



## **ATT1MX10 (SPINNAKER) Quad MAC and Transceivers**

### **Features**

- Four 10 Mbits/s Ethernet transceivers and MACs integrated together with separate transmit and receive port FIFOs and a single DMA interface simplify the design of a frame-switching hub. Each port can be configured separately for MAC, AUI, or twisted pair.
- Provides extensive network management capabilities that network administrators demand in today's hub equipment.
  - There are nine per-port transmit event counters and 18 per-port receive event counters for hardware-based network management.
  - Event counters can be read through a high-speed CPU interface that is separate from the DMA (Frame Data) interface.
  - The 16-bit asynchronous CPU interface can be accessed while frames are being read or written.
  - Receive frame statistics are appended to each frame at the end of the DMA transfer for custom management implementations and routing and switching information.
  - Per-port interrupt signals alert the CPU of transmit, receive, or bus errors. Per-port interrupt status registers distinguish the errors as any of the following: packet not transmitted due to late collision, excessive collisions, excessive deferral, or Tx FIFO underrun; packet not received because of receive jabber or FIFO overrun; TP transceiver has detected reversed polarity and corrected itself; link pulses are no longer being received; parity error while writing to Tx FIFO.
- The ATT1MX10 complies with ISO 8802.3 by default, but also can be configured for higher frame switching performance.
  - Short preamble generation.
  - Immediate retransmission after collision.
  - Selectable number of collision retries.
- Separate 128-byte transmit and receive FIFOs are provided per-port.
  - Tx and Rx FIFOs can hold multiple frames.
  - Deep FIFOs minimize overflows and underruns.
  - Frames are retransmitted automatically after a collision if the collision occurs within the first 64 octets of a packet.
  - Individual Rx thresholds for start of frame and midframe bursts allows DA (destination address) and SA (source address) to be read fast.
  - Undersized and fragment frames along with short events and noise hits can be rejected before any bus activity.
  - Direct system interface to FIFOs allows 32-bit single-cycle DMA operations at 16 MHz to 25 MHz.
- The ATT1MX10 can be configured for full-duplex transmission and reception.
  - The event counters gather statistics separately for the receive and transmit ports.
  - Independent transmit and receive data paths, including FIFOs, allow simultaneous transmit and receive operations on the same port.
- The CRC generator can be enabled or disabled on a per-packet basis through hardware and software control.
- Transmit and receive operations are under hardware control so that CPU accesses are not needed to transmit or receive a frame.
- 208-pin SQFP.

Note: Advisories are issued as needed to update product information. When using this data sheet for design purposes, please contact your Lucent Technologies Microelectronics Group Account Manager to obtain the latest advisory on this product.

■ 0050026 0027371 015 ■

1

## Table of Contents

<b>Contents</b>	<b>Page</b>	<b>Tables</b>	<b>Page</b>
Features .....	1	Table 1. ATT1MX10 Signal According to Its Pin Number in Numeric Sequence .....	6
Descriptions and Applications .....	4	Table 2. ATT1MX10 Signal According to Its Name in Alphabetic Sequence .....	7
Pin Information .....	5	Table 3. Ethernet Media Interface Signals .....	8
Pin Descriptions .....	8	Table 4. DMA Interface Signals.....	10
Functional Description .....	16	Table 5. Transmit Control Signals .....	12
Register and Counter Overview.....	16	Table 6. Receive Control Signals .....	12
Receive Overview .....	16	Table 7. CPU Interface Signals.....	13
Transmit Overview .....	16	Table 8. Miscellaneous Signals.....	15
Register and Counter Maps .....	17	Table 9. Port Addressing Format.....	17
Global Configuration Register.....	20	Table 10. Register and Counter Address Maps .....	17
Revision ID Register .....	21	Table 11. 32-Bit Counter, 16-Bit CPU Data Bus.....	19
Transmit Frame Configuration Register.....	22	Table 12. Global Configuration Register Bit Map .....	20
Receive Frame Configuration Register.....	24	Table 13. Global Configuration Register Bit Descriptions.....	20
Global FIFO Configuration Register.....	26	Table 14. Revision ID Register Bit Map.....	21
Port Interrupt Identification Register .....	27	Table 15. Revision ID Register Bit Descriptions.....	21
Transmit Port Diagnostic Register .....	28	Table 16. Transmit Frame Configuration Register Bit Map .....	22
Receive Port Diagnostic Register .....	30	Table 17. Transmit Frame Configuration Register Bit Descriptions .....	22
Frame Interrupt Mask Register .....	32	Table 18. Receive Frame Configuration Register Bit Map .....	24
Frame Interrupt Register.....	33	Table 19. Receive Frame Configuration Register Bit Descriptions .....	24
Counter Overflow Indication Register .....	34	Table 20. Global FIFO Configuration Register Bit Map .....	26
CPU Interrupt Mask Register .....	36	Table 21. Global FIFO Configuration Register Bit Descriptions .....	26
DMA Interface .....	38	Table 22. Port Interrupt Identification Register Bit Map .....	27
FIFO Operation .....	40	Table 23. Port Interrupt Identification Register Bit Descriptions .....	27
ISO 8802.3 MAC Functionality.....	44	Table 24. Transmit Port Diagnostic Register Bit Map .....	28
ISO 8802.3 Attachment Unit Interface (AUI).....	45	Table 25. Transmit Port Diagnostic Register Bit Descriptions .....	28
ISO 8802.3 Twisted-Pair (TP) Transceiver.....	46	Table 26. Receive Port Diagnostic Register Bit Map .....	30
Summary of ATT1MX10 Registers and Counters...	48		
ATT1MX10 Registers.....	49		
ATT1MX10 Counters .....	49		
Counter Definitions .....	50		
CPU Interface .....	52		
Application Information .....	54		
RMON Cross Reference .....	54		
Absolute Maximum Ratings.....	55		
Handling Precautions .....	56		
Electrical Characteristics .....	56		
Timing Characteristics.....	57		
Outline Diagram .....	77		
208-Pin SQFP, 1.3 mm Lead Frame.....	77		
Ordering Information.....	77		

## Table of Contents

Tables	Page	Figures	Page
Table 27. Receive Port Diagnostic Register		Figure 1. ATT1MX10 Block Diagram .....	4
Bit Descriptions .....	30	Figure 2. Single-Port Block Diagram .....	4
Table 28. Frame Interrupt Mask Register Bit Map .....	32	Figure 3. ATT1MX10 Pinout Diagram .....	5
Table 29. Frame Interrupt Mask Register Bit		Figure 4. AUI Driver Conditions .....	45
Descriptions .....	32	Figure 5. Capacitive Coupling .....	45
Table 30. Frame Interrupt Register Bit Map .....	33	Figure 6. The AUI Receivers Internal Bias	
Table 31. Frame Interrupt Mask Register Bit		Network .....	46
Descriptions .....	33	Figure 7. Pre-equalization Control Format .....	46
Table 32. Counter Overflow Indication Register		Figure 8. Recommended TP Port Configuration .....	47
Bit Map—High Word .....	34	Figure 9. Rx FIFO Read, 4-Word, Single-Cycle	
Table 33. Counter Overflow Indication Register		Access .....	58
Bit Map—Low Word .....	34	Figure 10. DMA Bus 3-State Timing .....	59
Table 34. Counter Overflow Indication Register		Figure 11. Receive Statistics (32-Bit Bus) .....	60
Bit Descriptions .....	35	Figure 12. Rx FIFO Read, 2-Word, Multiple Cycle	
Table 35. CPU Interrupt Mask Register		Access .....	61
Bit Map—High Word .....	36	Figure 13. Rx FIFO Read, 2-Word, Multiple FIFO	
Table 36. CPU Interrupt Mask Register		Accesses .....	62
Bit Map—Low Word .....	36	Figure 14. Rx FIFO Read, 2-Word, Multiple Chip	
Table 37. CPU Interrupt Mask Register Bit		Accesses .....	63
Descriptions .....	37	Figure 15. Byte Ordering on DATA Bus During Rx	
Table 38. Valid Byte (VB) Alignment .....	38	and Tx FIFO Transfers .....	63
Table 39. Word Count Threshold (WCTH)		Figure 16. Byte Order on DATA Bus During	
Settings .....	40	Rx Status, 32-Bit Bus .....	64
Table 40. RMON Cross Reference Objects .....	54	Figure 17. Tx FIFO Write, 4-Word, Single-Cycle	
Table 41. Rx FIFO Read, 4-Word, Single-Cycle		Access .....	65
Access .....	57	Figure 18. Tx FIFO Write, 2-Word, Multiple Cycle	
Table 42. DATA Bus 3-State Timing .....	59	Access .....	66
Table 43. Receive Statistics (32-bit Bus) Timing .....	60	Figure 19. Tx FIFO Write, 2-Word, Single-Cycle	
Table 44. Tx FIFO Write, 4-Word, Single-Cycle		Access, Last Word Written .....	67
Access Timing .....	64	Figure 20. Tx FIFO Write, 2-Word, Multiple FIFO	
Table 45. Tx FIFO Write, 2-Word, Single Cycle		Accesses .....	68
Access, Last Word Written .....	67	Figure 21. Tx FIFO Write, 2-Word, Multiple Chip	
Table 46. Rx FIFO Read to Tx FIFO Write Bus		Accesses .....	69
Turnaround Timing .....	70	Figure 22. Rx Read Followed by Tx Write Bus	
Table 47. 16-Bit, Asynchronous CPU Write		Turnaround Timing .....	70
Timing .....	72	Figure 23. Tx FIFO Write to Rx FIFO Read	
Table 48. 16-Bit, Asynchronous CPU Read		Timing .....	71
Timing .....	73	Figure 24. 16-Bit, Asynchronous CPU Write	
Table 49. MAC Transmit Timing .....	74	Timing .....	72
Table 50. MAC Receive Timing .....	74	Figure 25. 16-Bit, Asynchronous CPU Read	
Table 51. MAC Receive Clock Timing .....	75	Timing .....	73
Table 52. TP Driver Specification Timing .....	75	Figure 26. MAC Transmit Timing .....	74
Table 53. TP Link-Integrity Timing .....	76	Figure 27. MAC Receive Timing .....	74
		Figure 28. MAC Receive Clock Timing .....	75
		Figure 29. TP Driver Specification Timing .....	75
		Figure 30. Link-Integrity Timing .....	76

## Description and Applications

The ATT1MX10 provides four ISO 8802.3 10 Mb/s standard MACs and TP/AUI transceivers in a single 208-pin SQFP package. With deep internal FIFOs and a high-speed system interface, the ATT1MX10 is intended for Ethernet frame switching and multiple-port bridging and routing applications.

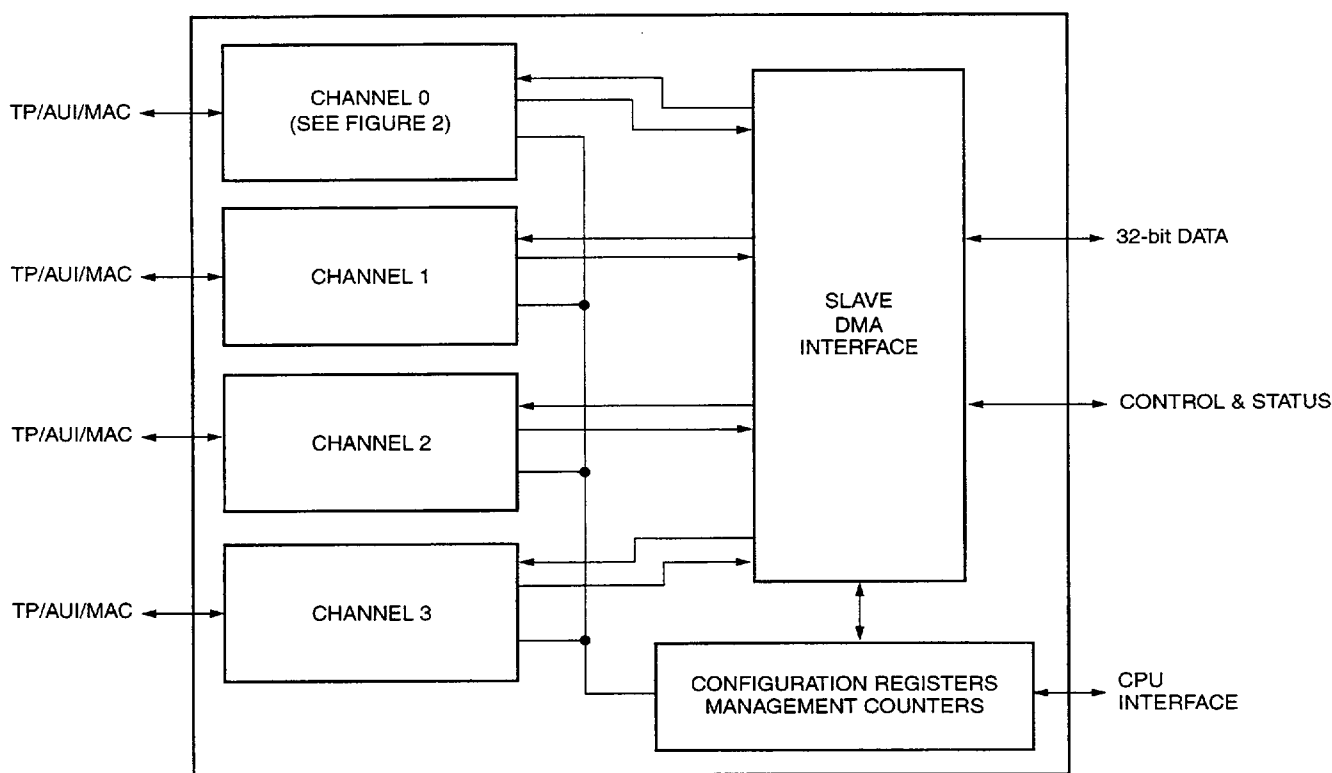
The ATT1MX10 allows single-cycle DMA transfers directly to and from its internal transmit and receive FIFOs. The deep FIFOs enable storing multiple frames on-chip, retransmitting a frame after a collision, and rejecting undersized frames before any DMA activity.

The ATT1MX10 also provides extensive on-chip counters and registers for out-of-band network management.

Typical applications of the ATT1MX10 include:

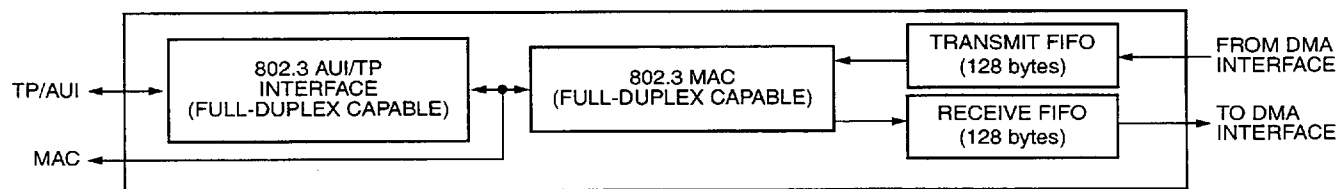
- Frame switched Ethernet hubs supporting store and forward, crosspoint switch, cut through, and other system architectures.
- Ethernet bridges and routers.
- Multiple-port server cards.

Figure 1 shows a block diagram of the ATT1MX10. An individual port block diagram is shown in Figure 2.



5-3640aF

**Figure 1. ATT1MX10 Block Diagram**



5-3641aF

**Figure 2. Single-Port Block Diagram**

## Pin Information

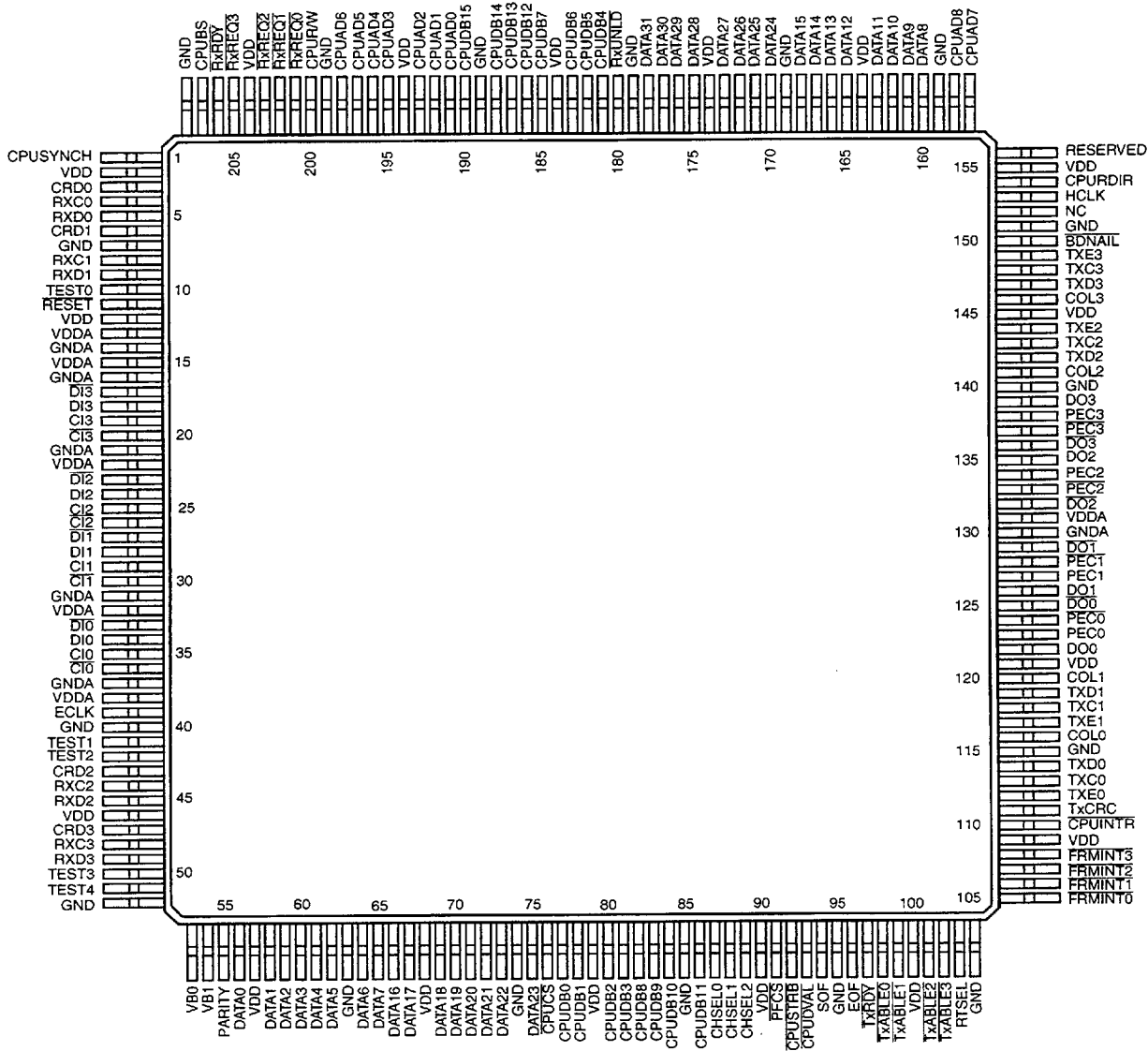


Figure 3. ATT1MX10 Pinout Diagram

5-3668(F).dR2

**Pin Information** (continued)

**Table 1. ATT1MX10 Signal According to Its Pin Number in Numeric Sequence**

Pin	Name	Pin	Name	Pin	Name	Pin	Name	Pin	Name	Pin	Name
1	NC3	36	$\overline{\text{CI0}}$	71	DATA20	106	$\overline{\text{FRMINT1}}$	141	COL2	176	DATA29
2	V <sub>DD</sub>	37	GND <sub>A</sub>	72	DATA21	107	$\overline{\text{FRMINT2}}$	142	TXD2	177	DATA30
3	CRD0	38	V <sub>DDA</sub>	73	DATA22	108	$\overline{\text{FRMINT3}}$	143	TXC2	178	DATA31
4	RXC0	39	ECLK	74	GND	109	V <sub>DD</sub>	144	TXE2	179	GND
5	RXD0	40	GND	75	DATA23	110	$\overline{\text{CPUINTR}}$	145	V <sub>DD</sub>	180	$\overline{\text{RxUNLD}}$
6	CRD1	41	TEST1	76	CPUCS	111	TxCRC	146	COL3	181	CPUDB4
7	GND	42	TEST2	77	CPUDB0	112	TXE0	147	TXD3	182	CPUDB5
8	RXC1	43	CRD2	78	CPUDB1	113	TXC0	148	TXC3	183	CPUDB6
9	RXD1	44	RXC2	79	V <sub>DD</sub>	114	TXD0	149	TXE3	184	V <sub>DD</sub>
10	TEST0	45	RXD2	80	CPUDB2	115	GND	150	BDNAIL	185	CPUDB7
11	$\overline{\text{RESET}}$	46	V <sub>DD</sub>	81	CPUDB3	116	COL0	151	GND	186	CPUDB12
12	V <sub>DD</sub>	47	CRD3	82	CPUDB8	117	TXE1	152	NC	187	CPUDB13
13	V <sub>DDA</sub>	48	RXC3	83	CPUDB9	118	TXC1	153	HCLK	188	CPUDB14
14	GND <sub>A</sub>	49	RXD3	84	CPUDB10	119	TXD1	154	CPURDIR	189	GND
15	V <sub>DDA</sub>	50	TEST3	85	GND	120	COL1	155	V <sub>DD</sub>	190	CPUDB15
16	GND <sub>A</sub>	51	TEST4	86	CPUDB11	121	V <sub>DD</sub>	156	NC1	191	CPUAD0
17	$\overline{\text{DI3}}$	52	GND	87	CHSEL0	122	DO0	157	CPUAD7	192	CPUAD1
18	DI3	53	VB0	88	CHSEL1	123	PEC0	158	CPUAD8	193	CPUAD2
19	CI3	54	VB1	89	CHSEL2	124	$\overline{\text{PEC0}}$	159	GND	194	V <sub>DD</sub>
20	$\overline{\text{CI3}}$	55	PARITY	90	V <sub>DD</sub>	125	$\overline{\text{DO0}}$	160	DATA8	195	CPUAD3
21	GND <sub>A</sub>	56	DATA0	91	PFC <sub>S</sub>	126	DO1	161	DATA9	196	CPUAD4
22	V <sub>DDA</sub>	57	V <sub>DD</sub>	92	$\overline{\text{CPUSTRB}}$	127	PEC1	162	DATA10	197	CPUAD5
23	$\overline{\text{DI2}}$	58	DATA1	93	$\overline{\text{CPUDVAL}}$	128	$\overline{\text{PEC1}}$	163	DATA11	198	CPUAD6
24	DI2	59	DATA2	94	SOF	129	$\overline{\text{DO1}}$	164	V <sub>DD</sub>	199	GND
25	CI2	60	DATA3	95	GND	130	GND <sub>A</sub>	165	DATA12	200	CPUR/W
26	$\overline{\text{CI2}}$	61	DATA4	96	EOF	131	V <sub>DDA</sub>	166	DATA13	201	$\overline{\text{RxREQ0}}$
27	$\overline{\text{DI1}}$	62	DATA5	97	$\overline{\text{TxRDY}}$	132	$\overline{\text{DO2}}$	167	DATA14	202	$\overline{\text{RxREQ1}}$
28	DI1	63	GND	98	$\overline{\text{TxABLE0}}$	133	PEC2	168	DATA15	203	$\overline{\text{RxREQ2}}$
29	CI1	64	DATA6	99	$\overline{\text{TxABLE1}}$	134	PEC2	169	GND	204	V <sub>DD</sub>
30	$\overline{\text{CI1}}$	65	DATA7	100	V <sub>DD</sub>	135	DO2	170	DATA24	205	$\overline{\text{RxREQ3}}$
31	GND <sub>A</sub>	66	DATA16	101	$\overline{\text{TxABLE2}}$	136	$\overline{\text{DO3}}$	171	DATA25	206	$\overline{\text{RxRDY}}$
32	V <sub>DDA</sub>	67	DATA17	102	$\overline{\text{TxABLE3}}$	137	PEC3	172	DATA26	207	NC2
33	$\overline{\text{DI0}}$	68	V <sub>DD</sub>	103	RTSEL	138	PEC3	173	DATA27	208	GND
34	DI0	69	DATA18	104	GND	139	DO3	174	V <sub>DD</sub>		
35	CI0	70	DATA19	105	$\overline{\text{FRMINT0}}$	140	GND	175	DATA28		

## Pin Information (continued)

Table 2. ATT1MX10 Signal According to Its Name in Alphabetic Sequence

Name	Pin	Name	Pin	Name	Pin	Name	Pin	Name	Pin	Name	Pin
BDNAIL	150	CPUDB10	84	DATA19	70	GND	7	PEC3	138	TXD1	119
CHSEL0	87	CPUDB11	86	DATA20	71	GND	40	PFC5	91	TXD2	142
CHSEL1	88	CPUDB12	186	DATA21	72	GND	52	RESET	11	TXD3	147
CHSEL2	89	CPUDB13	187	DATA22	73	GND	63	RTSEL	103	TXE0	112
CI0	35	CPUDB14	188	DATA23	75	GND	74	RXC0	4	TXE1	117
CI0	36	CPUDB15	190	DATA24	170	GND	85	RXC1	8	TXE2	144
CI1	29	CPUCS	76	DATA25	171	GND	95	RXC2	44	TXE3	149
CI1	30	CPUDVAL	93	DATA26	172	GND	104	RXC3	48	TxRDY	97
CI2	25	CPUINTR	110	DATA27	173	GND	115	RXD0	5	VB0	53
CI2	26	CPURDIR	154	DATA28	175	GND	140	RXD1	9	VB1	54
CI3	19	CPUR/W	200	DATA29	176	GND	151	RXD2	45	VDD	2
CI3	20	CPUSTRB	92	DATA30	177	GND	159	RXD3	49	VDD	12
COL0	116	CRD0	3	DATA31	178	GND	169	RxRDY	206	VDD	46
COL1	120	CRD1	6	DI0	33	GND	179	RxREQ0	201	VDD	57
COL2	141	CRD2	43	DI0	34	GND	189	RxREQ1	202	VDD	68
COL3	146	CRD3	47	DI1	27	GND	199	RxREQ2	203	VDD	79
CPUAD0	191	DATA0	56	DI1	28	GND	208	RxREQ3	205	VDD	90
CPUAD1	192	DATA1	58	DI2	23	GNDA	14	RxUNLD	180	VDD	100
CPUAD2	193	DATA2	59	DI2	24	GNDA	16	SOF	94	VDD	109
CPUAD3	195	DATA3	60	DI3	18	GNDA	21	TEST0	10	VDD	121
CPUAD4	196	DATA4	61	DI3	17	GNDA	31	TEST1	41	VDD	145
CPUAD5	197	DATA5	62	DO0	122	GNDA	37	TEST2	42	VDD	155
CPUAD6	198	DATA6	64	DO0	125	GNDA	130	TEST3	50	VDD	164
CPUAD7	157	DATA7	65	DO1	126	HCLK	153	TEST4	51	VDD	174
CPUAD8	158	DATA8	160	DO1	129	NC1	156	NC	152	VDD	184
CPUDB0	77	DATA9	161	DO2	135	NC2	207	TxABLE0	98	VDD	194
CPUDB1	78	DATA10	162	DO2	132	NC3	1	TxABLE1	99	VDD	204
CPUDB2	80	DATA11	163	DO3	139	PARITY	55	TxABLE2	101	VDDA	13
CPUDB3	81	DATA12	165	DO3	136	PEC0	123	TxABLE3	102	VDDA	15
CPUDB4	181	DATA13	166	ECLK	39	PEC0	124	TXC0	113	VDDA	22
CPUDB5	182	DATA14	167	EOF	96	PEC1	127	TXC1	118	VDDA	32
CPUDB6	183	DATA15	168	FRMINT0	105	PEC1	128	TXC2	143	VDDA	38
CPUDB7	185	DATA16	66	FRMINT1	106	PEC2	133	TXC3	148	VDDA	131
CPUDB8	82	DATA17	67	FRMINT2	107	PEC2	134	TxCRC	111		
CPUDB9	83	DATA18	69	FRMINT3	108	PEC3	137	TXD0	114		

## Pin Information (continued)

## Pin Descriptions

Table 3. Ethernet Media Interface Signals

Pin	Signal	Type	Name/Description
122, 125 126, 129 135, 132 139, 136	DO0, $\overline{\text{DO0}}$ DO1, $\overline{\text{DO1}}$ DO2, $\overline{\text{DO2}}$ DO3, $\overline{\text{DO3}}$	O	<b>Transmit Data Differential Pair.</b> If the media bits in the port's receive frame configuration register are 00, these pins are used for twisted-pair transmit data and should be connected to a filter module through a 48.7 $\Omega$ resistor. The data is Manchester encoded with a nominal bit rate of 10 Mbits/s. If the media bits are 01, these pins should be connected to external 100 $\Omega$ pull-up and pull-down resistors to complete the AUI transmit driver. If the media bits are 10 (MAC mode), these signals can be left unconnected.
123, 124 127, 128 134, 133 138, 137	PEC0, $\overline{\text{PEC0}}$ PEC1, $\overline{\text{PEC1}}$ PEC2, $\overline{\text{PEC2}}$ PEC3, $\overline{\text{PEC3}}$	O	<b>Pre-Equalization Control Differential Pair.</b> If the media bits in the port's receive frame configuration register are 00, these pins are used for TP Manchester data pre-equalization and should be connected to a filter module through a 464 $\Omega$ resistor. If the media bits are 01, these pins are connected, via isolation, to the transmit (DO) pair of the AUI transceiver cable. In AUI configuration, a 78 $\Omega$ resistor should be connected, in parallel, across the PEC and $\overline{\text{PEC}}$ signals. The data is Manchester encoded with a nominal bit rate of 10 Mbits/s. If the media bits are 10 (MAC mode), these signals can be left unconnected.
34, 33 28, 27 24, 23 18, 17	DI0, $\overline{\text{DI0}}$ DI1, $\overline{\text{DI1}}$ DI2, $\overline{\text{DI2}}$ DI3, $\overline{\text{DI3}}$	I	<b>Differential Data Inputs.</b> If the media bits in the port's receive frame configuration register are 00, these pins are differential Manchester-encoded receive data from the twisted-pair isolation transformers and filters. If the media bits are 01, these pins are connected via isolation to the receive (DI) pair of the AUI transceiver cable. Data on these pairs is Manchester encoded at the nominal rate of 10 Mbits/s plus any jitter. These pairs include internal 20 k $\Omega$ common-mode bias networks. If the media bits are 10 (MAC mode), these signals can be left unconnected.
35, 36 29, 30 25, 26 19, 20	CI0, $\overline{\text{CI0}}$ CI1, $\overline{\text{CI1}}$ CI2, $\overline{\text{CI2}}$ CI3, $\overline{\text{CI3}}$	I	<b>AUI Collision Differential Pair.</b> If the media bits in the port's receive frame configuration register are 01, these pins are connected, via isolation, to the collision presence (CI) pair of the AUI transceiver cable. The collision presence signal is a 10 MHz $\pm$ 15% square wave. These pairs include internal 20 k $\Omega$ common-mode bias networks. If the media bits are not 01, these pins are not used and can be left unconnected.
3 6 43 47	CRD0 CRD1 CRD2 CRD3	I/O	<b>Carrier Detect.</b> While the MAC interface is selected (Media = 10), carrier detect is an active-high input driven by an external device to initiate the reception of a frame. Carrier detect is also sensed during a transmission to tell if the media is being driven by another device. While the media bits are not 10, this signal is an output and reflects the state of the internal carrier detect output from the internal transceiver. This input is ignored while the port is programmed for full duplex.
4 8 44 48	RXC0 RXC1 RXC2 RXC3	I/O	<b>Receive Clock.</b> While the MAC interface is selected (Media = 10), this clock input is driven by an external device. The clock is derived from the recovered Manchester data by a transceiver/clock recovery device. Data is latched into the ATT1MX10 on the rising edge of RXC. During idle condition and reset, RXC can be held high or kept running. Five RXC cycles must be received after CRD is deasserted low at the end of a frame reception. While the media bits are not 10, this signal is an output and represents the internal state of the receive clock output from the internal receiver.



