PM7383

FREEDM™-32A256

FRAME ENGINE AND DATALINK MANAGER 32A256

DATASHEET

PROPRIETARY AND CONFIDENTIAL

RELEASED

ISSUE 2: AUGUST 2001

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 256 bi-directional HDLC channels assigned to a maximum of 32 H-MVIP digital telephony buses at 2.048 Mbps per link. The links are grouped into 4 logical groups of 8 links. A common clock and a type 0 frame pulse is shared among links in each logical group. The number of time-slots assigned to an HDLC channel is programmable from 1 to 32.

- Supports up to 256 bi-directional HDLC channels assigned to a maximum of 8 H-MVIP digital telephony buses at 8.192 Mbps per link. The links share a common clock and a type 0 frame pulse. The number of time-slots assigned to an HDLC channel is programmable from 1 to 128.

- Supports up to 256 bi-directional HDLC channels assigned to a maximum of 32 channelised T1/J1 or E1 links. The number of time-slots assigned to an HDLC channel is programmable from 1 to 24 (for T1/J1) and from 1 to 31 (for E1).

- Supports up to 32 bi-directional HDLC channels each assigned to an unchannelised arbitrary rate link, subject to a maximum aggregate link clock rate of 64 MHz in each direction. Channels assigned to links 0 to 2 support a clock rate of up to 51.84 MHz. Channels assigned to links 3 to 31 support a clock rate of up to 10 MHz.

- Supports three bi-directional HDLC channels each assigned to an unchannelised arbitrary rate link of up to 51.84 MHz when SYSCLK is running at 45 MHz.

- Supports a mix of up to 32 channelised, unchannelised and H-MVIP links, subject to the constraint of a maximum of 256 channels and a maximum aggregate link clock rate of 64 MHz in each direction.

- Links configured for channelised T1/J1/E1 or unchannelised operation support the gapped-clock method for determining time-slots which is backwards compatible with the FREEDM-8 and FREEDM-32 devices.

- 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.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 256 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 5 Volt tolerant I/O (except APPI).

- Low power 2.5 Volt 0.25 μm CMOS technology.

- 329 pin plastic ball grid array (PBGA) package.

2 APPLICATIONS

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

3 REFERENCES

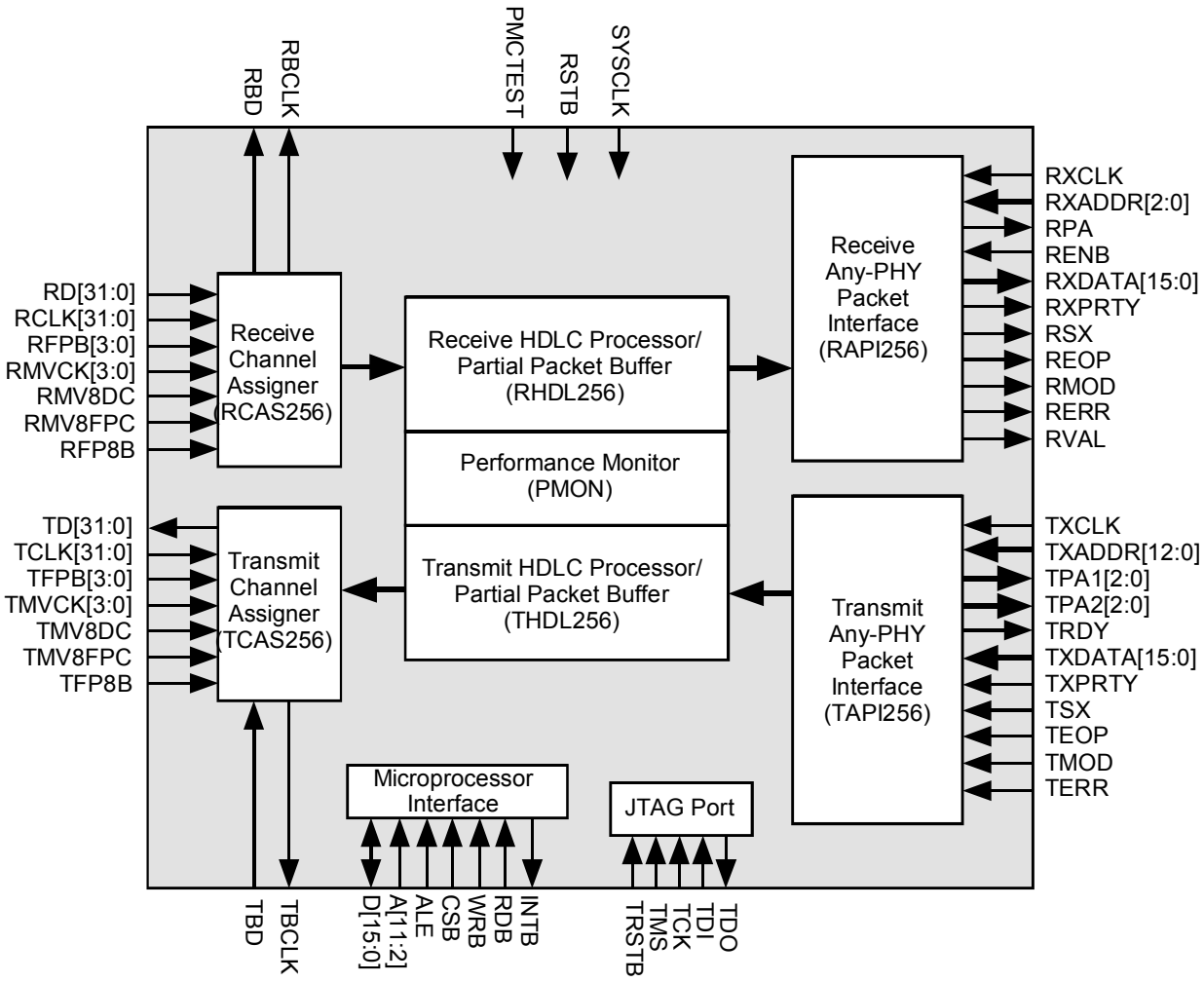
1. International Organization for Standardization, ISO Standard 3309-1993, "Information Technology - Telecommunications and information exchange between systems - High-level data link control (HDLC) procedures - Frame structure", December 1993.

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

3. GO-MVIP, “MVIP-90 Standard”, October 1994, release 1.1.

4. GO-MVIP, “H-MVIP Standard”, January 1997, release 1.1a.

4 BLOCK DIAGRAM



5 DESCRIPTION

The PM7383 FREEDM-32A256 Frame Engine and Datalink Manager device is a monolithic integrated circuit that implements HDLC processing for a maximum of 256 bi-directional channels.

The FREEDM-32A256 may be configured to support H-MVIP, channelised T1/J1/E1 or unchannelised traffic across 32 physical links.

The FREEDM-32A256 may be configured to interface with H-MVIP digital telephony buses at 2.048 Mbps. For 2.048 Mbps H-MVIP links, the FREEDM-32A256 allows up to 256 bi-directional HDLC channels to be assigned to individual time-slots within a maximum of 32 H-MVIP links. The channel assignment supports the concatenation of time-slots (N x DS0) up to a maximum of 32 concatenated time-slots for each 2.048 Mbps H-MVIP link. Time-slots assigned to any particular channel need not be contiguous within the H-MVIP link. When configured for 2.048 Mbps H-MVIP operation, the FREEDM-32A256 partitions the 32 physical links into 4 logical groups of 8 links. Links 0 through 7, 8 through 15, 16 through 23 and 24 through 31 make up the 4 logical groups. Links in each logical group share a common clock and a common type 0 frame pulse in each direction.

The FREEDM-32A256 may be configured to interface with H-MVIP digital telephony buses at 8.192 Mbps. For 8.192 Mbps H-MVIP links, the FREEDM-32A256 allows up to 256 bi-directional HDLC channels to be assigned to individual time-slots within a maximum of 8 H-MVIP links. The channel assignment supports the concatenation of time-slots (N x DS0) up to a maximum of 128 concatenated time-slots for each 8.192 H-MVIP link. Time-slots assigned to any particular channel need not be contiguous within the H-MVIP link. When configured for 8.192 Mbps H-MVIP operation, the FREEDM-32A256 partitions the 32 physical links into 8 logical groups of 4 links. Only the first link, which must be located at physical links numbered 4m ($0 \leq m \leq 7$), of each logical group can be configured for 8.192 Mbps operation. The remaining 3 physical links in the logical group (numbered 4m+1, 4m+2 and 4m+3) are unused. All links configured for 8.192 Mbps H-MVIP operation will share a common type 0 frame pulse, a common frame pulse clock and a common data clock.

For channelised T1/J1/E1 links, the FREEDM-32A256 allows up to 256 bi-directional HDLC channels to be assigned to individual time-slots within a maximum of 32 independently timed T1/J1 or E1 links. The gapped clock method to determine time-slot positions as per the FREEDM-8 and FREEDM-32 devices is retained. 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 the T1/J1 or E1 link.

For unchannelised links, the FREEDM-32A256 processes up to 32 bi-directional HDLC channels within 32 independently timed links. The links can be of arbitrary frame format. When limited to three unchannelised links, each link can be rated at up to 51.84 MHz provided SYSCLK is running at 45 MHz. For lower rate unchannelised links, the FREEDM-32A256 processes up to 32 links each rated at up to 10 MHz. In this case, the aggregate clock rate of all the links is limited to 64 MHz.

The FREEDM-32A256 supports mixing of up to 32 channelised T1/J1/E1, unchannelised and H-MVIP links. The total number of channels in each direction is limited to 256. The aggregate instantaneous clock rate over all 32 possible links is limited to 64 MHz.

The FREEDM-32A256 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-32A256 devices may share a single APPI. For each of the transmit and receive APPI, the external controller is the master of each FREEDM-32A256 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-32A256 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-32A256 device, or a channel within a FREEDM-32A256 device, for data transfer. The data transfer signals provide a means of transferring data across the APPI between the external controller and a FREEDM-32A256 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-32A256 device. In response to a positive poll, the external controller may select that FREEDM-32A256 device for data transfer. Once selected, the FREEDM-32A256 prepends an in-band channel address to each partial packet transfer across the receive APPI to associate the data with a channel. A FREEDM-32A256 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-32A256 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-32A256 performs channel assignment and packet extraction and validation. For each provisioned HDLC channel, the FREEDM-32A256 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-32A256 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-32A256 reports the status of each packet on the receive APPI at the end of each packet transfer.

Alternatively, in the receive direction, the FREEDM-32A256 supports a transparent operating mode. For each provisioned transparent channel, the FREEDM-32A256 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-32A256 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-32A256 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-32A256 will not attempt to re-transmit aborted packets.

Alternatively, in the transmit direction, the FREEDM-32A256 supports a transparent operating mode. For each provisioned transparent channel, the FREEDM-32A256 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-32A256 receives new data on the transmit APPI.

The FREEDM-32A256 is configured, controlled and monitored using the microprocessor interface. The FREEDM-32A256 is implemented in low power2.5 Volt 0.25 μm CMOS technology. All FREEDM-32A256 I/O except those belonging to the APPI are 5 volt tolerant. The APPI I/O are 3.3 volt tolerant. The FREEDM-32A256 is packaged in a 329 pin plastic ball grid array (PBGA) package.

6 PIN DIAGRAM

The FREEDM-32A256 is manufactured in a 329 pin plastic ball grid array package.

	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
A	RMVCK[2]	RD[16]	RCLK[17]	RCLK[19]	RD[21]	RD[22]	RD[23]	RD[24]	RCLK[25]	RCLK[27]	RCLK[29]	VDD2V5	RDB	A[10]	A[6]	A[3]	D[13]	D[11]	D[9]	D[6]	D[2]	D[0]	TPA2[2]	A
B	RFPB[2]	RCLK[16]	RD[18]	RD[20]	VDD2V5	RCLK[22]	RFPB[3]	RCLK[24]	RCLK[26]	RD[28]	RD[30]	RCLK[31]	INTB	A[11]	A[8]	A[4]	D[15]	D[12]	VDD2V5	D[7]	D[3]	TPA2[0]	TPA2[1]	B
C	RD[15]	RSTB	RCLK[15]	RCLK[18]	RCLK[20]	RCLK[21]	RMVCK[3]	RD[25]	RD[27]	RCLK[28]	RCLK[30]	RD[31]	CSB	ALE	A[9]	A[5]	A[2]	D[10]	D[8]	D[4]	D[1]	N.C.	TPA1[1]	C
D	RCLK[13]	RD[13]	RCLK[14]	RD[17]	RD[19]	VDD3V3	RCLK[23]	VSS	RD[26]	VDD3V3	RD[29]	VSS	WRB	VDD3V3	A[7]	VSS	D[14]	VDD3V3	D[5]	TPA1[2]	TPA1[0]	TXADDR[10]	TXADDR[11]	D
E	RD[12]	VDD2V5	RCLK[12]	RD[14]																TXADDR[12]	TXADDR[9]	VDD2V5	TXADDR[8]	E
F	RD[11]	RCLK[10]	RCLK[11]	VSS																VSS	TXADDR[7]	TXADDR[5]	TXADDR[6]	F
G	RD[10]	RD[9]	RCLK[8]	RCLK[9]							BOTTOM VIEW									TXADDR[3]	TXADDR[1]	TXADDR[2]	TXADDR[4]	G
H	RD[8]	RMVCK[1]	RFPB[1]	VDD3V3																VDD3V3	TXDATA[15]	TXCLK	TXADDR[0]	H
J	RCLK[7]	RCLK[6]	RD[6]	RD[7]																TXDATA[13]	TXDATA[11]	TXDATA[12]	TXDATA[14]	J
K	SYSCLK	RCLK[5]	RD[5]	VSS						VSS	VSS	VSS	VSS	VSS						VSS	TXDATA[8]	TXDATA[9]	TXDATA[10]	K
L	RD[4]	RD[3]	RCLK[3]	RCLK[4]						VSS	VSS	VSS	VSS	VSS						TXPRTY	TXDATA[6]	TXDATA[7]	TSX	L
M	VDD2V5	RCLK[2]	RD[2]	VDD3V3						VSS	VSS	VSS	VSS	VSS						VDD3V3	TXDATA[5]	TXDATA[4]	VDD2V5	M
N	RCLK[0]	RD[1]	RCLK[1]	RD[0]						VSS	VSS	VSS	VSS	VSS						TXDATA[0]	TXDATA[2]	TXDATA[3]	TXDATA[1]	N
P	RMV8FPC	RFPB[0]	RMVCK[0]	VSS						VSS	VSS	VSS	VSS	VSS						VSS	TEOP	TMOD	TERR	P
R	RBD	RMV8DC	RFP8B	RBCLK																RPA	TRDY	RVAL	RENB	R
T	TCK	TMS	TRSTB	VDD3V3																VDD3V3	REOP	RMOD	RERR	T
U	TFP8B	TDO	TDI	TMV8DC																RXDATA[14]	RXPRTY	RXDATA[15]	RXDATA[13]	U
V	TFPB[0]	TMV8FPC	TMVCK[0]	VSS																VSS	RXDATA[10]	RXDATA[12]	RXDATA[11]	V
W	TD[0]	VDD2V5	TCLK[0]	TD[2]																RXDATA[6]	RXDATA[8]	VDD2V5	RXDATA[9]	W
Y	TCLK[1]	TD[1]	TCLK[2]	TD[4]	TMVCK[1]	VDD3V3	TD[12]	VSS	TCLK[15]	VDD3V3	TCLK[17]	VSS	TD[22]	VDD3V3	TD[24]	VSS	TCLK[27]	VDD3V3	TBD	RXADDR[1]	RXDATA[5]	RSX	RXDATA[7]	Y
AA	TCLK[3]	TD[3]	TCLK[6]	TFPB[1]	TD[9]	TD[10]	TD[13]	TCLK[14]	TMVCK[2]	TD[17]	TCLK[18]	TD[20]	TCLK[20]	TCLK[22]	TFPB[3]	TD[25]	TCLK[26]	TCLK[29]	TCLK[30]	TBCLK	RXDATA[2]	RXDATA[4]	RXDATA[3]	AA
AB	TD[5]	TCLK[4]	TCLK[7]	TCLK[8]	VDD2V5	TD[11]	TCLK[12]	TD[14]	TFPB[2]	TCLK[16]	TD[19]	TCLK[19]	TD[21]	TD[23]	TMVCK[3]	TCLK[25]	TD[27]	TCLK[28]	VDD2V5	TD[31]	PMCTEST	RXADDR[2]	RXDATA[1]	AB
AC	TCLK[5]	TD[6]	TD[7]	TD[8]	TCLK[9]	TCLK[10]	TCLK[11]	TCLK[13]	TD[15]	TD[16]	TD[18]	VDD2V5	TCLK[21]	TCLK[23]	TCLK[24]	TD[26]	TD[28]	TD[29]	TD[30]	TCLK[31]	RXADDR[0]	RXCLK	RXDATA[0]	AC
	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

7 PIN DESCRIPTION

Table 1 – Line Side Interface Signals (154)

Pin Name	Type	Pin No.	Function
RCLK[0] RCLK[1] RCLK[2] RCLK[3] RCLK[4] RCLK[5] RCLK[6] RCLK[7] RCLK[8] RCLK[9] RCLK[10] RCLK[11] RCLK[12] RCLK[13] RCLK[14] RCLK[15] RCLK[16] RCLK[17] RCLK[18] RCLK[19] RCLK[20] RCLK[21] RCLK[22] RCLK[23] RCLK[24] RCLK[25] RCLK[26] RCLK[27] RCLK[28] RCLK[29] RCLK[30] RCLK[31]	Input	N23 N21 M22 L21 L20 K22 J22 J23 G21 G20 F22 F21 E21 D23 D21 C21 B22 A21 C20 A20 C19 C18 B18 D17 B16 A15 B15 A14 C14 A13 C13 B12	The receive line clock signals (RCLK[31:0]) contain the recovered line clock for the 32 independently timed links. Processing of the receive links are on a priority basis, in descending order from RCLK[0] to RCLK[31]. Therefore, the highest rate link should be connected to RCLK[0] and the lowest to RCLK[31]. For channelised T1/J1 or E1 links, RCLK[n] must be gapped during the framing bit (for T1/J1 interfaces) or during time-slot 0 (for E1 interfaces) of the RD[n] stream. The FREEDM-32A256 uses the gapping information to determine the time-slot alignment in the receive stream. RCLK[31:0] is nominally a 50% duty cycle clock of frequency 1.544 MHz for T1/J1 links and 2.048 MHz for E1 links. For unchannelised 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. RCLK[31:3] is nominally a 50% duty cycle clock between 0 and 10 MHz. The RCLK[n] inputs are invalid and should be forced to a low state when their associated link is configured for operation in H-MVIP mode.

Pin Name	Type	Pin No.	Function
RD[0] RD[1] RD[2] RD[3] RD[4] RD[5] RD[6] RD[7] RD[8] RD[9] RD[10] RD[11] RD[12] RD[13] RD[14] RD[15] RD[16] RD[17] RD[18] RD[19] RD[20] RD[21] RD[22] RD[23] RD[24] RD[25] RD[26] RD[27] RD[28] RD[29] RD[30] RD[31]	Input	N20 N22 M21 L22 L23 K21 J21 J20 H23 G22 G23 F23 E23 D22 E20 C23 A22 D20 B21 D19 B20 A19 A18 A17 A16 C16 D15 C15 B14 D13 B13 C12	The receive data signals (RD[31:0]) contain the recovered line data for the 32 independently timed links in normal mode (PMCTEST set low). Processing of the receive links is on a priority basis, in descending order from RD[0] to RD[31]. Therefore, the highest rate link should be connected to RD[0] and the lowest to RD[31]. For H-MVIP links, RD[n] contains 32/128 time-slots, depending on the H-MVIP data rate configured (2.048 or 8.192 Mbps). When configured for 2.048 Mbps H-MVIP operation, RD[31:24], RD[23:16], RD[15:8] and RD[7:0] are sampled on every 2nd rising edge of RMVCK[3], RMVCK[2], RMVCK[1] and RMVCK[0] respectively (at the ¾ point of the bit interval). When configured for 8.192 Mbps H-MVIP operation, RD[4m] ($0 \leq m \leq 7$) are sampled on every 2nd rising edge of RMV8DC (at the ¾ point of the bit interval). For channelised links, RD[n] contains the 24 (T1/J1) or 31 (E1) time-slots that comprise the channelised link. RCLK[n] must be gapped during the T1/J1 framing bit position or the E1 frame alignment signal (time-slot 0). The FREEDM-32A256 uses the location of the gap to determine the channel alignment on RD[n]. RD[31:0] are sampled on the rising edge of the corresponding RCLK[31:0]. For unchannelised links, RD[n] contains the HDLC packet data. For certain transmission formats, RD[n] 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-32A256 supports a maximum data rate of 10 Mbit/s

Pin Name	Type	Pin No.	Function
			on an individual RD[31:3] link and a maximum data rate of 51.84 Mbit/s on RD[2:0]. RD[31:0] are sampled on the rising edge of the corresponding RCLK[31:0].
RMVCK[0] RMVCK[1] RMVCK[2] RMVCK[3]	Input	P21 H22 A23 C17	The receive MVIP data clock signals (RMVCK[3:0]) provide the receive data clock for the 32 links when configured to operate in 2.048 Mbps H-MVIP mode. When configured for 2.048 Mbps H-MVIP operation, the 32 links are partitioned into 4 groups of 8, and each group of 8 links share a common data clock. RMVCK[0], RMVCK[1], RMVCK[2] and RMVCK[3] sample the data on links RD[7:0], RD[15:8], RD[23:16] and RD[31:24] respectively. Each RMVCK[n] is nominally a 50% duty cycle clock with a frequency of 4.096 MHz. RMVCK[n] is ignored and should be tied low when no physical link within the associated logical group of 8 links is configured for operation in 2.048 Mbps H-MVIP mode.

Pin Name	Type	Pin No.	Function
RFPB[0] RFPB[1] RFPB[2] RFPB[3]	Input	P22 H21 B23 B17	The receive frame pulse signals (RFPB[3:0]) reference the beginning of each frame for the 32 links when configured for operation in 2.048 Mbps H-MVIP mode. When configured for 2.048 Mbps H-MVIP operation, the 32 links are partitioned into 4 groups of 8, and each group of 8 links share a common frame pulse. RFPB[0], RFPB[1], RFPB[2] and RFPB[3] reference the beginning of a frame on links RD[7:0], RD[15:8], RD[23:16] and RD[31:24] respectively. When configured for operation in 2.048 Mbps H-MVIP mode, RFPB[n] is sampled on the falling edge of RMVCK[n]. Otherwise, RFPB[n] is ignored and should be tied low.
RFP8B	Input	R21	The receive frame pulse for 8.192 Mbps H-MVIP signal (RFP8B) references the beginning of each frame for links configured for operation in 8.192 Mbps H-MVIP mode. RFP8B references the beginning of a frame for any link configured for 8.192 Mbps H-MVIP operation. Only links RD[4m] ($0 \leq m \leq 7$) may be configured for 8.192 Mbps H-MVIP operation. When one or more links are configured for 8.192 Mbps H-MVIP operation, RFP8B is sampled on the falling edge of RMV8FPC. When no links are configured for 8.192 Mbps H-MVIP operation, RFP8B is ignored and should be tied low.

Pin Name	Type	Pin No.	Function
RMV8FPC	Input	P23	The receive 8.192 Mbps H-MVIP frame pulse clock signal (RMV8FPC) provides the receive frame pulse clock for links configured for operation in 8.192 Mbps H-MVIP mode. RMV8FPC is used to sample RFP8B. RMV8FPC is nominally a 50% duty cycle, clock with a frequency of 4.096 MHz. The falling edge of RMV8FPC must be aligned with the falling edge of RMV8DC with no more than ±10 ns skew. RMV8FPC is ignored and should be tied low when no physical links are configured for operation in 8.192 Mbps H-MVIP mode.
RMV8DC	Input	R22	The receive 8.192 Mbps H-MVIP data clock signal (RMV8DC) provides the receive data clock for links configured to operate in 8.192 Mbps H-MVIP mode. RMV8DC is used to sample data on RD[4m] ($0 \leq m \leq 7$) when link 4m is configured for 8.192 Mbps H-MVIP operation. RMV8DC is nominally a 50% duty cycle clock with a frequency of 16.384 MHz. RMV8DC is ignored and should be tied low when no physical links are configured for operation in 8.192 Mbps H-MVIP mode.
RBD	Tristate Output	R23	The receive BERT data signal (RBD) contains the receive bit error rate test data. RBD reports the data on the selected one of the receive data signals (RD[31:0]) and is updated on the falling edge of RBCLK. RBD may be tristated by setting the RBEN bit in the FREEDM-32A256 Master BERT Control register low. BERT is not supported for H-MVIP links.

Pin Name	Type	Pin No.	Function
RBCLK	Tristate Output	R20	The receive BERT clock signal (RBCLK) contains the receive bit error rate test clock. RBCLK is a buffered version of the selected one of the receive clock signals (RCLK[31:0]). RBCLK may be tristated by setting the RBEN bit in the FREEDM-32A256 Master BERT Control register low. BERT is not supported for H-MVIP links.

Pin Name	Type	Pin No.	Function
TCLK[0] TCLK[1] TCLK[2] TCLK[3] TCLK[4] TCLK[5] TCLK[6] TCLK[7] TCLK[8] TCLK[9] TCLK[10] TCLK[11] TCLK[12] TCLK[13] TCLK[14] TCLK[15] TCLK[16] TCLK[17] TCLK[18] TCLK[19] TCLK[20] TCLK[21] TCLK[22] TCLK[23] TCLK[24] TCLK[25] TCLK[26] TCLK[27] TCLK[28] TCLK[29] TCLK[30] TCLK[31]	Input	W21 Y23 Y21 AA23 AB22 AC23 AA21 AB21 AB20 AC19 AC18 AC17 AB17 AC16 AA16 Y15 AB14 Y13 AA13 AB12 AA11 AC11 AA10 AC10 AC9 AB8 AA7 Y7 AB6 AA6 AA5 AC4	The transmit line clock signals (TCLK[31:0]) contain the transmit clocks for the 32 independently timed links. Processing of the transmit links is on a priority basis, in descending order from TCLK[0] to TCLK[31]. Therefore, the highest rate link should be connected to TCLK[0] and the lowest to TCLK[31]. For channelised T1/J1 or E1 links, TCLK[n] must be gapped during the framing bit (for T1/J1 interfaces) or during time-slot 0 (for E1 interfaces) of the TD[n] stream. The FREEDM-32A256 uses the gapping information to determine the time-slot alignment in the transmit stream. For unchannelised 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. TCLK[31:3] is nominally a 50% duty cycle clock between 0 and 10 MHz. Typical values for TCLK[31:0] include 1.544 MHz (for T1/J1 links) and 2.048 MHz (for E1 links). The TCLK[n] inputs are invalid and should be tied low when their associated link is configured for operation in H-MVIP mode.

