indicate the receive channel to be configured or interrogated in the indirect access.

CRWB:

The channel indirect access control bit (CRWB) selects between a configure (write) or interrogate (read) access to the receive channel provision RAM. Writing a logic zero to CRWB triggers an indirect write operation. Data to be written is taken from the Indirect Channel Data registers. Writing a logic one to CRWB triggers an indirect read operation. The data read can be found in the Indirect Channel Data registers.

BUSY:

The indirect access status bit (BUSY) reports the progress of an indirect access. BUSY is set high when this register is written to trigger an indirect access, and will stay high until the access is complete. At which point, BUSY will be set low. This register should be polled to determine when data from an indirect read operation is available in the RHDL Indirect Channel Data #1 and #2 registers or to determine when a new indirect write operation may commence.

Register 0x204 : RHDL Indirect Channel Data Register #1

Bit	Type	Function	Default
Bit 15	R/W	PROV	0
Bit 14	R/W	STRIP	0
Bit 13	R/W	DELIN	0
Bit 12	R	TAVAIL	X
Bit 11	W	Reserved	X
Bit 10	W	FPTR[10]	X
Bit 9	W	FPTR[9]	X
Bit 8	W	FPTR[8]	X
Bit 7	W	FPTR[7]	X
Bit 6	W	FPTR[6]	X
Bit 5	W	FPTR[5]	X
Bit 4	W	FPTR[4]	X
Bit 3	W	FPTR[3]	X
Bit 2	W	FPTR[2]	X
Bit 1	W	FPTR[1]	X
Bit 0	W	FPTR[0]	X

This register contains data read from the channel provision RAM after an indirect read operation or data to be inserted into the channel provision RAM in an indirect write operation.

FPTR[10:0]:

The indirect FIFO block pointer (FPTR[10:0]) identifies one of the blocks of the circular linked list in the partial packet buffer used in the logical FIFO of the current channel. The FIFO pointer to be written to the channel provision RAM, in an indirect write operation, must be set up in this register before triggering the write. The FIFO pointer value can be any one of the blocks provisioned to form the circular buffer.

Reserved:

The reserved bit must be set low for correct operation of the FREEDM-84A672 device.

TAVAIL:

The indirect transaction available bit (TAVAIL) reports the fill level of the partial packet buffer used in the logical FIFO of the current channel. TAVAIL is set high when the FIFO of the current channel contains sufficient data, as controlled by XFER[3:0], to result in a transfer across the receive APPI. TAVAIL is set low when the amount of receive data is too small to result in a transfer across the receive APPI. TAVAIL is updated by an indirect channel read operation.

DELIN:

The indirect delineate enable bit (DELIN) configures the HDLC processor to perform flag sequence delineation and bit de-stuffing on the incoming data stream. The delineate enable bit to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. When DELIN is set high, flag sequence delineation and bit de-stuffing is performed on the incoming data stream. When DELIN is set low, the HDLC processor does not perform any processing (flag sequence delineation, bit de-stuffing nor CRC verification) on the incoming stream. DELIN reflects the value written until the completion of a subsequent indirect channel read operation.

STRIP:

The indirect frame check sequence discard bit (STRIP) configures the HDLC processor to remove the CRC from the incoming frame when writing the data to the channel FIFO. The FCS discard bit to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. When STRIP is set high and CRC[1:0] is not equal to “00”, the received CRC value is not written to the FIFO. When STRIP is set low, the received CRC value is written to the FIFO. The bytes in buffer field of the RPD correctly reflect the presence/absence of CRC bytes in the buffer. The value of STRIP is ignored when DELIN is low. STRIP reflects the value written until the completion of a subsequent indirect channel read operation.

PROV:

The indirect provision enable bit (PROV) reports the channel provision enable flag read from the channel provision RAM after an indirect channel read operation has completed. The provision enable flag to be written to the

channel provision RAM, in an indirect write operation, must be set up in this register before triggering the write. When PROV is set high, the HDLC processor will process data on the channel specified by CHAN[9:0]. When PROV is set low, the HDLC processor will ignore data on the channel specified by CHAN[9:0]. PROV reflects the value written until the completion of a subsequent indirect channel read operation.

Register 0x208 : RHDL Indirect Channel Data Register #2

Bit	Type	Function	Default
Bit 15	R/W	7BIT	0
Bit 14	R/W	PRIORITY	0
Bit 13	R/W	INVERT	0
Bit 12		Unused	X
Bit 11	R/W	CRC[1]	0
Bit 10	R/W	CRC[0]	0
Bit 9	R/W	OFFSET[1]	0
Bit 8	R/W	OFFSET[0]	0
Bit 7		Unused	X
Bit 6		Unused	X
Bit 5		Unused	X
Bit 4		Unused	X
Bit 3	R/W	XFER[3]	0
Bit 2	R/W	XFER[2]	0
Bit 1	R/W	XFER[1]	0
Bit 0	R/W	XFER[0]	0

This register contains data read from the channel provision RAM after an indirect read operation or data to be inserted into the channel provision RAM in an indirect write operation.

XFER[3:0]:

The indirect channel transfer size (XFER[3:0]) configures the amount of data transferred in each transaction. The channel transfer size to be written to the channel provision RAM, in an indirect write operation, must be set up in this register before triggering the write. 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. Channel transfer size is measured in 16 byte blocks. The amount of data transferred and the depth threshold are specified by given setting is:

XFER[3:0] + 1 blocks = 16 * (XFER[3:0] + 1) bytes

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. XFER[3:0] reflects the value written until the completion of a subsequent indirect channel read operation.

OFFSET[1:0]:

The packet byte offset (OFFSET[1:0]) configures the partial packet processor to insert invalid bytes at the beginning of a packet stored in the channel FIFO. The value of OFFSET[1:0] to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. The number of bytes inserted before the beginning of a HDLC packet is defined by the binary value of OFFSET[1:0]. OFFSET[1:0] reflects the value written until the completion of a subsequent indirect channel read operation.

CRC[1:0]:

The CRC algorithm bits (CRC[1:0]) configures the HDLC processor to perform CRC verification on the incoming data stream. The value of CRC[1:0] to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. CRC[1:0] is ignored when DELIN is low. CRC[1:0] reflects the value written until the completion of a subsequent indirect channel read operation.

Table 19 – CRC[1:0] Settings

CRC[1]	CRC[0]	Operation
0	0	No Verification
0	1	CRC-CCITT
1	0	CRC-32
1	1	Reserved

INVERT:

The HDLC data inversion bit (INVERT) configures the HDLC processor to logically invert the incoming HDLC stream from the RCAS672 before processing it. The value of INVERT to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. When INVERT is set to one, the HDLC stream is logically inverted before processing. When INVERT is set to zero, the HDLC

stream is not inverted before processing. INVERT reflects the value written until the completion of a subsequent indirect channel read operation.

PRIORITY:

The channel FIFO priority bit (PRIORITY) informs the partial packet processor that the channel has precedence over other channels when being serviced by the RAPI672 block for transfer across the receive APPI. The value of PRIORITY to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. Channel FIFOs with PRIORITY set to one are serviced by the RAPI672 before channel FIFOs with PRIORITY set to zero. Channels with an HDLC data rate to FIFO size ratio that is significantly higher than other channels should have PRIORITY set to one. PRIORITY reflects the value written until the completion of a subsequent indirect channel read operation.

7BIT:

The 7BIT enable bit (7BIT) configures the HDLC processor to ignore the least significant bit of each octet in the incoming channel stream. The value of 7BIT to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. When 7BIT is set high, the least significant bit (last bit of each octet received), is ignored. When 7BIT is set low, the entire receive data stream is processed. 7BIT reflects the value written until the completion of a subsequent indirect channel read operation.

Register 0x210 : RHDL Indirect Block Select

Bit	Type	Function	Default
Bit 15	R	BUSY	X
Bit 14	R/W	BRWB	X
Bit 13		Unused	X
Bit 12		Unused	X
Bit 11	R/W	Reserved	X
Bit 10	R/W	BLOCK[10]	X
Bit 9	R/W	BLOCK[9]	X
Bit 8	R/W	BLOCK[8]	X
Bit 7	R/W	BLOCK[7]	X
Bit 6	R/W	BLOCK[6]	X
Bit 5	R/W	BLOCK[5]	X
Bit 4	R/W	BLOCK[4]	X
Bit 3	R/W	BLOCK[3]	X
Bit 2	R/W	BLOCK[2]	X
Bit 1	R/W	BLOCK[1]	X
Bit 0	R/W	BLOCK[0]	X

This register provides the block number used to access the block pointer RAM. Writing to this register triggers an indirect block register access.

BLOCK[10:0]:

The indirect block number (BLOCK[10:0]) indicate the block to be configured or interrogated in the indirect access.

Reserved:

The reserved bit must be set low for correct operation of the FREEDM-84A672 device.

BRWB:

The block indirect access control bit (BRWB) selects between a configure (write) or interrogate (read) access to the block pointer RAM. Writing a logic

zero to BRWB triggers an indirect block write operation. Data to be written is taken from the Indirect Block Data register. Writing a logic one to BRWB triggers an indirect block read operation. The data read can be found in the Indirect Block Data register.

BUSY:

The indirect access status bit (BUSY) reports the progress of an indirect access. BUSY is set high when this register is written to trigger an indirect access, and will stay high until the access is complete. At which point, BUSY will be set low. This register should be polled to determine when data from an indirect read operation is available in the RHDL Indirect Block Data register or to determine when a new indirect write operation may commence.

Register 0x214 : RHDL Indirect Block Data

Bit	Type	Function	Default
Bit 15 to Bit 12		Unused	XH
Bit 11	R/W	Reserved	X
Bit 10	R/W	BPTR[10]	0
Bit 9	R/W	BPTR[9]	0
Bit 8	R/W	BPTR[8]	0
Bit 7	R/W	BPTR[7]	0
Bit 6	R/W	BPTR[6]	0
Bit 5	R/W	BPTR[5]	0
Bit 4	R/W	BPTR[4]	0
Bit 3	R/W	BPTR[3]	0
Bit 2	R/W	BPTR[2]	0
Bit 1	R/W	BPTR[1]	0
Bit 0	R/W	BPTR[0]	0

This register contains data read from the block pointer RAM after an indirect block read operation or data to be inserted into the block pointer RAM in an indirect block write operation.

BPTR[10:0]:

The indirect block pointer (BPTR[10:0]) configures the block pointer of the block specified by the Indirect Block Select register. The block pointer to be written to the block pointer RAM, in an indirect write operation, must be set up in this register before triggering the write. The block pointer value is the block number of the next block in the linked list. A circular list of blocks must be formed in order to use the block list as a receive channel FIFO buffer. BPTR[10:0] reflects the value written until the completion of a subsequent indirect block read operation. When provisioning a channel FIFO, all block pointers must be re-written to properly initialize the FIFO.

Reserved:

The reserved bit must be set low for correct operation of the FREEDM-84A672 device.

Register 0x220 : RHDL Configuration

Bit	Type	Function	Default
Bit 15 to Bit 10		Unused	XXH
Bit 9	R/W	LENCHK	0
Bit 8	R/W	TSTD	0
Bit 7		Unused	X
Bit 6		Unused	X
Bit 5		Unused	X
Bit 4		Unused	X
Bit 3		Unused	X
Bit 2		Unused	X
Bit 1		Unused	X
Bit 0		Unused	X

This register configures all provisioned receive channels.

TSTD:

The telecom standard bit (TSTD) controls the bit ordering of the HDLC data transferred across the receive APPI. When TSTD is set low, the least significant bit of each byte on the receive APPI bus (AD[0] and AD[8]) is the first HDLC bit received and the most significant bit of each byte (AD[7] and AD[15]) is the last HDLC bit received (datacom standard). When TSTD is set high, AD[0] and AD[8] are the last HDLC bits received and AD[7] and AD[15] are the first HDLC bits received (telecom standard).

LENCHK:

The packet length error check bit (LENCHK) controls the checking of receive packets that are longer than the maximum programmed length. When LENCHK is set high, receive packets are aborted and the remainder of the frame discarded when the packet exceeds the maximum packet length given by MAX[15:0]. When LENCHK is set low, receive packets are not checked for maximum size and MAX[15:0] must be set to ‘hFFFF.

Register 0x224 : RHDL Maximum Packet Length

Bit	Type	Function	Default
Bit 15	R/W	MAX[15]	1
Bit 14	R/W	MAX[14]	1
Bit 13	R/W	MAX[13]	1
Bit 12	R/W	MAX[12]	1
Bit 11	R/W	MAX[11]	1
Bit 10	R/W	MAX[10]	1
Bit 9	R/W	MAX[9]	1
Bit 8	R/W	MAX[8]	1
Bit 7	R/W	MAX[7]	1
Bit 6	R/W	MAX[6]	1
Bit 5	R/W	MAX[5]	1
Bit 4	R/W	MAX[4]	1
Bit 3	R/W	MAX[3]	1
Bit 2	R/W	MAX[2]	1
Bit 1	R/W	MAX[1]	1
Bit 0	R/W	MAX[0]	1

This register configures the maximum legal HDLC packet byte length.

MAX[15:0]:

The maximum HDLC packet length (MAX[15:0]) configures the FREEDM-84A672 to reject HDLC packets longer than a maximum size when LENCHK is set high. Receive packets with total length, including address, control, information and FCS fields, greater than MAX[15:0] bytes are aborted. When LENCHK is set low, aborts are not generated regardless of packet length and MAX[15:0] must be set to ‘hFFFF.

Register 0x380 : THDL Indirect Channel Select

Bit	Type	Function	Default
Bit 15	R	BUSY	X
Bit 14	R/W	CRWB	0
Bit 13 to Bit 10		Unused	XH
Bit 9	R/W	CHAN[9]	0
Bit 8	R/W	CHAN[8]	0
Bit 7	R/W	CHAN[7]	0
Bit 6	R/W	CHAN[6]	0
Bit 5	R/W	CHAN[5]	0
Bit 4	R/W	CHAN[4]	0
Bit 3	R/W	CHAN[3]	0
Bit 2	R/W	CHAN[2]	0
Bit 1	R/W	CHAN[1]	0
Bit 0	R/W	CHAN[0]	0

This register provides the channel number used to access the transmit channel provision RAM. Writing to this register triggers an indirect channel register access.

CHAN[9:0]:

The indirect channel number bits (CHAN[9:0]) indicate the channel to be configured or interrogated in the indirect access.

CRWB:

The channel indirect access control bit (CRWB) selects between a configure (write) or interrogate (read) access to the channel provision RAM. Writing a logic zero to CRWB triggers an indirect write operation. Data to be written is taken from the Indirect Channel Data registers. Writing a logic one to CRWB triggers an indirect read operation. The data read can be found in the Indirect Channel Data registers.

BUSY:

The indirect access status bit (BUSY) reports the progress of an indirect access. BUSY is set high when this register is written to trigger an indirect access, and will stay high until the access is complete. At which point, BUSY will be set low. This register should be polled to determine when data from an indirect read operation is available in the THDL Indirect Channel Data #1, #2 and #3 registers or to determine when a new indirect write operation may commence.

Register 0x384 : THDL Indirect Channel Data #1

Bit	Type	Function	Default
Bit 15	R/W	PROV	0
Bit 14	R/W	CRC[1]	0
Bit 13	R/W	CRC[0]	0
Bit 12	R/W	DELIN	0
Bit 11	W	Reserved	X
Bit 10	W	FPTR[10]	0
Bit 9	W	FPTR[9]	0
Bit 8	W	FPTR[8]	0
Bit 7	W	FPTR[7]	0
Bit 6	W	FPTR[6]	0
Bit 5	W	FPTR[5]	0
Bit 4	W	FPTR[4]	0
Bit 3	W	FPTR[3]	0
Bit 2	W	FPTR[2]	0
Bit 1	W	FPTR[1]	0
Bit 0	W	FPTR[0]	0

This register contains data read from the channel provision RAM after an indirect channel read operation or data to be inserted into the channel provision RAM in an indirect channel write operation.

FPTR[10:0]:

The indirect FIFO block pointer (FPTR[10:0]) informs the partial packet buffer processor about the circular linked list of blocks to use for a FIFO for the channel. The FIFO pointer to be written to the channel provision RAM, in an indirect write operation, must be set up in this register before triggering the write. The FIFO pointer value can be any one of the block numbers provisioned, by indirect block write operations, to form the circular buffer.

Reserved:

The reserved bit must be set low for correct operation of the FREEDM-84A672 device.

DELIN:

The indirect delineate enable bit (DELIN) configures the HDLC processor to perform flag sequence insertion and bit stuffing on the outgoing data stream. The delineate enable bit to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. When DELIN is set high, flag sequence insertion, bit stuffing and ,optionally, CRC generation is performed on the outgoing HDLC data stream. When DELIN is set low, the HDLC processor does not perform any processing (flag sequence insertion, bit stuffing nor CRC generation) on the outgoing stream. DELIN reflects the value written until the completion of a subsequent indirect channel read operation.

CRC[1:0]:

The CRC algorithm (CRC[1:0]) configures the HDLC processor to perform CRC generation on the outgoing HDLC data stream. The value of CRC[1:0] to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. CRC[1:0] is ignored when DELIN is low. CRC[1:0] reflects the value written until the completion of a subsequent indirect channel read operation.

Table 20 – CRC[1:0] Settings

CRC[1]	CRC[0]	Operation
0	0	No CRC
0	1	CRC-CCITT
1	0	CRC-32
1	1	Reserved

PROV:

The indirect provision enable bit (PROV) reports the channel provision enable flag read from the channel provision RAM after an indirect channel read operation has completed. The provision enable flag to be written to the channel provision RAM, in an indirect write operation, must be set up in this register before triggering the write. When PROV is set high, the HDLC processor will service requests for data from the TCAS672 block. When PROV is set low, the HDLC processor will ignore requests from the TCAS672

block. PROV reflects the value written until the completion of a subsequent indirect channel read operation.

Register 0x388 : THDL Indirect Channel Data #2

Bit	Type	Function	Default
Bit 15	R/W	7BIT	0
Bit 14	R/W	Reserved	0
Bit 13	R/W	INVERT	0
Bit 12	R/W	DFCS	0
Bit 11	W	Reserved	0
Bit 10	W	FLEN[10]	0
Bit 9	W	FLEN[9]	0
Bit 8	W	FLEN[8]	0
Bit 7	W	FLEN[7]	0
Bit 6	W	FLEN[6]	0
Bit 5	W	FLEN[5]	0
Bit 4	W	FLEN[4]	0
Bit 3	W	FLEN[3]	0
Bit 2	W	FLEN[2]	0
Bit 1	W	FLEN[1]	0
Bit 0	W	FLEN[0]	0

This register contains data to be inserted into the channel provision RAM in an indirect write operation.

FLEN[10:0]:

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. The FIFO length to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write.

Reserved:

The reserved bits must be set low for correct operation of the FREEDM-84A672 device.

DFCS:

The diagnose frame check sequence bit (DFCS) controls the inversion of the FCS field inserted into the transmit packet. The value of DFCS to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. When DFCS is set to one, the FCS field in the outgoing HDLC stream is logically inverted allowing diagnosis of downstream FCS verification logic. The outgoing FCS field is not inverted when DFCS is set to zero. DFCS reflects the value written until the completion of a subsequent indirect channel read operation.

INVERT:

The HDLC data inversion bit (INVERT) configures the HDLC processor to logically invert the outgoing HDLC stream. The value of INVERT to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. When INVERT is set to one, the outgoing HDLC stream is logically inverted. The outgoing HDLC stream is not inverted when INVERT is set to zero. INVERT reflects the value written until the completion of a subsequent indirect channel read operation.

7BIT:

The least significant stuff enable bit (7BIT) configures the HDLC processor to stuff the least significant bit of each octet in the outgoing channel stream. The value of 7BIT to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. When 7BIT is set high, 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. When 7BIT is set low, the entire octet contains valid data and BIT8 is ignored. 7BIT reflects the value written until the completion of a subsequent indirect channel read operation.

Register 0x38C : THDL Indirect Channel Data #3

Bit	Type	Function	Default
Bit 15	R/W	TRANS	0
Bit 14	R/W	IDLE	0
Bit 13		Unused	X
Bit 12		Unused	X
Bit 11	R/W	LEVEL[3]	0
Bit 10	R/W	LEVEL[2]	0
Bit 9	R/W	LEVEL[1]	0
Bit 8	R/W	LEVEL[0]	0
Bit 7	R/W	FLAG[2]	0
Bit 6	R/W	FLAG[1]	0
Bit 5	R/W	FLAG[0]	0
Bit 4		Unused	X
Bit 3	R/W	XFER[3]	0
Bit 2	R/W	XFER[2]	0
Bit 1	R/W	XFER[1]	0
Bit 0	R/W	XFER[0]	0

This register contains data read from the channel provision RAM after an indirect read operation or data to be inserted into the channel provision RAM in an indirect write operation.

XFER[3:0]:

The indirect channel transfer size (XFER[3:0]) specifies the amount of data the partial packet processor requests from the TAPI672 block. The channel transfer size to be written to the channel provision RAM, in an indirect write operation, must be set up in this register before triggering the write. 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. FIFO free space and transfer size are measured in number of 16-byte blocks. XFER[3:0] reflects the value written until the completion of a subsequent indirect channel read operation.

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.

FLAG[2:0]:

The flag insertion control (FLAG[2:0]) configures the minimum number of flags or bytes of idle bits the HDLC processor inserts between HDLC packets. The value of FLAG[2:0] to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. The minimum number of flags or bytes of idle (8 bits of 1's) inserted between HDLC packets is shown in the table below. FLAG[2:0] reflects the value written until the completion of a subsequent indirect channel read operation.