## Pin Information (continued)

## Pin Descriptions (continued)

Table 3. Ethernet Media Interface Signals (continued)

Pin	Signal	Type	Name/Description
5 9 45 49	RXD0 RXD1 RXD2 RXD3	I/O	<b>Receive Data.</b> While the MAC interface is enabled (Media = 10), this data input is driven by an external device. NRZ data recovered from the transceiver/clock recovery device is driven to the ATT1MX10 on this input. RXD is latched by the ATT1MX10 on the rising edge of RXC. While the media bits are not 10, this signal is an output and represents the internal state of receive data driven by the internal receiver.
112 117 144 149	TXE0 TXE1 TXE2 TXE3	O	<b>Transmit Enable.</b> While the MAC interface is selected (Media = 10), transmit enable is driven active-high on the rising edge of TXC by the ATT1MX10 to initiate a frame transmission. While media is not equal to 10, this signal represents the internal state of transmit enable output from the internal MAC.
113 118 143 148	TXC0 TXC1 TXC2 TXC3	I/O	<b>Transmit Clock.</b> While the MAC interface is selected (Media = 10), transmit clock is driven with a 10 MHz clock that must be generated by an external device connected to the ATT1MX10. The ATT1MX10 uses this clock to drive TXD synchronous to the transceiver. When the media bits are not set to 10, this signal is an output and represents the internal state of the transmit clock driven by the internal transceiver.
114 119 142 147	TXD0 TXD1 TXD2 TXD3	O	<b>Transmit Data.</b> While the MAC interface is enabled (Media = 10), the ATT1MX10 drives NRZ data synchronous to TXC on transmit data. TXD is driven high during idle conditions and reset. While media is not equal to 10, this signal represents the internal state of transmit data output from the internal MAC.
116 120 141 146	COL0 COL1 COL2 COL3	I/O	<b>Collision.</b> While the MAC interface is selected (Media = 10), collision is driven active-high by an external device when a collision is detected on the Ethernet network. If COL is asserted high and the ATT1MX10 is transmitting, the ATT1MX10 will commence its collision backoff algorithm according to the options set in the transmit frame configuration register. When the media bits are not set to 10, this signal is an output and represents the internal state of the collision output from the internal transceiver. This input is ignored while the port is programmed for full duplex.
39	ECLK	I	<b>Ethernet Clock.</b> 20 MHz $\pm$ 0.01%, 50% nominal, 40—60 worst-case duty cycle. This clock must be driven regardless of the port configurations.  <b>Note:</b> ISO 8802.3 requires the $\pm$ 0.01% frequency tolerance. The tolerance and duty cycle figures are specified only to limit the range over which the ATT1MX10 operates correctly. However, since this clock is used for Manchester data transmission, jitter performance degrades if clock sources with relatively large tolerances are used.

## Pin Information (continued)

## Pin Descriptions (continued)

Table 4. DMA Interface Signals

Pin	Signal	Type	Name/Description
56, 66 58, 67 59, 69 60, 70 61, 71 62, 72 64, 73 65, 75 160, 170 161, 171 162, 172 163, 173 165, 175 166, 176 167, 177 168, 178	DATA0, DATA16 DATA1, DATA17 DATA2, DATA18 DATA3, DATA19 DATA4, DATA20 DATA5, DATA21 DATA6, DATA22 DATA7, DATA23 DATA8, DATA24 DATA9, DATA25 DATA10, DATA26 DATA11, DATA27 DATA12, DATA28 DATA13, DATA29 DATA14, DATA30 DATA15, DATA31	I/O 3-state	<b>Data Bus.</b> The data bus is a 32-bit, bidirectional bus that interfaces to the transmit FIFO and the receive FIFO. The state of RTSEL determines whether the data bus is an input or an output. The ATT1MX10 requires one HCLK cycle to switch the data bus from Tx to Rx or from Rx to Tx.
55	Parity	I/O 3-state	<b>Parity.</b> Parity is generated during a DMA read cycle by the ATT1MX10. This bit represents the parity of the data being driven on DATA[31:0]. Parity is expected to be driven on this signal during a DMA write cycle. Parity as an input is not defined; the device is not differentiated between processing of packets with good parity and processing of packets with bad parity. This signal is valid based on the parity selection bits in the global configuration register. Parity should be tied to ground through a 100 k $\Omega$ resistor if this function is not being used.
54 53	VB1 VB0	I/O 3-state	<b>Valid Bytes.</b> As an input, valid bytes are sampled on every rising edge of HCLK while PFCS and TxRDY are active and are qualified by EOF and CHSEL. VB signifies the valid bytes in the doubleword transferred to the Tx FIFO. VB is latched for retransmission during a collision. As an output, VB is driven while PFCS, RxUNLD, RxRDY, and CHSEL are active to indicate which bytes in the current doubleword are valid at the end of an Rx frame. VB is qualified with EOF. Note that only the last transfer of frame data can contain invalid bytes. Refer to the section on the global FIFO configuration register for a description of the valid byte decode. The state of RTSEL determines whether VB is an input or an output.
153	HCLK	I	<b>Host Clock.</b> The host clock is used to clock data to and from the transmit and receive FIFOs, as well as all the internal digital logic of the ATT1MX10. Data can be read or written to one of the FIFOs through the data bus on each rising edge of HCLK. The host clock can be run up from 16 MHz to 25 MHz. HCLK must be active during reset. HCLK should have a 50% nominal duty cycle, $\pm 5\%$ if $22.5 \text{ MHz} \leq \text{HCLK} \leq 25 \text{ MHz}$ . If $16 \text{ MHz} \leq \text{HCLK} < 22.5 \text{ MHz}$ , then the tolerance on the duty cycle can be $\pm 10\%$ .

## Pin Information (continued)

## Pin Descriptions (continued)

Table 4. DMA Interface Signals (continued)

Pin	Signal	Type	Description																								
91	$\overline{\text{PFCS}}$	I	<b>Port FIFO Chip Select (Active-Low).</b> The port FIFO chip select input selects the ATT1MX10 FIFO interface. The data bus will be driven one HCLK cycle after $\overline{\text{PFCS}}$ becomes active-low with RTSEL high. The bus will be inactive (3-state) after $\overline{\text{PFCS}}$ is deasserted high or RTSEL is driven low. All DMA inputs are invalid if $\overline{\text{PFCS}}$ is high (inactive).																								
89 88 87	CHSEL2 CHSEL1 CHSEL0	I	<p><b>Channel Select.</b> The channel select inputs are used to select one of the four ATT1MX10 transmit or receive ports during an access to the Tx or Rx FIFO. This signal is qualified with RTSEL and <math>\overline{\text{PFCS}}</math>. When the bus is parked, all transfer attempts are ignored.</p> <table> <tr> <th>CHSEL2</th><th>CHSEL1</th><th>CHSEL0</th><th>Definition</th></tr> <tr> <td>1</td><td>X</td><td>X</td><td>Bus Parked</td></tr> <tr> <td>0</td><td>0</td><td>0</td><td>Port 0 selected</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>Port 1 selected</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>Port 2 selected</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>Port 3 selected</td></tr> </table>	CHSEL2	CHSEL1	CHSEL0	Definition	1	X	X	Bus Parked	0	0	0	Port 0 selected	0	0	1	Port 1 selected	0	1	0	Port 2 selected	0	1	1	Port 3 selected
CHSEL2	CHSEL1	CHSEL0	Definition																								
1	X	X	Bus Parked																								
0	0	0	Port 0 selected																								
0	0	1	Port 1 selected																								
0	1	0	Port 2 selected																								
0	1	1	Port 3 selected																								
103	RTSEL	I	<b>Receive-Transmit Select.</b> This input controls the direction of the ATT1MX10 data bus. When high, the ATT1MX10 DATA bus is an output (Rx FIFO accesses). When low, the data bus is an input (Tx FIFO accesses). This input is qualified with $\overline{\text{PFCS}}$ and CHSEL. When RTSEL changes from low to high, the ATT1MX10 asserts SOF, EOF, VB1, VB0, Parity, and DATA[31:0] after the rising edge of HCLK. These signals are deasserted after a combinatorial delay when RTSEL switches to a low.																								
94	SOF	I/O 3-state	<b>Start of Frame Indication.</b> While RTSEL is high (read), SOF is an output and indicates that the current DMA slave transfer contains the first byte of a received frame. SOF is driven on the rising edge of HCLK while $\overline{\text{PFCS}}$ is low and CHSEL and $\overline{\text{RxRDY}}$ are valid. When RTSEL is low (write), SOF is an input and should be driven active when the first word of a transmit frame is valid during the current DMA slave access. SOF is qualified by the ATT1MX10 on the rising edge of HCLK by $\overline{\text{PFCS}}$ , CHSEL, and $\overline{\text{TxRDY}}$ .																								
96	EOF	I/O 3-state	<b>End of Frame Indication.</b> While RTSEL is high (read), EOF is an output and indicates that the current DMA slave transfer contains the last byte of the received frame data. EOF is driven on the rising edge of HCLK while $\overline{\text{PFCS}}$ is low and CHSEL and $\overline{\text{RxRDY}}$ are valid. When RTSEL is low (write), EOF is an input and should be driven active when the last word of a transmit frame is valid during the current DMA slave access. EOF is qualified by the ATT1MX10 on the rising edge of HCLK by $\overline{\text{PFCS}}$ , CHSEL, and $\overline{\text{TxRDY}}$ .																								

## Pin Information (continued)

## Pin Descriptions (continued)

Table 5. Transmit Control Signals

Pin	Signal	Type	Description
98 99 101 102	$\overline{\text{TxABLE0}}$ $\overline{\text{TxABLE1}}$ $\overline{\text{TxABLE2}}$ $\overline{\text{TxABLE3}}$	O	<b>Transmit FIFO DMA Request (Active-Low).</b> The ATT1MX10 will assert this signal low when the transmit FIFO is capable of accepting data. The amount of free bytes in the FIFO will be equal to or greater than the word count threshold set in the global FIFO configuration register. Note that when a transmit channel is selected, the ATT1MX10 will deassert the corresponding $\overline{\text{TxABLE}}$ signal after one HCLK cycle and $\overline{\text{TxABLE}}$ cannot be reasserted while the corresponding transmit channel remains selected.
97	$\overline{\text{TxRDY}}$	I	<b>Transmit FIFO Data Ready (Active-Low).</b> The ATT1MX10 will clock data on the rising edge of HCLK from the data bus into the transmit FIFO while this input is active-low. When $\overline{\text{TxRDY}}$ transitions to an inactive state, high, the ATT1MX10 stops clocking data into the transmit FIFO. This signal can be used to insert wait-states in the DMA cycle. This input is qualified with $\overline{\text{TxABLE}}$ , $\overline{\text{PFCS}}$ , $\text{RTSEL}$ , and $\text{CHSEL}$ .
111	$\text{TxCRC}$	I	<b>Transmit CRC Checksum.</b> This input is sampled on every HCLK while $\overline{\text{TxRDY}}$ is active and is qualified by $\overline{\text{PFCS}}$ , $\overline{\text{TxRDY}}$ , $\text{CHSEL}$ , $\text{RTSEL}$ , and $\text{EOF}$ . If $\text{TxCRC}$ is high, the CRC checksum value will be calculated and appended to the transmitted frame. If it is low, the CRC will not be calculated and appended. This input is latched for retransmission during a collision. $\text{TxCRC}$ is ignored when $\text{HWCRC}$ bit = 0.

Table 6. Receive Control Signals

Pin	Signal	Type	Description
201 202 203 205	$\overline{\text{RxREQ0}}$ $\overline{\text{RxREQ1}}$ $\overline{\text{RxREQ2}}$ $\overline{\text{RxREQ3}}$	O	<b>Receive FIFO DMA Request (Active-Low).</b> When data in the receive FIFO reaches the threshold indicated by the $\text{RxSFTH}$ (receive start of frame threshold), the ATT1MX10 asserts $\overline{\text{RxREQ}}$ indicating that the system can read data. Subsequent assertions of $\overline{\text{RxREQ}}$ during the same frame will occur when the number of data bytes in the FIFO is greater than or equal to the word count threshold. Both thresholds are set in the global FIFO configuration register. Note that when a receive channel is selected, the ATT1MX10 will deassert the corresponding $\overline{\text{RxREQ}}$ signal after one HCLK cycle and the ATT1MX10 cannot assert $\overline{\text{RxREQ}}$ again while the corresponding receive channel is selected.
180	$\overline{\text{RxUNLD}}$	I	<b>Receive FIFO DMA Acknowledge (Active-Low).</b> The host asserts this signal low to acknowledge an $\overline{\text{RxREQ}}$ when it is able to read data from the Rx FIFO on the ATT1MX10 data bus. When $\overline{\text{RxUNLD}}$ is deasserted high, the ATT1MX10 stops clocking data from the receive FIFO. This signal can be used to insert wait-states into the DMA cycle. This input is qualified with $\overline{\text{PFCS}}$ , $\text{RTSEL}$ , and $\text{CHSEL}$ .
206	$\overline{\text{RxRDY}}$	O	<b>Receive FIFO Data Ready (Active-Low).</b> This output is asserted low when valid Rx data is driven on the ATT1MX10 DATA bus. This signal will be asserted while $\overline{\text{RxUNLD}}$ is held low. When $\overline{\text{RxRDY}}$ is driven low, $\text{CHSEL}$ can be changed to start reading data from the next channel.

**Pin Information** (continued)

**Pin Descriptions** (continued)

**Table 7. CPU Interface Signals**

Pin	Symbol	Type	Description
76	$\overline{\text{CPUCS}}$	I	<b>CPU Chip Select (Active-Low).</b> CPUCS must be asserted low in order to access a register or counter in the ATT1MX10. All CPU accesses are qualified with $\overline{\text{CPUCS}}$ .
154	CPURDIR	I	<b>Reserved.</b> Connect this signal to ground.
191, 197 192, 198 193, 157 195, 158 196	CPUAD0, CPUAD5 CPUAD1, CPUAD6 CPUAD2, CPUAD7 CPUAD3, CPUAD8 CPUAD4	I	<b>CPU Address Bus.</b> The address bus is used during CPU accesses to the ATT1MX10 configuration and control registers and the ATT1MX10 counters. The address specifies which counter or register is being read or written. Data is latched by address; if CPUAD changes state, new data will be driven. CPUAD is qualified by $\overline{\text{CPUCS}}$ and $\overline{\text{CPUSTRB}}$ .
77, 82 78, 83 80, 84 81, 86 181, 186 182, 187 183, 188 185, 190	CPUDB0, CPUDB8 CPUDB1, CPUDB9 CPUDB2, CPUDB10 CPUDB3, CPUDB11 CPUDB4, CPUDB12 CPUDB5, CPUDB13 CPUDB6, CPUDB14 CPUDB7, CPUDB15	I/O 3-state	<b>CPU Data Bus.</b> The 16-bit CPU data bus is used for reading and writing counters and registers in the ATT1MX10. CPUDB is driven by the ATT1MX10 while $\overline{\text{CPUCS}}$ and $\overline{\text{CPUSTRB}}$ are asserted and CPUR/W is high. When doing back-to-back read writes, the ATT1MX10 requires one HCLK cycle to 3-state the bus.
200	CPUR/W	I	<b>CPU Read/Not Write.</b> The CPUR/W signal indicates which direction the CPU data bus will be in for the current register or counter access. The signal should be driven high when reading a register/counter and low when writing a register. This signal is qualified with $\overline{\text{CPUCS}}$ and $\overline{\text{CPUSTRB}}$ .
92	$\overline{\text{CPUSTRB}}$	I	<b>CPU Data Strobe (Active-Low).</b> The CPU data strobe qualifies an access to the CPU interface. This signal is used to latch data and address into the ATT1MX10 during a write operation. During a read, this signal is used by the CPU to latch data from the ATT1MX10 data bus.

**Pin Information** (continued)

**Pin Descriptions** (continued)

**Table 7. CPU Interface Signals** (continued)

Pin	Symbol	Type	Description
93	$\overline{\text{CPUDVAL}}$	O	<b>CPU Data Valid (Active-Low).</b> This signal is an output and is driven by the ATT1MX10 during a read when data becomes valid. During a write, the ATT1MX10 will drive this signal low when it latches data from the bus.
110	$\overline{\text{CPUINTR}}$	O	<b>CPU Interrupt (Active-Low).</b> The CPU interrupt will be active when any one of the event counters reaches its maximum limit. (Counters will roll over and continue to count). The particular counter that caused the interrupt can be determined by reading the counter overflow indication register. As an option, the $\overline{\text{FRMINT}}$ interrupts can be routed to $\overline{\text{CPUINTR}}$ . In this case, $\overline{\text{CPUINTR}}$ will be driven low when any of the $\overline{\text{FRMINT}}$ pins would be driven.
105 106 107 108	$\overline{\text{FRMINT0}}$ $\overline{\text{FRMINT1}}$ $\overline{\text{FRMINT2}}$ $\overline{\text{FRMINT3}}$	O	<b>Frame Interrupt (Active-Low).</b> Each port of the ATT1MX10 is capable of interrupting a host when an error occurs while a frame is being received or transmitted. The frame interrupt signal is maskable by the frame interrupt mask register. The frame interrupt register indicates which events caused the interrupt.

## Pin Information (continued)

## Pin Descriptions (continued)

Table 8. Miscellaneous Signals

Pin	Signal	Type	Description
11	RESET	I	<b>Reset (Active-Low).</b> This input places the ATT1MX10 into a reset state. When RESET is deasserted high, the ATT1MX10 will be in an idle state. All counters will be cleared and configuration registers will be set to their default values. In order to effectively reset the ATT1MX10, RESET must be held active until ECLK and HCLK are running. RESET needs to be driven after powerup to put the ATT1MX10 into a known configuration. After a RESET cycle, the ATT1MX10 will be ready to transmit and receive frames; configuration registers only need to be written if the default configuration needs to be changed.
10 41 42 50 51 152	TEST0 TEST1 TEST2 TEST3 TEST4 NC	I	<b>Test Pins.</b> (6 pins.) These pins should be tied low for normal operation.
2, 121, 12, 145, 46, 155, 57, 164, 68, 174, 79, 184, 90, 194, 100, 204, 109	VDD	—	<b>Digital Power.</b> (17 pins.)
13, 32, 15, 38, 22, 131,	VDDA	—	<b>Analog Power.</b> (6 pins.)
7, 140, 40, 151, 52, 159, 63, 169, 74, 179, 85, 189, 95, 199, 104, 208, 115	GND	—	<b>Digital Ground.</b> (17 pins.)
14, 31, 16, 37, 21, 130,	GNDA	—	<b>Analog Ground.</b> (6 pins.)
156	NC1	—	<b>Reserved.</b> This pin is reserved and should be tied low.
207	NC2	—	<b>Reserved.</b> This pin is reserved and can either be left unconnected or tied low.
1	NC3	—	<b>Reserved.</b> This pin is reserved and should be tied low.
150	BDNAIL	I	<b>Bed-of-Nails Test (Active-Low).</b> This signal, when driven low, places all digital pins in 3-state for bed-of-nails testing. For normal operation, this signal should be pulled high with a 100 kΩ resistor.

## Functional Description

The ATT1MX10 can be viewed as four separate Ethernet transmitters and four separate Ethernet receivers and as a memory device where the host moves Ethernet frames on a bus. Host overhead is held to a minimum while the ATT1MX10 handles the transmission and reception of frames in accordance with the ISO 8802.3 specification. A host system can be built to read and write the appropriate FIFOs in the ATT1MX10 in order to receive and transmit frames.

The ATT1MX10 also provides an elegant, high-speed CPU interface. An exhaustive list of network management counters can be read through this interface which makes out-of-band network management possible.

## Register and Counter Overview

The ATT1MX10 resets to a default configuration and can also be configured for custom operation through several registers. The registers are addressable in memory space through the ATT1MX10 CPU interface. This bank of registers also includes the Event Counters that monitor network statistics. Most of these registers are duplicated for each of the four ports of the ATT1MX10. Where appropriate, a single register is used globally. The counters are all specific to a single port. The upper two address bits select which port register or counter is being accessed; these inputs must be driven low when accessing a global register. Some counters are 32 bits wide while others are 16 bits wide. The CPU data bus is 16 bits wide.

## Receive Overview

As frames are received from the Ethernet media through either the twisted-pair port, the attachment unit interface (AUI), or the MAC interface, data is recovered and stored in a 32-bit wide FIFO. When a programmable threshold is reached (a minimum number of bytes is stored in the FIFO) or EOF is met, the  $\overline{\text{RxREQ}}$  signal for the appropriate port becomes active-low. This output signal serves as an interrupt to the host system signifying that there is data in the FIFO that can be read. When the host drives the  $\overline{\text{RxUNLD}}$  input of the ATT1MX10 low, the ATT1MX10 will start unloading the Rx FIFO by driving data on to the 32-bit data bus on every rising edge of HCLK. It is the host's responsibility to address and strobe the external memory system.

The threshold setting guarantees that an exact number of bytes are in the FIFO so that a burst transfer can be used. The transfer can also be preempted by deasserting the  $\overline{\text{RxUNLD}}$  input. The last transfer of the frame is indicated by the signal EOF being asserted high during

the DMA transfer. If EOF is asserted in the middle of a burst, the burst will terminate after the receive statistics are read. In other words, data from two separate frames cannot be read from the ATT1MX10 during the same burst. If a second frame is stored in the FIFO behind the one currently being read out and there is enough data in the FIFO to satisfy the programmed threshold,  $\overline{\text{RxREQ}}$  will be reasserted, and a new cycle needs to be started.  $\overline{\text{RxREQ}}$  will only be reasserted after the current channel is deselected.

The last 4 bytes of each received frame is status. These 4 bytes of status will contain 2 bytes of frame length and 2 bytes of receive status. Status will be driven on the cycle following EOF.

## Transmit Overview

Frames are transmitted from the FIFO as long as there is data to be transmitted. It is the responsibility of the host system to keep the transmit FIFO from underrunning. Once the FIFO has detected that a certain amount of data (specified by the TxSFTH bits in the global FIFO configuration register) has been loaded in the FIFO, the ATT1MX10 will commence the transmit operation. When the Ethernet media is clear, the actual transmission will begin. In the event that a legal collision occurs, the ATT1MX10 will back off and retry the transmission, provided the maximum number of retries has not been met. The ATT1MX10 will preserve the first 64 bytes of data in the FIFO so that if there is a collision during the transmission of these bytes, the ATT1MX10 can retransmit the frame without the host system having to reload the FIFO. If a collision occurs after 64 bytes have been transmitted, an interrupt can be generated.

A transmit FIFO threshold size can be set in the global FIFO configuration register. When at least WCTH of space is available in the Tx FIFO, the  $\overline{\text{TxBLE}}$  signal will be asserted low. (When RESET is deasserted high, the Tx FIFOs will be empty and the  $\overline{\text{TxBLE}}$  signals for each of the FIFOs will be asserted low.) The ATT1MX10 will start loading data into the Tx FIFO on every HCLK while  $\overline{\text{TxRDY}}$  is returned low. This allows an exact burst transfer. The transfer can also be stopped immediately if  $\overline{\text{TxRDY}}$  is deasserted high. During the transfer of the last word of a frame, the host system should drive the following signals to transfer important transmit information to the ATT1MX10: EOF should be driven high to indicate this word contains the last byte of the frame; VB should be asserted to indicate how many bytes of the word contain valid data (note that VB is only valid while EOF is asserted); and TxCRC should be asserted high if the host system requires the ATT1MX10 to calculate and transmit the CRC checksum.



## Functional Description (continued)

### Transmit Overview (continued)

If a frame is less than 64 bytes, the ATT1MX10 will transmit out pad data to make the frame legal length if the PAD bit is set in the transmit frame configuration register. If the PAD bit is not set, the ATT1MX10 will only transmit the data that was written to the FIFO.

Also, the ATT1MX10 will keep track of multiple frames in the Tx FIFO and transmit them back-to-back in conformance with the ISO 8802.3 specification.

### Register and Counter Maps

The following tables list the register and counter mapping of the ATT1MX10. Note that the upper two address bits select registers that are paged.

Global registers must be accessed by driving the upper two address bits low. All other registers are accessed on pages where CPUAD[8:7] select the page as shown

in Table 9. Each page provides counters and registers for each port. The address map is shown in Table 10.

**Table 9. Port Addressing Format**

CPUAD[8:7]	Port
00	0
01	1
10	2
11	3

All registers are accessible on 16-bit boundaries. When reading 32-bit counters, multiple reads of the counter must be done by the host depending on the size of the CPU data bus. Since the CPU data bus is 16 bits, CPUAD0 is ignored.

When a counter needs to be read in more than one cycle, a snapshot of the counter memory location is stored in a register that guarantees that the count is valid. The byte order of the counters when they are read is shown in Table 10. Each register is described in detail in the following sections of this data sheet.

**Table 10. Register and Counter Address Maps**

CPUAD[8:7]	CPUAD[6:0]	Register/Counter Name	Read/Write (R/W)*
<b>Registers</b>			
00	0x00	Global Configuration Register	R/W
00	0x02	Revision ID Register	R
00:11	0x04	Transmit Frame Configuration Register	R/W
00:11	0x06	Receive Frame Configuration Register	R/W
00	0x08	Global FIFO Configuration Register	R/W
00	0x0a	Port Interrupt Identification Register	R
00:11	0x0c	Transmit Port Diagnostic Register	R
00:11	0x0e	Receive Port Diagnostic Register	R
00:11	0x10	Frame Interrupt Mask Register	R/W
00:11	0x12	Frame Interrupt Register	R
00:11	0x14	Counter Overflow Indication Register, High Word	R
00:11	0x16	Counter Overflow Indication Register, Low Word	R
00:11	0x18	CPU Interrupt Mask Register, High Word	R/W
00:11	0x1a	CPU Interrupt Mask Register, Low Word	R/W
00	0x1c	Reserved	NA
00	0x1e	Reserved	NA

\* NA = not applicable.

**Functional Description** (continued)**Register and Counter Maps** (continued)**Table 10. Register and Counter Address Map** (continued)

CPUAD[8:7]	CPUAD[6:0]	Register/Counter Name	Read/Write (R/W)*
<b>Transmit Event Counters</b>			
00:11	0x20	Total Octets Transmitted High (including errors)	R/W
00:11	0x22	Total Octets Transmitted Low (including errors)	R/W
00:11	0x24	Total Frames Transmitted High	R/W
00:11	0x26	Total Frames Transmitted Low	R/W
00:11	0x28	Total Collisions High	R/W
00:11	0x2a	Total Collisions Low	R/W
00:11	0x2c	Multicast Frames Transmitted	R/W
00:11	0x2e	Broadcast Frames Transmitted	R/W
00:11	0x30	Single Collisions	R/W
00:11	0x32	Late Collisions	R/W
00:11	0x34	Excess Deferrals	R/W
00:11	0x36	Transmit Retry Errors	R/W
00:11	0x38	Reserved	NA
<b>Receive Event Counters</b>			
00:11	0x40	Total Octets Received High (including errors)	R/W
00:11	0x42	Total Octets Received Low (including errors)	R/W
00:11	0x44	Misaligned Receive Frames High	R/W
00:11	0x46	Misaligned Receive Frames Low	R/W
00:11	0x48	Frames Received with Bad CRC High	R/W
00:11	0x4a	Frames Received with Bad CRC Low	R/W
00:11	0x4c	Total Octets Received High (not including errors)	R/W
00:11	0x4e	Total Octets Received Low (not including errors)	R/W
00:11	0x50	Frames Received High	R/W
00:11	0x52	Frames Received Low	R/W
00:11	0x54	Frames Received (64 octets) High (including errors)	R/W
00:11	0x56	Frames Received (64 octets) Low (including errors)	R/W
00:11	0x58	Rx Jabber Frames	R/W
00:11	0x5a	Multicast Frames Received	R/W
00:11	0x5c	Broadcast Frames Received	R/W
00:11	0x5e	Reserved	NA
00:11	0x60	Missed Frames	R/W
00:11	0x62	Reserved	NA
00:11	0x64	Reserved	NA
00:11	0x66	Fragments Received	R/W
00:11	0x68	Undersized Frames Received	R/W

\* NA = not applicable.

## Functional Description (continued)

### Register and Counter Maps (continued)

Table 10. Register and Counter Address Map (continued)

CPUAD[8:7]	CPUAD[6:0]	Register/Counter Name	Read/Write (R/W)*
<b>Receive Event Counters</b>			
00:11	0x6a	Frames Received (65 to 127 octets and errors)	R/W
00:11	0x6c	Frames Received (128 to 255 octets and errors)	R/W
00:11	0x6e	Frames Received (256 to 511 octets and errors)	R/W
00:11	0x70	Frames Received (512 to 1023 octets and errors)	R/W
00:11	0x72	Frames Received (1024 to 1518 octets and errors)	R/W
00:11	0x74	Long Frames Received (>1518 octets and no errors)	R/W
00:11	0x76	Reserved	NA

\* NA = not applicable.