Pin Name	Type	Pin No.	Function
TD[0] TD[1] TD[2] TD[3] TD[4] TD[5] TD[6] TD[7] TD[8] TD[9] TD[10] TD[11] TD[12] TD[13] TD[14] TD[15] TD[16] TD[17] TD[18] TD[19] TD[20] TD[21] TD[22] TD[23] TD[24] TD[25] TD[26] TD[27] TD[28] TD[29] TD[30] TD[31]	Output	W23 Y22 W20 AA22 Y20 AB23 AC22 AC21 AC20 AA19 AA18 AB18 Y17 AA17 AB16 AC15 AC14 AA14 AC13 AB13 AA12 AB11 Y11 AB10 Y9 AA8 AC8 AB7 AC7 AC6 AC5 AB4	The transmit data signals (TD[31:0]) contain the transmit data for the 32 independently timed links in normal mode (PMCTEST set low). Processing of the transmit links is on a priority basis, in descending order from TD[0] to TD[31]. Therefore, the highest rate link should be connected to TD[0] and the lowest to TD[31]. For H-MVIP links, TD[n] contain 32/128 time-slots, depending on the H-MVIP data rate configured (2.048 or 8.192 Mbps). When configured for 2.048 Mbps H-MVIP operation, TD[31:24], TD[23:16], TD[15:8] and TD[7:0] are updated on every 2nd falling edge of TMVCK[3], TMVCK[2], TMVCK[1] and TMVCK[0] respectively. When configured for 8.192 Mbps H-MVIP operation, TD[4m] ($0 \leq m \leq 7$) are updated on every 2nd falling edge of TMV8DC. For channelised links, TD[n] contains the 24 (T1/J1) or 31 (E1) time-slots that comprise the channelised link. TCLK[n] must be gapped during the T1/J1 framing bit position or during the E1 frame alignment signal (time-slot 0). The FREEDM-32A256 uses the location of the gap to determine the channel alignment on TD[n]. TD[31:0] are updated on the falling edge of the corresponding TCLK[31:0]. For unchannelised links, TD[n] contains the HDLC packet data. For certain transmission formats, TD[n] 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-32A256 supports a maximum data rate of 10 Mbit/s on an individual TD[31:3] link and a maximum data rate of 51.84 Mbit/s on TD[2:0].

Pin Name	Type	Pin No.	Function
			TD[31:0] are updated on the falling edge of the corresponding TCLK[31:0] clock.
TMVCK[0] TMVCK[1] TMVCK[2] TMVCK[3]	Input	V21 Y19 AA15 AB9	The transmit MVIP data clock signals (TMVCK[3:0]) provide the transmit data clocks for the 32 links when configured to operate in 2.048 Mbps H-MVIP mode. When configured for 2.048 Mbps H-MVIP operation, the 32 links are partitioned into 4 groups of 8, and each group of 8 links share a common clock. TMVCK[0], TMVCK[1], TMVCK[2] and TMVCK[3] update the data on links TD[7:0], TD[15:8], TD[23:16] and TD[31:24] respectively. Each TMVCK[n] is nominally a 50% duty cycle clock with a frequency of 4.096 MHz. TMVCK[n] is unused and should be tied low when no physical links within the associated group of 8 logical links is configured for operation in 2.048 Mbps H-MVIP mode.
TFPB[0] TFPB[1] TFPB[2] TFPB[3]	Input	V23 AA20 AB15 AA9	The transmit frame pulse signals (TFPB[3:0]) reference the beginning of each frame when configured for operation in 2.048 Mbps H-MVIP mode. When configured for 2.048 Mbps H-MVIP operation, the 32 links are partitioned into 4 groups of 8, and each group of 8 links share a common frame pulse. TFPB[0], TFPB[1], TFPB[2] and TFPB[3] reference the beginning of a frame on links TD[7:0], TD[15:8], TD[23:16] and TD[31:24] respectively. When configured for operation in 2.048 Mbps H-MVIP mode, TFPB[n] is sampled on the falling edge of TMVCK[n]. Otherwise, TFPB[n] is ignored and should be tied low.

Pin Name	Type	Pin No.	Function
TFP8B	Input	U23	The transmit frame pulse for 8.192 Mbps H-MVIP signal (TFP8B) references the beginning of each frame for links configured to operate in 8.192 Mbps H-MVIP mode. TFP8B references the beginning of a frame for any link configured for 8.192 Mbps H-MVIP operation. Only links 4m ($0 \leq m \leq 7$) may be configured for 8.192 Mbps H-MVIP operation. When one or more links are configured for 8.192 Mbps H-MVIP operation, TFP8B is sampled on the falling edge of TMV8FPC. When no links are configured for 8.192 Mbps H-MVIP operation, TFPB[n] is ignored and should be tied low.
TMV8FPC	Input	V22	The transmit 8.192 Mbps H-MVIP frame pulse clock signal (TMV8FPC) provides the transmit frame pulse clock for links configured for operation in 8.192 Mbps H-MVIP mode. TMV8FPC is used to sample TFP8B. TMV8FPC is nominally a 50% duty cycle, clock with a frequency of 4.096 MHz. The falling edge of TMV8FPC must be aligned with the falling edge of TMV8DC with no more than ±10 ns skew. TMV8FPC[n] is ignored and should be tied low when no physical links are configured for operation in 8.192 Mbps H-MVIP mode.

Pin Name	Type	Pin No.	Function
TMV8DC	Input	U20	The transmit 8.192 Mbps H-MVIP data clock signal (TMV8DC) provides the transmit data clock for links configured to operate in 8.192 Mbps H-MVIP mode. TMV8DC is used to update data on TD[4m] ($0 \leq m \leq 7$) when link 4m is configured for 8.192 Mbps H-MVIP operation. TMV8DC is nominally a 50% duty cycle clock with a frequency of 16.384 MHz. TMV8DC is unused and should be tied low when no physical links are configured for operation in 8.192 Mbps H-MVIP mode.
TBD	Input	Y5	The transmit BERT data signal (TBD) contains the transmit bit error rate test data. When the TBERTEN bit in the BERT Control register is set high, the data on TBD is transmitted on the selected one of the transmit data signals (TD[31:0]). TBD is sampled on the rising edge of TBCLK. BERT is not supported for H-MVIP links.
TBCLK	Tristate Output	AA4	The transmit BERT clock signal (TBCLK) contains the transmit bit error rate test clock. TBCLK is a buffered version of the selected one of the transmit clock signals (TCLK[31:0]). TBCLK may be tristated by setting the TBEN bit in the FREEDM-32A256 Master BERT Control register low. BERT is not supported for H-MVIP links.

Table 2 – Any-PHY Packet Interface Signals (70)

Pin Name	Type	Pin No.	Function
TXCLK	Input	H2	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[10] TXADDR[11] TXADDR[12]	Input	H1 G3 G2 G4 G1 F2 F1 F3 D2 D1 E4	The transmit address signals (TXADDR[12:0]) provide a channel address for polling a transmit channel FIFO. The 8 least significant bits provide the channel number (0 to 255) while the 3 most significant bits select one of seven possible FREEDM-32A256 devices sharing a single external controller. (One address is reserved as a null address.) The Tx APPI of each FREEDM-32A256 device is identified by the base address in the TAPI256 Control register. The TXADDR[12:0] signals are sampled on the rising edge of TXCLK. Note that TXADDR[9:8] have been removed from the FREEDM-32A256 device. Pin numbering of TXADDR[7:0] and TXADDR[12:10] has been maintained to allow software compatibility with the FREEDM-32A672 device.

Pin Name	Type	Pin No.	Function
TPA1[0] TPA1[1] TPA1[2] TPA2[0] TPA2[1] TPA2[2]	Tristate Output	D3 C1 D4 B2 B1 A1	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-32A256'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	R3	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-32A256 device. TRDY is asserted one TXCLK cycle after TSX is sampled high. TRDY is asserted by the FREEDM-32A256 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 the 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	N4 N1 N3 N2 M2 M3 L3 L2 K3 K2 K1 J3 J2 J4 J1 H3	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-32A256 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 8 least significant bits provide the channel number (0 to 255) while the 3 most significant bits select one of seven possible FREEDM-32A256 devices sharing a single external controller. (One address is reserved as a null address.) The FREEDM-32A256 will not respond to channel addresses outside the range 0 to 255, nor to device addresses other than the base address stored in the TAPI256 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	L4	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	L1	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	P3	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	P2	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	P1	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-32A256 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	AC2	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	AC3 Y4 AB2	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-32A256 device for receive data available in any one of its 256 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-32A256 device enabling it to output data on the receive APPI. Note that up to seven FREEDM-32A256 devices may share a single external controller (one address is reserved as a null address). The Rx APPI of each FREEDM-32A256 device is identified by the base address in the RAPI256 Control register. The RXADDR[2:0] signals are sampled on the rising edge of RXCLK.

Pin Name	Type	Pin No.	Function
RPA	Tristate Output	R4	The receive packet available signal (RPA) reflects the status of a poll on the receive APPI of a FREEDM-32A256 device. When RPA is set high, the polled FREEDM-32A256 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-32A256 device does not have data ready to transfer. (XFER[3:0] is a per-channel programmable value – see description of register 0x208.) A FREEDM-32A256 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 RAPI256 Control register, that FREEDM-32A256 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-32A256'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	R1	The receive enable signal (RENB) qualifies the RXADDR[2:0] signals for selection of a FREEDM-32A256 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-32A256 device enabling it to output data on the receive APPI. The Rx APPI of each FREEDM-32A256 device is identified by the base address in the RAPI256 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-32A256 during data transfer on the Rx APPI. When the FREEDM-32A256 samples RENB high during data transfer, the FREEDM-32A256 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-32A256 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-32A256 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	AC1 AB1 AA3 AA1 AA2 Y3 W4 Y1 W3 W1 V3 V1 V2 U1 U4 U2	The receive data signals (RXDATA[15:0]) contain the receive Any-PHY packet interface (APPI) data output by the FREEDM-32A256 when selected. Data is presented in big endian format, i.e. the byte in RXDATA[15:8] was received by the FREEDM-32A256 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 8 least significant bits (RXDATA[7:0]) contain the channel number (0 to 255) 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-32A256 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 tristate when the FREEDM-32A256 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	U3	The receive parity signal (RXPRTY) reflects the odd parity calculated over the RXDATA[15:0] signals. RXPRTY is driven/tristate at the same time as RXDATA[15:0]. RXPRTY is updated on the rising edge of RXCLK.
RSX	Tristate Output	Y2	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-32A256 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 255. 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 tristate when the FREEDM-32A256 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	T3	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 tristate when the FREEDM-32A256 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	T2	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-32A256 device is programmed to overwrite RXDATA[7:0] with status information when errored packets are received. RMOD is tristate when the FREEDM-32A256 device is not selected via the RENB signal. RMOD is updated on the rising edge of RXCLK.
RERR	Tristate Output	T1	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-32A256 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-32A256 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	R2	The receive data valid (RVAL) is asserted when packet data is being output on RXDATA[15:0]. It is deasserted whenever the FREEDM-32A256 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-32A256 device is not selected via the RENB signal. RVAL is updated on the rising edge of RXCLK.

Table 3 – 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	A2 C3 A3 B3 C4 D5 A4 B4 C5 A5 C6 A6 B6 A7 D7 B7	The bi-directional data signals (D[15:0]) provide a data bus to allow the FREEDM-32A256 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-32A256 device.
A[2] A[3] A[4] A[5] A[6] A[7] A[8] A[9] A[10] A[11]	Input	C7 A8 B8 C8 A9 D9 B9 C9 A10 B10	The address signals (A[11:2]) provide an address bus to allow the FREEDM-32A256 device to interface to an external micro-processor. All microprocessor accessible registers are dword aligned.
ALE	Input	C10	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	D11	The write strobe signal (WRB) qualifies write accesses to the FREEDM-32A256 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	A11	The read strobe signal (RDB) qualifies read accesses to the FREEDM-32A256 device. When CSB is set low, the FREEDM-32A256 device drives the D[15:0] bus with the contents of the addressed register on the falling edge of RDB.
CSB	Input	C11	The chip select signal (CSB) qualifies read/write accesses to the FREEDM-32A256 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-32A256 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	B11	The interrupt signal (INTB) indicates that an interrupt source is active and unmasked. When INTB is set low, the FREEDM-32A256 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-32A256 device to determine the exact cause of the interrupt. INTB is an open drain output.

Table 4 – Miscellaneous Interface Signals (9)

Pin Name	Type	Pin No.	Function
SYSCLK	Input	K23	The system clock (SYSCLK) provides timing for the core logic. SYSCLK is nominally a 50% duty cycle, 25 to 45 MHz clock.
RSTB	Input	C22	The active low reset signal (RSTB) signal provides an asynchronous FREEDM-32A256 reset. RSTB is an asynchronous input. When RSTB is set low, all FREEDM-32A256 registers are forced to their default states. In addition, TD[31:0] are forced high and all APPI output pins are forced tristate and will remain high or tristated, respectively, until RSTB is set high.
PMCTEST	Input	AB3	The PMC production test enable signal (PMCTEST) places the FREEDM-32A256 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	T23	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	T22	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	U21	The test data input signal (TDI) carries test data into the FREEDM-32A256 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	U22	The test data output signal (TDO) carries test data out of the FREEDM-32A256 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	T21	The active low test reset signal (TRSTB) provides an asynchronous FREEDM-32A256 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.
NC	Open	C2	This pin must be left unconnected.

Table 5 – Production Test Interface Signals (0 - Multiplexed)

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	G23 F23 E23 D22 E20 C23 A22 D20 B21 D19 B20 A19	The test mode address bus (TA[11:0]) selects specific registers during production test (PMCTEST set high) read and write accesses. TA[11:0] replace RD[21:10] when PMCTEST is set high.
TA[12]/ TRS	Input	A16	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. TA[12]/TRS replaces RD[24] when PMCTEST is set high.
TRDB	Input	A18	The test mode read enable signal (TRDB) is set low during FREEDM-32A256 register read accesses during production test (PMCTEST set high). The FREEDM-32A256 drives the test data bus (TDAT[15:0]) with the contents of the addressed register while TRDB is low. TRDB replaces RD[22] when PMCTEST is set high.
TWRB	Input	A17	The test mode write enable signal (TWRB) is set low during FREEDM-32A256 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. TWRB replaces RD[23] when PMCTEST is set high.

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	AC14 AA14 AC13 AB13 AA12 AB11 Y11 AB10 Y9 AA8 AC8 AB7 AC7 AC6 AC5 AB4	The bi-directional test mode data bus (TDAT[15:0]) carries data read from or written to FREEDM-32A256 registers during production test. TDAT[15:0] replace TD[31:16] when PMCTEST is set high.

Table 6 – Power and Ground Signals (65)

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]	Power	D6 D10 D14 D18 H4 H20 M4 M20 T4 T20 Y6 Y10 Y14 Y18	The VDD3V3[14:1] 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.

Pin Name	Type	Pin No.	Function
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	E2 M1 W2 AB5 AC12 AB19 W22 M23 E22 B19 A12 B5	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.
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]	Ground	D8 D12 D16 F4 F20 K4 K20 P4 P20 V4 V20 Y8 Y12 Y16	The VSS[14:1] 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.

Pin Name	Type	Pin No.	Function
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] VSS[29] VSS[30] VSS[31] VSS[32] VSS[33] VSS[34] VSS[35] VSS[36] VSS[37] VSS[38] VSS[39]		K10 K11 K12 K13 K14 L10 L11 L12 L13 L14 M10 M11 M12 M13 M14 N10 N11 N12 N13 N14 P10 P11 P12 P13 P14	The VSS[39:15] DC ground pins should be connected to ground. They provide improved thermal properties for the 329 PBGA package.

Notes on Pin Description:

1. All FREEDM-32A256 inputs and bi-directionals present minimum capacitive loading and, with the exception of the Any-PHY interface, are 5V tolerant. (The Any-PHY interface is 3.3V tolerant.)

2. All FREEDM-32A256 digital outputs and bi-directionals have 4 mA drive capability except the RBCLK, TBCLK, RBD, D[15:0] and INTB outputs which have 8 mA drive capability and the Any-PHY outputs (TPAn[2:0], TRDY, RPA, RSX, REOP, RXDATA[15:0], RXPRTY, RMOD and RERR) which also have 8 mA drive capability.

3. All FREEDM-32A256 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 with the exception of the Any-PHY interface are 5 V tolerant when tristated. (The Any-PHY interface is 3.3V tolerant.)

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

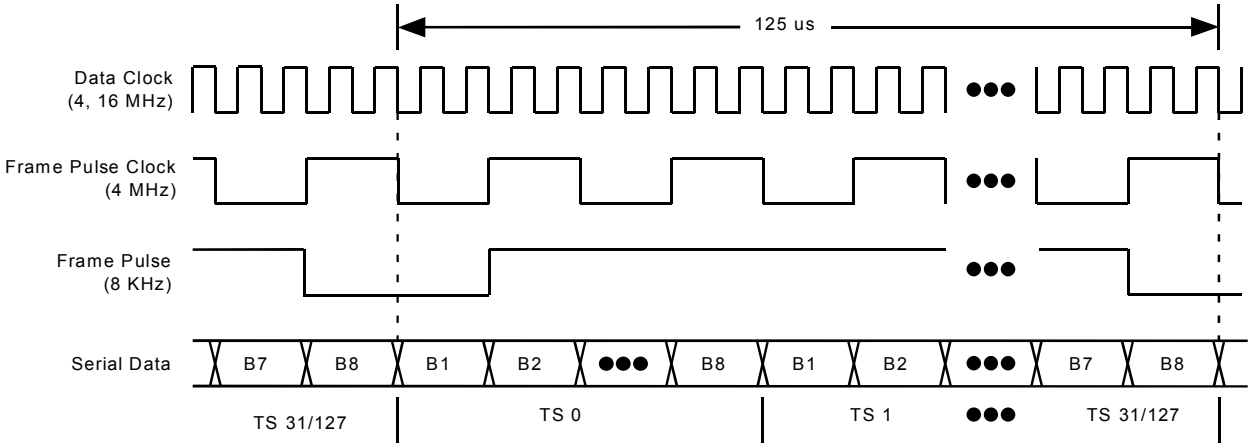
5. Power to the VDD3V3 pins should be applied before power to the VDD2V5 pins is applied. Similarly, power to the VDD2V5 pins should be removed before power to the VDD3V3 pins is removed.

8 FUNCTIONAL DESCRIPTION

8.1 High Speed Multi-Vendor Integration Protocol (H-MVIP)

H-MVIP defines a synchronous, time division multiplexed (TDM) bus of Nx64 Kbps constant bit rate (CBR) data streams. Each 64 Kbps data stream (time-slot) carries an 8-bit byte of HDLC traffic, as described in the following section, and is characterised by 8 KHz framing. H-MVIP supports higher bandwidth applications on existing telephony networks by fitting more time-slots into a 125 μs frame. The FREEDM-32A256 supports H-MVIP data rates of 2.048 Mbps and 8.192 Mbps with 32 or 128 time-slots per frame and associated clocking frequencies of 4.096 and 16.384 MHz respectively. Figure 1 shows a diagram of the H-MVIP protocol supported by the FREEDM-32A256 device.

Figure 1 – H-MVIP Protocol

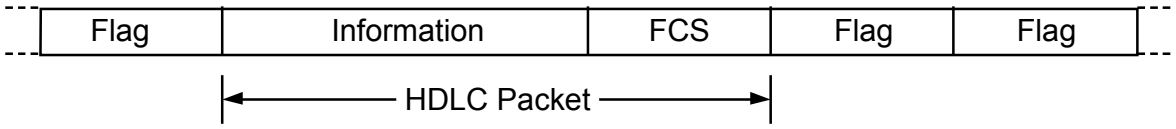


8.2 High-Level Data Link Control (HDLC) Protocol

Figure 2 shows a diagram of the synchronous HDLC protocol supported by the FREEDM-32A256 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

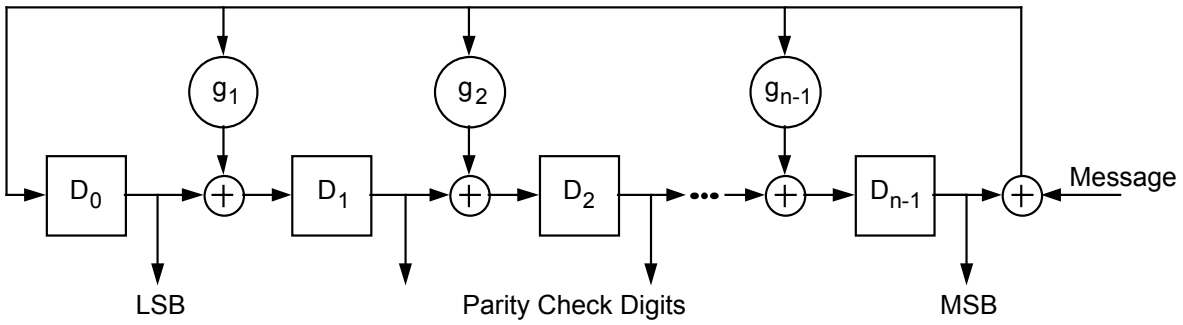
FCS, the minimum packet length is four bytes while those with CRC-32 as FCS, the minimum length is six bytes. An HDLC packet is aborted when seven contiguous "1" bits (with no inserted "0" bits) are received. At least one flag byte must exist between HDLC packets for delineation. Contiguous flag bytes, or all ones bytes between packets are used as an "inter-frame time fill". Adjacent flag bytes may share zeros.

Figure 2 – HDLC Frame



The CRC algorithm for the frame checking sequence (FCS) field is either a CRC-CCITT or CRC-32 function. Figure 3 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 3 – CRC Generator



8.3 Receive Channel Assigner

The Receive Channel Assigner block (RCAS256) processes up to 32 serial links. Links may be configured to support 2.048 or 8.192 Mbps H-MVIP traffic, to support T1/J1/E1 channelised traffic or to support unchannelised traffic. When configured to support 2.048 Mbps H-MVIP traffic, each group of 8 links share a clock and frame pulse. All links configured for 8.192 Mbps H-MVIP traffic share a

common clock and frame pulse. For T1/J1/E1 channelised traffic or for unchannelised traffic, each link is independent and has its own associated clock. For each link, the RCAS256 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 (RHDL256) 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 #31 the lowest.

From the point of view of the RCAS256, links configured for H-MVIP traffic behave identically to links configured for T1/J1/E1 channelised or unchannelised traffic in the back end, only differing on the link side as described herein. First, the number of time-slots in each frame is programmable to be 32 or 128 and has an associated data clock frequency that is double the data rate. This provides more bandwidth per link for applications requiring higher data densities on a single link. Second, H-MVIP links reference the start of each frame with a frame pulse, thereby avoiding having to gap the link clock during the framing bits/bytes of each frame. The frame pulse is provided by an H-MVIP bus master and ensures that all agents sharing the H-MVIP bus remain synchronized. When configured for operation in 2.048 Mbps mode, the frame pulse is sampled using the same clock which samples the data. When configured for operation in 8.192 Mbps H-MVIP mode, the frame pulse is sampled using a separate frame pulse clock provided by an H-MVIP bus master. The frame pulse clock has a synchronous timing relationship to the data clock. Third, not all links are independent. When configured for operation in 2.048 Mbps H-MVIP mode, each group of 8 links share a clock and a frame pulse. Links 0 through 7, 8 through 15, 16 through 23 and 24 through 31 each share a clock and a frame pulse. Not all 8 links within each group need to be configured for operation in 2.048 Mbps H-MVIP mode. However, any link within each logical group of 8 which is configured for 2.048 Mbps H-MVIP operation will share the same clock and frame pulse. When configured for operation in 8.192 Mbps H-MVIP mode, links 4m ($0 \leq m \leq 7$) share a frame pulse, a data clock and a frame pulse clock. Again, not all eight 4m ($0 \leq m \leq 7$) links need to be configured for operation in 8.192 Mbps H-MVIP mode, however, any link which is configured for 8.192 Mbps H-MVIP operation will share the same frame pulse, data clock and frame pulse clock. If link 4m is configured for 8.192 Mbps H-MVIP operation, then data transferred on that link is “spread” over links 4m, 4m+1, 4m+2 and 4m+3 from a channel assigner point of view. Accordingly, when link 4m is configured for operation in 8.192 Mbps H-MVIP mode, links 4m+1, 4m+2 and 4m+3 must also be configured for operation in 8.192 Mbps H-MVIP mode. In the back end, the RCAS256 extracts and processes the time-slots in the same way as channelised T1/J1/E1 traffic.

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 RCAS256 performs a table lookup to associate the link and time-slot identity with a channel. T1/J1 and E1 framing bits/bytes are identified by observing the gap in the link clock which is squelched during the framing bits/bytes. For unchannelised links, clock rates are limited to 51.84 MHz for links #0 to #2 and limited to 10 MHz for the remaining links. All data on each link belongs to one channel. For the case of a mixture of channelised, unchannelised and H-MVIP links, the total instantaneous link rate over all the links is limited to 64 MHz. The RCAS256 performs a table lookup using only the link number to determine the associated channel, as time-slots are non-existent in unchannelised links.

The RCAS256 provides diagnostic loopback that is selectable on a per channel basis. The RCAS256 does not support diagnostic loopback for links configured as H-MVIP. When a channel is in diagnostic loopback, stream data on the received links originally destined for that channel is ignored. Transmit data of that channel is substituted in its place.

8.3.1 Line Interface Translator (LIT)

The LIT block translates the information on the 32 physical links into a suitable format for interpretation by the Line Interface block. The LIT block performs three functions: data translation, clock translation and frame pulse generation.

When link 4m ($0 \leq m \leq 7$) is configured for operation in 8.192 Mbps H-MVIP mode, the LIT block translates the 128 time-slots on link 4m to the Line Interface block across links 4m, 4m+1, 4m+2 and 4m+3. The LIT block provides time-slots 0 through 31, 32 through 63, 64 through 95 and 96 through 127 to the Line Interface block on links 4m, 4m+1, 4m+2 and 4m+3 respectively. When link 4m is configured for operation in 8.192 Mbps H-MVIP mode, data cannot be received on inputs RD[4m+3:4m+1]. However, links 4m+1, 4m+2 and 4m+3 must be programmed in the RCAS256 Link Configuration register for 8.192 Mbps H-MVIP operation. When links are configured for operation in 2.048 Mbps H-MVIP mode, channelised T1/J1/E1 mode or unchannelised mode, the LIT block does not perform any translation on the link data.

