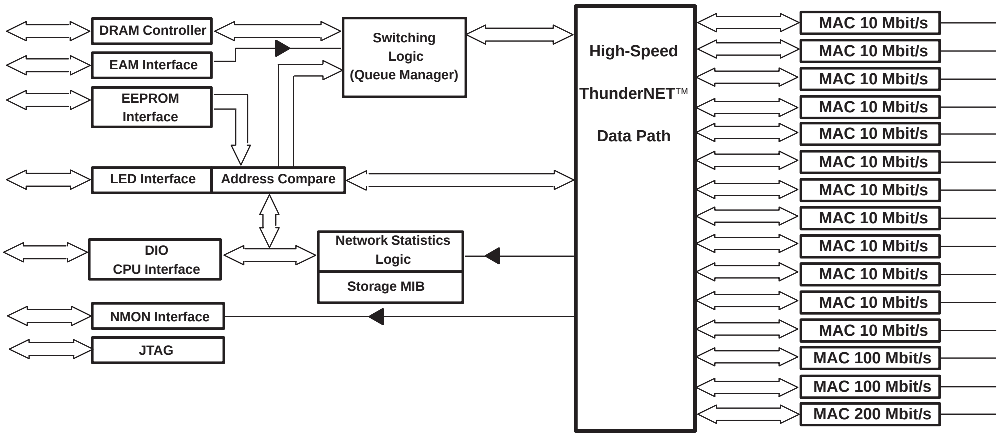
- **Highly Integrated Single-Chip Shared-Memory Switch**
- **Provides 12 10-Mbit/s Ethernet Ports, 2 10-/100-Mbit/s Ethernet Ports, and a 10-/100-/200-Mbit/s Ethernet Port**
- **Supports Cut-Through and Store-and-Forward Modes of Switching Operation**
- **All Ports Are Full-Duplex Capable**
- **10-/100-/200-Mbit/s Ethernet Port Capable of Cascading Multiple Devices, Bidirectional Tagging, and Unidirectional Flow Control**
- **Supports On-Chip-Per-Port Storage for Etherstat™ and Remote Monitoring (RMON) Management-Information Base (MIB)**
- **Network Monitoring (NMON) Supports Port Mirroring Feature**
- **Supports Multiple and Single Media-Access Controller (MAC) Address Switching Modes**
- **Virtual-LAN (VLAN) Capable**
- **Provides Direct Connection to Industry-Standard PHY Components:**
  - **10-Mbit/s Ports – Serial Network Interface (SNI)**
  - **10-/100-Mbit/s Ports – SNI for 10-Mbit/s and Media-Independent Interface (MII) for 100 Mbit/s**
- **Provides Support for Simple-Network Management Protocol (SNMP) and Spanning Tree (IEEE Std 802.1D)**
- **EEPROM Interface for Auto-Configuration, No CPU Needed, Enabling Lowest Cost Desktop Switching Solution**
- **Supports In-Order Broadcast/Multicast Traffic (All Packets Are Transmitted in Same Order as Received)**
- **Provides an External Address Match (EAM) Interface**
- **Packet Memory Utilizes Cost-Effective Extended Data Out (EDO) DRAM**
- **Provides Direct Input/Output (DIO) Interface for Configuration and Statistics Information**
- **Provides Transmit Pacing**
- **LED Interface for Port Status and TXQ Status**
- **Fabricated in 3.3-V Low-Voltage Technology**
- **Packaged in 352-Pin Plastic Ball Grid Array Package**
- **5-V Tolerant I/Os**
- **JTAG Compliant**

description

The ThunderSWITCH TNETX3150/TNETX3150A is a 15-port 10-/100-Mbit/s Ethernet shared-memory switch. The TNETX3150/TNETX3150A provides a complete low-cost switch solution. A desktop solution can be achieved by combining the TNETX3150/TNETX3150A, the necessary physical interfaces, and low-cost packet memory.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

ThunderNET and ThunderSWITCH are trademarks of Texas Instruments Incorporated.
Ethernet and Etherstat are trademarks of Xerox Corporation.

description (continued)

All 15 ports are full-duplex capable. With full-duplex capability, ports 03–14 can support 20-Mbit/s connections and ports 01–02 can support 200-Mbit/s connections to desktops or high-speed servers through external physical (PHY) layers. The uplink port 00 can provide a 400-Mbit/s (200 Mbit/s full duplex) connection for either external-switching fabric, external routing engine, or high-speed stacking links. ThunderSWITCH is capable of supporting VLAN for workgroup and segment-switching applications.

Data received from the media-access controller (MAC) interface is buffered in the RX FIFO before storage in external buffer memory under control of the queue management (QM) logic. Both QM and MAC blocks apply round-robin arbitration to maintain bandwidth and fast data transfer of the first-in, first-out (FIFO) subsystem without contention.

The TNETX3150/TNETX3150A has three modes for making forwarding decisions: internal single-address compare, external address match (EAM) interface, and frame tagging on the uplink port. All the 10-Mbit/s ports internally support switching a single MAC address per port. An EAM interface is supplied to support multiple addresses per port. External-address-matching hardware is required to support multiple users on one port. Ports 01 and 02 (10-/100-Mbit/s high-speed connection) are similarly restricted. The uplink port 00 does not have any internal address register and can support multiple addresses.

Upon transmission, frame data is obtained from the buffer memory and buffered temporarily into the TX FIFO before transmission on the relevant MAC port. If a collision occurs during transmission, data recovery and retransmission occur from the TX FIFO.

Statistics for the Etherstat and RMON management-information base are independently collected for each of the 15 ports. Access to the statistics counters is provided via the direct input/output (DIO) interface. The DIO interface is intended only for configuration and monitoring operations.

The TNETX3150/TNETX3150A utilizes a standard 60-ns memory solution (EDO DRAM).The packet memory has been implemented to effectively support both single-access operation and page-burst-access operation. All DRAM buffer transfers are made within a page boundary, permitting fast-burst accesses. A high-memory bandwidth is maintained, allowing all ports to be active without bottlenecking at the memory buffer. The TNETX3150/TNETX3150A allows any port configuration, including those that may exceed the maximum buffer-memory bandwidth. This can cause packets to be dropped. To avoid these conditions, the port configurations should be restricted so that the maximum allowable bandwidth to external buffer memory is not exceeded.

Packets are routed to local ports based on the destination MAC address. The cut-through feature enables transmission on the destination port before complete reception of the frame, which reduces the overall switch latency. The cut-through feature can be employed for all situations where the transmission port is slower or equal to the data rate on the receiving port. 100 Mbit/s can cut through to another 100-Mbit/s port or a 10-Mbit/s port. However, a 10-Mbit/s port cannot cut through to a 100-Mbit/s port. For this case, local cut-through is automatically disabled to prevent underflow. The cut-through feature can be disabled by either the receiving port or transmitting port, on a per-port basis, by setting the store-and-forward option bits in the port-control registers.

If the TNETX3150/TNETX3150A is operating in store-and-forward mode, the entire frame must be received before it is transmitted to the destination port. This provides the ability to filter frames that contain errors. This mode can be configured either by using the port control registers or is used automatically when transmitting from a low-speed to a high-speed port, since cut-through is not possible in this case.

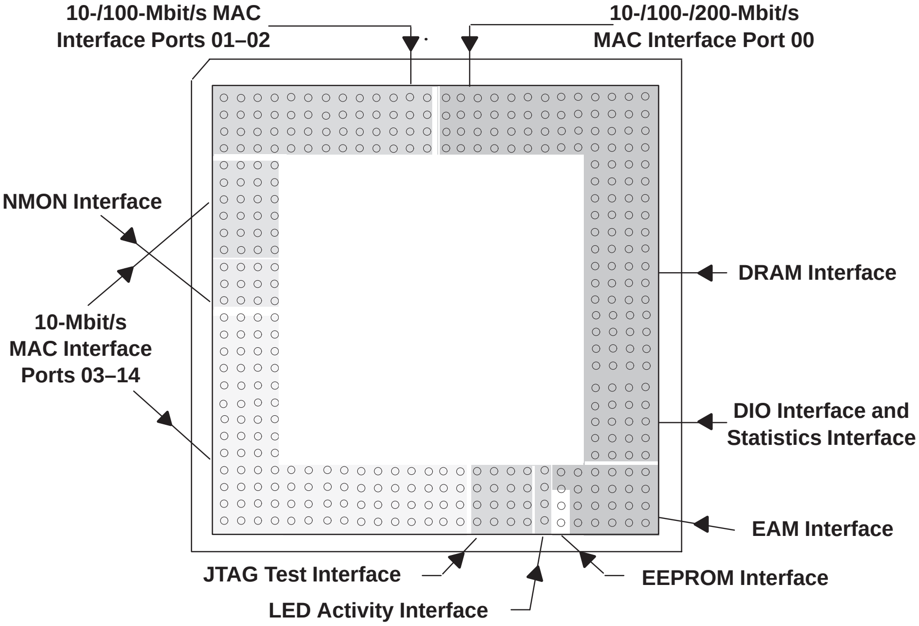
description (continued)

The TNETX3150/TNETX3150A supports VLANs. This can be done two ways: 1) In the single-address-per-port mode, internal VLAN registers are used to configure the destination ports for broadcast/multicast packets. 2) The EAM interface provides a hardware mechanism to control broadcast/multicast packets based on the code provided on that interface. See *EAM interface* and *VLAN support* for more details.

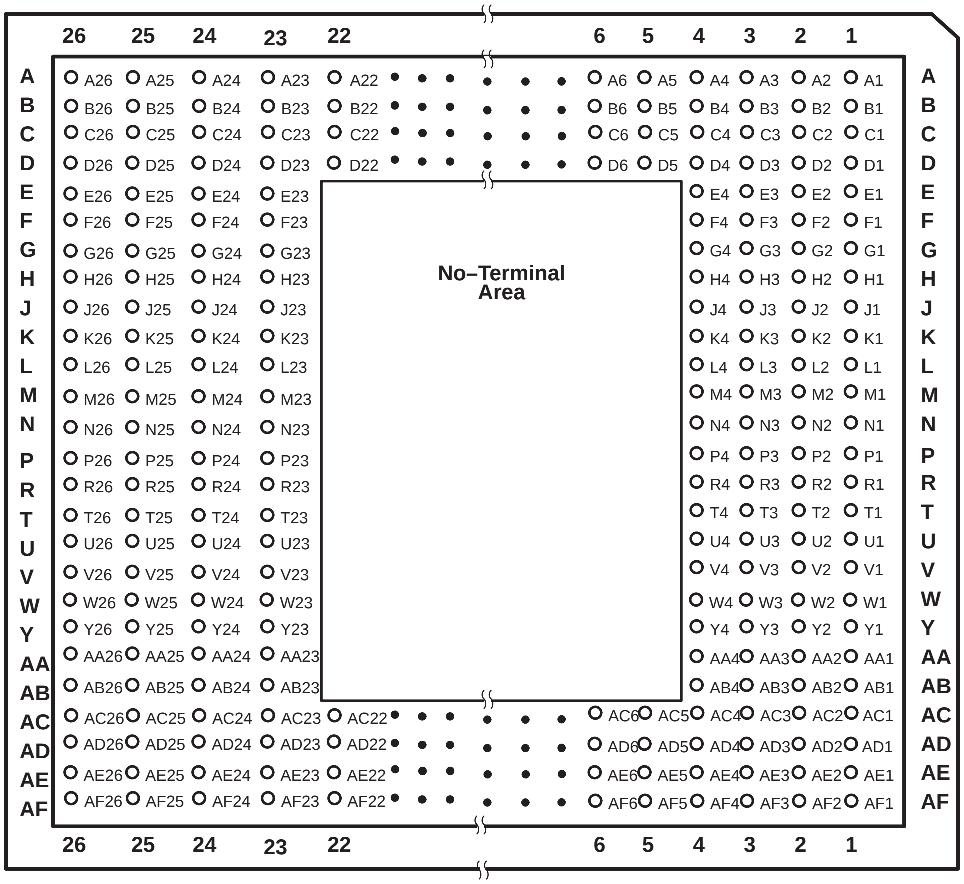
The TNETX3150/TNETX3150A is fabricated with a 3.3-V technology. The inputs are 5-V tolerant and the 3.3-V outputs can directly interface to 5-V TTL-logic levels. This provides the customer with a broad choice of interfacing-device options.

The TNETX3150/TNETX3150A is JTAG compliant with one exception; they do require external pullup resistors on the TDI, TMS, and $\overline{\text{TRST}}$ input terminals.

GGP PACKAGE
(TOP VIEW)


10-/100-Mbit/s MAC Interface Ports 01–02
10-/100-/200-Mbit/s MAC Interface Port 00
NMON Interface
10-Mbit/s MAC Interface Ports 03–14
DRAM Interface
DIO Interface and Statistics Interface
EAM Interface
JTAG Test Interface
EEPROM Interface
LED Activity Interface

VIEW FROM BOTTOM WITH SOLDER BALLS FACING VIEWER



	26	25	24	23	22							6	5	4	3	2	1	
A	○ A26	○ A25	○ A24	○ A23	○ A22	●	●	●	●	●	●	○ A6	○ A5	○ A4	○ A3	○ A2	○ A1	A
B	○ B26	○ B25	○ B24	○ B23	○ B22	●	●	●	●	●	●	○ B6	○ B5	○ B4	○ B3	○ B2	○ B1	B
C	○ C26	○ C25	○ C24	○ C23	○ C22	●	●	●	●	●	●	○ C6	○ C5	○ C4	○ C3	○ C2	○ C1	C
D	○ D26	○ D25	○ D24	○ D23	○ D22	●	●	●	●	●	●	○ D6	○ D5	○ D4	○ D3	○ D2	○ D1	D
E	○ E26	○ E25	○ E24	○ E23										○ E4	○ E3	○ E2	○ E1	E
F	○ F26	○ F25	○ F24	○ F23										○ F4	○ F3	○ F2	○ F1	F
G	○ G26	○ G25	○ G24	○ G23										○ G4	○ G3	○ G2	○ G1	G
H	○ H26	○ H25	○ H24	○ H23				No–Terminal Area						○ H4	○ H3	○ H2	○ H1	H
J	○ J26	○ J25	○ J24	○ J23										○ J4	○ J3	○ J2	○ J1	J
K	○ K26	○ K25	○ K24	○ K23										○ K4	○ K3	○ K2	○ K1	K
L	○ L26	○ L25	○ L24	○ L23										○ L4	○ L3	○ L2	○ L1	L
M	○ M26	○ M25	○ M24	○ M23										○ M4	○ M3	○ M2	○ M1	M
N	○ N26	○ N25	○ N24	○ N23										○ N4	○ N3	○ N2	○ N1	N
P	○ P26	○ P25	○ P24	○ P23										○ P4	○ P3	○ P2	○ P1	P
R	○ R26	○ R25	○ R24	○ R23										○ R4	○ R3	○ R2	○ R1	R
T	○ T26	○ T25	○ T24	○ T23										○ T4	○ T3	○ T2	○ T1	T
U	○ U26	○ U25	○ U24	○ U23										○ U4	○ U3	○ U2	○ U1	U
V	○ V26	○ V25	○ V24	○ V23										○ V4	○ V3	○ V2	○ V1	V
W	○ W26	○ W25	○ W24	○ W23										○ W4	○ W3	○ W2	○ W1	W
Y	○ Y26	○ Y25	○ Y24	○ Y23										○ Y4	○ Y3	○ Y2	○ Y1	Y
AA	○ AA26	○ AA25	○ AA24	○ AA23										○ AA4	○ AA3	○ AA2	○ AA1	AA
AB	○ AB26	○ AB25	○ AB24	○ AB23										○ AB4	○ AB3	○ AB2	○ AB1	AB
AC	○ AC26	○ AC25	○ AC24	○ AC23	○ AC22	●	●	●	●	●	●	○ AC6	○ AC5	○ AC4	○ AC3	○ AC2	○ AC1	AC
AD	○ AD26	○ AD25	○ AD24	○ AD23	○ AD22	●	●	●	●	●	●	○ AD6	○ AD5	○ AD4	○ AD3	○ AD2	○ AD1	AD
AE	○ AE26	○ AE25	○ AE24	○ AE23	○ AE22	●	●	●	●	●	●	○ AE6	○ AE5	○ AE4	○ AE3	○ AE2	○ AE1	AE
AF	○ AF26	○ AF25	○ AF24	○ AF23	○ AF22	●	●	●	●	●	●	○ AF6	○ AF5	○ AF4	○ AF3	○ AF2	○ AF1	AF
	26	25	24	23	22							6	5	4	3	2	1	

Table of Contents

Terminal Functions

10-Mbit/s MAC interface (ports 03–14)

TERMINAL		I/O	DESCRIPTION
NAME	NO.		

M03COL	F4	I	Collision sense. In carrier sense multiple access with collision detection (CSMA/CD) mode, assertion of MXXCOL indicates network collision.
			High = network collision
			Low = no network collision

M04COL	H4		
M05COL	L4		
M06COL	T4		
M07COL	W4		
M08COL	AA4		
M09COL	AC2		
M10COL	AC6		
M11COL	AC8		
M12COL	AC11		
M13COL	AC14		
M14COL	AD17		
M03CRS	F3	I	Carrier sense. MXXCRS indicates a frame carrier signal is being received. High = frame carrier signal is being received Low = no frame carrier signal is being received
M04CRS	H3		
M05CRS	L3		
M06CRS	T3		
M07CRS	W3		
M08CRS	AA3		
M09CRS	AF3		
M10CRS	AD6		
M11CRS	AD8		
M12CRS	AD11		
M13CRS	AC15		
M14CRS	AC17		
M03DUPLEX	E4	I/O	Duplex selector. MXXDUPLEX switches the interface between full- and half-duplex operation (low = half duplex, high = full duplex). M03DUPLEX–M14DUPLEX inputs have active pulldown circuits used to force the outputs low when the FORCEHD bit is set.
M04DUPLEX	G4		
M05DUPLEX	K4		
M06DUPLEX	R4		
M07DUPLEX	U4		
M08DUPLEX	Y4		
M09DUPLEX	AC3		
M10DUPLEX	AC5		
M11DUPLEX	AC7		
M12DUPLEX	AC10		
M13DUPLEX	AD12		
M14DUPLEX	AC16		

Terminal Functions (Continued)

10-Mbit/s MAC interface (ports 03–14) (continued)

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
M03LINK M04LINK M05LINK M06LINK M07LINK M08LINK M09LINK M10LINK M11LINK M12LINK M13LINK M14LINK	E3 G3 K3 R3 U3 AA2 AB4 AD5 AD7 AD10 AD14 AD16	I	Link selector. MXXLINK indicates the condition of the port connection (low = no link, high = link).
M03RCLK M04RCLK M05RCLK M06RCLK M07RCLK M08RCLK M09RCLK M10RCLK M11RCLK M12RCLK M13RCLK M14RCLK	F1 H1 K2 R2 V1 Y3 AC1 AF6 AF8 AE10 AD13 AE16	I	Receive clock. MXXRCLK is the receive clock from the attached PHY or test device.
M03RXD M04RXD M05RXD M06RXD M07RXD M08RXD M09RXD M10RXD M11RXD M12RXD M13RXD M14RXD	E1 H2 L2 T2 V3 Y2 AD1 AF5 AE8 AE11 AE13 AF16	I	Receive data. MXXRXD is the receive data. Data is synchronous to MXXRCLK.

Terminal Functions (Continued)

10-Mbit/s MAC interface (ports 03–14) (continued)

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
M03TCLK	G2	I	Transmit clock. MXXTCLK is the transmit clock from the attached PHY or test device.
M04TCLK	K1		
M05TCLK	M2		
M06TCLK	U2		
M07TCLK	Y1		
M08TCLK	AA1		
M09TCLK	AE4		
M10TCLK	AE7		
M11TCLK	AF10		
M12TCLK	AE12		
M13TCLK	AE15		
M14TCLK	AF18		
M03TXD	G1	O	Transmit data. MXXTXD is the transmit data leaving port XX when MXXTXEN is asserted.
M04TXD	J1		
M05TXD	M1		
M06TXD	U1		
M07TXD	W1		
M08TXD	AB1		
M09TXD	AF4		
M10TXD	AF7		
M11TXD	AF9		
M12TXD	AF12		
M13TXD	AF15		
M14TXD	AF17		
M03TXEN	F2	O	Transmit enable. MXXTXEN indicates valid transmit data on MXXTXD. High = valid transmit data on MXXTXD Low = no valid transmit data on MXXTXD
M04TXEN	J3		
M05TXEN	L1		
M06TXEN	T1		
M07TXEN	W2		
M08TXEN	AB3		
M09TXEN	AD4		
M10TXEN	AE6		
M11TXEN	AD9		
M12TXEN	AF11		
M13TXEN	AD15		
M14TXEN	AE17		

Terminal Functions (Continued)

10-/100-Mbit/s MAC interface (ports 01–02)

TERMINAL		I/O	DESCRIPTION
NAME	NO.		

M01COL
M02COL
	C8
	D7
		I	Collision sense. Assertion of MXXCOL indicates network collision.
			High = network collision
			Low = no network collision

M01CRS
M02CRS
	B8
	D3
		I	Carrier sense. MXXCRS indicates a frame carrier signal is being received.
			High = frame carrier
			Low = no frame carrier

M01PROTOCOL M02PROTOCOL	A8 D1	I	Interface protocol. MXXPROTOCOL indicates the protocol of the interface and is determined by the logic level on the terminal. When high, it indicates a request/grant interface for datastreaming applications. When low, it indicates a CSMA/CD interface. For standard Ethernet applications, MXXPROTOCOL should be held low.
M01DUPLEX M02DUPLEX	C9 C1	I/O	Duplex selector. MXXDUPLEX switches the interface between full- and half-duplex operation (low = half duplex, high = full duplex). The M01DUPLEX and M02DUPLEX inputs have active pulldown circuits used to force the outputs low when the FORCEHD bit is set.