Table 21 – FLAG[2:0] Settings

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

LEVEL[3:0]:

The indirect channel FIFO trigger level (LEVEL[3:0]), in concert with the TRANS bit, configure the various channel FIFO free space levels which trigger the HDLC processor to start transmission of a HDLC packet as well as trigger the partial packet buffer to request data from the TAPI672 as shown in the following table. The channel FIFO trigger level to be written to the channel provision RAM, in an indirect write operation, must be set up in this register before triggering the write. LEVEL[3:0] reflects the value written until the completion of a subsequent indirect channel read operation.

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 greater than or equal to 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 expedited requests to the TAPI672 to retrieve XFER[3:0] + 1 blocks of data.

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. The starving trigger level must always be set to a number of blocks greater than or equal to the channel transfer size.

IDLE:

The interframe time fill bit (IDLE) configures the HDLC processor to use flag bytes or HDLC idle as the interframe time fill between HDLC packets. The value of IDLE to be written to the channel provision RAM, in an indirect channel write operation, must be set up in this register before triggering the write. When IDLE is set low, the HDLC processor uses flag bytes as the interframe time fill. When IDLE is set high, the HDLC processor uses HDLC idle (all one's bit with no bit-stuffing pattern is transmitted) as the interframe time fill. IDLE reflects the value written until the completion of a subsequent indirect channel read operation.

TRANS:

The indirect transmission start bit (TRANS), in concert with the LEVEL[3:0] bits, configure the various channel FIFO free space levels which trigger the HDLC processor to start transmission of a HDLC packet as well as trigger the partial packet buffer to request data from the TAPI672 as shown in the following table. The transmission start mode to be written to the channel provision RAM, in an indirect write operation, must be set up in this register before triggering the write. TRANS reflects the value written until the completion of a subsequent indirect channel read operation.

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 greater than or equal to 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 expedited requests to the TAPI672 to retrieve XFER[3:0] + 1 blocks of data.

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. The starving trigger level must always be set to a number of blocks greater than or equal to the channel transfer size.

Table 22 – Level[3:0]/TRANS Settings

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)

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

Register 0x3A0 : THDL Indirect Block Select

Bit	Type	Function	Default
Bit 15	R	BUSY	X
Bit 14	R/W	BRWB	0
Bit 13 to Bit 12		Unused	XH
Bit 11	R/W	Reserved	X
Bit 10	R/W	BLOCK[10]	0
Bit 9	R/W	BLOCK[9]	0
Bit 8	R/W	BLOCK[8]	0
Bit 7	R/W	BLOCK[7]	0
Bit 6	R/W	BLOCK[6]	0
Bit 5	R/W	BLOCK[5]	0
Bit 4	R/W	BLOCK[4]	0
Bit 3	R/W	BLOCK[3]	0
Bit 2	R/W	BLOCK[2]	0
Bit 1	R/W	BLOCK[1]	0
Bit 0	R/W	BLOCK[0]	0

This register provides the block number used to access the block pointer RAM. Writing to this register triggers an indirect block register access.

BLOCK[10:0]:

The indirect block number (BLOCK[10:0]) indicate the block to be configured or interrogated in the indirect access.

Reserved:

The reserved bit must be set low for correct operation of the FREEDM-84A672 device.

BRWB:

The block indirect access control bit (BRWB) selects between a configure (write) or interrogate (read) access to the block pointer RAM. Writing a logic

zero to BRWB triggers an indirect block write operation. Data to be written is taken from the Indirect Block Data register. Writing a logic one to BRWB triggers an indirect block read operation. The data read can be found in the Indirect Block Data register.

BUSY:

The indirect access status bit (BUSY) reports the progress of an indirect access. BUSY is set high when this register is written to trigger an indirect access, and will stay high until the access is complete. At which point, BUSY will be set low. This register should be polled to determine when data from an indirect read operation is available in the THDL Indirect Block Data register or to determine when a new indirect write operation may commence.

Register 0x3A4 : THDL Indirect Block Data

Bit	Type	Function	Default
Bit 15	R/W	Reserved	0
Bit 14 to Bit 12		Unused	XH
Bit 11	R/W	Reserved	X
Bit 10	R/W	BPTR[10]	0
Bit 9	R/W	BPTR[9]	0
Bit 8	R/W	BPTR[8]	0
Bit 7	R/W	BPTR[7]	0
Bit 6	R/W	BPTR[6]	0
Bit 5	R/W	BPTR[5]	0
Bit 4	R/W	BPTR[4]	0
Bit 3	R/W	BPTR[3]	0
Bit 2	R/W	BPTR[2]	0
Bit 1	R/W	BPTR[1]	0
Bit 0	R/W	BPTR[0]	0

This register contains data read from the transmit block pointer RAM after an indirect block read operation or data to be inserted into the transmit block pointer RAM in an indirect block write operation.

BPTR[10:0]:

The indirect block pointer (BPTR[10:0]) configures the block pointer of the block specified by the Indirect Block Select register. The block pointer to be written to the transmit block pointer RAM, in an indirect write operation, must be set up in this register before triggering the write. The block pointer value is the block number of the next block in the linked list. A circular list of blocks must be formed in order to use the block list as a channel FIFO buffer. FPTR[10:0] reflects the value written until the completion of a subsequent indirect block read operation.

When provisioning a channel FIFO, all blocks pointers must be re-written to properly initialize the FIFO.

Reserved:

The reserved bits (Reserved) must be set low for correct operation of the FREEDM-84A672 device.

Register 0x3B0 : THDL Configuration

Bit	Type	Function	Default
Bit 15 to Bit 10		Unused	XXH
Bit 9	R/W	BIT8	0
Bit 8	R/W	TSTD	0
Bit 7	R/W	Reserved	0
Bit 6 to Bit 4		Unused	XH
Bit 3 to Bit 0	R/W	Reserved	0H

This register configures all provisioned channels.

Reserved:

The reserved bits must be set low for correct operation of the FREEDM-84A672 device.

TSTD:

The telecom standard bit (TSTD) controls the bit ordering of the HDLC data transferred on the transmit APPI. When TSTD is set low, the least significant bit of the each byte on the transmit APPI bus (AD[0] and AD[8]) is the first HDLC bit transmitted and the most significant bit of each byte (AD[7] and AD[15]) is the last HDLC bit transmitted (datacom standard). When TSTD is set high, AD[0] and AD[8] are the last HDLC bit transmitted and AD[7] and AD[15] are the first HDLC bit transmitted (telecom standard).

BIT8:

The least significant stuff control bit (BIT8) carries the value placed in the least significant bit of each octet when the HDLC processor is configured (7BIT set high) to stuff the least significant bit of each octet in the corresponding transmit link (TD[n]). When BIT8 is set high, the least significant bit (last bit of each octet transmitted) is forced high. When BIT8 is set low, the least significant bit is forced low. BIT8 is ignored when 7BIT is set low.

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

Bit	Type	Function	Default
Bit 15	R	BUSY	X
Bit 14	R/W	RWB	0
Bit 13		Unused	X
Bit 12	R/W	LINK[6]	0
Bit 11	R/W	LINK[5]	0
Bit 10	R/W	LINK[4]	0
Bit 9	R/W	LINK[3]	0
Bit 8	R/W	LINK[2]	0
Bit 7	R/W	LINK[1]	0
Bit 6	R/W	LINK[0]	0
Bit 5		Unused	X
Bit 4	R/W	TSLOT[4]	0
Bit 3	R/W	TSLOT[3]	0
Bit 2	R/W	TSLOT[2]	0
Bit 1	R/W	TSLOT[1]	0
Bit 0	R/W	TSLOT[0]	0

This register provides the link number and time-slot number used to access the transmit channel provision RAM. Writing to this register triggers an indirect register access and transfers the contents of the Indirect Channel Data register to an internal holding register.

TSLOT[4:0]:

The indirect time-slot number bits (TSLOT[4:0]) indicate the time-slot to be configured or interrogated in the indirect access. For a channelised T1/J1 link, time-slots 1 to 24 are valid. For a channelised E1 link, time-slots 1 to 31 are valid. For unchannelised or unframed links, only time-slot 0 is valid.

LINK[6:0]:

The indirect link number bits (LINK[6:0]) select amongst the 84 transmit links to be configured or interrogated in the indirect access.

RWB:

The indirect access control bit (RWB) selects between a configure (write) or interrogate (read) access to the transmit channel provision RAM. The address to the transmit channel provision RAM is constructed by concatenating the TSLOT[6:0] and LINK[4:0] bits. Writing a logic zero to RWB triggers an indirect write operation. Data to be written is taken from the PROV and the CHAN[9:0] bits of the Indirect Data register. Writing a logic one to RWB triggers an indirect read operation. Addressing of the RAM is the same as in an indirect write operation. The data read can be found in the PROV and the CHAN[9:0] bits of the Indirect Channel Data register.

BUSY:

The indirect access status bit (BUSY) reports the progress of an indirect access. BUSY is set high when this register is written to trigger an indirect access, and will stay high until the access is complete. At which point, BUSY will be set low. This register should be polled to determine when data from an indirect read operation is available in the TCAS Indirect Channel Data register or to determine when a new indirect write operation may commence.

Register 0x404 : TCAS Indirect Channel Data

Bit	Type	Function	Default
Bit 15	R/W	PROV	0
Bit 14 to Bit 10		Unused	XXH
Bit 9	R/W	CHAN[9]	0
Bit 8	R/W	CHAN[8]	0
Bit 7	R/W	CHAN[7]	0
Bit 6	R/W	CHAN[6]	0
Bit 5	R/W	CHAN[5]	0
Bit 4	R/W	CHAN[4]	0
Bit 3	R/W	CHAN[3]	0
Bit 2	R/W	CHAN[2]	0
Bit 1	R/W	CHAN[1]	0
Bit 0	R/W	CHAN[0]	0

This register contains the data read from the transmit channel provision RAM after an indirect read operation or the data to be inserted into the transmit channel provision RAM in an indirect write operation.

CHAN[9:0]:

The indirect data bits (CHAN[9:0]) report the channel number read from the transmit channel provision RAM after an indirect read operation has completed. Channel number to be written to the transmit channel provision RAM in an indirect write operation must be set up in this register before triggering the write. CHAN[9:0] reflects the value written until the completion of a subsequent indirect read operation.

PROV:

The indirect provision enable bit (PROV) reports the channel provision enable flag read from transmit channel provision RAM after an indirect read operation has completed. The provision enable flag to be written to the transmit channel provision RAM in an indirect write operation must be set up in this register before triggering the write. When PROV is set high, the current time-

slot is assigned to the channel as indicated by CHAN[9:0]. When PROV is set low, the time-slot does not belong to any channel. The transmit link data is set to the contents of the Idle Time-slot Fill Data register. PROV reflects the value written until the completion of a subsequent indirect read operation.

Register 0x40C : TCAS Idle Time-slot Fill Data

Bit	Type	Function	Default
Bit 15 to Bit 8		Unused	XXH
Bit 7	R/W	FDATA[7]	1
Bit 6	R/W	FDATA[6]	1
Bit 5	R/W	FDATA[5]	1
Bit 4	R/W	FDATA[4]	1
Bit 3	R/W	FDATA[3]	1
Bit 2	R/W	FDATA[2]	1
Bit 1	R/W	FDATA[1]	1
Bit 0	R/W	FDATA[0]	1

This register contains the data to be written to disabled time-slots of a channelised link.

FDATA[7:0]:

The fill data bits (FDATA[7:0]) are transmitted during disabled (PROV set low) time-slots of channelised links.

Register 0x410 : TCAS Channel Disable

Bit	Type	Function	Default
Bit 15	R/W	CHDIS	0
Bit 14 to Bit 10		Unused	XXH
Bit 9	R/W	DCHAN[9]	0
Bit 8	R/W	DCHAN[8]	0
Bit 7	R/W	DCHAN[7]	0
Bit 6	R/W	DCHAN[6]	0
Bit 5	R/W	DCHAN[5]	0
Bit 4	R/W	DCHAN[4]	0
Bit 3	R/W	DCHAN[3]	0
Bit 2	R/W	DCHAN[2]	0
Bit 1	R/W	DCHAN[1]	0
Bit 0	R/W	DCHAN[0]	0

This register controls the disabling of one specific channel to allow orderly provisioning of time-slots.

DCHAN[9:0]:

The disable channel number bits (DCHAN[9:0]) selects the channel to be disabled. When CHDIS is set high, the channel specified by DCHAN[9:0] is disabled. Data in time-slots associated with the specified channel is set to FDATA[7:0] in the Idle Time-slot Fill Data register. When CHDIS is set low, the channel specified by DCHAN[9:0] operates normally.

CHDIS:

The channel disable bit (CHDIS) controls the disabling of the channels specified by DCHAN[9:0]. When CHDIS is set high, the channel selected by DCHAN[9:0] is disabled. Data in time-slots associated with the specified channel is set to FDATA[7:0] in the Idle Time-slot Fill Data register. When CHDIS is set low, the channel specified by DCHAN[9:0] operates normally.

Register 0x440 : TCAS SBI SPE1 Configuration Register #1

Bit	Type	Function	Default
Bit 15	R/W	FEN[11]	0
Bit 14	R/W	FEN[10]	0
Bit 13	R/W	FEN[9]	0
Bit 12	R/W	FEN[8]	0
Bit 11	R/W	FEN[7]	0
Bit 10	R/W	FEN[6]	0
Bit 9	R/W	FEN[5]	0
Bit 8	R/W	FEN[4]	0
Bit 7	R/W	FEN[3]	0
Bit 6	R/W	FEN[2]	0
Bit 5	R/W	FEN[1]	0
Bit 4	R/W	FEN[0]	0
Bit 3		Unused	X
Bit 2	R/W	SBI_MODE[2]	0
Bit 1	R/W	SBI_MODE[1]	0
Bit 0	R/W	SBI_MODE[0]	0

This register configures the operational mode of transmit links 0, 3, 6, 9, … 33, 36, 39, …81, i.e. those links mapped to SPE 1 of the SBI ADD BUS.

SBI_MODE[2:0]:

The SBI mode select bits (SBI_MODE[2:0]) configure the transmit links of SPE1, as shown in the following table:

Table 23 – SBI Mode SPE1 Configuration

SBI_MODE [2:0]	SPE1 Configuration
000	Single unchannelised DS-3 on link 0
001	28 T1/J1 links
010	21 E1 links (links 63, 66, 69, … , 81 are unused)
011	Reserved
100	Reserved
101	Reserved
110	Reserved
111	Reserved

FEN[11:0]

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.

Register 0x444 : TCAS SBI SPE1 Configuration Register #2

Bit	Type	Function	Default
Bit 15	R/W	FEN[27]	0
Bit 14	R/W	FEN[26]	0
Bit 13	R/W	FEN[25]	0
Bit 12	R/W	FEN[24]	0
Bit 11	R/W	FEN[23]	0
Bit 10	R/W	FEN[22]	0
Bit 9	R/W	FEN[21]	0
Bit 8	R/W	FEN[20]	0
Bit 7	R/W	FEN[19]	0
Bit 6	R/W	FEN[18]	0
Bit 5	R/W	FEN[17]	0
Bit 4	R/W	FEN[16]	0
Bit 3	R/W	FEN[15]	0
Bit 2	R/W	FEN[14]	0
Bit 1	R/W	FEN[13]	0
Bit 0	R/W	FEN[12]	0

The bits of this register set are used to configure the framing modes of transmit links 36, 39, 42 … 81.

FEN[27:12]:

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.

Register 0x448 : TCAS SBI SPE2 Configuration Register #1

Bit	Type	Function	Default
Bit 15	R/W	FEN[11]	0
Bit 14	R/W	FEN[10]	0
Bit 13	R/W	FEN[9]	0
Bit 12	R/W	FEN[8]	0
Bit 11	R/W	FEN[7]	0
Bit 10	R/W	FEN[6]	0
Bit 9	R/W	FEN[5]	0
Bit 8	R/W	FEN[4]	0
Bit 7	R/W	FEN[3]	0
Bit 6	R/W	FEN[2]	0
Bit 5	R/W	FEN[1]	0
Bit 4	R/W	FEN[0]	0
Bit 3		Unused	X
Bit 2	R/W	SBI_MODE[2]	0
Bit 1	R/W	SBI_MODE[1]	0
Bit 0	R/W	SBI_MODE[0]	0

This register configures the operational mode of transmit links 1, 4, 7, 10, … 34, 37, …82, i.e. those links mapped to SPE 2 of the SBI ADD BUS.

SBI_MODE[2:0]:

The SBI mode select bits (SBI_MODE[2:0]) configure the transmit links of SPE2, as shown in the following table:

Table 24 – SBI Mode SPE2 Configuration

SBI_MODE [2:0]	SPE2 Configuration
000	Single unchannelised DS-3 on link 1
001	28 T1/J1 links
010	21 E1 links (links 64, 67, 70, … , 82 are unused)
011	Reserved
100	Reserved
101	Reserved
110	Reserved
111	Reserved

FEN[11:0]

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.

Register 0x44C : TCAS SBI SPE2 Configuration Register #2

Bit	Type	Function	Default
Bit 15	R/W	FEN[27]	0
Bit 14	R/W	FEN[26]	0
Bit 13	R/W	FEN[25]	0
Bit 12	R/W	FEN[24]	0
Bit 11	R/W	FEN[23]	0
Bit 10	R/W	FEN[22]	0
Bit 9	R/W	FEN[21]	0
Bit 8	R/W	FEN[20]	0
Bit 7	R/W	FEN[19]	0
Bit 6	R/W	FEN[18]	0
Bit 5	R/W	FEN[17]	0
Bit 4	R/W	FEN[16]	0
Bit 3	R/W	FEN[15]	0
Bit 2	R/W	FEN[14]	0
Bit 1	R/W	FEN[13]	0
Bit 0	R/W	FEN[12]	0

The bits of this register set are used to configure the framing modes of transmit links 37, 40, 43 … 82.

FEN[27:12]:

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.

Register 0x450 : TCAS SBI SPE3 Configuration Register #1

Bit	Type	Function	Default
Bit 15	R/W	FEN[11]	0
Bit 14	R/W	FEN[10]	0
Bit 13	R/W	FEN[9]	0
Bit 12	R/W	FEN[8]	0
Bit 11	R/W	FEN[7]	0
Bit 10	R/W	FEN[6]	0
Bit 9	R/W	FEN[5]	0
Bit 8	R/W	FEN[4]	0
Bit 7	R/W	FEN[3]	0
Bit 6	R/W	FEN[2]	0
Bit 5	R/W	FEN[1]	0
Bit 4	R/W	FEN[0]	0
Bit 3		Unused	X
Bit 2	R/W	SBI_MODE[2]	0
Bit 1	R/W	SBI_MODE[1]	0
Bit 0	R/W	SBI_MODE[0]	0

This register configures the operational mode of transmit links 2, 5, 8, 11, … 35, 38, …83, i.e. those links mapped to SPE 3 of the SBI ADD BUS.

SBI_MODE[2:0]:

The SBI mode select bits (SBI_MODE[2:0]) configure the transmit links of SPE3, as shown in the following table:

Table 25 – SBI Mode SPE3 Configuration

SBI_MODE [2:0]	SPE3 Configuration
000	Single unchannelised DS-3 on link 2
001	28 T1/J1 links
010	21 E1 links (links 65, 68, 71, … , 83 are unused)
011	Reserved
100	Reserved
101	Reserved
110	Reserved
111	Reserved

FEN[11:0]

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.

Register 0x454 : TCAS SBI SPE3 Configuration Register #2

Bit	Type	Function	Default
Bit 15	R/W	FEN[27]	0
Bit 14	R/W	FEN[26]	0
Bit 13	R/W	FEN[25]	0
Bit 12	R/W	FEN[24]	0
Bit 11	R/W	FEN[23]	0
Bit 10	R/W	FEN[22]	0
Bit 9	R/W	FEN[21]	0
Bit 8	R/W	FEN[20]	0
Bit 7	R/W	FEN[19]	0
Bit 6	R/W	FEN[18]	0
Bit 5	R/W	FEN[17]	0
Bit 4	R/W	FEN[16]	0
Bit 3	R/W	FEN[15]	0
Bit 2	R/W	FEN[14]	0
Bit 1	R/W	FEN[13]	0
Bit 0	R/W	FEN[12]	0

The bits of this register set are used to configure the framing modes of transmit links 38, 41, 44 … 83.

FEN[27:12]:

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.

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

Bit	Type	Function	Default
Bit 15 to Bit 5		Unused	XXXH
Bit 4	R/W	Reserved	0
Bit 3		Unused	X
Bit 2	R/W	Reserved	0
Bit 1	R/W	Reserved	0
Bit 0	R/W	Reserved	0

This register controls the operation of transmit links #0 to #2 when they are configured to transmit data on the TD[2:0] outputs (i.e. SPEn_EN is low).

Reserved:

The reserved bits must be set low for correct operation of the FREEDM-84A672 device.

Register 0x500 : PMON Status

Bit	Type	Function	Default
Bit 15 to Bit 6		Unused	XXXH
Bit 5	R	C2DET	X
Bit 4	R	C1DET	X
Bit 3	R	UFDET	X
Bit 2	R	OFDET	X
Bit 1		Unused	X
Bit 0		Unused	X

This register contains status information indicating whether a non-zero count has been latched in the count registers.

OFDET:

The overflow detect bit (OFDET) indicates the status of the PMON Receive FIFO Overflow Count register. OFDET is set high when overflow events have occurred during the latest PMON accumulation interval. OFDET is set low if no overflow events are detected.

UFDET:

The underflow detect bit (UFDET) indicates the status of the PMON Transmit FIFO Underflow Count register. UFDET is set high when underflow events have occurred during the latest PMON accumulation interval. UFDET is set low if no underflow events are detected.

C1DET:

The configurable event #1 detect bit (C1DET) indicates the status of the PMON Configurable Count #1 register. C1DET is set high when selected events have occurred during the latest PMON accumulation interval. C1DET is set low if no selected events are detected.