When a link is configured for operation in H-MVIP mode, the LIT block divides the appropriate clock (RMVCK[n] for 2.048 Mbps H-MVIP and RMV8DC for 8.192 Mbps H-MVIP) by two and provides this divided down clock to the Line Interface block. When a link is configured for operation in channelised T1/J1/E1 or unchannelised mode, the LIT block does not perform any translation on the link clock.

When a link is configured for operation in H-MVIP mode, the LIT block samples the appropriate frame pulse (RFPB[n] for 2.048 Mbps H-MVIP and RFP8B for 8.192 Mbps H-MVIP) and presents the sampled frame pulse to the Line Interface block. When a link is configured for operation in channelised T1/J1/E1 or unchannelised mode, the gapped clock is passed to the LIT block unmodified.

8.3.2 Line Interface

There are 32 identical line interface blocks in the RCAS256. Each line interface block contains 2 sub-blocks; one supporting channelised T1/J1/E1 streams and the other H-MVIP streams. Based on configuration, only one of the sub-blocks are active at one time; the other is held reset. Each sub-block contains a bit counter, an 8-bit shift register and a holding register. Each sub-block performs 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 would respond with the data residing in the holding register in the active sub-block.

To support H-MVIP links, each line interface block contains a time-slot counter. The time-slot counter is incremented each time the holding register is updated. When a frame pulse occurs, the time-slot counter is initialised to indicate that the next bit is the most significant bit of the first time-slot.

To support non H-MVIP channelised links, each line interface block contains a time-slot counter and a clock activity monitor. The time-slot counter is incremented each time the holding register is updated. The clock activity monitor is a counter that increments at the system clock (SYSCLK) rate and is cleared by a rising edge of the receive clock (RCLK[n]). A framing bit (T1/J1) or a framing byte (E1) is detected when the counter reaches a programmable threshold, in which case, the bit and time-slot counters are initialised to indicate that the next bit is the most significant bit of the first time-slot. For unchannelised links, the time-slot counter and the clock activity monitor are held reset.

8.3.3 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 RD[0] to that attached to RD[31]. Thus, simultaneous requests from RD[m] will be serviced ahead of RD[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.

8.3.4 Channel Assigner

The channel assigner block determines the channel number of the data byte currently being processed. The block contains a 1024 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 indicted by the channel number field.

8.3.5 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 (THDL256) is written into a 64 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.

8.4 Receive HDLC Processor / Partial Packet Buffer

The Receive HDLC Processor / Partial Packet Buffer block (RHDL256) processes up to 256 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 RHDL256 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 (RAPI256) to transfer, on the Rx APPI, data in channel FIFOs with depths exceeding their associated threshold.

8.4.1 HDLC Processor

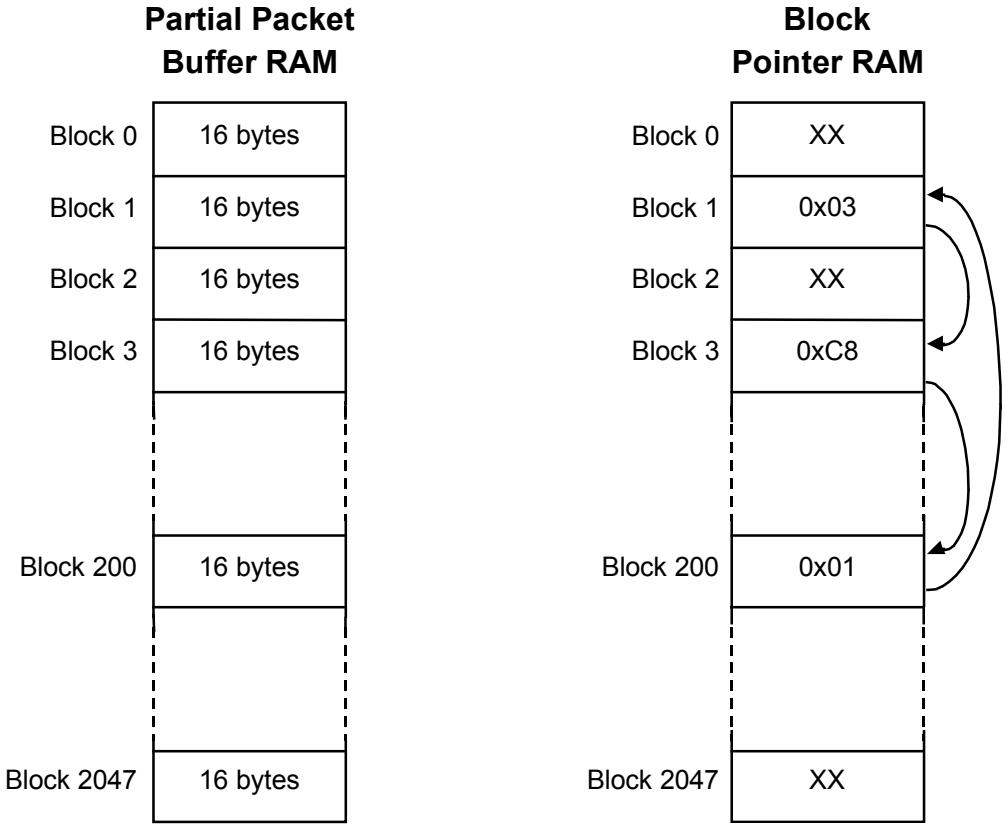
The HDLC processor is a time-slice state machine which can process up to 256 independent channels. The state vector and provisioning information for each channel is stored in a RAM. Whenever new channel data arrives, the appropriate state vector is read from the RAM, processed and written back to the 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 (RCAS256). Writing new provisioning data to a channel resets the channel's entire state vector.

8.4.2 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 4 shows an example of three blocks (blocks 1, 3, and 200) linked together to form a 48 byte channel FIFO.

Figure 4 – 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 (RAPI256). 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 RAPI256 block, a transaction is deleted. Thus, small packets less than the transfer size will be naturally transferred to the RAPI256 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 RAPI256 block. The roamer informs the partial packet reader of the channel to process. The reader transfers the data to the RAPI256 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 RAPI256 block.

8.5 Receive Any-PHY Interface

The Receive Any-PHY Interface (RAPI256) provides a low latency path for transferring data out of the partial packet buffer in the RHDL256 and onto the Receive Any-PHY Packet Interface (Rx APPI). The RAPI256 contains a FIFO block for latency control as well as to segregate the APPI timing domain from the SYSCLK timing domain. The RAPI256 contains the necessary logic to manage and respond to device polling from an upper layer device. The RAPI256 also provides the upper layer device with status information on a per packet basis.

8.5.1 FIFO Storage and Control

The FIFO block temporarily stores channel data during transfer across the Rx APPI. RAPI256 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 RAPI256 can maintain continuous full bandwidth transfer over the Rx APPI.

A maximum of 256 bytes can be stored in one 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 every burst data tansfer across the Rx APPI identify the FREEDM-32A256 device associated with the transfer and reflect the value of the base address programmed in the RAPI256 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 complete burst transfer of data, up to a maximum of 256 bytes, is written into that empty FIFO. (The transfer is sourced by the upstream RHDL256 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 RHDL256 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 RAPI256 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-32A256 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 RAPI256 provides packet status information on the Rx APPI at the end of every packet transfer. The RAPI256 asserts RERR at the end of packet reception (REOP high) to indicate that the packet is in error. The RAPI256 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). Overwriting of status information is enabled by setting the STATEN bit in the RAPI Control register.

8.5.2 Polling Control and Management

The RAPI256 only responds to device polls which match the base address programmed in the RAPI256 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.

8.6 Transmit Any-PHY Interface

The Transmit Any-PHY Interface (TAPI256) provides a low latency path for transferring data from the Transmit Any-PHY Packet Interface (Tx APPI) into the partial packet buffer in the THDL256. The TAPI256 contains a FIFO block for latency control as well as to segregate the APPI timing domain from the SYSCLK timing domain. The TAPI256 contains the necessary logic to manage and respond to channel polling from an upper layer device.

8.6.1 FIFO Storage and Control

The FIFO block temporarily stores channel data during transfer across the Tx APPI. TAPI256 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 TAPI256 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-32A256 may throttle the upper layer device if both FIFOs in the TAPI256 are full. When the FIFOs in the TAPI256 cannot accept data, the TAPI256 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 one 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 TAPI256 Control register for the TAPI256 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 THDL256 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 THDL256 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 the FIFO. Because the reader controller reads data out of the FIFOs in the order in which they were filled, the THDL256 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.

8.6.2 Polling Control and Management

The TAPI256 only responds to poll addresses which are in the range programmed in the base address field in the TAPI256 Control register. The TAPI256 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 TAPI256 provides three poll results for every poll address according to Table 7. 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 7 – 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 255	Channel 255	Channel 255	Channel 255	Channel 0	Channel 0	Channel 0

The TAPI256 maintains a mirror image of the status of each channel FIFO in the partial packet buffer. The THDL256 continuously reports the status of the 256 channel FIFOs to the TAPI256 and the TAPI256 updates the mirror image accordingly. The THDL256 also signals to the TAPI256 whenever an underflow event has occurred on a channel FIFO. At the beginning of every data transfer across the Tx APPI, the TAPI256 sets the mirror image status of the channel to “full”. Only the TAPI256 can cause the status to be set to “full” and only the THDL256 can cause the status to be set to “space” or “starving”. Only the THDL256 can cause the status to be set to “underflow” and only the TAPI256 can clear the “underflow” status when that channel FIFO is polled. In the event that

both the TAPI256 and the THDL256 try to change the mirror image status of a particular channel simultaneously, the TAPI256 takes precedence, except for the "underflow" status, where the THDL256 takes precedence.

8.7 Transmit HDLC Controller / Partial Packet Buffer

The Transmit HDLC Controller / Partial Packet Buffer block (THDL256) contains a partial packet buffer for Tx APPI latency control and a transmit HDLC controller. The THDL256 also contains logic to monitor the full/empty status of each channel FIFO and push this status onto the polling interface signals.

The THDL256 requests data from the TAPI256 in response to control information from the TAPI256 indicating the channel for which data is available and ready to be transferred. Packet data received from the TAPI256 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 (TCAS256) on demand. A packet in progress is aborted if an under-run occurs. The THDL256 is programmable to operate in transparent mode where packet data retrieved from the TAPI256 is transmitted verbatim.

8.7.1 Transmit HDLC Processor

The HDLC processor is a time-slice state machine which can process up to 256 independent channels. The state vector and provisioning information for each channel is stored in a RAM. Whenever the TCAS256 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.

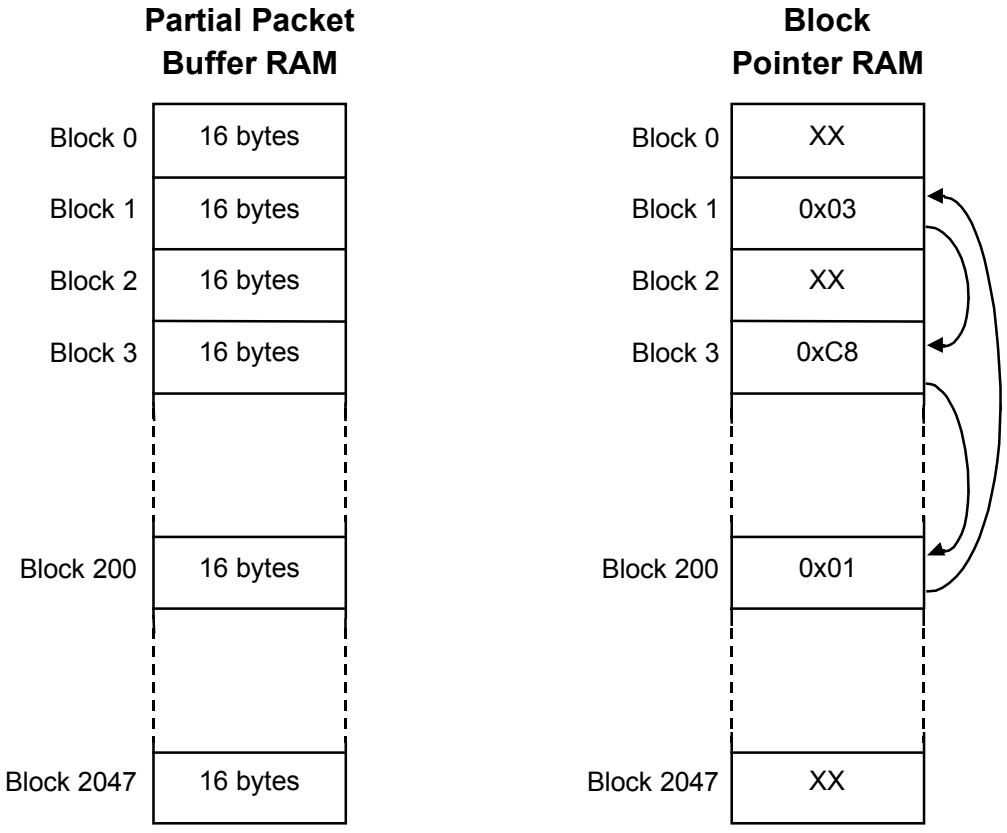
The configuration of the HDLC processor is accessed using indirect channel read and write operations. When an indirect operation is performed, the information is

accessed from RAM during a null clock cycle inserted by the TCAS256 block. Writing new provisioning data to a channel resets the channels entire state vector.

8.7.2 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 5 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 5 – 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 TAPI256 block and transfers packet data from the TAPI256 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, the THDL256 disables the channel as if it were unprovisioned and does not transmit any further data until that channel is reprovisioned. In both cases, 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 TAPI256 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 free space 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 TAPI256. The roamer listens to control information from the TAPI256 to decide which channel FIFO requests data from the TAPI256 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 TAPI256 block and writes the response data to the channel FIFO setting 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.

The roamer round-robins between all channel FIFOs and pushes the status to the TAPI256 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 THDL256 pushes a starving status for that channel FIFO to the TAPI256 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 TCAS256 block. Writing new provisioning data to a channel resets the entire state vector.

8.8 Transmit Channel Assigner

The Transmit Channel Assigner block (TCAS256) processes up to 256 channels. Data for all channels is sourced from a single byte-serial stream from the Transmit HDLC Controller / Partial Packet Buffer block (THDL256). The TCAS256 demultiplexes the data and assigns each byte to any one of 32 links. Each link may be configured to support 2.048 or 8.192 H-MVIP traffic, to support T1/J1/E1 channelised traffic or to support unchannelised traffic. When configured to support H-MVIP traffic, each group of 8 links share a clock and frame pulse, otherwise each link is independent and has its own associated clock. For each high-speed link (TD[2:0]), the TCAS provides a six byte FIFO. For the remaining links (TD[31:3]), the TCAS provides a single byte holding register. The TCAS256 also performs parallel to serial conversion to form a bit-serial stream. In the event where multiple links are in need of data, TCAS256 requests data from upstream blocks on a fixed priority basis with link TD[0] having the highest priority and link TD[31] the lowest.

From the point of view of the TCAS256, links configured for H-MVIP traffic behave identically to links configured for T1/J1/E1 channelised or unchannelised traffic in the back end, only differing on the link side as described herein. First, the number of time-slots in each frame is programmable to be 32 or 128 and has an associated data clock frequency that is double the data rate. This provides

more bandwidth per link for applications requiring higher data densities on a single link. Data at each time-slot may be independently assigned to be sourced from a different channel. Second, H-MVIP links reference the start of each frame with a frame pulse, thereby avoiding having to gap the link clock during the framing bits/bytes of each frame. The frame pulse is provided by an H-MVIP bus master and ensures that all agents sharing the H-MVIP bus remain synchronized. When configured for operation in 2.048 Mbps H-MVIP mode, the frame pulse is sampled using the same clock which samples the data. When configured for operation in 8.192 Mbps H-MVIP mode, the frame pulse is sampled using a separate frame pulse clock provided by an H-MVIP bus master. The frame pulse clock has a synchronous timing relationship to the data clock. Third, not all links are independent. When configured for operation in 2.048 Mbps H-MVIP mode, each group of 8 links share a clock and a frame pulse. Links 0 through 7, 8 through 15, 16 through 23 and 24 through 31 each share a clock and a frame pulse. Not all 8 links within each group need to be configured for operation in 2.048 Mbps H-MVIP mode. However, any link within each logical group of 8 which is configured for 2.048 Mbps H-MVIP operation will share the same clock and frame pulse. When configured for operation in 8.192 Mbps H-MVIP mode, links 4m ($0 \leq m \leq 7$) share a frame pulse, a data clock and a frame pulse clock. Again, not all eight 4m ($0 \leq m \leq 7$) links need to be configured for operation in 8.192 Mbps H-MVIP mode, however, any link which is configured for 8.192 Mbps H-MVIP operation will share the same frame pulse, data clock and frame pulse clock. If link 4m is configured for 8.192 Mbps H-MVIP operation, then data transferred on that link is “spread” over links 4m, 4m+1 4m+2 and 4m+3 from a channel assigner point of view. Accordingly, when link 4m is configured for operation in 8.192 Mbps H-MVIP mode, links 4m+1, 4m+2 and 4m+3 must also be configured for operation in 8.192 Mbps H-MVIP mode. In the back end, the TCAS256 extracts and processes the time-slots identically to channelised T1/J1/E1 traffic.

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 link clock is only active during time-slots 1 to 24 of a T1/J1 stream and is inactive during the frame bit. Similarly, the clock is only active during time-slots 1 to 31 of an E1 stream and is inactive during the FAS and NFAS framing bytes. The most significant bit of time-slot 1 of a channelised link is identified by noting the absence of the clock and its re-activation. With knowledge of the transmit link and time-slot identity, the TCAS256 performs a table look-up to identify the channel from which a data byte is to be sourced.

Links may also be unchannelised. Then, all data bytes on that link belong to one channel. The TCAS256 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. Link clocks are no longer limited to T1/J1 or E1 rates and may range up to a maximum clock rate of 51.84 MHz for TCLK[2:0] and 10 MHz for TCLK[31:3]. The link clock is only active during bit times containing data to be transmitted and inactive during bits that are to be ignored by the downstream devices, such as framing and overhead bits. For the case of three unchannelised links, the maximum link rate is 51.84 MHz. For the case of more numerous unchannelised links or a mixture of channelised, unchannelised and H-MVIP links, the total instantaneous link rate over all the links is limited to 64 MHz.

8.8.1 Line Interface Translator (LIT)

The LIT block translates the information between the 32 physical links and the Line Interface block. The LIT block performs three functions: data translation, clock translation and frame pulse generation.

When link 4m ($0 \leq m \leq 7$) is configured for operation in 8.192 Mbps H-MVIP mode, the LIT block translates the data arriving from the Line Interface block on links 4m, 4m+1, 4m+2 and 4m+3 onto the 128 time-slot link 4m. The LIT block translates data arriving from the Line Interface block on link 4m, 4m+1, 4m+2 and 4m+3 onto time-slots 0 through 31, 32 through 63, 64 through 95 and 96 through 127 respectively. When link 4m is configured for operation in 8.192 Mbps H-MVIP mode, outputs TD[4m+3:4m+1] are driven with constant ones. However, links 4m+1, 4m+2 and 4m+3 must be programmed in the TCAS256 Link Configuration register for 8.192 Mbps H-MVIP operation. When links are configured for operation in 2.048 Mbps H-MVIP mode, channelised T1/J1/E1 mode or unchannelised mode, the LIT block does not perform any translation on the link data.

When a link is configured for operation in H-MVIP mode, the LIT block divides the appropriate clock (TMVCK[n] for 2.048 Mbps H-MVIP and TMV8DC for 8.192 Mbps H-MVIP) by two and provides this divided down clock to the Line Interface block. When a link is configured for operation in channelised T1/J1/E1 or unchannelised mode, the LIT block does not perform any translation on the link clock.

When a link is configured for operation in H-MVIP mode, the LIT block samples the appropriate frame pulse (TFPB[n] for 2.048 Mbps H-MVIP and TFP8B for 8.192 Mbps H-MVIP) and presents the sampled frame pulse to the Line Interface block. When a link is configured for operation in channelised T1/J1/E1 or unchannelised mode, the gapped clock is passed to the LIT block unmodified.

8.8.2 Line Interface

There are 32 identical line interface blocks in the TCAS256. Each line interface block contains 2 sub-blocks; one supporting channelised T1/J1/E1 streams and the other H-MVIP streams. Based on configuration, only one of the sub-blocks are active at one time; the other is held reset. Each sub-block contains a bit counter, an 8-bit shift register and a holding register. Each sub-block performs 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 would respond by writing the data into the holding register in the active sub-block.

To support H-MVIP links, each line interface block contains a time-slot counter. The time-slot counter is incremented each time the holding register is updated. When a frame pulse occurs, the time-slot counter is cleared to indicate that the next byte belongs to the first time-slot.

To support non H-MVIP channelised links, each line interface block contains a time-slot counter and a clock activity monitor. The time-slot counter is incremented each time the shift register is updated. The clock activity monitor is a counter that increments at the system clock (SYSCLK) rate and is cleared by a rising edge of the transmit clock (TCLK[n]). A framing bit (T1/J1) or a framing byte (E1) is detected when the counter reaches a programmable threshold, at which point, the bit and time-slot counters are initialised to indicate that the next bit sampled is the most significant bit of the first time-slot. For unchannelised links, the time-slot counter and the clock activity monitor are held reset.

8.8.3 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 line interface TD[0] to line interface TD[31]. Thus, simultaneous requests from line interface TD[m] will be serviced ahead of line interface TD[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 downstream for CBI accesses to the channel provision RAM.

8.8.4 Channel Assigner

The channel assigner block determines the channel number of the request currently being processed. The block contains a 1024 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 THDL256 block which responds with packet data within one byte period of the transmit stream.

8.9 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-32A256 Master Clock / BERT 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.

8.10 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-32A256 identification code is 073830CD hexadecimal.

8.11 Microprocessor Interface

The FREEDM-32A256 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 8 – Normal Mode Microprocessor Accessible Registers

Address	Register
0x000	FREEDM-32A256 Master Reset
0x004	FREEDM-32A256 Master Interrupt Enable
0x008	FREEDM-32A256 Master Interrupt Status
0x00C	FREEDM-32A256 Master Clock / Frame Pulse / BERT Activity Monitor and Accumulation Trigger
0x010	FREEDM-32A256 Master Link Activity Monitor
0x014	FREEDM-32A256 Master Line Loopback #1
0x018	FREEDM-32A256 Master Line Loopback #2
0x01C	FREEDM-32A256 Reserved
0x020	FREEDM-32A256 Master BERT Control
0x024	FREEDM-32A256 Master Performance Monitor Control
0x028 – 0x0FC	Reserved
0x100	RCAS Indirect Channel and Time-slot Select
0x104	RCAS Indirect Channel Data
0x108	RCAS Framing Bit Threshold
0x10C	RCAS Channel Disable
0x110 – 0x17C	RCAS Reserved
0x180 – 0x1FC	RCAS Link #0 through #31 Configuration
0x200	RHDL Indirect Channel Select
0x204	RHDL Indirect Channel Data Register #1
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

Address	Register
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 Framing Bit Threshold
0x40C	TCAS Idle Time-slot Fill Data
0x410	TCAS Channel Disable
0x414 – 0x47C	TCAS Reserved
0x480 – 0x4FC	TCAS Link #0 through #31 Configuration
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

Address	Register
0x580	RAPI Control
0x584 – 0x5BC	RAPI Reserved
0x5C0 – 0x5FC	Reserved
0x600	TAPI Control
0x604	TAPI Indirect Channel Provisioning
0x608	TAPI Indirect Channel Data Register
0x60C – 0x63C	TAPI Reserved
0x640 – 0x7FC	Reserved

9 NORMAL MODE REGISTER DESCRIPTION

Normal mode registers are used to configure and monitor the operation of the FREEDM-32A256.

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-32A256 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-32A256 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-32A256 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-32A256 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-32A256 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]	0
Bit 9	R	TYPE[1]	1
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-32A256 to be reset under software control. If the RESET bit is a logic one, the entire FREEDM-32A256, except the microprocessor interface, is held in reset. This bit is not self-clearing. Therefore, a logic zero must be written to bring the FREEDM-32A256 out of reset. Holding the FREEDM-32A256 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

Like the hardware reset input (RSTB), RESET forces the FREEDM-32A256's transmit link data pins (TD[31:0]) high and the APPI outputs tristate.

TYPE[3:0]:

The Device Type bits (TYPE[3:0]) allow software to identify the device as the FREEDM-32A256 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-32A256.

Register 0x004 : FREEDM-32A256 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-32A256.

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-32A256 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-32A256. 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-32A256 Master Clock / Frame Pulse / BERT 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	R	TFPA[3]	X
Bit 10	R	TFPA[2]	X
Bit 9	R	TFPA[1]	X
Bit 8	R	TFPA[0]	X
Bit 7	R	RFPA[3]	X
Bit 6	R	RFPA[2]	X
Bit 5	R	RFPA[1]	X
Bit 4	R	RFPA[0]	X
Bit 3	R	TFP8A	X
Bit 2	R	RFP8A	X
Bit 1	R	TBDA	X
Bit 0	R	SYSCLKA	X

This register provides activity monitoring on the FREEDM-32A256 system clock, Any-PHY clocks, H-MVIP frame pulse and BERT port 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.

TBDA:

The transmit BERT data active bit (TBDA) monitors for low to high transitions on the TBD input. TBDA is set high on a rising edge of TDB, and is set low when this register is read.

RFP8A:

The receive 8.192 Mbps H-MVIP frame pulse activity bit (RFP8A) monitors for low to high transitions on the RFP8B input. RFP8A is set high when RFP8B has been sampled low and sampled high by falling edges of the RMV8FPC, and is set low when this register is read.

TFP8A:

The transmit 8.192 Mbps H-MVIP frame pulse activity bit (TFP8A) monitors for low to high transitions on the TFP8B input. TFP8A is set high when TFP8B has been sampled low and sampled high by falling edges of the TMV8FPC, and is set low when this register is read.

RFPA[3:0]:

The receive frame pulse activity bits (RFPA[3:0]) monitor for low to high transitions on the RFPB[3:0] inputs. RFPA[n] is set high when RFPB[n] has been sampled low and sampled high by falling edges of the corresponding RMVCK[n], and is set low when this register is read.

TFPA[3:0]:

The transmit frame pulse activity bits (TFPA[3:0]) monitor for low to high transitions on the TFPB[3:0] inputs. TFPA[n] is set high when TFPB[n] has been sampled low and sampled high by falling edges of the corresponding TMVCK[n], and is set low when this register is read.

RXCLKA / TXCLKA:

The Any-PHY clock active bits (RXCLKA, TXCLKA) monitor for low to high transitions on the RXCLK and TXCLK inputs respectively. RXCLKA and

TXCLKA are set high on a rising edge of the corresponding clock, and are set low when this register is read.