M01LINK
M02LINK
	D11
	D2
		I	Link selector. MXXLINK indicates the condition of the port connection.
			High = link
			Low = no link

M01RCLK M02RCLK	A9 A3	I	Receive clock. MXXRCLK source is from the external PHY or test device.

M01RXDV
M02RXDV
	C11
	C4
		I	Receive data valid. MXXRXDV indicates data on MXXRXD3–MXXRXD0 is valid.
			High = valid data
			Low = no valid data

M01RXER
M02RXER
	B11
	A5
		I	Receive error. MXXRXER indicates reception of a coding error on received data.
			High = coding error
			Low = no coding error

M01RXD3 M02RXD3	A10 C5	I	Receive data. Nibble receive data MXXRXD3–MXXRXD0 is from the physical media dependent (PMD) front end. Data is synchronous to MXXRCLK. If the MWIDTH bit in the port control register and the M00SPEED terminal are set low, then data is received on M01RXD0 and M02RXD0 terminals.
M01RXD2 M02RXD2	B10 D5		
M01RXD1 M02RXD1	C10 A4		
M01RXD0 M02RXD0	D10 B4		

M01SPEED
M02SPEED
	A11
	D6
		I	Bit-rate selection speed. The speed of the MAC interface is determined by the logic level on MXXSPEED.
			High = 100 Mbit/s
			Low = 10 Mbit/s

M01TCLK M02TCLK	C12 C6	I	Transmit clock. MXXTCLK source is from the external PHY or test device.

Terminal Functions (Continued)

10-/100-Mbit/s MAC interface (ports 01–02) (continued)

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
M01TXD3	C13	O	Transmit data. Nibble transmit data is from the TNETX3150/TNETX3150A. When MXXTXEN is asserted, these terminals carry transmit data. Data on these terminals is always synchronous to MXXTCLK. If the MWIDTH bit in the port control register and the M00SPEED terminal are set low, then data is transmitted on M01TXD0 and M02TXD0.
M02TXD3	B7		
M01TXD2	D13		
M02TXD2	C7		
M01TXD1	A12		
M02TXD1	A6		
M01TXD0	B12		
M02TXD0	B6		
M01TXEN	D12	O	Transmit enable. MXXTXEN indicates valid transmit data on MXXTXD. High = valid transmit data Low = no valid transmit data
M02TXEN	A7		

M01TXER	C14	O	Transmit error. MXXTXER allows coding errors to be propagated across the media-independent interface (MII).
			High = transmit error
			Low = no transmit error

M02TXER	D8		

10-/100-/200-Mbit/s uplink MAC interface (port 00)

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION

M00COL	B14	I	Collision sense. Assertion of M00COL indicates network collision. In full-duplex operation, M00COL can be used as a flow-control signal.
			High = network collision
			Low = no network collision

M00CRS	D15	I	Carrier sense. M00CRS indicates a frame carrier signal is being received.
			High = frame carrier
			Low = no frame carrier

M00PROTOCOL	C15	I	Interface protocol. M00PROTOCOL indicates the protocol of the interface and is determined by the logic level on the terminal. When high, it indicates a request/grant interface for datastreaming applications. When low, it indicates a CSMA/CD interface. For 200-Mbit/s uplink operation, M00PROTOCOL should be held high (refer to *uplink flow-control protocol* for details). For standard Ethernet applications, M00PROTOCOL should be held low.
M00DUPLEX	B15	I/O	Duplex selector. M00DUPLEX switches the interface between full- and half-duplex operation (low = half duplex, high = full duplex). The M00DUPLEX input has an active pulldown circuit used to force the output low when the FORCEHD bit is set.
M00LINK	A15	I	Link selector. M00LINK indicates the condition of the port connection (low = no link, high = link OK) (in 200-Mbit/s mode, M00LINK = 1).
M00RCLK	D16	I	Receive clock. M00RCLK source is from the external PHY device.
M00RXD0	C16	I	Receive data. Nibble/byte receive data is from the PMD front end. Data is synchronous to M00RCLK. Port 00 can receive 4- or 8-bit data and this is determined by strapping terminal $\overline{\text{M00UPLINK}}$: when low, the uplink operates in the wide 8-bit mode; when high, the upper-nibble bits (4–7) are not driven, leaving 4-bit data. If the MWIDTH bit in the port control register and the M00SPEED terminal are set low, then data is received on the M00RXD0 terminal.
M00RXD1	B16		
M00RXD2	A16		
M00RXD3	D17		
M00RXD4	C17		
M00RXD5	B17		
M00RXD6	A17		
M00RXD7	C18		

Terminal Functions (Continued)

10-/100-/200-Mbit/s uplink MAC interface (port 00) (continued)

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
M00RXDV	A18	I	Receive data valid. M00RXDV indicates data on M00RXD is valid for 10-/100-Mbit/s operation. When operating in 200-Mbit/s mode in conjunction with the M00RXDVX signal, M00RXDV indicates the following (in 200-Mbit/s mode): M00RXDVX (most-significant bit), M00RXDV (least-significant bit). 00 – idle (interframe gap) 01 – data frame available 10 – idle (waiting for keytag) 11 – post tag data available
M00RXDVX	D19	I	Receive data valid. In conjunction with the M00RXDV signal, M00RXDVX indicates the following: M00RXDVX (most-significant bit), M00RXDV (least-significant bit). 00 – idle (interframe gap) 01 – data frame available 10 – idle (waiting for keytag) 11 – post tag data available M00RXDVX must be pulled down with external resistors to operate in 10-/100-Mbit/s modes.

M00RXER	C19	I	Receive error. M00RXER indicates reception of a coding error on received data.
			High = coding error
			Low = no coding error

M00SPEED	B19	I	Bit-rate selection speed. The speed of the MAC interface is determined by the logic level on M00SPEED (low = 10 Mbit/s, high = 100 Mbit/s) (in 200-Mbit/s mode, M00LINK = 1).
M00TCLK	A19	I	Transmit clock. M00TCLK source is from the external PHY or test device.
M00TXD0 M00TXD1 M00TXD2 M00TXD3 M00TXD4 M00TXD5 M00TXD6 M00TXD7	C20 B20 A20 D21 C21 B21 A21 D22	O	Transmit data. Nibble/byte transmit data is from the TNETX3150/TNETX3150A. When M00TXEN is asserted, these signals carry transmit data. The source port number appears on M00TXD3–M00TXD0 one cycle before M00TXEN is asserted. Data on these terminals is always synchronous to M00TCLK. The uplink can transmit 4- or 8-bit data and is determined by strapping terminal M00UPLINK: when low, the uplink operates in the 8-bit wide mode; when high, the upper-nibble bits (4–7) are not driven, leaving 4-bit data. If the MWIDTH bit in the port control register and the M00SPEED terminal are set low, then data is transmitted on the M00TXD0 terminal.

M00TXEN	C22	O	Transmit enable. M00TXEN indicates valid transmit data on M00TXD7–M00TXD0.
			High = valid data
			Low = no valid data

M00TXER	D20	O	Transmit error. M00TXER allows coding errors to be propagated across the MIIs. When M00UPLINK is low (200-Mbit/s uplink), M00TXER is taken high when an underrun occurs in the TX FIFO port 00. This enables external logic to reconstruct and resend the frame. When M00UPLINK is high (nonuplink mode), M00TXER is asserted at the end of an under-running frame, enabling the TNETX3150/TNETX3150A to force a coding error.
M00UPLINK	A22	I	Uplink selector. M00UPLINK, when low, selects the 8-bit uplink (200 Mbit/s) protocol. When high, M00UPLINK selects normal 10-/100-Mbit/s protocol.

Terminal Functions (Continued)

network monitoring (NMON) interface (see system NMON register at 0x00A2)

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
NMON00	N3	O	Network monitor bit 00. The NMON00 function is determined by the logic state of MONRXTX and MONWIDE bits. See *system NMON register.*
NMON01	N2	O	Network monitor bit 01. The NMON01 function is determined by the logic state of MONRXTX and MONWIDE bits. See *system NMON register.*
NMON02	M4	O	Network monitor bit 02. The NMON02 function is determined by the logic state of MONRXTX and MONWIDE bits. See *system NMON register.*
NMON03	M3	O	Network monitor bit 03. The NMON03 function is determined by the logic state of MONRXTX and MONWIDE bits. See *system NMON register.*
NMON04	P3	O	Network monitor bit 04. The NMON04 function is determined by the logic state of MONRXTX and MONWIDE bits. See *system NMON register.*
NMON05	P4	O	Network monitor bit 05. The NMON05 function is determined by the logic state of MONRXTX and MONWIDE bits. See *system NMON register.*
NMON06	R1	O	Network monitor bit 06. The NMON06 function is determined by the logic state of MONRXTX and MONWIDE bits. See *system NMON register.*

DRAM interface

TERMINAL		I/O	DESCRIPTION
NAME†	NO.		
DA00 DA01 DA02 DA03 DA04 DA05 DA06 DA07	A23 A24 B23 C23 C26 D24 D25 D26	O	DRAM address bus. DA00–DA07 is time multiplexed with row and column address strobes.
DX00 DX01 DX02	E23 E24 E26	O	DRAM extended address lines. DX00–DX02 is time multiplexed with row and column address strobes.
$\overline{\text{DCAS}}$	F23	O	DRAM column address select. $\overline{\text{DCAS}}$ is the column address strobe.
$\overline{\text{DRAS}}$	F26	O	DRAM row address select. $\overline{\text{DRAS}}$ is the row address strobe

† DD00 is the least-significant bit.

Terminal Functions (Continued)

DRAM interface (continued)

TERMINAL		I/O	DESCRIPTION
NAME†	NO.		
DD00	G23	I/O	DRAM data bus. DD00–DD25 is bidirectional.
DD01	G24		
DD02	G25		
DD03	G26		
DD04	H23		
DD05	H24		
DD06	H25		
DD07	H26		
DD08	J24		
DD09	J26		
DD10	K23		
DD11	K24		
DD12	K25		
DD13	K26		
DD14	L23		
DD15	L24		
DD16	L25		
DD17	L26		
DD18	M23		
DD19	M24		
DD20	M25		
DD21	M26		
DD22	N23		
DD23	N24		
DD24	P24		
DD25	P25		

† DD00 is the least-significant bit.

Terminal Functions (Continued)

DRAM interface (continued)

TERMINAL		I/O	DESCRIPTION
NAME†	NO.		
DD26	R23	I/O	DRAM data bus. DD00–DD35 is bidirectional.
DD27	R24		
DD28	R25		
DD29	R26		
DD30	T23		
DD31	T24		
DD32	T25		
DD33	T26		
DD34	U24		
DD35	U25		
$\overline{\text{DOE}}$	F24	O	DRAM output enable. $\overline{\text{DOE}}$ is the DRAM output-enable command.
$\overline{\text{DWE}}$	F25	O	DRAM write enable. $\overline{\text{DWE}}$ is the DRAM read/write command.

† DD00 is the least-significant bit.

external address match (EAM) interface

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
EAM15 (mode select)	AD21	I	External address match. When EAM15 is high and EAM04 is low, it indicates that the least-significant nibble EAM03–EAM00 contains a routing code for the single-selected port.
EAM00	AB26	I	External address match. When EAM15 is low, a high on EAMXX (XX = 14–0) indicates the data frame is transmitted from that single-selected port. EAM00 is the least-significant bit and is associated with port 00.
EAM01	AB24		
EAM02	AB23		
EAM03	AC26		
EAM04	AC25		
EAM05	AC24		
EAM06	AD26		
EAM07	AF24		
EAM08	AD23		
EAM09	AE23		
EAM10	AF23		
EAM11	AC22		
EAM12	AD22		
EAM13	AF22		
EAM14	AC21		

Terminal Functions (Continued)

direct input/output (DIO) and statistics interface

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
SAD1 SAD0	W24 W23	I	DIO address. SAD1–SAD0 selects the TNETX3150/TNETX3150A host registers.
$\overline{\text{SCS}}$	W25	I	DIO chip select. When low, $\overline{\text{SCS}}$ indicates a port access is valid.
SDATA7 SDATA6 SDATA5 SDATA4 SDATA3 SDATA2 SDATA1 SDATA0	AA25 AA24 AA23 Y26 Y25 Y24 Y23 W26	I/O	DIO data. SDATA7–SDATA0 is bidirectional byte-wide data.
$\overline{\text{SRDY}}$	AA26	O	DIO ready. When low, $\overline{\text{SRDY}}$ indicates to the host that data is ready to be read. $\overline{\text{SRDY}}$ also indicates when data has been received for the write cycle. The TNETX3150/TNETX3150A drives $\overline{\text{SRDY}}$ high for one clock cycle before placing the output in the high-impedance (Z) state after $\overline{\text{SCS}}$ is taken high. $\overline{\text{SRDY}}$ should be pulled high with an external pullup resister.
SRNW	V26	I	DIO read not write. When low, SRNW indicates a write cycle on the DIO port.

EEPROM interface

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
ECLK	AF21	O	EEPROM data clock. ECLK is the serial EEPROM clock and requires an external pullup resistor.
EDIO	AE21	I/O	EEPROM data input/output. EDIO is the serial EEPROM data I/O signal that requires an external pullup (see EEPROM data sheet) for EEPROM operation. Tying this terminal to ground disables the EEPROM interface and prevents auto-configuration.

LED activity interface

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
LEDCLK	AC20	O	LED clock. LEDCLK is the serial-shift clock for the LED-status data.
$\overline{\text{LEDDATA}}$	AD20	O	LED data. $\overline{\text{LEDDATA}}$ is the active-low serial data signal for the LED-status data.
LEDSTR0	AE20	O	LED strobe (port status). One LEDCLK cycle after the end of port status valid data, LEDSTR0 pulses high for one LEDCLK cycle.
LEDSTR1	AF20	O	LED strobe (transmit-queue status). One LEDCLK cycle after the end of transmit-queue status valid data, LEDSTR1 pulses high for one LEDCLK cycle.

Terminal Functions (Continued)

JTAG interface

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
TCLK	AF19	I	Test clock. TCLK is used to clock state information and test data into and out of the device during operation of the test port.
TDI	AD18	I	Test data input. TDI is used to serially shift test data and test instructions into the device during operation of the test port.
TDO	AE19	O	Test data out. TDO is used to serially shift test data and test instructions out of the device during operation of the test port.
TMS	AD19	I	Test mode select. TMS is used to control the state of the test-port controller within the TNETX3150/TNETX3150A.
$\overline{\text{TRST}}$	AC19	I	Test reset. $\overline{\text{TRST}}$ is asynchronous reset of the test-port controller.

miscellaneous functions

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
DREF	U23	O	DRAM reference. DREF is the DRAM reference clock.
NC	V24		No connection.
OSCIN	AC12	I	Oscillator input. OSCIN is the TNETX3150/TNETX3150A clock input (50 MHz).
$\overline{\text{RESET}}$	U26	I	Reset. $\overline{\text{RESET}}$ (active low) resets the TNETX3150/TNETX3150A. $\overline{\text{RESET}}$ should be held low for 25 ms after the power supplies and clocks have stabilized. This input is synchronous and therefore, the system clock must be operational during reset.

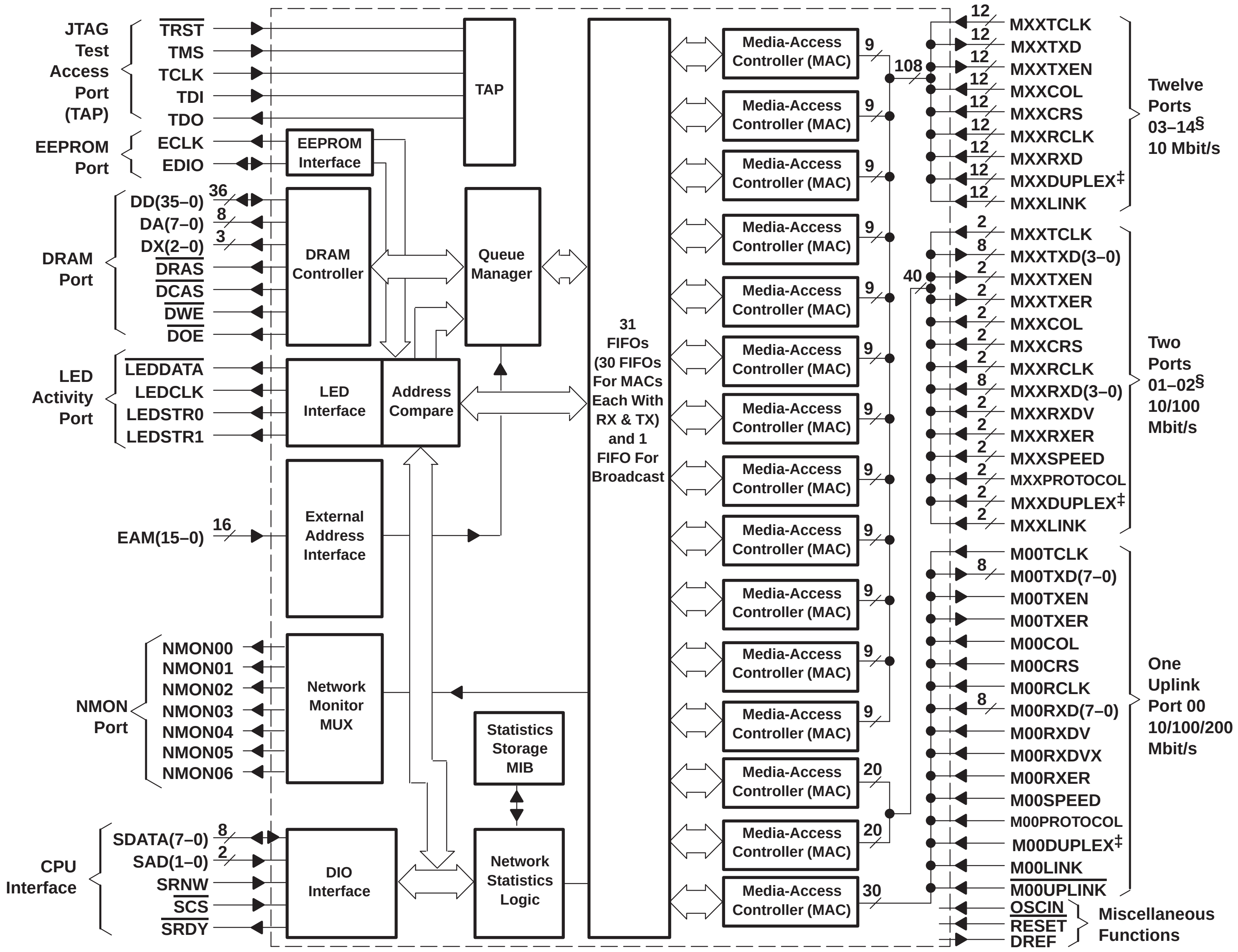
power interface

TERMINAL		DESCRIPTION
NAME	NO.	
GND	A1, A2, A13, A14, A25, A26, AD2, AD25, AE1, AE3, AE24, AE26, AF1, AF2, AF13, AF14, AF25, AF26, B1, B3, B24, B26, C2, C25, N1, N26, P1, P26	Ground. GND is the 0-V reference for the device.
V_{CC}	AC4, AC9, AC13, AC18, AC23, AD3, AD24, AE2, AE25, B2, B25, C3, C24, D4, D9, D14, D18, D23, J4, J23, N4, V4, V23, P23	Supply voltage. $V_{CC} = 3.3 \pm 0.3$ V
$V_{CC(5V)}$	AB2, AB25, AE5, AE9, AE14, AE18, AE22, B5, B9, B13, B18, B22, E2, E25, J2, J25, N25, P2, V2, V25	Supply voltage. $V_{CC(5V)} = 5 \pm 0.5$ V. $V_{CC(5V)}$ is for the I/O periphery only and is used for the 5-V tolerant pads only.

Terminal Functions (Continued)

summary of signal terminals by signal group function

PORT DESCRIPTION	NUMBER OF SIGNALS	MULTIPLIER	TOTAL
LED	4	1	4
10-Mbit/s port	9	12	108
10-/100-Mbit/s port	20	2	40
10-/100-/200-Mbit/s uplink	30	1	30
DIO	13	1	13
EAM port	16	1	16
EEPROM interface	2	1	2
DRAM interface	51	1	51
Miscellaneous	3	1	3
JTAG	5	1	5
NMON	7	1	7
Total design			279
SUMMARY			
Assigned balls			279
Unassigned balls (Reserved)			1
Power balls $V_{CC(5V)}$			20
Power balls V_{CC}			24
Power balls GND			28
Total number of balls			352



[†] See Table 1 for signal group breakdown.
[‡] The MXXDUPLEX inputs in the 15 media-access controllers are active-pulldown inputs.
[§] XX is the port number that is being monitored.

Figure 1. TNETX3150/TNETX3150A Interface Block Diagram

interface description

The TNETX3150/TNETX3150A functional interfaces are shown in Figure 1 and described in the following paragraphs.

10-/100-/200-Mbit/s MAC interfaces (ports 00–14)

The 10-/100-/200-Mbit/s interface links the MAC interfaces to the FIFO and data-handling mechanisms of TNETX3150/TNETX3150A.

The 10 Mbit/s ports (03–14) operate in serial mode only. When operating in 10-Mbit/s mode, the 10-/100-Mbit/s interfaces (00–02) operate in either nibble or serial mode; when operating in 100-Mbit/s mode, the 10-/100-Mbit/s interfaces operate in nibble mode only. In 200-Mbit/s mode, port 00 operates in byte mode only.

data reception

Figure 2 shows a simplified 10-/100-Mbit/s receiver block diagram.