The following table shows how 32-bit counters can be read through the 16-bit CPU data bus. The 16-bit counters are read in a similar format.

Table 11. 32-Bit Counter, 16-Bit CPU Data Bus

CPUAD	CPUDB[15:8]	CPUDB[7:0]
0x00	Byte 3 [MSB]*	Byte 2
0x02	Byte 1	Byte 0 [LSB]†

\* MSB = most significant byte.

† LSB = least significant byte.

## Functional Description (continued)

## Global Configuration Register

Address Offset = 0x00

The global configuration register is used to configure attributes of the ATT1MX10 that are common to all four ports. RESET default values are shown in parentheses. The ATT1MX10 will clear the appropriate reset bit in this register at the end of a software reset. Polling this register will allow the host to know when the software reset is complete.

Table 12. Global Configuration Register Bit Map

15	14	13	12	11	10	9	8
Reserved (1)	IntSel (0)	PSel (0)	PEn (0)	CntRes3 (0)	CntRes2 (0)	CntRes1 (0)	CntRes0 (0)
7	6	5	4	3	2	1	0
RxRes3 (0)	RxRes2 (0)	RxRes1 (0)	RxRes0 (0)	TxRes3 (0)	TxRes2 (0)	TxRes1 (0)	TxRes0 (0)

Table 13. Global Configuration Register Bit Descriptions

Bit	Name	Description
15	—	<b>Reserved.</b> This bit is reserved and must be 1.
14	IntSel	<b>Interrupt Select.</b> When this bit is set, the FRMINT signal will be disabled and routed to CPUINTR. When this bit is 0, all five interrupt signals operate individually.
13	PSel	<b>Parity Select.</b> This bit selects odd or even parity. When PSel is 0, parity will be even; the number of 1s on the data bus including the parity bit will be an even number. When PSel is 1, parity will be odd; the number of 1s on the data bus including the parity bit will be odd.
12	PEn	<b>Parity Enable.</b> This bit enables or disables parity generation. When PEn is 0, this function is disabled; when high, parity generation is enabled.
11:8	CntRes	<b>Counter Port Reset.</b> These bits provide individual counter port resets. Setting a particular CntRes bit will force a reset of all the counters on the specified port. At the end of reset, the ATT1MX10 will clear the appropriate CntRes bit.
7:4	RxRes	<b>Receive Port Reset.</b> These bits provide individual receive port resets. Setting a particular RxRes bit will force a reset of all the digital receive circuitry of that port except the counters. At the end of reset, the ATT1MX10 will clear the appropriate RxRes bit.
3:0	TxRes	<b>Transmit Port Reset.</b> These bits provide individual transmit port resets. Setting a particular TxRes bit will force a reset of all the digital transmit circuitry of that port except the counters. At the end of reset, the ATT1MX10 will clear the appropriate TxRes bit.

## Functional Description (continued)

### Revision ID Register

Address Offset = 0x02

This register is used to identify the revision of the ATT1MX10. Default values are shown in parentheses. These bits are read-only.

**Table 14. Revision ID Register Bit Map**

15	14	13	12	11	10	9	8
Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)
7	6	5	4	3	2	1	0
Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	RID3 (0)	RID2 (0)	RID1 (0)	RID0 (1)

**Table 15. Revision ID Register Bit Descriptions**

Bit	Name	Description
15:4	—	<b>Reserved.</b> These bits are reserved and must be 0.
3:0	RID	<b>Revision ID Nibble.</b> These 4 bits are read only and indicate the revision of the ATT1MX10.

**Functional Description** (continued)**Transmit Frame Configuration Register**

Address Offset = 0x04

The transmit frame configuration register defines the transmission rules for a specific port of the ATT1MX10. Note that there are four transmit frame configuration registers, one for each port. This register defaults to an ISO 8802.3 compliant mode after **RESET** is asserted. Default values are shown in parentheses. These can be changed to accommodate different options.

**Table 16. Transmit Frame Configuration Register Bit Map**

15	14	13	12	11	10	9	8
Reserved (0)	LBK (0)	FDup (0)	TxHalt (0)	HwCRC (1)	CRC (0)	Retry1 (0)	Retry0 (0)
7	6	5	4	3	2	1	0
EnSQE (1)	BSel (0)	Pad (1)	PreAm1 (0)	PreAm0 (0)	IgSQE (0)	Defer (1)	Enable (1)

**Table 17. Transmit Frame Configuration Register Bit Descriptions**

Bit	Name	Description
15	—	<b>Reserved.</b> This bit is reserved and must be 0.
14	LBK	<b>Loopback Enable.</b> The ATT1MX10 can be configured for loopback diagnostic operation by setting this bit. Loopback allows transmitted data to be looped back through the FIFOs of the ATT1MX10 in order to test host circuitry. Data will not be transmitted on the network regardless of the MEDIA bits. In loopback, data written to the Tx FIFO can be read back from the Rx FIFO.
13	FDup	<b>Full Duplex.</b> When this bit is set, the specific port will be in full duplex so that packets can be transmitted and received at the same time. In full duplex, the transmitter ignores receive carrier and collision indications before and during transmission. The AUI port cannot be configured for full duplex. When this bit is 0, the port will operate in normal 8802.3 half-duplex mode.
12	TxHalt	<b>Halt On Error Condition.</b> When this bit is set and an error occurs, the Enable bit will be cleared, halting TxFIFO. The FIFO can be restarted by resetting the enable bit in this register. The following error conditions can halt the Tx FIFO: excessive deferral, collision error, late collision, or a FIFO underrun. When TxHalt is 0, the Tx FIFO will continue to operate as normal even after error conditions.
11	HwCRC	<b>Hardware Select CRC.</b> When this bit is 1, the option to calculate and append CRC to each transmitted frame is determined by the TxCRC hardware signal. When HwCRC is 0, the CRC bit in the transmit frame configuration register determines whether or not the CRC is calculated and appended by the ATT1MX10. Note that if the pad bit in this register is set and the ATT1MX10 appends data to the end of a frame that is less than 64 bytes, the CRC will automatically be calculated and appended by the ATT1MX10, regardless of the state of this bit, the CRC bit, or the TxCRC pin input.
10	CRC	<b>CRC Enable.</b> This bit is valid when HwCRC is low. While CRC is 0, the CRC checksum is not calculated and appended to each transmitted frame. While CRC is 1, the CRC checksum is transmitted with each frame as the last four octets. This bit should only be set or reset during initialization.

**Functional Description** (continued)**Transmit Frame Configuration Register** (continued)

Address Offset = 0x04 (continued)

**Table 17. Transmit Frame Configuration Register Descriptions** (continued)

Bit	Name	Description															
9:8	Retry [1:0]	<p><b>Retry Attempt Select.</b> These bits set the number of attempts the transmitter makes to transmit a frame. After this number of attempts, the transmitter will abort the transmission of the frame resulting in an excessive collision error. They are selectable as:</p> <table> <tr> <th>Retry 1</th><th>Retry 0</th><th>Transmit Attempts</th></tr> <tr> <td>0</td><td>0</td><td>16 (default)</td></tr> <tr> <td>0</td><td>1</td><td>8</td></tr> <tr> <td>1</td><td>0</td><td>4</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	Retry 1	Retry 0	Transmit Attempts	0	0	16 (default)	0	1	8	1	0	4	1	1	1
Retry 1	Retry 0	Transmit Attempts															
0	0	16 (default)															
0	1	8															
1	0	4															
1	1	1															
7	EnSQE	<b>SQE Enable.</b> This bit enables SQE test for a port configured as twisted-pair. If set, a pulse is generated on COL after a successful transmission (SQE test). If clear, SQE test is disabled.															
6	BSel	<b>Backoff Select.</b> When this bit is set, the transmitter will only wait 9.6 $\mu$ s before retransmitting a frame after a collision. A random number of slot times is not calculated. When 0, the transmitter will function normally according to ISO 8802.3 transmission rules during collisions; a random number of slot times is calculated based on the following algorithm: backoff time = 9.6 $\mu$ s + 51.2 $\mu$ s * R, where $0 \leq R \leq 2^{\min(n, 10)}$ ; n is the number of collisions.															
5	Pad	<b>Pad Transmit Data.</b> While this bit is set, the Tx FIFO of the ATT1MX10 will automatically append data to a frame that is less than 64 bytes in length. Frames will be padded with a repeating F0 data pattern so that the frame length is equal to 64 bytes. If this bit is not set and a frame less than 64 bytes in length is DMAed to the Tx FIFO, this frame will be transmitted as is. Note that CRC will automatically be calculated and appended by the ATT1MX10 if this feature is enabled.															
4:3	PreAm [1:0]	<p><b>Preamble Bits Selection.</b> Setting these bits allows the user to select the number of preamble bits that the transmitter will send before the 8-bit SFD (10101011). The options are:</p> <table> <tr> <th>PreAm1</th><th>PreAm0</th><th>Number of PreAm Bits</th></tr> <tr> <td>0</td><td>0</td><td>56 (default)</td></tr> <tr> <td>0</td><td>1</td><td>48</td></tr> <tr> <td>1</td><td>0</td><td>40</td></tr> <tr> <td>1</td><td>1</td><td>32</td></tr> </table>	PreAm1	PreAm0	Number of PreAm Bits	0	0	56 (default)	0	1	48	1	0	40	1	1	32
PreAm1	PreAm0	Number of PreAm Bits															
0	0	56 (default)															
0	1	48															
1	0	40															
1	1	32															
2	IgSQE	<b>Ignore SQE Test.</b> When IgSQE is set, the transmitter will not wait for the SQE test on the COL input. The transmitter will immediately start the IFG timer and proceed with the next transmission. When IgSQE is 0, the transmitter will wait for the SQE test. Note that the IFG is timed in parallel with waiting for SQE test. In this case, if SQE test is not detected, the ATT1MX10 will set the SQE missed bit in the Tx diagnostic register.															
1	Defer	<b>Abort After Max Deferral.</b> When defer is set, the transmitter will abort a transmission if it is deferring for more than 24,288 bit times. The defer timer is reset at the beginning of each transmission attempt regardless of the number of collision retries. If defer is 0, the transmitter will wait indefinitely until the Ethernet media is available to accept a transmission.															
0	Enable	<b>Transmit Port Enable.</b> Setting this bit enables this particular MAC of the ATT1MX10 to transmit frames. If this bit is cleared during the transmission of a frame, the frame will be completely transmitted before the port is disabled. If a frame is being written to the Tx FIFO, EOF must be driven for the port to be disabled. If TxHalt is set and an error occurs, this bit will be cleared. It must be set again to enable the port transmitter.															

**Functional Description** (continued)**Receive Frame Configuration Register****Address Offset = 0x06**

The receive frame configuration register defines the rules for frame reception on a specific port of the ATT1MX10. There are four receive frame configuration registers, one for each port. This register resets to a default ISO 8802.3 compliant mode where each reception rule is in the compliant state. These can be changed to accommodate different options. RESET default values are shown in parentheses.

**Table 18. Receive Frame Configuration Register Bit Map**

<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
InhSIFG (0)	ID3 (0)	ID2 (0)	ID1 (0)	ID0 (0)	TPLink (1)	TPSque (0)	TPPol (1)
<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
Enable (1)	Media 1 (1)	Media 0 (0)	Reserved (0)	CRC (0)	UND (0)	FRAG (0)	FAE (0)

**Table 19. Receive Frame Configuration Register Bit Descriptions**

Bit	Name	Description						
15	InhSIFG	<b>Inhibit Short IFG.</b> This bit is used to adjust the minimum time after a transmission that a port responds to carrier. This implements the carrier inhibit timer function of the ISO 8802.3 specification.  <table><tr><td><b>InhSIFG</b></td><td><b>Time</b></td></tr><tr><td>0</td><td>≥200 ns</td></tr><tr><td>1</td><td>≥6.4 μs</td></tr></table>	<b>InhSIFG</b>	<b>Time</b>	0	≥200 ns	1	≥6.4 μs
<b>InhSIFG</b>	<b>Time</b>							
0	≥200 ns							
1	≥6.4 μs							
14:11	ID[3:0]	<b>Port ID Nibble.</b> These 4 bits are a programmable identification for the four ports of the ATT1MX10. These bits reset to 0 and are writable by the host. They are included in the receive port diagnostic register that is part of the end of packet statistics.						
10	TPLink	<b>Twisted-Pair Link Enable.</b> When this bit is set, the twisted-pair transceiver will generate and check link pulses. This bit is only valid while the media bits are 00.						
9	TPSque	<b>Twisted-Pair Squelch Enable.</b> When this bit is set, the twisted-pair receive squelch threshold will be reduced allowing extended cable to be used (>100 m). This bit is only valid while the media bits are 00.						
8	TPPol	<b>Twisted-Pair Auto Polarity Enable.</b> When this bit is set, the twisted-pair transceiver will monitor the polarity of the TP receive pair and automatically fix the polarity if it detects that it is reversed. When 0, the incorrect polarity will not be corrected. This bit is only valid while the media bits are 00. Counter values are invalid if TPPol = 0 and the polarity is reversed.						
7	Enable	<b>Receive Port FIFO Enable.</b> Setting this bit enables the FIFO of a particular port of the ATT1MX10 to receive frames. If Enable is set while a packet is being transmitted to the ATT1MX10, the packet will not be written to the FIFO. If this bit is cleared during the reception of a frame, the frame will be completely written to the FIFO before it is disabled. Regardless of the state of this bit, the event counters will continue to be updated.						



**Functional Description** (continued)**Receive Frame Configuration Register** (continued)

Address Offset = 0x06 (continued)

**Table 19. Receive Frame Configuration Register Bit Descriptions** (continued)

Bit	Name	Description															
6:5	Media[1:0]	<p><b>Media Select.</b> These 2 bits determine which media is selected for each port. The following options are available.</p> <table> <tr> <th>Media1</th><th>Media0</th><th>Mode of Operation</th></tr> <tr> <td>0</td><td>0</td><td>TP</td></tr> <tr> <td>0</td><td>1</td><td>AUI</td></tr> <tr> <td>1</td><td>0</td><td>MAC NRZ (Default)</td></tr> <tr> <td>1</td><td>1</td><td>Invalid</td></tr> </table>	Media1	Media0	Mode of Operation	0	0	TP	0	1	AUI	1	0	MAC NRZ (Default)	1	1	Invalid
Media1	Media0	Mode of Operation															
0	0	TP															
0	1	AUI															
1	0	MAC NRZ (Default)															
1	1	Invalid															
4	—	<b>Reserved.</b> This bit is reserved and must be a 0.															
3	CRC	<b>Reject CRC Error Frames.</b> When this bit is 0, the receiver will DMA the entire frame to the system even though there is a CRC error. If this bit is set, the receiver will discard the frame at the time the CRC error is detected providing the (RxSFTH – 4) bytes threshold has not been met. If the threshold has been met at the time that the ATT1MX10 determines there is a CRC error, the frame will be DMAed to the host (RxREQ asserted) and the Rx Status appended to the end of the frame will show a CRC error. This implies that the maximum size packet that will be rejected when this bit is 1 is (RxSFTH – 4) bytes where RxSFTH ≥ 68 bytes.															
2	UND	<b>Reject Undersized Frames.</b> When this bit is 0, the receiver will DMA the entire frame to the system even though the frame is undersized. If this bit is set, the receiver will discard the frame at the time the ATT1MX10 determines that the frame is undersized providing the (RxSFTH – 4) bytes threshold has not been met. If the threshold has been met, the ATT1MX10 will DMA the entire frame to the host (RxREQ asserted) and the Rx Status appended to the end of the frame will show an undersized frame error. This implies that the maximum size packet that will be rejected when this bit is 1 is (RxSFTH – 4) bytes where RxSFTH < 68 bytes.															
1	FRAG	<b>Reject Fragments.</b> When this bit is 0, the ATT1MX10 will DMA fragments to the host. If this bit is set, the ATT1MX10 will discard the frame at the time it determines the frame is a fragment, providing the (RxSFTH – 4) bytes threshold has not been met. If the threshold has been met, the ATT1MX10 will DMA the entire fragment to the host (RxREQ asserted) and the Rx Status appended at the end of the frame will show a fragment error. This implies that the maximum size packet that will be rejected when this bit is 1 is (RxSFTH – 4) bytes where RxSFTH < 68 bytes.															
0	FAE	<b>Reject Frame Alignment Errors.</b> When this bit is 0, the ATT1MX10 will DMA frames with alignment errors to the system. If this bit is set, the ATT1MX10 will discard the frame providing the (RxSFTH – 4) bytes threshold has not been met. If this threshold has been met, the ATT1MX10 will DMA the entire frame to the host (RxREQ asserted) and the Rx Status appended at the end of the frame will show an alignment error. This implies that the maximum size packet that will be rejected when this bit is 1 is (RxSFTH – 4) bytes where RxSFTH ≥ 68 bytes.															

**Functional Description** (continued)**Global FIFO Configuration Register****Address Offset = 0x08**

The global FIFO configuration register allows the user to configure the Tx and Rx FIFOs to match their system requirements. This register is globally set so that each of the four Rx FIFOs will have the same characteristics. The same is true for the four Tx FIFOs. See Table 39 for a complete description of the threshold settings and byte values. Reset default values are shown in parentheses.

**Table 20. Global FIFO Configuration Register Bit Map**

15	14:10	9:5	4:0
Reserved (0)	WCTH (00000)	TxSFTH (01111)	RxSFTH (01111)

**Table 21. Global FIFO Configuration Register Bit Descriptions**

Bit	Name	Description
15	—	<b>Reserved.</b>
14:10	WCTH	<b>Word Count Threshold.</b> This threshold is set for all four ports and controls when the $\overline{\text{TxABLE}}$ or $\overline{\text{RxREQ}}$ signal becomes active. This guarantees that at least WCTH words are free to either write to (Tx) or read from (Rx) the FIFO. This threshold should be set according to the burst capabilities of the system. WCTH is a binary number that equals the number of 4-byte words minus one. Note that, for receives, the WCTH is not valid until the RxSFTH has been met. Also, $\overline{\text{RxREQ}}$ will be asserted if a frame ends before the WCTH is met. For transmit, WCTH will always control the $\overline{\text{TxABLE}}$ signal.
9:5	TxSFTH	<b>Transmit Start of Frame Threshold.</b> This threshold is set for all four transmit ports and is used by the ATT1MX10 to indicate how many bytes should be written to the Tx FIFO before a transmit attempt is initiated. TxSFTH is a binary number that equals the number of 4-byte words minus one. If the frame size is less than the TxSFTH, the transmission will start after EOF is driven during the write cycle. This threshold can be set higher than the WCTH. $\text{TxSFTH} + \text{WCTH} \leq \text{Tx FIFO size} (= 128 \text{ bytes})$ .
4:0	RxSFTH	<b>Receive Start of Frame Threshold.</b> This threshold is set for all four receive ports and is used by the ATT1MX10 to indicate how many bytes of a frame must be received before the first DMA request is made by asserting $\overline{\text{RxREQ}}$ . RxSFTH is a binary number that equals the number of 4-byte words minus one. This threshold can be set higher than the WCTH so that undersized or fragment frames can be rejected. Once the RxSFTH is met, $\overline{\text{RxREQ}}$ is driven when the WCTH is met.

**Functional Description** (continued)**Port Interrupt Identification Register****Address Offset = 0x0a**

This register is used to identify which port caused the  $\overline{\text{CPUINTR}}$  signal to become asserted. This register should be read before reading one of the counter overflow indication registers. This register is cleared when read by the host.

**Table 22. Port Interrupt Identification Register Bit Map**

15:4	3	2	1	0
Reserved	P3	P2	P1	P0

**Table 23. Port Interrupt Identification Register Bit Descriptions**

Bit	Name	Description
15:4	—	<b>Reserved.</b> This bit is reserved and must be a 0.
3	P3	<b>Port 3 Interrupt.</b> When this bit is set, the $\overline{\text{CPUINT}}$ signal assertion was caused by a counter overflow on port 3.
2	P2	<b>Port 2 Interrupt.</b> When this bit is set, the $\overline{\text{CPUINT}}$ signal assertion was caused by a counter overflow on port 2.
1	P1	<b>Port 1 Interrupt.</b> When this bit is set, the $\overline{\text{CPUINT}}$ signal assertion was caused by a counter overflow on port 1.
0	P0	<b>Port 0 Interrupt.</b> When this bit is set, the $\overline{\text{CPUINT}}$ signal assertion was caused by a counter overflow on port 0.

**Functional Description** (continued)**Transmit Port Diagnostic Register****Address Offset = 0x0c**

The transmit port diagnostic register is updated after a frame is fully transmitted or the transmission of a frame is aborted due to an error. The register can be read to determine if the frame was successfully transmitted or to determine what errors occurred. This register should only be read during diagnostic testing unless the system can guarantee that it is read before a new frame is transmitted from the port. Reset default values are shown in parentheses.

**Table 24. Transmit Port Diagnostic Register Bit Map**

15	14	13	12	11	10	9	8
Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	PARITY (0)	LATE (0)	ExDEF (0)
7	6	5	4	3	2	1	0
DEF (0)	SCOL (0)	MCOL (0)	CERR (0)	LCar (0)	SQE (0)	FIFO (0)	GOOD (0)

**Table 25. Transmit Port Diagnostic Register Bit Descriptions**

Bit	Name	Description
15:11	—	<b>Reserved.</b> These bits are reserved and must be a 0.
10	PARITY	<b>Parity Error.</b> This bit will be set if there was a parity error on the data bus while the host is writing to the transmit FIFO.
9	LATE	<b>Late Collision.</b> This bit will be set when a transmission is aborted due to a collision occurring after the first 64 bytes of a frame have been transmitted. If TxHalt (in the transmit frame configuration register) is set, the occurrence of a late collision will halt the Tx FIFO. This bit is not valid while the corresponding port is configured for full duplex.
8	ExDEF	<b>Excessive Deferral.</b> This bit will be set if the transmission was aborted because of an excessive deferral as defined by the Defer bit in the transmit frame configuration register. If TxHalt (in transmit frame configuration register) is set, excessive deferring will halt the Tx FIFO. This bit is not valid while the corresponding port is configured for full duplex.
7	DEF	<b>Deferred.</b> This bit will be set if the frame transmission was delayed because of a deferral. This bit will be set when a frame is transmitted with a collision and the BSEL bit in the transmit frame configuration register is set (i.e., standard backoff is selected). This bit is not valid while the corresponding port is configured for full duplex.
6	SCOL	<b>Single Collision.</b> This bit will be set if the frame being transmitted collided once and was then transmitted successfully on the Ethernet. This bit will not be set if the transmission is aborted due to excess collisions or there are multiple collisions. This bit is not valid while the corresponding port is configured for full duplex.
5	MCOL	<b>Multiple Collisions.</b> This bit will be set if the frame being transmitted collided more than once and was then transmitted successfully on the Ethernet. This bit will not be set if the transmission is aborted due to excess collisions or there is a single collision. This bit is not valid while the corresponding port is configured for full duplex.

## Functional Description (continued)

### Transmit Port Diagnostic Register (continued)

Address Offset = 0x0c (continued)

Table 25. Transmit Port Diagnostic Register Bit Descriptions (continued)

Bit	Name	Description
4	CERR	<b>Collision Error.</b> This bit will be set if the frame transmission was aborted because of too many collisions. The number of collision retries allowed is specified in the transmit frame configuration register. If TxHalt (in transmit frame configuration register) is set, a collision error will halt the Tx FIFO. This bit is not valid while the corresponding port is configured for full duplex.
3	LCar	<b>Loss of Carrier.</b> This bit will be set if the CRD (for MAC NRZ) input is low at any time while COL is high during the transmission of a frame. (Internal signals are monitored while the port is configured for AUI or TP.) The loss of carrier may indicate that the connection to the MAC, AUI, or twisted-pair is not valid. (Data is not being looped back from the media.) This bit is not valid when the ATT1MX10 is configured for full duplex.
2	SQE	<b>Signal Quality Error Missed.</b> This bit will be set if the SQE test on the COL[3:0] signal line is not detected at the end of a transmission. This bit will only be set if the media bits in the receive frame configuration register are 01 or 10 and the SQE bit in the transmit frame configuration register is 0. This bit is not valid while the corresponding port is configured for loop-back.
1	FIFO	<b>FIFO Underrun.</b> This bit will be set if the transmitter runs out of data in the Tx FIFO while the port is transmitting. If TxHalt (in transmit frame configuration register) is set, the Tx FIFO will halt when it underruns.
0	GOOD	<b>Good Frame.</b> This bit will be set if the frame is transmitted without any error.

**Functional Description** (continued)**Receive Port Diagnostic Register****Address Offset = 0x0e**

The receive port diagnostic register is updated every time a frame is received or there is a status change in the twisted-pair transceiver. This register is also part of the end of frame statistics that are DMAed at the end of each frame. Error bits are set regardless of the state of the receive port diagnostic register. Note that this register is updated each time a frame is received and multiple frames can be stored in the Rx FIFO of each port. Therefore, reading this register in real time will not guarantee that the status read will represent the current frame being read from the DMA channel. This register is intended for diagnostic purposes and for end of frame statistics.

**Table 26. Receive Port Diagnostic Register Bit Map**

<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
ID3 (0)	ID2 (0)	ID1 (0)	ID0 (0)	AutoP (0)	LStat (0)	RxJab (0)	FAE (0)
<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
CRC (0)	UND (0)	FRAG (0)	LONG (0)	PHYS (0)	MULT (0)	BRD (0)	FIFO (0)

**Table 27. Receive Port Diagnostic Register Bit Descriptions**

Bit	Name	Description
15:12	ID[3:0]	<b>Port ID Nibble.</b> The four-port ID nibble bits are copied from the receive port diagnostic register, which is programmed by the host.
11	AutoP	<b>Polarity Change.</b> This bit will be set when the media bits in the receive port diagnostic register are 00 (twisted-pair configuration), the TPPol bit is set, and the ATT1MX10 flips the polarity of the TP receive signal pairs because the TP polarity is incorrect. Otherwise, this bit will be 0.
10	LStat	<b>Current Link Status.</b> While media is 00 (twisted-pair configuration), this bit represents the state of the link. When LStat = 0, the receiver is not receiving link pulses. While LStat = 1, the receiver is receiving link pulses properly.
9	RxJab	<b>Jabber on Reception.</b> This bit will be set when the current receive frame is greater than 1518 octets and has either a bad CRC with an integral number of octets or a bad CRC with a nonintegral number of octets.
8	FAE	<b>Frame Alignment Error.</b> This bit will be set if the current frame being received has a frame alignment error. An FAE occurs when the resultant remainder from the division between the number of bits in the frame and 8 is nonzero (nonintegral number of octets), the FCS is invalid, and the octet count is greater than or equal to 64, and less than or equal to 1518 octets.
7	CRC	<b>CRC Error.</b> This bit is set if the frame being received has a CRC error with an integral number of octets and the octet count is greater than or equal to 64 and less than or equal to 1518 octets.
6	UND	<b>Undersized Frame.</b> This bit is set if the frame being received is less than 64 octets, not including preamble, and is otherwise well formed.
5	FRAG	<b>Fragment.</b> This bit is set if the frame being received is less than 64 octets, not including preamble, has a valid SFD, and has either an invalid CRC with an integral number of octets or an invalid CRC with a nonintegral number of octets.

## Functional Description (continued)

### Receive Port Diagnostic Register (continued)

Address Offset = 0x0e (continued)

Table 27. Receive Port Diagnostic Register Bit Descriptions (continued)

Bit	Name	Description
4	LONG	<b>Frame Long Error.</b> This bit is set if the frame is beyond the legal length of 1518 octets and is otherwise well formed.
3	PHYS	<b>Physical Address.</b> This bit will be set if the destination address of the frame is a physical address. (A physical address may also be referred to as a unicast address.)
2	MULT	<b>Multicast Address.</b> This bit will be set if the destination address of the frame is a multicast (group) address. This does not include broadcast frames.
1	BRD	<b>Broadcast Address.</b> This bit will be set if the destination address of the frame is the broadcast address. This does not include multicast frames.
0	FIFO	<b>FIFO Overrun.</b> This bit will be set if, for some reason external to the ATT1MX10, the Rx FIFO for the associated port is not serviced in time and the frame is not buffered.

## Functional Description (continued)

## Frame Interrupt Mask Register

Address Offset = 0x10

This register masks the interrupt bit from the frame interrupt register to drive the  $\overline{\text{FRMINT}}$  signal. When an event occurs that sets one of the bits in the frame interrupt register, the  $\overline{\text{FRMINT}}$  signal will become active if the corresponding mask bit in this register is 0. When the bit in this register is set, the  $\overline{\text{FRMINT}}$  signal will not be driven when an event occurs in the frame interrupt register. Reset default values are shown in parentheses.