Register 0x010 : FREEDM-32A256 Master Link Activity Monitor

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

This register provides activity monitoring on FREEDM-32A256 receive and transmit link inputs. When a monitored input makes a low to high 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.

RLGA[0]:

The receive link group #0 active bit (RLGA[0]) monitors for transitions on the RD[3:0] and RCLK[3:0]/RMVCK[0]/RMV8DC inputs. RLGA[0] is set high when either:

1. Each of RD[3:0] has been sampled low and sampled high by rising edges of the corresponding RCLK[3:0] inputs, or

2. Each of RD[3:0] has been sampled low and sampled high by rising edges of the RMVCK[0] input, or
3. RD[0] has been sampled low and sampled high by rising edges of the RMV8DC input.

RLGA[0] is set low when this register is read.

RLGA[1]:

The receive link group #1 active bit (RLGA[1]) monitors for transitions on the RD[7:4] and RCLK[7:4]/RMVCK[0]/RMV8DC inputs. RLGA[1] is set high when either:

1. Each of RD[7:4] has been sampled low and sampled high by rising edges of the corresponding RCLK[7:4] inputs, or
2. Each of RD[7:4] has been sampled low and sampled high by rising edges of the RMVCK[0] input, or
3. RD[4] has been sampled low and sampled high by rising edges of the RMV8DC input.

RLGA[1] is set low when this register is read.

RLGA[2]:

The receive link group #2 active bit (RLGA[2]) monitors for transitions on the RD[11:8] and RCLK[11:8]/RMVCK[1]/RMV8DC inputs. RLGA[2] is set high when either:

1. Each of RD[11:8] has been sampled low and sampled high by rising edges of the corresponding RCLK[11:8] inputs, or
2. Each of RD[11:8] has been sampled low and sampled high by rising edges of the RMVCK[1] input, or
3. RD[8] has been sampled low and sampled high by rising edges of the RMV8DC input.

RLGA[2] is set low when this register is read.

RLGA[3]:

The receive link group #3 active bit (RLGA[3]) monitors for transitions on the RD[15:12] and RCLK[15:12]/RMVCK[1]/RMV8DC inputs. RLGA[3] is set high when either:

1. Each of RD[15:12] has been sampled low and sampled high by rising edges of the corresponding RCLK[15:12] inputs, or
2. Each of RD[15:12] has been sampled low and sampled high by rising edges of the RMVCK[1] input, or
3. RD[12] has been sampled low and sampled high by rising edges of the RMV8DC input.

RLGA[3] is set low when this register is read.

RLGA[4]:

The receive link group #4 active bit (RLGA[4]) monitors for transitions on the RD[19:16] and RCLK[19:16]/RMVCK[2]/RMV8DC inputs. RLGA[4] is set high when either:

1. Each of RD[19:16] has been sampled low and sampled high by rising edges of the corresponding RCLK[19:16] inputs, or
2. Each of RD[19:16] has been sampled low and sampled high by rising edges of the RMVCK[2] input, or
3. RD[16] has been sampled low and sampled high by rising edges of the RMV8DC input.

RLGA[4] is set low when this register is read.

RLGA[5]:

The receive link group #5 active bit (RLGA[5]) monitors for transitions on the RD[23:20] and RCLK[23:20]/RMVCK[2]/RMV8DC inputs. RLGA[5] is set high when either:

1. Each of RD[23:20] has been sampled low and sampled high by rising edges of the corresponding RCLK[23:20] inputs, or
2. Each of RD[23:20] has been sampled low and sampled high by rising edges of the RMVCK[2] input, or
3. RD[20] has been sampled low and sampled high by rising edges of the RMV8DC input.

RLGA[5] is set low when this register is read.

RLGA[6]:

The receive link group #6 active bit (RLGA[6]) monitors for transitions on the RD[27:24] and RCLK[27:24]/RMVCK[3]/RMV8DC inputs. RLGA[6] is set high when either:

1. Each of RD[27:24] has been sampled low and sampled high by rising edges of the corresponding RCLK[27:24] inputs, or
2. Each of RD[27:24] has been sampled low and sampled high by rising edges of the RMVCK[3] input, or
3. RD[24] has been sampled low and sampled high by rising edges of the RMV8DC input.

RLGA[6] is set low when this register is read.

RLGA[7]:

The receive link group #7 active bit (RLGA[7]) monitors for transitions on the RD[31:28] and RCLK[31:28]/RMVCK[3]/RMV8DC inputs. RLGA[7] is set high when either:

1. Each of RD[31:28] has been sampled low and sampled high by rising edges of the corresponding RCLK[31:28] inputs, or
2. Each of RD[31:28] has been sampled low and sampled high by rising edges of the RMVCK[3] input, or
3. RD[28] has been sampled low and sampled high by rising edges of the RMV8DC input.

RLGA[7] is set low when this register is read.

TLGA[0]:

The transmit link group #0 active bit (TLGA[0]) monitors for low to high transitions on the TCLK[3:0] & TMVCK[0] inputs. TLGA[0] is set high when either:

1. Rising edges have been observed on all four TCLK[3:0] inputs, or
2. A rising edge has been observed on the TMVCK[0] input.

TLGA[0] is set low when this register is read.

TLGA[1]:

The transmit link group #1 active bit (TLGA[1]) monitors for low to high transitions on the TCLK[7:4] & TMVCK[1] inputs. TLGA[1] is set high when either:

1. Rising edges have been observed on all four TCLK[7:4] inputs, or
2. A rising edge has been observed on the TMVCK[1] input.

TLGA[1] is set low when this register is read.

TLGA[2]:

The transmit link group #2 active bit (TLGA[2]) monitors for low to high transitions on the TCLK[11:8] & TMVCK[2] inputs. TLGA[2] is set high when either:

1. Rising edges have been observed on all four TCLK[11:8] inputs, or
2. A rising edge has been observed on the TMVCK[2] input.

TLGA[2] is set low when this register is read.

TLGA[3]:

The transmit link group #3 active bit (TLGA[3]) monitors for low to high transitions on the TCLK[15:12] & TMVCK[3] inputs. TLGA[3] is set high when either:

1. Rising edges have been observed on all four TCLK[15:12] inputs, or
2. A rising edge has been observed on the TMVCK[3] input.

TLGA[3] is set low when this register is read.

TLGA[4]:

The transmit link group #4 active bit (TLGA[4]) monitors for low to high transitions on the TCLK[19:16] inputs. TLGA[4] is set high when rising edges have been observed on all four TCLK[19:16] inputs, and is set low when this register is read.

TLGA[5]:

The transmit link group #5 active bit (TLGA[5]) monitors for low to high transitions on the TCLK[23:20] inputs. TLGA[5] is set high when rising edges have been observed on all four TCLK[23:20] inputs, and is set low when this register is read.

TLGA[6]:

The transmit link group #6 active bit (TLGA[6]) monitors for low to high transitions on the TCLK[27:24] inputs. TLGA[6] is set high when rising edges have been observed on all four TCLK[27:24] inputs, and is set low when this register is read.

TLGA[7]:

The transmit link group #7 active bit (TLGA[7]) monitors for low to high transitions on the TCLK[31:28] & TMV8DC inputs. TLGA[7] is set high when either:

1. Rising edges have been observed on all four TCLK[31:28] inputs, or
2. A rising edge has been observed on the TMV8DC input.

TLGA[7] is set low when this register is read.

Register 0x014 : FREEDM-32A256 Master Line Loopback #1

Bit	Type	Function	Default
Bit 15	R/W	LLBEN[15]	0
Bit 14	R/W	LLBEN[14]	0
Bit 13	R/W	LLBEN[13]	0
Bit 12	R/W	LLBEN[12]	0
Bit 11	R/W	LLBEN[11]	0
Bit 10	R/W	LLBEN[10]	0
Bit 9	R/W	LLBEN[9]	0
Bit 8	R/W	LLBEN[8]	0
Bit 7	R/W	LLBEN[7]	0
Bit 6	R/W	LLBEN[6]	0
Bit 5	R/W	LLBEN[5]	0
Bit 4	R/W	LLBEN[4]	0
Bit 3	R/W	LLBEN[3]	0
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 links #0 to #15.

LLBEN[15:0]:

The line loopback enable bits (LLBEN[15:0]) controls line loopback for links #15 to #0. When links #0 through #15 are configured for channelised T1/J1/E1 or unchannelised traffic and 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. Line loopback is not supported for H-MVIP traffic.

Register 0x018 : FREEDM-32A256 Master Line Loopback #2

Bit	Type	Function	Default
Bit 15	R/W	LLBEN[31]	0
Bit 14	R/W	LLBEN[30]	0
Bit 13	R/W	LLBEN[29]	0
Bit 12	R/W	LLBEN[28]	0
Bit 11	R/W	LLBEN[27]	0
Bit 10	R/W	LLBEN[26]	0
Bit 9	R/W	LLBEN[25]	0
Bit 8	R/W	LLBEN[24]	0
Bit 7	R/W	LLBEN[23]	0
Bit 6	R/W	LLBEN[22]	0
Bit 5	R/W	LLBEN[21]	0
Bit 4	R/W	LLBEN[20]	0
Bit 3	R/W	LLBEN[19]	0
Bit 2	R/W	LLBEN[18]	0
Bit 1	R/W	LLBEN[17]	0
Bit 0	R/W	LLBEN[16]	0

This register controls line loopback for links #16 to #31.

LLBEN[31:16]:

The line loopback enable bits (LLBEN[31:16]) controls line loopback for links #31 to #16. When links #16 through #31 are configured for channelised T1/J1/E1 or unchannelised traffic and 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. Line loopback is not supported for H-MVIP traffic.

Register 0x01C : FREEDM-32A256 Reserved

Bit	Type	Function	Default
Bit 15 to Bit 1		Unused	XXXXH
Bit 0	R/W	Reserved	0

Reserved:

The reserved bit must be set low for correct operation of the FREEDM-32A256 device.

Register 0x020 : FREEDM-32A256 Master BERT Control

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

This register controls the bit error rate testing of the receive and transmit links. Bit error rate testing is not supported for links configured for H-MVIP traffic.

RBSEL[4:0]:

The receive bit error rates testing link select bits (RBSEL[4:0]) controls the source of data on the RBD and RBCLK outputs when receive bit error rate testing is enabled (RBEN set high). RBSEL[4:0] is a binary number that selects a receive link configured for non H-MVIP traffic (RD[31:0]/RCLK[31:0]) to be the source of data for RBD and RBCLK outputs. RBSEL[4:0] is ignored when RBEN is set low. RBSEL[4:0] cannot select a link configured for H-MVIP traffic.

RBEN:

The receive bit error rates testing link enable bit (RBEN) controls the receive bit error rate testing port. When RBEN is set high, RBSEL[4:0] is a binary number that selects a receive link configured for non H-MVIP traffic (RD[31:0]/RCLK[31:0]) to be the source of data for RBD and RBCLK outputs. When RBEN is set low, RBD and RBCLK are held tristated.

TBSEL[4:0]:

The transmit bit error rates testing link select bits (TBSEL[4:0]) controls the over-writing of transmit data on TD[31:0] by data on TBD when transmit bit error rate testing is enabled (TBEN set high) and the selected link is not in line loopback (LLBEN[n] set low). TBSEL[4:0] is a binary number that selects a transmit link configured for non H-MVIP traffic (TD[31:0]/TCLK[31:0]) to carry the data on TBD. The TBCLK output is a buffered version of the selected one of TCLK[31:0]. TBSEL[4:0] is ignored when TBEN is set low. TBSEL[4:0] cannot select a link configured for H-MVIP traffic.

TBEN:

The transmit bit error rates testing link enable bit (TBEN) controls the transmit bit error rate testing port. When TBEN is set high and the associated link in not in line loopback (LLBEN set low), TBSEL[4:0] is a binary number that selects a transmit link data configured for non H-MVIP traffic (TD[31:0]) to carry the data on TBD and selects a transmit link clock (TCLK[31:0]) as the source of TBCLK. When TBEN is set low, all transmit links are processed normally and TBCLK is held tristated.

Register 0x024 : FREEDM-32A256 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 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	Reserved	0
Bit 11	R/W	Reserved	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 an H-MVIP link, time-slots 0 to 31 are valid. For unchannelised links, only time-slot 0 is valid.

LINK[4:0]:

The indirect link number bits (LINK[4:0]) select amongst the 32 receive links to be configured or interrogated in the indirect access.

Reserved:

The reserved bits must be set low for correct operation of the FREEDM-32A256 device.

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[4:0] 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[7:0] 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[7:0] 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	Reserved	0
Bit 8	R/W	Reserved	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[7:0]:

The indirect data bits (CHAN[7:0]) 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[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 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[7:0]. 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[7:0]. 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 0x108 : RCAS Framing Bit Threshold

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

This register contains the threshold used by the clock activity monitor to detect for framing bits/bytes.

FTHRES[6:0]:

The framing bit threshold bits (FTHRES[6:0]) contains the threshold used by the clock activity monitor to detect for the presence of framing bits. A counter in the clock activity monitor of each receive link increments on each rising edge of SYSCLK and is cleared, when the BSYNC bit of that link is set low, by each rising edge of the corresponding RCLK[n]. When the BSYNC bit of that link is set high, the counter is cleared at every fourth rising edge of the corresponding RCLK[n]. When the counter exceeds the threshold given by FTHRES[6:0], a framing bit/byte has been detected.

FTHRES[6:0] should be set as a function of the SYSCLK period and the expected gapping width of RCLK[n] during data bits and during framing bits/bytes. Legal range of FTHRESH[6:0] is ‘b0000001 to ‘b1111110.

Note: For operation with T1/J1 links and SYSCLK = 45 MHz, FTHRESH[6:0] should be set to ‘b0100101’. The default value of this register reflects this mode of 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	Reserved	0
Bit 8	R/W	Reserved	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[7:0]:

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

CHDIS:

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

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

Bit	Type	Function	Default
Bit 15 to Bit 5		Unused	XXXH
Bit 4	R/W	BSYNC	0
Bit 3		Unused	X
Bit 2	R/W	MODE[2]	0
Bit 1	R/W	MODE[1]	0
Bit 0	R/W	MODE[0]	0

This register configures operational modes of receive links #0 to #2.

MODE[2:0]:

The mode select bits (MODE[2:0]) configures the corresponding receive link. Table 9 details this procedure. When link 4m ($0 \leq m \leq 7$) is configured for operation in 8.192 Mbps H-MVIP mode (MODE[2:0]="111"), data cannot be received on links 4m+1, 4m+2 and 4m+3. However, links 4m+1, 4m+2 and 4m+3 must be configured for 8.192 Mbps H-MVIP mode for correct operation of the RCAS256. From a channel assignment point of view in the RCAS256 (Registers 0x100, 0x104), time-slots 0 through 31 of the H-MVIP link are treated as time-slots 0 through 31 of link 4m, time-slots 32 through 63 of the H-MVIP link are treated as time-slots 0 through 31 of link 4m+1, time-slots 64 through 95 of the H-MVIP link are treated as time-slots 0 through 31 of link 4m+2 and time-slots 96 through 127 of the H-MVIP link are treated as time-slots 0 through 31 of link 4m+3.

Table 9 – Receive Links #0 to #2 Configuration

MODE[2:0]	Link Configuration
000	Unchannelised
001	Channelised T1/J1 (24 time slots labeled 1-24)
010	Channelised E1 (31 time slots labeled 1-31)
011	2 Mbps H-MVIP (32 time slots labeled 0-31)
100	Reserved
101	Reserved
110	Reserved
111	8 Mbps H-MVIP (128 time slots mapped to time-slots 0 through 31 of links 4m, 4m+1, 4m+2 and 4m+3)

BSYNC:

The byte synchronization enable bit (BSYNC) controls the interpretation of gaps in RCLK[n] when link #n is in unchannelised mode (MODE[2:0]="000"). When BSYNC is set high, the data bit on RD[n] clocked in by the first rising edge of RCLK[n] after an extended quiescent period is considered to be the most significant bit of a data byte. When BSYNC is set low, gaps in RCLK[n] carry no special significance. BSYNC is ignored when MODE[2:0]≠"000".

Register 0x18C – 0x1FC : RCAS Links #3 to #31 Configuration

Bit	Type	Function	Default
Bit 15 to Bit 3		Unused	XXXXH
Bit 2	R/W	MODE[2]	0
Bit 1	R/W	MODE[1]	0
Bit 0	R/W	MODE[0]	0

This register configures operational modes of receive links #3 to #31.

MODE[2:0]:

The mode select bits (MODE[2:0]) configures the corresponding receive link. Table 10 details this procedure. When link 4m ($0 \leq m \leq 7$) is configured for operation in 8.192 Mbps H-MVIP mode (MODE[2:0]="111"), data cannot be received on links 4m+1, 4m+2 and 4m+3. However, links 4m+1, 4m+2 and 4m+3 must be configured for 8.192 Mbps H-MVIP mode for correct operation of the RCAS256. From a channel assignment point of view in the RCAS256 (Registers 0x100, 0x104), time-slots 0 through 31 of the H-MVIP link are treated as time-slots 0 through 31 of link 4m, time-slots 32 through 63 of the H-MVIP link are treated as time-slots 0 through 31 of link 4m+1, time-slots 64 through 95 of the H-MVIP link are treated as time-slots 0 through 31 of link 4m+2 and time-slots 96 through 127 of the H-MVIP link are treated as time-slots 0 through 31 of link 4m+3.

Table 10 – Receive Links #3 to #31 Configuration

MODE[2:0]	Link Configuration
000	Unchannelised
001	Channelised T1/J1 (24 time slots labeled 1-24)
010	Channelised E1 (31 time slots labeled 1-31)
011	2 Mbps H-MVIP (32 time slots labeled 0-31)
100	Reserved
101	Reserved
110	Reserved
111	8 Mbps H-MVIP (128 time slots mapped to time-slots 0 through 31 of links 4m, 4m+1, 4m+2 and 4m+3)

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	Reserved	0
Bit 8	R/W	Reserved	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[7:0]:

The indirect channel number bits (CHAN[7:0]) 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-32A256 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[7:0]. When PROV is set low, the HDLC processor will ignore data on the channel specified by CHAN[7:0]. 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-32A256 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 11 – 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 RCAS256 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 RAPI256 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 RAPI256 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 corresponding link RD[n]. 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-32A256 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-32A256 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-32A256 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	Reserved	0
Bit 8	R/W	Reserved	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[7:0]:

The indirect channel number bits (CHAN[7:0]) 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-32A256 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 12 – 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 TCAS256 block. When

PROV is set low, the HDLC processor will ignore requests from the TCAS256 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-32A256 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 corresponding transmit link (TD[n]). 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 TAPI256 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 TAPI256 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 13 – 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 TAPI256 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 TAPI256 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 TAPI256 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 TAPI256 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 14 – 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)

LEVEL[3:0]	Starving Trigger Level	Start Transmission Level (TRANS=0)	Start Transmission Level (TRANS=1)
1000	32 Blocks (512 bytes free)	24 Blocks (384 bytes free)	16 Blocks (256 bytes free)
1001	48 Blocks (768 bytes free)	32 Blocks (512 bytes free)	24 Blocks (384 bytes free)
1010	64 Blocks (1 Kbytes free)	48 Blocks (768 bytes free)	32 Blocks (512 bytes free)
1011	96 Blocks (1.5 Kbytes free)	64 Blocks (1 Kbytes free)	48 Blocks (768 bytes free)
1100	192 Blocks (3 Kbytes free)	128 Blocks (2 Kbytes free)	96 Blocks (1.5 Kbytes free)
1101	384 Blocks (6 Kbytes free)	256 Blocks (4 Kbytes free)	192 Blocks (2 Kbytes free)
1110	768 Blocks (12 Kbytes free)	512 Blocks (8 Kbytes free)	384 Blocks (4 Kbytes free)
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-32A256 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-32A256 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-32A256 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	Reserved	0
Bit 11	R/W	Reserved	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 a H-MVIP link, time-slots 0 to 31 are valid. For unchannelised links, only time-slot 0 is valid.

LINK[4:0]:

The indirect link number bits (LINK[4:0]) select amongst the 32 transmit links to be configured or interrogated in the indirect access.

Reserved:

The reserved bits must be set low for correct operation of the FREEDM-32A256 device.

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[4:0] 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[7:0] 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[7:0] 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	Reserved	0
Bit 8	R/W	Reserved	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[7:0]:

The indirect data bits (CHAN[7:0]) 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[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 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[7:0]. 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 0x408 : TCAS Framing Bit Threshold

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

This register contains the threshold used by the clock activity monitors to detect for framing bits/bytes.

FTHRES[6:0]:

The framing bit threshold bits (FTHRES[6:0]) contains the threshold used by the clock activity monitor to detect for the presence of framing bits. A counter in the clock activity monitor of each receive link increments on each rising edge of SYSCLK and is cleared, when the BSYNC bit of that link is set low, by each rising edge of the corresponding TCLK[n]. When the BSYNC bit of that link is set high, the counter is cleared at every fourth rising edge of the corresponding TCLK[n]. When the counter exceeds the threshold given by FTHRES[6:0], a framing bit/byte has been detected.

FTHRES[6:0] should be set as a function of the SYSCLK period and the expected gapping width of TCLK[n] during data bits and during framing bits/bytes. Legal range of FTHRES[6:0] is ‘b0000001 to ‘b1111110.

Note: For operation with T1/J1 links and SYSCLK = 45 MHz, FTHRESH[6:0] should be set to ‘b0100101’. The default value of this register reflects this mode of 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	Reserved	0
Bit 8	R/W	Reserved	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[7:0]:

The disable channel number bits (DCHAN[7:0]) selects the channel to be disabled. When CHDIS is set high, the channel specified by DCHAN[7:0] 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[7:0] operates normally.

CHDIS:

The channel disable bit (CHDIS) controls the disabling of the channels specified by DCHAN[7:0]. When CHDIS is set high, the channel selected by DCHAN[7:0] 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[7:0] operates normally.

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

Bit	Type	Function	Default
Bit 15 to Bit 5		Unused	XXXH
Bit 4	R/W	BSYNC	0
Bit 3		Unused	X
Bit 2	R/W	MODE[2]	0
Bit 1	R/W	MODE[1]	0
Bit 0	R/W	MODE[0]	0

This register configures operational modes of transmit links #0 to #2.

MODE[2:0]:

The mode select bits (MODE[2:0]) configures the corresponding transmit link. Table 15 details this procedure. When link 4m ($0 \leq m \leq 7$) is configured for operation in 8.192 Mbps H-MVIP mode, links 4m+1, 4m+2 and 4m+3 are driven with constant ones. However, links 4m+1, 4m+2 and 4m+3 must be configured for 8.192 Mbps H-MVIP mode for correct operation of the TCAS256. From a channel assignment point of view in the TCAS256 (Registers 0x400, 0x404), time-slots 0 through 31 of link 4m are mapped to time-slots 0 through 31 of the H-MVIP link, time-slots 0 through 31 of link 4m+1 are mapped to time-slots 32 through 63 of the H-MVIP link, time-slots 0 through 31 of link 4m+2 are mapped to time-slots 64 through 95 of the H-MVIP link and time-slots 0 through 31 of link 4m+3 are mapped to time-slots 96 through 127 of the H-MVIP link.

Table 15 – Transmit Links #0 to #2 Configuration

MODE[2:0]	Link Configuration
000	Unchannelised
001	Channelised T1/J1 (24 time slots labeled 1-24)
010	Channelised E1 (31 time slots labeled 1-31)
011	2 Mbps H-MVIP (32 time slots labeled 0-31)
100	Reserved
101	Reserved
110	Reserved
111	8 Mbps H-MVIP (128 time slots mapped to time-slots 0 through 31 of links 4m, 4m+1, 4m+2 and 4m+3)

BSYNC:

The byte synchronization enable bit (BSYNC) controls the interpretation of gaps in TCLK[n] when link #n is in unchannelised mode (MODE[2:0]="000"). When BSYNC is set high, the data bit on TD[n] clocked in by a downstream device on the first rising edge of TCLK[n] after an extended quiescent period is considered to be the most significant bit of a data byte. When BSYNC is set quiescent, gaps in TCLK[n] carry no special significance. BSYNC is ignored when MODE[2:0]≠"000".

Register 0x48C – 0x4FC : TCAS Links #3 to #31 Configuration

Bit	Type	Function	Default
Bit 15 to Bit 3		Unused	XXXXH
Bit 2	R/W	MODE[2]	0
Bit 1	R/W	MODE[1]	0
Bit 0	R/W	MODE[0]	0

This register configures operational modes of transmit links #3 to #31.

MODE[2:0]:

The mode select bits (MODE[2:0]) configures the corresponding transmit link. Table 16 details this procedure. When link 4m ($0 \leq m \leq 7$) is configured for operation in 8.192 Mbps H-MVIP mode, links 4m+1, 4m+2 and 4m+3 are driven with constant ones. However, links 4m+1, 4m+2 and 4m+3 must be configured for 8.192 Mbps H-MVIP mode for correct operation of the TCAS256. From a channel assignment point of view in the TCAS256 (Registers 0x100, 0x104), time-slots 0 through 31 of link 4m are mapped to time-slots 0 through 31 of the H-MVIP link, time-slots 0 through 31 of link 4m+1 are mapped to time-slots 32 through 63 of the H-MVIP link, time-slots 0 through 31 of link 4m+2 are mapped to time-slots 64 through 95 of the H-MVIP link and time-slots 0 through 31 of link 4m+3 are mapped to time-slots 96 through 127 of the H-MVIP link.

Table 16 – Transmit Links #3 to #31 Configuration

MODE[2:0]	Link Configuration
000	Unchannelised
001	Channelised T1/J1 (24 time slots labeled 1-24)
010	Channelised E1 (31 time slots labeled 1-31)
011	2 Mbps H-MVIP (32 time slots labeled 0-31)
100	Reserved
101	Reserved
110	Reserved
111	8 Mbps H-MVIP (128 time slots mapped to time-slots 0 through 31 of links 4m, 4m+1, 4m+2 and 4m+3)

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-32A256 Master Clock / BERT 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-32A256 Master Clock / BERT 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-32A256 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-32A256 Master Clock / BERT 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-32A256 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-32A256 Master Clock / BERT 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 RAPI256.

BADDR[2:0]:

The base address bits (BADDR[2:0]) configure the address space occupied by the FREEDM-32A256 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-32A256 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-32A256 to respond to receive polling and device selection when BADDR[2:0] = ‘111’. When ALL1ENB is zero, the FREEDM-32A256 responds to receive polling and device selection when BADDR[2:0] = RXADDR[2:0] = ‘111’. When ALL1ENB is one, the FREEDM-32A256 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-32A256 device.