C2DET:

The configurable event #2 detect bit (C2DET) indicates the status of the PMON Configurable Count #2 register. C2DET is set high when selected

events have occurred during the latest PMON accumulation interval. C2DET is set low if no selected events are detected.

Register 0x504 : PMON Receive FIFO Overflow Count

Bit	Type	Function	Default
Bit 15	R	OF[15]	X
Bit 14	R	OF[14]	X
Bit 13	R	OF[13]	X
Bit 12	R	OF[12]	X
Bit 11	R	OF[11]	X
Bit 10	R	OF[10]	X
Bit 9	R	OF[9]	X
Bit 8	R	OF[8]	X
Bit 7	R	OF[7]	X
Bit 6	R	OF[6]	X
Bit 5	R	OF[5]	X
Bit 4	R	OF[4]	X
Bit 3	R	OF[3]	X
Bit 2	R	OF[2]	X
Bit 1	R	OF[1]	X
Bit 0	R	OF[0]	X

This register reports the number of receive FIFO overflow events in the previous accumulation interval.

OF[15:0]:

The OF[15:0] bits reports the number of receive FIFO overflow events that have been detected since the last time this register was polled. This register is polled by writing to the FREEDM-84A672 Master Clock / Frame Pulse Activity Monitor and Accumulation Trigger register. The write access transfers the internally accumulated error count to the FIFO overflow register and simultaneously resets the internal counter to begin a new cycle of error accumulation.

Register 0x508 : PMON Transmit FIFO Underflow Count

Bit	Type	Function	Default
Bit 15	R	UF[15]	X
Bit 14	R	UF[14]	X
Bit 13	R	UF[13]	X
Bit 12	R	UF[12]	X
Bit 11	R	UF[11]	X
Bit 10	R	UF[10]	X
Bit 9	R	UF[9]	X
Bit 8	R	UF[8]	X
Bit 7	R	UF[7]	X
Bit 6	R	UF[6]	X
Bit 5	R	UF[5]	X
Bit 4	R	UF[4]	X
Bit 3	R	UF[3]	X
Bit 2	R	UF[2]	X
Bit 1	R	UF[1]	X
Bit 0	R	UF[0]	X

This register reports the number of transmit FIFO underflow events in the previous accumulation interval.

UF[15:0]:

The UF[15:0] bits reports the number of transmit FIFO underflow events that have been detected since the last time this register was polled. This register is polled by writing to the FREEDM-84A672 Master Clock / Frame Pulse Activity Monitor and Accumulation Trigger register. The write access transfers the internally accumulated error count to the FIFO underflow register and simultaneously resets the internal counter to begin a new cycle of error accumulation.

Register 0x50C : PMON Configurable Count #1

Bit	Type	Function	Default
Bit 15	R	C1[15]	X
Bit 14	R	C1[14]	X
Bit 13	R	C1[13]	X
Bit 12	R	C1[12]	X
Bit 11	R	C1[11]	X
Bit 10	R	C1[10]	X
Bit 9	R	C1[9]	X
Bit 8	R	C1[8]	X
Bit 7	R	C1[7]	X
Bit 6	R	C1[6]	X
Bit 5	R	C1[5]	X
Bit 4	R	C1[4]	X
Bit 3	R	C1[3]	X
Bit 2	R	C1[2]	X
Bit 1	R	C1[1]	X
Bit 0	R	C1[0]	X

This register reports the number events, selected by the FREEDM-84A672 Master Performance Monitor Control register, that occurred in the previous accumulation interval.

C1[15:0]:

The C1[15:0] bits reports the number of selected events that have been detected since the last time this register was polled. This register is polled by writing to the FREEDM-84A672 Master Clock / Frame Pulse Activity Monitor and Accumulation Trigger register. The write access transfers the internally accumulated error count to the configurable count #1 register and simultaneously resets the internal counter to begin a new cycle of event accumulation.

Register 0x510 : PMON Configurable Count #2

Bit	Type	Function	Default
Bit 15	R	C2[15]	X
Bit 14	R	C2[14]	X
Bit 13	R	C2[13]	X
Bit 12	R	C2[12]	X
Bit 11	R	C2[11]	X
Bit 10	R	C2[10]	X
Bit 9	R	C2[9]	X
Bit 8	R	C2[8]	X
Bit 7	R	C2[7]	X
Bit 6	R	C2[6]	X
Bit 5	R	C2[5]	X
Bit 4	R	C2[4]	X
Bit 3	R	C2[3]	X
Bit 2	R	C2[2]	X
Bit 1	R	C2[1]	X
Bit 0	R	C2[0]	X

This register reports the number events, selected by the FREEDM-84A672 Master Performance Monitor Control register, that occurred in the previous accumulation interval.

C2[15:0]:

The C2[15:0] bits reports the number of selected events that have been detected since the last time this register was polled. This register is polled by writing to the FREEDM-84A672 Master Clock / Frame Pulse Activity Monitor and Accumulation Trigger register. The write access transfers the internally accumulated error count to the configurable count #2 register and simultaneously resets the internal counter to begin a new cycle of event accumulation.

Register 0x580 : RAPI Control

Bit	Type	Function	Default
Bit 15	R/W	ENABLE	0
Bit 14	R/W	STATEN	0
Bit 13	R/W	Reserved	0
Bit 12 to Bit 4		Unused	XXXH
Bit 3	R/W	ALL1ENB	1
Bit 2	R/W	BADDR[2]	1
Bit 1	R/W	BADDR[1]	1
Bit 0	R/W	BADDR[0]	1

This register provides the base address of the Rx APPI for purposes of responding to polling and device selection. This register also enables the RAPI672.

BADDR[2:0]:

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.

ALL1ENB:

The All Ones Enable bit (ALL1ENB) permits the FREEDM-84A672 to respond to receive polling and device selection when BADDR[2:0] = ‘111’. When ALL1ENB is zero, the FREEDM-84A672 responds to receive polling and device selection when BADDR[2:0] = RXADDR[2:0] = ‘111’. When ALL1ENB is one, 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].

Reserved:

The reserved bit must be set to zero for correct operation of the FREEDM-84A672 device.

STATEN:

The RAPI672 Status Enable bit (STATEN) enables the RAPI672 to provide the status of an errored packet on RXDATA[7:0] during transfer of the final word of that packet on the receive APPI (REOP and RERR high). When STATEN is set high, the RAPI672 overwrites RXDATA[7:0] of the final word of an errored packet with status information for that packet. When STATEN is set low, the RAPI672 does not report detailed status information for an errored packet. The RXDATA[15:0] connector description details the errored packet status reporting when STATEN is set high.

ENABLE:

The RAPI672 Enable bit (ENABLE) enables normal operation of the RAPI672. When ENABLE is set low, the RAPI672 will not transfer data from the RHDL672 into its internal FIFOs. When ENABLE is set high, the RAPI672 operates normally.

Register 0x5C0 : SBI EXTRACT Control

Bit	Type	Function	Default
Bit 15 to Bit 8		Unused	XXH
Bit 7	R/W	Reserved	0
Bit 6		Unused	X
Bit 5		Unused	X
Bit 4	R/W	Reserved	0
Bit 3	R/W	Reserved	0
Bit 2	R/W	Reserved	0
Bit 1	R/W	SBI_PERR_EN	0
Bit 0	R/W	SBI_PAR_CTL	1

This register controls the operation of the SBI EXTRACT block.

SBI_PAR_CTL

The SBI_PAR_CTL bit is used to configure the Parity mode for checking of the SBI parity signal, DDP as follows: When SBI_PAR_CTL is ’0’ parity is even. When SBI_PAR_CTL is ‘1’ parity is odd.

SBI_PERR_EN

The SBI_PERR_EN bit is used to enable SBI Parity Error interrupt generation. When SBI_PERR_EN is ‘0’, SBI Parity Error Interrupts are disabled. When SBI_PERR_EN is ‘1’, SBI Parity Error Interrupts are enabled. In both cases the SBI Parity checker logic will update the SBI EXTRACT Parity Error Interrupt Reason Register when a parity error occurs.

Reserved:

The reserved bits must be set low for correct operation of the FREEDM-84A672 device.

Register 0x5CC : SBI EXTRACT Tributary RAM Indirect Access Address

Bit	Type	Function	Default
Bit 15 to Bit 8		Unused	XXH
Bit 7	R/W	Reserved	0
Bit 6	R/W	SPE[1]	0
Bit 5	R/W	SPE[0]	0
Bit 4	R/W	TRIB[4]	0
Bit 3	R/W	TRIB[3]	0
Bit 2	R/W	TRIB[2]	0
Bit 1	R/W	TRIB[1]	0
Bit 0	R/W	TRIB[0]	0

This register provides the receive SPE and link number used to access the SBI EXTRACT tributary control configuration RAM.

TRIB[4:0] and SPE[1:0]

The TRIB[4:0] and SPE[1:0] fields 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'.

Reserved:

The reserved bit must be set low for correct operation of the FREEDM-84A672 device.

Register 0x5D0 : SBI EXTRACT Tributary RAM Indirect Access Control

Bit	Type	Function	Default
Bit 15 to Bit 8		Unused	XXH
Bit 7	R/W	BUSY	X
Bit 6 to Bit 2		Unused	XXH
Bit 1	R/W	RWB	0
Bit 0	R/W	Reserved	0

This register controls access the SBI EXTRACT tributary control configuration RAM. Writing to this register triggers an indirect register access.

Reserved:

The reserved bit must be set low for correct operation of the FREEDM-84A672 device.

RWB

The indirect access control bit (RWB) selects between a configure (write) or interrogate (read) access to the tributary control configuration RAM. Writing a '0' to RWB triggers an indirect write operation. Data to be written is taken from the SBI EXTRACT Tributary RAM Indirect Access Data Register. Writing a '1' to RWB triggers an indirect read operation. The data read can be found in the SBI EXTRACT Tributary RAM Indirect Access Data Register.

BUSY

The indirect access status bit (BUSY) reports the progress of an indirect access. BUSY is set high when a write to the SBI EXTRACT Tributary RAM Indirect Access Control Register triggers an indirect access and will stay high until the access is complete. This register should be polled to determine when data from an indirect read operation is available in the SBI EXTRACT Tributary RAM Indirect Access Data Register or to determine when a new indirect write operation may commence.

Register 0x5D8 : SBI EXTRACT Tributary RAM Indirect Access Data

Bit	Type	Function	Default
Bit 15 to Bit 7		Unused	XXXH
Bit 6	R/W	Reserved	0
Bit 5	R/W	Reserved	0
Bit 4	R/W	Reserved	0
Bit 3	R/W	TRIB_TYP[1]	0
Bit 2	R/W	TRIB_TYP[0]	0
Bit 1	R/W	Reserved	0
Bit 0	R/W	ENBL	0

This register contains data read from the SBI EXTRACT tributary control configuration RAM after an indirect read operation or data to be written to the tributary control configuration RAM in an indirect write operation.

ENBL

The ENBL bit is used to enable the Tributary. Writing to the SBI EXTRACT 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.

Reserved:

The reserved bits must be set low for correct operation of the FREEDM-84A672 device.

TRIB_TYP[1:0]

The TRIB_TYP[1:0] field is used to specify the characteristics of the SBI tributary as shown in Table 26 below:

Table 26 – TRIB_TYP Encoding

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

Register 0x5DC : SBI EXTRACT Parity Error Interrupt Reason

Bit	Type	Function	Default
Bit 15 to Bit 8		Unused	XXH
Bit 7	R/W	SPE[1]	0
Bit 6	R/W	SPE[0]	1
Bit 5	R/W	TRIB[4]	0
Bit 4	R/W	TRIB[3]	0
Bit 3	R/W	TRIB[2]	0
Bit 2	R/W	TRIB[1]	0
Bit 1	R/W	TRIB[0]	1
Bit 0	R	PERRI	0

This register provides information about the most recent parity error on the SBI DROP BUS.

PERRI

When set PERRI indicates that an SBI parity error has been detected. Reading the SBI EXTRACT Parity Error Interrupt Reason Register clears this bit.

TRIB[4:0] and SPE[1:0]

The TRIB[4:0] and SPE[1:0] fields specify the SBI tributary for which a parity error was detected. These fields are only valid when PERRI is set.

Register 0x600 : TAPI Control

Bit	Type	Function	Default
Bit 15	R/W	ENABLE	0
Bit 14	R/W	Reserved	0
Bit 13	R/W	Reserved	0
Bit 12 to Bit 4		Unused	XXXH
Bit 3	R/W	ALL1ENB	1
Bit 2	R/W	BADDR[2]	1
Bit 1	R/W	BADDR[1]	1
Bit 0	R/W	BADDR[0]	1

This register provides the base address of the Tx APPI for purposes of responding to polling and Tx APPI data transfers. This register also enables the TAPI672.

BADDR[2:0]:

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.

ALL1ENB:

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

Reserved:

The reserved bits must be set to zero for correct operation of the FREEDM-84A672 device.

ENABLE:

The TAPI672 Enable bit (ENABLE) enables normal operation of the TAPI672. When ENABLE is set low, the TAPI672 will complete the current data transfer and will respond to any further transactions on the Tx APPI normally (by setting TRDY high), but data provided will be ignored. When ENABLE is set high, the TAPI672 operates normally.

Register 0x604 : TAPI Indirect Channel Provisioning

Bit	Type	Function	Default
Bit 15	R	BUSY	X
Bit 14	R/W	RWB	0
Bit 13 to Bit 10		Unused	XH
Bit 9	R/W	CHAN[9]	0
Bit 8	R/W	CHAN[8]	0
Bit 7	R/W	CHAN[7]	0
Bit 6	R/W	CHAN[6]	0
Bit 5	R/W	CHAN[5]	0
Bit 4	R/W	CHAN[4]	0
Bit 3	R/W	CHAN[3]	0
Bit 2	R/W	CHAN[2]	0
Bit 1	R/W	CHAN[1]	0
Bit 0	R/W	CHAN[0]	0

The Indirect Channel Provisioning Register provides the channel number used to access the TAPI672 channel provisioning RAM. Writing to this register triggers an indirect channel register access.

CHAN[9:0]:

The indirect channel number bits (CHAN[9:0]) indicate the channel to be configured or interrogated in the indirect access.

RWB:

The Read/Write Bar (RWB) bit selects between a provisioning/unprovisioning operation (write) or a query operation (read). Writing a logic 0 to RWB triggers the provisioning or unprovisioning of the channel specified by CHAN[9:0]. Writing a logic 1 to RWB triggers a query of the channel specified by CHAN[9:0].

BUSY:

The indirect access status bit (BUSY) reports the progress of an indirect access. BUSY is set high when this register is written to trigger an indirect access, and will stay high until the access is complete. At which point, BUSY will be set low. This register should be polled to determine when data from an indirect read operation is available or to determine when a new indirect write operation may commence.

Register 0x608 : TAPI Indirect Channel Data Register

Bit	Type	Function	Default
Bit 15	R/W	PROV	0
Bit 14 to Bit 8		Unused	XH
Bit 7	R/W	BLEN[7]	0
Bit 6	R/W	BLEN[6]	0
Bit 5	R/W	BLEN[5]	0
Bit 4	R/W	BLEN[4]	0
Bit 3	R/W	BLEN[3]	0
Bit 2	R/W	BLEN[2]	0
Bit 1	R/W	BLEN[1]	0
Bit 0	R/W	BLEN[0]	0

The TAPI Indirect Channel Data Register contains data read from the TAPI672 channel provision RAM after an indirect read operation or data to be written to channel provision RAM in an indirect write operation.

BLEN[7:0]:

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. 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.

PROV:

The indirect provision enable bit (PROV) reports the channel provision enable flag read from the TAPI672 channel provision RAM after an indirect read operation has completed. The provision enable flag to be written to the TAPI672 channel provision RAM, in an indirect write operation, must be set up in this register before triggering the write. When PROV is set high, the

channel as indicated by CHAN[9:0] is provisioned. When PROV is set low, the channel indicated by CHAN[9:0] is unprovisioned. PROV reflects the value written until the completion of a subsequent indirect read operation.

Register 0x680 : SBI INSERT Control

Bit	Type	Function	Default
Bit 15 to Bit 8		Unused	XXH
Bit 7	R/W	Reserved	0
Bit 6		Unused	X
Bit 5		Unused	X
Bit 4	R/W	Reserved	0
Bit 3	R/W	Reserved	0
Bit 2	R/W	Reserved	0
Bit 1		Unused	X
Bit 0	R/W	SBI_PAR_CTL	1

This register controls the operation of the SBI INSERT block.

SBI_PAR_CTL

The SBI_PAR_CTL bit is used to configure the Parity mode for generation of the SBI parity signal, ADP as follows: When SBI_PAR_CTL is ’0’ parity is even. When SBI_PAR_CTL is ‘1’ parity is odd.

Reserved:

The reserved bits must be set low for correct operation of the FREEDM-84A672 device.

Register 0x68C : SBI INSERT Tributary RAM Indirect Access Address

Bit	Type	Function	Default
Bit 15 to Bit 8		Unused	XXH
Bit 7	R/W	Reserved	0
Bit 6	R/W	SPE[1]	0
Bit 5	R/W	SPE[0]	0
Bit 4	R/W	TRIB[4]	0
Bit 3	R/W	TRIB[3]	0
Bit 2	R/W	TRIB[2]	0
Bit 1	R/W	TRIB[1]	0
Bit 0	R/W	TRIB[0]	0

This register provides the transmit SPE and link number used to access the SBI INSERT tributary control configuration RAM.

TRIB[4:0] and SPE[1:0]

The TRIB[4:0] and SPE[1:0] fields 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'.

Reserved:

The reserved bit must be set low for correct operation of the FREEDM-84A672 device.

Register 0x690 : SBI INSERT Tributary RAM Indirect Access Control

Bit	Type	Function	Default
Bit 15 to Bit 8		Unused	XXH
Bit 7	R/W	BUSY	X
Bit 6 to Bit 2		Unused	XXH
Bit 1	R/W	RWB	0
Bit 0	R/W	Reserved	0

This register controls access to the SBI INSERT tributary control configuration RAM. Writing to this register triggers an indirect register access.

Reserved:

The reserved bit must be set low for correct operation of the FREEDM-84A672 device.

RWB

The indirect access control bit (RWB) selects between a configure (write) or interrogate (read) access to the tributary control configuration RAM. Writing a '0' to RWB triggers an indirect write operation. Data to be written is taken from the SBI INSERT Tributary RAM Indirect Access Data Register. Writing a '1' to RWB triggers an indirect read operation. The data read can be found in the SBI INSERT Tributary RAM Indirect Access Data Register.

BUSY

The indirect access status bit (BUSY) reports the progress of an indirect access. BUSY is set high when a write to the SBI INSERT Tributary RAM Indirect Access Control Register triggers an indirect access and will stay high until the access is complete. This register should be polled to determine when data from an indirect read operation is available in the SBI INSERT Tributary RAM Indirect Access Data Register or to determine when a new indirect write operation may commence.

Register 0x698 : SBI INSERT Tributary RAM Indirect Access Data

Bit	Type	Function	Default
Bit 15 to Bit 5		Unused	XXXH
Bit 4	R/W	CLK_MSTR	0
Bit 3	R/W	TRIB_TYP[1]	0
Bit 2	R/W	TRIB_TYP[0]	0
Bit 1	R/W	Reserved	0
Bit 0	R/W	ENBL	0

This register contains data read from the SBI INSERT tributary control configuration RAM after an indirect read operation or data to be written to the tributary control configuration RAM in an indirect write operation.

ENBL

The ENBL bit is used to enable the Tributary. Writing to the SBI INSERT tributary control configuration RAM with the ENBL bit set enables the SBI INSERT block to output tributary data on an SBI tributary.

Reserved:

The reserved bit must be set low for correct operation of the FREEDM-84A672 device.

TRIB_TYP[1:0]

The TRIB_TYP[1:0] field is used to specify the characteristics of the SBI tributary as shown in Table 27 below:

Table 27 – TRIB_TYP Encoding

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

CLK_MSTR

The CLK_MSTR bit 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).

10 TEST FEATURES DESCRIPTION

The FREEDM-84A672 also supports a standard IEEE 1149.1 five signal JTAG boundary scan test port for use in board testing. All device inputs may be read and all device outputs may be forced via the JTAG test port.

10.1 Test Mode Registers

Test mode registers are used to apply test vectors during production testing of the FREEDM-84A672. Production testing is enabled by asserting the PMCTEST pin. During production tests, FREEDM-84A672 registers are selected by the TA[12:0] pins. Read accesses are enabled by asserting TRDB low while write accesses are enabled by asserting TWRB low. Test mode register data is conveyed on the TDAT[15:0] pins. Test mode registers (as opposed to normal mode registers) are selected when TA[12]/TRS is set high.

Table 28 – Test Mode Register Memory Map

Address TA[12:0]	Register
0x0000 - 0x07FE	Normal Mode Registers
0x0800 - 0x10FE	Reserved
0x1100 - 0x11FE	RCAS672 Test Registers
0x1200 - 0x123E	RHDL672 Test Registers
0x1240 - 0x137E	Reserved
0x1380 - 0x13BE	THDL672 Test Registers
0x13C0 - 0x13FE	Reserved
0x1400 - 0x14FE	TCAS672 Test Registers
0x1500 - 0x151E	PMON Test Registers
0x1520 - 0x157E	Reserved
0x1580 - 0x15BE	RAPI672 Test Registers
0x15C0 - 0x15FE	SBI EXTRACT Test Registers
0x1600 - 0x163E	TAPI672 Test Registers
0x1640 - 0x167E	Reserved
0x1680 - 0x16FE	SBI INSERT Test Registers
0x1700 - 0x17FE	Reserved
0x1800 - 0x18FE	SBI PISO#1 Test Registers
0x1900 - 0x19FE	SBI PISO#2 Test Registers
0x1A00 - 0x1AFE	SBI PISO#3 Test Registers
0x1B00 - 0x1BFE	SBI SIPO#1 Test Registers
0x1C00 - 0x1CFE	SBI SIPO#2 Test Registers
0x1D00 - 0x1DFE	SBI SIPO#3 Test Registers
0x1E00 - 0x1FFE	Reserved

Notes on Test Mode Register Bits:

1. 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 must 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 when read.