Table 28. Frame Interrupt Mask Register Bit Map

15	14	13	12	11	10	9	8
LATEM (1)	ExDEFM (1)	CERRM (1)	TFIFOM (1)	AutoPM (1)	LStatChM (1)	RxJabM (1)	RFIFOM (1)
7	6	5	4	3	2	1	0
ParityM (1)	Reserved (1)	Reserved (1)	Reserved (1)	Reserved (1)	Reserved (1)	Reserved (1)	Reserved (1)

Table 29. Frame Interrupt Mask Register Bit Descriptions

Bit	Name	Description
15	LATEM	<b>Late Collision Mask.</b> While this bit is 0, $\overline{\text{FRMINT}}$ will be driven active when LATE is set in the frame interrupt register. While this bit is 1, $\overline{\text{FRMINT}}$ will not be affected by the state of LATE.
14	ExDEFM	<b>Excessive Deferral Mask.</b> While this bit is 0, $\overline{\text{FRMINT}}$ will be driven active when ExDEF is set in the frame interrupt register. While this bit is 1, $\overline{\text{FRMINT}}$ will not be affected by the state of ExDEF.
13	CERRM	<b>Collision Error Mask.</b> While this bit is 0, $\overline{\text{FRMINT}}$ will be driven active when CERR is set in the frame interrupt register. While this bit is 1, $\overline{\text{FRMINT}}$ will not be affected by the state of CERR.
12	TFIFOM	<b>Transmit FIFO Underrun Error Mask.</b> When this bit is 0, $\overline{\text{FRMINT}}$ will be driven active-low when the TFIFO bit in the frame interrupt register is set. While this bit is 1, $\overline{\text{FRMINT}}$ will not be affected by the state of TFIFO.
11	AutoPM	<b>Polarity Change Mask.</b> While this bit is 0, $\overline{\text{FRMINT}}$ will be driven active when AutoPM is set in the frame interrupt register. While this bit is 1, $\overline{\text{FRMINT}}$ will not be affected by the state of AutoPM. This bit should be programmed to 1 for AUI and MAC mode configurations.
10	LStatChgM	<b>Change in TP Link Status Mask.</b> While this bit is 0, $\overline{\text{FRMINT}}$ will be driven active when LStatChg is set in the frame interrupt register. While this bit is 1, $\overline{\text{FRMINT}}$ will not be affected by the state of LStatChg. This bit should be programmed to 1 for AUI and MAC mode configurations.
9	RxJabM	<b>Jabber on Reception Mask.</b> While this bit is 0, $\overline{\text{FRMINT}}$ will be driven active when RxJab is set in the frame interrupt register. While this bit is 1, $\overline{\text{FRMINT}}$ will not be affected by the state of RxJab.
8	RFIFOM	<b>Receive FIFO Overrun Mask.</b> While this bit is 0, $\overline{\text{FRMINT}}$ will be driven active when RFIFO is set in the frame interrupt register. While this bit is 1, $\overline{\text{FRMINT}}$ will not be affected by the state of RFIFO.
7	PARITYM	<b>Parity Error Mask.</b> While this bit is 0, $\overline{\text{FRMINT}}$ will be driven active-low when the Parity bit is set in the frame interrupt register. While this bit is 1, $\overline{\text{FRMINT}}$ will not be driven based on the state of parity.
6:0	—	<b>Reserved.</b> This bit is reserved and must be a 0.



**Functional Description** (continued)**Frame Interrupt Register****Address Offset = 0x12**

The events in this register will cause an interrupt event to occur on  $\overline{\text{FRMINT}}$  if the event is enabled by setting the appropriate bit in the frame interrupt mask register. The bits in this register are cleared when the register is read.  $\overline{\text{FRMINT}}$  will also be deasserted when this register is read. Reset default values are shown in parentheses. Note that there are four  $\overline{\text{FRMINT}}$  signals and four of these registers. During initialization, this register must be read to clear it before normal operation proceeds.

**Table 30. Frame Interrupt Register Bit Map**

<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
LATE (0)	ExDEF (0)	CERR (0)	TFIFO (0)	AutoP (0)	LStatChg (0)	RxJab (0)	RFIFO (0)
<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
PARITY (0)	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)

**Table 31. Frame Interrupt Mask Register Bit Descriptions**

Bit	Name	Description
15	LATE	<b>Late Collision.</b> This bit will be set when the LATE bit is set in the transmit port diagnostic register. $\overline{\text{FRMINT}}$ will be driven active if this event is unmasked.
14	ExDEF	<b>Excessive Deferral.</b> This bit will be set when the ExDEF bit is set in the transmit port diagnostic register. $\overline{\text{FRMINT}}$ will be driven active if this event is unmasked.
13	CERR	<b>Collision Error.</b> This bit will be set when the CERR is set in the transmit port diagnostic register. $\overline{\text{FRMINT}}$ will be driven active if this event is unmasked.
12	TFIFO	<b>Transmit FIFO Underrun Error.</b> This bit will be set when the FIFO bit in the transmit port diagnostic register is set. $\overline{\text{FRMINT}}$ will be driven active if this event is unmasked.
11	AutoP	<b>Polarity Change.</b> This bit will be set when the AutoP bit is set in the port diagnostic register. $\overline{\text{FRMINT}}$ will be driven active if this event is unmasked.
10	LStatChg	<b>Change in TP Link Status.</b> This bit will be set when the LStat bit changes state in the receive port diagnostic register. $\overline{\text{FRMINT}}$ will be driven active if this event is unmasked.
9	RxJab	<b>Jabber on Reception.</b> This bit will be set when the RxJab bit is set in the receive port diagnostic register. $\overline{\text{FRMINT}}$ will be driven active if this event is unmasked.
8	RFIFO	<b>Receive FIFO Overrun.</b> This bit will be set when the FIFO bit is set in the receive port diagnostic register. $\overline{\text{FRMINT}}$ will be driven active if this event is unmasked.
7	PARITY	<b>Parity Error.</b> This bit will be set when there is a parity error on the data bus while the host is writing data to the ATT1MX10 Tx FIFO. This bit will remain low when the parity function is disabled. This bit is not set when the host is reading from the Rx FIFO and there is a parity error.
6:0	—	<b>Reserved.</b> This bit is reserved and must be a 0.

**Functional Description** (continued)**Counter Overflow Indication Register****Address Offset = 0x14 (High Word), 0x16 (Low Word)**

This register can be used to identify any counters that have reached the overflow threshold and need to be read. When read after an interrupt, this register will indicate which counter(s) caused the interrupt. This register is port specific.

The counter overflow indication register is a 32-bit register:

- The data bus bits 31:16 can be read from the high word.
- The data bus bits 15:0 can be read from the low word.

This register is a clear-on-read register. This register must be read during initialization in order to initialize it. Reset default values are shown in parentheses. The counters are described in detail in the Counter Definitions section of this data sheet.

**Table 32. Counter Overflow Indication Register Bit Map—High Word**

31	30	29	28	27	26	25	24
Reserved (0)	RxTotOct (0)	RxJab (0)	TxRty (0)	— (0)	ExDef (0)	RxFrag (0)	RxUnd (0)
23	22	21	20	19	18	17	16
RxCRC (0)	RxLong (0)	RxF (0)	RxE (0)	RxD (0)	RxC (0)	RxB (0)	RxA (0)

**Table 33. Counter Overflow Indication Register Bit Map—Low Word**

15	14	13	12	11	10	9	8
RxFA (0)	Reserved (0)	RxBCM (0)	RxMC (0)	TxBC (0)	TxMC (0)	TxLCOL (0)	RxOct (0)
7	6	5	4	3	2	1	0
TxSCol (0)	Reserved (0)	TxOct (0)	Reserved (0)	RxMF (0)	RxFrm (0)	TxCol (0)	TxFrm (0)

## Functional Description (continued)

### Counter Overflow Indication Register (continued)

Address Offset = 0x14 (High Word), 0x16 (Low Word) (continued)

Table 34. Counter Overflow Indication Register Bit Descriptions

Bit	Name	Description
<b>High Word</b>		
31	—	<b>Reserved.</b> This bit is reserved.
30	RxTotOct	Total Octets Received Counter Overflow.
29	RxJab	Jabber Frames Received Counter Overflow.
28	TxRty	Transmit Retry Errors Counter Overflow.
27	—	<b>Reserved.</b> This bit is reserved.
26	ExDef	Excessive Deferrals Counter Overflow.
25	RxFrag	Fragments Received Counter Overflow.
24	RxUnd	Undersized Frames Received Counter Overflow.
23	RxCRC	Frames Received with Bad CRC Counter Overflow.
22	RxLong	Long Frames Received Counter Overflow.
21	RxF	Frames Received (1024 to 1518 Octets) Counter Overflow.
20	RxE	Frames Received (512 to 1023 Octets) Counter Overflow.
19	RxD	Frames Received (256 to 511 Octets) Counter Overflow.
18	RxC	Frames Received (128 to 255 Octets) Counter Overflow.
17	RxB	Frames Received (65 to 127 Octets) Counter Overflow.
16	RxA	Frames Received (64 Octets) Counter Overflow.
<b>Low Word</b>		
15	RxFA	Misaligned Frames Counter Overflow.
14	—	<b>Reserved.</b> This bit is reserved.
13	RxBC	Broadcast Frames Received Counter Overflow.
12	RxMC	Multicast Frames Received Counter Overflow.
11	TxBC	Broadcast Frames Transmitted Counter Overflow.
10	TxMC	Multicast Frames Transmitted Counter Overflow.
9	TxLCol	Late Collisions Counter Overflow.
8	RxOct	Total Good Octets Received Counter Overflow.
7	TxSCol	Single Collisions Counter Overflow.
6	—	<b>Reserved.</b> This bit is reserved.
5	TxOct	Total Octets Transmitted Counter Overflow.
4	—	<b>Reserved.</b> This bit is reserved.
3	RxMF	Missed Frames Counter Overflow.
2	RxFrm	Total Good Frames Received Counter Overflow.
1	TxCol	Total Collisions Counter Overflow.
0	TxFrm	Total Good Frames Transmitted Counter Overflow.

**Functional Description** (continued)**CPU Interrupt Mask Register****Address Offset = 0x18 (High Word), 0x1a (Low Word)**

This register is used to control which counters drive the  $\overline{\text{CPUINTR}}$  when they roll over. Clearing these bits will allow the corresponding counter to drive the  $\overline{\text{CPUINTR}}$  signal when the counter reaches FFFF (for 16-bit counters) or FFFFFFFF (for 32-bit counters). Note that when a counter reaches its terminal count, it will roll over to 0 on the next event. Setting a mask bit will disable the corresponding counter overrun to drive the  $\overline{\text{CPUINTR}}$  signal. Note that all counters will roll over and keep counting. Counters values are valid regardless of the state of these register bits. Reset default values are shown in parentheses.

**Table 35. CPU Interrupt Mask Register Bit Map—High Word**

31	30	29	28	27	26	25	24
Reserved (1)	RxTotOctM (1)	RxJabM (1)	TxRtyM (1)	Reserved (1)	ExDefM (1)	RxFragM (1)	RxUndM (1)
23	22	21	20	19	18	17	16
RxCRCM (1)	RxLongM (1)	RxFM (1)	RxEM (1)	RxDM (1)	RxCM (1)	RxBM (1)	RxAM (1)

**Table 36. CPU Interrupt Mask Register Bit Map—Low Word**

15	14	13	12	11	10	9	8
RxFAM (1)	Reserved (1)	RxBCM (1)	RxMCM (1)	TxBCM (1)	TxMCM (1)	TxLCOLM (1)	RxOctM (1)
7	6	5	4	3	2	1	0
TxSColM (1)	Reserved (1)	TxOctM (1)	Reserved (1)	RxMFM (1)	RxFrmM (1)	TxColM (1)	TxFrmM (1)

## Functional Description (continued)

### CPU Interrupt Mask Register (continued)

Address Offset = 0x18 (High Word), 0x1a (Low Word) (continued)

Table 37. CPU Interrupt Mask Register Bit Descriptions

Bit	Name	Description
<b>High Word</b>		
31	—	<b>Reserved.</b> This bit is reserved and must be set.
30	RxTotOctM	Total Octets Received Counter Mask.
29	RxJabM	Jabber Frames Received Counter Mask.
28	TxRtyM	Transmit Retry Errors Counter Mask.
27	—	<b>Reserved.</b> This bit is reserved and must be set.
26	ExDefM	Excessive Deferral Counter Mask.
25	RxFragM	Fragments Received Counter Mask.
24	RxUndM	Undersized Frames Received Counter Mask.
23	RxCRCM	Frames Received with Bad CRC Counter Mask.
22	RxLongM	Long Frames Received Counter Mask.
21	RxFM	Frames Received (1024 to 1518 Octets) Counter Mask.
20	RxEM	Frames Received (512 to 1023 Octets) Counter Mask.
19	RxDM	Frames Received (256 to 511 Octets) Counter Mask.
18	RxCM	Frames Received (128 to 255 Octets) Counter Mask.
17	RxBM	Frames Received (65 to 127 Octets) Counter Mask.
16	RxAM	Frames Received (64 Octets) Counter Mask.
<b>Low Word</b>		
15	RxFAM	Misaligned Frames Counter Mask.
14	—	<b>Reserved.</b> This bit is reserved and must be set.
13	RxBCM	Broadcast Frames Received Counter Mask.
12	RxMCM	Multicast Frames Received Counter Mask.
11	TxBMCM	Broadcast Frames Transmitted Counter Mask.
10	TxMCM	Multicast Frames Transmitted Counter Mask.
9	TxLCoIM	Late Collisions Counter Mask.
8	RxOctM	Total Good Octets Received Counter Mask.
7	TxSCoIM	Single Collisions Counter Mask.
6	—	<b>Reserved.</b> This bit is reserved and must be set.
5	TxOctM	Total Octets Transmitted Counter Mask.
4	—	<b>Reserved.</b> This bit is reserved and must be set.
3	RxMFM	Missed Frames Counter Mask.
2	RxFrmM	Total Good Frames Received Counter Mask.
1	TxCollM	Total Collisions Counter Mask.
0	TxFrmM	Total Good Frames Transmitted Counter Mask.

## Functional Description (continued)

### DMA Interface

The ATT1MX10 provides a slave DMA interface that is capable of reading or writing data at 100 Mbytes/s. The ATT1MX10 uses eight signals to request FIFO access; each of the four ports has a receive FIFO and transmit FIFO request signal. These signals are under the control of the FIFO threshold settings in the global FIFO configuration register. All other control is driven by the host DMA subsystem.

### Common Tx and Rx Functions

Although the transmit FIFO and receive FIFO are under individual control, the two functions share several of the DMA signals. (For full-duplex systems, the DATA bus is time multiplexed.) This allows the fastest possible access time with the least number of pins. The direction of the data bus can be changed through the RTSEL input.

The data bus is a 32-bit bidirectional bus. All control on the data bus is synchronous to HCLK. HCLK must be between 16 MHz and 25 MHz. The data bus also has a parity bit. Parity generation and checking can be set to odd, even, or disabled.

### Chip and Channel Selection

In the event that a multiple port product is designed and more than one ATT1MX10 is used in a subsystem, a chip select signal is provided to enable the data bus. While  $\overline{PFCS}$  is high, the data bus will be in high impedance.

Beyond the chip select, a particular port within an ATT1MX10 must also be selected. This is done by driving the CHSEL[2:0] inputs. When driven in the 1XX state, no channel is selected and the bus is 'parked'. The ATT1MX10 continues to drive the data bus when CHSEL = 1XX and RTSEL = 1; however, no DMA activity is permitted by the ATT1MX10.

A transmit or receive channel must be deselected before the ATT1MX10 can assert the corresponding  $\overline{TxABLE}$  or  $RxREQ$ .

When using more than one ATT1MX10 in a system, no bus activity can occur when switching control from one chip to the next. Therefore,  $\overline{RxUNLD}$  (for RX) and  $\overline{TxRDY}$  (for Tx) cannot be asserted while switching control. The bus does not have to be parked by forcing CHSEL = 1XX.  $\overline{PFCS}$  may remain asserted when switching channels. When the most significant CHSEL bit is 0, the lower 2 bits determine which port is being selected. For example, 000 selects port 0, and 010 selects port 2.

### Start and End of Frame Notification

Each frame within the ATT1MX10, whether transmit or receive, is delineated by two signals on the data bus: SOF indicates that the current word is the first word of a frame, and EOF indicates that the current word is the last word of a frame.

These signals are bidirectional. During a write to the Tx FIFO, the host drives SOF and EOF to bound a frame. During a read of the Rx FIFO, the ATT1MX10 drives these signals to bound the frame. For short frames, SOF and EOF can be asserted during the same burst, by either the ATT1MX10 or the host. Also, SOF and EOF can be asserted on the same HCLK edge.

### Valid Byte (VB) Alignment

Most frames will not end on an even word boundary. To facilitate this, the ATT1MX10 uses the VB[1:0] signals to indicate how many bytes within a word contain valid frame data. It requires that all DMA transfers are word aligned except for the last word of the frame. Therefore, VB1 and VB0 are only valid when EOF is asserted. These bidirectional signals should be driven by the host during the last write to the Tx FIFO, and read by the host during the last read of the Rx FIFO. The states of VB[1:0] that determine which bytes of a word are valid are described in Table 38.

Table 38. Valid Byte (VB) Alignment

VB1	VB0	DATA 31:24	DATA 23:16	DATA 15:8	DATA 7:0	EOF
0	0	Valid	—	—	—	1
0	1	Valid	Valid	—	—	1
1	0	Valid	Valid	Valid	—	1
1	1	Valid	Valid	Valid	Valid	1
x	x	Valid	Valid	Valid	Valid	0

### Tx Specific Functions

The Tx FIFO uses a two-signal handshake for request and acknowledge synchronization. The first signal,  $\overline{TxABLE}$ , is asserted low by the ATT1MX10 when the Tx FIFO is able of having data written to it. The amount of data is determined by the word threshold count (WCTH). For example, if the WCTH is equal to 48 bytes and there are at least 48 bytes empty in the Tx FIFO,  $\overline{TxABLE}$  is asserted by the ATT1MX10.

The second signal,  $\overline{TxRDY}$ , is driven by the host to drive data into the Tx FIFO. The burst size will be, at most, the number of bytes represented by WCTH.  $\overline{TxABLE}$  will be deasserted one HCLK cycle after the transmit channel is asserted.  $\overline{TxRDY}$  is qualified by the ATT1MX10 with  $\overline{PFCS}$ , CHSEL, and RTSEL.

## Functional Description (continued)

### DMA Interface (continued)

#### CRC Generation

The Tx FIFO has the capability of either transmitting a frame with or without the CRC checksum calculated and appended by the ATT1MX10. This option can be turned on or off two ways based on the status of the HwCRC bit in the transmit frame configuration register. If HwCRC = 0, the CRC bit in the transmit frame configuration register determines whether the transmitter will calculate and append CRC or not. The CRC bit cannot be changed dynamically. If HwCRC = 1, TxCRC is qualified with PFCS, CHSEL, EOF, and RTSEL. The transmitter will calculate and append CRC based on the state of this input. This method allows the host to calculate and append CRC dynamically.

When there is a FIFO underrun or parity error, the ATT1MX10 will transmit a packet with a CRC error regardless of the state of the CRC bit or the TxCRC signal. Also, if the pad bit in the transmit frame configuration register is set, the CRC will always be calculated and appended if the ATT1MX10 pads a frame that is less than 64 bytes in length.

#### Rx Specific Functions

The receive FIFO uses a similar handshake as the transmit FIFO with the addition of one signal. When a new frame is received by the ATT1MX10, the first  $\overline{RxREQ}$  is controlled by the receive start of frame threshold (RxSFTH). This threshold is set in the global FIFO configuration register and determines how many bytes of data from a new frame must be received before the initial DMA request ( $\overline{RxREQ}$  low) is made by the ATT1MX10. Once this threshold is met, the entire frame needs to be read out of the ATT1MX10, regardless of any error conditions and the bits set in the receive frame configuration register. If the ATT1MX10 detects any errors before the (RxSFTH - 4) bytes are met, the ATT1MX10 will discard the frame depending on the state of the appropriate configuration bit in the receive frame configuration register.

For example, if the RxSFTH is set to 64 bytes, the ATT1MX10 receives a frame and detects a CRC error after 32 bytes are received into the Rx FIFO and the CRC bit is set in the receive frame configuration register, the ATT1MX10 will not assert the  $\overline{RxREQ}$  signal for this frame. Instead, the frame will be discarded by the ATT1MX10, and the appropriate receive event counters will be incremented. If, however, the CRC error is detected after 68 bytes are received,  $\overline{RxREQ}$  will be asserted and the frame will have to be read by the host.

Once the RxSFTH is met, each successive assertion of  $\overline{RxREQ}$  will be based on the WCTH, which may or may not be the same as the RxSFTH. In most cases, the WCTH will be equal to the burst size of the host DMA while the RxSFTH is used to filter frames with errors.

In order to read data from the ATT1MX10, the host must assert  $\overline{RxUNLD}$  low at some time after the ATT1MX10 asserts  $\overline{RxREQ}$ . (The maximum time that the host has to service the  $\overline{RxREQ}$  is dependent on the RxSFTH and whether or not this is the only frame in the FIFO.) Along with  $\overline{RxUNLD}$ , the host must drive PFCS low and RTSEL high, and select the proper channel with CHSEL[2:0]. CHSEL[2:0] must be held true until the ATT1MX10 drives  $\overline{RxRDY}$  low.  $\overline{RxRDY}$  indicates that data is valid on the data bus.  $\overline{RxREQ}$  will be asserted if the WCTH is met after the Rx channel is deselected.

If a frame ends before a threshold is met,  $\overline{RxREQ}$  will automatically be asserted, unless the frame has not met the RxSFTH, the frame has an error, and the ATT1MX10 is configured to reject frames with the same error.

#### End of Frame Status

After the frame data has been read from the Rx FIFO, the end of frame status will be DMAed to the host. The end of frame status consists of the frame byte length and a copy of the receive port diagnostic register. The frame byte length is 11 bits.

The end of frame status will always be word aligned, regardless of where frame data ends. End of frame status is valid on the HCLK cycle after EOF is driven by the ATT1MX10. The byte count is driven on DATA[32:16] and status is driven on DATA[15:0].

EOF status is written in the FIFO after the frame. While  $\overline{RxUNLD}$  is driven low and  $\overline{RxRDY}$  is asserted low by the ATT1MX10, EOF status will be driven on the DATA bus on the rising edge of HCLK following the assertion of EOF. If  $\overline{RxUNLD}$  is deasserted by the host after the EOF signal is detected high, EOF status will not be driven on the DATA bus. In this case, after the channel is deselected,  $\overline{RxREQ}$  will be reasserted, indicating that there is status to be read. Note that  $\overline{RxREQ}$  will be asserted even though there may not be any data, other than EOF status in the FIFO. The host can read status by selecting the chip and channel and asserting  $\overline{RxUNLD}$ .  $\overline{RxUNLD}$  should be deasserted after the status is read.  $\overline{RxREQ}$  will be asserted based on RxSFTH when the next frame arrives.

**Functional Description** (continued)**FIFO Operation**

FIFOs are used internally to buffer frames before they are transmitted on the network or before they are read by the host. The FIFO thresholds need to be set to handle latencies associated with the arbitration of multiple ports. The Tx FIFO is deep enough to support the retransmission of a frame if a collision occurs within the first 51.2  $\mu$ s of transmission. The Rx FIFO is deep enough to filter undersized and fragment frames without having to interrupt the host DMA system. A maximum amount of FIFO control is governed by the host, creating a flexible interface for various applications.

**Common Tx and Rx FIFO Functions**

There are a couple of functions that are shared by the Tx and Rx FIFOs, each of which are described in the following sections.

**Word Count Threshold (WCTH)**

Both FIFOs have their own start of frame thresholds. The Rx FIFO has a specific threshold (RxSFTH) which designates when the first FIFO request will be asserted after a frame is received. The Tx FIFO uses a separate threshold (TxSFTH) to start a transmission after a certain number of bytes are written to the Tx FIFO. Each successive request is based on a common threshold value, WCTH. The word count threshold should be based on the burst capability of the host DMA system. The WCTH bits can be set in the global FIFO configuration register. The following table shows the different threshold possibilities. WCTH is a binary number that represents the number of bytes in a transfer where the number of bytes =  $[(WCTH(dec) + 1) \times 4]$ .

For frame reception,  $\overline{RxREQ}$  will be asserted as data is received and the WCTH is met; assuming the channel is not currently selected. WCTH represents the number of bytes in the Rx FIFO that have not been read. For transmission,  $\overline{TxABLE}$  is asserted as data is transmitted and WCTH bytes of space is available in the FIFO; assuming the channel is not currently selected. In this case, WCTH represents the number of empty bytes in the Tx FIFO.

**Table 39. Word Count Threshold (WCTH) Settings**

WCTH	Bytes	WCTH	Bytes
00000	4	10000	68
00001	8	10001	72
00010	12	10010	76
00011	16	10011	80
00100	20	10100	84
00101	24	10101	88
00110	28	10110	92
00111	32	10111	96
01000	36	11000	100
01001	40	11001	104
01010	44	11010	108
01011	48	11011	112
01100	52	11100	116
01101	56	11101	120
01110	60	11110	124
01111	64	11111	128

**Loopback**

The ATT1MX10 supports loopback operation for diagnostic purposes. Loopback should not be used under normal operating conditions. Loopback is programmable from the transmit frame configuration register. For loopback, data is written to the Tx FIFO and transmitted internal to the ATT1MX10. As the looped back data is received, the host system should read the frame from the Rx FIFO. The three threshold values are maintained as well as the Tx and Rx FIFO handshake signals. Loopback allows testing of the host DMA that the host can perform by writing and reading the same frame to and from the ATT1MX10.

**Tx FIFO Specific Functions**

The Tx FIFO provides the mechanism for sending frame data through the MAC and onto the network. The FIFO input is controlled by the host DMA and provides a fast, efficient method for data transfer. Several signals are needed to provide frame boundaries and CRC generation options.



## Functional Description (continued)

### FIFO Operation (continued)

#### Tx FIFO—Start of Frame Signal (SOF)

In order for the Tx FIFO to mark the first byte of a frame, the SOF signal must be driven active by the host system during the DMA transfer of the first word. The frame must start on an even byte boundary; VB[1:0] is ignored. If TxRDY is asserted and SOF is not driven to indicate the start of frame, the Tx FIFO will assume that SOF was missed due to an error. The Tx FIFO, assuming that the data in the FIFO is part of an incomplete frame, will ignore data until the SOF signal is asserted. The presence of SOF will clear this condition.

#### Tx FIFO—End of Frame Signal (EOF)

In addition to marking the start of frame, the host must also drive EOF to mark the last word of a frame. EOF must be driven during the DMA transfer of the last word in the frame. VB[1:0] must also be driven by the host to indicate which bytes in the last word are valid. Only the last word in the DMA transfer can contain invalid bytes (up to 3 bytes can be invalid).

#### Tx FIFO—Frame Transmission

The transmission of a frame starts automatically when a certain number of words is written to the Tx FIFO or when EOF is written. This threshold is indicated by the TxSFTH [4:0] bits in the global FIFO configuration register. If, for example, TxSFTH = 10000, the transmitter will begin the transmission of the frame when the host has written 64 bytes of data to the FIFO. The number of bytes that the threshold is set to is equal to  $[(\text{TxSFTH}(\text{dec}) + 1) \times 4]$ . Refer to Table 39 for a list of thresholds. TxSPTH and WCTH use the same decoding.

If the Ethernet media is free, transmission will start 9.6  $\mu\text{s}$  after TxSFTH is met. Care **must** be taken in setting the TxSFTH and WCTH thresholds. If it is not, deadlock may occur where the TxSFTH is not met (transmission will not start) and there is not enough room in the Tx FIFO to meet the WCTH (TxABLE will not be asserted). The two must be selected such that  $\text{WCTH} + \text{TxSFTH} \leq 128$  bytes and TxSFTH can be greater than WCTH.

The FIFO logic will continue to transmit the frame until the EOF signal is driven active by the host. If the FIFO runs out of data before EOF is asserted, the fragment of the frame will be transmitted with an incorrect CRC regardless of the state of the CRC bit in the transmit frame configuration register or the TxCRC signal.

Any subsequent words up to and including the next EOF will be discarded by the ATT1MX10. An interrupt may also be generated when the Tx FIFO underruns if the TFIFOM bit is not set in the frame interrupt mask register. If TxHalt is set in the transmit frame configuration register, the Tx FIFO will halt and become disabled until the host sets the enable bit in the same register.

#### Tx FIFO—Collision Retransmission

The Tx FIFO will not write over the first 64 bytes of a frame until, regardless of TxSFTH, they are transmitted successfully. In the event of a collision, the FIFO pointers are reset to the beginning of the frame and these bytes are retransmitted after a collision back off period. Once the first 64 bytes are transmitted, the entire FIFO is recovered.

Sometimes, a collision will occur after the first 64 bytes of the frame have been transmitted. According to the ISO 8802.3 specification, a collision that occurs after 51.2  $\mu\text{s}$  is a late collision and a violation of the specification. The ATT1MX10 will not attempt to retransmit the frame when a late collision occurs. The transmission, instead, will be aborted and data is discarded. If the TxHalt bit in the transmit frame configuration register is set, the Tx FIFO will be halted until the enable bit in the same register is set. The condition can also cause an interrupt on the FRMINT signal if the LATTEM bit is not set in the frame interrupt mask register.

The ATT1MX10 also supports additional collision related features that violate the ISO 8802.3 specification; however, these features make the ATT1MX10 more aggressive when transmitting. This, in turn, helps keep data moving through a frame switch.

The Retry[1:0] bits in the transmit frame configuration register set the number of times the transmitting port will attempt retransmissions after collisions. The default is 16. This can be changed to 8, 4, or 1 transmission attempts.

The BSEL bit in the same register can be set, which turns off the random backoff timer used for collision deference. When the timer is turned off and a collision occurs, the ATT1MX10 will only wait 9.6  $\mu\text{s}$  before it retries the transmission. If the BSEL bit is not set, the ATT1MX10 will use the normal backoff algorithm.

## **Functional Description** (continued)

### **FIFO Operation** (continued)

#### **Tx FIFO—Additional IEEE\* Features**

The FDup bit, also in the transmit frame configuration register, allows the ATT1MX10 to transmit and receive on the same port simultaneously. When FDup is set, the ATT1MX10 will transmit a frame when the 9.6  $\mu$ s interframe gap timer has expired; the transmitter will not defer or back off due to carrier or collision assertion.

The PreAm[1:0] bits in the transmit frame configuration register select the number of preamble bits that are transmitted at the start of a frame. The default is 56 bits. This can be reduced to 48, 40, or 32 bits. Note that the 8-bit SFD pattern (10101011) is always transmitted, regardless of the length of preamble.

#### **Tx FIFO—CRC Generation**

The transmitter in the ATT1MX10 has the capability of calculating and appending the cyclic redundancy check (CRC) to each transmitted frame. There are two ways to accomplish this that are selectable by the HwCRC bit in the transmit frame configuration register. If this bit is set, the ATT1MX10 latches the state of the TxCRC signal when  $\overline{\text{TxRDY}}$ ,  $\overline{\text{PFCs}}$ , and EOF are asserted. If TxCRC is high, the CRC will be calculated and appended to the frame by the ATT1MX10.

If TxCRC is low, the CRC is not calculated and appended. This method allows the host to dynamically append the CRC to specific frames. If HwCRC is low, the transmitter will calculate and append CRC to every frame based on the state of the CRC bit in the transmit frame configuration register. When using the software method, every frame will have the CRC calculated and appended if the CRC bit is high. The CRC will never be calculated and appended if the CRC bit is low.