STATEN:

The RAPI256 Status Enable bit (STATEN) enables the RAPI256 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 RAPI256 overwrites RXDATA[7:0] of the final word of an errored packet with status information for that packet. When STATEN is set low, the RAPI256 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 RAPI256 Enable bit (ENABLE) enables normal operation of the RAPI256. When ENABLE is set low, the RAPI256 will not transfer data from the RHDL256 into its internal FIFOs. When ENABLE is set high, the RAPI256 operates normally.

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 TAPI256.

BADDR[2:0]:

The base address bits (BADDR[2:0]) configure the address space occupied by the FREEDM-32A256 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-32A256 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-32A256 device.

ALL1ENB:

The All Ones Enable bit (ALL1ENB) permits the FREEDM-32A256 to respond to transmit polling and device selection when BADDR[2:0] = ‘111’. When ALL1ENB is zero, the FREEDM-32A256 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-32A256 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-32A256 device.

ENABLE:

The TAPI256 Enable bit (ENABLE) enables normal operation of the TAPI256. When ENABLE is set low, the TAPI256 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 TAPI256 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	Reserved	0
Bit 8	R/W	Reserved	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 TAPI256 channel provisioning RAM. Writing to this register triggers an indirect channel register access.

CHAN[7:0]:

The indirect channel number bits (CHAN[7:0]) 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[7:0]. Writing a logic 1 to RWB triggers a query of the channel specified by CHAN[7:0].

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 TAPI256 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 TAPI256 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 TAPI256 channel provision RAM after an indirect read operation has completed. The provision enable flag to be written to the TAPI256 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[7:0] is provisioned. When PROV is set low, the channel indicated by CHAN[7:0] is unprovisioned. PROV reflects the value written until the completion of a subsequent indirect read operation.

10 TEST FEATURES DESCRIPTION

The FREEDM-32A256 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-32A256. Production testing is enabled by asserting the PMCTEST pin. During production tests, FREEDM-32A256 registers are selected by the TA[12:0] pins. The address of a register on TA[12:0] is identical to the offset of that register when production testing is disabled (PMCTEST low). 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 17 – Test Mode Register Memory Map

Address TA[12:0]	Register
0x0000 - 0x07FE	Normal Mode Registers
0x0800 - 0x10FE	Reserved
0x1100 - 0x11FE	RCAS256 Test Registers
0x1200 - 0x123E	RHDL256 Test Registers
0x1240 - 0x137E	Reserved
0x1380 - 0x13BE	THDL256 Test Registers
0x13C0 - 0x13FE	Reserved
0x1400 - 0x14FE	TCAS256 Test Registers
0x1500 - 0x151E	PMON Test Registers
0x1520 - 0x157E	Reserved
0x1580 - 0x15BE	RAPI256 Test Registers
0x15C0 - 0x15FE	Reserved
0x1600 - 0x163E	TAPI256 Test Registers
0x1640 - 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-32A256 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 18 – 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 - 7383H

Manufacturer's identification code - 0CDH

Device identification - 273830CDH

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 273830CDH. The cells are arranged as follows:

Table 19 – Boundary Scan Chain

Pin/ Enable	Register Bit	Cell Type	Device I.D.
RBCLK_OEN	0	OUT_CELL	-
RBCLK	1	OUT_CELL	-
RBD_OEN	2	OUT_CELL	-
RBD	3	OUT_CELL	-
TMV8DC	4	IN_CELL	-
TFP8B	5	IN_CELL	-
TMV8FPC	6	IN_CELL	-
TFPB[0]	7	IN_CELL	-
TMVCK[0]	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	-
TD_OEN[3]	18	OUT_CELL	-
TD[3]	19	OUT_CELL	-
TCLK[3]	20	IN_CELL	-
TD_OEN[4]	21	OUT_CELL	-
TD[4]	22	OUT_CELL	-
TCLK[4]	23	IN_CELL	-
TD_OEN[5]	24	OUT_CELL	-
TD[5]	25	OUT_CELL	-

Pin/ Enable	Register Bit	Cell Type	Device I.D.
TCLK[5]	26	IN_CELL	-
TD_OEN[6]	27	OUT_CELL	-
TD[6]	28	OUT_CELL	-
TCLK[6]	29	IN_CELL	-
TD_OEN[7]	30	OUT_CELL	-
TD[7]	31	OUT_CELL	-
TCLK[7]	32	IN_CELL	-
TFPB[1]	33	IN_CELL	-
TMVCK[1]	34	IN_CELL	-
TD_OEN[8]	35	OUT_CELL	-
TD[8]	36	OUT_CELL	-
TCLK[8]	37	IN_CELL	-
TD_OEN[9]	38	OUT_CELL	-
TD[9]	39	OUT_CELL	-
TCLK[9]	40	IN_CELL	-
TD_OEN[10]	41	OUT_CELL	-
TD[10]	42	OUT_CELL	-
TCLK[10]	43	IN_CELL	-
TD_OEN[11]	44	OUT_CELL	-
TD[11]	45	OUT_CELL	-
TCLK[11]	46	IN_CELL	-
TD_OEN[12]	47	OUT_CELL	-
TD[12]	48	OUT_CELL	-
TCLK[12]	49	IN_CELL	-
TD_OEN[13]	50	OUT_CELL	-
TD[13]	51	OUT_CELL	-
TCLK[13]	52	IN_CELL	-

Pin/ Enable	Register Bit	Cell Type	Device I.D.
TD_OEN[14]	53	OUT_CELL	-
TD[14]	54	OUT_CELL	-
TCLK[14]	55	IN_CELL	-
TD_OEN[15]	56	OUT_CELL	-
TD[15]	57	OUT_CELL	-
TCLK[15]	58	IN_CELL	-
TFPB[2]	59	IN_CELL	-
TMVCK[2]	60	IN_CELL	-
TD_OEN[16]	61	OUT_CELL	-
TD[16]	62	IO_CELL	-
TCLK[16]	63	IN_CELL	-
TD_OEN[17]	64	OUT_CELL	-
TD[17]	65	IO_CELL	-
TCLK[17]	66	IN_CELL	-
TD_OEN[18]	67	OUT_CELL	-
TD[18]	68	IO_CELL	-
TCLK[18]	69	IN_CELL	-
TD_OEN[19]	70	OUT_CELL	-
TD[19]	71	IO_CELL	-
TCLK[19]	72	IN_CELL	-
TD_OEN[20]	73	OUT_CELL	-
TD[20]	74	IO_CELL	-
TCLK[20]	75	IN_CELL	-
TD_OEN[21]	76	OUT_CELL	-
TD[21]	77	IO_CELL	-
TCLK[21]	78	IN_CELL	-
TD_OEN[22]	79	OUT_CELL	-

Pin/ Enable	Register Bit	Cell Type	Device I.D.
TD[22]	80	IO_CELL	-
TCLK[22]	81	IN_CELL	-
TD_OEN[23]	82	OUT_CELL	-
TD[23]	83	IO_CELL	-
TCLK[23]	84	IN_CELL	-
TFPB[3]	85	IN_CELL	-
TMVCK[3]	86	IN_CELL	-
TD_OEN[24]	87	OUT_CELL	-
TD[24]	88	IO_CELL	-
TCLK[24]	89	IN_CELL	-
TD_OEN[25]	90	OUT_CELL	-
TD[25]	91	IO_CELL	-
TCLK[25]	92	IN_CELL	-
TD_OEN[26]	93	OUT_CELL	-
TD[26]	94	IO_CELL	-
TCLK[26]	95	IN_CELL	-
TD_OEN[27]	96	OUT_CELL	-
TD[27]	97	IO_CELL	-
TCLK[27]	98	IN_CELL	-
TD_OEN[28]	99	OUT_CELL	-
TD[28]	100	IO_CELL	-
TCLK[28]	101	IN_CELL	-
TD_OEN[29]	102	OUT_CELL	-
TD[29]	103	IO_CELL	-
TCLK[29]	104	IN_CELL	-
TD_OEN[30]	105	OUT_CELL	-
TD[30]	106	IO_CELL	-

Pin/ Enable	Register Bit	Cell Type	Device I.D.
TCLK[30]	107	IN_CELL	-
TD_OEN[31]	108	OUT_CELL	-
TD[31]	109	IO_CELL	-
TCLK[31]	110	IN_CELL	-
TBD	111	IN_CELL	-
TBCLK_OEN	112	OUT_CELL	-
TBCLK	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	-

Pin/ Enable	Register Bit	Cell Type	Device I.D.
RENB	161	IN_CELL	-
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	-

Pin/ Enable	Register Bit	Cell Type	Device I.D.
TXDATA_OEN[7]	188	OUT_CELL	-
TXDATA[7]	189	IO_CELL	-
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	-

Pin/ Enable	Register Bit	Cell Type	Device I.D.
Unconnected	215	OUT_CELL	-
Unconnected	216	OUT_CELL	-
TXADDR[3]	217	IN_CELL	-
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	-

Pin/ Enable	Register Bit	Cell Type	Device I.D.
D_OEN[1]	242	OUT_CELL	-
D[1]	243	IO_CELL	-
D_OEN[2]	244	OUT_CELL	-
D[2]	245	IO_CELL	-
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	-

Pin/ Enable	Register Bit	Cell Type	Device I.D.
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	-
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	-
RCLK[31]	288	IN_CELL	-
RD[31]	289	IN_CELL	-
RCLK[30]	290	IN_CELL	-
RD[30]	291	IN_CELL	-
RCLK[29]	292	IN_CELL	-
RD[29]	293	IN_CELL	-
RCLK[28]	294	IN_CELL	-
RD[28]	295	IN_CELL	-

Pin/ Enable	Register Bit	Cell Type	Device I.D.
RCLK[27]	296	IN_CELL	-
RD[27]	297	IN_CELL	-
RCLK[26]	298	IN_CELL	-
RD[26]	299	IN_CELL	-
RCLK[25]	300	IN_CELL	-
RD[25]	301	IN_CELL	-
RCLK[24]	302	IN_CELL	-
RD[24]	303	IN_CELL	-
RMVCK[3]	304	IN_CELL	-
RFPB[3]	305	IN_CELL	-
RCLK[23]	306	IN_CELL	-
RD[23]	307	IN_CELL	-
RCLK[22]	308	IN_CELL	-
RD[22]	309	IN_CELL	-
RCLK[21]	310	IN_CELL	-
RD[21]	311	IN_CELL	-
RCLK[20]	312	IN_CELL	-
RD[20]	313	IN_CELL	-
RCLK[19]	314	IN_CELL	-
RD[19]	315	IN_CELL	-
RCLK[18]	316	IN_CELL	-
RD[18]	317	IN_CELL	-
RCLK[17]	318	IN_CELL	-
RD[17]	319	IN_CELL	-
RCLK[16]	320	IN_CELL	-
RD[16]	321	IN_CELL	-
RMVCK[2]	322	IN_CELL	-

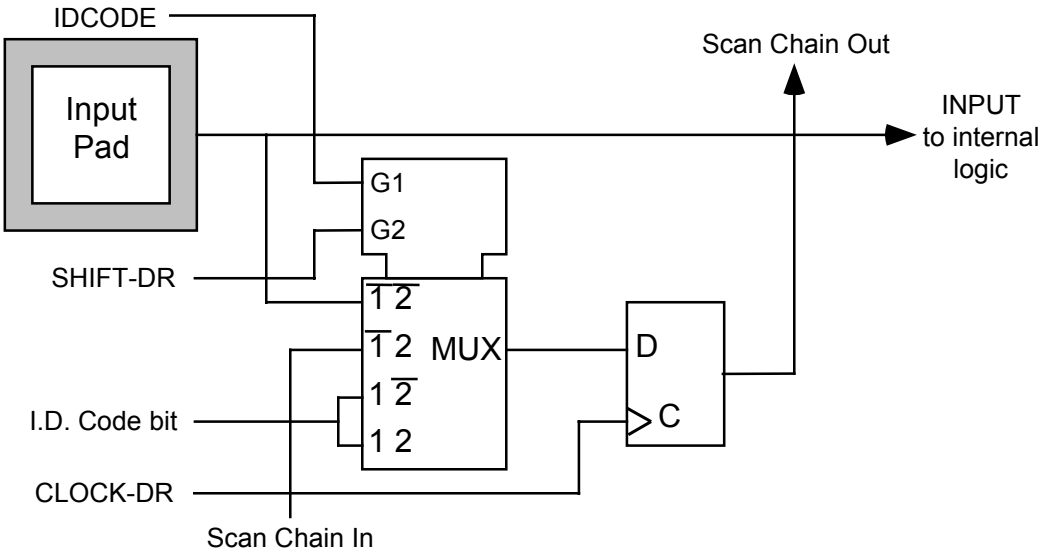
Pin/ Enable	Register Bit	Cell Type	Device I.D.
RFPB[2]	323	IN_CELL	-
RCLK[15]	324	IN_CELL	-
RD[15]	325	IN_CELL	-
RSTB	326	IN_CELL	-
RCLK[14]	327	IN_CELL	-
RD[14]	328	IN_CELL	-
RCLK[13]	329	IN_CELL	-
RD[13]	330	IN_CELL	-
RCLK[12]	331	IN_CELL	-
RD[12]	332	IN_CELL	-
RCLK[11]	333	IN_CELL	1
RD[11]	334	IN_CELL	0
RCLK[10]	335	IN_CELL	1
RD[10]	336	IN_CELL	1
RCLK[9]	337	IN_CELL	0
RD[9]	338	IN_CELL	0
RCLK[8]	339	IN_CELL	1
RD[8]	340	IN_CELL	1
RMVCK[1]	341	IN_CELL	0
RFPB[1]	342	IN_CELL	0
RCLK[7]	343	IN_CELL	0
RD[7]	344	IN_CELL	0
RCLK[6]	345	IN_CELL	1
RD[6]	346	IN_CELL	1
SYSCLK	347	IN_CELL	0
RCLK[5]	348	IN_CELL	0
RD[5]	349	IN_CELL	0

Pin/ Enable	Register Bit	Cell Type	Device I.D.
RCLK[4]	350	IN_CELL	0
RD[4]	351	IN_CELL	0
RCLK[3]	352	IN_CELL	1
RD[3]	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
RCLK[0]	358	IN_CELL	1
RD[0]	359	IN_CELL	1
RMVCK[0]	360	IN_CELL	0
RFPB[0]	361	IN_CELL	0
RMV8FPC	362	IN_CELL	1
RFP8B	363	IN_CELL	0
RMV8DC	364	IN_CELL	0
TDO		TAP Output	-
TDI		TAP Input	-
TCK		TAP Clock	-
TMS		TAP Input	-
TRSTB		TAP Input	-

Notes:

1. RMV8DC is the first bit of the scan chain (closest to TDI).
2. Enable cell pinname_OEN, tristates pin pinname when set high.
3. 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 6 – 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 7 – Output Cell (OUT_CELL)

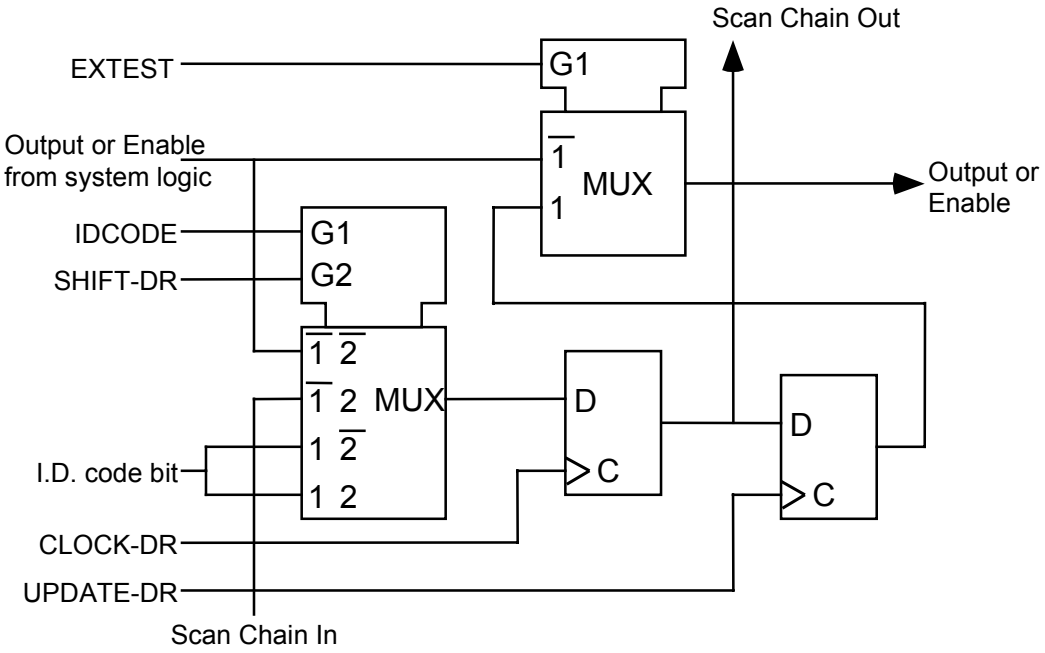


Figure 8 – Bi-directional Cell (IO_CELL)

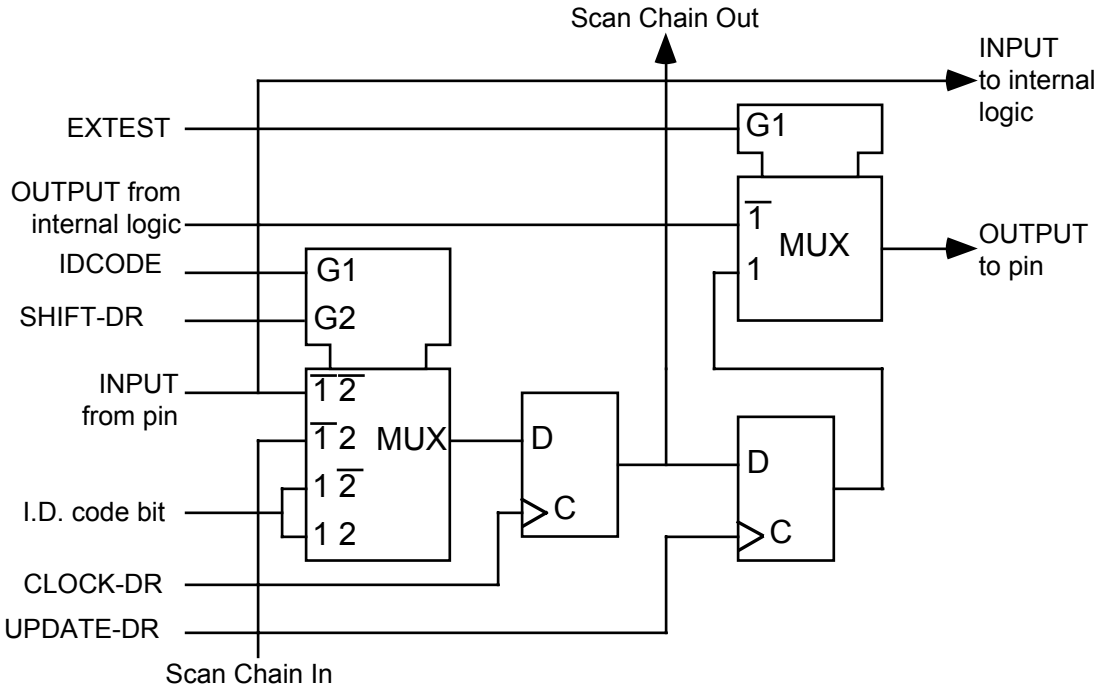
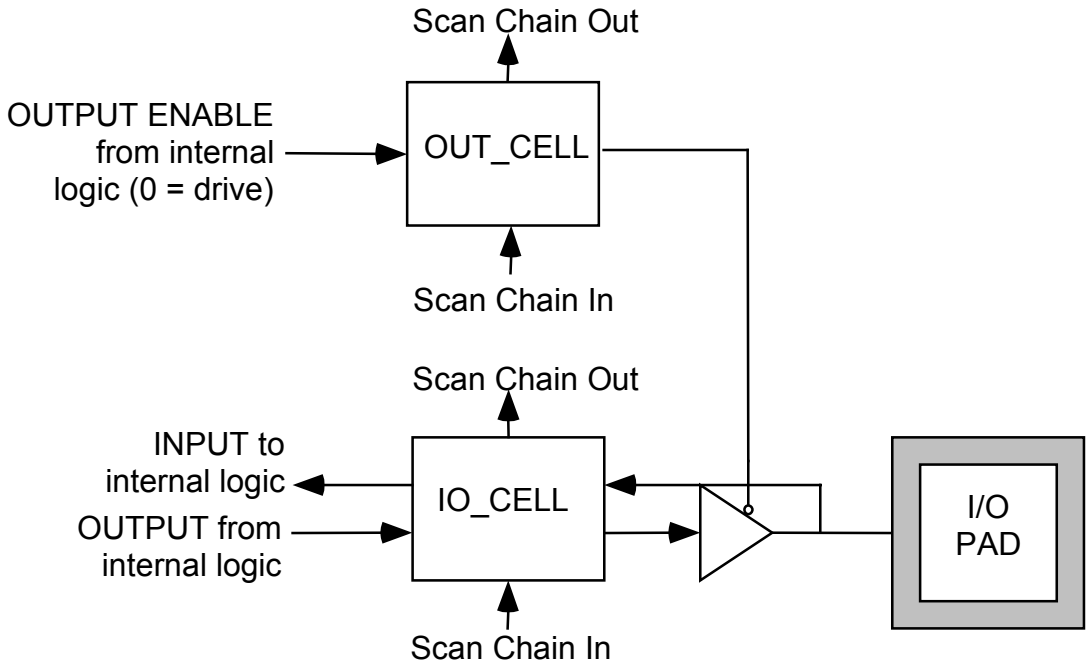


Figure 9 – Layout of Output Enable and Bi-directional Cells



11 OPERATIONS

This section presents connection details to the PM4388 TOCTL device, and operating details for the JTAG boundary scan feature.

11.1 TOCTL Connections

The required connections between the PM4388 TOCTL and the FREEDM-32A256 are shown in the following table:

Table 20 – FREEDM–TOCTL Connections

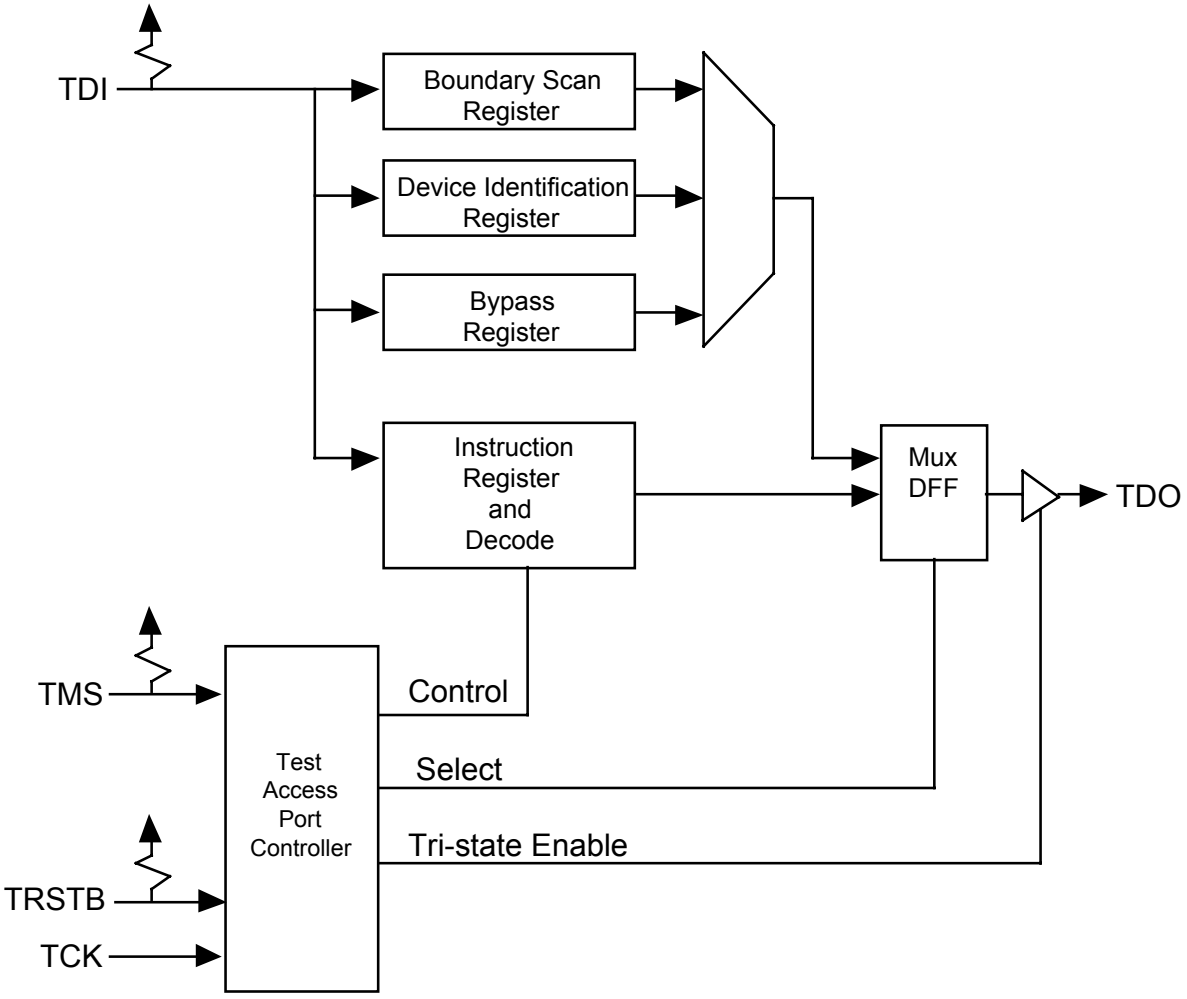
FREEDM Pin	Direction	TOCTL Pin
RCLK[n]	←	ICLK/ISIG[m]
RD[n]	←	ID[m]
n.c.	←	IFP[m]
TCLK[n]	←	EFP/RCLK/ESIG[m]
TD[n]	→	ED[m]

All 8 framers in the TOCTL should be programmed to operate in “Clock Master: NxDS0” mode in both the ingress and egress direction.

11.2 JTAG Support

The FREEDM-32A256 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 10 – 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.