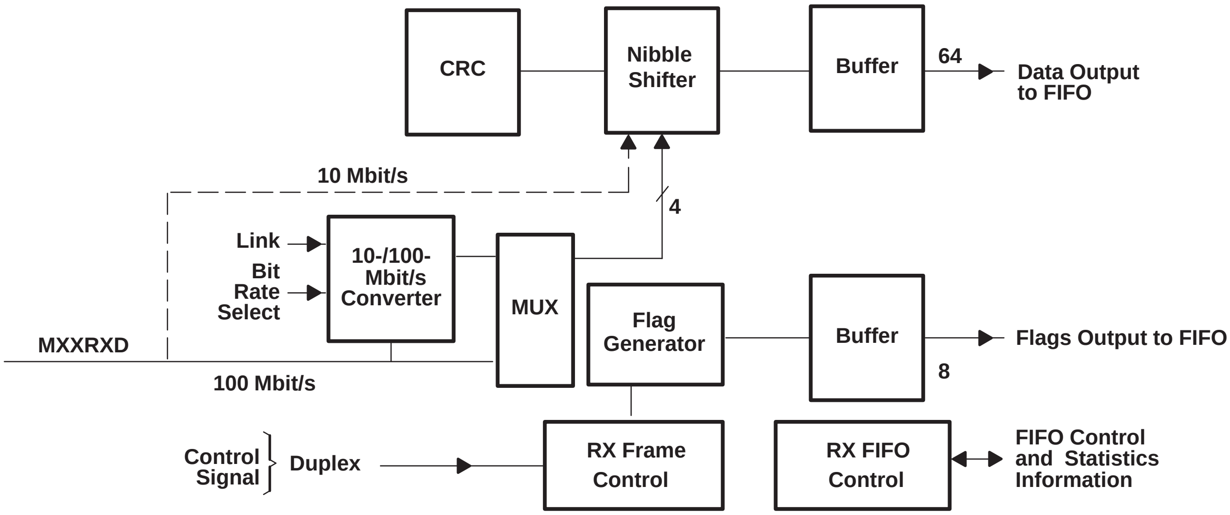


Figure 2. 10-/100-Mbit/s Receive Block Diagram

The data received from the external PHY forms 64-bit words. The data is synchronized to the internal clock of TNETX3150/TNETX3150A. After deserialization, a flag byte is assigned to the data word, identifying attributes for later data handling. The format of the flag-byte data is common for all ports. Once the 100-Mbit/s data has been deserialized, it is handled the same as the 10-Mbit/s data.

giant (long) frames

The TNETX3150/TNETX3150A can handle frames up to 1536 bytes to support IEEE Std 802.10. All frames longer than 1563 bytes are truncated by the TNETX3150/TNETX3150A. Frames greater than 1536 received in cut-through mode are passed. Frames greater than 1536 received in store-and-forward mode are discarded and not transmitted. The LONG option bit governs how the statics for long frames are recorded. The byte count for long frames with good cyclic redundancy checks (CRCs) are recorded in the RX + TX-frames 1024–1518 statistic, (which effectively becomes RX + TX-frames 1024–1531 when LONG is set). Long frames received with bad CRCs are recorded as jabbers.

short frames

All frames less than 64 bytes received into any port are filtered by the TNETX3150/TNETX3150A within the RX FIFOs. They do not appear on the DRAM bus.

RX filtering of non-cut-through frames

The TNETX3150/TNETX3150A filters RX frames (CRC, alignment, jabber, etc.) that contain errors when operating in store-and-forward mode. A frame that might be cut-through can be non-cut-through if its destination is busy. If it contains any error, it is filtered. The error is recorded in the relevant statistic counter, with all used buffers being recovered and returned to the free queue.

deserialization

The raw data is deserialized before further processing. The data is formed into 64-bit words. The received data is synchronized with the internal clock of the TNETX3150/TNETX3150A.

receive control

Receiver control is partitioned into two blocks:

1. The frame-control block schedules all receive operations (detection and removal of the preamble, extraction of the addresses and frame length, data handling, and CRC checking. Also included is a jabber-detection timer to detect greater-than-maximum-length frames being received on the network.
2. The FIFO-control block places the received data into the FIFO buffers, while also detecting and flagging erroneous data conditions in the flag byte.

TNETX3150/TNETX3150A MXXDUPLEX terminals

The MXXDUPLEX terminals are implemented as inputs with active pulldown circuitry, producing a pseudo-bidirectional terminal.

An external PHY can weakly drive the DUPLEX line high, indicating an intention for duplex operation. the TNETX3150/TNETX3150A can override this DUPLEX terminal input by pulling the line low. This is detected by the PHY, which monitors the sense of the DUPLEX signal, causing it to operate in half-duplex mode. Thus, the TNETX3150/TNETX3150A can force the PHY into half-duplex operation when desired (during testing for example). See Figure 3.

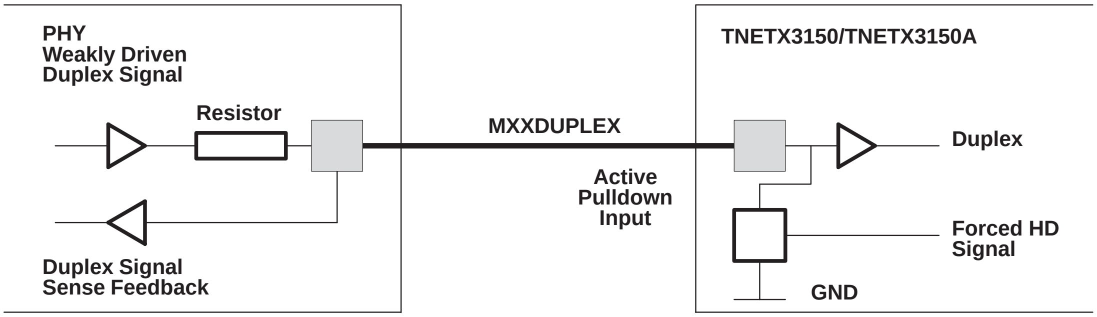


Figure 3. MXXDUPLEX Signal Connection

If the PHY is driven only in half-duplex operation, a pulldown resistor should be permanently attached to the DUPLEX signal.

If the PHY is operated in full duplex (with the option of the TNETX3150/TNETX3150A forcing half duplex), a pullup resistor should be placed on the DUPLEX signal.

receive arbitration

Receive arbitration biases the prioritization of the arbitration for received frames over transmitted frames. This utilizes the TNETX3150/TNETX3150A buffering capability during heavy traffic loading, while increasing transmission latency of the TNETX3150/TNETX3150A. Receive arbitration is selected by setting the RXARB bit (bit 5) in the serial input/output (SIO) register.

The normal arbitration scheme is extended to bias the receive priority and active transmissions over inactive transmissions. The queue manager services buffer transfer requests between the port FIFOs and DRAM. Receive requests and ongoing transmit requests take priority over transmissions that have not started (inactive transmissions). If there are spare DRAM accesses available, an inactive transmit request is promoted to an active transmit request. If there are no spare DRAM accesses, the transmit requests are arbitrated and all ongoing transmissions are allowed to finish, with no new transmission started until the receive requests are exhausted.

Port 00, when operated in uplink mode, is always assigned the transmit inactive priority. Even when granted an active TX slot, one buffer is transferred (following the initial two buffers accrued before a frame start) before the port renegotiates another TX active slot. Thus, port 00 TX in uplink mode has the lowest possible priority, reducing the probability of frame loss through oversubscribed bandwidth, while increasing frame latency and buffering requirements. When operating in this mode, external hardware must be provided to reconstruct the frame due to port 00 uplink operation.

data transmission

Figure 4 shows a simplified 10-/100-/200-Mbit/s transmit block diagram.

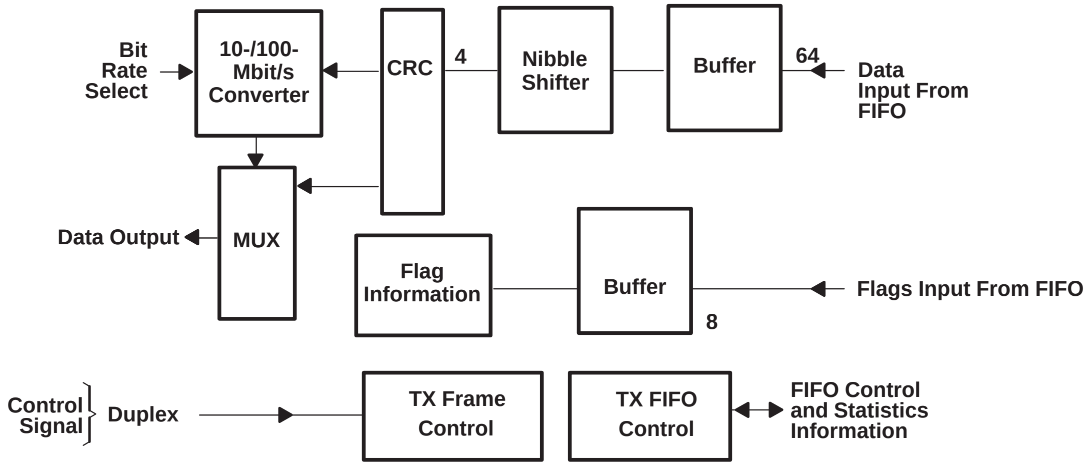


Figure 4. 10-/100-/200-Mbits Transmit Block Diagram

serialization

Data entering from the FIFO as a 64-bit word is serialized for transmission at the transmit clock rate. This also requires the data to be synchronized to the transmit clock rate from the TNETX3150/TNETX3150A internal clock. The CRC block is used only to check that the frame still has a valid CRC. It is not used to recalculate a new CRC for the frame. If the CRC does not match, the frame contents were corrupted through the switch and are counted in the TX data-error counter.

backoff

The TNETX3150/TNETX3150A implements the IEEE Std 802.3 binary exponential backoff algorithm.

transmit control

Transmit control is partitioned into two blocks:

1. The frame-control block handles the output of data into the PHY interfaces. A number of error states are handled. If a collision is detected, the state machine jams the output. If the collision is late (after the first 64-byte buffer has been transmitted), the frame is lost. If it is an early collision, the controller backs off before retrying. While operating in full duplex, both carrier sense (CRS) mode and collision sensing modes are disabled.
2. The FIFO-control block handles the flow of data from the FIFO buffers to the MAC interface for transmission. The data within a FIFO buffer is cleared only after the data has been successfully transmitted without collision (for the half-duplex ports). Transmission recovery also is handled in this state machine. If a collision is detected, frame recovery and retransmission are initiated.

interframe-gap enforcement

The measurement reference for the interframe gap of 96 µs (when transmitting at 10 Mbit/s) is changed, depending on frame traffic conditions. If a frame is successfully transmitted (without collision), 96 µs is measured from MXXTXEN. If the frame suffered a collision, 96 µs is measured from MXXCRS, the TNETX3150/TNETX3150A can receive frames with an interframe gap of less than 9.6 µs, and it always transmits its frames with an interframe gap at 9.6 µs. The 100-Mbit/s ports can receive frames with an interframe gap of less than 0.96 µs, and it always transmits its frames with an interframe gap of 0.96 µs.

transmit pacing

When transmit pacing is enabled, the ThunderSWITCH architecture is capable of altering its transmission routine during times of heavy network activity. The TNETX3150/TNETX3150A is intelligent enough to sense heavy network traffic and alter its transmission routing by intentionally inserting an extra amount of delay between transmission attempts. The added delay reduces collision rates, and thus reduces the number of transmission attempts, which helps reduce CPU utilization, lighten overall network traffic, and allows the network time to normalize before attempting transmission. If the delay was not added, the TNETX3150/TNETX3150A would attempt to transmit on an already heavily loaded network, adding to the network traffic’s unsuccessful transmission attempts.

Each Ethernet MAC incorporates transmit-pacing logic. This is enabled on an individual basis by setting the TXPACE bit (bit 1) of the port control registers. When set, the MACs use transmit pacing to enhance performance (when connected on networks using other transmit-pacing-capable MACs). Transmit pacing introduces delays into the normal transmission of frames, which delays transmission attempts between stations, reducing the probability of collisions occurring during heavy traffic (as indicated by frame deferrals and collisions). This increases the chance of successful transmission.

When a frame is deferred, suffers a single collision, multiple collisions, or excessive collisions, the pacing counter is loaded with the initial value that is loaded into the PACTST register (bits 4–0). When a frame is transmitted successfully (without experiencing a deferral, single collision, multiple collisions, or excessive collisions), the pacing counter is decremented by one, down to zero.

With pacing enabled, and after one interframe-pacing-gap (IPG) delay, a frame is permitted to immediately attempt transmission if the pacing counter is zero. If the pacing counter is zero, normal IPG rules apply. Pacing delays are not inserted when the pacing counter is zero. If the pacing counter is nonzero, the frame is delayed by the pacing delay (a delay of approximately four IPGs).

uplink port interface (port 00)

The uplink can be used as a 15th 10-/100-Mbit/s switched port, even though no address compare register exists for it. Packets are switched to it by default if the destination address is not matched to any of the other 14 ports.

The uplink port implementation is similar to the 10-/100-Mbit/s port described previously. However, the uplink port is capable of 200 Mbit/s by using byte-wide transfers rather than nibble transfers. This 200-Mbit/s wide uplink mode is selected by asserting $\overline{\text{M00UPLINK}}$ low.

uplink multiplexing operation

With $\overline{\text{M00UPLINK}}$ set low, all packets are sent to the uplink port by default. Broadcast and unicast traffic received on ports 01–14 are treated similarly (forwarded to the uplink only, if no local addressing is enabled). Identification of broadcast traffic is retained for statistical counting purposes. The address-compare disable option bits (ADRDIS) (port control register) are set for all ports except port 00. Local-address comparison is possible by clearing the ADRDIS bits for the ports that take part in address comparison. Alternatively, the EAM interface can be used in the normal manner. Setting $\overline{\text{M00UPLINK}}$ low also selects store-and-forward operation on all ports to prevent data underflows and to permit frame-error filtering. If local-frame switching is employed, clearing the relevant STFORRX bits for ports 01–14 and ensuring both STFORRX and STFORTX bits are set for port 00 (uplink), improves performance by permitting cut-through where possible (store-and-forward operation permits frame-error filtering, but cut-through does not).

uplink TX pretagging

When $\overline{\text{M00UPLINK}}$ is low, it is desirable to have an indication of which port received the frame. This permits an address lookup device to be connected to the uplink port, allowing incorporation of the TNETX3150/TNETX3150A into a larger switch fabric. The TNETX3150/TNETX3150A provides one byte of information (to identify the source port) on the MII data terminals before M00TXEN is asserted. See Figure 5.

1-Byte Tag Field Added to Frame
No Preamble Transmitted at 200 Mbit/s.

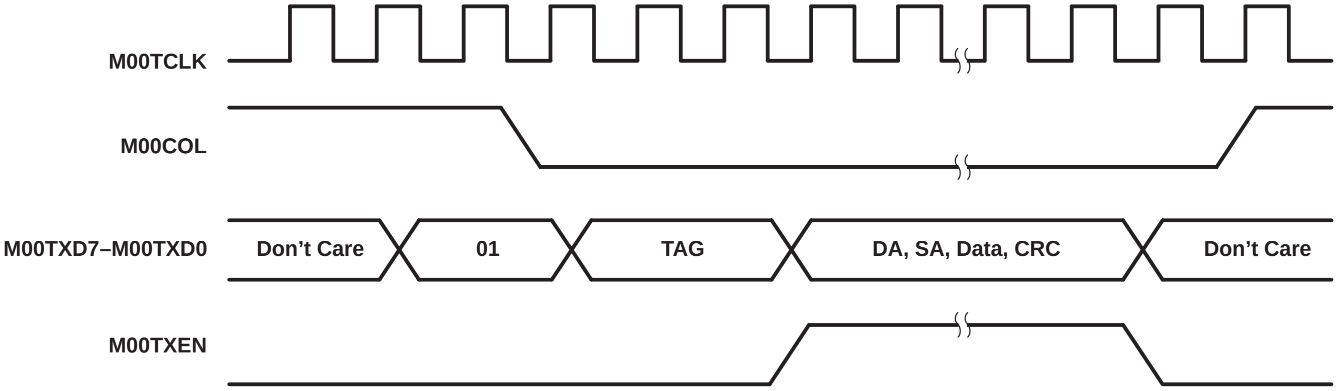


Figure 5. Multiplex Operation for Uplink-Port Transmit Frames

The pretag byte format is as follows:

BIT 7	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		Source Port Number			

The source port number is coded as shown in Table 1.

uplink TX pretagging (continued)

Table 1. Source Port Number Codes

SOURCE PORT NUMBER	PORT
0000	Reserved
0001	Port 01 (10/100 Mbit/s)
0010	Port 02 (10/100 Mbit/s)
0011	Port 03 (10 Mbit/s)
0100	Port 04 (10 Mbit/s)
0101	Port 05 (10 Mbit/s)
0110	Port 06 (10 Mbit/s)
0111	Port 07 (10 Mbit/s)
1000	Port 08 (10 Mbit/s)
1001	Port 09 (10 Mbit/s)
1010	Port 10 (10 Mbit/s)
1011	Port 11 (10 Mbit/s)
1100	Port 12 (10 Mbit/s)
1101	Port 13 (10 Mbit/s)
1110	Port 14 (10 Mbit/s)
1111	Reserved

Port 00, when operated at 100-Mbit/s ($\overline{\text{M00UPLINK}}$ = 1), provides a tag nibble one cycle before M00TXEN is asserted. A preamble is provided on this port when operated at 100 Mbit/s. The nibble format is shown in Table 1.

uplink RX pretagging

When port 00 (uplink) is operated in 200-Mbit/s mode, a pretag can be placed one cycle before MRXDV goes high for the frame. This pretag is not operated on internally by the TNETX3150/TNETX3150A. The tag is passed to the DRAM interface, where it appears on bits 31–28 of the forward pointer of the buffer transfer for the frame received on the uplink. Tag data that is applied to M00RXD0–M00RXD3 is passed, and tag data that is applied on M00RXD4–M00RXD7 is ignored. This allows an upstream device, connected to the uplink, to pass information through the TNETX3150/TNETX3150A to the DRAM bus where an external lookup device can operate on that information. It also enables two ThunderSWITCH devices, when cascaded, to pass frame-source port information from the receiving TNETX3150/TNETX3150A to the cascaded external address-lookup devices. See Figure 6.

The pretag is qualified with the channel number of the port that received the frame. Only pretags for frames whose port number is 00 are valid. For frames that are not received on the uplink (port 00), the pretag is zero. A pretag received on the uplink (port 00) is retained for the duration of the frame.

uplink RX pretagging (continued)

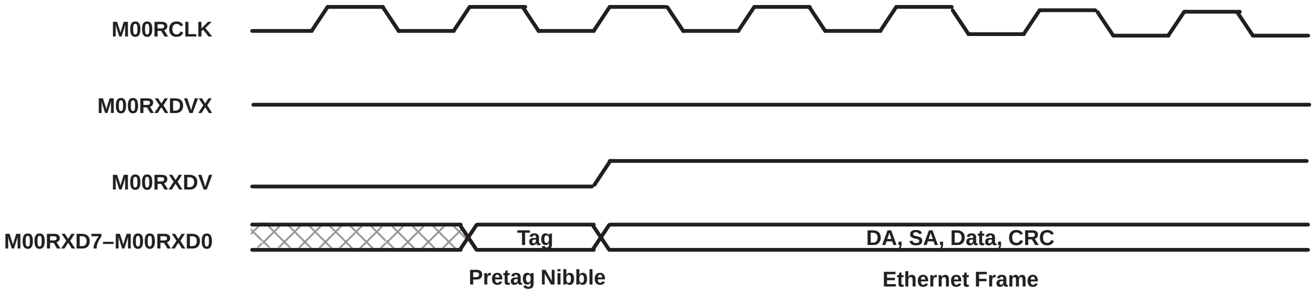


Figure 6. Data Format for Uplink RX Frames With Pretag

cascade connection

In uplink mode, the TNETX3150/TNETX3150A can route uplink frames by employing the EAM or internal address registers. To allow this, the TAGOFF bit (bit 4 of RSIZE register 0x00C2) must be set. When set, the TNETX3150/TNETX3150A does not require a post frame tag for frame-routing information. It routes the frame according to the destination address. This permits the following:

- The TNETX3150/TNETX3150A can be cascaded with a second TNETX3150/TNETX3150A device to permit larger port configurations to be supported with minimal additional hardware.
- The separate TNETX3150/TNETX3150A devices can use either single-address-per-port capability, or with an external address-lookup device, provide network-address-support capability.
- Additional logic can be used in the interface to implement more sophisticated flow control on the uplink.
- Using EEPROM devices to provide initialization data, the devices can be used with or without a management CPU, presenting a low-cost switching solution. A cascade connection is shown in Figure 7.