#### **Tx FIFO—Error Conditions**

The Tx FIFO has the capability of halting its operation when an error occurs. This option can be selected by the TxHALT bit in the transmit frame configuration register. The following conditions will cause this to occur: excessive deferral, excess collisions, late collision, or a FIFO underrun. When the FIFO is halted, the Enable bit in the transmit frame configuration register is cleared and  $\overline{\text{TxABLE}}$  will be deasserted high until the condition is cleared by re-enabling the Tx FIFO. The Tx FIFO is enabled by setting the Enable bit in the transmit frame configuration register. Besides halting the FIFO, the ATT1MX10 can interrupt the host when one of

these errors occur through the  $\overline{\text{FRMINT}}$ . Each of these error conditions can be enabled in the frame interrupt mask register. In the case of a FIFO underrun, the ATT1MX10 will append an incorrect CRC to the frame. This prohibits an incomplete frame from having a valid CRC.

If the error occurs while the system is still writing the frame to the Tx FIFO, the EOF signal still needs to be asserted by the system. This allows both the system and the ATT1MX10 transmit FIFO to recover from the error. The next packet should be started by asserting the SOF signal.

#### **Tx FIFO—Enabling**

Before transmissions start and after a halt condition, the Tx FIFO must be enabled. This is done by setting the Enable bit in the transmit frame configuration register. While the port is disabled, the port's  $\overline{\text{TxABLE}}$  will be deasserted high. It is possible for a port's transmitter to be disabled while the port's receiver is enabled.

#### **Tx FIFO—Short Frames**

The ATT1MX10 does not require transmit frames to be greater than or equal to 64 bytes. The ATT1MX10 will transmit a frame of any length with a correct CRC. If the PAD bit is set in the transmit frame configuration register, the ATT1MX10 will automatically pad the frame with an F0 pattern so that the frame is 64 bytes in length. The ATT1MX10 will also calculate and append the CRC checksum, regardless of the state of HwCRC and TxCRC.

When writing short frames to the Tx FIFO, SOF and EOF may be asserted during the same burst ( $\overline{\text{TxRDY}}$  asserted) or during the same HCLK cycle. If the frame is smaller than TxSFTH, transmission of that frame will begin after EOF is asserted.

#### **Tx FIFO—Multiple Frames**

The Tx FIFO is capable of holding multiple frames. The ATT1MX10 will assert  $\overline{\text{TxABLE}}$  based on the two thresholds set in the global FIFO configuration register. The Tx FIFO will in turn transmit these frames on the network according to the ISO 8802.3 rules. The Tx FIFO will save the first 64 bytes of the frame until they have all been transmitted so that the packet can be retransmitted after a collision. Frame boundaries in the FIFO are recognized by the SOF and EOF signals. If SOF is not driven as the host starts to write the frame data to the FIFO, the data will be ignored.

\* IEEE is a registered trademark of The Institute of Electrical and Electronics Engineers, Inc.

## Functional Description (continued)

### FIFO Operation (continued)

#### TXFIFO—Multiple Frames (continued)

Due to the multiple frame capabilities of the Tx FIFO, it is difficult to maintain status on a single frame. Therefore, transmit statistics are recorded in the transmit event counters. Errors that are unrecoverable (excessive deferral, collision error, late collision, parity error, or a FIFO underrun) can cause the FIFO to halt until the host re-enables it.

This feature can be turned off by clearing the TxHalt bit in the transmit frame configuration register. If TxHalt is not set and one of these errors occurs (excessive deferral, collision error, late collision, or a FIFO underrun), the Tx FIFO will cancel the attempt to transmit the current frame and start the transmission of the next frame in the FIFO.

#### Rx FIFO Specific Functions

The Rx FIFO buffers frames as they are received from the network and provides a slave DMA interface for the host. There are two different threshold settings that are used by the Rx FIFO: the start of frame threshold (RxSFTH), and the word threshold count (WCTH). Both thresholds use the same decoding.

RxSFTH sets the threshold at which  $\overline{\text{RxREQ}}$  is asserted for the first burst of a frame. Refer to Table 39 for a list of thresholds. Each successive Rx FIFO burst is governed by the WCTH. This allows the first burst of the frame to be different than the typical burst capability of the host. For example, a 12-byte RxSFTH can be used so that the source and destination address can be read as soon as possible, or, a 64-byte RxSFTH can be used to guarantee that undersized frames or fragments do not cause unneeded bus activity (if the corresponding reject bits are set in the receive frame configuration register).

#### Rx FIFO—Multiple Frames

The Rx FIFO has the ability to hold multiple frames at any given time. In the event that several short frames are received before the host has time to process them, the ATT1MX10 will store them. This will be transparent to the host, however. The Rx FIFO will request DMA service from the host by asserting  $\overline{\text{RxREQ}}$  low when the data in the FIFO meets the threshold requirements set in the global FIFO configuration register. If there are back-to-back frames, the ATT1MX10 will not drive EOF and SOF in the same burst. In the event that the Rx FIFO becomes full and frames cannot be stored, a FIFO overflow error will occur and the missed frames

counter will be incremented for each frame that is received until the overflow condition is cleared. Rx event counters are maintained while the Rx FIFO is in an overflow state so that error statistics are correctly maintained.

#### Rx FIFO—Start and End of Frame Signals

During the first DMA transfer of a received frame, the ATT1MX10 will drive SOF indicating that the current transfer includes the first byte of the frame. Note that the receive frame will be byte aligned and VB[1:0] are not used.

In addition to marking the start of frame, the ATT1MX10 also marks the end of frame by asserting the EOF signal during the last DMA burst ( $\overline{\text{RxRDY}}$  asserted low). While EOF is asserted, VB[1:0] will indicate which bytes are valid. VB[1:0] are only valid while EOF is asserted; all other reads are byte aligned. Receive statistics are driven on the HCLK cycle following EOF.

#### Rx FIFO—Receive Statistics

As frames are received, status is monitored by the ATT1MX10 and recorded in two places. First, a receive port diagnostic register is updated at the end of every receive frame. It is not practical for the system to read this register after each frame is read. Instead, the contents of this register are buffered with the frame data as part of the end of frame statistics. Second, event counters are updated based on status at the end of a receive frame. These counters can be read at any time in order to be monitored. It is more practical for the system to gather statistics through the event counters. The number of counters provides a good foundation for out of band RMON support.

#### Rx FIFO—Undersized and Fragment Rejection

Undersized and fragment frame filtering can be achieved by setting the RxSFTH to 68 bytes and by setting the UND and or the FRAG bits in the receive frame configuration register. If a frame is received that is less than 64 bytes and has a good CRC (undersized) or has a bad CRC or frame alignment error (fragment), the ATT1MX10 will not assert  $\overline{\text{RxREQ}}$  and will remove the frame from the Rx FIFO. The ATT1MX10 will increment the undersized frame or fragment frame counter and recover the space in the FIFO the frame was occupying. The ATT1MX10 will not DMA the status word if a frame is rejected. If the UND and or FRAG bits are not set, the ATT1MX10 will DMA the frame and append status, regardless if the byte count reached the RxSFTH or not.

## **Functional Description** (continued)

### **FIFO Operation** (continued)

#### **Rx FIFO—Error Conditions**

The ATT1MX10 provides four interrupt bits for receive error conditions. These are found in the frame interrupt register and are the following: auto polarity has detected reverse polarity, change in link status, FIFO overrun, and jabber frame received.

Each of these interrupts can be masked in the frame interrupt mask register. The Rx FIFO will remain operational under each of these conditions, although it is unlikely that frames are still being received, except in the case of autopolarity change. The link status change bit is set when there is a change in link status. The presence of link pulses can be determined by reading the LStat bit in the receive port diagnostic register.

#### **Rx FIFO—Enabling**

Before a port of the ATT1MX10 can receive frames, the port must be enabled. If the port is not enabled, RxREQ will remain deasserted high and the ATT1MX10 will not store received frames in the RxFIFO. The receive event counters, however, will be maintained properly. If a port is disabled by clearing the enable bit in the receive frame configuration register and a frame is currently being received, the port will be disabled after the frame is fully received.

Upon hardware reset ( $\overline{\text{RESET}}$  asserted), the Rx Enable bit will be set and the ATT1MX10 will be ready to receive frames.

## **ISO 8802.3 MAC Functionality**

The ATT1MX10 incorporates an 802.3 compliant MAC for each of its four ports. As an option, the traditional 7-pin MAC interface is accessible at the pin interface. When operating in this mode, the internal transceiver is not used. The 7-pin interface includes transmit and receive clocks, transmit and receive data in NRZ (non-return to zero) format, active-high transmit and receive enables, and active-high collision. When the MAC port is not enabled (TP or AUJ is being used), the 7-pin interface will be configured as outputs. The data on the outputs will reflect the internal signals of the 7-pin interface between the MAC and transceiver.

## **Tx MAC Functionality**

When the MAC interface is being used, a 10 MHz clock is required on TXC, the transmit clock. This clock needs to be synchronized with the transmitting device being used with the ATT1MX10.

If individual transceivers are being used, each port's TXC needs to be synchronized to the specific transceiver. Therefore, each TXC may have a different source clock. The ATT1MX10 drives transmit data (TXD) on the rising edge of TXC while transmit enable (TXE, active-high) is asserted high. The ATT1MX10 will assert and deassert TXE on the rising edge of TXC.

While configured for half-duplex operation, the ATT1MX10 will monitor the collision (COL, active-high) input and react properly when it is asserted. The ATT1MX10 will transmit 32 bits of JAM after COL is asserted and then backs off the Ethernet for a random period of time. The jam pattern that the ATT1MX10 uses is all 1s on TXD.

At the end of the backoff, the ATT1MX10 will retry the transmission attempt, according to the configuration setting in the transmit frame configuration register.

## **Rx MAC Functionality**

When the MAC interface is being used, the MAC receiver uses three signals: carrier detect (CRD, active-high), receive data (RXD), and receive clock (RXC). The ATT1MX10 samples serial data on RXD on the rising edge of RXC while CRD is high. The ATT1MX10 does not require that RXC be driven with a valid clock while CRD is low, with the one exception that five RXC cycles must be received after the falling edge of CRD.

## Functional Description (continued)

### ISO 8802.3 Attachment Unit Interface (AUI)

The AUI transceivers include DO drivers, DI receivers, and CI receivers.

#### Tx Functionality

The ATT1MX10 provides AUI drive capability by including another driver in addition to the standard slew-controlled drivers. During normal (TP) transmission, these drivers are 3-stated and do not affect the device operation.

In order to use the AUI drivers, the four output pins should be connected as shown in Figure 4, except for the test load. In this case, there are no pre-equalization outputs and the PEC and  $\overline{\text{PEC}}$  pins are used as DATA (DO) and  $\overline{\text{DATA}}$  ( $\overline{\text{DO}}$ ) outputs. The pull-up and pull-down resistors control the peak voltage and must be connected. The load resistor must be connected to properly terminate the transmission line.

Capacitive coupling of an AUI port to an on-board MAU can be achieved as shown in Figure 5, provided the circuit that drives the ATT1MX10 does not cause large common-mode shifts.

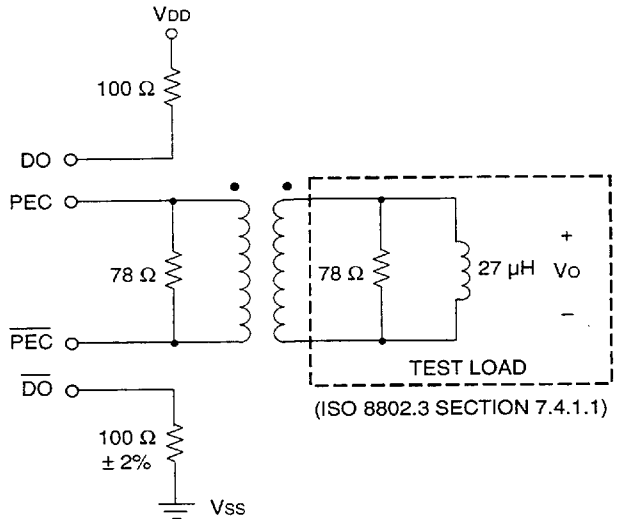


Figure 4. AUI Driver Conditions

For testing purposes, an AUI test load must be connected for proper termination. If this is done, an output voltage between  $\pm 0.6$  V and  $\pm 1.2$  V is measured differentially between PEC and  $\overline{\text{PEC}}$ .

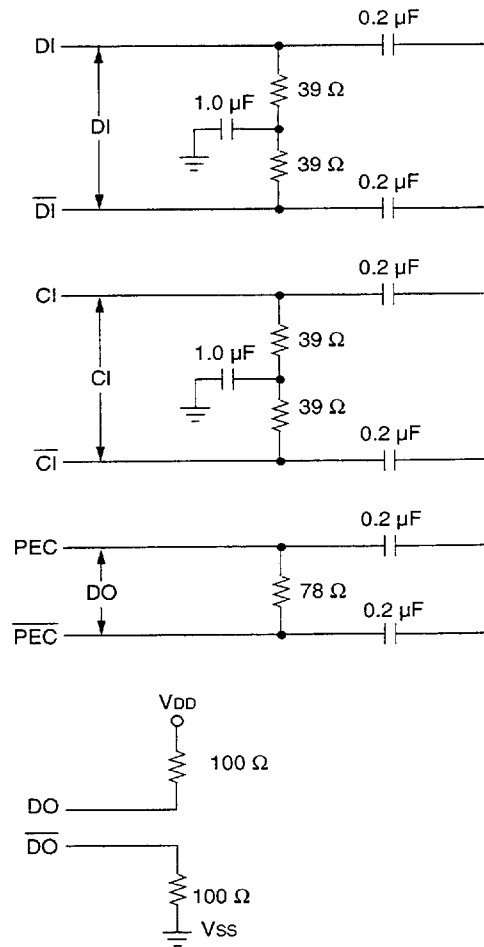


Figure 5. Capacitive Coupling

#### Rx Functionality

The AUI receiver accepts data from the medium attachment unit (MAU) through the DI and  $\overline{\text{DI}}$  pins. The input must be transformer-coupled to the AUI circuit. The differential dc input impedance of the DI and  $\overline{\text{DI}}$  pins is  $20 \text{ k}\Omega \pm 50\%$ .

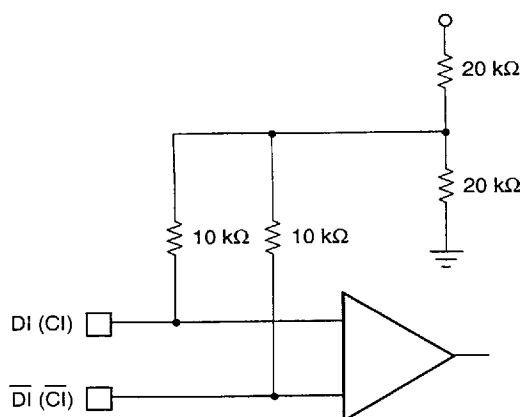
This block includes a squelch circuit to prevent noise from falsely indicating that a packet is being received. This circuit is used to reject spurious noise on DI and  $\overline{\text{DI}}$  that would otherwise cause unintentional start-up. The squelch remains on if the differential input signal is less than 160 mV peak or exceeds this magnitude for less than 20 ns. All signals on DI and  $\overline{\text{DI}}$  are ignored while the squelch is on. The squelch is turned off if the differential input exceeds 300 mV peak for more than 75 ns, and it remains off until an IDL pulse is detected. The squelch will also turn on if the input signal does not exceed the detection threshold for  $500 \text{ ns} \pm 20\%$ .

## Functional Description (continued)

### ISO 8802.3 Attachment Unit Interface (AUI) (continued)

The AUI receivers were designed to be transformer coupled, although capacitive coupling can be used. The capacitive coupling may cause a shift of the ATT1MX10 internal receiver bias. If the bias is upset, the receiver will not function properly. The  $\overline{\text{DI}}$  and  $\overline{\text{DI}}$  signals along with the  $\overline{\text{CI}}$  and  $\overline{\text{CI}}$  signals are internally biased as shown in Figure 6.

The Thevenin equivalent of the bias network of Figure 5 shows two  $39\ \Omega$  resistors with a center tap bypass capacitor ( $1.0\ \mu\text{F}$ ) which will help control the bias on the AUI receivers. The external circuit can also have  $2.5\ \text{V}$  tied to the center tap of the  $39\ \Omega$  resistors. The AUI receiver pins must be biased at  $2.5\ \text{V} \pm 0.5\ \text{V}$  to operate correctly.



5-4057

Figure 6. The AUI Receivers Internal Bias Network

### Collision Functionality

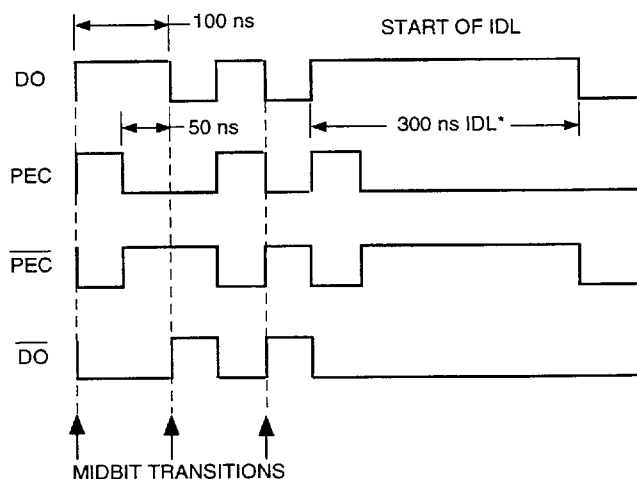
The ATT1MX10  $\overline{\text{CI}}$  and  $\overline{\text{CI}}$  pins must be connected to the AUI  $\overline{\text{CI}}$  and  $\overline{\text{CI}}$  terminals in a fashion similar to the  $\overline{\text{DI}}$  and  $\overline{\text{DI}}$  terminals. Internally, the  $\overline{\text{CI}}$  squelch and receive circuitry is similar to the  $\overline{\text{DI}}$  circuitry. However, the  $\overline{\text{CI}}$  and  $\overline{\text{CI}}$  squelch is turned off if the differential input exceeds 300 mV peak for more than 45 ns to accommodate the 10 MHz collision signal.

## ISO 8802.3 Twisted-Pair (TP) Transceiver

The twisted-pair (TP) transceivers include the TP drivers and receivers. The TP output driver is driven with CMOS logic levels that have a source resistance of less than  $10\ \Omega$  and a maximum dc current source rating of 25 mA. All TP output driver pins are driven low, 300 ns after the rising edge of IDL.

### TP Driver Characteristics

When the TP interface is being used, four output driver signals are connected together through a resistive summing network as shown in Figure 7. The  $\overline{\text{DO}}$  and  $\overline{\text{DO}}$  outputs are Manchester data; the  $\overline{\text{PEC}}$  and  $\overline{\text{PEC}}$  outputs are pre-equalization control. Adding these two signals enhances the high-frequency components of the signal, thus increasing the transmission distance. The outputs of these four signals are shown in Figure 7.



5-3027C.a

\* IDL will be 300 ns if the last bit is 1 or 0.

Figure 7. Pre-equalization Control Format

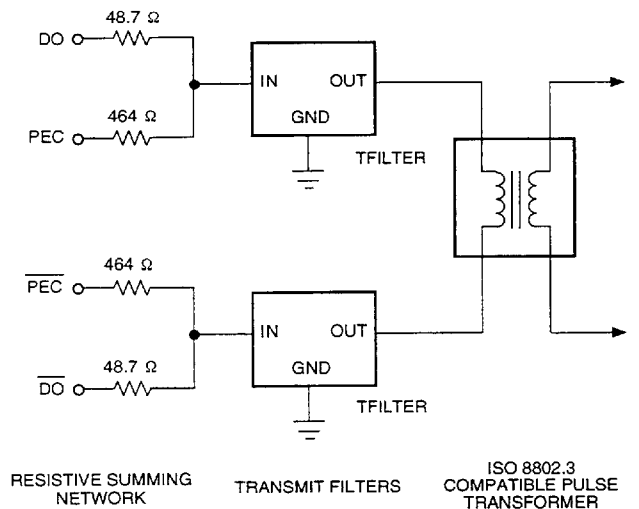
In order to help limit ground bounce and EMI on the transmission lines, the output drivers are designed to have a low output impedance. The output drivers have also been designed so that the rise and fall times are between 3 ns and 10 ns.

## Functional Description (continued)

### ISO 8802.3 Twisted-Pair (TP) Transceiver (continued)

Figure 8 shows an example of a typical output port configuration. All TP ports on the ATT1MX10 should be connected to a circuit like this. The selection of the proper summing resistors and filter characteristics has been discussed extensively in the *10Base-T Twisted-Pair Interface Design Considerations* Application Note.

The ATT1MX10 is designed to drive a 100  $\Omega$  resistive load. Large capacitive loads should not be driven directly. If a large capacitive load is being driven, series resistance should be used between the ATT1MX10 and the load. This will help control ground bounce.



5-3028C

Figure 8. Recommended TP Port Configuration

### TP Receiver Characteristics

The TP receiver is able to resolve differential signals as small as 350 mV peak. To accomplish this, the common-mode voltage of the input is required to be a nominal 2.5 V. Internal circuitry provides this bias.

The ATT1MX10 can also be configured for long line length operation by setting the TPSque bit in the receive frame configuration register. The differential, dc input impedance of the DI and  $\overline{\text{DI}}$  pins is 20 k $\Omega \pm 20\%$ .

The ATT1MX10 squelch function is divided into two parts:

- The first part is amplitude based. The differential input signal must exceed certain amplitude requirements. For normal line lengths, this amplitude is nominally 450 mV. When the squelch circuit has turned off, the amplitude threshold is dropped to 400 mV, nominal. The hysteresis prevents the chance of carrier dropout. When the long line length is enabled, the nominal amplitude is reduced to 350 mV. The sustained threshold is reduced to 300 mV.
- The second part of the squelch function is time based. If a signal exceeds the required amplitude, the amplitude must be maintained for 25 ns. Several pulses must occur with the proper polarities and amplitudes in order for the squelch circuit to turn off. Once the squelch has been turned off, all signals on the DI and  $\overline{\text{DI}}$  pair are received until an IDL is detected or there is no input activity for 5 bit times.

### TP Receiver Smart Squelch

The receiver rejects the following signals in accordance with Section 14.3.1.3.2 of the ISO 8802.3 standard:

- All signals that, when measured through the recommended input filter, produce a peak magnitude of less than 300 mV.
- All continuous sinusoidal signals of amplitude less than 3.1 V peak and frequency less than 2 MHz.
- All single sinusoidal cycles with amplitude less than 3.1 V peak and starting with either polarity where the frequency is between 2 MHz and 15 MHz. For a period of 4 bit times before and after this single cycle, the received signal must conform to step 1 above.

## **Functional Description** (continued)

### **Summary of ATT1MX10 Registers and Counters**

The ATT1MX10 uses a set of registers for port configuration and status. These registers are accessible through the 16-bit CPU interface. While most of these registers are specific for each port, there are three global registers: the global configuration register, the global FIFO configuration register, and the port identification register. The per-port registers are replicated four times and either configure a port or maintain status of a port.

The network event counters are accessible in the address space above the configuration and status registers. Each of the counters are replicated four times so that statistics from each port are gathered individually. The exact address locations of both the counters and registers are shown in Table 10.

Some registers and counters are 16 bits wide while others are 32 bits wide. Because the CPU data bus is 16 bits wide, 16-bit counters and registers can be read or written directly, with one access. 32-bit counters and registers must be read and written in two cycles. When reading a 32-bit counter or register and the upper byte is addressed (MSB = D31—D16), data is read from the counter or register directly and stored in a holding register. The addressed word will be driven on the data bus. When the second word is read, data will be driven from the holding register to ensure a valid count.

Data written to a 32-bit register or counter is stored in a holding register. Data from the holding register is then written to the addressed counter or register when the lower word is written by the host. Therefore, it is important to write the upper word (D31—D16) first, and then the lower word.

When writing or reading registers and counters, there is no time constraint on the successive reads of words from the same counter or register other than the specific timing on the CPU bus. The host should allow a two HCLK cycle delay between reads of different counters.

As a general rule for reading registers or counters, the upper word should be read first so that the current count or status is frozen in the holding register. Then, the lower word can be read. For writing, the upper word should be written first to load data into the holding register. The lower word should be written last in order to load the correct data in the register or counter. Note that although some registers and counters may have an undefined upper word, the upper word still needs to be read or written to update the holding register.

Also, note that partial reads and writes should not be intermixed. Doing so will destroy the contents of the holding register. For example, do not read a word of counter A, write a word of register B, and then read the rest of counter A. Completely read the contents of counter A, and then write to register B.



## Functional Description (continued)

### ATT1MX10 Registers

The ATT1MX10 registers are used primarily for configuration and error notification. There are also two diagnostic registers that may be helpful in system diagnostic testing.

Configuration registers should only be written at system start-up or whenever the chip is reset. (RESET driven low; software resets do not change the contents of configuration registers.) They should not be written to while the ATT1MX10 is transmitting or receiving frames.

Diagnostic registers can be read at any time and reflect the current status of frames that have been completely received or transmitted. The frames that are partially received or transmitted at the time of the diagnostic register read will not be reflected in the status. Diagnostic registers represent the status of the last frame received or transmitted.

### Configuration Registers

The configuration registers all reset to known default values that enable the ATT1MX10 to transmit and receive frames on the MAC interface. These registers should be updated if the default configuration does not suit the needs of the host system. The ATT1MX10 has six configuration registers:

- The global configuration register configures the bus size, interrupt type, and parity selection. The register also has self-clearing reset bits that can individually reset the port counters, port transmitters, or port receivers. None of these reset bits clear the configuration registers.
- The transmit frame configuration register sets the transmit rules that the ATT1MX10 will use. It also has an enable bit that can enable or disable the transmission port.
- The receive frame configuration register sets the media type (TP, AUI, or MAC). There are also ID bits which can be set. These ID bits then become part of the end of frame statistics and would indicate that port the frame was received on. This register also has an enable bit that enables or disables the receive port. In addition, this register determines what error frames the ATT1MX10 will DMA to the host and which ones it will discard.
- The global FIFO configuration register is a global register that sets the FIFO thresholds on all four ports.

- The frame interrupt mask register enables specific transmit and receive error conditions to drive the FRMINT signal.
- The CPU interrupt mask register determines which counters can drive the CPUINTR signal active when they overflow.

### Status Registers

Another set of registers is the status and interrupt registers. Each port has a transmit port diagnostic register and a receive port diagnostic register. These registers are updated after a frame is completely transmitted or completely received. Both registers contain status for only one frame. In other words, they do not accumulate status. Under normal network traffic conditions, it is unlikely that these registers can be read accurately. (Status from some frames will most likely be missed.) Therefore, these registers should only be read during testing and diagnosis where the network conditions are known. The ATT1MX10 status registers include:

- The port interrupt identification register is a first level interrupt notification register and identifies which port caused the CPUINTR to become asserted.
- The counter overflow indication register indicates which counter or counters are currently overrun. This register is cleared when it is read.
- The frame interrupt register will indicate any error conditions on a port. This register is also cleared when it is read.

### ATT1MX10 Counters

The ATT1MX10 provides an extensive list of network event counters. The entire set of counters is replicated for each port. The counters clear to 0 upon a hardware reset (RESET asserted low) or a software reset (CntRes set in the global configuration register). The counters will count all events even when the port is disabled or the Rx FIFO is overrun.

When counters reach FFFF, an interrupt is generated and they continue counting. Note that when a counter reaches FFFF, the next event will increment the counter to 0000 (any external counter should be incremented by 1 [FFFF + 1] on the next event after a counter overflows).

Internal arbitration allows counters to be read efficiently by the host and updated by the ATT1MX10 concurrently. The number of counters, coupled with their accuracy and accessibility, allow out-of-band statistic gathering for RMON (remote network monitoring MIB).

## **Functional Description** (continued)

### **Counter Definitions**

This section defines the conditions that increment each of the counters listed in Table 10. Counters are cleared by a hardware reset or by setting the CntRes bit in the global configuration register. Counters can also be written to through the CPU interface. It is not recommended that counters be written during normal operation. It is intended that counters be preloaded with a known value for counter overflow testing during diagnostics.

#### **Total Octets Transmitted**

This 32-bit counter counts the number of octets that are transmitted out of a port including the 4 octets of the frame check sequence (CRC). An octet is defined as 8 bits of transmitted data and is counted after SFD is transmitted. This count does not include framing octets (preamble). This does include octets that are transmitted before a collision, excess deferral, or FIFO underrun occurs. This count is the exact number of octets transmitted on the media by a single port.

#### **Total Good Frames Transmitted**

This 32-bit counter counts the number of frames that are transmitted out of an individual port. This count does not include those frames that are transmitted with errors (i.e., collision fragments and partial frames due to FIFO underruns).

#### **Total Collisions**

This 32-bit counter counts the number of collisions that occur on an individual port during transmission attempts. Collisions that occur while the port is not in the transmit state are not counted. (Receive collisions are not counted.)

#### **Multicast Frames Transmitted**

This 16-bit counter counts the number of good multicast frames that are transmitted from an individual port. A multicast frame is identified by having a 1 in the least significant bit of the most significant byte of the destination address. (The first bit transmitted is a 1.) Frames transmitted to the broadcast address are not included in this count.

#### **Broadcast Frames Transmitted**

This 16-bit counter counts the number of good broadcast frames that are transmitted from an individual port. A broadcast frame is a special type of multicast frame where all the bits of the destination address are 1.

### **Single Collisions**

This 16-bit counter counts the number of times a frame is transmitted successfully from a port and one collision occurs.

### **Late Collisions**

This 16-bit counter counts the number of late collisions that occur on a single port. A late collision is defined as a collision that occurs 512 bit times after the transmission starts. The 512 bit time is measured from the start of preamble.

### **Excessive Deferrals**

This 16-bit counter counts the number of transmissions that failed because the transmitter deferred past 24,288 bit times and the transmission never started. The deferral timer is started at the initiation of each transmission attempt, regardless of the retry attempts.

### **Transmit Retry Errors**

This 16-bit counter counts the number of frames that are not transmitted because the number of collision retry attempts was greater than the threshold defined in the transmit frame configuration register.

### **Total Octets Received**

This 32-bit counter counts the number of octets that are received by an individual port. The octet count includes the frame checksum as well as octets from erroneous frames. Octets are 8-bit bytes of receive data received after the SFD.