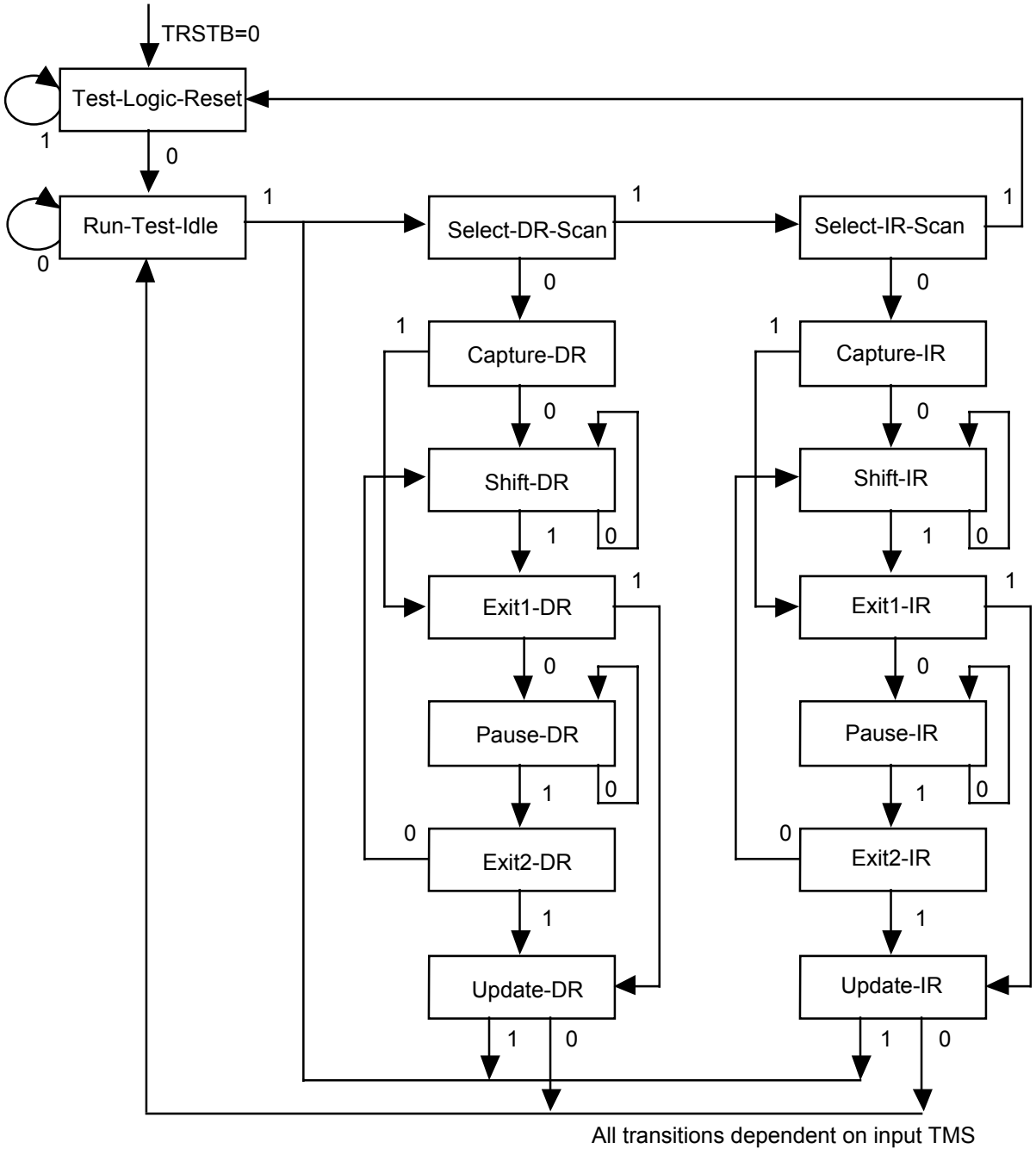
The boundary scan register allows testing of board inter-connectivity. The boundary scan register consists of a shift register placed in series with device

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 11 – 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

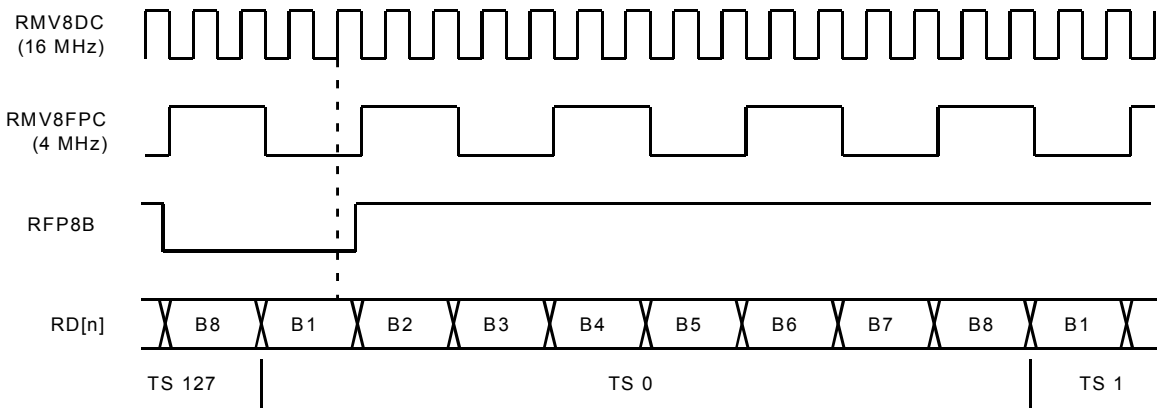
The internal test instruction is used to exercise the device's internal core logic. When this instruction is the current instruction, the boundary scan register is connected between TDI and TDO. During the Update-DR state, patterns shifted in on input, TDI are used to drive primary inputs. During the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

12 FUNCTIONAL TIMING

12.1 Receive H-MVIP Link Timing

The timing relationship of the receive data clock (RMV8DC), frame pulse clock (RMV8FPC), data (RD[n]) and frame pulse (RFP8B) signals of a link configured for 8.192 Mbps H-MVIP operation with a type 0 frame pulse is shown in Figure 12. The falling edges of each RMV8FPC are aligned to a falling edge of the corresponding RMV8DC for 8.192 Mbps H-MVIP operation. The FREEDM-32A256 samples RFP8B low on the falling edge of RMV8FPC and references this point as the start of the next frame. The FREEDM-32A256 samples the data provided on RD[n] at the ¾ point of the data bit using the rising edge of RMV8DC as indicated for bit 1 (B1) of time-slot 0 (TS 0) in Figure 12. B1 is the most significant bit and B8 is the least significant bit of each octet. Time-slots can be ignored by setting the PROV bit in the corresponding word of the receive channel provision RAM in the RCAS256 block to low.

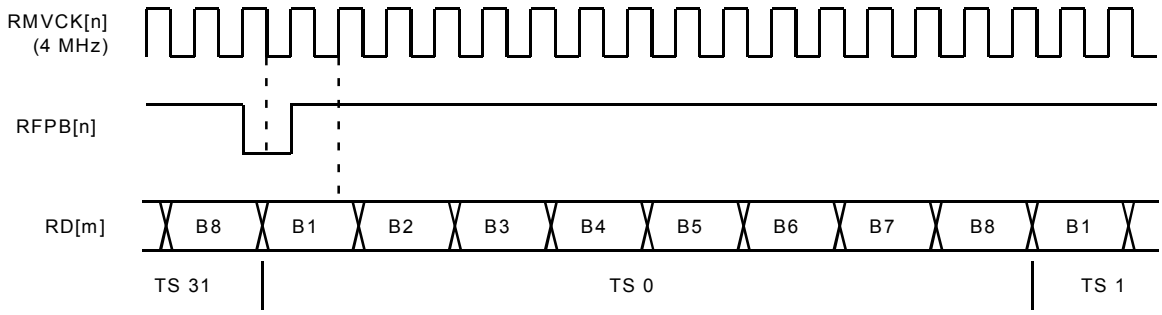
Figure 12 – Receive 8.192 Mbps H-MVIP Link Timing



The timing relationship of the receive data clock (RMVCK[n]), data (RD[m], where $8n \leq m \leq 8n+7$) and frame pulse (RFPB[n]) signals of a link configured for 2.048 Mbps H-MVIP operation with a type 0 frame pulse is shown in Figure 13. The FREEDM-32A256 samples RFPB[n] low on the falling edge of the corresponding RMVCK[n] and references this point as the start of the next frame. The FREEDM-32A256 samples the data provided on RD[m] at the ¾ point of the data bit using the rising edge of the corresponding RMVCK[n] as indicated for bit 1 (B1) of time-slot 0 (TS 0) in Figure 13. B1 is the most significant bit and B8 is the least significant bit of each octet. Time-slots can be ignored by setting the PROV

bit in the corresponding word of the receive channel provision RAM in the RCAS256 block to low.

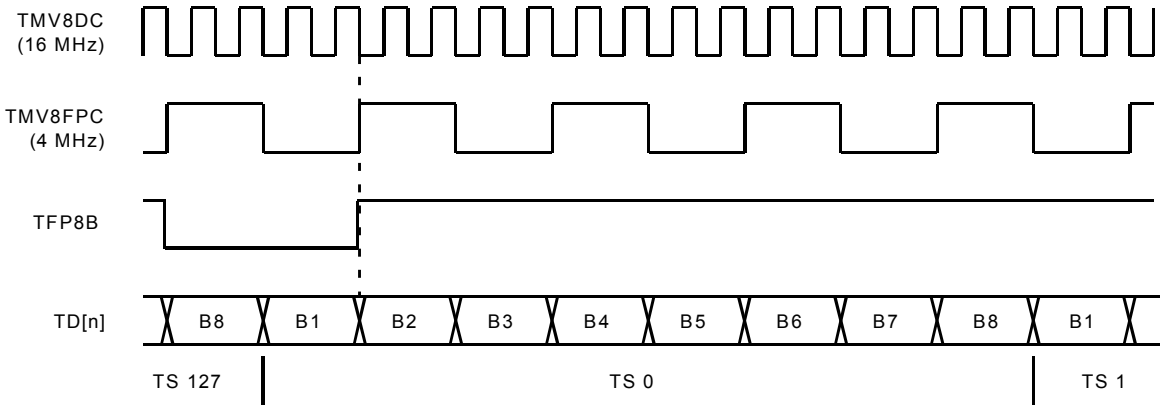
Figure 13 – Receive 2.048 Mbps H-MVIP Link Timing



12.2 Transmit H-MVIP Link Timing

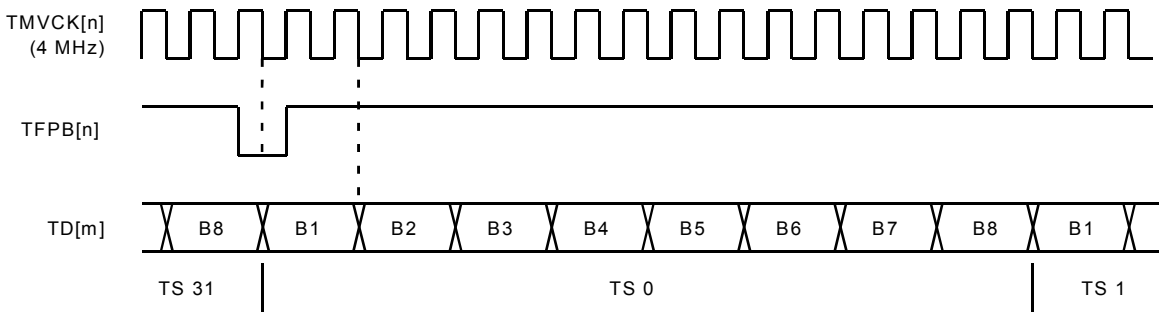
The timing relationship of the transmit data clock (TMV8DC), frame pulse clock (TMV8FPC), data (TD[n]) and frame pulse (TFP8B) signals of a link configured for 8.192 Mbps H-MVIP operation with a type 0 frame pulse is shown in Figure 14. The falling edges of each TMV8FPC are aligned to a falling edge of the corresponding TMV8DC for 8.192 Mbps H-MVIP operation. The FREEDM-32A256 samples TFP8B low on the falling edge of TMV8FPC and references this point as the start of the next frame. The FREEDM-32A256 updates the data provided on TD[n] on every second falling edge of TMV8DC as indicated for bit 2 (B2) of time-slot 0 (TS 0) in Figure 14. The first bit of the next frame is updated on TD[n] on the falling TMV8DC clock edge for which TFP8B is also sampled low. B1 is the most significant bit and B8 is the least significant bit of each octet. Time-slots that are not provisioned to belong to any channel (PROV bit in the corresponding word of the transmit channel provision RAM in the TCAS256 block set low) transmits the contents of the Idle Fill Time-slot Data register.

Figure 14 – Transmit 8.192 Mbps H-MVIP Link Timing



The timing relationship of the transmit data clock (TMVCK[n]), data (TD[m], where 8n≤m≤8n+7) and frame pulse (TFPB[n]) signals of a link configured for 2.048 Mbps H-MVIP operation with a type 0 frame pulse is shown in Figure 15. The FREEDM-32A256 samples TFPB[n] low on the falling edge of the corresponding TMVCK[n] and references this point as the start of the next frame. The FREEDM-32A256 updates the data provided on TD[m] on every second falling edge of the corresponding TMVCK[n] as indicated for bit 2 (B2) of time-slot 0 (TS 0) in Figure 15. The first bit of the next frame is updated on TD[m] on the falling TMVCK[n] clock edge for which TFPB[n] is also sampled low. B1 is the most significant bit and B8 is the least significant bit of each octet. Time-slots that are not provisioned to belong to any channel (PROV bit in the corresponding word of the transmit channel provision RAM in the TCAS256 block set low) transmits the contents of the Idle Fill Time-slot Data register.

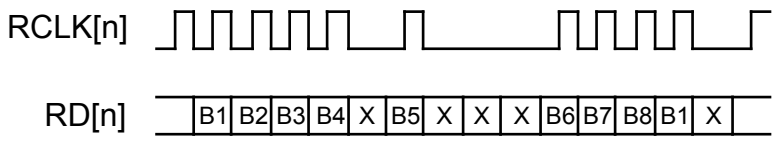
Figure 15 – Transmit 2.048 Mbps H-MVIP Link Timing



12.3 Receive non H-MVIP Link Timing

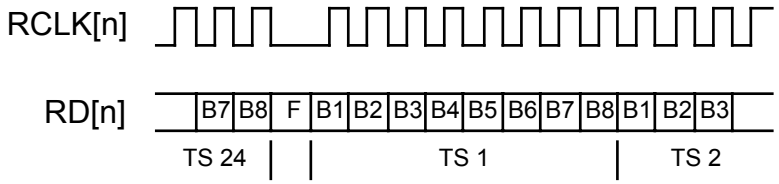
The timing relationship of the receive clock (RCLK[n]) and data (RD[n]) signals of an unchannelised link is shown in Figure 16. The receive data is viewed as a contiguous serial stream. There is no concept of time-slots in an unchannelised link. Every eight bits are grouped together into a byte with arbitrary alignment. The first bit received (B1 in Figure 16) 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-32A256 are clocked in on the rising edge of RCLK[n]. Bits that should be ignored (X in Figure 16) are squelched by holding RCLK[n] quiescent. In Figure 16, 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 16 – Unchannelised Receive Link Timing



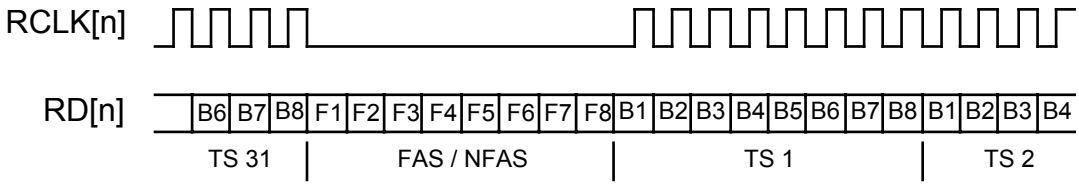
The timing relationship of the receive clock (RCLK[n]) and data (RD[n]) signals of a channelised T1/J1 link is shown in Figure 17. The receive data stream is a T1/J1 frame with a single framing bit (F in Figure 17) followed by octet bound time-slots 1 to 24. RCLK[n] is held quiescent during the framing bit. The RD[n] data bit (B1 of TS1) clocked in by the first rising edge of RCLK[n] after the framing bit is the most significant bit of time-slot 1. The RD[n] bit (B8 of TS24) clocked in by the last rising edge of RCLK[n] before the framing bit is the least significant bit of time-slot 24. In Figure 17, the quiescent period is shown to be a low level on RCLK[n]. A high level, effected by extending the high phase of bit B8 of time-slot TS24, is equally acceptable. In channelised T1/J1 mode, RCLK[n] can only be gapped during the framing bit. It must be active continuously at 1.544 MHz during all time-slot bits. Time-slots can be ignored by setting the PROV bit in the corresponding word of the receive channel provision RAM in the RCAS256 block to low.

Figure 17 – Channelised T1/J1 Receive Link Timing



The timing relationship of the receive clock (RCLK[n]) and data (RD[n]) signals of a channelised E1 link is shown in Figure 18. The receive data stream is an E1 frame with a singe framing byte (F1 to F8 in Figure 18) followed by octet bound time-slots 1 to 31. RCLK[n] is held quiescent during the framing byte. The RD[n] data bit (B1 of TS1) clocked in by the first rising edge of RCLK[n] after the framing byte is the most significant bit of time-slot 1. The RD[n] bit (B8 of TS31) clocked in by the last rising edge of RLCLK[n] before the framing byte is the least significant bit of time-slot 31. In Figure 18, the quiescent period is shown to be a low level on RCLK[n]. A high level, effected by extending the high phase of bit B8 of time-slot TS31, is equally acceptable. In channelised E1 mode, RCLK[n] can only be gapped during the framing byte. It must be active continuously at 2.048 MHz during all time-slot bits. Time-slots can be ignored by setting the PROV bit in the corresponding word of the receive channel provision RAM in the RCAS256 block to low.

Figure 18 – Channelised E1 Receive Link Timing

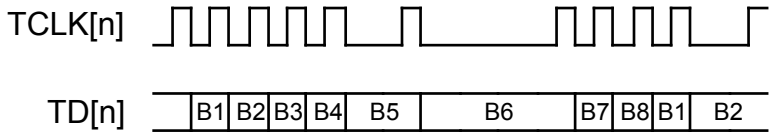


12.4 Transmit non H-MVIP Link Timing

The timing relationship of the transmit clock (TCLK[n]) and data (TD[n]) signals of a unchannelised link is shown in Figure 19. The transmit data is viewed as a contiguous serial stream. There is no concept of time-slots in an unchannelised link. Every eight bits are grouped together into a byte with arbitrary byte alignment. Octet data is transmitted from most significant bit (B1 in Figure 19) and ending with the least significant bit (B8 in Figure 19). Bits are updated on the falling edge of TCLK[n]. A transmit link may be stalled by holding the

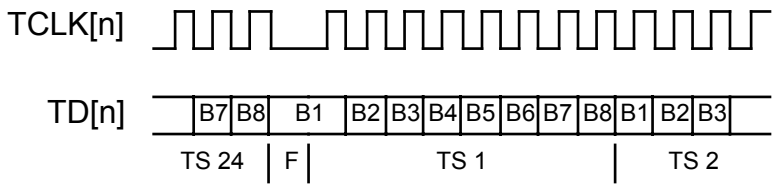
corresponding TCLK[n] quiescent. In Figure 19, bits B5 and B2 are shown to be stalled for one cycle while bit B6 is shown to be stalled for three cycles. In Figure 19, the quiescent period is shown to be a low level on TCLK[n]. A high level, effected by extending the high phase of the previous valid bit, is also acceptable. Gapping of TCLK[n] can occur arbitrarily without regard to byte nor frame boundaries.

Figure 19 – Unchannelised Transmit Link Timing



The timing relationship of the transmit clock (TCLK[n]) and data (TD[n]) signals of a channelised T1/J1 link is shown in Figure 20. The transmit data stream is a T1/J1 frame with a single framing bit (F in Figure 20) followed by octet bound time-slots 1 to 24. TCLK[n] is held quiescent during the framing bit. The most significant bit of each time-slot is transmitted first (B1 in Figure 20). The least significant bit of each time-slot is transmitted last (B8 in Figure 20). The TD[n] bit (B8 of TS24) before the framing bit is the least significant bit of time-slot 24. In Figure 20, the quiescent period is shown to be a low level on TCLK[n]. A high level, effected by extending the high phase of bit B8 of time-slot TS24, is equally acceptable. In channelised T1/J1 mode, TCLK[n] can only be gapped during the framing bit. It must be active continuously at 1.544 MHz during all time-slot bits. Time-slots that are not provisioned to belong to any channel (PROV bit in the corresponding word of the transmit channel provision RAM in the TCAS256 block set low) transmit the contents of the Idle Fill Time-slot Data register.

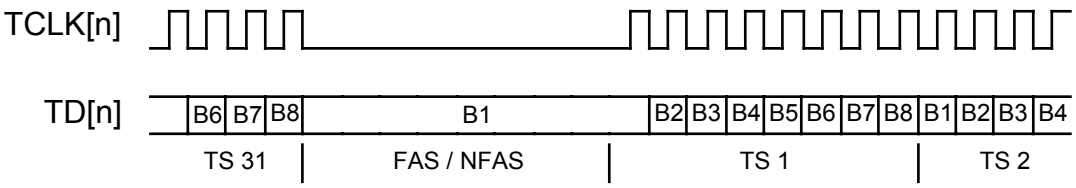
Figure 20 – Channelised T1/J1 Transmit Link Timing



The timing relationship of the transmit clock (TCLK[n]) and data (TD[n]) signals of a channelised E1 link is shown in Figure 21. The transmit data stream is an E1 frame with a singe framing byte (FAS/NFAS in Figure 21) followed by octet bound time-slots 1 to 31. TCLK[n] is held quiescent during the framing byte. The most

significant bit of each time-slot is transmitted first (B1 in Figure 21). The least significant bit of each time-slot is transmitted last (B8 in Figure 21). The TD[n] bit (B8 of TS31) before the framing byte is the least significant bit of time-slot 31. In Figure 21, the quiescent period is shown to be a low level on TCLK[n]. A high level, effected by extending the high phase of bit B8 of time-slot 31, is equally acceptable. In channelised E1 mode, TCLK[n] can only be gapped during the framing byte. It must be active continuously at 2.048 MHz during all time-slot bits. Time-slots that are not provisioned to belong to any channel (PROV bit in the corresponding word of the transmit channel provision RAM in the TCAS256 block set low) transmit the contents of the Idle Time-slot Fill Data register.

Figure 21 – Channelised E1 Transmit Link Timing



12.5 Receive APPI Timing

The receive Any-PHY packet interface (APPI) timing is shown in Figure 22 through Figure 24. The FREEDM-32A256 device provides data to an external controller using the receive APPI. The following discussion surrounding the receive APPI functional timing assumes that multiple FREEDM-32A256 devices share a single external controller. All Rx APPI signals are shared between the FREEDM-32A256 devices.

Figure 22 – Receive APPI Timing (Normal Transfer)

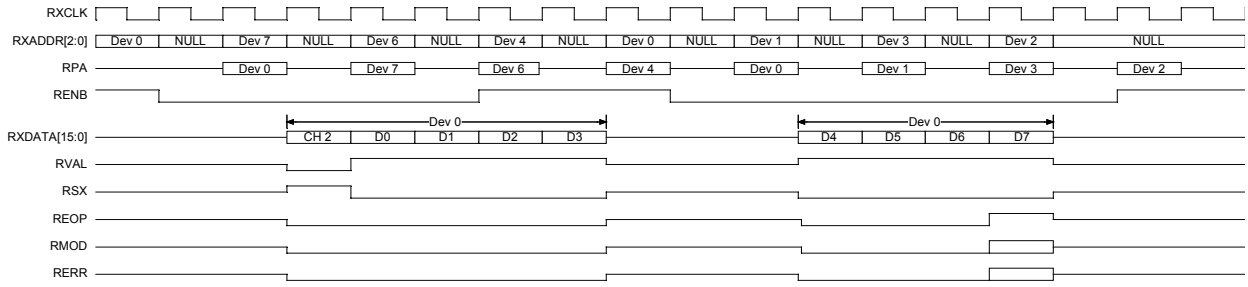


Figure 22 shows the transfer of an 8 word packet across the Rx APPI from FREEDM-32A256 device 0, channel 2. In this example, seven FREEDM-32A256 devices are sharing the Rx APPI, with device 5 being the null address.

The data transfer begins when the external controller selects FREEDM-32A256 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-32A256 samples RENB low and responds by asserting RSX two RXCLK cycles 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-32A256 tristates the Rx APPI. The external controller may hold RENB high for an indeterminate number of RXCLK cycles. The FREEDM-32A256 will wait until the external controller returns RENB low. Because the FREEDM-32A256 does not support interrupted data transfers on the Rx APPI, the external controller must reselect FREEDM-32A256 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-32A256 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-32A256 device 0. This is shown in Figure 25.

The FREEDM-32A256 will not pause burst data transfers across the Rx APPI.

The FREEDM-32A256 automatically deselects at the end of all burst data transfers. The FREEDM-32A256 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-32A256 may be programmed to overwrite RXDATA[7:0] 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-32A256 devices as well as for selection. During data transfer, the RXADDR[2:0] signals continue to poll the FREEDM-32A256 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-32A256 devices from briefly driving RPA. If RPA is a point-to-point signal for each FREEDM-32A256 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 RAPI256. 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-32A256 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-32A256 device (via RENB) to maximize bandwidth on the Rx APPI (shown in Figure 24).

Figure 23 – Receive APPI Timing (Auto Deselection)

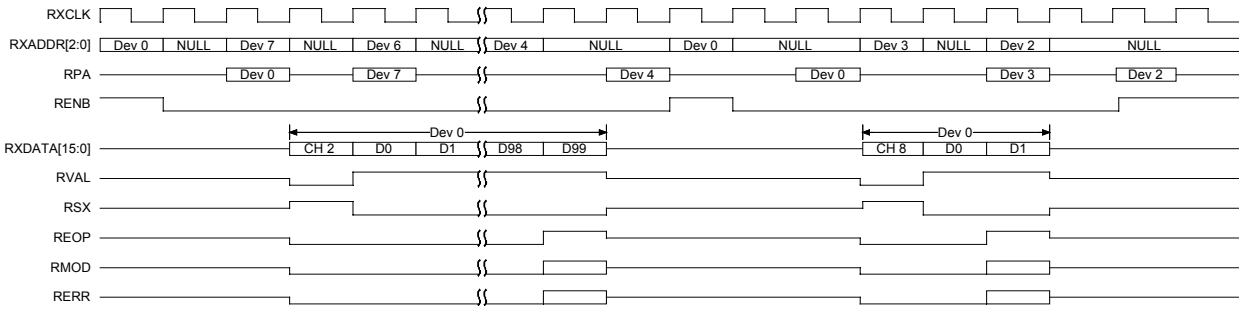


Figure 23 shows the transfer of a 100 word packet across the Rx APPI from FREEDM-32A256 device 0, channel 2 followed by the transfer of a 2 word packet from FREEDM-32A256 device 0, channel 8. More importantly, Figure 23 illustrates that, for back-to-back transfers from the same FREEDM-32A256 (device 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-32A256 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-32A256 device 0 by setting RENB high and placing address 0 on the RXADDR[2:0] signals. When

the FREEDM-32A256 samples RENB low, it begins the next data transfer as before.