2. Writable test mode register bits are not initialized upon reset unless otherwise noted.

10.2 JTAG Test Port

The FREEDM-84A672 JTAG Test Access Port (TAP) allows access to the TAP controller and the 4 TAP registers: instruction, bypass, device identification and boundary scan. Using the TAP, device input logic levels can be read, device outputs can be forced, the device can be identified and the device scan path can be bypassed. For more details on the JTAG port, please refer to the Operations section.

Table 29 – Instruction Register

Length - 3 bits

Instructions	Selected Register	Instruction Code IR[2:0]
EXTEST	Boundary Scan	000
IDCODE	Identification	001
SAMPLE	Boundary Scan	010
BYPASS	Bypass	011
BYPASS	Bypass	100
STCTEST	Boundary Scan	101
BYPASS	Bypass	110
BYPASS	Bypass	111

10.2.1 Identification Register

Length - 32 bits

Version number - 2H

Part Number - 7385H

Manufacturer's identification code - 0CDH

Device identification - 273850CDH

10.2.2 Boundary Scan Register

The boundary scan register is made up of 365 boundary scan cells, divided into input observation (in_cell), output (out_cell), and bi-directional (io_cell) cells. These cells are detailed in the following pages. The first 32 cells form the ID code register, and carry the code 273850CDH. The cells are arranged as follows:

Table 30 – Boundary Scan Chain

Pin/Enable	Register Bit	Cell Type	Device I.D.
Unconnected	0	OUT_CELL	-
Unconnected	1	OUT_CELL	-
Unconnected	2	OUT_CELL	-
Unconnected	3	OUT_CELL	-
FASTCLK	4	IN_CELL	-
Logic 0	5	IN_CELL	-
SPE1_EN	6	IN_CELL	-
SPE2_EN	7	IN_CELL	-
SPE3_EN	8	IN_CELL	-
TD_OEN[0]	9	OUT_CELL	-
TD[0]	10	OUT_CELL	-
TCLK[0]	11	IN_CELL	-
TD_OEN[1]	12	OUT_CELL	-
TD[1]	13	OUT_CELL	-
TCLK[1]	14	IN_CELL	-
TD_OEN[2]	15	OUT_CELL	-
TD[2]	16	OUT_CELL	-
TCLK[2]	17	IN_CELL	-
C1FPOUT_OEN	18	OUT_CELL	-
C1FPOUT	19	OUT_CELL	-
REFCLK	20	IN_CELL	-
APL_OEN	21	OUT_CELL	-

Pin/Enable	Register Bit	Cell Type	Device I.D.
APL	22	OUT_CELL	-
DPL	23	IN_CELL	-
AV5_OEN	24	OUT_CELL	-
AV5	25	OUT_CELL	-
DV5	26	IN_CELL	-
ADP_OEN	27	OUT_CELL	-
ADP	28	OUT_CELL	-
DDP	29	IN_CELL	-
AACTIVE_OEN	30	OUT_CELL	-
AACTIVE	31	OUT_CELL	-
C1FP	32	IN_CELL	-
Logic 0	33	IN_CELL	-
Logic 0	34	IN_CELL	-
ADATA_OEN[0]	35	OUT_CELL	-
ADATA[0]	36	OUT_CELL	-
DDATA[0]	37	IN_CELL	-
ADATA_OEN[1]	38	OUT_CELL	-
ADATA[1]	39	OUT_CELL	-
DDATA[1]	40	IN_CELL	-
ADATA_OEN[2]	41	OUT_CELL	-
ADATA[2]	42	OUT_CELL	-
DDATA[2]	43	IN_CELL	-
ADATA_OEN[3]	44	OUT_CELL	-
ADATA[3]	45	OUT_CELL	-
DDATA[3]	46	IN_CELL	-
ADATA_OEN[4]	47	OUT_CELL	-
ADATA[4]	48	OUT_CELL	-
DDATA[4]	49	IN_CELL	-

Pin/Enable	Register Bit	Cell Type	Device I.D.
ADATA_OEN[5]	50	OUT_CELL	-
ADATA[5]	51	OUT_CELL	-
DDATA[5]	52	IN_CELL	-
ADATA_OEN[6]	53	OUT_CELL	-
ADATA[6]	54	OUT_CELL	-
DDATA[6]	55	IN_CELL	-
ADATA_OEN[7]	56	OUT_CELL	-
ADATA[7]	57	OUT_CELL	-
DDATA[7]	58	IN_CELL	-
Logic 0	59	IN_CELL	-
Logic 0	60	IN_CELL	-
TDAT_OEN[0]	61	OUT_CELL	-
TDAT[0]	62	IO_CELL	-
AJUST_REQ	63	IN_CELL	-
TDAT_OEN[1]	64	OUT_CELL	-
TDAT[1]	65	IO_CELL	-
ADETECT[0]	66	IN_CELL	-
TDAT_OEN[2]	67	OUT_CELL	-
TDAT[2]	68	IO_CELL	-
ADETECT[1]	69	IN_CELL	-
TDAT_OEN[3]	70	OUT_CELL	-
TDAT[3]	71	IO_CELL	-
Logic 0	72	IN_CELL	-
TDAT_OEN[4]	73	OUT_CELL	-
TDAT[4]	74	IO_CELL	-
Logic 0	75	IN_CELL	-
TDAT_OEN[5]	76	OUT_CELL	-
TDAT[5]	77	IO_CELL	-

Pin/Enable	Register Bit	Cell Type	Device I.D.
Logic 0	78	IN_CELL	-
TDAT_OEN[6]	79	OUT_CELL	-
TDAT[6]	80	IO_CELL	-
Logic 0	81	IN_CELL	-
TDAT_OEN[7]	82	OUT_CELL	-
TDAT[7]	83	IO_CELL	-
Logic 0	84	IN_CELL	-
Logic 0	85	IN_CELL	-
Logic 0	86	IN_CELL	-
TDAT_OEN[8]	87	OUT_CELL	-
TDAT[8]	88	IO_CELL	-
Logic 0	89	IN_CELL	-
TDAT_OEN[9]	90	OUT_CELL	-
TDAT[9]	91	IO_CELL	-
Logic 0	92	IN_CELL	-
TDAT_OEN[10]	93	OUT_CELL	-
TDAT[10]	94	IO_CELL	-
Logic 0	95	IN_CELL	-
TDAT_OEN[11]	96	OUT_CELL	-
TDAT[11]	97	IO_CELL	-
Logic 0	98	IN_CELL	-
TDAT_OEN[12]	99	OUT_CELL	-
TDAT[12]	100	IO_CELL	-
Logic 0	101	IN_CELL	-
TDAT_OEN[13]	102	OUT_CELL	-
TDAT[13]	103	IO_CELL	-
Logic 0	104	IN_CELL	-
TDAT_OEN[14]	105	OUT_CELL	-

Pin/Enable	Register Bit	Cell Type	Device I.D.
TDAT[14]	106	IO_CELL	-
Logic 0	107	IN_CELL	-
TDAT_OEN[15]	108	OUT_CELL	-
TDAT[15]	109	IO_CELL	-
Logic 0	110	IN_CELL	-
Logic 0	111	IN_CELL	-
Unconnected	112	OUT_CELL	-
Unconnected	113	OUT_CELL	-
PMCTEST	114	IN_CELL	-
RXADDR[0]	115	IN_CELL	-
RXADDR[1]	116	IN_CELL	-
RXADDR[2]	117	IN_CELL	-
RXCLK	118	IN_CELL	-
RXDATA_OEN[0]	119	OUT_CELL	-
RXDATA[0]	120	IO_CELL	-
RXDATA_OEN[1]	121	OUT_CELL	-
RXDATA[1]	122	IO_CELL	-
RXDATA_OEN[2]	123	OUT_CELL	-
RXDATA[2]	124	IO_CELL	-
RXDATA_OEN[3]	125	OUT_CELL	-
RXDATA[3]	126	IO_CELL	-
RXDATA_OEN[4]	127	OUT_CELL	-
RXDATA[4]	128	IO_CELL	-
RXDATA_OEN[5]	129	OUT_CELL	-
RXDATA[5]	130	IO_CELL	-
RXDATA_OEN[6]	131	OUT_CELL	-
RXDATA[6]	132	IO_CELL	-
RXDATA_OEN[7]	133	OUT_CELL	-

Pin/Enable	Register Bit	Cell Type	Device I.D.
RXDATA[7]	134	IO_CELL	-
RSX_OEN	135	OUT_CELL	-
RSX	136	IO_CELL	-
RXDATA_OEN[8]	137	OUT_CELL	-
RXDATA[8]	138	IO_CELL	-
RXDATA_OEN[9]	139	OUT_CELL	-
RXDATA[9]	140	IO_CELL	-
RXDATA_OEN[10]	141	OUT_CELL	-
RXDATA[10]	142	IO_CELL	-
RXDATA_OEN[11]	143	OUT_CELL	-
RXDATA[11]	144	IO_CELL	-
RXDATA_OEN[12]	145	OUT_CELL	-
RXDATA[12]	146	IO_CELL	-
RXDATA_OEN[13]	147	OUT_CELL	-
RXDATA[13]	148	IO_CELL	-
RXDATA_OEN[14]	149	OUT_CELL	-
RXDATA[14]	150	IO_CELL	-
RXDATA_OEN[15]	151	OUT_CELL	-
RXDATA[15]	152	IO_CELL	-
RXPRTY_OEN	153	OUT_CELL	-
RXPRTY	154	IO_CELL	-
RERR_OEN	155	OUT_CELL	-
RERR	156	IO_CELL	-
RMOD_OEN	157	OUT_CELL	-
RMOD	158	IO_CELL	-
REOP_OEN	159	OUT_CELL	-
REOP	160	IO_CELL	-
RENB	161	IN_CELL	-

Pin/Enable	Register Bit	Cell Type	Device I.D.
RPA_OEN	162	OUT_CELL	-
RPA	163	IO_CELL	-
RVAL_OEN	164	OUT_CELL	-
RVAL	165	IO_CELL	-
TRDY_OEN	166	OUT_CELL	-
TRDY	167	IO_CELL	-
TERR_OEN	168	OUT_CELL	
TERR	169	IO_CELL	-
TMOD_OEN	170	OUT_CELL	-
TMOD	171	IO_CELL	-
TEOP_OEN	172	OUT_CELL	-
TEOP	173	IO_CELL	-
TXDATA_OEN[0]	174	OUT_CELL	-
TXDATA[0]	175	IO_CELL	-
TXDATA_OEN[1]	176	OUT_CELL	-
TXDATA[1]	177	IO_CELL	-
TXDATA_OEN[2]	178	OUT_CELL	-
TXDATA[2]	179	IO_CELL	-
TXDATA_OEN[3]	180	OUT_CELL	-
TXDATA[3]	181	IO_CELL	-
TXDATA_OEN[4]	182	OUT_CELL	-
TXDATA[4]	183	IO_CELL	-
TXDATA_OEN[5]	184	OUT_CELL	-
TXDATA[5]	185	IO_CELL	-
TXDATA_OEN[6]	186	OUT_CELL	-
TXDATA[6]	187	IO_CELL	-
TXDATA_OEN[7]	188	OUT_CELL	-
TXDATA[7]	189	IO_CELL	-

Pin/Enable	Register Bit	Cell Type	Device I.D.
TSX	190	IN_CELL	-
TXPRTY_OEN	191	OUT_CELL	-
TXPRTY	192	IO_CELL	-
TXDATA_OEN[8]	193	OUT_CELL	-
TXDATA[8]	194	IO_CELL	-
TXDATA_OEN[9]	195	OUT_CELL	-
TXDATA[9]	196	IO_CELL	-
TXDATA_OEN[10]	197	OUT_CELL	-
TXDATA[10]	198	IO_CELL	-
TXDATA_OEN[11]	199	OUT_CELL	-
TXDATA[11]	200	IO_CELL	-
TXDATA_OEN[12]	201	OUT_CELL	-
TXDATA[12]	202	IO_CELL	-
TXDATA_OEN[13]	203	OUT_CELL	-
TXDATA[13]	204	IO_CELL	-
TXDATA_OEN[14]	205	OUT_CELL	-
TXDATA[14]	206	IO_CELL	-
TXDATA_OEN[15]	207	OUT_CELL	-
TXDATA[15]	208	IO_CELL	-
Unconnected	209	OUT_CELL	-
Unconnected	210	IO_CELL	-
TXCLK	211	IN_CELL	-
TXADDR[0]	212	IN_CELL	-
TXADDR[1]	213	IN_CELL	-
TXADDR[2]	214	IN_CELL	-
Unconnected	215	OUT_CELL	-
Unconnected	216	OUT_CELL	-
TXADDR[3]	217	IN_CELL	-

Pin/Enable	Register Bit	Cell Type	Device I.D.
TXADDR_OEN[4]	218	OUT_CELL	-
TXADDR[4]	219	IO_CELL	-
TXADDR[5]	220	IN_CELL	-
TXADDR[6]	221	IN_CELL	-
TXADDR[7]	222	IN_CELL	-
TXADDR[8]	223	IN_CELL	-
TXADDR[9]	224	IN_CELL	-
TXADDR[10]	225	IN_CELL	-
TXADDR[11]	226	IN_CELL	-
TXADDR[12]	227	IN_CELL	-
TPA1_OEN[0]	228	OUT_CELL	-
TPA1[0]	229	IO_CELL	-
TPA1_OEN[1]	230	OUT_CELL	-
TPA1[1]	231	IO_CELL	-
TPA1_OEN[2]	232	OUT_CELL	-
TPA1[2]	233	IO_CELL	-
TPA2_OEN[0]	234	OUT_CELL	-
TPA2[0]	235	IO_CELL	-
TPA2_OEN[1]	236	OUT_CELL	-
TPA2[1]	237	IO_CELL	-
TPA2_OEN[2]	238	OUT_CELL	-
TPA2[2]	239	IO_CELL	-
D_OEN[0]	240	OUT_CELL	-
D[0]	241	IO_CELL	-
D_OEN[1]	242	OUT_CELL	-
D[1]	243	IO_CELL	-
D_OEN[2]	244	OUT_CELL	-
D[2]	245	IO_CELL	-

Pin/Enable	Register Bit	Cell Type	Device I.D.
D_OEN[3]	246	OUT_CELL	-
D[3]	247	IO_CELL	-
D_OEN[4]	248	OUT_CELL	-
D[4]	249	IO_CELL	-
D_OEN[5]	250	OUT_CELL	-
D[5]	251	IO_CELL	-
D_OEN[6]	252	OUT_CELL	-
D[6]	253	IO_CELL	-
D_OEN[7]	254	OUT_CELL	-
D[7]	255	IO_CELL	-
D_OEN[8]	256	OUT_CELL	-
D[8]	257	IO_CELL	-
D_OEN[9]	258	OUT_CELL	-
D[9]	259	IO_CELL	-
D_OEN[10]	260	OUT_CELL	-
D[10]	261	IO_CELL	-
D_OEN[11]	262	OUT_CELL	-
D[11]	263	IO_CELL	-
D_OEN[12]	264	OUT_CELL	-
D[12]	265	IO_CELL	-
D_OEN[13]	266	OUT_CELL	-
D[13]	267	IO_CELL	-
D_OEN[14]	268	OUT_CELL	-
D[14]	269	IO_CELL	-
D_OEN[15]	270	OUT_CELL	-
D[15]	271	IO_CELL	-
A[2]	272	IN_CELL	-
A[3]	273	IN_CELL	-

Pin/Enable	Register Bit	Cell Type	Device I.D.
A[4]	274	IN_CELL	-
A[5]	275	IN_CELL	-
A[6]	276	IN_CELL	-
A[7]	277	IN_CELL	-
A[8]	278	IN_CELL	-
A[9]	279	IN_CELL	-
A[10]	280	IN_CELL	-
A[11]	281	IN_CELL	-
ALE	282	IN_CELL	-
WRB	283	IN_CELL	-
RDB	284	IN_CELL	-
CSB	285	IN_CELL	-
INTB_OEN	286	OUT_CELL	-
INTB	287	OUT_CELL	-
Logic 0	288	IN_CELL	-
Logic 0	289	IN_CELL	-
Logic 0	290	IN_CELL	-
Logic 0	291	IN_CELL	-
Logic 0	292	IN_CELL	-
Logic 0	293	IN_CELL	-
Logic 0	294	IN_CELL	-
Logic 0	295	IN_CELL	-
Logic 0	296	IN_CELL	-
Logic 0	297	IN_CELL	-
Logic 0	298	IN_CELL	-
Logic 0	299	IN_CELL	-
Logic 0	300	IN_CELL	-
Logic 0	301	IN_CELL	-

Pin/Enable	Register Bit	Cell Type	Device I.D.
Logic 0	302	IN_CELL	-
TA[12]	303	IN_CELL	-
Logic 0	304	IN_CELL	-
Logic 0	305	IN_CELL	-
Logic 0	306	IN_CELL	-
TWRB	307	IN_CELL	-
Logic 0	308	IN_CELL	-
TRDB	309	IN_CELL	-
Logic 0	310	IN_CELL	-
TA[11]	311	IN_CELL	-
Logic 0	312	IN_CELL	-
TA[10]	313	IN_CELL	-
Logic 0	314	IN_CELL	-
TA[9]	315	IN_CELL	-
Logic 0	316	IN_CELL	-
TA[8]	317	IN_CELL	-
Logic 0	318	IN_CELL	-
TA[7]	319	IN_CELL	-
Logic 0	320	IN_CELL	-
TA[6]	321	IN_CELL	-
Logic 0	322	IN_CELL	-
Logic 0	323	IN_CELL	-
Logic 0	324	IN_CELL	-
TA[5]	325	IN_CELL	-
RSTB	326	IN_CELL	-
Logic 0	327	IN_CELL	-
TA[4]	328	IN_CELL	-
Logic 0	329	IN_CELL	-

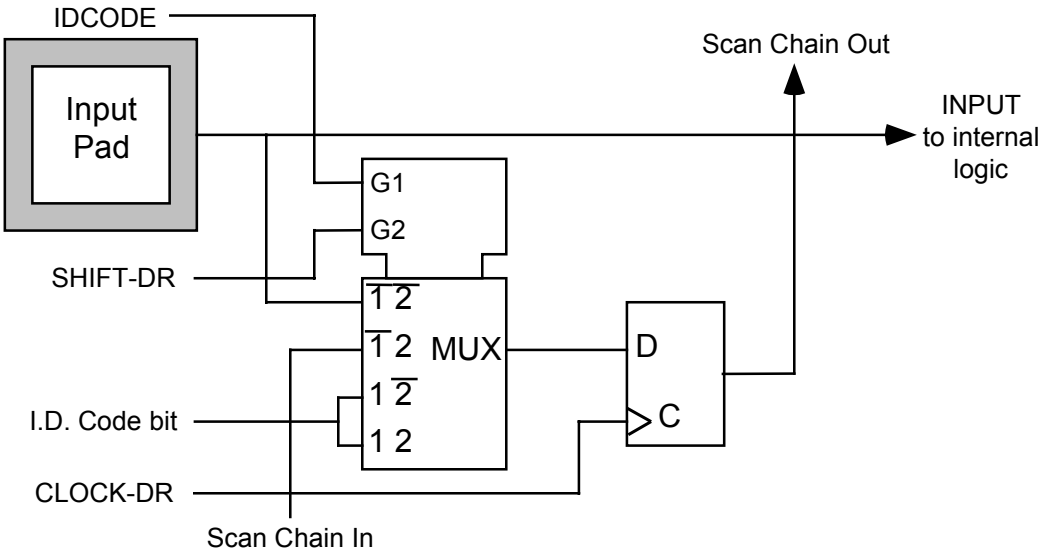
Pin/Enable	Register Bit	Cell Type	Device I.D.
TA[3]	330	IN_CELL	-
Logic 0	331	IN_CELL	-
TA[2]	332	IN_CELL	-
Logic 0	333	IN_CELL	1
TA[1]	334	IN_CELL	0
Logic 0	335	IN_CELL	1
TA[0]	336	IN_CELL	1
Logic 0	337	IN_CELL	0
Logic 0	338	IN_CELL	0
Logic 0	339	IN_CELL	1
Logic 0	340	IN_CELL	1
Logic 0	341	IN_CELL	0
Logic 0	342	IN_CELL	0
Logic 0	343	IN_CELL	0
Logic 0	344	IN_CELL	0
Logic 0	345	IN_CELL	1
Logic 0	346	IN_CELL	0
SYSCLK	347	IN_CELL	1
Logic 0	348	IN_CELL	0
Logic 0	349	IN_CELL	0
Logic 0	350	IN_CELL	0
Logic 0	351	IN_CELL	0
Logic 0	352	IN_CELL	1
Logic 0	353	IN_CELL	1
RCLK[2]	354	IN_CELL	1
RD[2]	355	IN_CELL	0
RCLK[1]	356	IN_CELL	0
RD[1]	357	IN_CELL	1

Pin/Enable	Register Bit	Cell Type	Device I.D.
RCLK[0]	358	IN_CELL	1
RD[0]	359	IN_CELL	1
Logic 0	360	IN_CELL	0
Logic 0	361	IN_CELL	0
Logic 0	362	IN_CELL	1
Logic 0	363	IN_CELL	0
Logic 0	364	IN_CELL	0
TDO		TAP Output	-
TDI		TAP Input	-
TCK		TAP Clock	-
TMS		TAP Input	-
TRSTB		TAP Input	-

Notes:

1. Register bit 364 is the first bit of the scan chain (closest to TDI).
2. Enable cell pinname_OEN, tristates pin pinname when set high.
3. Cells ‘Logic 0’ and ‘Logic 1’ are Input Observation cells whose input pad is bonded to VSS or VDD internally.
4. Cells titled ‘Unconnected’ are Output or Bi-directional cells whose pad is unconnected to the device package. In the case of bi-directional cells, the pad always drives (i.e. never tri-states) and the pad input is the same logic value as the pad output.