### **Misaligned Frames**

This 32-bit counter counts the number of frames that are received by an individual port where the frame check sequence (CRC) is invalid and the frame does not align on an 8-bit boundary (nonintegral number of octets). The frame must be greater than or equal to 64 octets and less than or equal to 1518 octets, including FCS, but not including preamble bits.

### **Frames Received with Bad CRC**

This 32-bit counter counts the number of frames that are received by an individual port where the frame check sequence (CRC) is invalid and has an integral number of octets. The frame must be greater than or equal to 64 octets and less than or equal to 1518 octets, including FCS, but not including preamble bits. Also, the frame must align on an 8-bit boundary.

## Functional Description (continued)

### Counter Definitions (continued)

#### Total Good Octets Received

This 32-bit counter counts the number of octets that are received by an individual port. The octet count includes the frame checksum but does not include octets from erroneous frames. Octets are 8-bit bytes of receive data received after the SFD.

#### Total Good Frames Received

This 32-bit counter counts the number of good frames that are received by a particular port. This count includes broadcast and multicast frames. Frames with errors are not included in this count, but are counted individually based on the error.

#### Frames Received (64 Octets)

This 32-bit counter counts the number of frames received by an individual port (including frames with errors) that are 64 octets in length, including CRC, excluding preamble bits.

#### Jabber Frames Received

This 16-bit counter counts the number of frames that are received on an individual port that are longer than 1518 octets including the frame check sequence (CRC), and have either an invalid CRC with an integral number of octets or an invalid CRC with a nonintegral number of octets.

#### Multicast Frames Received

This 16-bit counter is the number of good multicast frames that are received by an individual port. Multicast frames that have errors are not included in this count. The number of broadcast frames received are not included in this count.

#### Broadcast Frames Received

This 16-bit counter counts the number of good broadcast frames that are received by an individual port. It does not include broadcast frames that have errors.

#### Missed Frames

This 16-bit counter counts the number of frames that would otherwise be received by an individual port but are not able to be stored in the Rx FIFO because it has overflowed.

#### Fragments Received

This 16-bit counter counts the number of packet fragments that are received on an individual port. A frag-

ment is defined as being a packet shorter than 64 octets, including the CRC, but not including preamble, has a valid SFD, and has either an invalid CRC with an integral number of octets or an invalid CRC with a non-integral number of octets.

#### Undersized Frames Received

This 16-bit counter counts the number of frames that are received on an individual port that are shorter than 64 octets including the frame check sequence (CRC), and do not have any errors. SFD must be received so that the CRC can be calculated.

#### Frames Received (65 to 127 Octets)

This 16-bit counter counts the number of frames received by an individual port (including frames with errors) that are greater than or equal to 65 octets and less than or equal to 127 octets, including CRC, but not including preamble bits.

#### Frames Received (128 to 255 Octets)

This 16-bit counter counts the number of frames received by an individual port (including frames with errors) that are greater than or equal to 128 octets and less than or equal to 255 octets, including CRC, but not including preamble bits.

#### Frames Received (256 to 511 Octets)

This 16-bit counter counts the number of frames received by an individual port (including frames with errors) that are greater than or equal to 256 octets and less than or equal to 511 octets, including CRC, but not including preamble bits.

#### Frames Received (512 to 1023 Octets)

This 16-bit counter counts the number of frames received by an individual port (including frames with errors) that are greater than or equal to 512 octets and less than or equal to 1023 octets, including CRC, but not including preamble bits.

#### Frames Received (1024 to 1518 Octets)

This 16-bit counter counts the number of frames received by an individual port (including frames with errors) that are greater than or equal to 1024 octets and less than or equal to 1518 octets, including CRC, but not including preamble bits.

#### Long Frames Received

This 16-bit counter counts the number of frames that are received on an individual port that are longer than 1518 octets including the frame check sequence (CRC) and do not have any errors.

## **Functional Description** (continued)

### **CPU Interface**

The registers and counters of the ATT1MX10 are accessible through the CPU interface. The CPU interface has been designed to accommodate CPUs that memory map or I/O map the ATT1MX10. The CPU interface can be accessed independently from the DMA interface.

### **CPU Cycles**

The ATT1MX10 uses several signals for control on the CPU bus.  $\overline{\text{CPUCS}}$  selects the ATT1MX10 and enables the CPU interface.  $\overline{\text{CPUSTRB}}$  provides a strobe to start and finish the access to the CPU interface.  $\overline{\text{CPUDVAL}}$  indicates when data is valid on the CPU data bus.  $\text{CPUR/W}$  sets the direction of the data bus. Each of these signals is fully described in the Pin Descriptions section.

The ATT1MX10 provides internal arbitration for counter access. Both the CPU and the ATT1MX10 have access to the event counters; however, the CPU cannot access a counter while the ATT1MX10 is updating it. For this reason, the ATT1MX10 reads the counter value from memory and stores the value in a snapshot register. The CPU is then able to read the counter. The ATT1MX10 will update the snapshot register when the upper word of a 32-bit counter is addressed. There is a recovery time of two HCLK cycles needed between successive counter reads. This allows the ATT1MX10 to update the event counters. Refer to the Summary of ATT1MX10 Registers and Counters section for a complete description on reading counters.

### **Asynchronous Accesses**

A read cycle is started by asserting  $\overline{\text{CPUCS}}$  and  $\overline{\text{CPUSTRB}}$  low along with a valid address. On the next rising edge of HCLK, the ATT1MX10 will latch the address and commence the read of the counter or register. Within four HCLK cycles, the ATT1MX10 will drive  $\overline{\text{CPUDVAL}}$  and  $\text{CPUDB}$  valid. Note that the  $\overline{\text{CPUDVAL}}$  signal will be asserted by the ATT1MX10 one clock before the data is valid.  $\text{CPUAD}$  needs to be held valid for the entire cycle ( $\overline{\text{CPUSTRB}}$  low).

When writing to the ATT1MX10, the host needs to drive  $\overline{\text{CPUCS}}$ ,  $\overline{\text{CPUSTRB}}$ ,  $\text{CPUAD}$ , and  $\text{CPUDB}$  to their valid states. Each of these signals needs to be driven for at least two HCLK cycles so that the ATT1MX10 can latch the data.  $\text{CPUAD}$  needs to be held valid for the entire cycle ( $\overline{\text{CPUSTRB}}$  low).

## Functional Description (continued)

### CPU Interface (continued)

#### Interrupts

The ATT1MX10 supports two kinds of interrupts. One is a port interrupt, and the other is a counter interrupt. Each port within the ATT1MX10 has its own port interrupt signal,  $\overline{\text{FRMINT}}$ .

The counter interrupt is shared by all four ports. The port interrupt is used mainly for notifying the CPU that a transmit or receive error has occurred on one of the ports, which may cause the port to stop operating correctly.

Each port has a pair of registers that are used to control and identify the interrupt:

- The frame interrupt mask register allows the system to configure the  $\overline{\text{FRMINT}}$  signal to interrupt the system for one, many, or all of the interrupt events.
- The frame interrupt register can be read by a CPU after  $\overline{\text{FRMINT}}$  is asserted to determine what event caused the interrupt.

Events that are masked in the frame interrupt mask register will still be recorded in the frame interrupt register. However, the masked event will not cause  $\overline{\text{FRMINT}}$  to be asserted. When the frame interrupt register is read by the host, it will be cleared and  $\overline{\text{FRMINT}}$  will be deasserted.

The  $\overline{\text{CPUINTR}}$  is a single interrupt that will be driven active-low when a counter overflows. A counter will overflow and set the interrupt when it reaches its terminal value (FF for 16-bit counters and FFFF for 32-bit counters). The counter will then roll over and continue counting. If a counter is not being used by the host, it can be masked so that it does not cause  $\overline{\text{CPUINTR}}$  to be asserted when it overflows. Each counter can be masked by setting the corresponding bit in the CPU interrupt mask register. Note that although a counter interrupt may be masked, its count is still valid.

When  $\overline{\text{CPUINTR}}$  is asserted by the ATT1MX10, the port interrupt ID register must be read by the host to determine which counter or counters have overflowed, followed by reading the appropriate counter overflow indication register. When these registers are read, the port interrupt ID register is cleared and  $\overline{\text{CPUINTR}}$  is deasserted.

As an option, the four  $\overline{\text{FRMINT}}$  signals can be routed to the  $\overline{\text{CPUINTR}}$  signal so that one interrupt line is used. This can be done by setting the IntSel bit in the global configuration register.

#### Hardware Reset

The ATT1MX10 has a single hardware reset pin,  $\overline{\text{RESET}}$ , which, when asserted low, will reset the entire ATT1MX10. When  $\overline{\text{RESET}}$  is driven high, registers will return to default values, state machines will return to idle, counters will be reset to 00 and the FIFOs will be cleared. Any packets being transmitted or received at the time of reset will be lost. In order for reset to be successful,  $\overline{\text{RESET}}$  must be held low for 20 HCLK cycles. HCLK must be valid during reset. Upon powerup,  $\overline{\text{RESET}}$  must be asserted when the power supply reaches its full power and HCLK has stabilized. This will ensure proper operation of the ATT1MX10.

#### Software Reset

The ATT1MX10 offers software reset bits that are found in the global configuration register. There are three different reset bits: counter reset, transmit port reset, and receive port reset.

Each of these bits are particular to an ATT1MX10 port. The receive and transmit port reset bits reset the receive and transmit FIFOs as well as the other digital logic associated with the port. These resets do not have any effect on the counter values, although during the transmit or receive port reset, associated counters will not be updated. The counter reset bit will reset the port event counters to the 00 value. All of the reset bits are cleared by the ATT1MX10 when the reset is complete.

## Application Information

### RMON Cross Reference

The ATT1MX10 fully supports the objects defined in the remote network monitoring MIB. The following table shows which ATT1MX10 counters can be used to keep the statistics (objects) in RMON. In some cases, several ATT1MX10 counters need to be added together to support the RMON objects. There are other RMON objects that use the ATT1MX10 counters, but these are the most obvious:

- Rx and Tx refer to whether the counter is adding receive statistics or transmit statistics.
- Frm and Oct refer to whether the count is done on a frame basis or on an octet basis.
- Err indicates whether frames with errors are included in the count or not.
- A check mark (✓) denotes which counters are applicable to the RMON object.

**Table 40. RMON Cross Reference Objects**

RMON Object	Rx	Tx	Frm	Oct	Err	ATT1MX10 Counter(s)
etherStatsDropEvents	✓	—	✓	—	✓	Missed Frames
etherStatsOctets	✓	—	—	✓	✓	Total Octets Received
etherStatsPkts	✓	—	✓	—	✓	Frames Received Multicast Frames Received Broadcast Frames Received Misaligned Receive Frames Undersized Frames Received Fragments Received Frames Received with Bad CRC
etherStatsBroadcastPkts	✓	—	✓	—	—	Broadcast Frames Received
etherStatsMulticastPkts	✓	—	✓	—	—	Multicast Frames Received
etherStatsCRCAlignErrors	✓	—	✓	—	✓	Frames Received with Bad CRC Misaligned Receive Frames
etherStats Undersize Pkts	✓	—	✓	—	✓	Undersized Frames Received
etherStats Oversize Pkts	✓	—	✓	—	✓	Long Frames Received
etherStats Fragments	✓	—	✓	—	✓	Fragments Received
etherStats Jabbers	✓	—	✓	—	✓	Jabber Frames Received
etherStats Collisions	—	✓	✓	—	✓	Total Collisions Late Collisions
etherStatsPkts 64 Octets	✓	—	✓	—	—	Frames Received (64 Octets)
etherStatsPkts 64 to 127 Octets	✓	—	✓	—	—	Frames Received (65—127 Octets)
etherStatsPkts 128 to 255 Octets	✓	—	✓	—	—	Frames Received (128—255 Octets)
etherStatsPkts 256 to 511 Octets	✓	—	✓	—	—	Frames Received (256—511 Octets)
etherStatsPkts 512 to 1023 Octets	✓	—	✓	—	—	Frames Received (512—1023 Octets)
etherStatsPkts 1024 to 1518 Octets	✓	—	✓	—	—	Frames Received (1024—1518 Octets)
etherHistoryDropEvents	✓	—	✓	—	✓	Missed Frames
etherHistoryOctets	✓	—	—	✓	✓	Total Octets Received

## Application Information (continued)

### RMON Cross Reference (continued)

Table 40. RMON Cross Reference Objects (continued)

RMON Object	Rx	Tx	Frm	Oct	Err	ATT1MX10 Counter(s)
etherHistoryPkts	√	—	√	—	√	Frames Received Multicast Frames Received Broadcast Frames Received Misaligned Receive Frames Undersized Frames Received Fragments Received Frames Received with Bad CRC
etherHistoryBroadcastPkts	√	—	√	—	—	Broadcast Frames Received
etherHistoryMulticastPkts	√	—	√	—	—	Multicast Frames Received
etherHistoryCRCAlignErrors	√	—	√	—	√	Frames Received with Bad CRC Misaligned Frames Received
etherHistoryUndersizedPkts	√	—	√	—	√	Undersized Frames
etherHistoryOversizedPkts	√	—	√	—	√	Long Frames Received
etherHistoryFragments	√	—	√	—	√	Fragments Received
etherHistoryJabbers	√	—	√	—	√	Jabber Receive Frames
etherHistoryCollisions	—	√	√	—	√	Total Collisions Late Collisions
hostInPkts	√	—	√	—	—	Frames Received
hostOutPkts	—	√	√	—	√	Total Frames Transmitted
hostInOctets	√	—	—	√	—	Total Good Octets Received
hostOutOctets	—	√	—	√	√	Total Octets Transmitted
hostOutErrors	—	√	√	—	√	Total Collisions Late Collisions
hostOutBroadcastPkts	—	√	√	—	—	Broadcast Frames Transmitted
hostOutMulticastPkts	—	√	√	—	—	Multicast Frames Transmitted

## Absolute Maximum Ratings

Stresses in excess of the absolute maximum ratings can cause immediate or latent permanent damage to the device. These are absolute stress ratings only. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operational sections of the data sheet. Exposure to absolute maximum ratings for extended periods can adversely affect device reliability.

Parameter	Symbol	Min	Max	Unit
Ambient Operating Temperature Range	T <sub>A</sub>	0	70	°C
Storage Temperature Range	T <sub>stg</sub>	–40	125	°C
Voltage on Any Pin with Respect to Ground	—	–0.5	V <sub>DD</sub> + 0.5	V
V <sub>DD</sub>	—	—	7.0	V

## Handling Precautions

Although protection circuitry has been designed into this device, proper precautions should be taken to avoid exposure to electrostatic discharge (ESD) during handling and mounting. Lucent Technologies employs a human-body model (HBM) and charged-device model (CDM) for ESD-susceptibility testing and protection design evaluation. ESD voltage thresholds are dependent on the circuit parameters used in the defined model. No industry-wide standard has been adopted for the CDM. However, a standard HBM (resistance = 1500  $\Omega$ , capacitance = 100 pF) is widely used and, therefore, can be used for comparison. The HBM ESD threshold presented here was obtained by using these circuit parameters:

Device	Voltage
ATT1MX10	1000 V

## Electrical Characteristics

Ambient temperature 0 °C to 70 °C, V = 5.0 V  $\pm$  5%, GND = 0.0 V.

Parameter	Symbol	Test Conditions	Min	Max	Unit
Input Voltage:					
Low	V <sub>IL</sub>	—	−0.3	0.8	V
High	V <sub>IH</sub>	—	2.0	V <sub>DD</sub> + 0.3	V
Output Voltage:					
Low	V <sub>OL</sub>	—	—	0.5	V
High	V <sub>OH</sub>	—	2.4	—	V
Input Leakage Current	I <sub>ILH</sub> , I <sub>ILL</sub>	V = 5.5 V	—	2	$\mu$ A
Power Supply Current	I <sub>DDS</sub>	0 °C, V <sub>DD</sub> = 5.0 V	—	500	mA
Power Dissipation	P <sub>D</sub>	0 °C, V <sub>DD</sub> = 5.0 V	—	2.5	W
Pin Capacitance:					
CMOS Input	—	—	—	5	pF
CMOS Output	—	—	—	5	pF
CMOS 3-state	—	—	—	5	pF
AUI Receiver:					
Unsquench	—	—	160	300	mV
TP Receiver:					
Unsquench	—	—	300	585	mV
AUI Driver:					
Amplitude	—	—	450	1315	mV
VCOM Shift	—	—	—	40	mV
Rise/Fall Time	—	—	—	3	ns
Undershoot	—	—	—	100	mV
Ringing	—	—	—	200	mV
Jitter	—	—	—	500	ps
TP Driver Specifications:					
Output Resistance	R <sub>O</sub>	—	—	10	$\Omega$
Output Low Voltage	V <sub>OL</sub>	—	—	GND + 0.25	V
Output High Voltage	V <sub>OH</sub>	—	V <sub>DD</sub> − 0.25	—	V
Output Low Match Voltage	DV <sub>OL</sub>	—	−50	50	mV
Output High Match Voltage	DV <sub>OH</sub>	—	−50	50	mV
Rise/Fall Time	—	—	3	10	ns
Transmit Jitter	—	—	—	2	ns
Jitter Tolerance	—	—	$\pm$ 18	—	ns



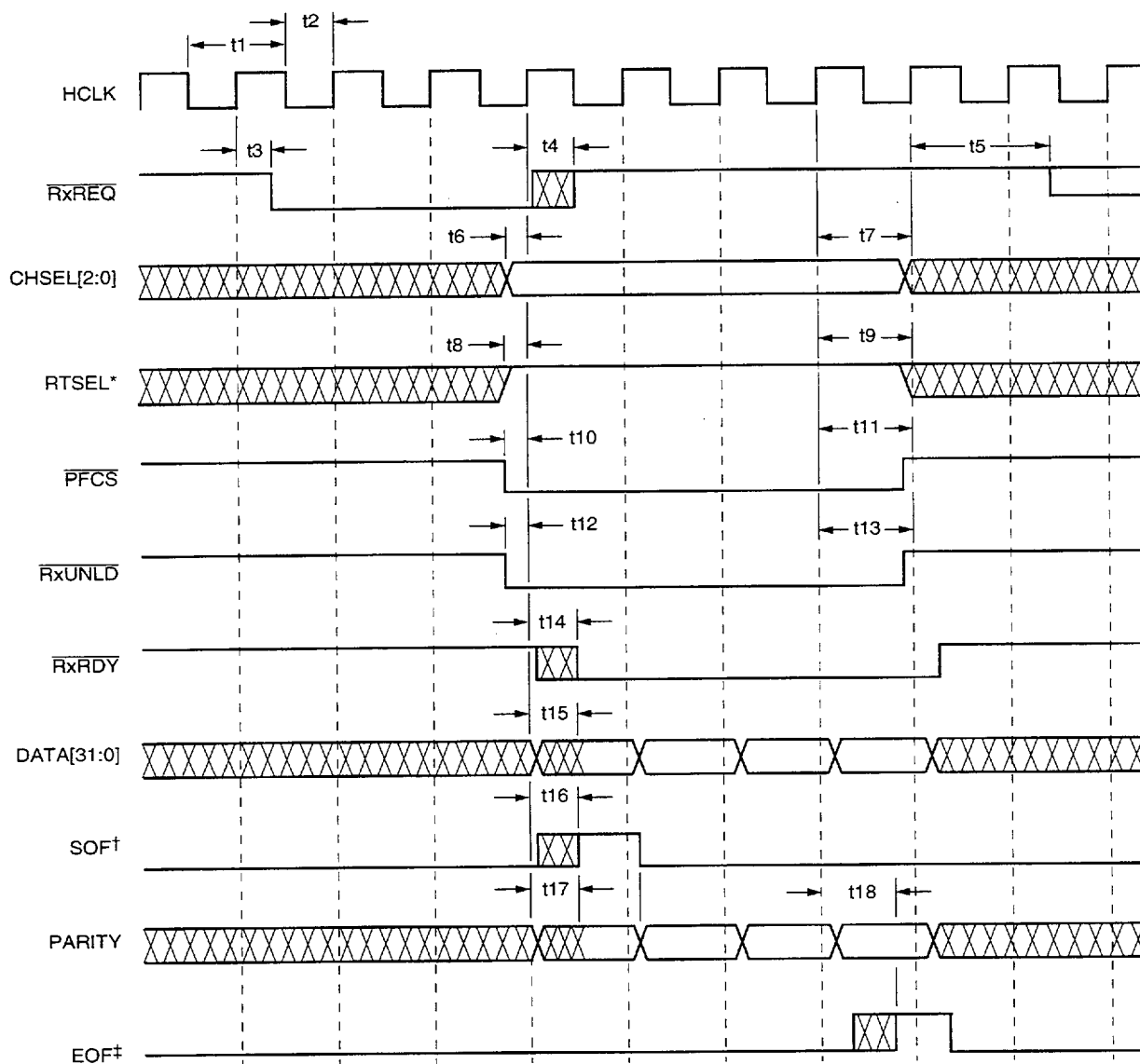
## Timing Characteristics

This section identifies ac timing for the ATT1MX10. The timing assumes a load capacitance of 50 pF.

**Table 41. Rx FIFO Read, 4-Word, Single-Cycle Access**

Name	Parameter	Min	Max	Unit
t1	Clock Frequency	16	25	MHz
t2	Minimum Clock Time Low	18	—	ns
t3	RxREQ Delay from HCLK Rising Edge	2	21	ns
t4	RxREQ Deactive Delay from HCLK Rising Edge	2	21	ns
t5	RxREQ Reasserted (based on threshold)	1 HCLK cycle	—	—
t6	CHSEL[2:0] Setup to HCLK Rising Edge	18	—	ns
t7	CHSEL[2:0] Hold from HCLK Rising Edge	2	—	ns
t8	RTSEL Setup to HCLK Rising Edge	18	—	ns
t9	RTSEL Hold from HCLK Rising Edge	2	—	ns
t10	PFCS Setup to HCLK Rising Edge	15	—	ns
t11	PFCS Hold from HCLK Rising Edge	2	—	ns
t12	RxUNLD Setup to HCLK Rising Edge	17	—	ns
t13	RxUNLD Hold from HCLK Rising Edge	2	—	ns
t14	RxRDY Active Delay from HCLK Rising Edge	2	23	ns
t15	DATA Active Delay from HCLK Rising Edge	2	28	ns
t16	SOF Active Delay from HCLK Rising Edge	2	28	ns
t17	PARITY Active Delay from HCLK Rising Edge	2	28	ns
t18	EOF Active Delay from HCLK Rising Edge	2	23	ns

# Timing Characteristics (continued)



5-4365F

\* When RTSEL changes from low to high, the ATT1MX10 asserts SOF, EOF, VB1, VB0, PARITY, and DATA[31:0] after the rising edge of HCLK.

These signals are deasserted after a combinatorial delay when RTSEL switches to a low.

† SOF will be asserted if this is the first transfer of a new frame.

‡ EOF will be asserted if this is the last transfer of a frame.

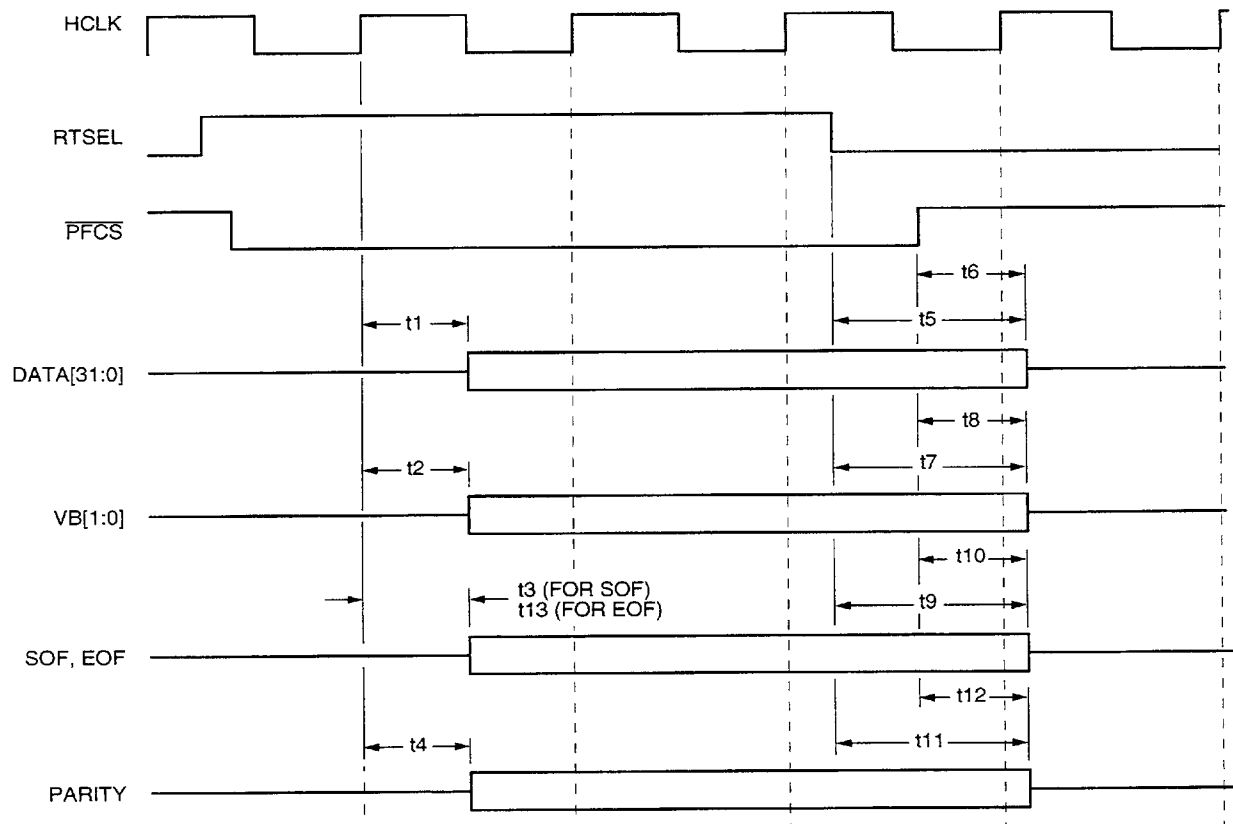
Note: Refer to Figure 10 and Table 42 for 3-state timing.

**Figure 9. Rx FIFO Read, 4-Word, Single-Cycle Access**

## Timing Characteristics (continued)

Table 42. DATA Bus 3-State Timing

Name	Parameter	Min	Max	Unit
t1	DATA 3-state Active Delay from HCLK Rising Edge	—	28	ns
t2	VB 3-state Active Delay from HCLK Rising Edge	—	25	ns
t3	SOF 3-state Active Delay from HCLK Rising Edge	—	28	ns
t4	Parity 3-state Active Delay from HCLK Rising Edge	—	28	ns
t5	DATA 3-state from RTSEL Falling Edge	—	15	ns
t6	DATA 3-state from $\overline{\text{PFCS}}$ Rising Edge	—	15	ns
t7	VB 3-state from RTSEL Falling Edge	—	15	ns
t8	VB 3-state from $\overline{\text{PFCS}}$ Rising Edge	—	15	ns
t9	SOF, EOF 3-state from RTSEL Falling Edge	—	15	ns
t10	SOF, EOF 3-state from $\overline{\text{PFCS}}$ Rising Edge	—	15	ns
t11	PARITY 3-state from RTSEL Falling Edge	—	15	ns
t12	PARITY 3-state from $\overline{\text{PFCS}}$ Rising Edge	—	15	ns
t13	EOF 3-state Active Delay from HCLK Rising Edge	—	23	ns



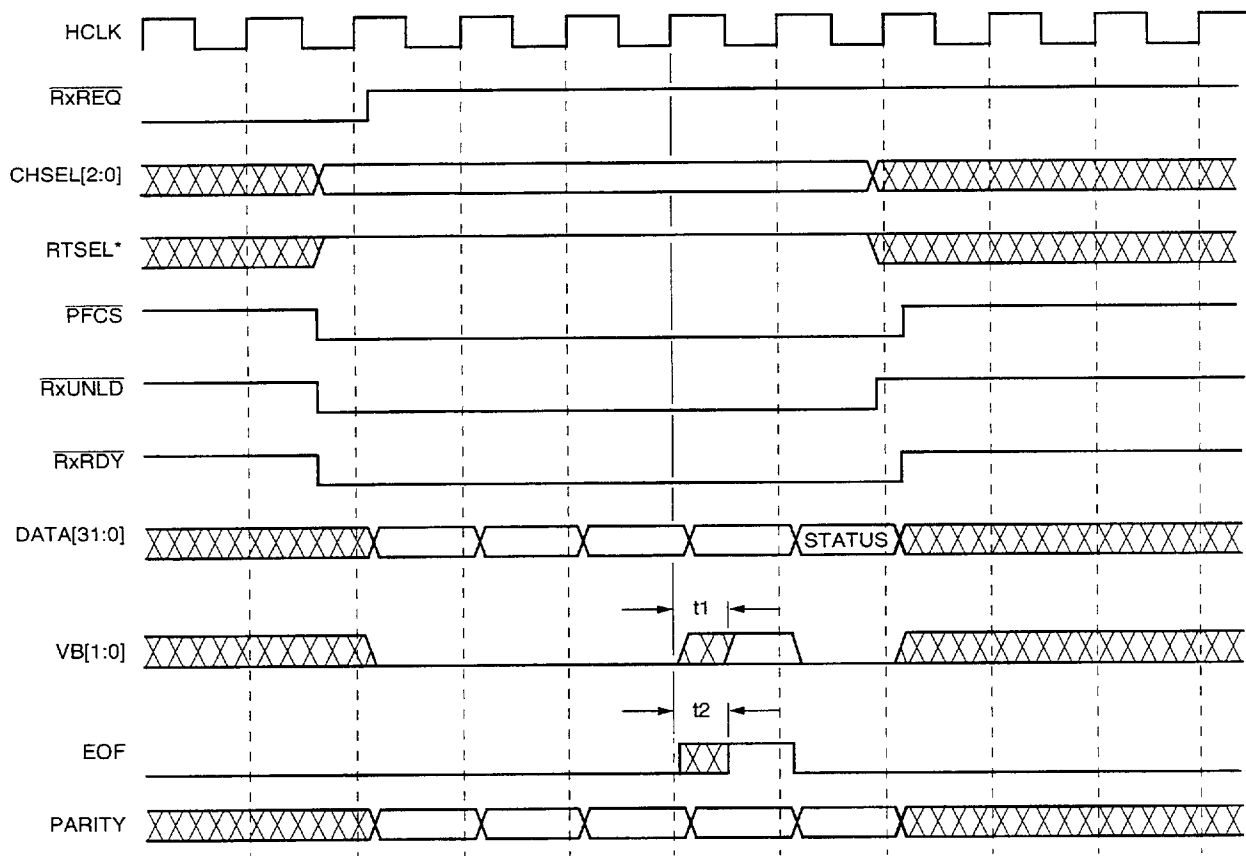
5-4366F

Figure 10. DMA Bus 3-State Timing

Timing Characteristics (continued)

Table 43. Receive Statistics (32-bit Bus) Timing

Name	Parameter	Min	Max	Unit
t1	VB Active Delay from HCLK Rising Edge	2	25	ns
t2	EOF Active Delay from HCLK Rising Edge	2	23	ns

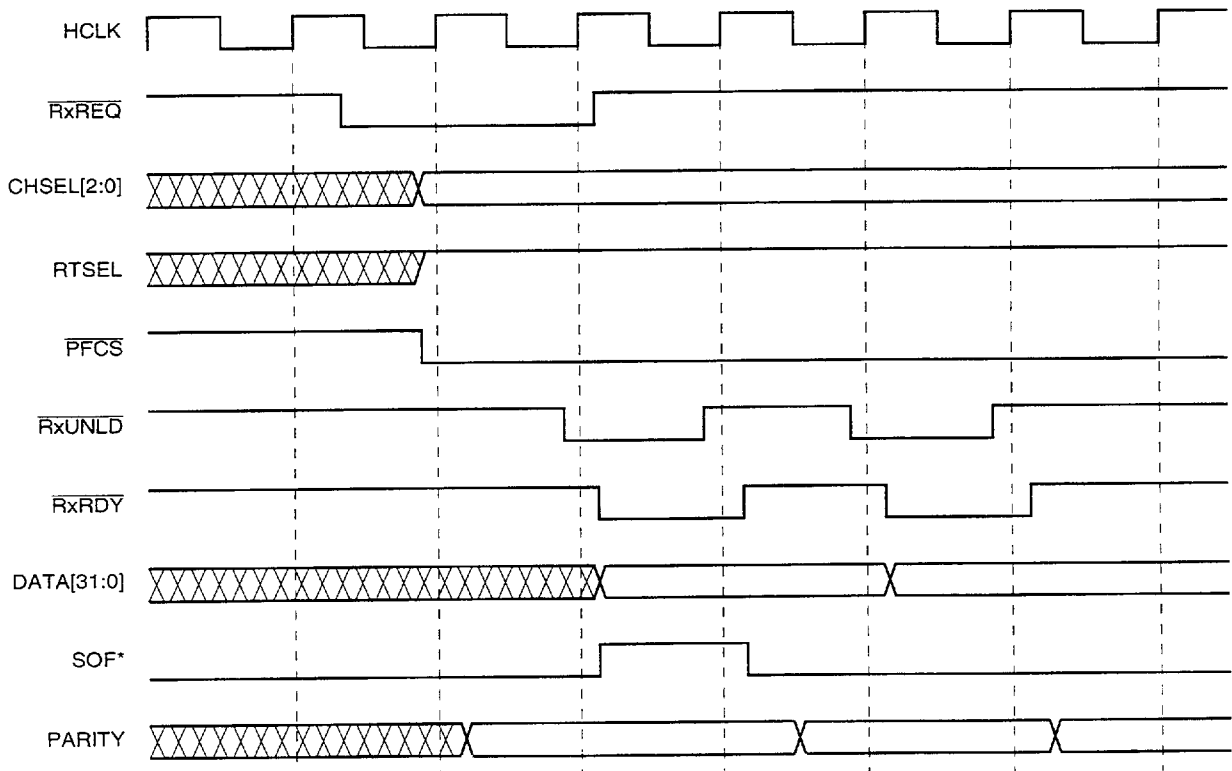


5-4367F

\* When RTSEL changes from low to high, the ATT1MX10 asserts SOF, EOF, VB1, VB0, PARITY, and DATA[31:0] after the rising edge of HCLK. These signals are deasserted after a combinatoric delay when RTSEL switches to a low.