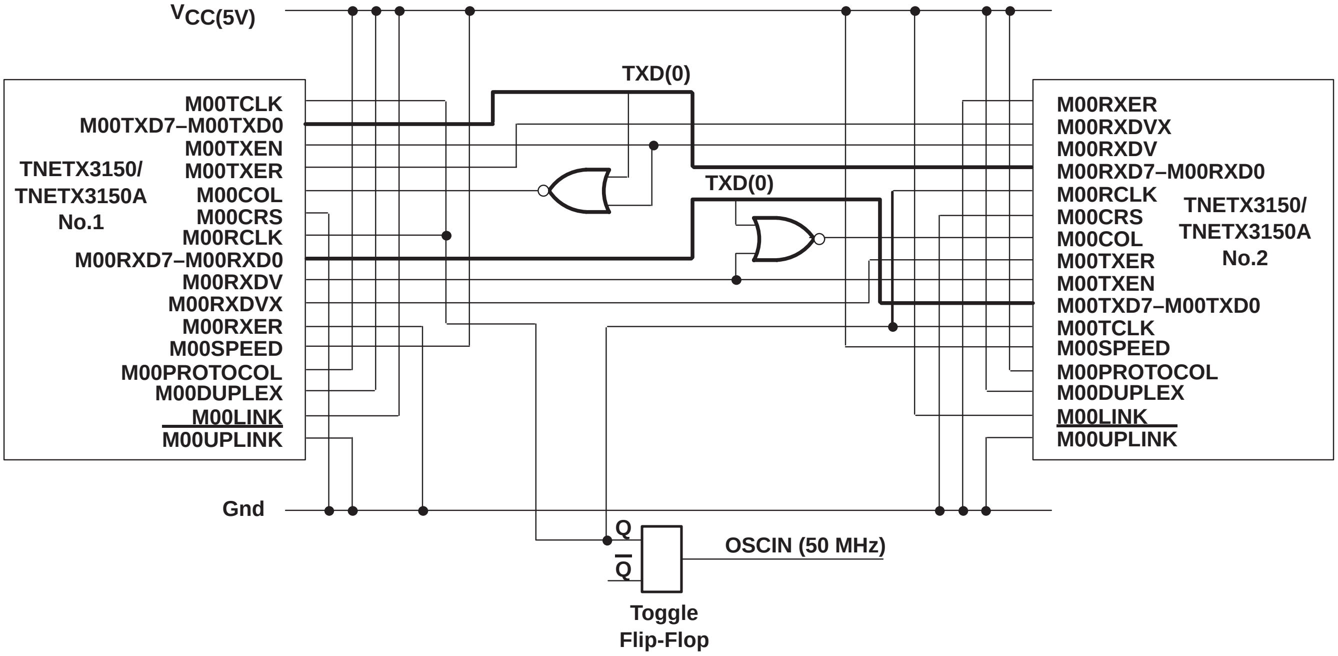


Figure 7. TNETX3150/TNETX3150A Cascade Connection

cascade connection (continued)

Cascade employs the TNETX3150/TNETX3150A uplink connection of 200 Mbit/s. The necessity to provide routing information with post tagging is disabled by option bit TAGOFF (bit 4) of DIO 0x00C2 RSIZE register.

The frame-control signal (M00TXER is asserted during FIFO underrun) is connected to M00RXDVX, permitting the receiver to temporarily stall before continuing frame reception. This allows the TNETX3150/TNETX3150A to reconstruct the frame. Flow control, implemented by holding M00COL high before frame transmission, is retained to enable flow-controlled interfaces when required.

uplink frame RX routing via post tagging

If the TNETX3150/TNETX3150A relies on the external switch logic to make switching decisions (200-Mbit/s mode and the TAGOFF option bit is reset), the external hardware must provide an indication of the destination ports for the frame received on the uplink. This indication consists of four bytes. The EAM interface and the post tag should not be used at the same time. When post tagging is enabled, the tag bytes overwrite the value received on the EAM interface (EAM interface has no effect). When post tagging is turned off (designed for local switching), the post tag is ignored, and the EAM interface or the internal address match supplies the routing information.

There is no handshake or flow control for the receive uplink path on the TNETX3150/TNETX3150A. If required, this must be implemented in upstream devices. No preamble is expected on data received by the uplink at 200-Mbit/s (see Figure 8). The tag fields are coded as shown in Figure 9.

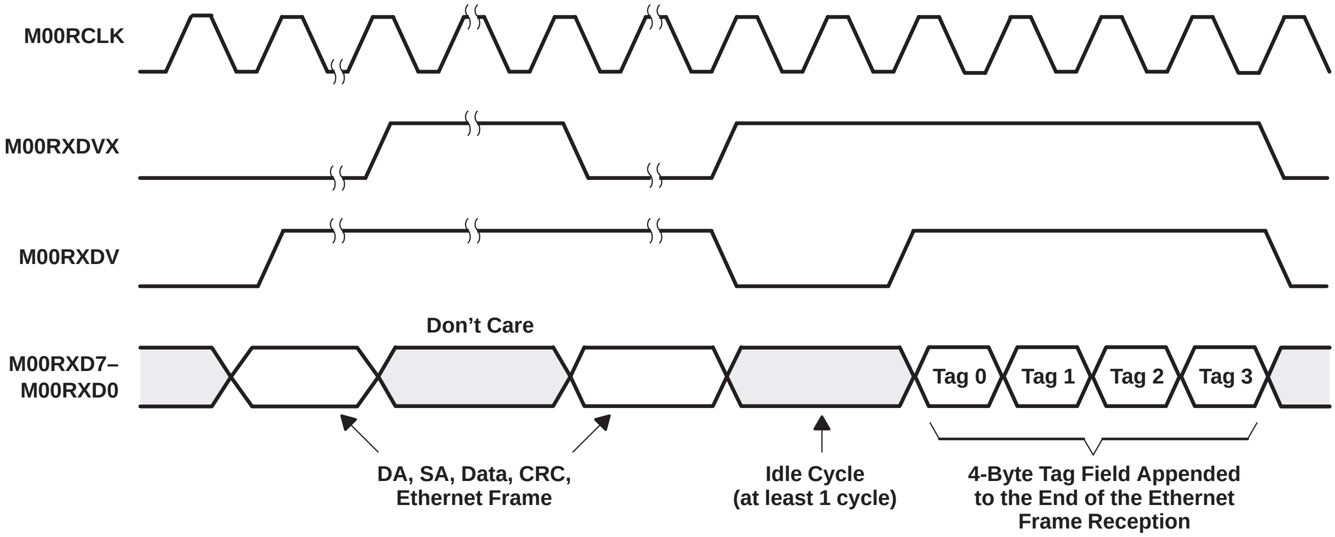


Figure 8. Data Format for Uplink RX Frames

uplink frame RX routing via post tagging (continued)

tag 0

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Destination Port 08 10 Mbit/s	Destination Port 07 10 Mbit/s	Destination Port 06 10 Mbit/s	Destination Port 05 10 Mbit/s	Destination Port 04 10 Mbit/s	Destination Port 03 10 Mbit/s	Destination Port 02 10/100 Mbit/s	Destination Port 01 10/100 Mbit/s

tag 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Reserved		Destination Port 14 10 Mbit/s	Destination Port 13 10 Mbit/s	Destination Port 12 10 Mbit/s	Destination Port 11 10 Mbit/s	Destination Port 10 10 Mbit/s	Destination Port 09 10 Mbit/s

tag 2

BIT 7	BIT 0
Reserved	

tag 3

BIT 7	BIT 0
Reserved	

Figure 9. Tag-Field Coding

If only one bit is set in the destination port field, the packet is sent to a single port. For example, with tag 0 = 00000000 and tag 1 = xx000100, the packet is unicast and destined for port 11.

If more than one bit is set, the packet is sent to multiple ports. For example, with tag 0 = 11001010 and tag 1 = xx001001, the packet is transmitted from ports 12, 09, 08, 07, 04, and 02. This allows broadcast and multicast traffic to be limited in supporting external VLANs.

If all bits are clear in the tags, the packet is invalid and is discarded.

NOTE:

The four tag fields do not immediately follow the frame data, but are presented after the end of data (following an idle period), and require that M00RXDVX = 1 and M00RXDV = 1.

uplink flow control

Flow control is available only on the uplink port and is applicable in full-duplex mode only. In this mode, asserting the collision signal before the TNETX3150/TNETX3150A begins the transmission of a frame forces the TNETX3150/TNETX3150A to wait for the collision signal to be deasserted before the frame is transmitted. The collision pin is sampled immediately before transmission. If it is not asserted, frame transmission continues. If the collision signal is asserted after transmission, the current frame continues to be transmitted. The TNETX3150/TNETX3150A delays all future-frame transmissions until the collision signal is deasserted. The interfacing hardware must be capable of storing up to a maximum-length Ethernet frame to prevent dropping frames due to congestion.

The frame is transmitted immediately following the deassertion of the collision signal. The flow-control-requesting device must be ready to accept data when the collision signal is deasserted following a flow-controlled frame. No interframe gap is imposed by the TNETX3150/TNETX3150A in this mode of operation. This provides maximum flexibility and control to the interfacing hardware on the uplink.

uplink-flow control protocol

The 200-Mbit/s handshake protocol is as follows:

- The upstream device holds flow-control signal (M00COL) high, preventing the TNETX3150/TNETX3150A from transmitting frames on the uplink.
- When a frame is ready to transmit, the TNETX3150/TNETX3150A makes a request to the upstream device by taking M00TXD00 high.
- When ready to receive, the upstream device, in response to seeing M00TXD00 go high, takes M00COL low.
- The TNETX3150/TNETX3150A places the source port number on bits M00TXD00–M00TXD03.
- Four M00TCLK clock cycles after M00COL is driven low, M00TXEN is taken high and normal data transfer occurs, starting with the destination address. No preamble is provided before the destination address in the frame.
- When M00TXEN is taken low at the end of the frame, M00COL is taken high in preparation for the next handshake. If the upstream device is busy, M00COL should be kept high even after M00TXD00 is taken high, until the upstream congestion has cleared and transmission can continue. The next frame transmission does not proceed until the handshake is performed. M00COL must be cycled before each transmission.
- To operate in this mode $\overline{\text{M00UPLINK}}$ is held low, M00DUPLEX and M00PROTOCOL are held high, and the in-order broadcast (IOB) option bit in the SYSCTRL register is set.

uplink frame control

If the uplink transmit FIFO underruns, a frame control-signal is provided on M00TXER during 200-Mbit/s uplink operation to permit the reconstruction of frames using external logic (see Figure 10).

In uplink mode, M00TXER is low throughout a successful transmission. If a FIFO underrun occurs (due to exceeding maximum memory bandwidth), the data in the FIFO continues to be transmitted until empty, at which point the M00TXER signal is taken high. While M00TXER is high, the data transmitted from the uplink should be discarded. When the next 64-byte data buffer is forwarded to the uplink transmit port, M00TXER is taken low and normal transmission continues. If the following buffer updates are delayed, the FIFO again underruns, causing M00TXER to go high once the data present in the FIFO is transmitted.

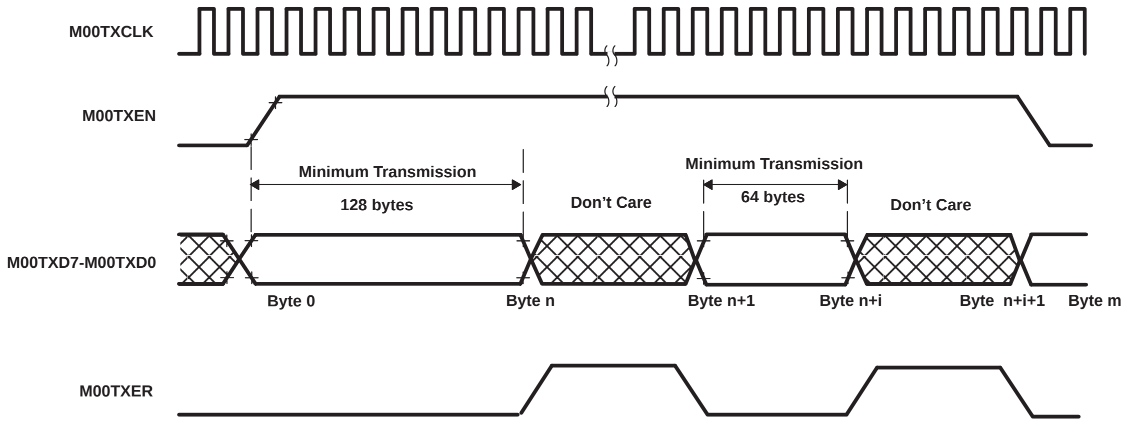


Figure 10. Uplink Frame Control for External Frame Reconstruction

uplink frame control (continued)

The FIFO is loaded with two buffers before transmission begins; this ensures a minimum transmission of 128 bytes before potential underrun can occur. Following an underrun, only one buffer is transferred with a minimum of 64 bytes following an underrun. During transmission of a long frame (during high-traffic loads), multiple underruns can occur.

If M00RXDVX is taken high during the data receive state of an uplink-frame reception, the data reception is stalled until M00RXDVX is returned low. This directly connects M00TXER to M00RXDVX. This simplifies the cascade interface, permitting the transmitting TNETX3150/TNETX3150A to apply frame control (due to underunning) and the receiving TNETX3150/TNETX3150A to reconstruct the received frame before transmission.

NMON interface

The NMON interface (see Figure 11) provides complete network monitoring capability at 10 Mbit/s and a partial capability at 100 Mbit/s for the 10/100-Mbits ports. TNETX3150/TNETX3150A port selection is based on the NMON register (0xA2). See Table 2.

The interface permits two modes of operation:

- 10-Mbit/s mode (seven-wire SNI): ports 00, 01, and 02 must be used in 10-Mbit/s SNI bit-serial mode of operation. The signals that are provided by the interface are 10-Mbit/s bit-serial, MXXRXD, MXXRCLK, MXXCRS, MXXCOL, MXXTXD, MXXTCLK, and MXXTXEN.
- 100-Mbit/s mode (four-bit, MII either RX or TX): if ports 00, 01, and 02 are operated in 100-Mbit/s mode (or 10-Mbit/s non-SNI), the user can select the interface to access RX or TX. If ports 03–14 are monitored in this mode and enabled by setting the MONWIDE bit high, only the least-significant bit of the interface contains network data (bits 1–3 are not driven). When monitoring RX data, MXXRXD3–MXXRXD0, MXXRXDV, MXXRCLK, and MXXSPEED are provided. When monitoring TX data, MXXTXD3–MXXTXD0, MXXTXEN, MXXTCLK, and MXXSPEED are provided.

This interface monitors the signals before any MAC processing is performed by the TNETX3150/TNETX3150A. An RMON probe can monitor every packet on the segment connected to the port. The port selection is made by writing to the NMON port select field.

NMON interface (continued)

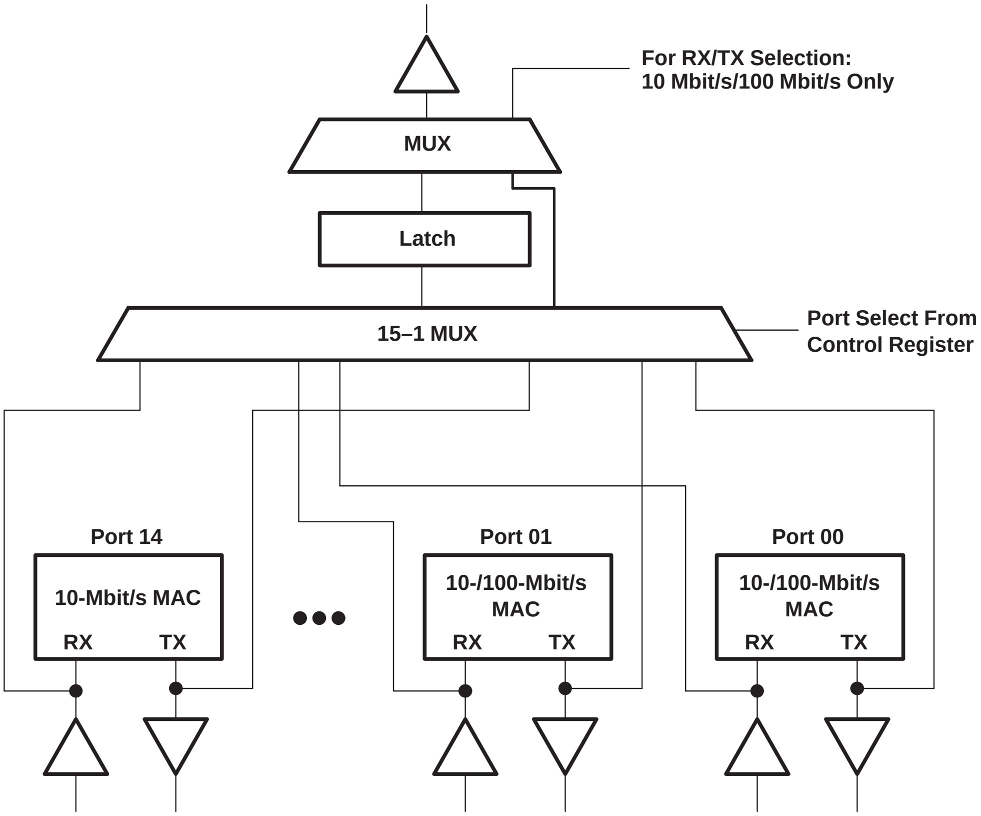


Figure 11. NMON Port Block Diagram

Table 2. NMON Code Port Selection

NMON CODE	MONITORING PORT
	UPLINK 200-Mbit/s SIGNALS PORT NUMBER
0000	00 (10/100 Mbit/s)
0001	01 (10/100 Mbit/s)
0010	02 (10/100 Mbit/s)
0011	03 (10 Mbit/s)
0100	04 (10 Mbit/s)
0101	05 (10 Mbit/s)
0110	06 (10 Mbit/s)
0111	07 (10 Mbit/s)
1000	08 (10 Mbit/s)
1001	09 (10 Mbit/s)
1010	10 (10 Mbit/s)
1011	11 (10 Mbit/s)
1100	12 (10 Mbit/s)
1101	13 (10 Mbit/s)
1110	14 (10 Mbit/s)
1111	Disable NMON

NMON interface (continued)

The NMON control field is mapped to the lower four bits of the system NMON register DIO register address 0xA2.

For 10-Mbit/s monitoring, the NMON signals are provided as shown in Table 3 (NMON register option bits: MONRXTX = X, MONWIDE = 0).

Table 3. 10-Mbit/s NMON Signals

TERMINAL NAME	NMON MODE (UPLINK)
NMON00	MXXRXD
NMON01	MXXCRS
NMON02	MXXRCLK
NMON03	MXXTXD
NMON04	MXXTXEN
NMON05	MXXTCLK
NMON06	MXXCOL

For 100-Mbit/s monitoring, network-monitoring signals are provided for TX as shown in Table 4 (NMON register operation bits: MONRXTX = 1, MONWIDE = 1).

Table 4. 100-Mbit/s TX NMON Signals

NORMAL OPERATION TERMINAL DESCRIPTION	NMON MODE (UPLINK)
NMON00	MXXTXD(0)
NMON01	MXXTXD(1)
NMON02	MXXTXD(2)
NMON03	MXXTXD(3)
NMON04	MXXTXEN
NMON05	MXXTCLK
NMON06	MXXSPEED

For 100-Mbit/s monitoring, network-monitoring signals are provided for RX as shown in Table 5 (NMON register operation bits: MONRXTX = 0, MONWIDE = 1).

Table 5. 100-Mbit/s RX NMON Signals

NORMAL OPERATION TERMINAL DESCRIPTION	NMON MODE (UPLINK)
NMON00	MXXRXD(0)
NMON01	MXXRXD(1)
NMON02	MXXRXD(2)
NMON03	MXXRXD(3)
NMON04	MXXRXDV
NMON05	MXXRCLK
NMON06	MXXSPEED

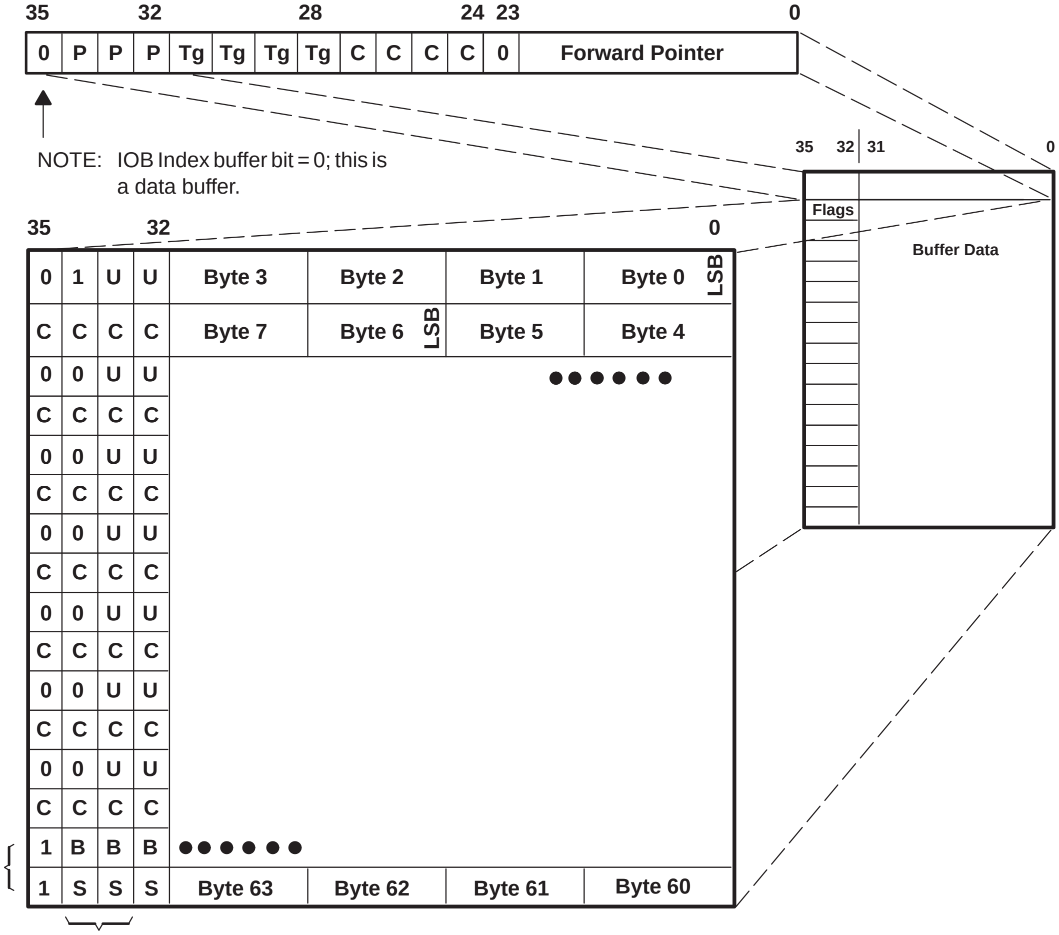
RX signals are latched and provided after a delay of one RX clock cycle. TX signals are the same as the TX terminals (no latching) after a delay time.