Figure 5 – Input Observation Cell (IN_CELL)



In this diagram and those that follow, CLOCK-DR is equal to TCK when the current controller state is SHIFT-DR or CAPTURE-DR, and unchanging otherwise. The multiplexor in the center of the diagram selects one of four inputs, depending on the status of select lines G1 and G2. The ID Code bit is as listed in the table above.

Figure 6 – Output Cell (OUT_CELL)

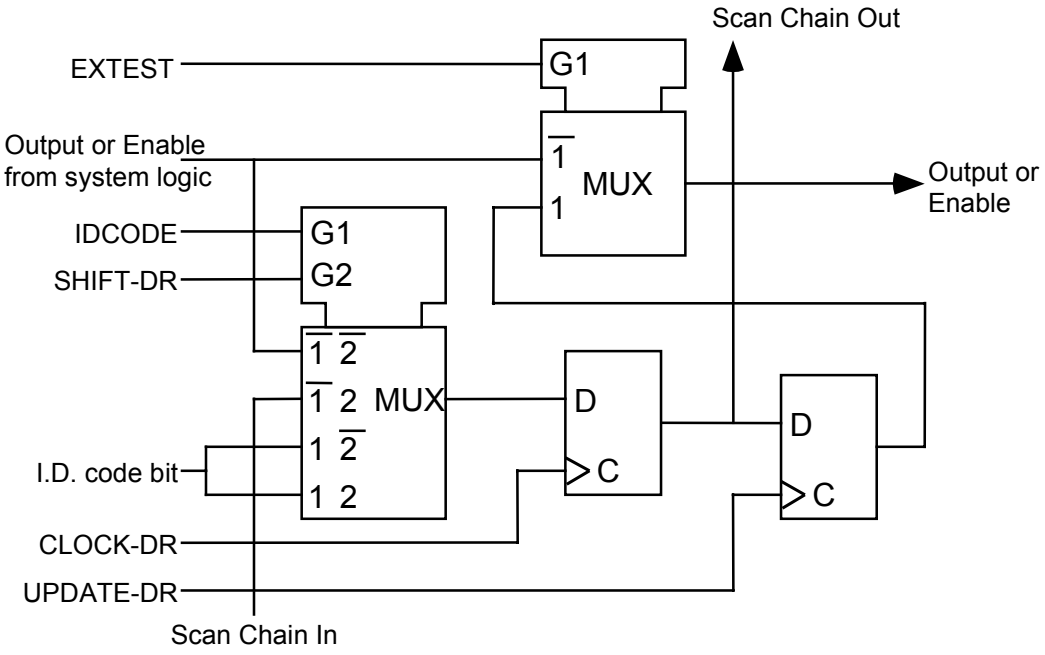


Figure 7 – Bi-directional Cell (IO_CELL)

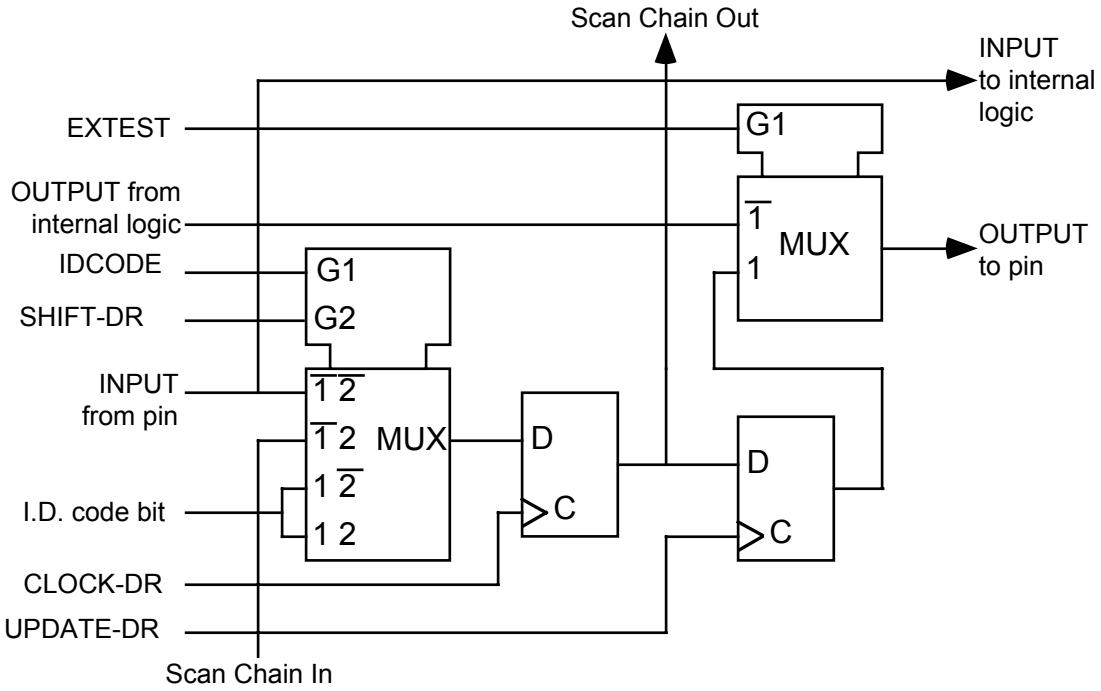
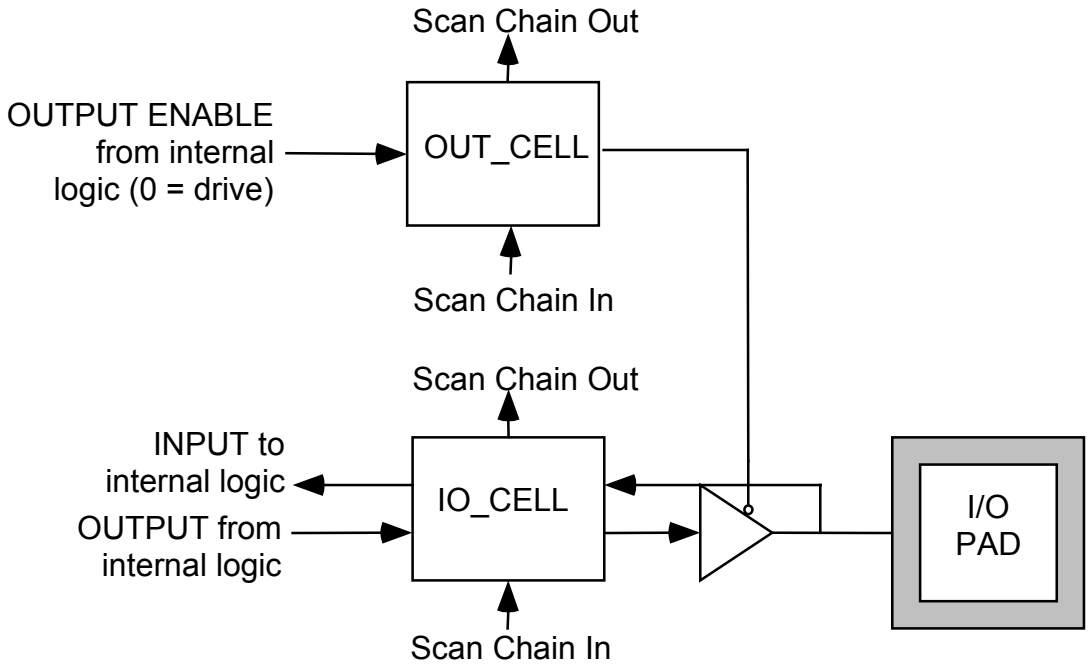


Figure 8 – Layout of Output Enable and Bi-directional Cells



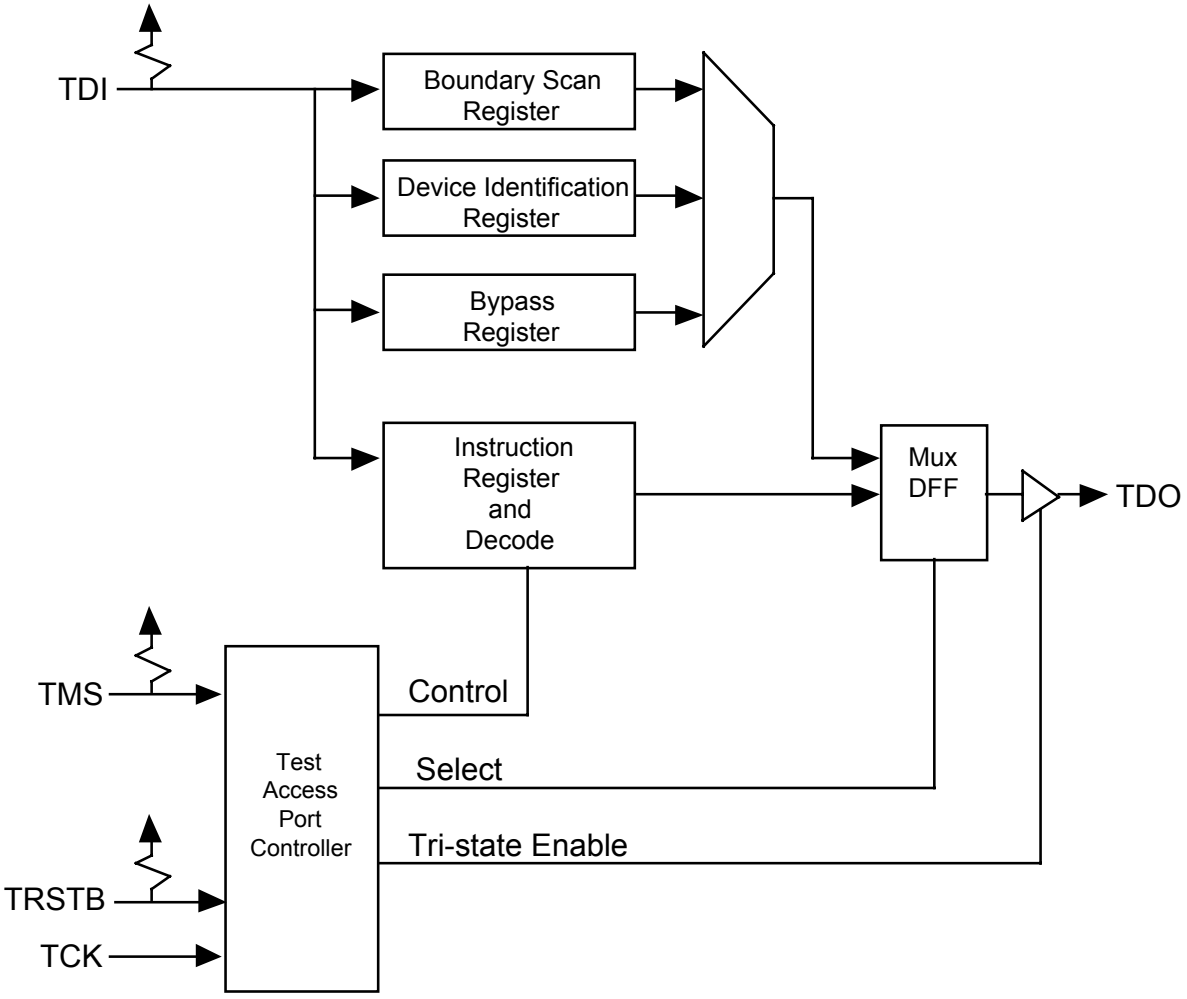
11 OPERATIONS

This section presents operating details for the JTAG boundary scan feature.

11.1 JTAG Support

The FREEDM-84A672 supports the IEEE Boundary Scan Specification as described in the IEEE 1149.1 standards. The Test Access Port (TAP) consists of the five standard pins, TRSTB, TCK, TMS, TDI and TDO used to control the TAP controller and the boundary scan registers. The TRSTB input is the active low reset signal used to reset the TAP controller. TCK is the test clock used to sample data on input, TDI and to output data on output, TDO. The TMS input is used to direct the TAP controller through its states. The basic boundary scan architecture is shown below.

Figure 9 – Boundary Scan Architecture



The boundary scan architecture consists of a TAP controller, an instruction register with instruction decode, a bypass register, a device identification register and a boundary scan register. The TAP controller interprets the TMS input and generates control signals to load the instruction and data registers. The instruction register with instruction decode block is used to select the test to be executed and/or the register to be accessed. The bypass register offers a single bit delay from primary input, TDI to primary output , TDO. The device identification register contains the device identification code.

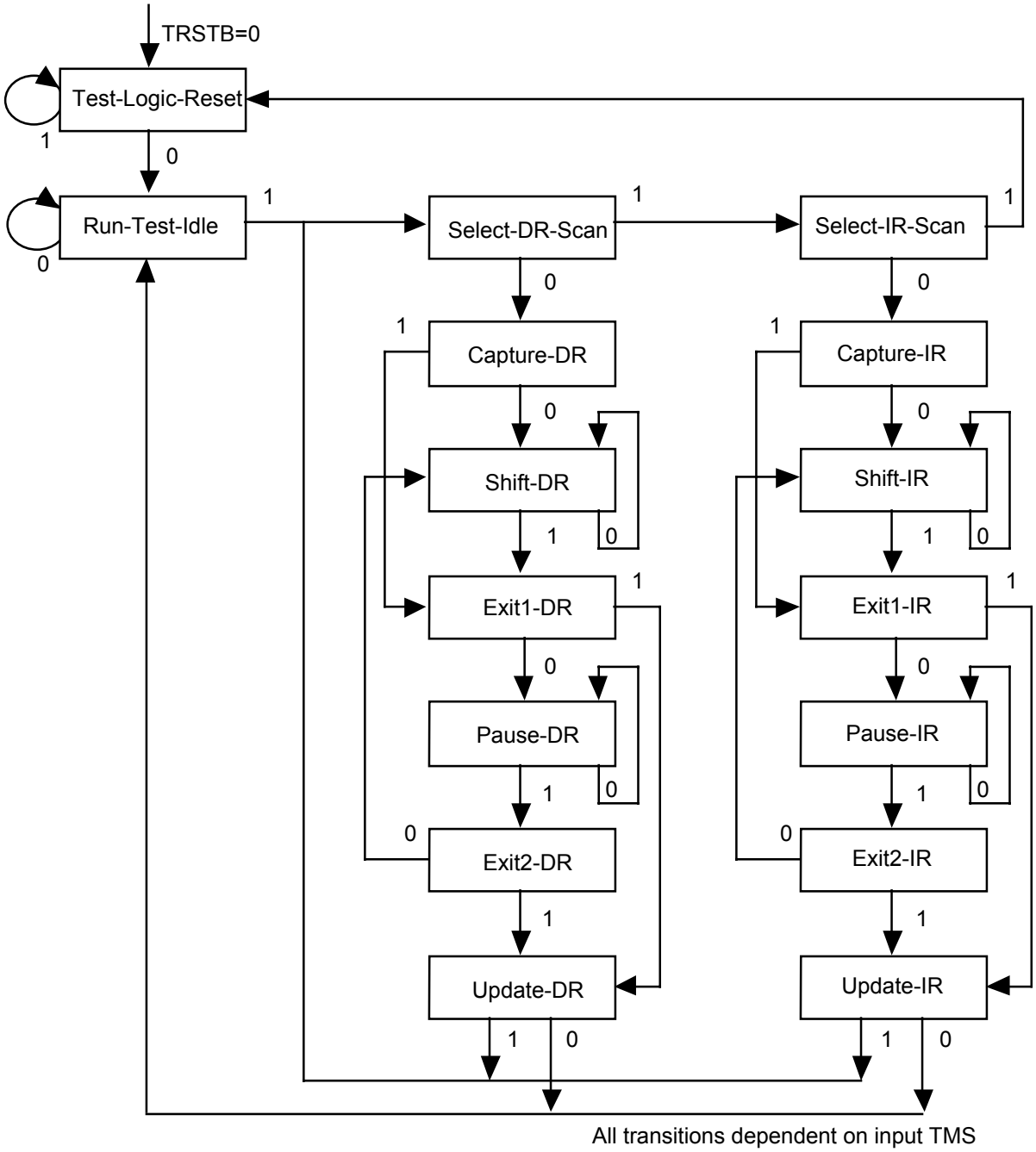
The boundary scan register allows testing of board inter-connectivity. The boundary scan register consists of a shift register placed in series with device inputs and outputs. Using the boundary scan register, all digital inputs can be

sampled and shifted out on primary output TDO. In addition, patterns can be shifted in on primary input, TDI and forced onto all digital outputs.

TAP Controller

The TAP controller is a synchronous finite state machine clocked by the rising edge of primary input, TCK. All state transitions are controlled using primary input, TMS. The finite state machine is described below.

Figure 10 – TAP Controller Finite State Machine



Test-Logic-Reset

The test logic reset state is used to disable the TAP logic when the device is in normal mode operation. The state is entered asynchronously by asserting input, TRSTB. The state is entered synchronously regardless of the current TAP controller state by forcing input, TMS high for 5 TCK clock cycles. While in this state, the instruction register is set to the IDCODE instruction.

Run-Test-Idle

The run test/idle state is used to execute tests.

Capture-DR

The capture data register state is used to load parallel data into the test data registers selected by the current instruction. If the selected register does not allow parallel loads or no loading is required by the current instruction, the test register maintains its value. Loading occurs on the rising edge of TCK.

Shift-DR

The shift data register state is used to shift the selected test data registers by one stage. Shifting is from MSB to LSB and occurs on the rising edge of TCK.

Update-DR

The update data register state is used to load a test register's parallel output latch. In general, the output latches are used to control the device. For example, for the EXTEST instruction, the boundary scan test register's parallel output latches are used to control the device's outputs. The parallel output latches are updated on the falling edge of TCK.

Capture-IR

The capture instruction register state is used to load the instruction register with a fixed instruction. The load occurs on the rising edge of TCK.

Shift-IR

The shift instruction register state is used to shift both the instruction register and the selected test data registers by one stage. Shifting is from MSB to LSB and occurs on the rising edge of TCK.

Update-IR

The update instruction register state is used to load a new instruction into the instruction register. The new instruction must be scanned in using the Shift-IR state. The load occurs on the falling edge of TCK.

The Pause-DR and Pause-IR states are provided to allow shifting through the test data and/or instruction registers to be momentarily paused.

Boundary Scan Instructions

The following is a description of the standard instructions. Each instruction selects a serial test data register path between input, TDI and output, TDO.

BYPASS

The bypass instruction shifts data from input, TDI to output, TDO with one TCK clock period delay. The instruction is used to bypass the device.

EXTEST

The external test instruction allows testing of the interconnection to other devices. When the current instruction is the EXTEST instruction, the boundary scan register is place between input, TDI and output, TDO. Primary device inputs can be sampled by loading the boundary scan register using the Capture-DR state. The sampled values can then be viewed by shifting the boundary scan register using the Shift-DR state. Primary device outputs can be controlled by loading patterns shifted in through input TDI into the boundary scan register using the Update-DR state.

SAMPLE

The sample instruction samples all the device inputs and outputs. For this instruction, the boundary scan register is placed between TDI and TDO. Primary device inputs and outputs can be sampled by loading the boundary scan register using the Capture-DR state. The sampled values can then be viewed by shifting the boundary scan register using the Shift-DR state.

IDCODE

The identification instruction is used to connect the identification register between TDI and TDO. The device's identification code can then be shifted out using the Shift-DR state.

STCTEST

The single transport chain instruction is used to test out the TAP controller and the boundary scan register during production test. When this instruction is the current instruction, the boundary scan register is connected between TDI and TDO. During the Capture-DR state, the device identification code is loaded into the boundary scan register. The code can then be shifted out output, TDO using the Shift-DR state.

INTEST

The internal test instruction is used to exercise the device's internal core logic. When this instruction is the current instruction, the boundary scan register is connected between TDI and TDO. During the Update-DR state, patterns shifted in on input, TDI are used to drive primary inputs. During the Capture-DR state, primary outputs are sampled and loaded into the boundary scan register.

12 FUNCTIONAL TIMING

12.1 SBI DROP BUS Interface Timing

Figure 11 – T1/E1 DROP BUS Functional Timing

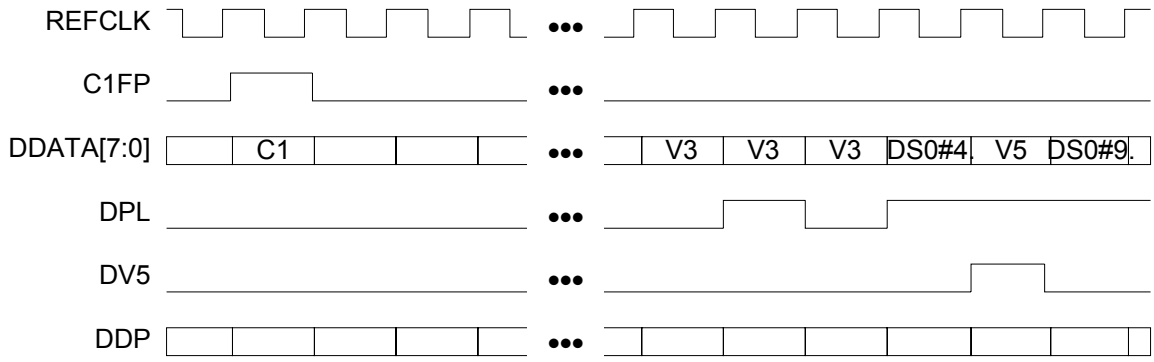


Figure 11 illustrates the operation of the SBI DROP BUS, using a negative justification on the second to last V3 octet as an example. The justification is indicated by asserting DPL high during the V3 octet. The timing diagram also shows the location of one of the tributaries by asserting DV5 high during the V5 octet.