Figure 11. Receive Statistics (32-Bit Bus)

## Timing Characteristics (continued)

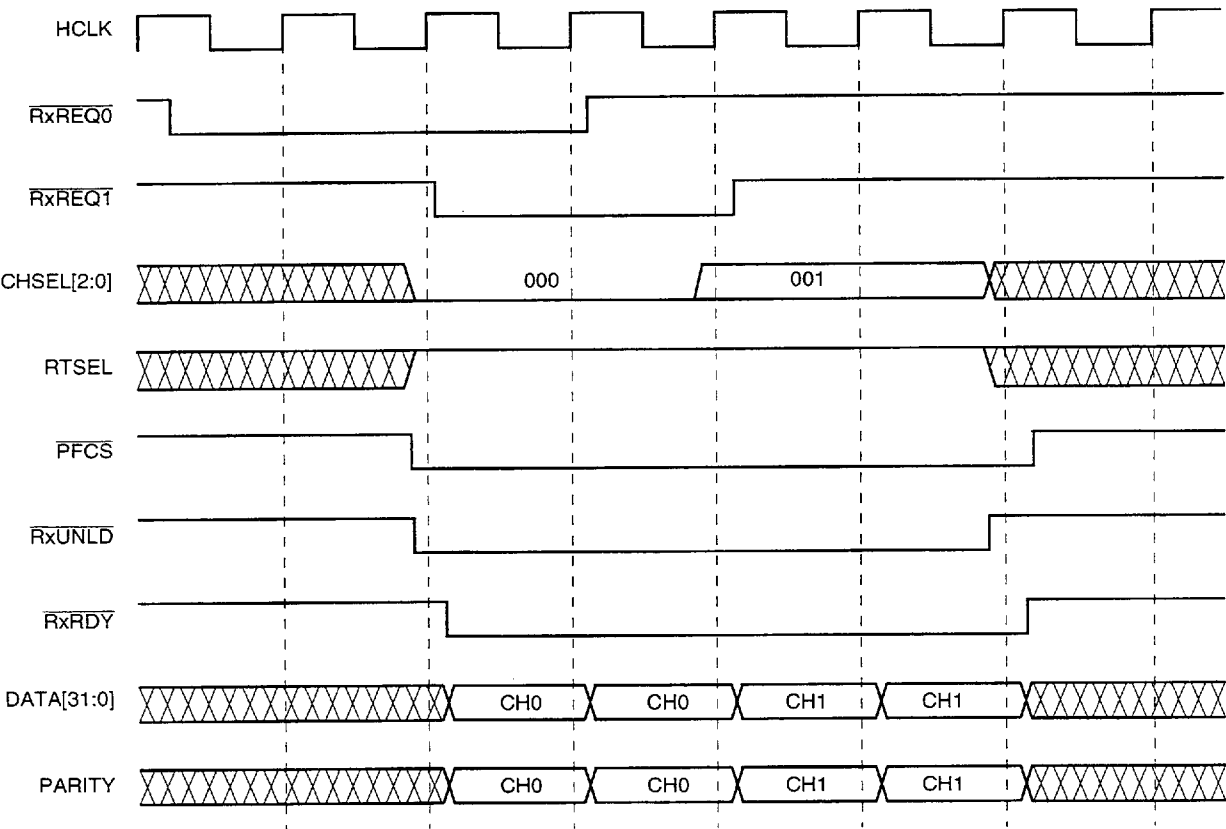


5-4368F

\* SOF will be asserted if this is the first transfer of a new frame.

**Figure 12. Rx FIFO Read, 2-Word, Multiple Cycle Access**

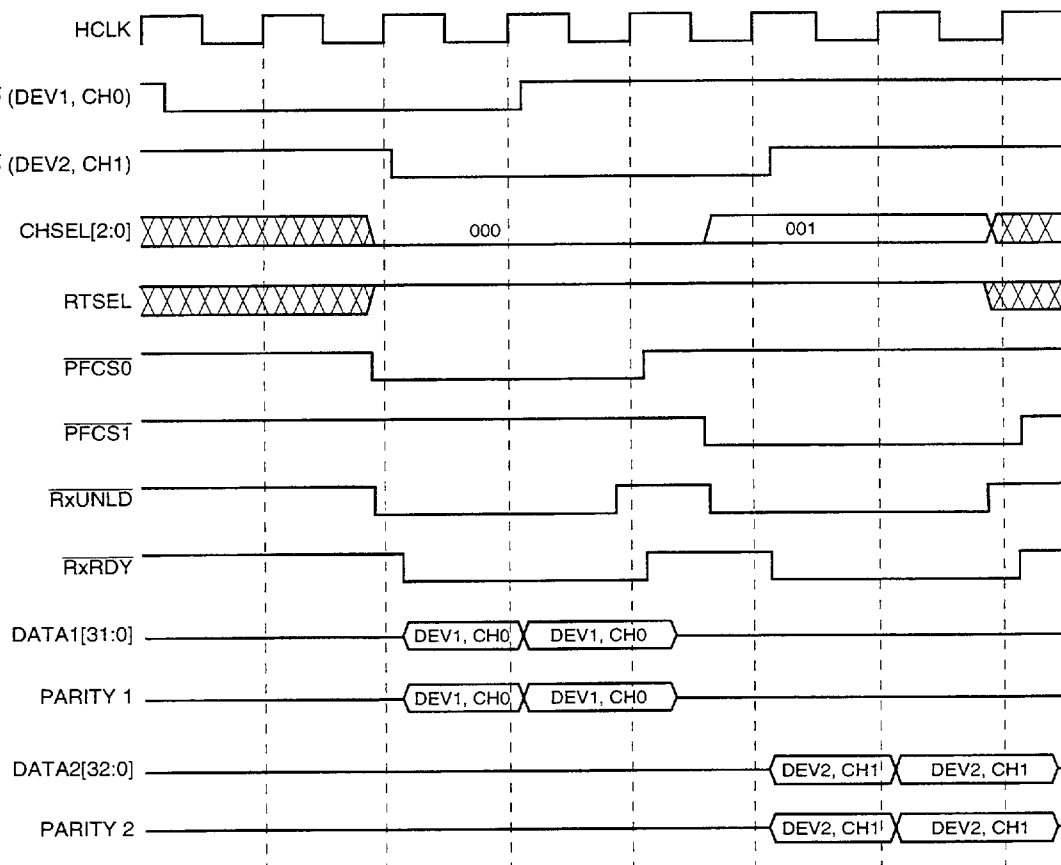
Timing Characteristics (continued)



5-4369F

Figure 13. Rx FIFO Read, 2-Word, Multiple FIFO Accesses

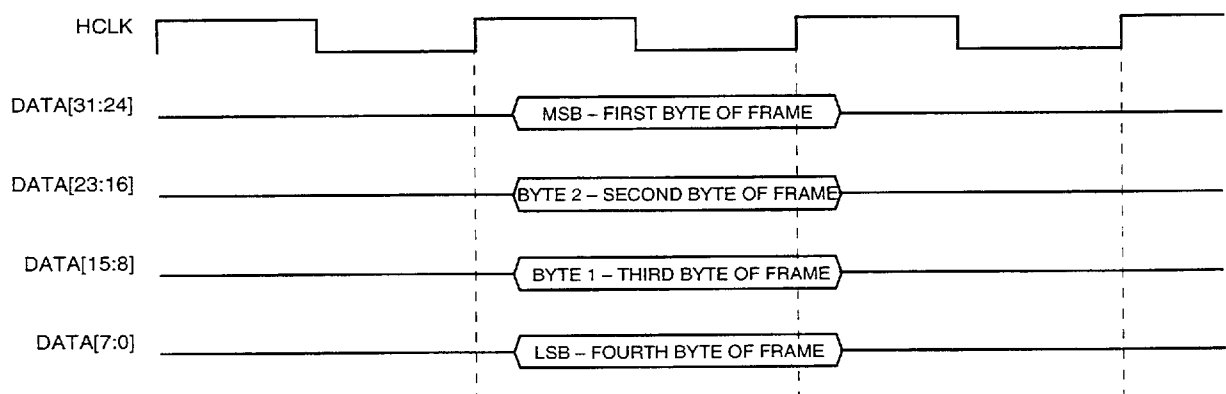
## Timing Characteristics (continued)



5-4370F

Note:  $\overline{PFCS0}$  belongs to device 1 (DEV1) and  $\overline{PFCS1}$  belongs to device 2 (DEV2).

**Figure 14. Rx FIFO Read, 2-Word, Multiple Chip Accesses**

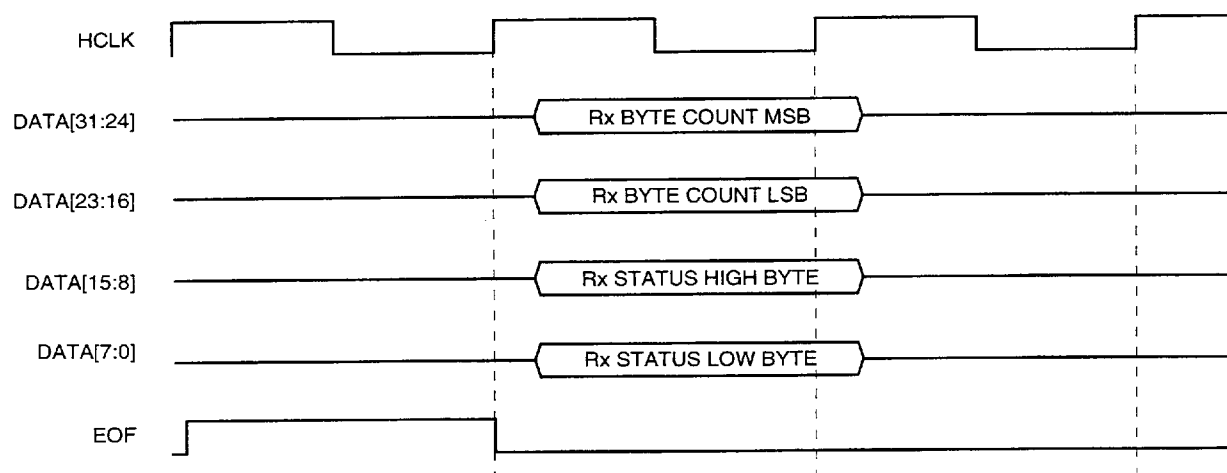


5-4371F

Note: For 16-bit bus operations, the first byte of a frame will be present on DATA[15:8]. During end of frame operations (EOF valid), DATA [31:24] must have valid data.

**Figure 15. Byte Ordering on DATA Bus During Rx and Tx FIFO Transfers**

## Timing Characteristics (continued)



5-4372F

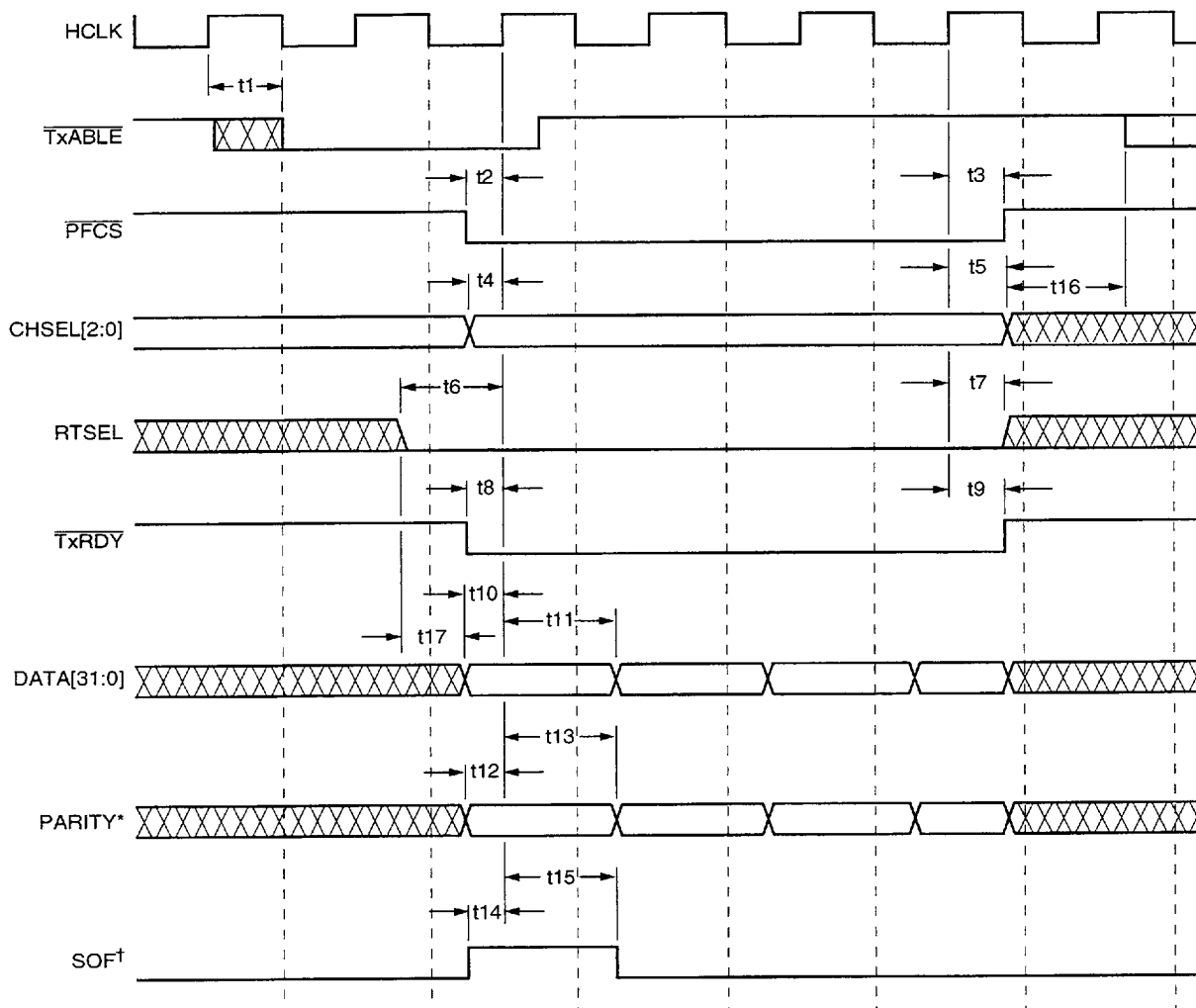
Figure 16. Byte Order on DATA Bus During Rx Status, 32-Bit Bus

Table 44. Tx FIFO Write, 4-Word, Single-Cycle Access Timing

Name	Parameter	Min	Max	Unit
t1	TxABLE Active Delay from HCLK Rising Edge	2	21	ns
t2	PFCS Setup to HCLK Rising Edge	15	—	ns
t3	PFCS Hold from HCLK Rising Edge	2	—	ns
t4	CHSEL[2:0] Setup to HCLK Rising Edge	18	—	ns
t5	CHSEL[2:0] Hold from HCLK Rising Edge	2	—	ns
t6	RTSEL Setup to HCLK Rising Edge	18	—	ns
t7	RTSEL Hold from HCLK Rising Edge	2	—	ns
t8	TxRDY Setup to HCLK Rising Edge	12	—	ns
t9	TxRDY Hold from HCLK Rising Edge	2	—	ns
t10	DATA Setup to HCLK Rising Edge	13	—	ns
t11	DATA Hold from HCLK Rising Edge	2	—	ns
t12	PARITY Setup to HCLK Rising Edge	15	—	ns
t13	PARITY Hold from HCLK Rising Edge	2	—	ns
t14	SOF Setup to HCLK Rising Edge	12	—	ns
t15	SOF Hold from HCLK Rising Edge	2	—	ns
t16	TxABLE Reassert (based on threshold)	1 HCLK cycle	—	ns
t17	RTSEL Setup to DATA Driven	1 HCLK cycle	—	ns



## Timing Characteristics (continued)



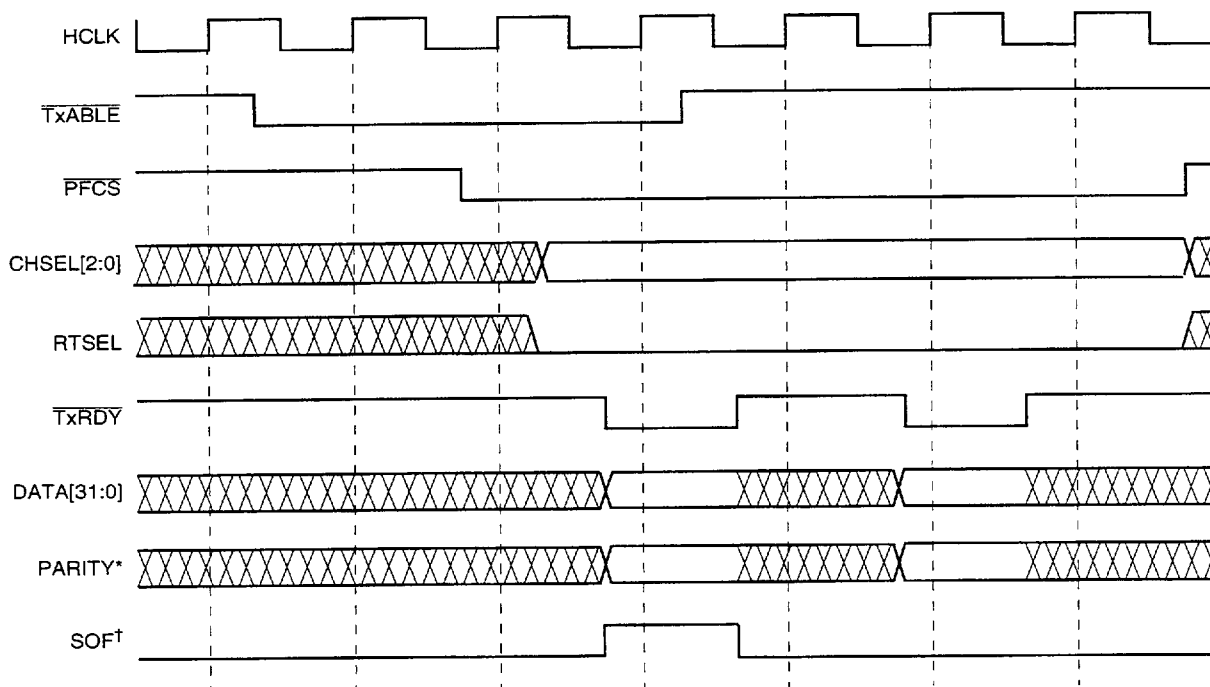
5-4374F.r2

\* PARITY as an input is not defined; the device will not differentiate between processing of packets with good parity and processing of packets with bad parity.

† SOF must be asserted if this is the first transfer of a new frame.

**Figure 17. Tx FIFO Write, 4-Word, Single-Cycle Access**

# Timing Characteristics (continued)



5-4375F.b

\* PARITY as an input is not defined; the device will not differentiate between processing of packets with good parity and processing of packets with bad parity.

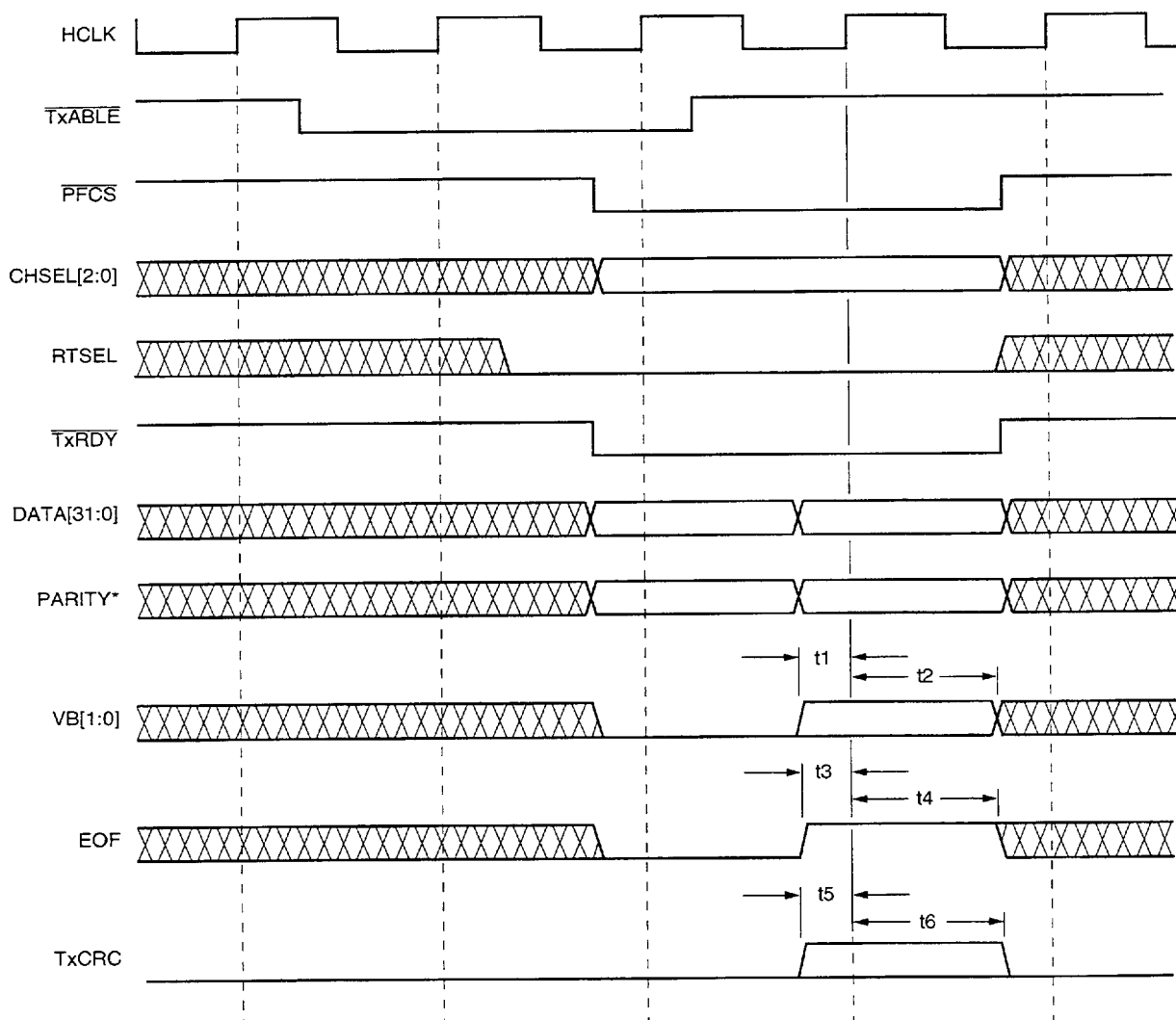
† SOF must be asserted if this is the first transfer of a new frame.

**Figure 18. Tx FIFO Write, 2-Word, Multiple Cycle Access**

## Timing Characteristics (continued)

Table 45. Tx FIFO Write, 2-Word, Single Cycle Access, Last Word Written

Name	Parameter	Min	Max	Unit
t1	VB Setup to HCLK Rising Edge	12	—	ns
t2	VB Hold from HCLK Rising Edge	2	—	ns
t3	EOF Setup to HCLK Rising Edge	12	—	ns
t4	EOF Hold from HCLK Rising Edge	2	—	ns
t5	TxCRC Setup to HCLK Rising Edge	12	—	ns
t6	TxCRC Hold from HCLK Rising Edge	2	—	ns

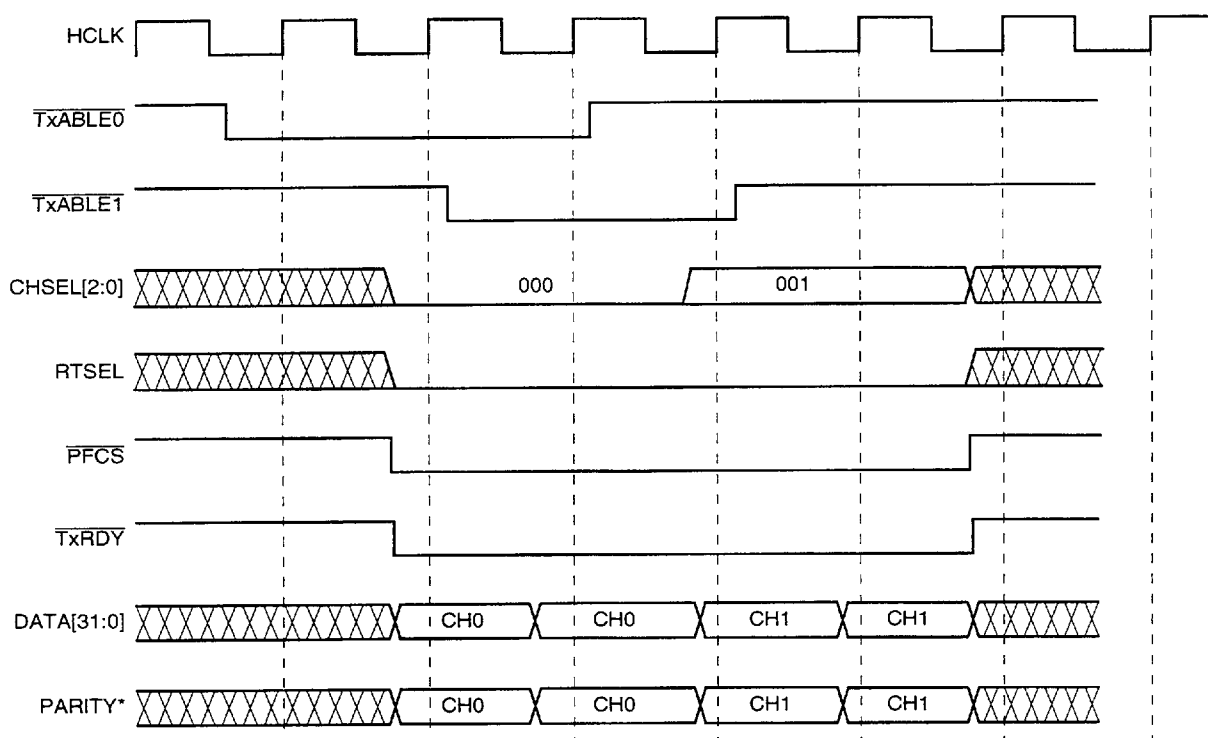


5-4376F.r2

\* PARITY as an input is not defined; the device will not differentiate between processing of packets with good parity and processing of packets with bad parity.