DRAM and EAM interfaces

All valid frames are passed across the DRAM interface (see Figure 14). The EAM hardware can detect the start of a new frame from the flag byte information. The first-flag nibble on the DRAM bus (bits 35–32) corresponds to bits 7–4 of the frame flag (see Figure 12). In conjunction with the DRAM column address strobe, external logic can access the frame addresses and perform external address lookup, as detailed in the following:

- Use row-address strobe ($\overline{\text{DRAS}}$) and column-address strobe ($\overline{\text{DCAS}}$) to identify the position of the forward pointer, the top nibble of the flag byte, and whether the nibble contains the start of frame code 01XX. Bit 35 of the forward pointer should be zero to denote a start of frame. If it is high, the frame is an IOB link buffer and not the start-of-data frame (bits 34, 33, and 32 contain parity information for the three forward point or data bytes).
- Bits 31–28 of the forward pointer contain the pre-tag nibble applied to the uplink (when used in uplink mode). See description of Port 00 uplink port.
- Bits 27–24 denote the active port number (port 00 = 0000, port 01 = 0001, etc.).
- Use the DRAM column-address strobe ($\overline{\text{DCAS}}$) to identify the presence of destination and source address data on the DRAM interface.
- Perform address processing.
- Present the destination-channel bit map not more than 12 memory cycles after the high nibble of the start flag is transmitted on the DRAM interface.
- External address timing is shown in Figure 41.

DRAM and EAM interfaces (continued)


35 32 28 24 23 0
0 P P P Tg Tg Tg Tg C C C C 0 Forward Pointer
35 32 31 0
Flags
Buffer Data
35 32 0
0 1 U U Byte 3 Byte 2 Byte 1 Byte 0 LSB
C C C C Byte 7 Byte 6 LSB Byte 5 Byte 4
0 0 U U
C C C C
0 0 U U
C C C C
0 0 U U
C C C C
0 0 U U
C C C C
0 0 U U
C C C C
0 0 U U
C C C C
1 B B B
1 S S S Byte 63 Byte 62 Byte 61 Byte 60

NOTE: IOB Index buffer bit = 0; this is a data buffer.

NOTE: Diagram shows end-of-frame flag format. Multiple buffer frames use this flag format only on the last buffer. (End-of buffer flags are used between a frame’s buffers.)

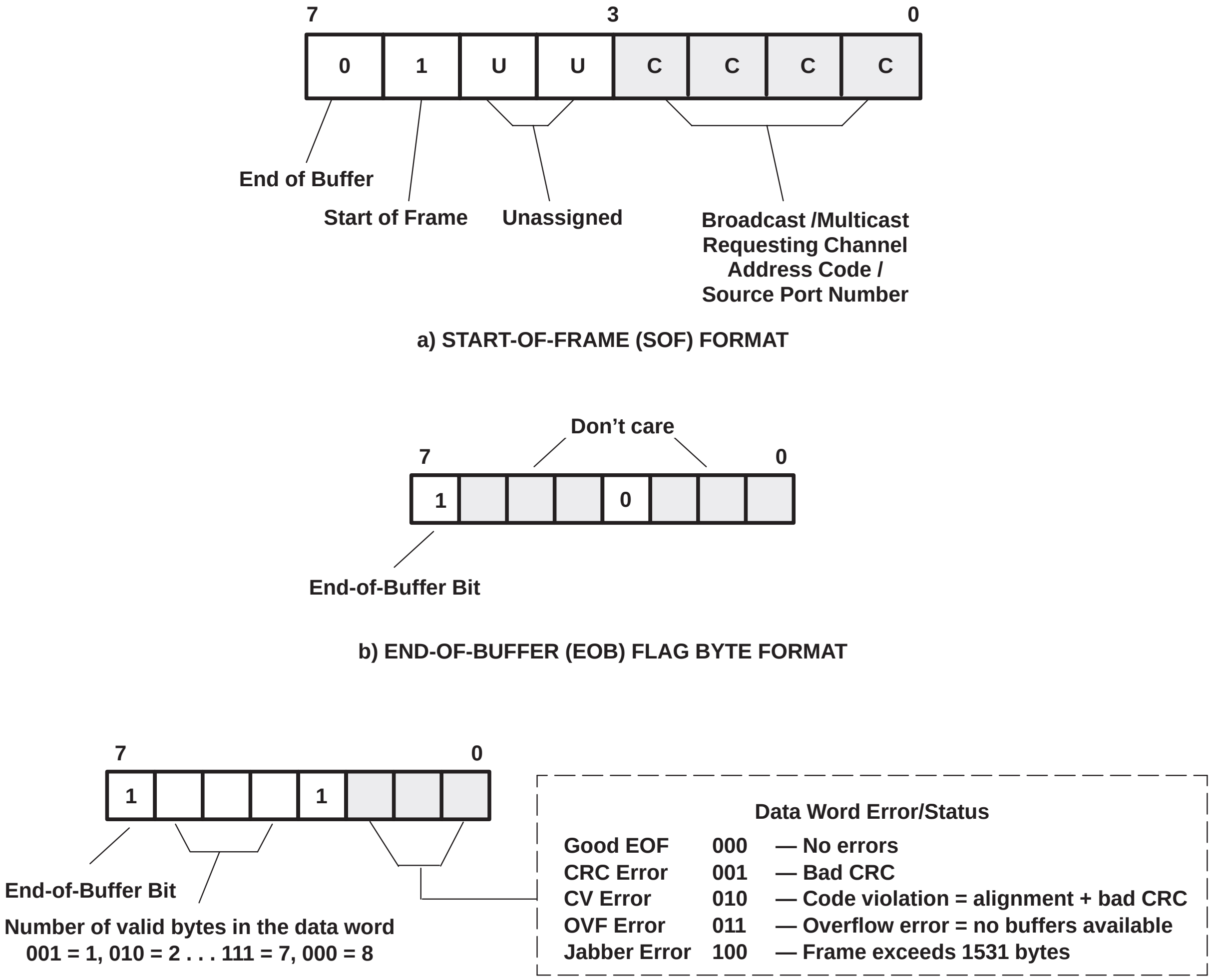
C = Channel code/source port code
P = Parity bits for forward pointer
B = Number of valid bytes in data word
S = Frame status (error conditions)
Tg = Pretag (uplink mode)
U = Reserved

Figure 12. Buffer Diagram Showing Flag Format for a 64-Byte Frame

DRAM and EAM interfaces (continued)

flag bytes

Flag attributes are assigned to the deserialized data word, identifying key attributes. The flags are used in later data handling. The flag field is assigned to every eight data bytes. The format of the subfields within the flag byte changes, depending on the flag information. The formats shown in Figure 13 are the end-of-buffer and the end-of-frame flag formats. When the most-significant bit (end-of-buffer bit) is set, the remaining bits of the most-significant nibble contain the number of bytes in the data word, while the least-significant nibble contains error/status information.



a) START-OF-FRAME (SOF) FORMAT

b) END-OF-BUFFER (EOB) FLAG BYTE FORMAT

Data Word Error/Status

Good EOF	000	— No errors
CRC Error	001	— Bad CRC
CV Error	010	— Code violation = alignment + bad CRC
OVF Error	011	— Overflow error = no buffers available
Jabber Error	100	— Frame exceeds 1531 bytes

c) END-OF-FRAME (EOF) FLAG BYTE FORMAT

NOTE A: The EOB bit is asserted after each 64-byte data transfer. Only at the EOF is bit 3 of the flag byte set.

Figure 13. EOB, EOF, and SOF Flag Formats

DRAM and EAM interfaces (continued)

The TNETX3150/TNETX3150A uses the external channel address, in priority over the internal channel address match information, to route the frame to the appropriate channel. To disable the EAM interface, the no-operation code should be used. If there is no EAM hardware present, the no-operation code should be hardwired onto the interface. The no-operation code causes the internal destination selection to be used. See Figure 14.

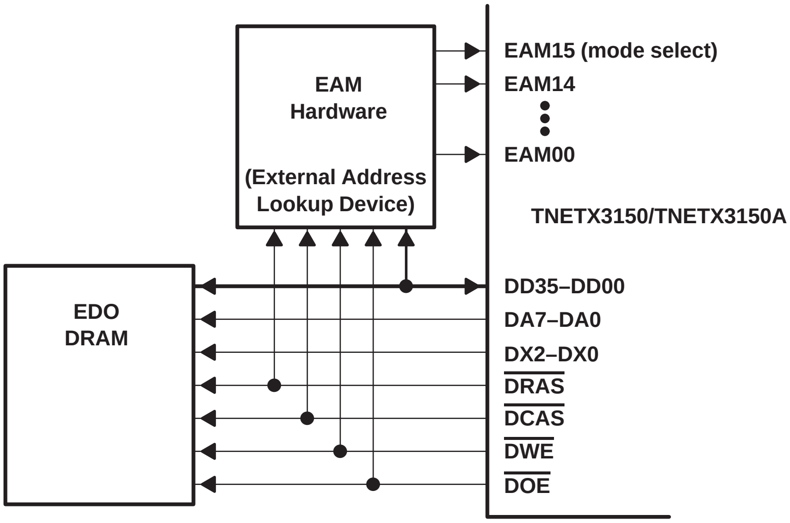


Figure 14. EAM Interface

Table 6 provides the 4-bit code needed to identify the destination port when using the EAM interface with EAM15 (mode select) bit set.

- With the EAM04 bit and EAM15 bit (mode select) both set, all other EAM bits are ignored (this is a no-operation code) and the frame uses the internal address match information (if enabled). With this code, any external device is signaling that it does not participate in address matching (at least during this cycle). These two bits need to be hardwired to 1 if no external device is present to disable this interface.
- With the EAM04 bit reset and EAM15 bit (mode select) set, EAM03–EAM00 code is used to identify a single destination port.

When the EAM15 bit (mode select) is reset:

- The other bits EAM14–EAM00 represent a mask of the ports to which the frame is forwarded. For example, if the frame is copied to ports 00, 07, and 14, the value signaled is 0x4081 or 0b0100000010000001.
- To discard a frame, the external interface provides an all-zero code (0x0000 or 0b0000000000000000), which indicates that the external device is going to specify all the ports to forward the frame to, and there are no valid 1s for this frame.

All internal address registers should be disabled with the address-disable bit (port-control register bit 3), if the external device is used.

An external device has the choice of two codes to forward a frame to a single port. To forward to port 3, one can enter 0x0004 (multiport capable with only one port specified), or 0x8003 (single-port capable only). While using the first method can make specifying unicast and multicast forwarding more unified, it forces the TNETX3150/TNETX3150A to use internal cycles to build an IOB table with only one entry, and thereby unicast traffic takes a large drop in performance. Unicast forwarding is done with the EAM15 set, and the port specified in the lower four bits.

DRAM and EAM interfaces (continued)

Table 6. EAM Port Codes

TNETX3150/TNETX3150A PORT	EAM15 MODE SELECT	EAM14–EAM05	EAM04	EAM03–EAM00
Port 00 (uplink)	1	xxxxxxxxxx	0	0000
Port 01 (10/100 Mbit/s)	1	xxxxxxxxxx	0	0001
Port 02 (10/100 Mbit/s)	1	xxxxxxxxxx	0	0010
Port 03 (10 Mbit/s)	1	xxxxxxxxxx	0	0011
Port 04 (10 Mbit/s)	1	xxxxxxxxxx	0	0100
Port 05 (10 Mbit/s)	1	xxxxxxxxxx	0	0101
Port 06 (10 Mbit/s)	1	xxxxxxxxxx	0	0110
Port 07 (10 Mbit/s)	1	xxxxxxxxxx	0	0111
Port 08 (10 Mbit/s)	1	xxxxxxxxxx	0	1000
Port 09 (10 Mbit/s)	1	xxxxxxxxxx	0	1001
Port 10 (10 Mbit/s)	1	xxxxxxxxxx	0	1010
Port 11 (10 Mbit/s)	1	xxxxxxxxxx	0	1011
Port 12 (10 Mbit/s)	1	xxxxxxxxxx	0	1100
Port 13 (10 Mbit/s)	1	xxxxxxxxxx	0	1101
Port 14 (10 Mbit/s)	1	xxxxxxxxxx	0	1110
Broadcast channel	1	xxxxxxxxxx	0	1111
No operation (discard)	1	xxxxxxxxxx	1	xxxx
Frame discard	0	0000000000	0	0000
Bit-map mode	0	EAM14–EAM0 equals port destination bit map		

VLAN support

For the single-address-per-port mode, TNETX3150/TNETX3150A provides a VLAN register per port. Each register contains a bit map to indicate the VLAN group for the port. All broadcast/multicast traffic received on that port is sent only to the ports that are a part of the same VLAN.

The EAM14–EAM00 inputs, when EAM15 (mode select) is low, provide a mechanism for the EAM interface to specify which destination port or group of destination ports must be used to transmit the frame. Each signal represents one destination port. Asserting just one signal sends the frame to one destination port and asserting more than one signal allows the same frame to be transmitted to multiple ports. This allows the EAM interface to limit the broadcast/multicast traffic within the VLAN. This mode of operation employs the IOB mechanism to append the frames onto the transmit queues of the ports used for frame transmission. The IOB mechanism is an inefficient way to send frames to single ports when individual port codes can be used.

DIO interface

The DIO interface allows host access to the TNETX3150/TNETX3150A. The DIO interface provides access to the on-chip registers and statistics.

Information that DIO provides for access follows:

- Network statistics counters – provides access to the network statistics information compiled in the statistics RAM
- System configuration registers – to set or change the operation of the TNETX3150/TNETX3150A
- RAM access – to permit test access, allowing functional testing
- Port registers – to access port control, port status, and port address registers, permitting port management and status interrogation

A byte-wide asynchronous interface is defined to reduce design overheads and to simplify interfacing logic.

Access to the internal TNETX3150/TNETX3150A registers is available, indirectly, via the TNETX3150/TNETX3150A host registers (see *host registers* for more detail).

The four host registers are addressed directly from DIO interface address lines SAD1 and SAD0 (see Table 7).

Table 7. SAD1/SAD0 Address Lines

SAD1	SAD0	DESCRIPTION
0	0	DIO address low
0	1	DIO address high
1	0	DIO data
1	1	DIO data increment

Data can be read from or written to the address registers using the data lines SDATA7–SDATA0 under the control of chip select ($\overline{\text{SCS}}$), read not write (SRNW), and ready ($\overline{\text{SRDY}}$) signals. The DIO interface is shown in Figure 15. The DIO interface timing is shown in Figures 39 and 40.

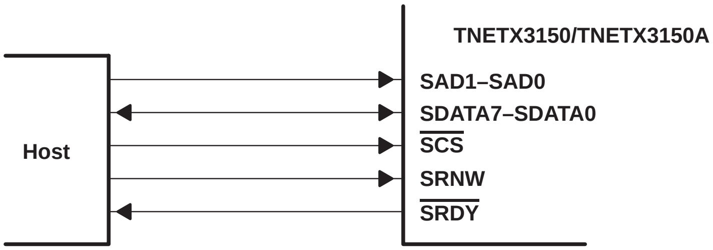


Figure 15. DIO Interface Signals

remote hardware reset

A hardware reset can be initiated by the DIO interface, permitting remote recovery of the TNETX3150/TNETX3150A, without requiring external glue logic to decode a DIO address and toggle the external reset signal of the TNETX3150/TNETX3150A. A hard DIO reset is instigated by writing 0x40000 to the DIO address high register (SAD1=0, SAD0=0). This address is directly decoded by the TNETX3150/TNETX3150A and forces a hardware reset.

port statistics compilation

The port statistics are updated, depending on the frequency of updates required to maintain a constant bandwidth to the statistics RAM. This ensures that a recordable event is not ignored.

The memory map for one port of the statistics RAM is shown in the section port statistics map.

DIO access to counter statistics

When accessing the statistics values from the DIO port, it is necessary to perform four 1-byte DIO reads, to obtain the full 32-bit counter. To prevent the chance of the counter being updated while reading the four bytes, the user should access the low byte first, followed by the upper three bytes. On reading the low byte, the counter statistic value is transferred to a 32-bit holding register before being placed on the DIO bus. The register is updated only when reading the low byte of the counter statistic. The user does not see spurious data due to an update occurring during the read.

test access to statistics memory

Test access to the statistics RAM is provided via the DIO port after the TNETX3150/TNETX3150A has been soft reset (or following power on before the START bit has been set). In this mode, all locations of the RAM can be written to and read from. Once the start bit has been set, only read access is permitted to the RAM.

clearing statistics counter memory

The statistics RAM can be requested to clear at any time during operation. This is achieved by setting the CLRSTS bit in the system control register (DIO 0xC3). This bit is latched. When set, the next statistics update cycle writes zero to all counters in the statistics RAM before resetting the latched bit. If the CLRSTS bit has not been reset by the user, the latched bit is set again, causing the TNETX3150/TNETX3150A to load zero into the statistics counters again. This continues until the CLRSTS bit is reset by the user. Soft reset has no effect on the statistics counters (their contents are not cleared during a soft reset). A hard reset causes the statistics counters to reset to zero. Port statistics cannot be cleared on a per-port basis.

FIFO RAM access

FIFO RAM access for test is provided via the DIO interface. This allows full RAM access for RAM purposes. User access to the FIFO is allowed only following a soft reset, but before the start bit is written (or after power up, but before the start bit is written). The soft reset bit should be asserted then deasserted. If the soft reset bit is not cleared, the TNETX3150/TNETX3150A holds the DRAM refresh state machine in reset and the contents of the external memory become invalid.

EEPROM read/write

EEPROM terminals ECLK and EDIO are a direct reflection of the ECLOK and EDATA bits (SIO/XCTL register) respectively. See *EEPROM interface* for additional information.

DRAM read/write

DRAM is accessed by properly configuring the DRAM test access resistors and setting the MTEST bit at 00C2.

TNETX3150/TNETX3150A condensed DIO memory map

Table 8 illustrates the condensed DIO memory map. Each section is expanded in detail.

TNETX3150/TNETX3150A condensed DIO memory map (continued)

Table 8. TNETX3150/TNETX3150A Condensed DIO Memory Map

DIO ADDRESS (BIT 15–0) SHOWN IN HEX FORMAT	DESCRIPTION	AREA OF TNETX3150/ TNETX3150A
0000–0077	Port 00–14 control/status and address registers†	Configuration
0080–009D	Port 00–14 transmit queue length†	Configuration
009E–009F	Reserved†	Configuration
00A0	Revision register†	Configuration
00A1	SIO/SCTRL register†	Configuration
00A2	NMON register†	Configuration
00A3	Reserved†	Configuration
00A4–00C1	Port 00–14 VLAN registers†	Configuration
00C2	RAM size register†	Configuration
00C3	System control register†	Configuration
00C4–00D3	Reserved	Configuration
00D4–00D7	DRAM data	Internal test
00D8	DRAM flag	Internal test
00D9–00DB	DRAM address	Internal test
00DC	Reserved	Internal test
00DD	DIATST	Internal test
00DE–3FFF	Reserved	Internal test
4000–7FFF	Hardware reset if written	Internal test
8000–877F	Port 00–14 statistics	Statistics RAM
8780–87F7	Port 00–14 collisions and RX overruns	Statistics RAM
87F8–87FF	Reserved	Statistics RAM
8800–8877	Port 00–14 TXQ registers	Statistics RAM
8878–887F	Reserved	Statistics RAM
8880–88F7	Port 00–14 IMQ registers	Statistics RAM
88F8–88FF	Reserved	Statistics RAM
8900–8977	Port 00–14 RXQ registers	Statistics RAM
8978–9FFF	Reserved	Statistics RAM
A000–A1FF	Port 13 RX FIFO block 0–3	FIFO RAM block 3
A200–A3FF	Port 13 TX FIFO block 0–3	FIFO RAM block 3
A400–A5FF	Port 14 RX FIFO block 0–3	FIFO RAM block 3
A600–A7FF	Port 14 TX FIFO block 0–3	FIFO RAM block 3
A800–CBFF	Reserved	FIFO RAM block 3
CC00-CDFF	Port 00 RX FIFO block 0–3	FIFO RAM block 1
CE00–CFFF	Port 00 TX FIFO block 0–3	FIFO RAM block 1

† These registers are automatically loaded from the EEPROM.