Figure 12 – DS3 DROP BUS Functional Timing

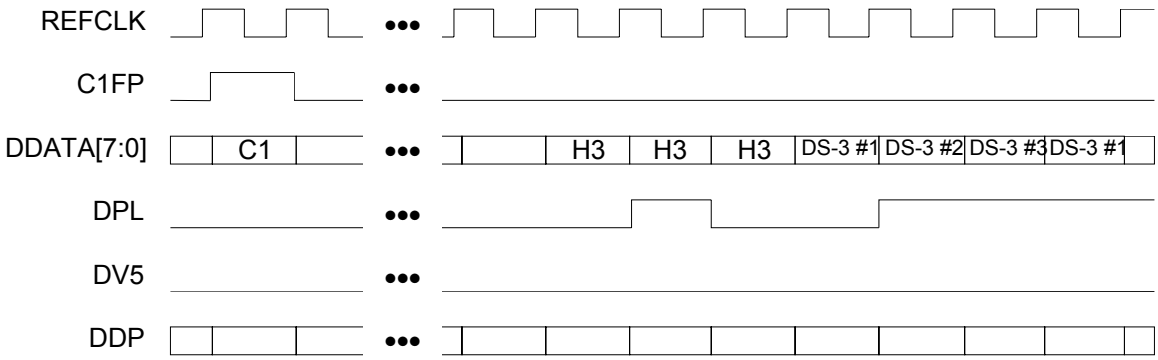


Figure 12 shows three DS-3 tributaries mapped onto the SBI bus. A negative justification is shown for DS-3 #2 during the H3 octet with DPL asserted high. A positive justification is shown for DS-3#1 during the first DS-3#1 octet after H3 which has DPL asserted low.

12.2 SBI ADD BUS Interface Timing

Figure 13 – DS3 Add Bus Adjustment Request Functional Timing

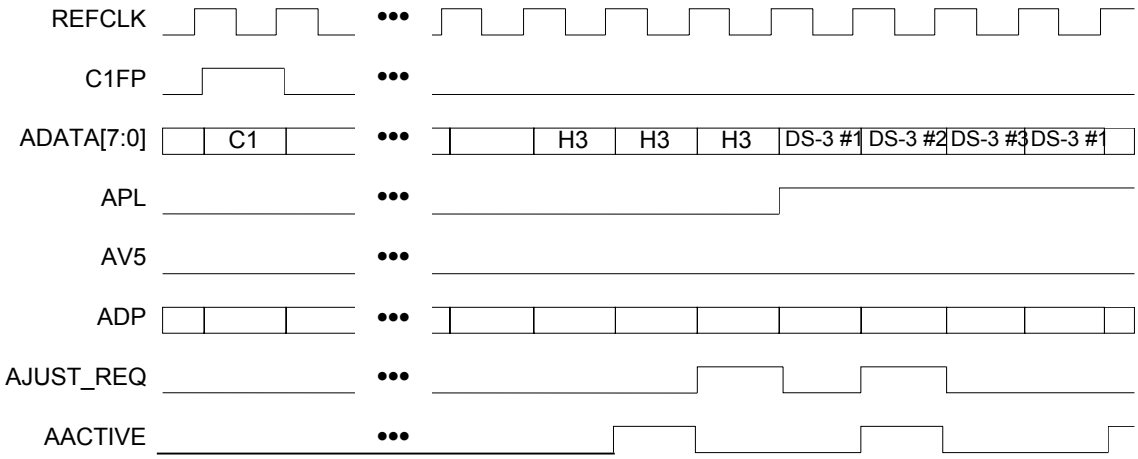
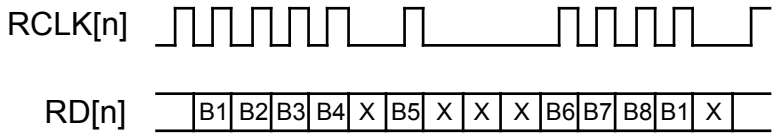


Figure 13 illustrates the operation of the SBI ADD BUS, using positive and negative justification requests as an example. (The responses to the justification requests would take effect during the next multi-frame.) The negative justification request occurs on the DS-3#3 tributary when AJUST_REQ is asserted high during the H3 octet. The positive justification occurs on the DS-3#2 tributary when AJUST_REQ is asserted high during the first DS-3#2 octet after the H3 octet. The AACTIVE signal is shown for the case in which FREEDM-84A672 is only driving DS-3#2 onto the SBI ADD bus.

12.3 Receive Link Timing

The timing relationship of the receive clock (RCLK[n]) and data (RD[n]) signals is shown in Figure 14. 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 14) 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 14) are squelched by holding RCLK[n] quiescent. In Figure 14, 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.

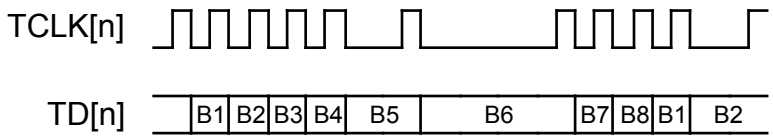
Figure 14 – Receive Link Timing



12.4 Transmit Link Timing

The timing relationship of the transmit clock (TCLK[n]) and data (TD[n]) signals is shown in Figure 15. 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 15) and ending with the least significant bit (B8 in Figure 15). 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 15, 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 15, 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 15 – Transmit Link Timing



12.5 Receive APPI Timing

The receive Any-PHY packet interface (APPI) timing is shown in Figure 16 through Figure 19. The FREEDM-84A672 device provides data to an external controller using the receive APPI. The following discussion surrounding the receive APPI functional timing assumes that multiple FREEDM-84A672 devices share a single external controller. All Rx APPI signals are shared between the FREEDM-84A672 devices.

Figure 16 – Receive APPI Timing (Normal Transfer)

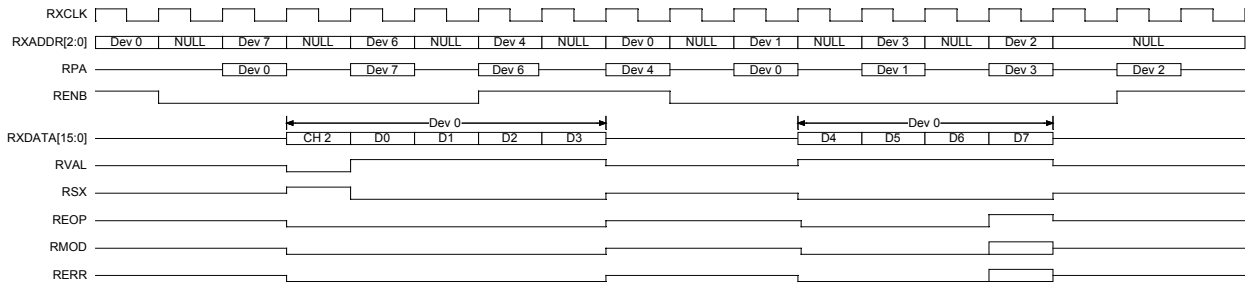


Figure 16 shows the transfer of an 8 word packet across the Rx APPI from FREEDM-84A672 device 0, channel 2. In this example, seven FREEDM-84A672 devices are sharing the Rx APPI, with device 5 being the null address.

The data transfer begins when the external controller selects FREEDM-84A672 device 0 by placing that address on the RXADDR[2:0] inputs and setting RENB high. The external controller sets RENB low in the next RXCLK cycle to commence data transfer across the Rx APPI. The FREEDM-84A672 samples RENB low and responds by asserting RSX one RXCLK cycle later. The start of all burst data transfers is qualified with RSX and an in-band channel address on RXDATA[15:0] to associate the data to follow with a HDLC channel.

During the cycle when D2 is placed on RXDATA[15:0], the external controller is unable to accept any further data and sets RENB high. Two RXCLK cycles later, the FREEDM-84A672 tristates the Rx APPI. The external controller may hold RENB high for an indeterminate number of RXCLK cycles. The FREEDM-84A672 will wait until the external controller returns RENB low. Because the FREEDM-84A672 does not support interrupted data transfers on the Rx APPI, the external controller must reselect FREEDM-84A672 device 0 or output a null address during the clock cycle before it returns RENB low. However, while RENB remains high, the address on the RXADDR[2:0] signals may change. When the FREEDM-84A672 device 0 samples RENB low, it continues data transfer by providing D4 on RXDATA[15:0]. Note that if D3 were the final word of the packet (Status), in response to sampling REOP high, the external controller does not have to reselect FREEDM-84A672 device 0. This is shown in Figure 19.

The FREEDM-84A672 will not pause burst data transfers across the Rx APPI.

The FREEDM-84A672 automatically deselects at the end of all burst data transfers. The FREEDM-84A672 must be reselected before any further data will be transferred across the Rx APPI.

The RVAL and REOP signals indicate the presence and end of valid packet data respectively. The RERR and RMOD signals are only valid at the end of a packet and are qualified with the REOP signal. When a packet is errored, the FREEDM-84A672 may be programmed to overwrite RXDATA[15:0] in the final word of packet transfer with status information indicating the cause of the error. RXDATA[15:0] is not modified if a packet is error free.

The RXADDR[2:0] signals serve to poll FREEDM-84A672 devices as well as for selection. During data transfer, the RXADDR[2:0] signals continue to poll the FREEDM-84A672 devices sharing the Rx APPI. Polled results are returned on the RPA signal. Note that each poll address is separated by a NULL address to generate tristate turn-around cycle in order to prevent multiple FREEDM-84A672 devices from briefly driving RPA. If RPA is a point-to-point signal for each FREEDM-84A672 device on the board, then the tristate turn-around cycle is not required, thereby effectively doubling the polling bandwidth at the expense of extra signals.

Polled results reflect the status of the two FIFOs in the RAPI672. Polled responses always refer to the next data transfer. In other words, polled responses during or after the RXCLK cycle where RSX is set high refer to the FIFO which is not involved in the current data transfer. For example, once FIFO one begins transferring data on the Rx APPI (RSX set high), any polls against that FREEDM-84A672 device respond with the status of FIFO two. This allows the external controller to gather knowledge about the FIFO not involved in the current data transfer so that it can anticipate reselecting that FREEDM-84A672 device (via RENB) to maximize bandwidth on the Rx APPI (shown in Figure 18).

Figure 17 – Receive APPI Timing (Auto Deselection)

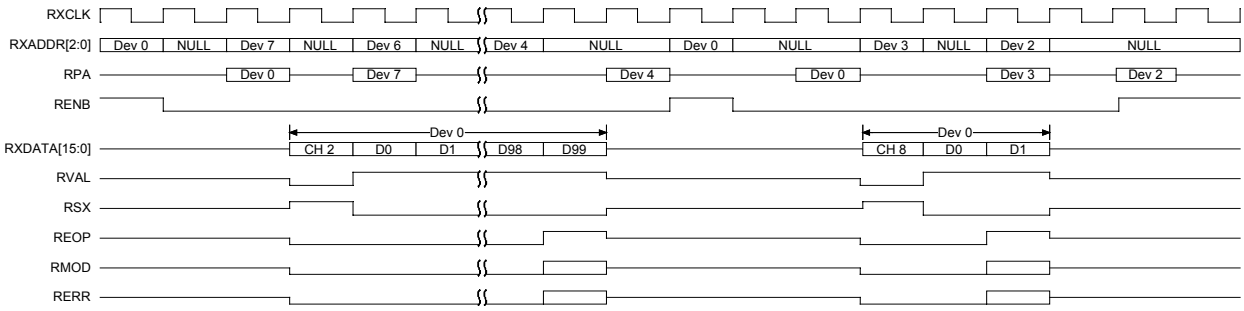


Figure 17 shows the transfer of a 100 word packet across the Rx APPI from FREEDM-84A672 device 0, channel 2 followed by the transfer of a 2 word packet from FREEDM-84A672 device 0, channel 8. More importantly, Figure 17 illustrates that, for back-to-back transfers from the same FREEDM-84A672

(device 0), it must be reselected before any further data is provided on the Rx APPI.

At the end of the first 100 word packet transfer across the Rx APPI, the FREEDM-84A672 automatically deselects and must be reselected before the second two word packet is transferred. When the external controller samples REOP high, it recognizes that the burst transfer has completed. Two RXCLK cycles later, the external controller reselects FREEDM-84A672 device 0 by setting RENB high and placing address 0 on the RXADDR[2:0] signals. When the FREEDM-84A672 samples RENB low, it begins the next data transfer as before.

Figure 18 – Receive APPI Timing (Optimal Reselection)

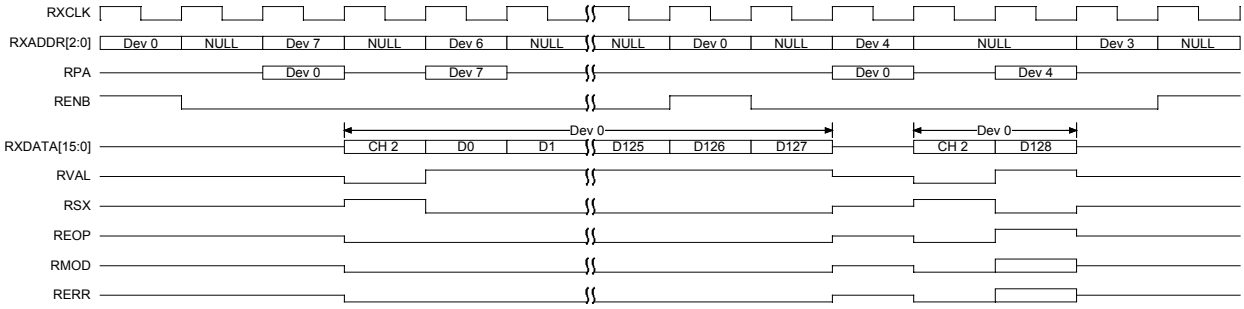


Figure 18 shows optimal bandwidth utilization across the Rx APPI.

With knowledge that the maximum burst data transfer (excluding channel address prepend) is 256 bytes, i.e. 128 words, the external controller sets RENB high when the 127th word (D126) is placed on RXDATA[15:0] in anticipation of the end of a burst transfer. The FREEDM-84A672 completes the burst data transfer and tristates the Rx APPI one RXCLK cycle after RENB is sampled high. Because the burst data transfer is complete and RENB is immediately returned low following selection, the FREEDM-84A672 immediately begins the next data transfer following the single turn-around cycle.

The protocol dictates that at least one tristate turn-around cycle be inserted between data transfers, even if the external controller is reselecting the same FREEDM-84A672 device. In other words, Figure 18 shows the earliest possible time that the external controller could have set RENB high to reselect FREEDM-84A672 device 0.

Figure 19 – Receive APPI Timing (Boundary Condition)

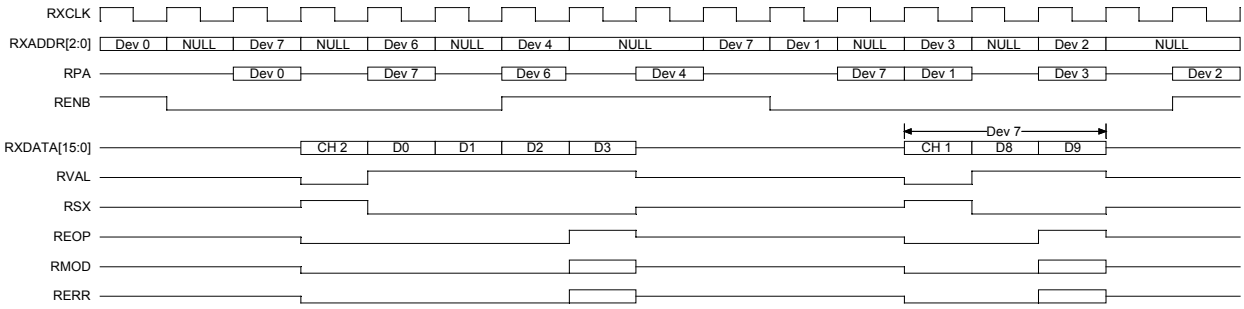


Figure 19 shows the boundary condition where a packet transfer completes shortly after the external controller has set RENB high to pause the FREEDM-84A672 device. The second data transfer is the final two words of a packet for FREEDM-84A672 device 7, channel 1.

When FREEDM-84A672 device 0 places D2 on RXDATA[15:0], the external controller sets RENB high to pause the FREEDM-84A672 device. In the following RXCLK cycle, the FREEDM-84A672 provides D3 on RXDATA[15:0] and sets REOP high to conclude packet transfer. The external controller samples REOP high while RENB is high and recognizes that the packet transfer is complete. The external controller now knows that it doesn't need to reselect FREEDM-84A672 device 0, but can select another FREEDM-84A672 device sharing the Rx APPI. The external controller decides to select FREEDM-84A672 device 7 by placing this address on the RXADDR[2:0] signals. The external controller sets RENB low to commence data transfer from FREEDM-84A672 device 7.

12.6 Transmit APPI Timing

The transmit Any-PHY packet interface (APPI) timing is shown in Figure 20. An external controller provides data to the FREEDM-84A672 device using the transmit APPI. The following discussion surrounding the transmit APPI functional timing assumes that multiple FREEDM-84A672 devices share a single external controller. The three most significant bits of TXADDR[12:0] perform device selection for purposes of polling while the ten least significant bits provide the channel poll address. All Tx APPI signals are shared between the FREEDM-84A672 devices.

Figure 20 – Transmit APPI Timing (Normal Transfer)

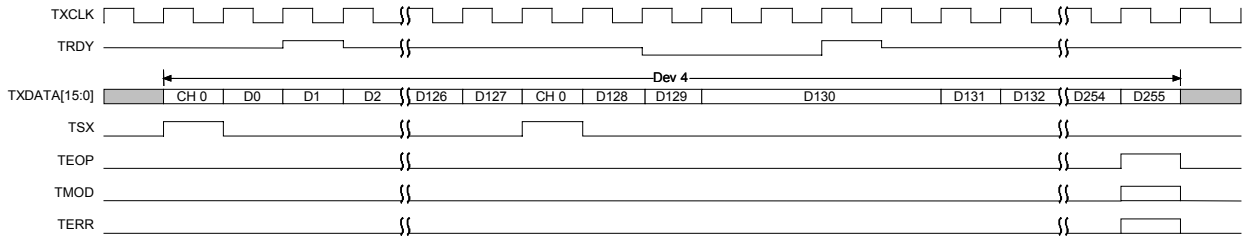


Figure 20 shows transfer of a 256 word packet on the Tx APPI of FREEDM-84A672 device 4, channel 0. The maximum burst data transfer (excluding channel address prepend) is 128 words, so two data transfers are required to complete the transfer of the 256 word packet.

The start of all burst data transfers is qualified with the TSX signal and an in-band channel address on TXDATA[15:0] to associate the data to follow with a HDLC channel. The TEOP signal indicates the end of valid packet data. The TMOD and TERR signals held low except at the end of a packet (TEOP set high).

The FREEDM-84A672 starts driving the TRDY signal one TXCLK cycle after TSX is sampled high. Upon sampling the TRDY signal high, the external controller completes the current burst data transfer. The FREEDM-84A672 tristates the TRDY signal one TXCLK cycle after it has been driven high. This is the case for the first burst data transfer in Figure 20. In the second burst data transfer, the FREEDM-84A672 drives the TRDY signal low to indicate that the FIFOs in the TAPI672 are full and no further data may be transferred. Upon sampling the TRDY signal low, the external controller must hold the last valid word of data on TXDATA[15:0]. The FREEDM-84A672 may drive TRDY low for an indeterminate number of TXCLK cycles. During this time, the external controller must wait and is not permitted to begin another burst data transfer until TRDY is sampled high. When the TAPI672 has at least one empty FIFO, the FREEDM-84A672 drives the TRDY signal high. Upon sampling the TRDY signal high, the external controller completes the current burst data transfer. The FREEDM-84A672 tristates the TRDY signal one TXCLK cycle after it has been driven high.

The external controller must sample the TRDY signal high before it can begin the next burst data transfer. This prevents the external controller from bombarding the FREEDM-84A672 device with small packets and allows the FREEDM-84A672 to perform the necessary house-keeping and clean-up associated with the ending of burst data transfers. This protocol also ensures that transitions between burst data transfers do not require any extra per channel storage, thereby simplifying implementation of both the external controller and the FREEDM-84A672 device. Figure 21 illustrates this condition.

Figure 21 – Transmit APPI Timing (Special Conditions)

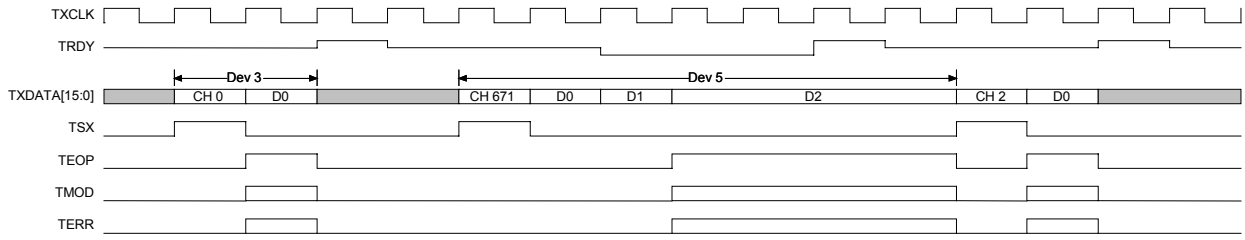
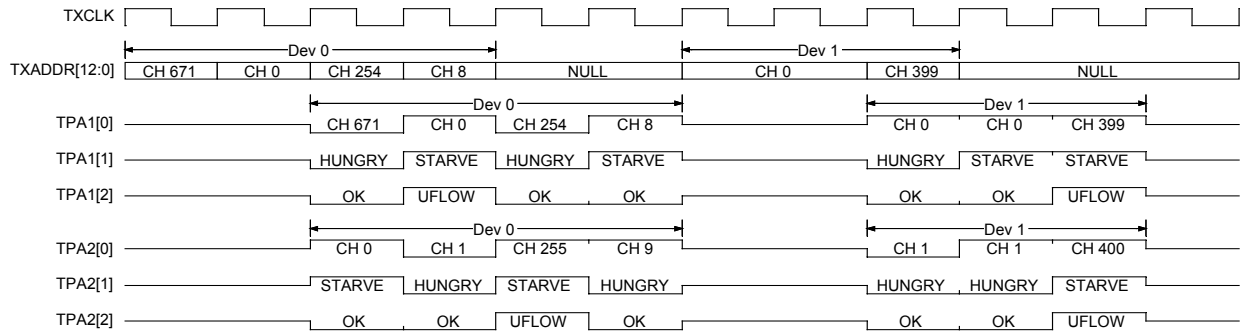


Figure 21 shows two special conditions – (1) the transfer of a one word packet illustrating how the external controller must wait until TRDY has been sampled high before the next data transfer can begin, and (2) the transfer of a packet which completes when TRDY is set low illustrating that although the packet has been completely transferred, the external controller must still wait until TRDY has been sampled high before the next data transfer can begin.