Figure 24 – Receive APPI Timing (Optimal Reselection)

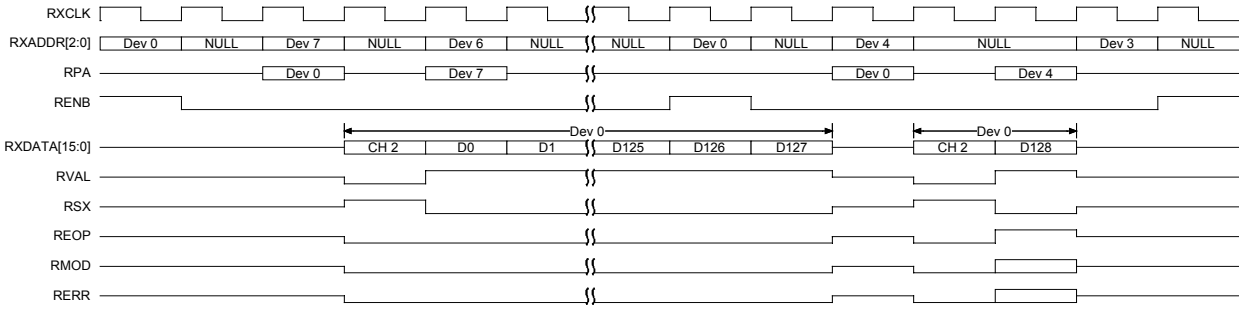


Figure 24 shows optimal bandwidth utilization across the Rx APPI.

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-32A256 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-32A256 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-32A256 device. In other words, Figure 24 shows the earliest possible time that the external controller could have set RENB high to reselect FREEDM-32A256 device 0.

Figure 25 – Receive APPI Timing (Boundary Condition)

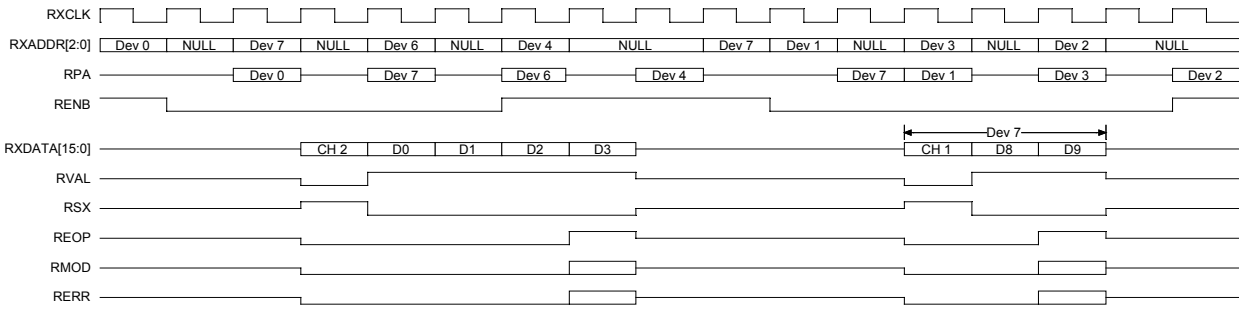


Figure 25 shows the boundary condition where a packet transfer completes shortly after the external controller has set RENB high to pause the FREEDM-32A256 device. The second data transfer is the final two words of a packet for FREEDM-32A256 device 7, channel 1.

When FREEDM-32A256 device 0 places D2 on RXDATA[15:0], the external controller sets RENB high to pause the FREEDM-32A256 device. In the following RXCLK cycle, the FREEDM-32A256 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-32A256 device 0, but can select another FREEDM-32A256 device sharing the Rx APPI. The external controller decides to select FREEDM-32A256 device 7 by placing this address on the RXADDR[2:0] signals. The external controller sets RENB low to commence data transfer from FREEDM-32A256 device 7.

12.6 Transmit APPI Timing

The transmit Any-PHY packet interface (APPI) timing is shown in Figure 26. An external controller provides data to the FREEDM-32A256 device using the transmit APPI. The following discussion surrounding the transmit APPI functional timing assumes that multiple FREEDM-32A256 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-32A256 devices.

Figure 26 – Transmit APPI Timing (Normal Transfer)

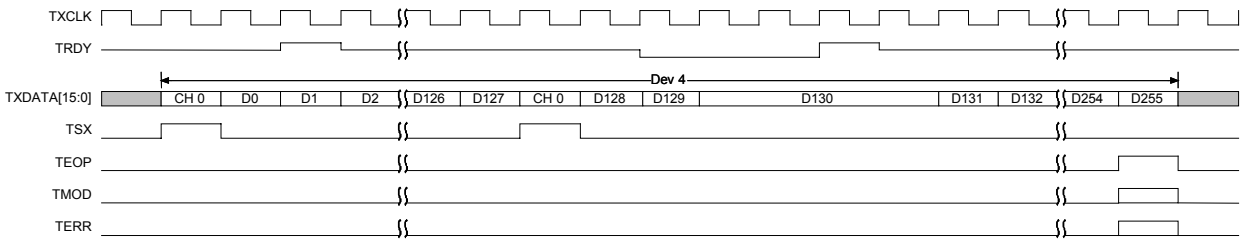


Figure 26 shows transfer of a 256 word packet on the Tx APPI of FREEDM-32A256 device 4, channel 0. The maximum burst data transfer (excluding channel address prepend) is 128 words, so two data transfers are required to complete the transfer of the 256 word packet.

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-32A256 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-32A256 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 26. In the second burst data transfer, the FREEDM-32A256 drives the TRDY signal low to indicate that the FIFOs in the TAPI256 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-32A256 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 TAPI256 has at least one empty FIFO, the FREEDM-32A256 drives the TRDY signal high. Upon sampling the TRDY signal high, the external controller completes the current burst data transfer. The FREEDM-32A256 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-32A256 device with small packets and allows the FREEDM-32A256 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-32A256 device. Figure 27 illustrates this condition.

Figure 27 – Transmit APPI Timing (Special Conditions)

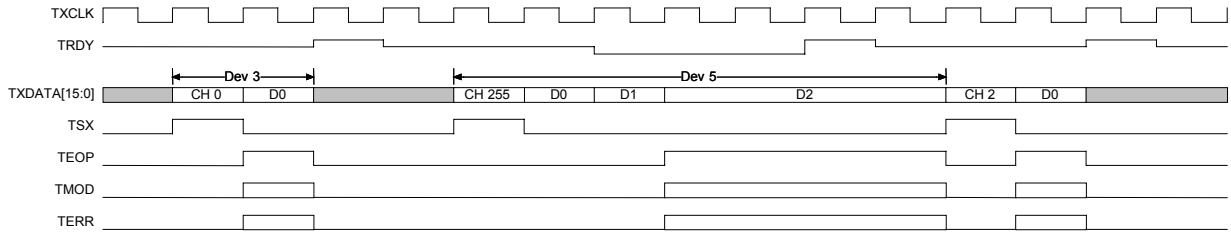


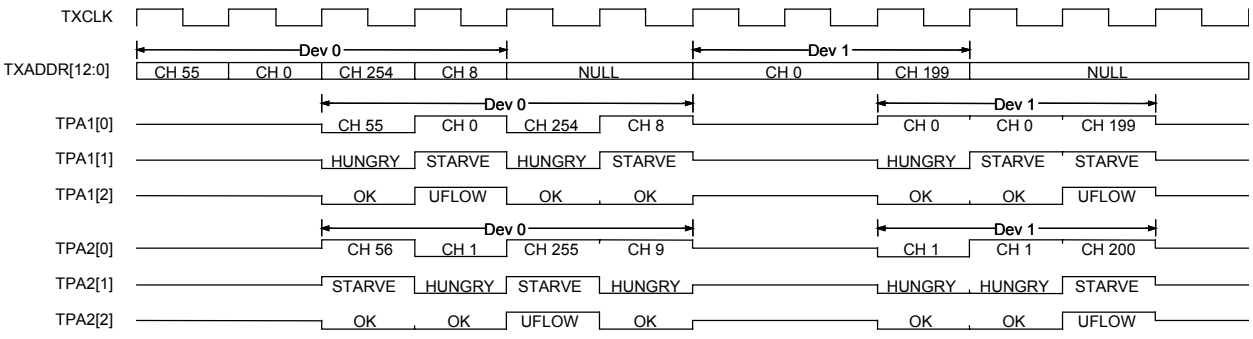
Figure 27 shows two special conditions – (1) the transfer of a one word packet illustrating how the external controller must wait until TRDY has been sampled

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-32A256 device 3, channel 0. The FREEDM-32A256 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-32A256 device 5, channel 255. 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-32A256 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-32A256 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-32A256 tristates the TRDY signal one TXCLK cycle after it has been driven high.

Figure 28 – Transmit APPI Timing (Polling)



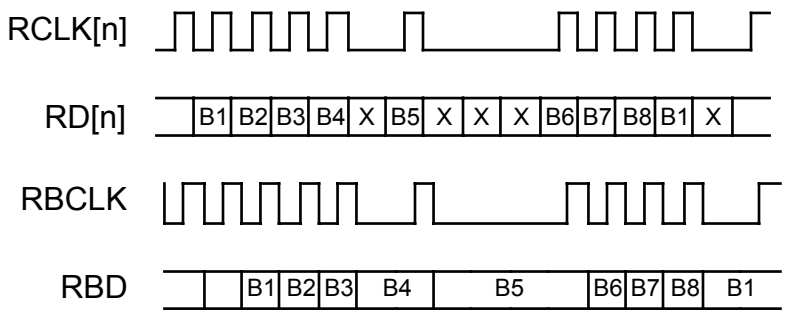
Polling is completely decoupled from device and channel selection on the Tx APPI. Accordingly, the TXADDR[12:0] signals continue to provide only a poll address for any of the FREEDM-32A256 devices sharing the Tx APPI. The most

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 28, channel 55 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 256 channels.

12.7 BERT Interface

The timing relationship between the receive link clock and data (RCLK[n] / RD[n]) and the receive BERT port signals (RBCLK / RBD) is shown in Figure 29. BERT is not supported for H-MVIP links. For non H-MVIP links, the selected RCLK[n] is placed on RBCLK after an asynchronous delay. The selected receive link data (RD[n]) is sampled on the rising edge of the associated RCLK[n] and transferred to RBD on the falling edge of RBCLK.

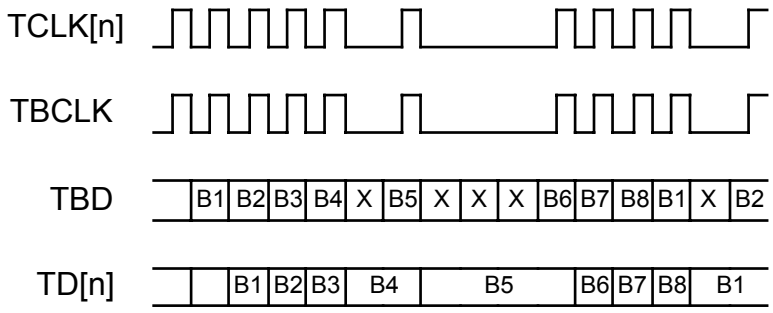
Figure 29 – Receive BERT Port Timing



The timing relationship between the transmit link clock and data (TCLK[n] / TD[n]) and the transmit BERT port signals (TBCLK / TBD) is shown in Figure 30. BERT is not supported for H-MVIP links. TCLK[n] is shown to have an arbitrary gapping. When TCLK[n] is quiescent, TBD is ignored (X in Figure 30). The

selected TCLK[n] is buffered and placed on TBCLK. The transmit BERT data (TBD) is sampled on the rising edge of the TBCLK and transferred to the selected TD[n] on the falling edge of TCLK[n].

Figure 30 – Transmit BERT Port Timing



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 21 – FREEDM-32A256 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
Voltage on Any Pin (non Any-PHY)	-0.3V to +6.0V
Volatge on Any Pin (Any-PHY)	-0.5V to $V_{DD3.3}$ + 0.5V
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 V to 3.6 V, $V_{DD2.5}$ = 2.3 V to 2.7V)

Table 22 – FREEDM-32A256 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} = -4 mA for all outputs except RBCLK, TBCLK, RBD, D[15:0], INTB, TPAn[2:0], TRDY, RPA, RSX, REOP, RXDATA[15:0], RXPRTY, RMOD and RERR where I_{OL} = -8 mA. Note 3.
V_{OH}	Output or Bi-directional High Voltage	2.4			Volts	I_{OH} = 4 mA for all outputs except RBCLK, TBCLK, RBD, D[15:0], INTB, TPAn[2:0], TRDY, RPA, RSX, REOP, RXDATA[15:0], RXPRTY, RMOD and RERR where I_{OH} = 8 mA. Note 3.
V_{T+}	Schmitt Triggered Input High Voltage	2.0		5.5	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.

Symbol	Parameter	Min	Typ	Max	Units	Conditions
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.
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.		220		mA	$V_{DD2.5}$ = 2.7V, Outputs Unloaded. RCLK[31:0]= TCLK[31:0]= 2.048 MHz. RMVCK[3:0] and TMVCK[3:0] tied low. RMV8DC and TMV8DC tied low. Note 4.
I_{DDOP}	Operating Current.		300		mA	$V_{DD2.5}$ = 2.7V, Outputs Unloaded. RCLK[2:0] = TCLK[2:0] = 51.84 MHz. RCLK[31:3], TCLK[31:3], RMVCK[3:0] and TMVCK[3:0] tied low. RMV8DC and TMV8DC tied low. 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-32A256 TIMING CHARACTERISTICS

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

Table 23 – FREEDM-32A256 Link Input (Figure 31 to Figure 34)

Symbol	Description	Min	Max	Units
	RCLK[31:0] Frequency (See Note 3)	1.542	1.546	MHz
	RCLK[31:0] Frequency (See Note 4)	2.046	2.050	MHz
	RCLK[2:0] Frequency (See Note 5)		51.84	MHz
	RCLK[31:3] Frequency (See Note 5)		10	MHz
	RCLK[31:0] Duty Cycle	40	60	%
	RMVCK[3:0] Frequency (See Note 6)	4.092	4.100	MHz
	RMVCK[3:0] Duty Cycle	40	60	%
	RMV8DC Frequency (See Note 7)	16.368	16.400	MHz
	RMV8DC Duty Cycle	40	60	%
	RMV8FPC Frequency (See Note 8)	4.092	4.100	MHz
	RMV8FPC Duty Cycle	40	60	%
tP_{MVC}	RMV8DC to RMV8FPC skew	-10	10	ns
	SYSCLK Frequency	25	45	MHz
	SYSCLK Duty Cycle	40	60	%
tS_{RD}	RD[2:0] Set-Up Time	1		ns
tH_{RD}	RD[2:0] Hold Time	2		ns
tS_{RD}	RD[31:3] Set-Up Time	5		ns
tH_{RD}	RD[31:3] Hold Time	5		ns
tS_{RD_2MVIP}	RD[31:0] Set-Up Time (2.048 Mbps H-MVIP Mode)	5		ns
tH_{RD_2MVIP}	RD[31:0] Hold Time (2.048 Mbps H-MVIP Mode)	5		ns

Symbol	Description	Min	Max	Units
tS_{RD_8MVIP}	RD[31:0] Set-Up Time (8.192 Mbps H-MVIP Mode)	5		ns
tH_{RD_8MVIP}	RD[31:0] Hold Time (8.192 Mbps H-MVIP Mode)	5		ns
tS_{RFPB}	RFPB[3:0] Set-Up Time	50		ns
tH_{RFPB}	RFPB[3:0] Hold Time	50		ns
TS_{RFP8B}	RFP8B Set-Up Time	50		ns
TH_{RFP8B}	RFP8B Hold Time	50		ns
tS_{TBD}	TBD Set-Up Time (See Note 9)	15		ns
tH_{TBD}	TBD Hold Time	0		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.

3. Applicable only to channelised T1/J1 links and measured between framing bits.

4. Applicable only to channelised E1 links and measured between framing bytes.

5. Applicable only to unchannelised links of any format and measured between any two RCLK rising edges.

6. Applicable only to 2.048 Mbps H-MVIP links and measured between any two RMVCK[n] falling edges.

7. Applicable only to 8.192 Mbps H-MVIP links and measured between any two RMV8DC falling edges.

8. Applicable only to H-MVIP links and measured between any two RMV8FPC falling edges.

9. TBD set-up time is measured with a 20 pF load on TBCLK. The set-up time increases by typically 1 ns for each 10 pF of extra load on TBCLK.

Figure 31 – Receive Data & Frame Pulse Timing (2.048 Mbps H-MVIP Mode)

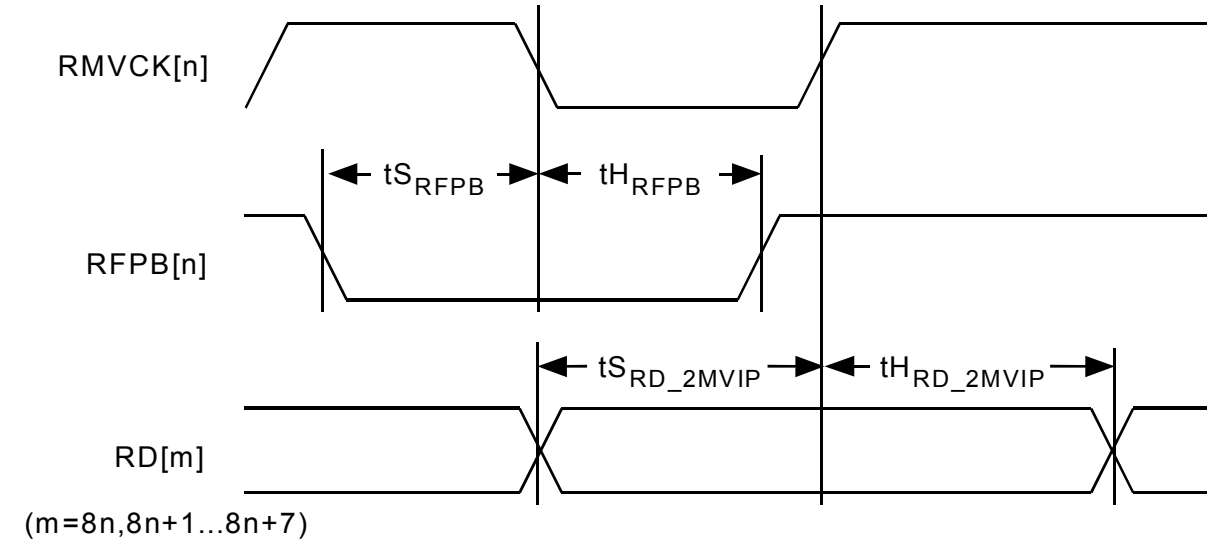


Figure 32 – Receive Data & Frame Pulse Timing (8.192 Mbps H-MVIP Mode)

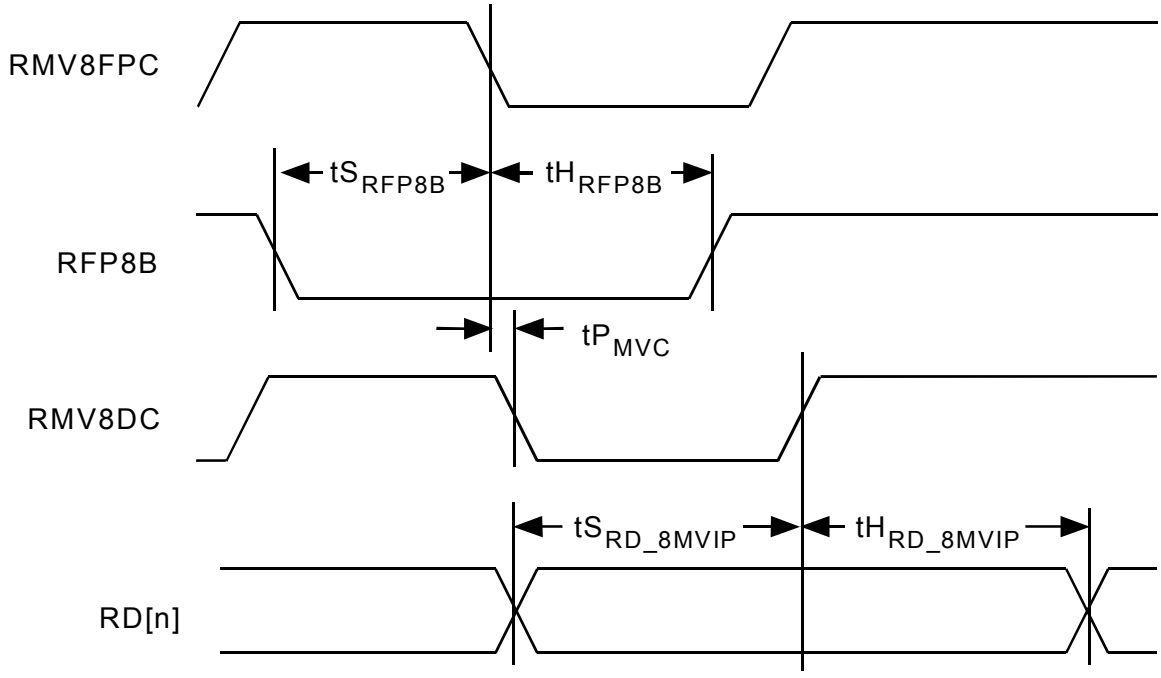


Figure 33 – Receive Data Timing (Non H-MVIP Mode)

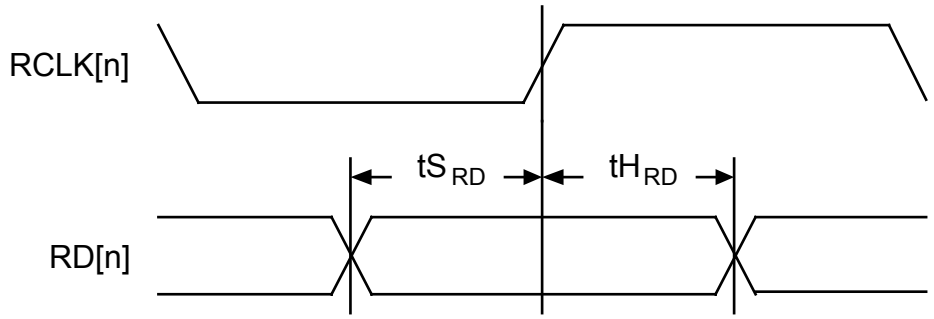


Figure 34 – BERT Input Timing

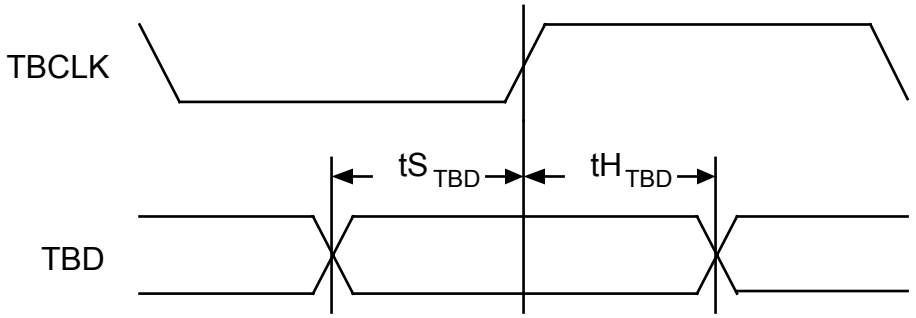


Table 24 – FREEDM-32A256 Link Output (Figure 35 to Figure 38)

Symbol	Description	Min	Max	Units
	TCLK[31:0] Frequency (See Note 4)	1.542	1.546	MHz
	TCLK[31:0] Frequency (See Note 5)	2.046	2.050	MHz
	TCLK[2:0] Frequency (See Note 6)		51.84	MHz
	TCLK[31:3] Frequency (See Note 6)		10	MHz
	TCLK[31:0] Duty Cycle	40	60	%
	TMVCK[3:0] Frequency (See Note 7)	4.092	4.100	MHz
	TMVCK[3:0] Duty Cycle	40	60	%
	TMV8DC Frequency (See Note 8)	16.368	16.400	MHz
	TMV8DC Duty Cycle	40	60	%
	TMV8FPC Frequency (See Note 9)	4.092	4.100	MHz

Symbol	Description	Min	Max	Units
	TMV8FPC Duty Cycle	40	60	%
tP_{MVC}	TMV8DC to TMV8FPC skew	-10	10	ns
tS_{TFPB}	TFPB[3:0] Set-Up Time	50		ns
tH_{TFPB}	TFPB[3:0] Hold Time	50		ns
TS_{TFP8B}	TFP8B Set-Up Time	50		ns
TH_{TF8PB}	TFP8B Hold Time	50		ns
tP_{TD}	TCLK[2:0] Low to TD[2:0] Valid	3	12	ns
tP_{TD}	TCLK[31:3] Low to TD[31:3] Valid	4	25	ns
tP_{TD_2MVIP}	TMVCK[3:0] Low to TD[31:0] Valid (2.048 Mbps H-MVIP Mode)	4	25	ns
tP_{TD_8MVIP}	TMV8DC Low to TD[31:0] Valid (8.192 Mbps H-MVIP Mode)	4	25	ns
tP_{RBD}	RBCLK Low to RBD Valid	-1	5	ns

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 50 pF load on all the outputs, except for the TD[2:0] outputs. For TD[2:0] outputs, propagation delays are measured with a 20 pF load. Maximum propagation delay for TD[2:0] increases by typically 1 ns for each 10 pF of extra load.

3. Output propagation delays of signal outputs that are specified in relation to a reference output are measured with a 50 pF load on both the signal output and the reference output.