TNETX3150/TNETX3150A condensed DIO memory map (continued)

Table 8. TNETX3150/TNETX3150A Condensed DIO Memory Map (Continued)

DIO ADDRESS (BIT 15–0) SHOWN IN HEX FORMAT	DESCRIPTION	AREA OF TNETX3150/ TNETX3150A
D000–DIFF	Port 01 RX FIFO Block 0–3	FIFO RAM block 1
D200–D3FF	Port 01 TX FIFO block 0–3	FIFO RAM block 1
D400–D5FF	Port 02 RX FIFO block 0–3	FIFO RAM block 1
D600–D7FF	Port 02 TX FIFO block 0–3	FIFO RAM block 1
D800–D9FF	Port 03 RX FIFO block 0–3	FIFO RAM block 1
DA00–DBFF	Port 03 TX FIFO block 0–3	FIFO RAM block 1
DC00–DDFF	Port 04 RX FIFO block 0–3	FIFO RAM block 1
DE00–DFFF	Port 04 TX FIFO block 0–3	FIFO RAM block 2
E000–E1FF	Port 05 RX FIFO block 0–3	FIFO RAM block 2
E200–E3FF	Port 05 TX FIFO block 0–3	FIFO RAM block 2
E400–E5FF	Port 06 RX FIFO block 0–3	FIFO RAM block 2
E600–E7FF	Port 06 TX FIFO block 0–3	FIFO RAM block 2
E800–E9FF	Port 07 RX FIFO block 0–3	FIFO RAM block 2
EA00–EBFF	Port 07 TX FIFO block 0–3	FIFO RAM block 2
EC00–EDFF	Port 08 RX FIFO block 0–3	FIFO RAM block 2
EE00–EFFF	Port 08 TX FIFO block 0–3	FIFO RAM block 2
F000–F1FF	Port 09 RX FIFO block 0–3	FIFO RAM block 2
F200–F3FF	Port 09 TX FIFO block 0–3	FIFO RAM block 2
F400–F5FF	Port 10 RX FIFO block 0–3	FIFO RAM block 2
F600–F7FF	Port 10 TX FIFO block 0–3	FIFO RAM block 2
F800–F9FF	Port 11 RX FIFO block 0–3	FIFO RAM block 2
FA00–FBFF	Port 11 TX FIFO block 0–3	FIFO RAM block 2
FC00–FDFF	Port 12 RX FIFO block 0–3	FIFO RAM block 2
FE00–FFFF	Port 12 TX FIFO block 0–3	FIFO RAM block 2

port-N control register at 0x8*N + 0000 (N = port number in hex)

RESET CONDITIONS	**INITIAL VALUES AFTER RESET**							
	BIT							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	DISABLE	ENABLE	STFORTX	STFORRX	ADRDIS	MWIDTH	TXPACE	FORCEHD
$\overline{\text{M00UPLINK}}$ = 1	0	0	0	0	0	0	0	0
$\overline{\text{M00UPLINK}}$ = 0 ports 01–14	0	0	0	1	1	0	0	0
$\overline{\text{M00UPLINK}}$ = 0 port 00	0	0	0	1	0	0	0	0

BIT	**NAME**	**FUNCTION**
7	DISABLE	Port disable. Writing a 1 to DISABLE disables the port. Frames are not forwarded from or to a disabled port. The port, however, attempts to transmit any previously queued frames. DISABLE is a latched bit. It is set by both hard and soft reset.
6	ENABLE	Port enable. Writing a 1 to ENABLE enables the port if the DISABLE bit is not set. Writing a 0 to ENABLE has no effect. ENABLE is always read as 0.
5	STFORTX	Store and forward on transmission. Cut-through to this port is not allowed when STFORTX = 1.
4	STFORRX	Store and forward on receive. Cut-through from this port is disabled when STFORRX = 1.
3	ADRDIS	Address match disable. When ADRDIS = 1, the port does not take part in address-matching activity. Addresses are not captured for this port and any stored address is invalidated. Frames are not forwarded to the port except by EAM or BRUN functions. This permits selection between the ports that use external and internal address mappings. This allows the external address-match engine to be restricted to a subset of TNETX3150/TNETX3150A ports using the internal single-address lookup. If all ADRDIS bits = 1, all ports rely on the external address-match hardware. If a no-match code is received, the frame is discarded. If the uplink ADRDIS bit is set and a frame address has not been matched, the frame is discarded. ADRDIS should be set for all ports using external address hardware to discard frames using the EAM no-operation code.
2	MWIDTH	MII width selection. MWIDTH is valid only on 10-/100-Mbit/s capable ports (ports 00, 01, 02). When MWIDTH = 1 and the port is operated in 10-Mbit/s mode, the interface is operated in nibble-serial mode. When MWIDTH = 0, the interface is operated in bit-serial mode.
1	TXPACE	Transmit pacing. When TXPACE = 1, the port uses transmission pacing to enhance performance. When TXPACE = 0, transmit pacing is disabled.
0	FORCEHD	Force half duplex. When FORCEHD = 1, the DUPLEX terminal is pulled down (active pulldown on the input), forcing the PHY to operate in half-duplex mode.

port-N status register at 0x8*N + 0x0001 (read only) (N = port number in hex)

INITIAL VALUES AFTER RESET							
BIT							
7	6	5	4	3	2	1	0
UPDATE	NLINK	PROTOCOL	SPEED	DUPLX	Port state		
—	—	—	—	—	1	0	0

BIT	NAME	FUNCTION
7	UPDATE	Transmit queue-length update pending. UPDATE indicates when the transmit queue-length information has been updated for this port. UPDATE = 1 pending a transmit queue-length at initialization and when a queue-length update is pending. UPDATE = 0 when the update is complete. UPDATE is maintained only for ports with the LINK signal active. Any port with LINK inactive is not updated.
6	NLINK	Not link. NLINK indicates that the port’s link is inactive. NLINK reports the inverse of the state of the port’s MXXLINK terminal.
5	PROTOCOL	Interface protocol. PROTOCOL indicates the protocol of the interface and is determined by the logic level on MXXPROTOCOL. When PROTOCOL = 1, it indicates a request/grant interface for datastreaming applications. When PROTOCOL = 0, it indicates a CSMA/CD interface. For 200-Mbit/s uplink operation, PROTOCOL should be 1 (refer to uplink section for details). For standard Ethernet applications, PROTOCOL should be 0.
4	SPEED	Network speed. SPEED indicates the speed of a network port. When set to a 1, it indicates 100 Mbit/s. When set to a 0, it indicates 10 Mbit/s. SPEED is a direct reflection of the state of the port’s MXXSPEED terminal (non-10-Mbit/s ports). Ports with 10 Mbit/s always have a 0 in SPEED.
3	DUPLX	Full-duplex network. DUPLX indicates that a network port is operating in full-duplex mode. When set to a 1, it indicates full duplex. When set to a 0, it indicates half duplex. DUPLX is a direct reflection of the state of the port’s MXXDUPLX terminal.
2–0	Port state	This field indicates the state of the port. 000 – Enabled 001 – Suspended due to link failure 010 – Suspended due to address duplication 011 – Suspended due to address mismatch 100 – Disabled by management 101 – Reserved 110 – Disabled due to address duplication 111 – Disabled due to address mismatch Reset places all ports in state 100 (disabled by management). Completion of buffer memory initialization (START complete) places all ports in state 000 (enabled) unless port DISABLE bit is set.

port-N status register at 0x8*N + 0x0001 (read only) (N = port number in hex) (continued)

The uplink port (port 00) does not have a port address, so it cannot enter either address mismatch state. It can receive frames with source addresses securely assigned to other ports. In such cases, if SECDIS bit is set, the port enters state 110 (disabled due to address duplication). Port suspension is not supported because a network port naturally receives frames with different source addresses. Waiting for the source address to change is not useful.

PORT STATE	DESCRIPTION
000	Enabled. This is the normal state of a port. This is the only port state when frames are forwarded to and from the port. In all other states, no new frames are forwarded to or from the port.
001	Suspended due to link failure. The port has been suspended due to the absence of link activity at the port, as indicated by an inactive (zero) state of the port's MXXLINK terminal. This can indicate cable failure or no station attached to the port. The port is re-enabled once link activity is detected at the port, as indicated by an active (1) state of the port's MXXLINK terminal. If link is lost during transmission of a frame, the transmitted frame is lost and transmission continues until the start of the next frame.
010	Suspended due to address duplication. The port has been suspended due to the reception at the port of a frame with a source address securely assigned to another port. The port is re-enabled if a frame is received at the port with a source address not securely assigned to another port. A port in this state also can be re-enabled by writing a 1 to the ENABLE control bit.
011	Suspended due to address mismatch. The port has been suspended due to the reception at the port of a frame with a source address different from that securely assigned to it. The port is re-enabled if a frame is received at the port with a source address equal to the address securely assigned to it. A port in this state also can be re-enabled by writing a 1 to the ENABLE control bit.
100	Disabled by management. The port has been explicitly disabled by a DISABLE control bit write (or it is in the buffer initialization state). In this state, the port can be re-enabled only by writing a 1 to the ENABLE control bit, or by clearing the disable bit.
101	Reserved
110	Disabled due to address duplication. The port has been disabled due to the reception at the port of a frame with a source address securely assigned to another port. In this state, no frames are forwarded to or from the port, and no address learning takes place. A port in this state can only be re-enabled by writing a 1 to the ENABLE control bit.
111	Disabled due to address mismatch. The port has been disabled due to the reception at the port of a frame with a source address different from that securely assigned to it. In this state, no frames are forwarded to or from the port. A port in this state can be re-enabled only by writing a 1 to the ENABLE control bit.

port-N address registers at 0x8*N + 0x0002 through 0x8*N + 0x0007 (N = port number in hex)

+2	+3	+4	+5	+6	+7
Port address (47–0)					

These six byte-wide registers hold the port’s assigned source address, and are used to control address assignment and security for the port. Together, these six registers contain a 47-bit IEEE Std 802-specific MAC address and a security enable bit. This bit is in the address group/specific (G/S) bit. The G/S bit is the first bit of address from the wire, but because of the least-significant bit-first addressing Ethernet scheme, this corresponds to the least-significant bit of the first byte, or address bit 40.

The security enable bit (port address 40) is used to indicate the use of secure addressing on a port. In the secure addressing mode, once an address is assigned to a port, that source address can be used only with that port, and that port only with that source address. Use of that source address on another port causes it to be suspended or disabled. Use of a different source address on the secured port causes it to be suspended or disabled.

The uplink port (port 00) does not have a port address. The port address registers for port 00 (DIO addresses 0x0002–0x0007) cannot be written and are always read as zero.

An address can be assigned to a port in two different ways: explicitly or dynamically. An address is explicitly assigned by writing it to the port address registers. An address is assigned dynamically by the TNETX3150/TNETX3150A hardware loading the register from the source address field of received frames. If a port is in secured mode, the address is loaded only once from the first received frame. In unsecured mode, the address is updated on every frame received. (TNETX3150/TNETX3150A never assigns a duplicate port address. If the address is securely assigned to another port, then this port is placed in an unaddressed state – the address is set to zero – null address. If the address is assigned to another port, but not securely, then the other port is placed in an unaddressed state.)

- Writing 0x00.00.00.00.00.00 to the registers places the port in an unsecured, unaddressed state.
- Writing 0x01.00.00.00.00.00 to the registers places the port in a secured, unaddressed state.
- Writing a non-zero address (with bit 40 clear) sets the port address in an unsecured state.
- Writing a non-zero address (with bit 40 set) sets the port address in a secured state.

To prevent dynamic updating of the port address during DIO writes to the address registers, (which would create a corrupt address), dynamic updating is disabled by writes to the first address register (47–40) and reenabled by writes to the last address register (7–0). Ensure that all six bytes are always written and in the correct order.

transmit queue-length registers at 0x2*N + 0x0080 through 0x2*N + 0x0081

+3	+2	+1	+0	DIO ADDRESS
Transmit queue 1 length		Transmit queue 0 length		0x80
Transmit queue 3 length		Transmit queue 2 length		0x84
Transmit queue 5 length		Transmit queue 4 length		0x88
Transmit queue 7 length		Transmit queue 6 length		0x8C
Transmit queue 9 length		Transmit queue 8 length		0x90
Transmit queue 11 length		Transmit queue 10 length		0x94
Transmit queue 13 length		Transmit queue 12 length		0x98
Reserved		Transmit queue 14 length		0x9C
Reserved	NMON	XCTRL/SIO	Revision register	0xA0

Transmit queues use a residual queue length to control their behavior. Its value indicates how many more buffers can be added to the queue, rather than how many buffers are on the queue. This is an advantage because it is easy to detect that the queue is full (length goes negative) and can be adjusted dynamically (2's complement addition to the length).

After reset, all transmit queue-length registers are initialized to zero. These registers are part of the address range that can be read from the EEPROM. The first time value written to the transmit queue-length registers is the initial value. When the user writes again, the value entered is added as a signed 16-bit integer to the current value.

As frames are placed on the queue, the transmit queue length is decremented by the number of buffers queued. Should the transmit queue length become negative (most-significant bit set and the queue is full), no new frames are added until the length becomes positive by the transmission of buffers. Since a maximum-size frame (1518 bytes) is 24 buffers long, and whole frames are queued based on the current transmit queue-length value, the queue can consume 23 more buffers than the initial residual length (i.e., if the transmit queue is set to length = 1, a full-size Ethernet frame still can be queued). As buffers are transmitted, the transmit queue length is incremented.

The transmit queue registers are used to initialize, alter, and provide status on transmit queue lengths. They are used in three different ways:

1. To assign initial transmit queue-length value. The value in the register is used as its initial value when the first frame is put on the queue.
2. To indicate current transmit queue-length value. The register is loaded with the transmit queue-length value when it is updated.
3. To adjust transmit queue length.

After transmit queue initialization, a value written to this register is added to the current transmit queue-length value the next time it is updated. The update bit in port status is used to detect initialization or that an update operation is complete. The operation is a 16-bit addition, allowing the current queue length to be increased or decreased. The update operation is enabled only when the most-significant byte of the register (15–8) is written to prevent possible length corruption. Ensure that length bytes are always written least-significant byte first.

revision register at 0x00A0†

BIT							
7	6	5	4	3	2	1	0
Revision							

BIT	NAME	FUNCTION
7–4	Revision	Major hardware revision code for this device. This field is read only. Initial samples have a revision code of 0x00.
3–0	Revision	Minor hardware revision code for this device. This field is read only. Initial samples have a revision code of 0x00.

† The TNETX3150 has a revision register of 0x01xx. The TNETX3150A has a revision register of 0x02xx.

SIO/XCTRL register at 0x00A1

XCTRL REGISTER				SIO REGISTER			
BIT				BIT			
7	6	5	4	3	2	1	0
WUPLINK	CUT100	RXARB	BRUN	Reserved	ECLOK	ETXEN	EDATA
Initial Values After Reset							
–	0	0	0	0	0	0	0

BIT	NAME	FUNCTION
7	WUPLINK	Wide uplink mode. WUPLINK reflects the status of the $\overline{\text{M00UPLINK}}$ strapping terminal (note that $\overline{\text{M00UPLINK}}$ is active low). Logic 1 = wide uplink mode (this bit is read only)
6	CUT100	Single-buffer cut-through operation on 100-Mbit/s ports only. CUT100 disables single-buffer cut-through operation for frames received on 10-Mbit/s source ports. A frame is transmitted only when two buffers have been transferred to the transmit FIFO or an end of frame (before two buffers) has been received. Although it increases latency, enabling CUT100 reduces the probability of dropping frames due to FIFO underrun in heavy bursty traffic.
5	RXARB	Receive arbitration mode. RXARB = 1 ensures that no frames are dropped during bursty conditions. When RXARB = 0, errored frames can be transmitted.
4	BRUN	Broadcast to unassigned ports. If no port address is matched, BRUN = 1 forces the TNETX3150/TNETX3150A to broadcast a unicast frame to all unassigned ports (ports that are enabled but do not have a port address). When BRUN = 0, all unmatched unicast frames are sent to the uplink port. BRUN = 0 if an external address-lookup device is used. The external address-lookup device should provide the function.
3	Reserved	Reserved.
2	ECLOK	EEPROM SIO clock. ECLOK controls the state of the ECLK terminal. When ECLOK = 1, ECLK is asserted. When ECLOK = 0, ECLK is deasserted. ECLOK also is used to determine the state of the EEPROM interface. If the EEPROM port is disabled, then ECLOK is always read as a 0, even if a value of 1 is written to the bit. the TNETX3150/TNETX3150A detects that the EEPROM port is disabled by sensing the state of the EDIO terminal during reset. If EDIO is read as a 0 during reset (due to an external pulldown resistor), then the EEPROM interface is disabled and no attempt is made to read configuration information.
1	ETXEN	EEPROM SIO transmit enable. ETXEN controls the direction of the EDIO terminal. When ETXEN = 1, EDIO is driven with the value in the EDATA bit. When ETXEN = 0, EDATA is loaded with the value on the EDIO terminal.
0	EDATA	EEPROM SIO data. EDATA is used to read or write the state of the EDIO terminal. When ETXEN = 1, EDIO is driven with the value in EDATA. When ETXEN = 0, EDATA is loaded with the value on the EDIO terminal.

system NMON register at 0x00A2

BIT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		MONRXTX	MONWIDE	NMON			
Initial Values After Reset							
0	0	0	0	0	0	0	0

<table>
<tr><th>BIT</th><th>NAME</th><th colspan="4">FUNCTION</th></tr>
<tr><td>7–6</td><td>Reserved</td><td colspan="4">Reserved</td></tr>
<tr><td>5</td><td>MONRXTX</td><td colspan="4">Selection of RX or TX signals when monitoring ports 00, 01, and 02 operating in nibble-interface format</td></tr>
<tr><td rowspan="3">4</td><td rowspan="3">MONWIDE</td><td colspan="4">Selection of monitor-port format width. When NMON = 0, the interface provides the SNI data format (only available for ports operating in SNI). When MONWIDE = 1:
● The NMON interface is configured for nibble (4-bit) data. If MONWIDE = 1 when a port is operating in SNI mode, only NMON00 is driven with data (NMON (01–03) are undriven).
● NMON06 is driven with an indication of the speed of the port; 0 = 10 Mbit/s, 1 = 100 Mbit/s.</td></tr>
<tr><td>NMON Terminal Name</td><td>MONWIDE = 0
MONRXTX = X</td><td>MONWIDE = 1
MONRXTX = 0</td><td>MONWIDE =1
MONRXTX = 1</td></tr>
<tr><td>NMON00
NMON01
NMON02
NMON03
NMON04
NMON05
NMON06</td><td>MXXRXD
MXXCRS
MXXRCLK
MXXTXD
MXXTXEN
MXXTCLK
MXXCOL</td><td>MXXRXD(0)
MXXRXD(1)
MXXRXD(2)
MXXRXD(3)
MXXRXDV
MXXRCLK
MXXSPEED</td><td>MXXTXD(0)
MXXTXD(1)
MXXTXD(2)
MXXTXD(3)
MXXTXEN
MXXTCLK
MXXSPEED</td></tr>
<tr><td rowspan="4">3–0</td><td rowspan="4">NMON</td><td colspan="4">This nibble controls which port is monitored when using the NMON function.</td></tr>
<tr><th colspan="2">NMON FIELD CODE</th><th colspan="2">DESCRIPTION</th></tr>
<tr><td colspan="2">0000–1110</td><td colspan="2">Ports 00–14 selected for monitoring. Port 00 (uplink) can be monitored only when $\overline{\text{M00UPLINK}}$ is high.</td></tr>
<tr><td colspan="2">1111</td><td colspan="2">Disables the NMON function</td></tr>
</table>

port VLAN registers at 0x2*N + 0x00A4 through 0x2*N + 0x00A5 (N = port number in hex)

The VLAN registers hold broadcast destination masks for each source port when IOB is in operation. Each bit in the VLAN register (with exception of bit 15) directly corresponds to a port (bit 14 = port 14 through bit 00 = port 00). Broadcast and multicast frames are directed according to the VLAN register setting for the port on which the broadcast or multicast frame was received. Each VLAN register is initialized at reset to send frames to all other ports except itself. After reset, the registers contain the following initial values (see Table 9).

Table 9. VLAN Register Initial Values

REGISTER NAME	INITIAL VALUE		DIO ADDRESS
	+1	+0	
VLAN 0 MASK	01111111	11111110	0x00A4
VLAN 1 MASK	01111111	11111101	0x00A6
VLAN 2 MASK	01111111	11111011	0x00A8
VLAN 3 MASK	01111111	11110111	0x00AA
VLAN 4 MASK	01111111	11101111	0x00AC
VLAN 5 MASK	01111111	11011111	0x00AE
VLAN 6 MASK	01111111	10111111	0x00B0
VLAN 7 MASK	01111111	01111111	0x00B2
VLAN 8 MASK	01111110	11111111	0x00B4
VLAN 9 MASK	01111101	11111111	0x00B6
VLAN 10 MASK	01111011	11111111	0x00B8
VLAN 11 MASK	01110111	11111111	0x00BA
VLAN 12 MASK	01101111	11111111	0x00BC
VLAN 13 MASK	01011111	11111111	0x00BE
VLAN 14 MASK	00111111	11111111	0x00C0

When EAM bit-map direction is in use, the VLAN registers are used to store the bit mask from the EAM. VLAN registers can be loaded only before the DRAM initialization (before the START bit is set).