The first data transfer is a single word packet for FREEDM-84A672 device 3, channel 0. The FREEDM-84A672 asserts TRDY high one TXCLK cycle after TSX is sampled high. The Tx APPI protocol dictates that the external controller must wait until TRDY is sampled high before beginning the next data transfer for FREEDM-84A672 device 5, channel 671. The external controller must hold the last valid word on TXDATA[15:0] until TRDY is sampled high. In this case, that data is a don't care. The FREEDM-84A672 tristates the TRDY signal one TXCLK cycle after it has been driven high.

The second transfer is a three word packet which completes transfer in the same TXCLK cycle that TRDY is sampled low by the external controller. Again, the external controller must hold the last valid word on TXDATA[15:0] until TRDY is sampled high. In this case, that data is D2, the last word of the packet. The FREEDM-84A672 may drive TRDY low for an indeterminate number of TXCLK cycles. During this time, the external controller must wait and is not permitted to begin another burst data transfer until TRDY is sampled high. When the external controller samples TRDY high, the current burst transfer is deemed to be complete and the external controller may begin the next data transfer. The FREEDM-84A672 tristates the TRDY signal one TXCLK cycle after it has been driven high.

Figure 22 – Transmit APPI Timing (Polling)



Polling is completely decoupled from device and channel selection on the Tx APPI. Accordingly, the TXADDR[12:0] signals continue to provide only a poll address for any of the FREEDM-84A672 devices sharing the Tx APPI. The most significant three bits provide the device address and the least significant ten bits provide the channel address. Poll results are returned on the TPAn[2:0] signals. The TPAn[0] bit indicates whether or not space exists in the channel FIFO for data (high means space exists in the channel FIFO) and the TPAn[1] bit indicates whether or not that polled channel FIFO is at risk of underflowing and should be provided data soon (high means the channel FIFO is at risk of underflowing). The TPAn[2] bit indicates whether or not an underflow condition has occurred on the polled channel FIFO (high means an underflow condition occurred on that channel). In Figure 22, channel 671 in device 0 reports that space does not exist for data in the channel FIFO, that there is currently no risk of underflow on that channel (hungry) and that an underflow event has not occurred on this channel since the last poll. Channel 0 in device 0 reports that space exists for data in the channel FIFO, that there is currently a risk of underflow on that channel (starving) and that an underflow event has occurred on this channel since the last poll. Polled results for two channels provide a two fold increase in the polling bandwidth on the Tx APPI to accommodate the high density of 672 channels.

13 ABSOLUTE MAXIMUM RATINGS

Maximum rating are the worst case limits that the device can withstand without sustaining permanent damage. They are not indicative of normal operating conditions.

Table 31 – FREEDM-84A672 Absolute Maximum Ratings

Case Temperature under Bias	-40°C to +85°C
Storage Temperature	-40°C to +125°C
Supply Voltage (+3.3 Volt $V_{DD3.3}$)	-0.3V to +4.6V
Supply Voltage (+2.5 Volt $V_{DD2.5}$)	-0.3V to +3.5V
Volatge on Any Pin	-0.3V to $V_{DD3.3}$ + 0.3V
Static Discharge Voltage	±1000 V
Latch-Up Current	±100 mA
DC Input Current	±20 mA
Lead Temperature	+230°C
Absolute Maximum Junction Temperature	+150°C

14 D.C. CHARACTERISTICS

(T_A = -40°C to +85°C, $V_{DD3.3}$ = 3.0 to 3.6 V, $V_{DD2.5}$ = 2.3 to 2.7 V)

Table 32 – FREEDM-84A672 D.C. Characteristics

Symbol	Parameter	Min	Typ	Max	Units	Conditions
$V_{DD3.3}$	3.3V Power Supply	3.0	3.3	3.6	Volts	Note 4.
$V_{DD2.5}$	2.5V Power Supply	2.3	2.5	2.7	Volts	Note 4.
V_{IL}	Any-PHY Input Low Voltage	-0.5		0.6	Volts	
V_{IH}	Any-PHY Input High Voltage	2.0		$V_{DD3.3}$ + 0.5	Volts	
V_{OL}	Output or Bi-directional Low Voltage			0.4	Volts	I_{OL} = -8 mA for all outputs except TDO where I_{OL} = -4 mA. Note 3.
V_{OH}	Output or Bi-directional High Voltage	2.4			Volts	I_{OH} = 8 mA for all outputs except TDO where I_{OH} = 4 mA. Note 3.
V_{T+}	Schmitt Triggered Input High Voltage	2.0		$V_{DD3.3}$ + 0.5	Volts	
V_{T-}	Schmitt Triggered Input Low Voltage	-0.2		0.6	Volts	
I_{ILPU}	Input Low Current	+10	45	+100	µA	V_{IL} = GND, Notes 1, 3, 4.
I_{IHPU}	Input High Current	-10	0	+10	µA	V_{IH} = V_{DD}, Notes 1, 3
I_{IL}	Input Low Current	-10	0	+10	µA	V_{IL} = GND, Notes 2, 3
I_{IH}	Input High Current	-10	0	+10	µA	V_{IH} = V_{DD}, Notes 2, 3
C_{IN}	Input Capacitance		5		pF	Excludes package. Package typically 2 pF. Note 4.
C_{OUT}	Output Capacitance		5		pF	All pins. Excludes package. Package typically 2 pF. Note 4.

Symbol	Parameter	Min	Typ	Max	Units	Conditions
C_{IO}	Bi-directional Capacitance		5		pF	All pins. Excludes package. Package typically 2 pF. Note 4.
L_{PIN}	Pin Inductance		2		nH	All pins. Note 4.
I_{DDOP}	Operating Current.		440		mA	$V_{DD2.5}$ = 2.7V, Outputs Unloaded. All 3 SPEs on SBI interface active. Note 4.

Notes on D.C. Characteristics:

1. Input pin or bi-directional pin with internal pull-up resistor.
2. Input pin or bi-directional pin without internal pull-up resistor.
3. Negative currents flow into the device (sinking), positive currents flow out of the device (sourcing).
4. Typical values are given as a design aid. The product is not tested to the typical values given in the data sheet.

15 FREEDM-84A672 TIMING CHARACTERISTICS

(T_A = -40°C to +85°C, $V_{DD3.3}$ = 3.0 to 3.6 V, $V_{DD2.5}$ = 2.3 to 2.7 V)

Table 33 – Clocks and SBI Frame Pulse (Figure 23)

Symbol	Description	Min	Max	Units
	REFCLK Frequency	19.44 -50 ppm	19.44 +50 ppm	MHz
	REFCLK Duty Cycle	40	60	%
	FASTCLK Frequency (51.84 MHz)	51.84 -50 ppm	51.84 +50 ppm	MHz
	FASTCLK Frequency (44.928 MHz)	44.928 -50 ppm	44.928 +50 ppm	MHz
	FASTCLK Frequency (66 MHz)	66 -50 ppm	66 +50 ppm	MHz
	FASTCLK Duty Cycle	40	60	%
	SYSCLK Frequency	45 -50 ppm	45 +50 ppm	MHz
	SYSCLK Duty Cycle	40	60	%
TS_{C1FP}	C1FP Set-Up Time to REFCLK	4		ns
TH_{C1FP}	C1FP Hold Time to REFCLK	1		ns
$TP_{C1FPOUT}$	REFCLK to C1FPOUT Valid	2	20	ns

Notes on Input Timing:

1. When a set-up time is specified between an input and a clock, the set-up time is the time in nanoseconds from the 1.4 Volt point of the input to the 1.4 Volt point of the clock.

2. When a hold time is specified between an input and a clock, the hold time is the time in nanoseconds from the 1.4 Volt point of the clock to the 1.4 Volt point of the input.

Notes on Output Timing:

1. Output propagation delay time is the time in nanoseconds from the 1.4 Volt point of the reference signal to the 1.4 Volt point of the output.

2. Maximum and minimum output propagation delays are measured with a 100 pF load on the Any-PHY interface outputs, 20 pF load on the TD[2:0] outputs and 50pF load on all other outputs. Maximum propagation delay for TD[2:0] increases by typically 1 ns for each 10 pF of extra load.

3. Output tristate delay is the time in nanoseconds from the 1.4 Volt point of the reference signal to the point where the total current delivered through the output is less than or equal to the leakage current.

Figure 23 – SBI Frame Pulse Timing

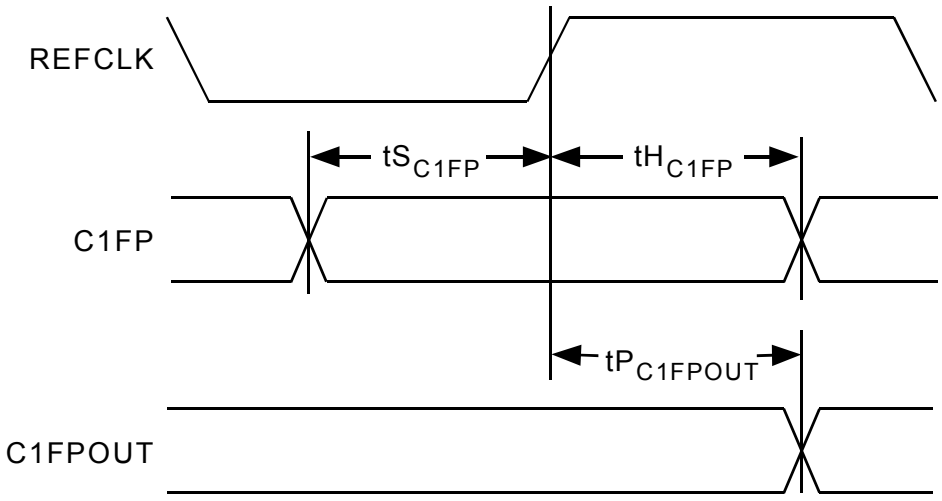


Table 34 – SBI DROP BUS (Figure 24)

Symbol	Description	Min	Max	Units
$tS_{SBIDROP}$	All SBI DROP BUS Inputs Set-Up Time to REFCLK	4		ns
$tH_{SBIDROP}$	All SBI DROP BUS Inputs Hold Time to REFCLK	1		ns

Figure 24 – SBI DROP BUS Timing

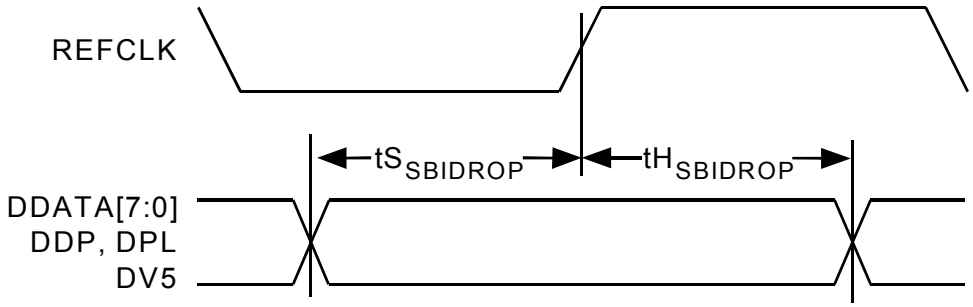


Table 35 – SBI ADD BUS (Figure 25 to Figure 26)

Symbol	Description	Min	Max	Units
tS_{SBIADD}	AJUST_REQ Set-Up Time to REFCLK	4		ns
tH_{SBIADD}	AJUST_REQ Hold Time to REFCLK	1		ns
$tP_{AACTIVE}$	REFCLK to AACTIVE Valid	2	18	ns
tP_{SBIADD}	REFCLK to All SBI ADD BUS Outputs (except AACTIVE) Valid	2	20	ns
tZ_{SBIADD}	REFCLK to All SBI ADD BUS Outputs (except AACTIVE) Tristate	2	20	ns
tP_{OUTEN}	ADETECT[1] and ADETECT[0] low to All SBI ADD BUS Outputs (except AACTIVE) Valid	0	15	ns
tZ_{OUTEN}	ADETECT[1] or ADETECT[0] high to All SBI ADD BUS Outputs (except AACTIVE) Tristate	0	15	ns

Figure 25 – SBI ADD BUS Timing

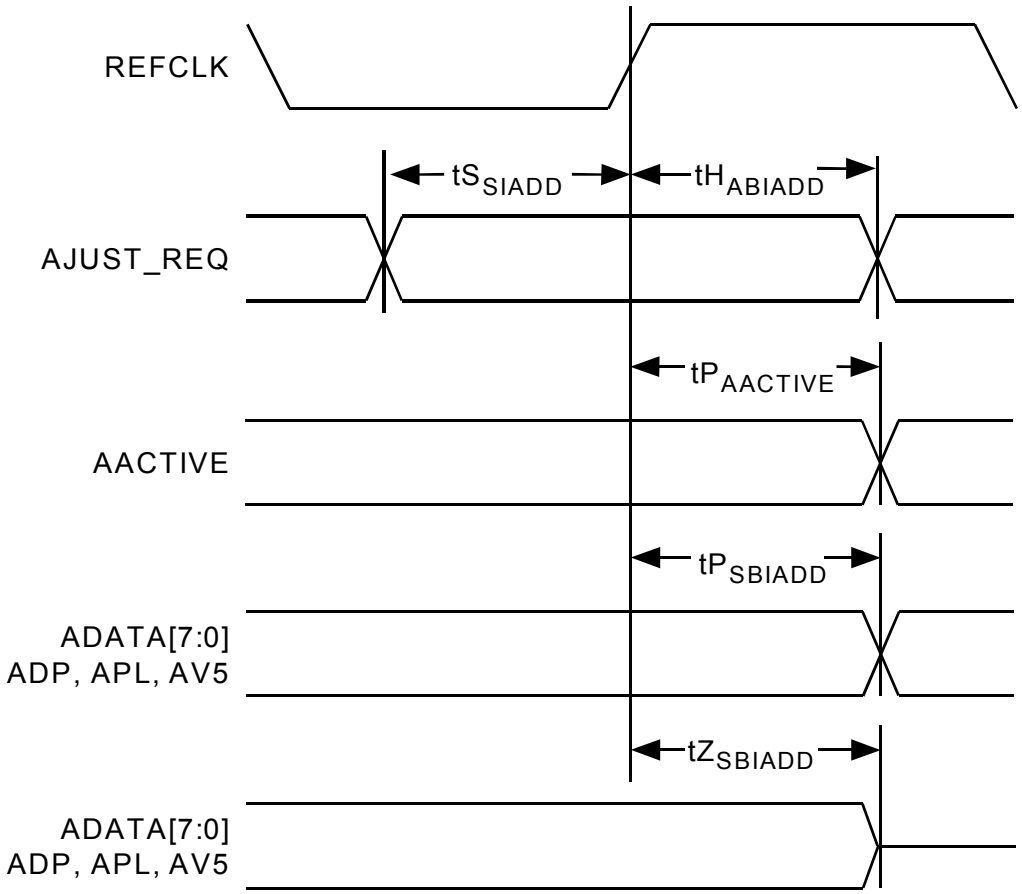


Figure 26 – SBI ADD BUS Collision Avoidance Timing

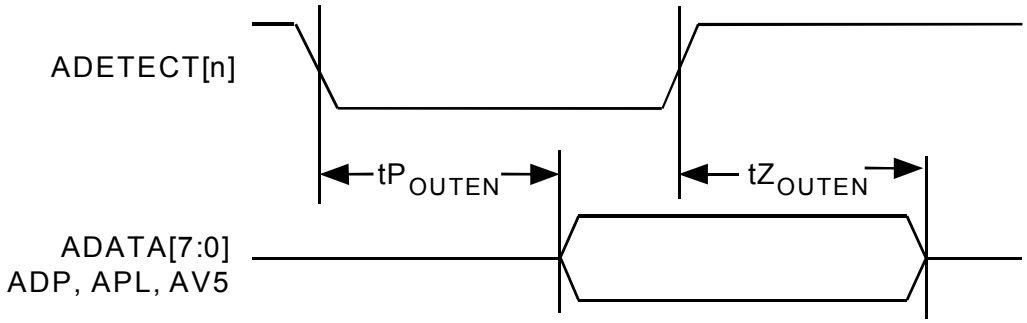


Table 36 – Clock/Data Input (Figure 27)

Symbol	Description	Min	Max	Units
	RCLK[2:0] Frequency		51.84	MHz
	RCLK[2:0] Duty Cycle	40	60	%
tS_{RD}	RD[2:0] Set-Up Time	1		ns
tH_{RD}	RD[2:0] Hold Time	2		ns

Figure 27 – Receive Data Timing

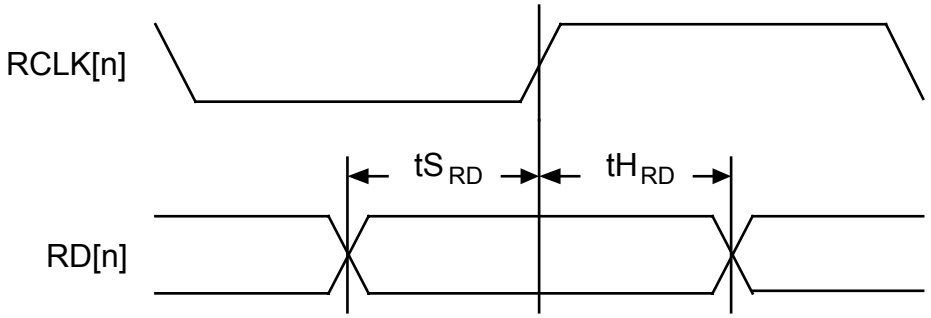


Table 37 – Clock/Data Output (Figure 28)

Symbol	Description	Min	Max	Units
	TCLK[2:0] Frequency		51.84	MHz
	TCLK[2:0] Duty Cycle	40	60	%
tP_{TD}	TCLK[2:0] Low to TD[2:0] Valid	3	12	ns

Figure 28 – Transmit Data Timing

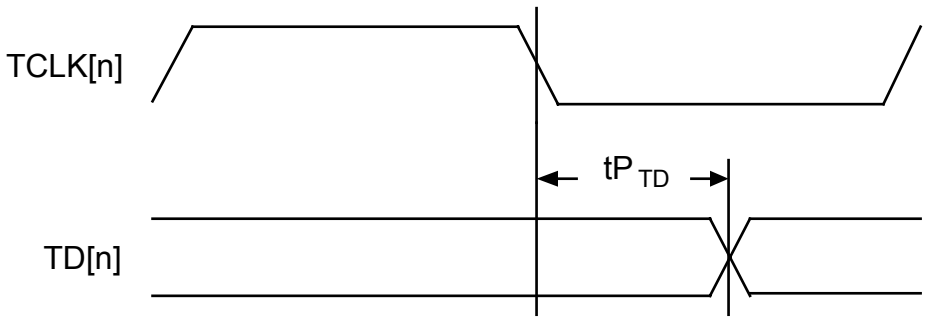


Table 38 – Any-PHY Packet Interface (Figure 29 to Figure 30)

Symbol	Description	Min	Max	Units
	RXCLK Frequency	25	50	MHz
	RXCLK Duty Cycle	40	60	%
	TXCLK Frequency	25	50	MHz
	TXCLK Duty Cycle	40	60	%
tS_{APPI}	All APPI Inputs Set-up time to RXCLK, TXCLK	4		ns
tH_{APPI}	All APPI Inputs Hold time to RXCLK, TXCLK	1		ns
tP_{APPI}	RXCLK, TXCLK to all APPI Outputs Valid	2	12	ns
tZ_{APPI}	RXCLK, TXCLK to APPI Outputs Tristate	2	12	ns
tZD_{APPI}	RXCLK, TXCLK to APPI Outputs Driven	2		ns