4. Applicable only to channelised T1/J1 links and measured between framing bits.

5. Applicable only to channelised E1 links and measured between framing bytes.

6. Applicable only to unchannelised links of any format and measured between any two TCLK rising edges.

7. Applicable only to 2.048 Mbps H-MVIP links and measured between any two TMVCK[n] falling edges.

8. Applicable only to 8.192 Mbps H-MVIP links and measured between any two TMV8DC falling edges.

9. Applicable only to H-MVIP links and measured between any two TMV8FPC falling edges.

10. 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 35 – Transmit Data & Frame Pulse Timing (2.048 Mbps H-MVIP Mode)

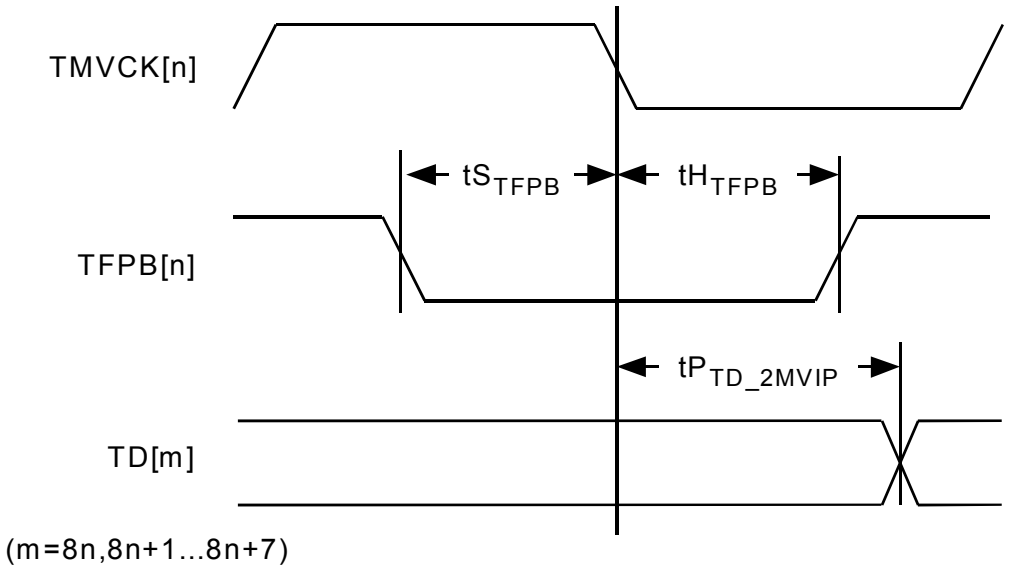


Figure 36 – Transmit Data & Frame Pulse Timing (8.192 Mbps H-MVIP Mode)

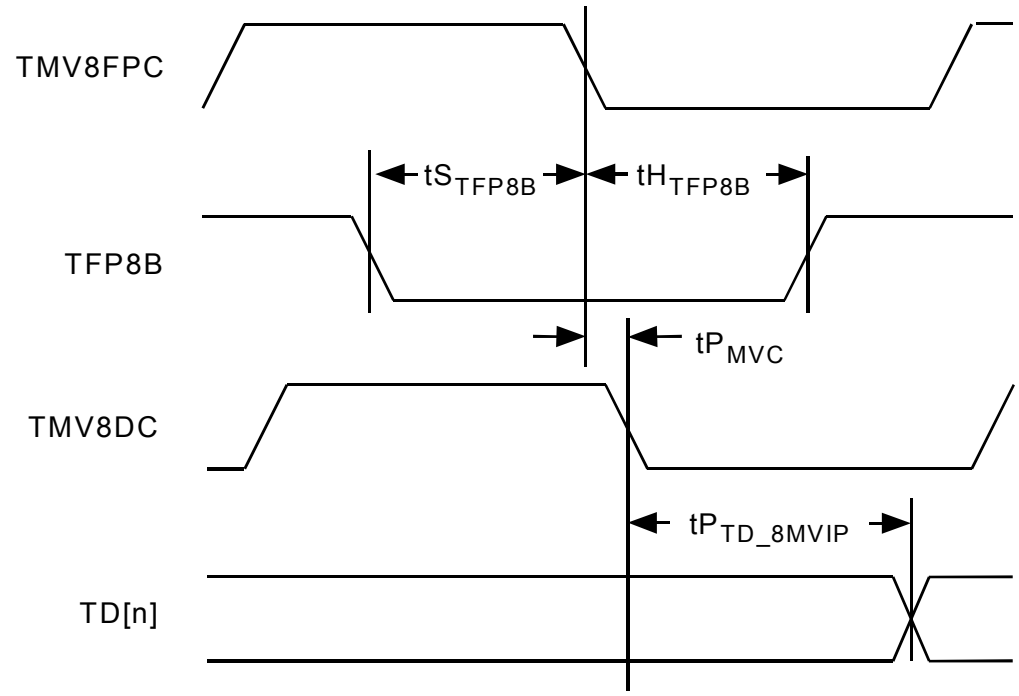


Figure 37 – Transmit Data Timing (Non H-MVIP Mode)

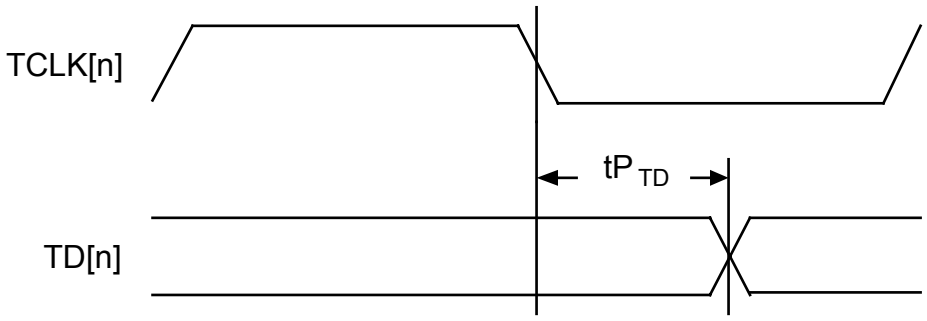


Figure 38 – BERT Output Timing

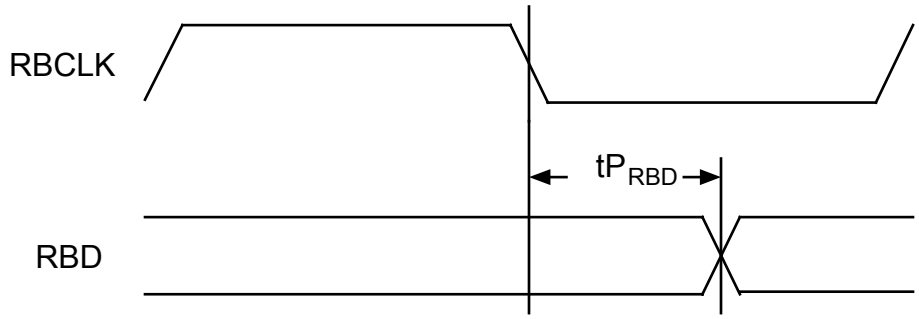


Table 25 – Any-PHY Packet Interface (Figure 39 to Figure 40)

Symbol	Description	Min	Max	Units
	RXCLK Frequency	0	50	MHz
	RXCLK Duty Cycle	40	60	%
	TXCLK Frequency	0	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

Notes on Any-PHY Packet Interface Output Timing:

1. Maximum and minimum output propagation delays are measured with a 100 pF load on all the outputs.

Figure 39 – Receive Any-PHY Packet Interface Timing

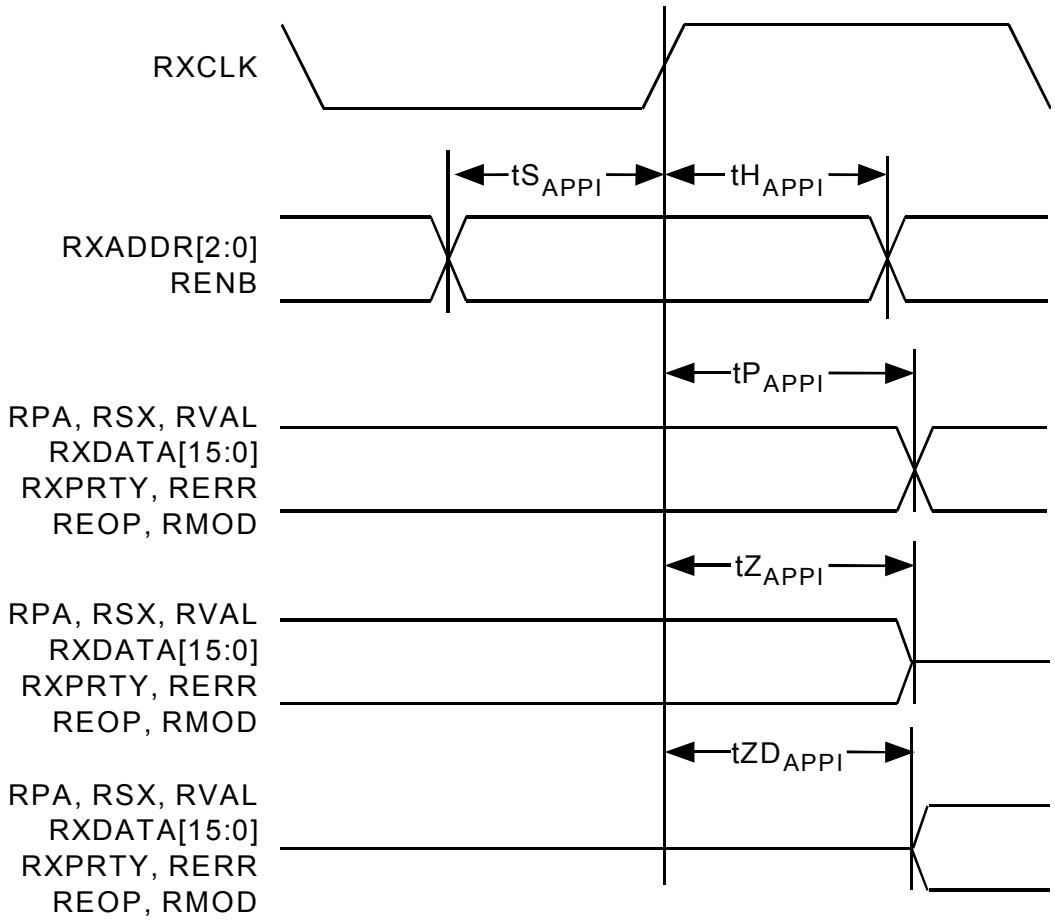


Figure 40 – Transmit Any-PHY Packet Interface Timing

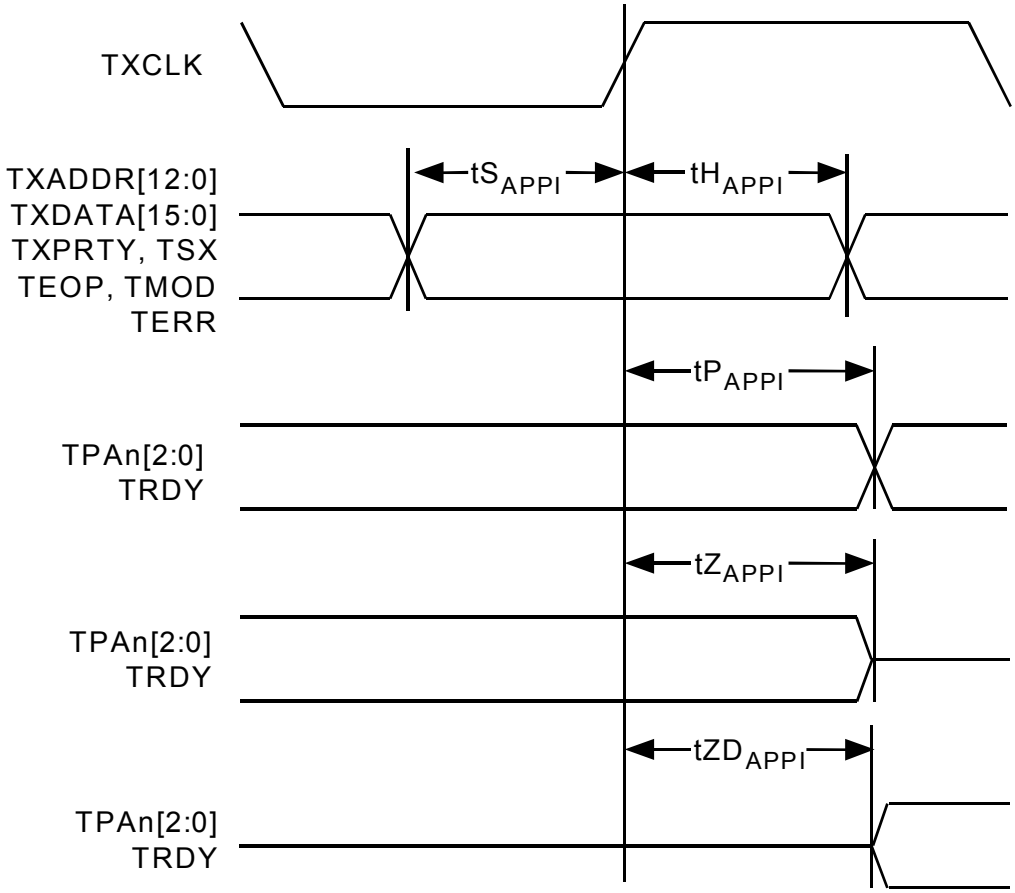
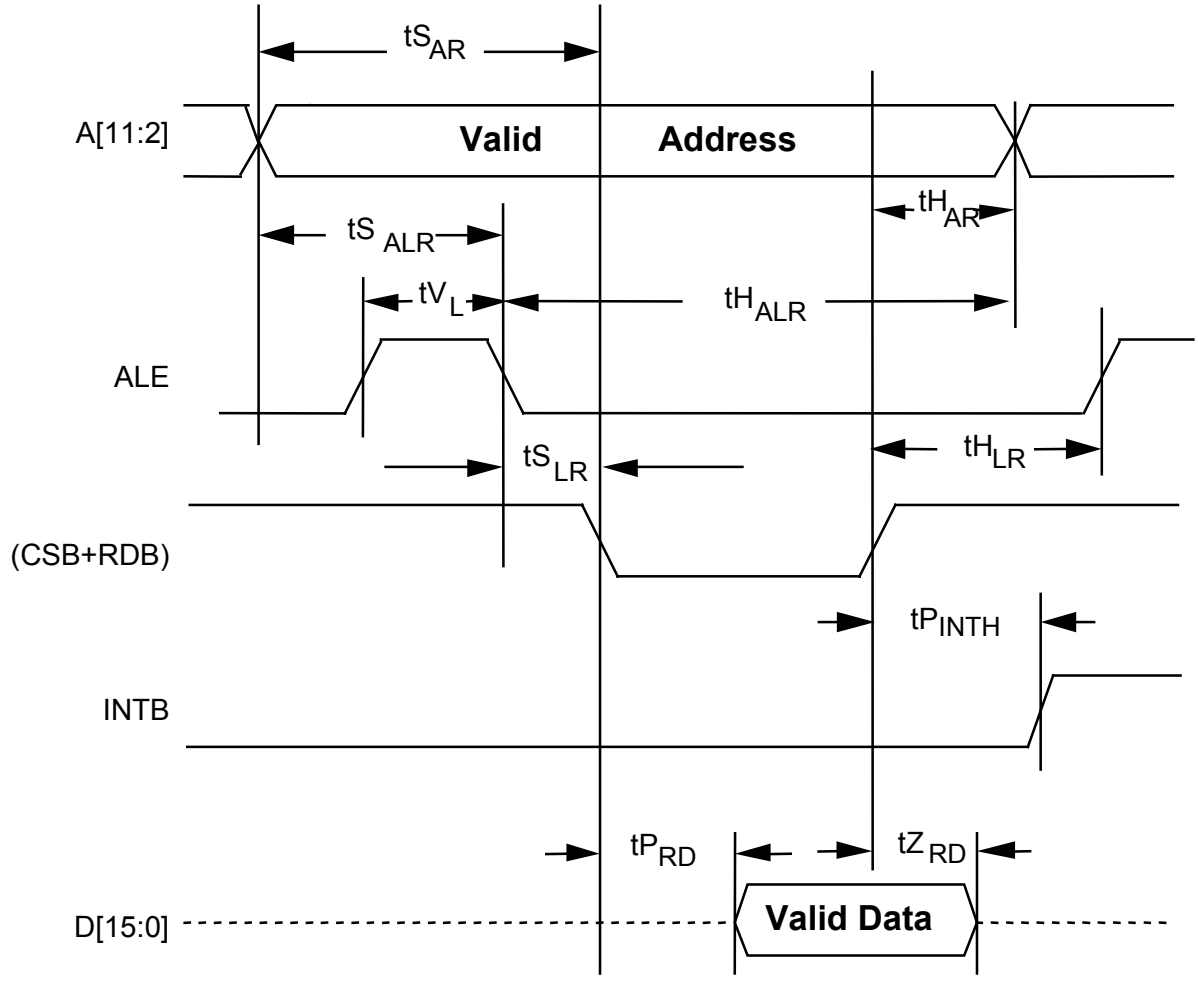


Table 26 – Microprocessor Interface Read Access (Figure 41)

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
tS_{LR}	Latch to Read Set-up	0		ns
tH_{LR}	Latch to Read Hold	5		ns

Symbol	Description	Min	Max	Units
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 41 – 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.

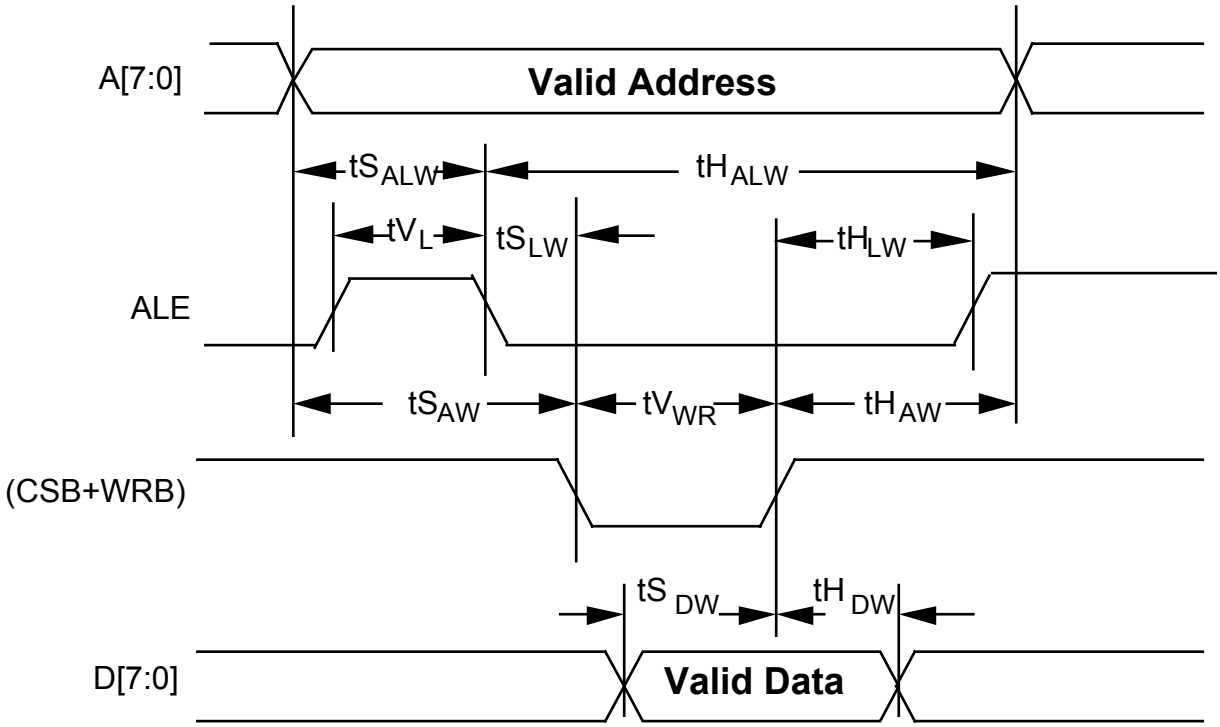
3. In non-multiplexed address/data bus applications, ALE should be held high, parameters tS_{ALR}, tH_{ALR}, tV_L, tS_{LR}, and tH_{LR} are not applicable.

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

Table 27 – Microprocessor Interface Write Access (Figure 42)

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 42 – Microprocessor Write Access Timing



Notes on Microprocessor Write Timing:

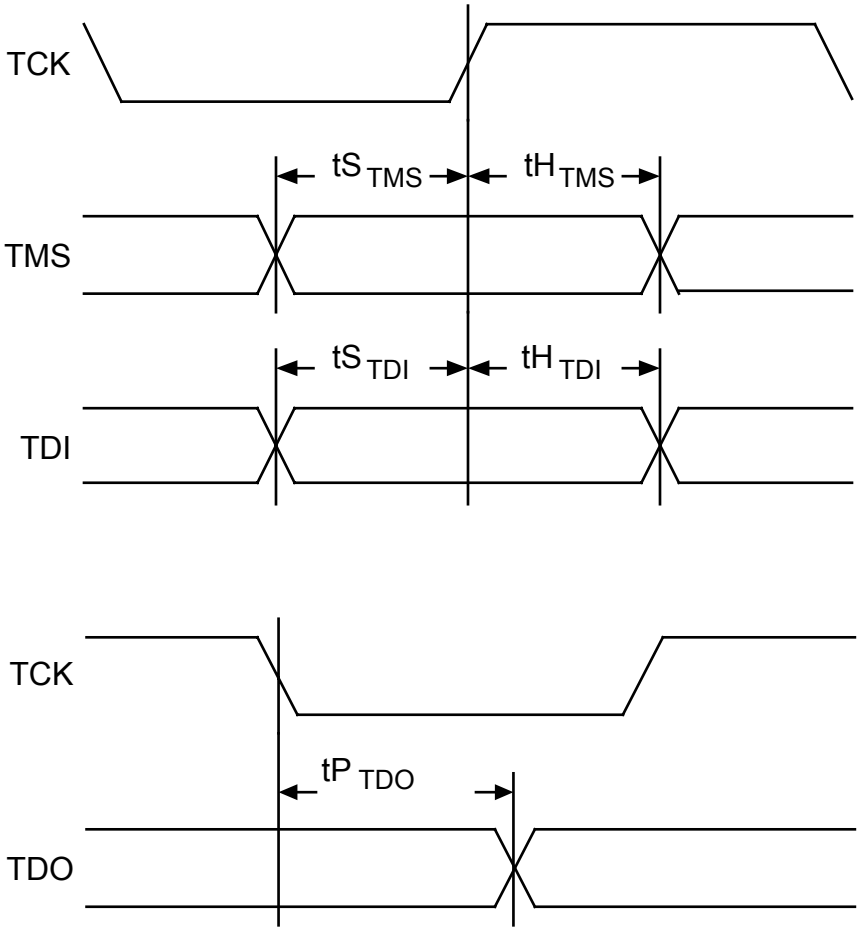
1. A valid write cycle is defined as a logical OR of the CSB and the WRB signals.
2. Microprocessor Interface timing applies to normal mode register accesses only.
3. In non-multiplexed address/data bus architectures, ALE should be held high, parameters tS_{ALW}, tH_{ALW}, tV_L, tS_{LW}, and tH_{LW}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

Table 28 – JTAG Port Interface (Figure 43)

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

Symbol	Description	Min	Max	Units
tH_{TMS}	TMS Hold time to TCK	50		ns
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 43 – JTAG Port Interface Timing



16 ORDERING AND THERMAL INFORMATION

Table 29 – FREEDM-32A256 Ordering Information

PART NO.	DESCRIPTION
PM7383-PI	329 Plastic Ball Grid Array (PBGA)

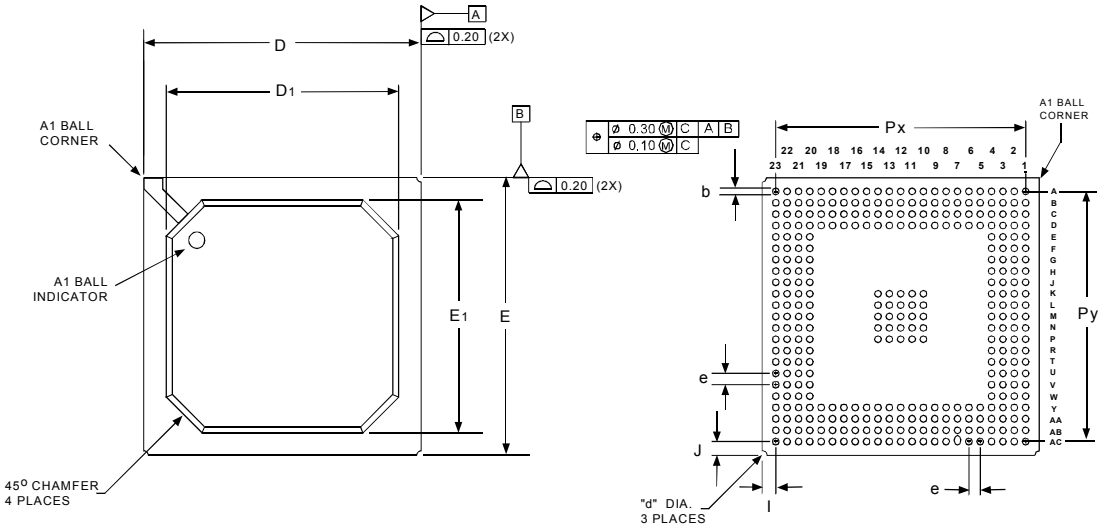
Table 30 – FREEDM-32A256 Thermal Information

PART NO.	AMBIENT TEMPERATURE	Theta Ja
PM7383-PI	-40°C to +85°C	25 °C/W

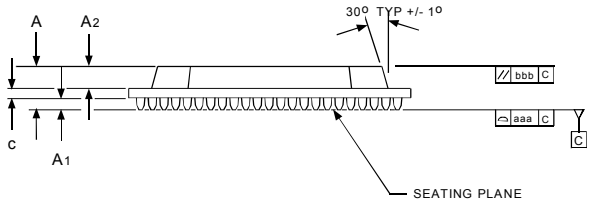
17 MECHANICAL INFORMATION

Figure 44 – 329 Pin Plastic Ball Grid Array (PBGA)

Note – The FREEDM-32P256 is manufactured using the 4-layer variant of the package shown in the drawing below.



TOP VIEW BOTTOM VIEW



SIDE VIEW

NOTES: 1) ALL DIMENSIONS IN MILLIMETER.
2) DIMENSION aaa DENOTES COPLANARITY.
3) DIMENSION bbb DENOTES PARALLEL.

PACKAGE TYPE : 329 PLASTIC BALL GRID ARRAY - PBGA																		
BODY SIZE : 31 x 31 x 2.33 MM (4 layer)																		
Dim.	A (2 layer)	A (4 layer)	A1	A2	D	D1	E	E1	l	J	b	c (2 layer)	c (4 layer)	d	e	Px	Py	aaa	bbb
Min.	2.12	2.12	0.50	1.12	30.80	25.50	30.80	25.50	-	-	0.60	-	-	-	-	27.84	27.84	-	-
Nom	2.33	2.33	0.60	1.17	31.00	26.00	31.00	26.00	1.53	1.53	0.76	0.56	0.56	1.0	1.27	27.94	27.94	-	-
Max.	2.54	2.56	0.70	1.22	31.20	26.70	31.20	26.70	-	-	0.90	-	-	-	-	28.04	28.04	0.15	0.15

CONTACTING PMC-SIERRA, INC.

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

Tel: +1 (604) 415-6000

Fax: +1 (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

PMC-2010336 (R1) Issue date: August 2001