RAM size register at 0x00C2

BIT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MTEST	Reserved		TAGOFF	RSIZE			
Initial Values After Reset		Initial Values After Reset					
0	X	X	0	0	1	0	0

BIT	NAME	FUNCTION
7	MTEST	Test access bit. MTEST = 1 before DIO accesses are made to the DIO test registers (address range 0xDC–0xFF). MTEST = 0 when the EEPROM is used to initialize registers 0x00–0xC3. The default value after reset is 0. MTEST allows access to the DIO registers to enable frame-wrap test modes.
6–5	Reserved	Reserved
4	TAGOFF	Post-frame tag disable bit. TAGOFF disables the requirement for post-frame tagging when operating in 200-Mbit/s wide uplink mode. In this mode (TAGOFF = 1), the TNETX3150/TNETX3150A uses either the EAM interface or the internal single-address lookup registers to route frames received on the uplink. This permits two TNETX3150/TNETX3150A devices to be cascaded.
3–0	RSIZE	RAM size select. This field indicates the size of the external DRAM and, therefore, the number of 64-byte data buffers available.† This field is used by the TNETX3150/TNETX3150A to determine how many buffers to initialize. The code values are:
3–0	RSIZE	
3–0	RSIZE	

BIT		CODE VALUES	NUMBER OF BUFFERS
3	**0**		
0000–0011		Reserved	—
0100		4K × 36	240
0101		8K × 36	480
0110		16K × 36	961
0111		32K × 36	1922
1000		64K × 36	3845
1001		128K × 36	7690
1010		256K × 36	15,380
1011		512K × 36	30,720
1100		1M × 36	61,440
1101		2M × 36	122,880
1110		4M × 36	245,760
1111		Reserved	—

† Buffers are 76.5 bytes long (four bytes of forward pointer). Fifteen 76.5-byte buffers are allocated per 1K-byte page. The first word of every 1K-byte page is not used. Buffers never straddle page boundaries.

system control register at 0x00C3

BIT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESET	LOAD	START	CLRSTS	STMAP	SECDIS	LONG	IOBMOD
Initial Values After Reset							
0	0	0	0	0	0	0	0

BIT	NAME	FUNCTION
7	RESET	RESET = 1 places the TNETX3150/TNETX3150A in a software reset state. Writing a 0 clears the reset state and allows for internal testing while keeping the ports disabled. RESET clears the port control register but does not affect any other configuration registers. Configuration registers are cleared by a hardware reset (write to DIO address 4000-7FFF).
6	LOAD	Load system. Writing a 1 to LOAD causes the TNETX3150/TNETX3150A DIO registers to auto-load from an external EEPROM (if present). All registers in the DIO address range 0x00–0xC3 are loaded from the corresponding EEPROM locations. Writing a 0 to LOAD has no effect. LOAD is read as a 1 until the auto-load is complete. LOAD is not auto-loaded. LOAD is always set to 0 by auto-load.
5	START	Start system. Writing a 1 to START causes the TNETX3150/TNETX3150A to begin operation. START is read as a 1 until buffer memory initialization is complete. While buffers are being initialized, all ports are disabled. Writing a 0 to START has no effect.
4	CLRSTS	Clear statistics. Writing a 1 to CLRSTS causes the TNETX3150/TNETX3150A to clear all its statistics counters. The TNETX3150/TNETX3150A repeats clearing the statistics counters until CLRSTS is cleared.
3	STMAP	Statistic mapping. STMAP selects which statistic is recorded in multiple-function statistic counters. Setting STMAP to a 1 selects the statistic to record the number of TX frames discarded on TX due to lack of resources. If STMAP is reset to a 0, the statistic records the number of data errors at TX.
2	SECDIS	Disable ports on security violations. When SECDIS is set to a 1, address security violations cause a port to be disabled. When SECDIS is set to a 0, address security violations cause a port to be suspended. Suspended ports are reenabled when the offending condition is removed. Disabled ports can be reenabled only by management (by setting port ENABLE bit).
1	LONG	Long frame handling. When LONG is set to a 1, the statistics counter for giant frames is recorded in the RX and TX frame 1024–1518 bucket counter, which for this mode is redefined to become RX and TX frames 1024–1531. Frames 1536 bytes or larger are truncated.
0	IOBMOD	IOB mode. When IOBMOD is set to a 1, broadcast/multicast frames are sent to a destination in sequence with unicast frames from the same source port, using the IOB buffer linking mechanism. To ensure proper frame forwarding, IOBMOD is set to a 1.

DRAM data register at 0x00D4–0x00D7

BIT	
31	**0**
DRAM DATA	

BIT	NAME	FUNCTION
31–0	DRAM DATA	DRAM Data. DRAM DATA holds a 32-bit data value that maps to the forward pointer field of a DRAM buffer when accessed in DRAM test-access mode.

DRAM flag register at 0x00D8

BIT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DRAMACT	Reserved			DRAMFLAG			

BIT	NAME	FUNCTION
7	DRAMACT	DRAM test access. DRAMACT contains the status of a DRAM test access READ or WRITE. When this activity bit is high, the DRAM access is being performed. When this bit is low, the DRAM access has completed. After a DRAM test access buffer is read, the user should detect a falling edge on this bit before proceeding to use the accessed data.
6–4	Reserved	Reserved
3–0	DRAMFLAG	DRAM flag field. DRAMFLAG holds a value that maps to the flag field of a DRAM buffer when accessed in DRAM test-access mode.

DRAM addr register at 0x00D9–0x00DB

BIT																							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R/W	DRAMADDRESS																						

BIT	NAME	FUNCTION
23	R/W	DRAM test access read/write. R/W determines whether the contents of channel 0's FIFO, DRAM data, and DRAMFLAG are read from DRAM or written to DRAM. When high, the write operation is performed. When low, a read operation is performed.
22–0	DRAMADDRESS	DRAM starting word address. DRAMADDRESS is a 23-bit DRAM address marking the starting word location for a DRAM test access buffer operation.

The DRAM address space (as used in this register) is not flat. It is partitioned in the following manner:

BIT																						
22	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	DX02 RAS	DX02 CAS	DX01 RAS	DX01 CAS	DX00 RAS	DX00 CAS	Row address (8 bits)								Column address (8 bits)							

BIT	NAME	FUNCTION
22	Reserved	
21	DX02	Extended address bit 2 (RAS)
20	DX02	Extended address bit 2 (CAS)
19	DX01	Extended address bit 1 (RAS)
18	DX01	Extended address bit 1 (CAS)
17	DX00	Extended address bit 0 (RAS)
16	DX00	Extended address bit 0 (CAS)
15–8	RAS	Row address for DRAM (most-significant bit = bit 15)
7–0	CAS	Column address for DRAM (most-significant bit = bit 7)

DRAM test-access operation

+3	+2	+1	+0	DIO ADDRESS
DRAM data				0x00D4–0x00D7
DRAM addr			DRAM flag	0x00D8–0x00DB
INIST	PACTST	DIATST	Reserved	0x00DC–0x00DF

The user can write and read a repeating DRAM test by using the following procedure.

- Soft reset the TNETX3150/TNETX3150A (do not set the start bit). Perform the following:

```
DIO write (address 00C3) = 80
DIO write (address 00C3) = 00
```

- Write to the port 00 TX FIFO block (72 bytes total):

```
DIO write (address CE00) = A5
DIO write (address CE01) = A5
DIO write (address CE02) = A5
DIO write (address CE03) = A5
DIO write (address CE04) = A5
DIO write (address CE05) = A5
DIO write (address CE06) = A5
DIO write (address CE07) = A5
DIO write (address CE08) = A5
```

- The FIFO can be filled by writing to the following offset:

```
DIO write (address C208 + (10 * n)) = A5    where n = 0–7
```

- Write the DRAM forward pointer, flag,and DRAM address:

```
DIO write (address 00D4) = FF      ;Forward pointer least-significant bit
DIO write (address 00D5) = FF      ;Forward pointer
DIO write (address 00D6) = FF      ;Forward pointer
DIO write (address 00D7) = FF      ;Forward pointer most-significant bit
DIO write (address 00D8) = xF      ;Flag
DIO write (address 00D9) = 00      ;DRAM address least-significant bit
DIO write (address 00DA) = 00      ;DRAM address
DIO write (address 00DB) = 80      ;DRAM address most-significant bit
```

DRAM test-access operation (continued)

- The DRAM access occurs after the high byte (location 00DB) of the DRAM_address register is completed. To verify the DRAM access is complete, read the DRAMACT bit within the DRAM flag register:

  ```
  DIO read (address 00D8) = 8x      DMA transfer not complete
  DIO read (address 00D8) = 0x      DMA transfer complete
  ```

- Clear the forward pointer, flag and perform a DRAM read:

  ```
  DIO write (address 00D4) = 00     ;Forward pointer least-significant bit
  DIO write (address 00D5) = 00     ;Forward pointer
  DIO write (address 00D6) = 00     ;Forward pointer
  DIO write (address 00D7) = 00     ;Forward pointer most-significant bit
  DIO write (address 00D8) = x0     ;Flag
  DIO write (address 00D9) = 00     ;DRAM address least-significant bit
  DIO write (address 00DA) = 00     ;DRAM address
  DIO write (address 00DB) = 00     ;DRAM address most-significant bit
  ```

- Now wait for the read from DRAM to complete by polling the DRAMACT bit within the DRAM flag register:

  ```
  DIO read (address 00D8) = 8x      DMA transfer not complete
  DIO read (address 00D8) = 0x      DMA transfer complete
  ```

- Read the forward pointer, flag, and DRAM data:

  ```
  DIO read (address 00D4–00D7)
  DIO read (address 00D8)
  DIO read (address CC00 + (10 * n)) through (address CC08 + (10 * n))    where n = 0–7
  ```

test registers

DIATST register at 0x00DD

BIT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				DPWRAP	INTWRAP		Reserved
Initial Values After Reset							
X	X	X	X	0	0	0	0

BIT	NAME	FUNCTION
7–4	Reserved	Reserved
3	DPWRAP	Duplex wrap mode. When DPWRAP = 1, all ports are forced into full-duplex mode and all ports can receive frames they transmit. This enables external wrap testing at the PHY.
2–1	INTWRAP	Internal wrap mode. Ports 1–14 internally wrap back according to the following two-bit coding (INTWRAP bit 2 and bit 1, respectively). 00 No internal wrapping 01 All ports internally wrapped except port 00 (uplink) 10 All ports internally wrapped except port 02 11 All ports internally wrapped except port 14 (The port that is not wrapped (00, 02, or 14) should be used to inject and observe test data frames from the internally wrapped ports.)
0	Reserved	

*port N statistics at 0x80*N + 0x8000 through 0x80*N + 0x807F (N = port number in hex)*

+7	+6	+5	+4	+3	+2	+1	+0	ADDRESS	
Good RX frames				RX octets				0x80*N + 0x8000	+0x00–0x07
Multicast RX frames				Broadcast RX frames				0x80*N + 0x8000	+0x08–0x0F
RX align/code errors				RX CRC errors				0x80*N + 0x8000	+0x10–0x17
RX jabbers				Oversize RX frames				0x80*N + 0x8000	+0x18–0x1F
RX fragments				Undersize RX frames				0x80*N + 0x8000	+0x20–0x27
Frames 65–127				Frame 64				0x80*N + 0x8000	+0x28–0x2F
Frames 256–511				Frames 128–255				0x80*N + 0x8000	+0x30–0x37
Frames 1024–1518				Frames 512–1023				0x80*N + 0x8000	+0x38–0x3F
SQE test errors				Net octets				0x80*N + 0x8000	+0x40–0x47
Good TX frames				TX octets				0x80*N + 0x8000	+0x48–0x4F
Multi-collision TX frames				Single-collision TX frames				0x80*N + 0x8000	+0x50–0x57
Deferred TX frames				Carrier sense errors				0x80*N + 0x8000	+0x58–0x5F
Excessive collisions				Late collisions				0x80*N + 0x8000	+0x60–0x67
Multicast TX frames				Broadcast TX frames				0x80*N + 0x8000	+0x68–0x6F
TX data errors†				Filtered RX frames				0x80*N + 0x8000	+0x70–0x77
Address changes/mismatches				Address duplications				0x80*N + 0x8000	+0x78–0x7F

† The operation of this counter is controlled by the STMAP bit in the system control register.

*port N RX overrun and collision statistics at 0x8*N + 0x8780 through 0x8*N + 0x87FF (N = port number in hex)*

+7	+6	+5	+4	+3	+2	+1	+0	ADDRESS
RX overrun port 00				Collision port 00				0x8780 +0x00–0x07
RX overrun port 01				Collision port 01				0x8780 +0x08–0x0F
RX overrun port 02				Collision port 02				0x8780 +0x10–0x17
RX overrun port 03				Collision port 03				0x8780 +0x18–0x1F
RX overrun port 04				Collision port 04				0x8780 +0x20–0x27
RX overrun port 05				Collision port 05				0x8780 +0x28–0x2F
RX overrun port 06				Collision port 06				0x8780 +0x30–0x37
RX overrun port 07				Collision port 07				0x8780 +0x38–0x3F
RX overrun port 08				Collision port 08				0x8780 +0x40–0x47
RX overrun port 09				Collision port 09				0x8780 +0x48–0x4F
RX overrun port 10				Collision port 10				0x8780 +0x50–0x57
RX overrun port 11				Collision port 11				0x8780 +0x58–0x5F
RX overrun port 12				Collision port 12				0x8780 +0x60–0x67
RX overrun port 13				Collision port 13				0x8780 +0x68–0x6F
RX overrun port 14				Collision port 14				0x8780 +0x70–0x77
Reserved				Reserved				0x8780 +0x78–0x7F

When accessing the statistics values from the DIO port, it is necessary to perform four 1-byte DIO reads to obtain the full 32-bit counter. To prevent the chance of the counter being updated while reading the four bytes, the user should access the low byte first, followed by the upper three bytes. On reading the low byte, the counter statistic value is transferred to a 32-bit holding register before being placed on the DIO bus. The register is updated only when reading the low byte of the counter statistic. When the statistics values are accessed in this way, the user does not see spurious updates.

The statistics, RX overrun, and collision registers are cleared only during a reset or when bit 4 (CLRSTS) of the system control register at 0x00C3 is set. When the registers roll over, a roll-over indication is not given.

transmit queue (TXQ) structures address map at 0x8*N + 0x8800 through 0x8*N + 0x8807 (N = port number in hex)

+7	+6	+5	+4	+3	+2	+1	+0	ADDRESS
Transmit queue 0 head			Transmit queue 0 tail			Transmit queue 0 length		0x8800–0x8807
Transmit queue 1 head			Transmit queue 1 tail			Transmit queue 1 length		0x8808–0x880F
Transmit queue 2 head			Transmit queue 2 tail			Transmit queue 2 length		0x8810–0x8817
Transmit queue 3 head			Transmit queue 3 tail			Transmit queue 3 length		0x8818–0x881F
Transmit queue 4 head			Transmit queue 4 tail			Transmit queue 4 length		0x8820–0x8827
Transmit queue 5 head			Transmit queue 5 tail			Transmit queue 5 length		0x8828–0x882F
Transmit queue 6 head			Transmit queue 6 tail			Transmit queue 6 length		0x8830–0x8837
Transmit queue 7 head			Transmit queue 7 tail			Transmit queue 7 length		0x8838–0x883F
Transmit queue 8 head			Transmit queue 8 tail			Transmit queue 8 length		0x8840–0x8847
Transmit queue 9 head			Transmit queue 9 tail			Transmit queue 9 length		0x8848–0x884F
Transmit queue 10 head			Transmit queue 10 tail			Transmit queue 10 length		0x8850–0x8857
Transmit queue 11 head			Transmit queue 11 tail			Transmit queue 11 length		0x8858–0x885F
Transmit queue 12 head			Transmit queue 12 tail			Transmit queue 12 length		0x8860–0x8867
Transmit queue 13 head			Transmit queue 13 tail			Transmit queue 13 length		0x8868–0x886F
Transmit queue 14 head			Transmit queue 14 tail			Transmit queue 14 length		0x8870–0x8877
Reserved								0x8878–0x887F

immediate queue (IMQ) structures map at 0x8*N + 0x8800 through 0x8*N + 0x8887 (N = port number in hex)

+7	+6	+5	+4	+3	+2	+1	+0	ADDRESS
immediate queue 0 head			immediate queue 0 tail			immediate queue 0 length		0x8880–0x8887
immediate queue 1 head			immediate queue 1 tail			immediate queue 1 length		0x888–0x888F
immediate queue 2 head			immediate queue 2 tail			immediate queue 2 length		0x8890–0x8897
immediate queue 3 head			immediate queue 3 tail			immediate queue 3 length		0x8898–0x889F
immediate queue 4 head			immediate queue 4 tail			immediate queue 4 length		0x88A0–0x88A7
immediate queue 5 head			immediate queue 5 tail			immediate queue 5 length		0x88A8–0x88AF
immediate queue 6 head			immediate queue 6 tail			immediate queue 6 length		0x88B0–0x88B7
immediate queue 7 head			immediate queue 7 tail			immediate queue 7 length		0x88B8–0x88BF
immediate queue 8 head			immediate queue 8 tail			immediate queue 8 length		0x88C0–0x88C7
immediate queue 9 head			immediate queue 9 tail			immediate queue 9 length		0x88C8–0x88CF
immediate queue 10 head			immediate queue 10 tail			immediate queue 10 length		0x88D0–0x88D7
immediate queue 11 head			immediate queue 11 tail			immediate queue 11 length		0x88D8–0x88DF
immediate queue 12 head			immediate queue 12 tail			immediate queue 12 length		0x88E0–0x88E7
immediate queue 13 head			immediate queue 13 tail			immediate queue 13 length		0x88E8–0x88EF
immediate queue 14 head			immediate queue 14 tail			immediate queue 14 length		0x88F0–0x88F7
Reserved								0x88F8–0x88FF

receive queue (RXQ) structures address map at 0x8*N + 0x8900 through 0x8*N + 0x8907 (N = port number in hex)

+7	+6	+5	+4	+3	+2	+1	+0	ADDRESS
Receive queue 0 head			Receive queue 0 tail			Receive queue 0 length		0x8900–0x8907
Receive queue 1 head			Receive queue 1 tail			Receive queue 1 length		0x8908–0x890F
Receive queue 2 head			Receive queue 2 tail			Receive queue 2 length		0x8910–0x8917
Receive queue 3 head			Receive queue 3 tail			Receive queue 3 length		0x8918–0x891F
Receive queue 4 head			Receive queue 4 tail			Receive queue 4 length		0x8920–0x8927
Receive queue 5 head			Receive queue 5 tail			Receive queue 5 length		0x8928–0x892F
Receive queue 6 head			Receive queue 6 tail			Receive queue 6 length		0x8930–0x8937
Receive queue 7 head			Receive queue 7 tail			Receive queue 7 length		0x8938–0x893F
Receive queue 8 head			Receive queue 8 tail			Receive queue 8 length		0x8940–0x8947
Receive queue 9 head			Receive queue 9 tail			Receive queue 9 length		0x8948–0x894F
Receive queue 10 head			Receive queue 10 tail			Receive queue 10 length		0x8950–0x8957
Receive queue 11 head			Receive queue 11 tail			Receive queue 11 length		0x8958–0x895F
Receive queue 12 head			Receive queue 12 tail			Receive queue 12 length		0x8960–0x8967
Receive queue 13 head			Receive queue 13 tail			Receive queue 13 length		0x8968–0x896F
Receive queue 14 head			Receive queue 14 tail			Receive queue 14 length		0x8970–0x8977
Reserved								0x8978–0x897F

EEPROM interface

The EEPROM interface is provided so the system-level manufacturer can optionally provide a preconfigured system to their customers. Customers also can change or reconfigure their system and retain their preferences between system power downs.

The EEPROM contains configuration and initialization information, which is accessed infrequently (typically at power up and reset).

The TNETX3150/TNETX3150A uses the 24C02 serial EEPROM device (2048 bits organized as 256 bits × 8). See Table 10.

The organization of the EEPROM data is the same format as the TNETX3150/TNETX3150A registers 0x00–0xC3 (see section on internal registers). This allows a complete initialization to be performed by downloading the contents of the EEPROM into the TNETX3150/TNETX3150A. During the download, no DIO operations are permitted. The LOAD bit in the system control registers cannot be set during a download, preventing a download loop. The LOAD bit is reset after completion of the download.

TNETX3150/TNETX3150A detects the presence/absence of the EEPROM. If no EEPROM is installed, the EDIO terminal should be tied low. For EEPROM operation, the terminal requires an external pullup (see EEPROM data sheet). If no EEPROM is detected, the TNETX3150/TNETX3150A assumes default modes of operation at power up. Downloading the configuration from the EEPROM terminals is disabled when no EEPROM is present. The timing information for the EEPROM interface is provided in Figures 20 and 21.

EEPROM auto-configuration from an external x24C02 EEPROM

The EEPROM can be initialized or reprogrammed through the DIO/host interface using a suitable software driver.

The organization of the EEPROM data is shown in Table 10. The last register loaded is the control register. This allows a complete initialization by downloading the contents of the EEPROM into the external-address lookup devices. During the download, no DIO operations are permitted. The LOAD and RESET bits in the control register cannot be set during a download, preventing a download loop.

The TNETX3150/TNETX3150A detects the presence/absence of the EEPROM. If it is not installed, the EDIO terminal should be tied low. For EEPROM operation, the terminal requires an external pullup (see EEPROM data sheet). When no EEPROM is detected, the TNETX3150/TNETX3150A assumes default register values at power up and is halted. Downloading a configuration from the EEPROM terminals is disabled when no EEPROM is present.

The first bit written to or read from the EEPROM is the most-significant bit of the byte, i.e., data (7). Therefore, writing the address 0xC0h is accomplished by writing a 1 and then 1, 0, 0, 0, 0, 0, 0.

The TNETX3150/TNETX3150A expects data to be stored in the EEPROM in a specific format. The range from 0x0000h – 0x00C3h in the EEPROM is reserved for use by the adapter. The contents of the remaining bytes are undefined. The EEPROM can be read/written by a software driver through the SIO register.

A 32-bit CRC value must be calculated from the EEPROM data and placed in the EEPROM in the location following the bytes loaded into the internal register. the TNETX3150/TNETX3150A uses this 32-bit CRC to validate the EEPROM data. If the CRC fails, the TNETX3150/TNETX3150A registers are set to their default (hardwired) values. The TNETX3150/TNETX3150A is then placed in a post-reset halted state. The TNETX3150/TNETX3150A must be started through the DIO interface control register START bit.