Figure 29 – Receive Any-PHY Packet Interface Timing

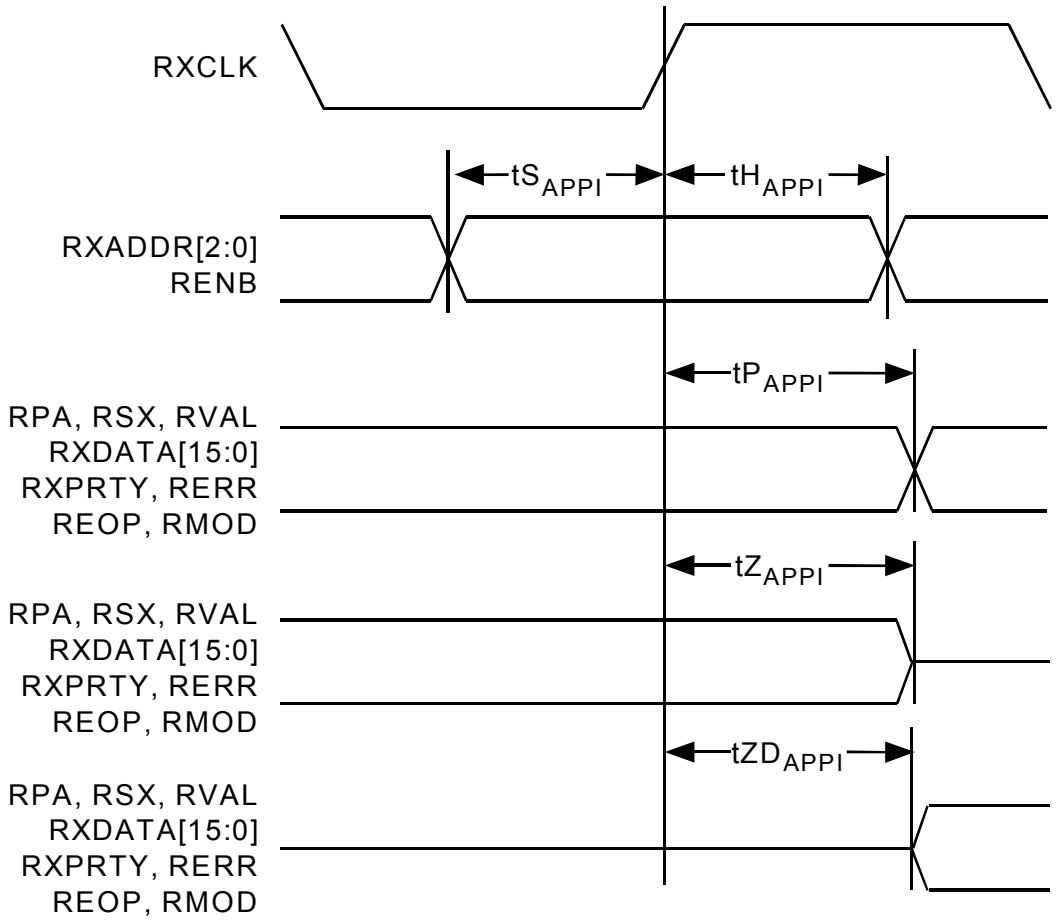


Figure 30 – Transmit Any-PHY Packet Interface Timing

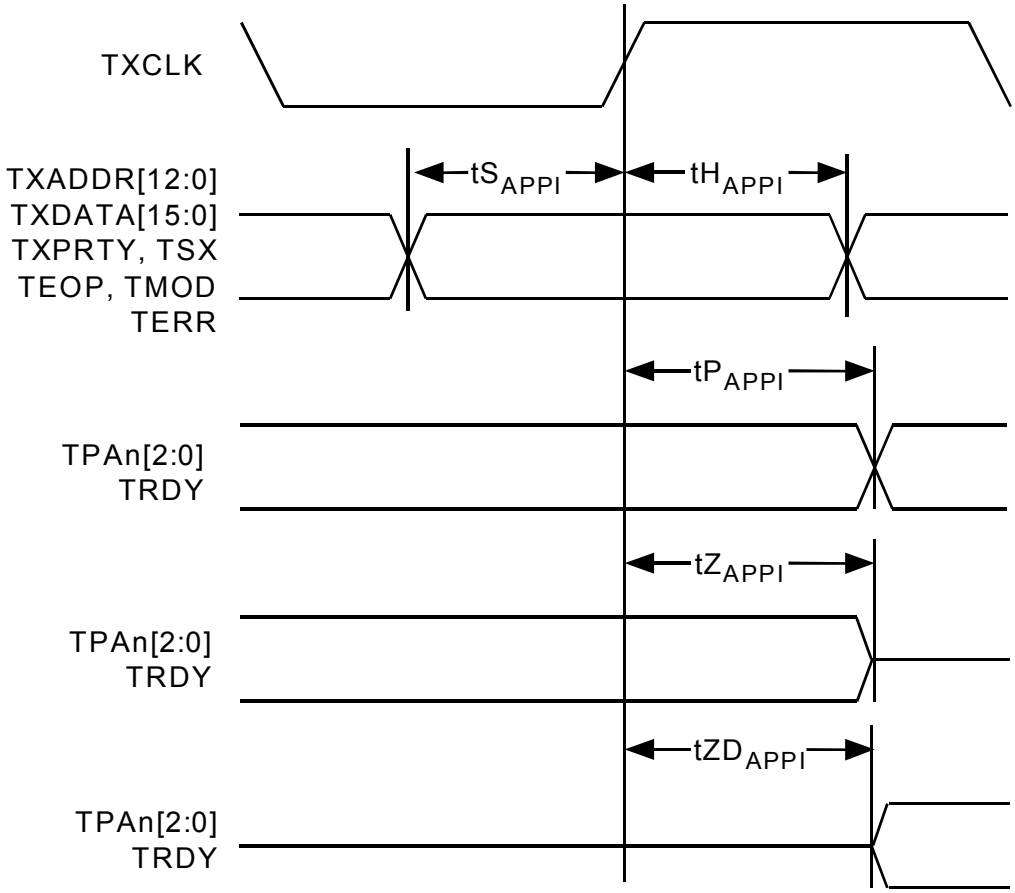
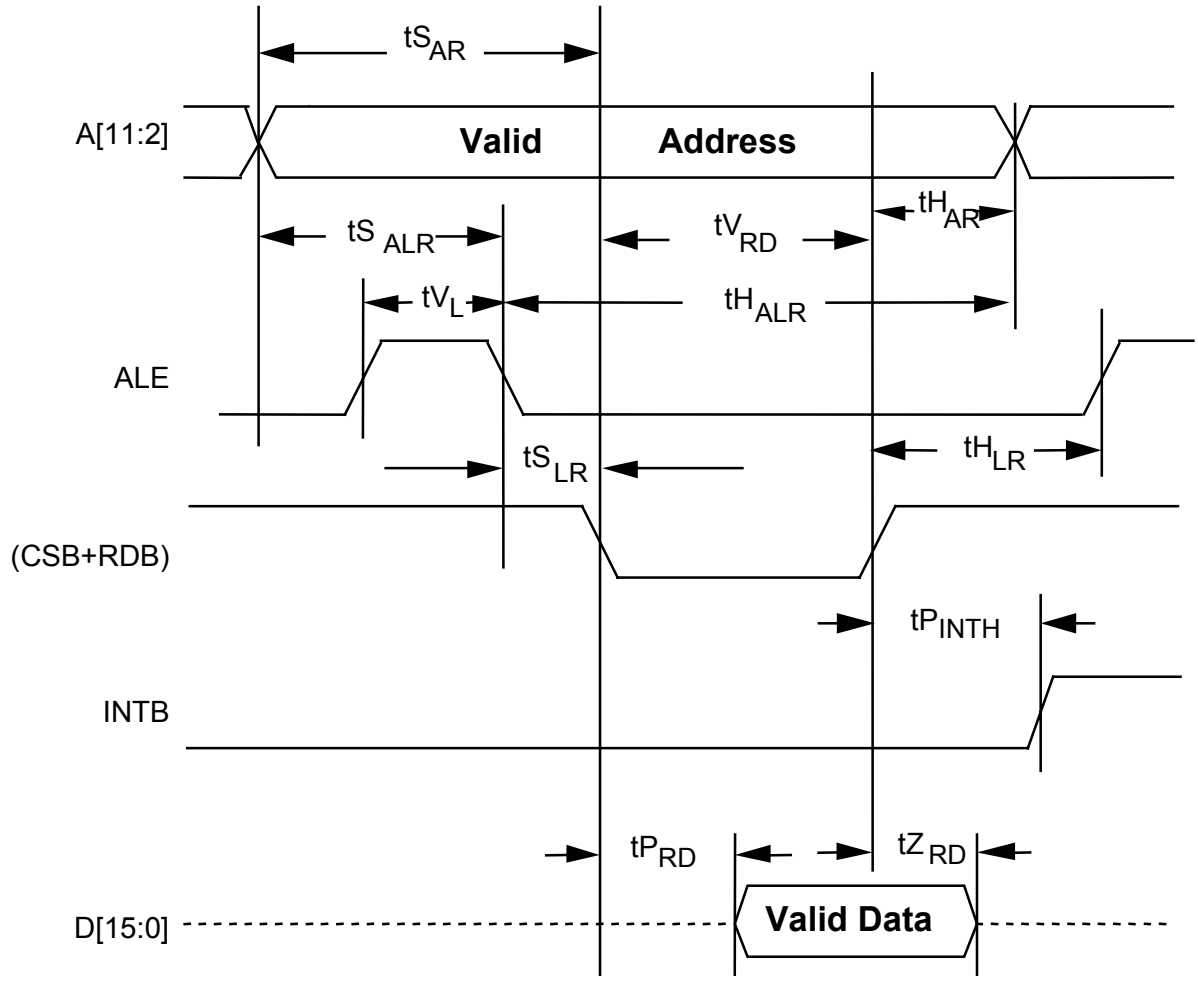


Table 39 – Microprocessor Interface Read Access (Figure 31)

Symbol	Description	Min	Max	Units
tS_{AR}	Address to Valid Read Set-up Time	10		ns
tH_{AR}	Address to Valid Read Hold Time	5		ns
tS_{ALR}	Address to Latch Set-up Time	10		ns
tH_{ALR}	Address to Latch Hold Time	10		ns
tV_{L}	Valid Latch Pulse Width	5		ns
tV_{RD}	Valid Read Pulse Width		100	ns
tS_{LR}	Latch to Read Set-up	0		ns

Symbol	Description	Min	Max	Units
tH_{LR}	Latch to Read Hold	5		ns
tP_{RD}	Valid Read to Valid Data Propagation Delay		40	ns
tZ_{RD}	Valid Read Deasserted to Output Tristate		20	ns
tP_{INTH}	Valid Read Deasserted to INTB High		50	ns

Figure 31 – Microprocessor Read Access Timing



Notes on Microprocessor Read Timing:

1. A valid read cycle is defined as a logical OR of the CSB and the RDB signals.

2. Microprocessor Interface timing applies to normal mode register accesses only.

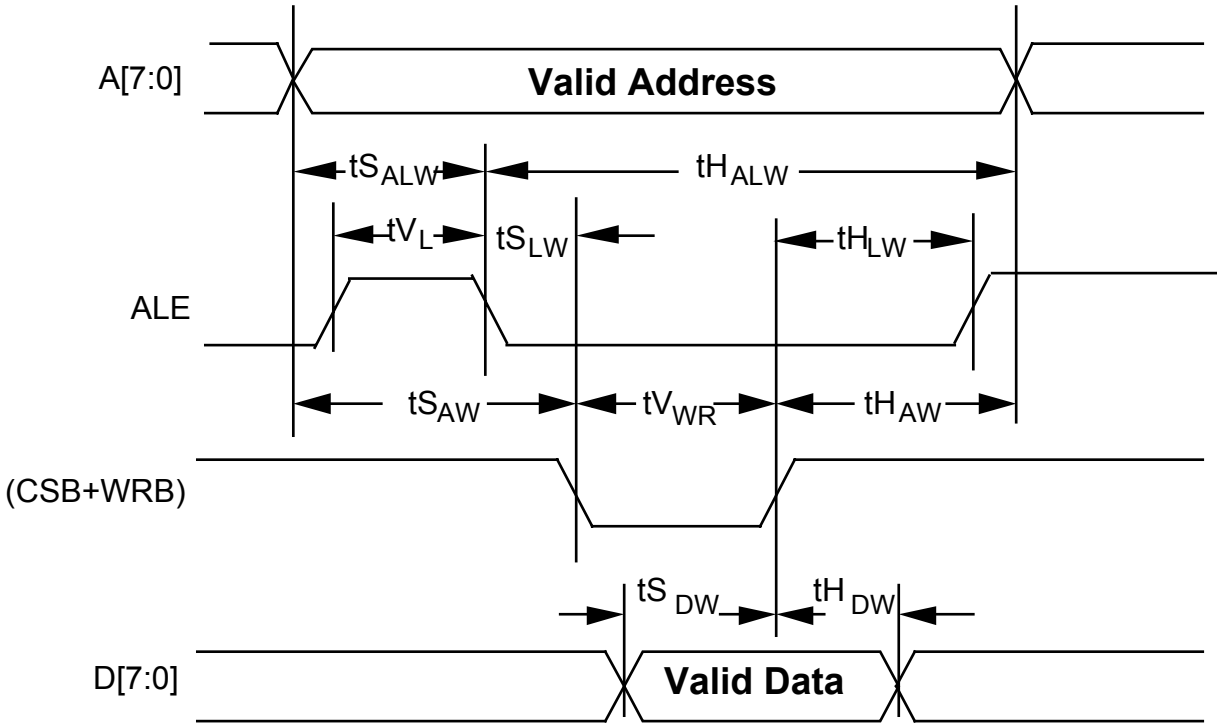
3. In non-multiplexed address/data bus applications, ALE should be held high, parameters tS_{ALR}, tH_{ALR}, tV_L, tS_{LR}, and tH_{LR} are not applicable.

4. Parameter tH_{AR} is not applicable if address latching is used.

Table 40 – Microprocessor Interface Write Access (Figure 32)

Symbol	Description	Min	Max	Units
tS_{AW}	Address to Valid Write Set-up Time	10		ns
tS_{DW}	Data to Valid Write Set-up Time	20		ns
tS_{ALW}	Address to Latch Set-up Time	10		ns
tH_{ALW}	Address to Latch Hold Time	10		ns
tV_L	Valid Latch Pulse Width	5		ns
tS_{LW}	Latch to Write Set-up	0		ns
tH_{LW}	Latch to Write Hold	5		ns
tH_{DW}	Data to Valid Write Hold Time	5		ns
tH_{AW}	Address to Valid Write Hold Time	5		ns
tV_{WR}	Valid Write Pulse Width	20		ns

Figure 32 – Microprocessor Write Access Timing



Notes on Microprocessor Write Timing:

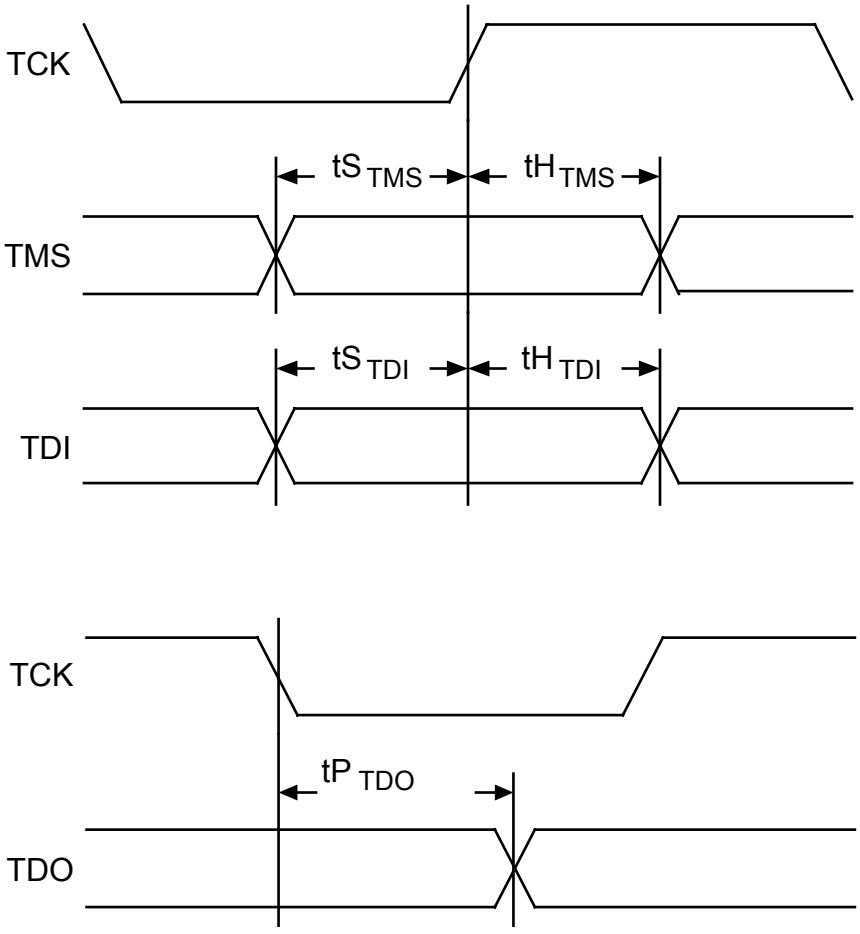
1. A valid write cycle is defined as a logical OR of the CSB and the WRB signals.
2. Microprocessor Interface timing applies to normal mode register accesses only.
3. In non-multiplexed address/data bus architectures, ALE should be held high, parameters tS_{ALW}, tH_{ALW}, tV_L, tS_{LW}, and tH_{LW} are not applicable.
4. Parameters tH_{AW} and tS_{AW} are not applicable if address latching is used.

Table 41 – JTAG Port Interface (Figure 33)

Symbol	Description	Min	Max	Units
	TCK Frequency		1	MHz
	TCK Duty Cycle	40	60	%
tS_{TMS}	TMS Set-up time to TCK	50		ns
tH_{TMS}	TMS Hold time to TCK	50		ns

Symbol	Description	Min	Max	Units
tS_{TDI}	TDI Set-up time to TCK	50		ns
tH_{TDI}	TDI Hold time to TCK	50		ns
tP_{TDO}	TCK Low to TDO Valid	2	60	ns

Figure 33 – JTAG Port Interface Timing



16 ORDERING AND THERMAL INFORMATION

Table 42 – FREEDM-84A672 Ordering Information

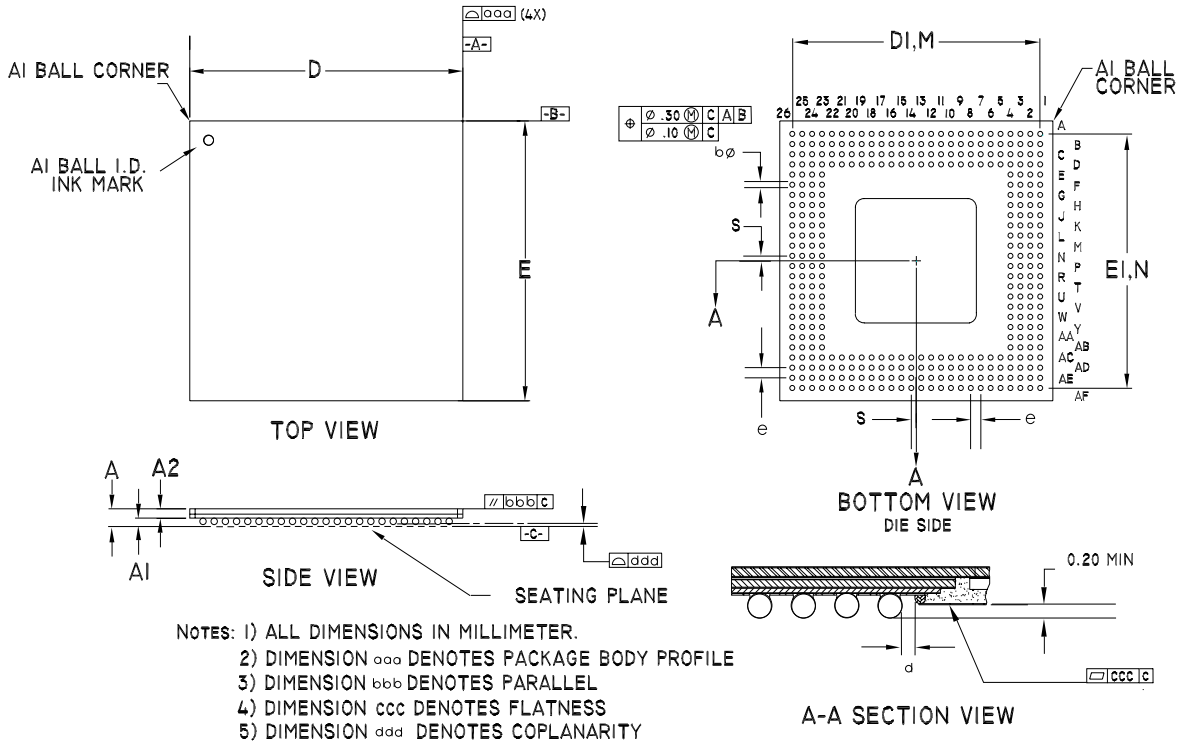
PART NO.	DESCRIPTION
PM7385-BI	352 Enhanced Ball Grid Array (SBGA)

Table 43 – FREEDM-84A672 Thermal Information

PART NO.	CASE TEMPERATURE	Theta Ja	Theta Jc
PM7385-BI	-40°C to +85°C	19 °C/W	< 1 °C/W

17 MECHANICAL INFORMATION

Figure 34 – 352 Pin Enhanced Ball Grid Array (SBGA)



NOTES: 1) ALL DIMENSIONS IN MILLIMETER.
2) DIMENSION aaa DENOTES PACKAGE BODY PROFILE
3) DIMENSION bbb DENOTES PARALLEL
4) DIMENSION ccc DENOTES FLATNESS
5) DIMENSION ddd DENOTES COPLANARITY

PACKAGE TYPE: 352 PIN THERMAL BALL GRID ARRAY																
BODY SIZE: 35 x 35 x 1.51 MM																
DIM.	A	A1	A2	D	D1	E	E1	M,N	e	b	aaa	bbb	ccc	ddd	d	S
MIN.	1.30	0.50	0.80	34.90	31.65	34.90	31.65			0.60					0.50	
NOM.	1.51	0.60	0.91	35.00	31.75	35.00	31.75	26x26	1.27	0.75						0.635
MAX.	1.70	0.70	1.00	35.10	31.85	35.10	31.85			0.90	0.20	0.25	0.20	0.20		

CONTACTING PMC-SIERRA, INC.

PMC-Sierra, Inc.
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
Web Site: http://www.pmc-sierra.com