Figure 19. Tx FIFO Write, 2-Word, Single-Cycle Access, Last Word Written

## Timing Characteristics (continued)

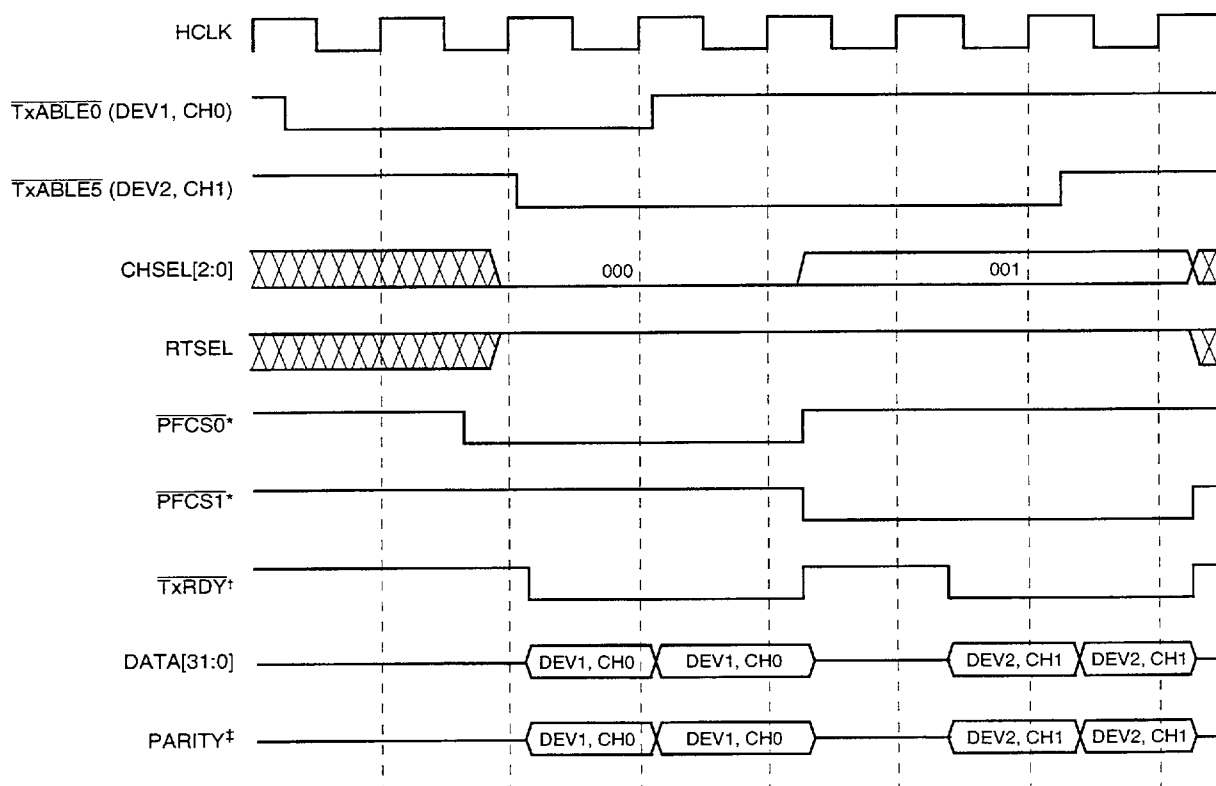


5-4377F.r3

\* PARITY as an input is not defined; the device will not differentiate between processing of packets with good parity and processing of packets with bad parity.

**Figure 20. Tx FIFO Write, 2-Word, Multiple FIFO Accesses**

## Timing Characteristics (continued)



5-4378Fr3

\*  $\overline{\text{PFCS0}}$  belongs to device 1 (DEV1) and  $\overline{\text{PFCS1}}$  belongs to device 2 (DEV2)

† If  $\text{TxRDY}$  is held low between chip accesses, the data can be driven one clock cycle earlier than shown.

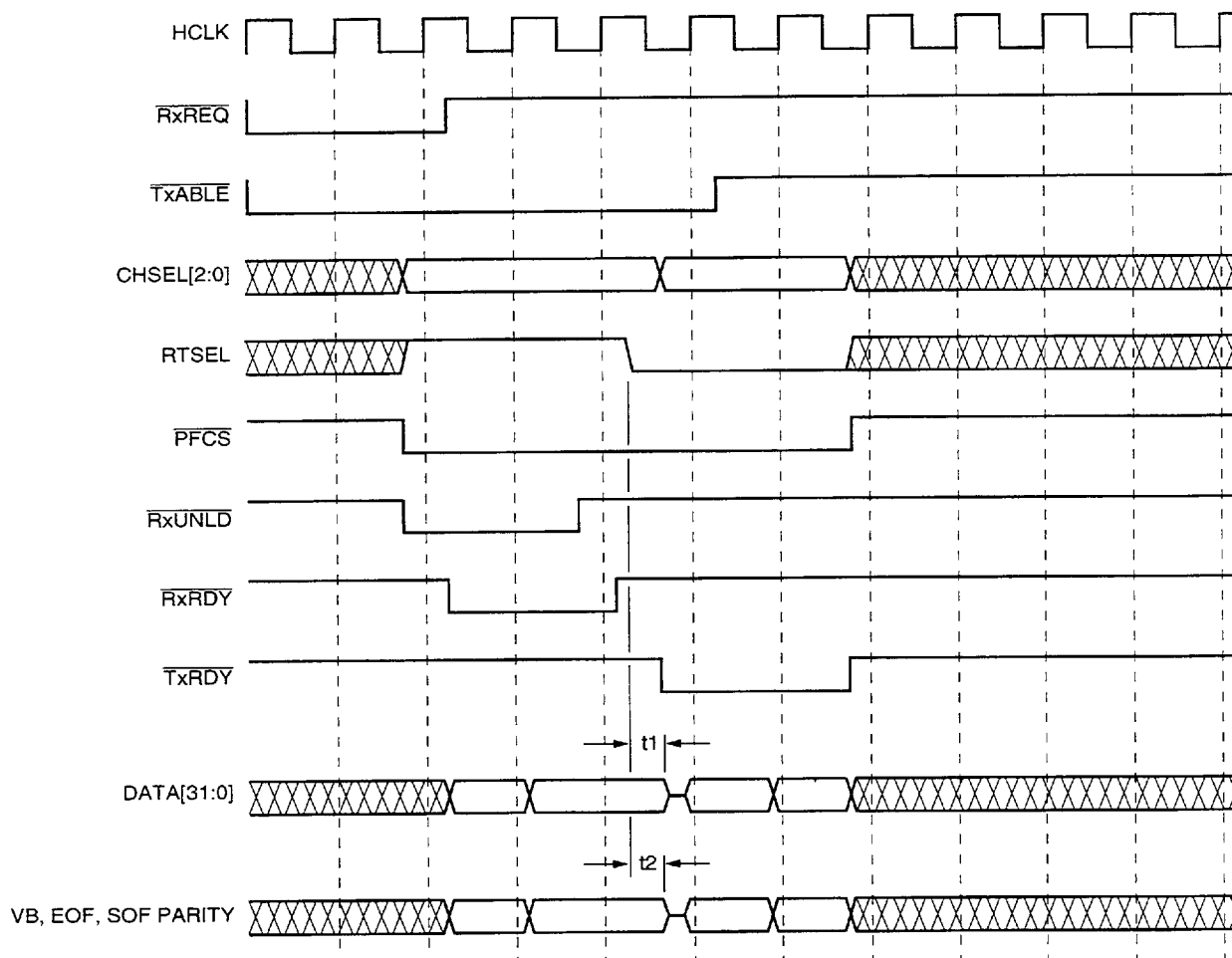
‡  $\text{PARITY}$  as an input is not defined; the device will not differentiate between processing of packets with good parity and processing of packets with bad parity.

**Figure 21. Tx FIFO Write, 2-Word, Multiple Chip Accesses**

## Timing Characteristics (continued)

Table 46. Rx FIFO Read to Tx FIFO Write Bus Turnaround Timing

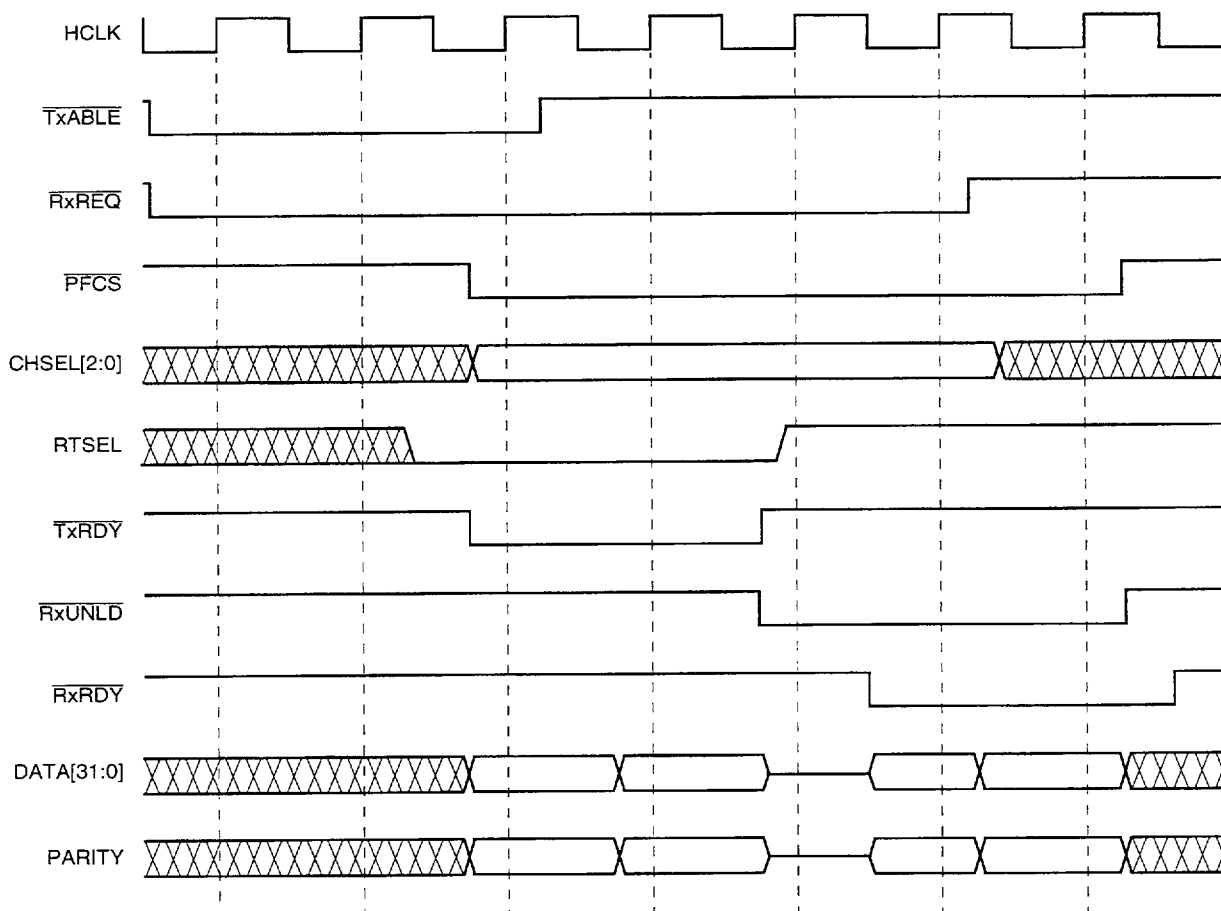
Name	Parameter	Min	Max	Unit
t1	DATA 3-state from RTSEL Low	—	15	ns
t2	PARITY 3-state from RTSEL Low	—	15	ns



5-4379F

Figure 22. Rx Read Followed by Tx Write Bus Turnaround Timing

## Timing Characteristics (continued)



5-4380F

Figure 23. Tx FIFO Write to Rx FIFO Read Timing

Timing Characteristics (continued)

Table 47. 16-Bit, Asynchronous CPU Write Timing

Name	Parameter	Min	Max	Unit
t1	CPUCS Setup to CPUSTRB Falling Edge	0	—	ns
t3	CPUAD Setup to CPUSTRB Falling Edge	0	—	ns
t4	CPUDB Setup to CPUSTRB Falling Edge	0	—	ns
t5	CPUDVAL Delay from CPUSTRB Falling Edge	0	—	ns
t6	CPUDVAL Hold from CPUSTRB Rising Edge	0	—	ns
t7	CPUAD Hold from CPUSTRB Rising Edge	0	—	ns
t8	CPUDB Hold from CPUSTRB Rising Edge	0	—	ns
t9	CPUSTRB Minimum Pulse Width	2 HCLK cycles	—	ns

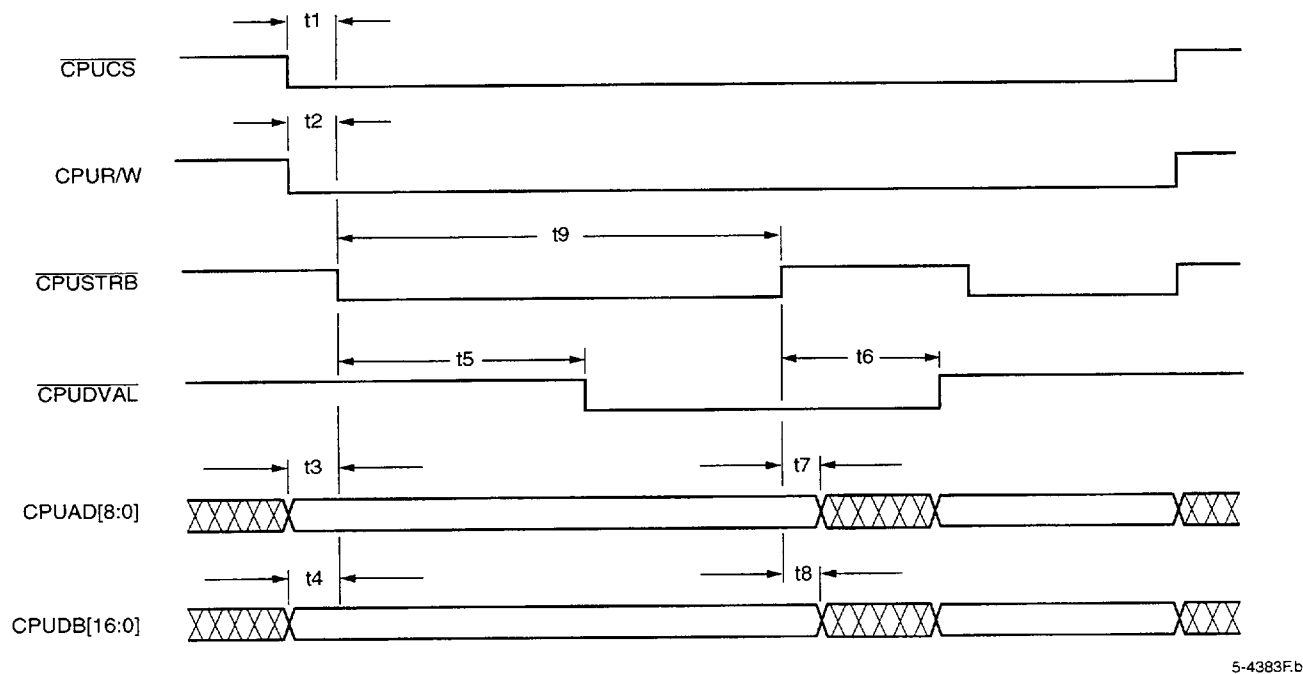


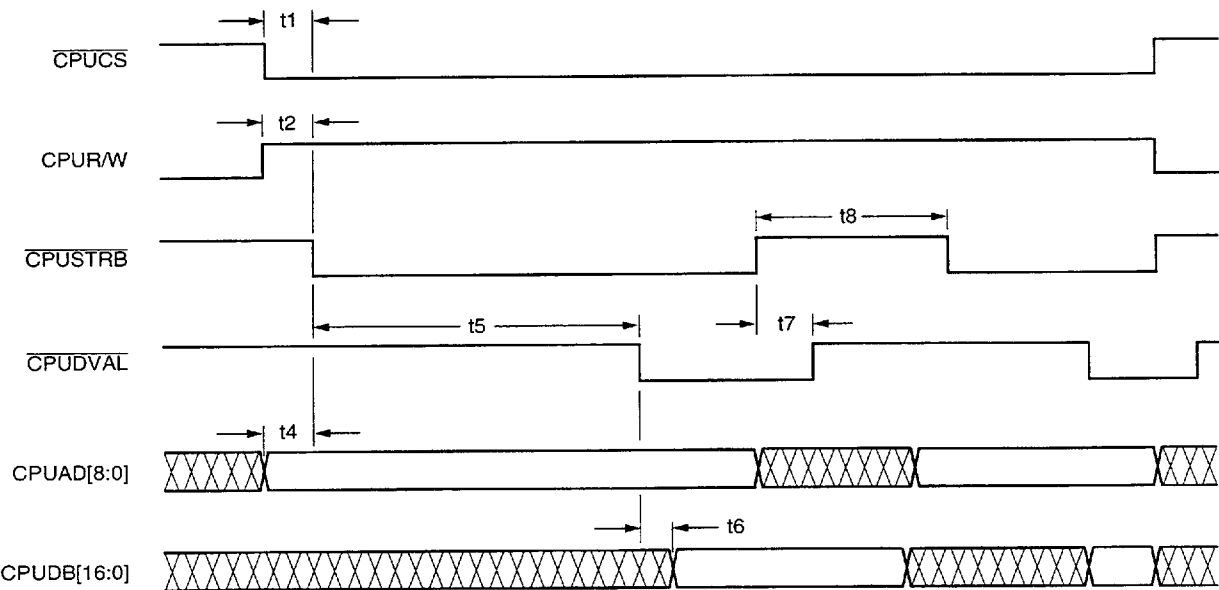
Figure 24. 16-Bit, Asynchronous CPU Write Timing



Timing Characteristics (continued)

Table 48. 16-Bit, Asynchronous CPU Read Timing

Name	Parameter	Min	Max	Unit
t1	CPUCS Setup to CPUSTRB Falling Edge	0	—	ns
t2	CPUR/W Setup to CPUSTRB Falling Edge	0	—	ns
t4	CPUAD Setup to CPUSTRB Falling Edge	0	—	ns
t5	CPUDVAL Delay from CPUSTRB Falling Edge	3 HCLK cycles	4 HCLK cycles	ns
t6	CPUDB Delay from CPUDVAL Falling Edge	10	50	ns
t7	CPUDVAL Deasserted from CPUSTRB	0	—	ns
t8	CPU Read Cycle Recovery Time	1 HCLK cycle	—	ns



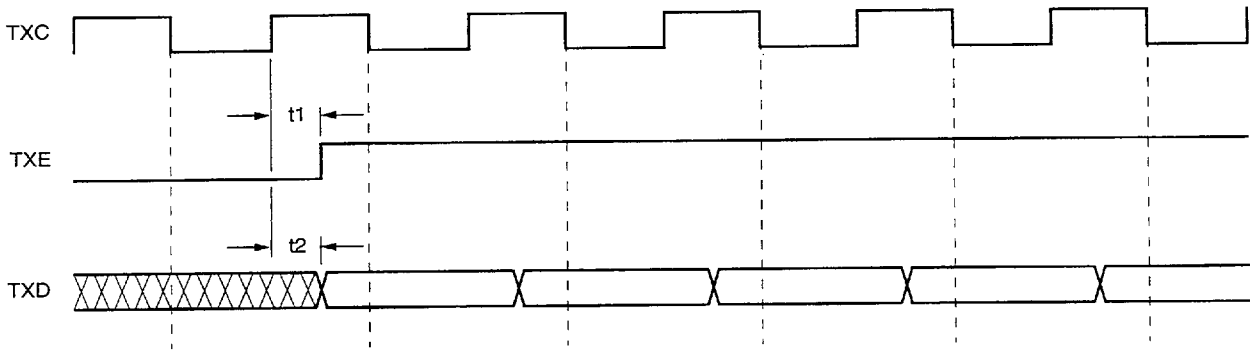
5-4384F.b

Figure 25. 16-Bit, Asynchronous CPU Read Timing

Timing Characteristics (continued)

Table 49. MAC Transmit Timing

Name	Parameter	Min	Max	Unit
t1	TXE High Delay from TXC Rising Edge	—	50	ns
t2	TXD Delay from TXC Rising Edge	—	50	ns

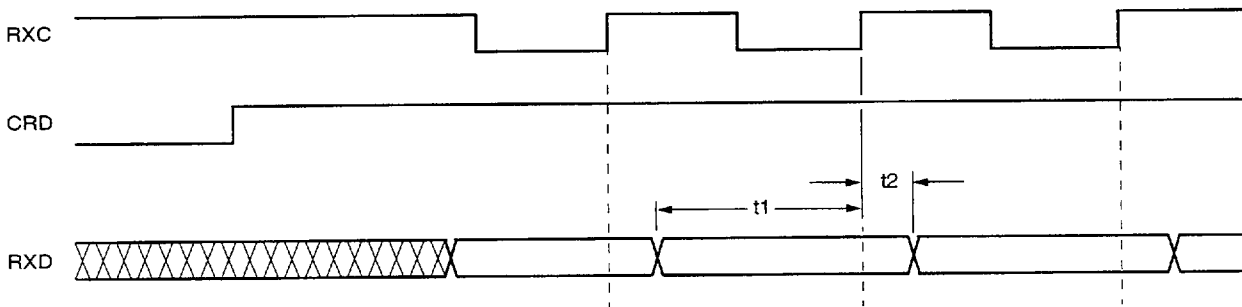


5-4385F

Figure 26. MAC Transmit Timing

Table 50. MAC Receive Timing

Name	Parameter	Min	Max	Unit
t1	RXD Setup to RXC Rising Edge	40	—	ns
t2	RXD Hold from RXC Rising Edge	30	—	ns



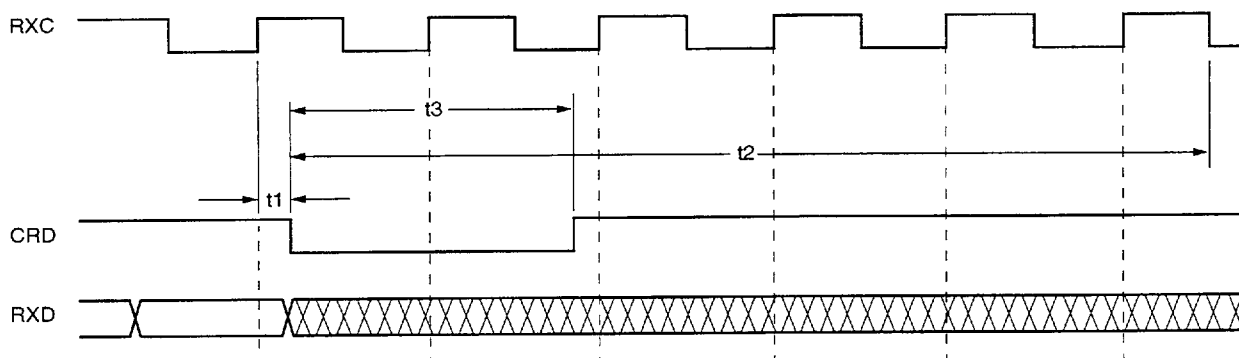
5-4386F

Figure 27. MAC Receive Timing

## Timing Characteristics (continued)

Table 51. MAC Receive Clock Timing

Name	Parameter	Min	Max	Unit
t1	CRS Hold From RXC Rising Edge	—	150	ns
t2	Number of Clocks Required After CRD	5 RXCs	—	ns
t3	Carrier Reassert Time	5 RXCs	—	ns

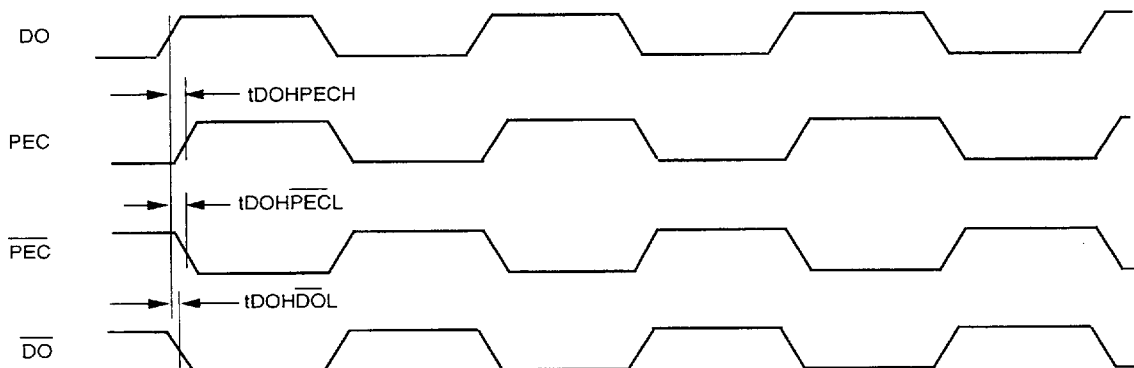


5-4387F

Figure 28. MAC Receive Clock Timing

Table 52. TP Driver Specification Timing

Name	Parameter	Min	Max	Unit
tDOHPECL	Pre-Equalization Skew	0	±10	ns
tDOHPECH	Pre-Equalization Skew	0	±10	ns
tDOHDOL	Pre-Equalization Skew	0	±10	ns



5-3037C

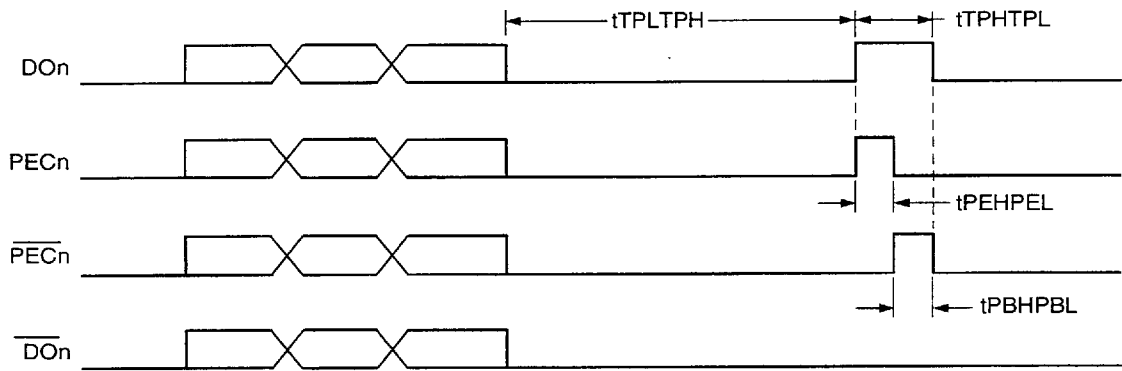
Figure 29 TP Driver Specification Timing

Timing Characteristics (continued)

Table 53. TP Link-Integrity Timing\*

Name	Parameter	Min	Max	Unit
tTPLTPH	Time Between Consecutive Transmitted Link-Integrity Pulses	8	24	ms
tPHTPL	Width of DO for Link-Integrity Pulse	80	120	ns
tPEHPEL, tPBHPBL	Width of PEC for Link-Integrity Pulse	40	60	ns
tSKEW(DO-PEC)	Skew Between Rising Edge of DO and Rising Edge of PEC During Link Pulse Transmission	0	±10	ns
tSKEW(PEC-PEC)	Skew Between Falling Edge of PEC and Rising Edge of PEC During Link Pulse Transmission	0	±10	ns
tSKEW(DO-PEC)	Skew Between Falling Edge of DO and Rising Edge of PEC During Link Pulse Transmission	0	±10	ns

\* Refer to Figure 30.



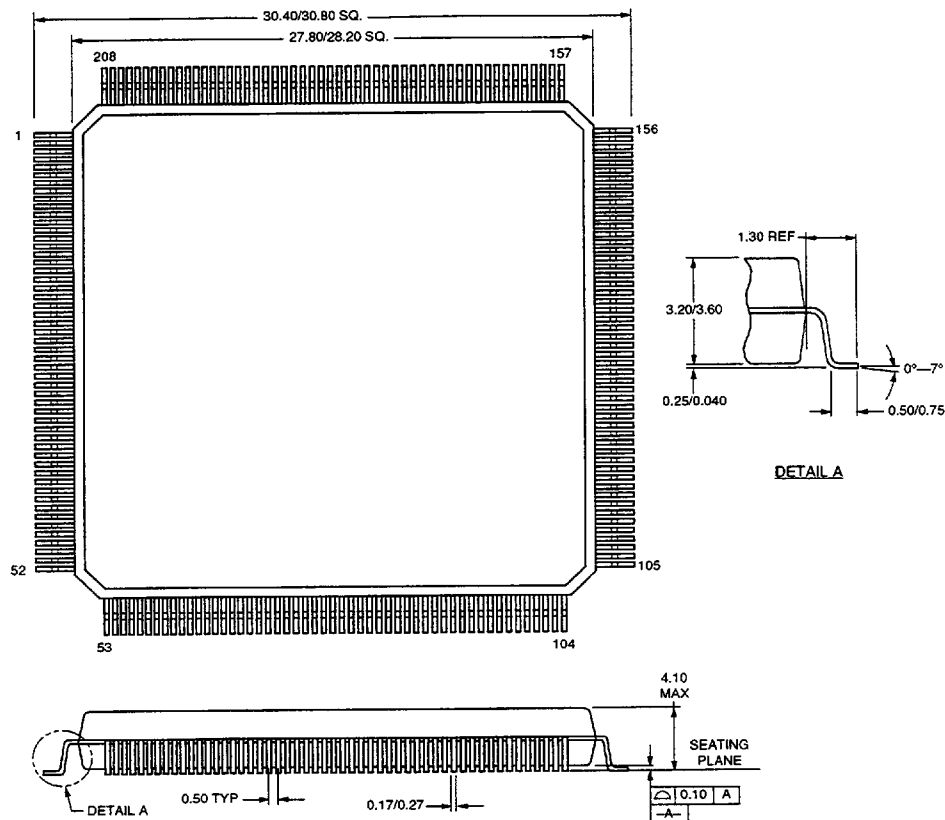
5-1089C

Figure 30. Link-Integrity Timing

Outline Diagram

208-Pin SQFPH, 1.3 mm Lead Frame

The controlling dimensions are in millimeters.



5-2196.r13

Ordering Information

Device Code	Package	Temperature
ATT1MX10-SC	208-Pin SQFPH	0 °C to 70 °C