The EEPROM algorithm, which is the same as the algorithm for the TNETX3150/TNETX3150A EEPROM CRC, is used by IEEE Std 802.3 for the packet CRC calculation. EEPROM bytes are processed by the internal logic in sequence just as bytes of a packet are processed. For a description of the algorithm, see Section 3.2.8. of the IEEE Std 802.3u specification. For reference, the equation is:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

EEPROM auto-configuration from an external x24C02 EEPROM (continued)

Table 10. EEPROM Address Map

EEPROM PHYSICAL ADDRESS	DESCRIPTION
00	Port 00 control
01	Port 00 status
02–07	Port 00 address register (bits 47–0)
08	Port 01 control
09	Port 01 status
0A–0F	Port 01 address register (bits 47–0)
10	Port 02 control
11	Port 02 status
12–17	Port 02 address register (bits 47–0)
18	Port 03 control
19	Port 03 status
1A–1F	Port 03 address register (bits 47–0)
20	Port 04 control
21	Port 04 status
22–27	Port 04 address register (bits 47–0)
28	Port 05 control
29	Port 05 status
2A–2F	Port 05 address register (bits 47–0)
30	Port 06 control
31	Port 06 status
32–37	Port 06 address register (bits 47–0)
38	Port 07 control
39	Port 07 status
3A–3F	Port 07 address register (bits 47–0)
40	Port 08 control
41	Port 07 status
42–47	Port 08 address register (bits 47–0)
48	Port 09 control
49	Port 09 status
4A–4F	Port 09 address register (bits 47–0)
50	Port 10 control
51	Port 10 status
52–57	Port 10 address register (bits 47–0)
58	Port 11 control
59	Port 11 status
5A–5F	Port 11 address register (bits 47–0)
60	Port 12 control
61	Port 12 status
62–67	Port 12 address register (bits 47–0)
68	Port 13 control
69	Port 13 status
6A–6F	Port 13 address register (bits 47–0)
70	Port 14 control
71	Port 14 status
72–77	Port 14 address register (bits 47–0)
78–7F	Reserved
80–81	Port 00 transmit queue length
82–83	Port 01 transmit queue length
84–85	Port 02 transmit queue length
86–87	Port 03 transmit queue length
88–89	Port 04 transmit queue length
8A–8B	Port 05 transmit queue length
8C–8D	Port 06 transmit queue length
8E–8F	Port 07 transmit queue length
90–91	Port 08 transmit queue length
92–93	Port 09 transmit queue length
94–95	Port 10 transmit queue length
96–97	Port 11 transmit queue length
98–99	Port 12 transmit queue length
9A–9B	Port 13 transmit queue length
9C–9D	Port 14 transmit queue length
9E–9F	Reserved
A0	Revision register
A1	SIO/SCTRL register
A2	NMON register
A3	Reserved
A4–A5	Port 00 VLAN register
A6–A7	Port 01 VLAN register
A8–A9	Port 02 VLAN register
AA–AB	Port 03 VLAN register
AC–AD	Port 04 VLAN register
AE–AF	Port 05 VLAN register
B0–B1	Port 06 VLAN register
B2–B3	Port 07 VLAN register
B4–B5	Port 08 VLAN register
B6–B7	Port 09 VLAN register
B8–B9	Port 10 VLAN register
BA–BB	Port 11 VLAN register
BC–BD	Port 12 VLAN register
BE–BF	Port 13 VLAN register
C0	Port 14 VLAN register
C1	Port 14 VLAN register
C2	RAM size register
C3	System control register
C4	CRC byte 3
C5	CRC byte 2
C6	CRC byte 1
C7	CRC byte 0
C8–FF	Reserved

write cycle timing

Following a start condition, the master must output the address of the slave it is accessing. The most-significant four bits of the slave address are those of the device type identifier (see Figure 16). This is fixed as 1010 for all devices. The EEPROM device address must be 000.

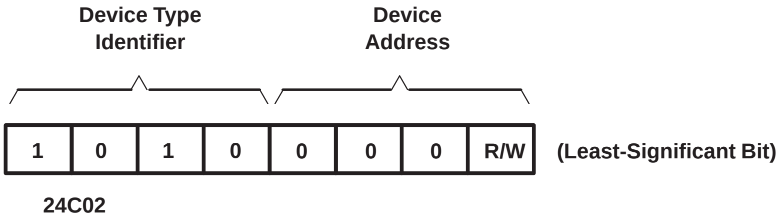


Figure 16. Device-Type Identifier and Address

The ECLK is an output from the TNETX3150/TNETX3150A. EDIO is an input if the TNETX3150/TNETX3150A is reading the EEPROM or an output if it is written to the EEPROM (see Figures 17 through 19).

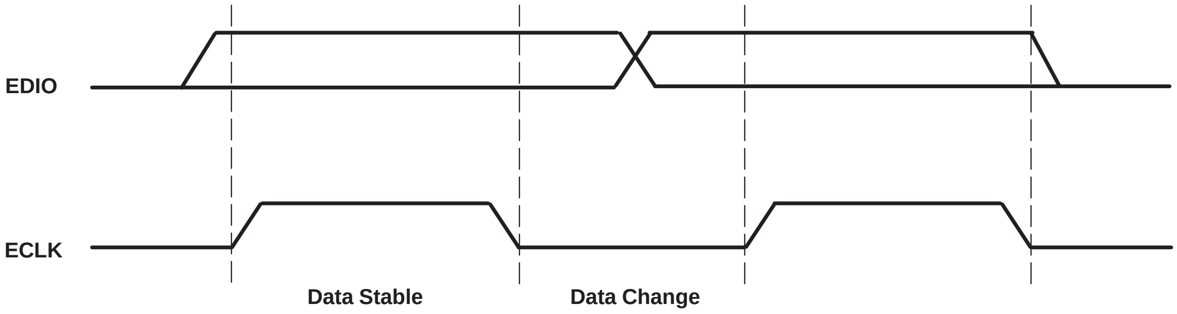


Figure 17. Data Validity

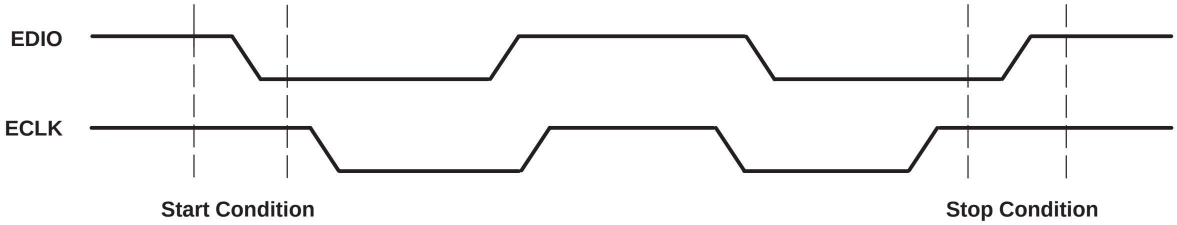


Figure 18. Start and Stop Definition

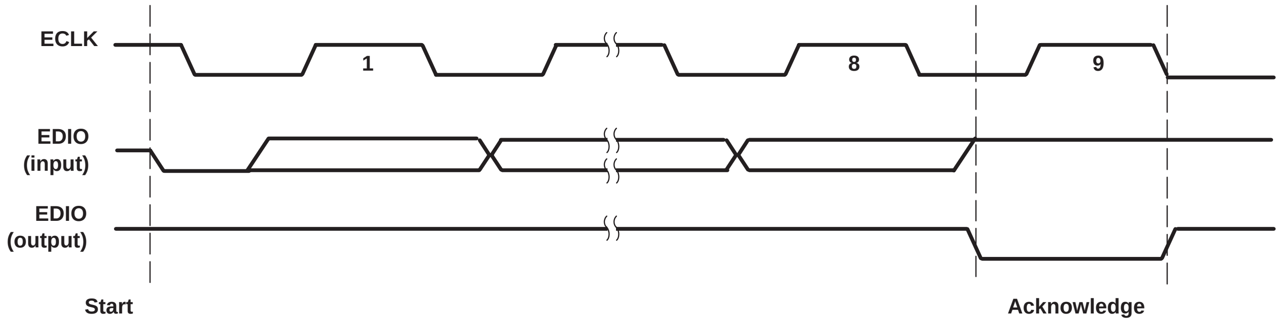
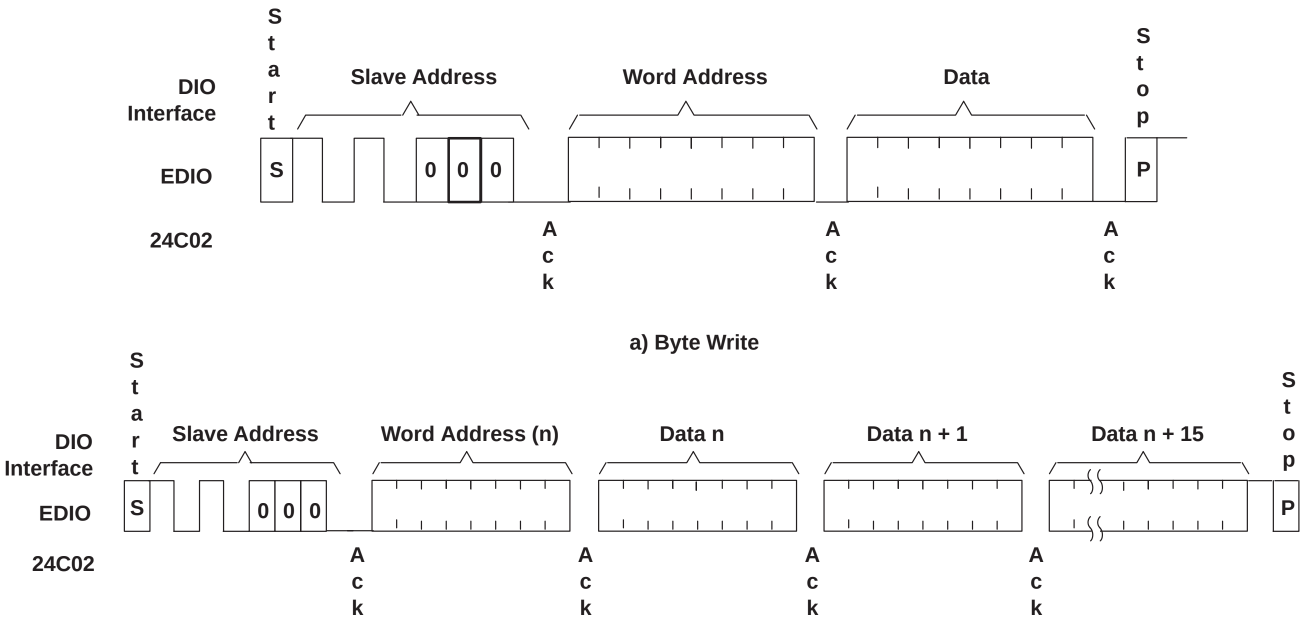


Figure 19. Acknowledge Response from Receiver

write cycle timing (continued)

When accessing the EEPROM through the DIO interface, the ECLOK and EDATA bits (SIOL/CTRL register at 0x00A1) are a direct reflection of the ECLK and EDIO terminals, respectively. To read and write to the EEPROM, these bits must be toggled properly to provide the proper requests and data.

A typical write operation is shown in Figure 20.



a) Byte Write

b) Page Write

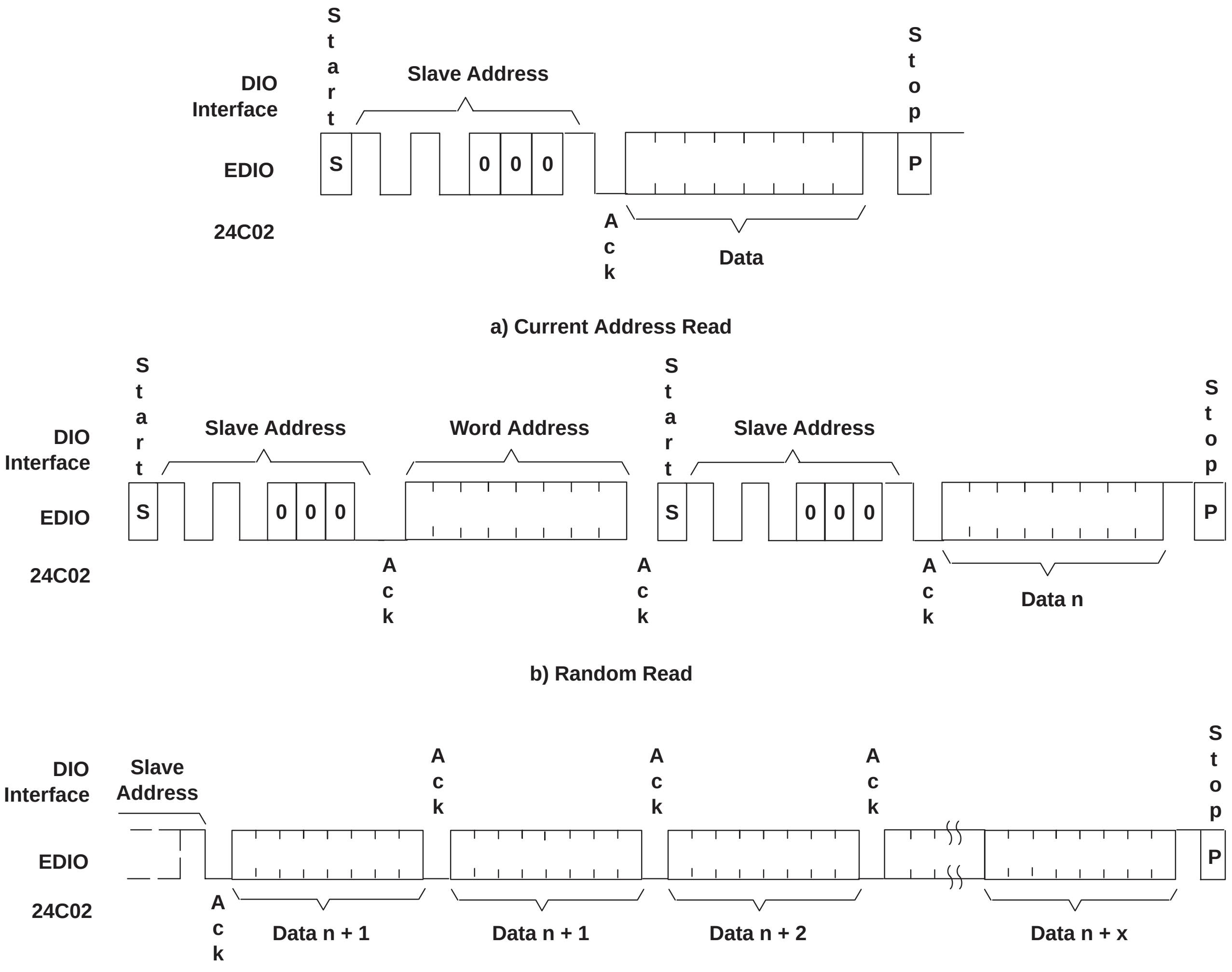
Figure 20. Typical Write Operation

read cycle timing

Read operations are initiated in the same manner as write operations, with the exception that the R/$\overline{W}$ bit of the slave address is set to a 1. There are three basic read operations: current address read, random read, and sequential read.

Typical read operations are shown in Figure 21.

read cycle timing (continued)



a) Current Address Read

b) Random Read

c) Sequential Read

Figure 21. Typical Read Operations

LED interface

LED interface allows a visual status for each port to be displayed (see Figure 22). The data supplied is multiplexed between port status and transmit queue congestion information. The data type is determined by the two strobe signals. Port status information is latched on the LEDSTR0 signal. Transmit queue congestion information is latched on the LEDSTR1 signal.

The LED port status output is driven low when the port state is suspended or disabled, except where suspension is caused by link loss. During normal operation, the output is high. The transmit queue congestion status is driven low when the transmit queue length has become negative for a port (indicating no further frames can be queued). For uncongested operation, the latched output is high. The $\overline{\text{LEDDATA}}$ is active low for simplified implementations using standard TTL parts. During hardware reset, the TNETX3150/TNETX3150A turns on the LEDs attached to it to provide indication that the TNETX3150/TNETX3150A is in reset. The LEDCLK is free running, LEDSTR0 and LEDSTR1 are held high, and $\overline{\text{LEDDATA}}$ is taken low.

LED interface (continued)

With LEDSTR0 and LEDSTR1 high, the LED latch is transparent and the LEDs are on.

When RESET is released, a normal LED cycle is initiated with $\overline{\text{LEDDATA}}$ held low and LEDSTR0 pulsed. The LEDs remain on until the START bit is set (an LED cycle updating LEDSTR1 is not performed).

When the START option bit is set, the TNETX3150/TNETX3150A initiates another LED cycle with $\overline{\text{LEDDATA}}$ held high, extinguishing the LEDs.

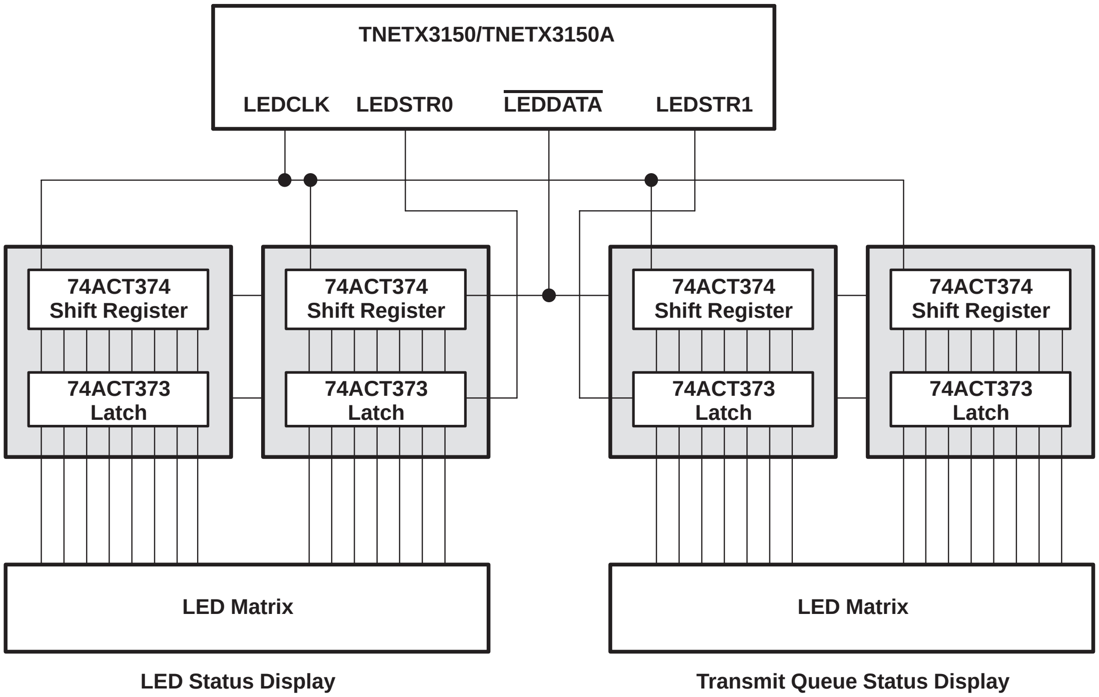


Figure 22. LED Interface

When a change is detected in the port status or transmit queue congestion status, the interface updates the LED data. Sixteen bits of status are shifted out serially at each update. The 16th bit is reserved. The LEDSTR0 or LEDSTR1 signal is pulsed once upon completion of the shift to latch the data. See the timing diagram in Figure 42.

JTAG interface

The TNETX3150/TNETX3150A is fully JTAG compliant with one exception; they do require external pullup resistors on the following terminals: TDI, TMS, and $\overline{\text{TRST}}$.

external pullup resistors

To implement internal pullup resistors, the TNETX3150/TNETX3150A would require the use of non-5V-tolerant input pads. The use of 5V-tolerant pads was deemed more important for mixed-voltage system boards than to integrate the required pullup resistors to be in strict compliance with the JTAG specification.

supported JTAG instructions

Mandatory: EXTEST, BYPASS and SAMPLE / PRELOAD
Optional public: HIGHZ and IDCODE

The opcodes for the various instructions (4-bit instruction register) are listed in Table 11.

Table 11. Opcodes for JTAG Instructions

INSTRUCTION TYPE	INSTRUCTION NAME	TNETX3150 JTAG OPCODE	TNETX3150A JTAG OPCODE
Mandatory	EXTEST	0000	000000
Mandatory	SAMPLE/PRELOAD	0001	000001
Optional	IDCODE	0100	000100
Optional	HIGHZ	0101	000101
Mandatory	BYPASS	1111	111111

The IDCODE for the TNETX3150/TNETX3150A is shown in Table 12.

Table 12. IDCODE Code

VARIANT	PART NUMBER	MANUFACTURE		LEAST-SIGNIFICANT BIT
bit 31 bit 28	bit 27 bit 12	bit 11	bit1	bit 0
0000	0000000000111000	00000010111		1

detailed description

FIFO

The FIFO RAM provides temporary storage of network data to allow burst transfers to and from the external DRAM, to allow network retries, and to run frame filtering.

The FIFO buffers the data between the MAC interfaces and QM block. There are independent FIFOs allocated for transmit and receive on each port. The depth of FIFO storage is 256 bytes per port for each TX and RX direction. Each port (256 bytes) is further subdivided into four 64-byte buffers. Each MAC access provides eight bytes of data and one byte of flag information. The flag information is generated by the MAC interfaces and passed with the data through the FIFO, providing useful status and control information through the TNETX3150/TNETX3150A.

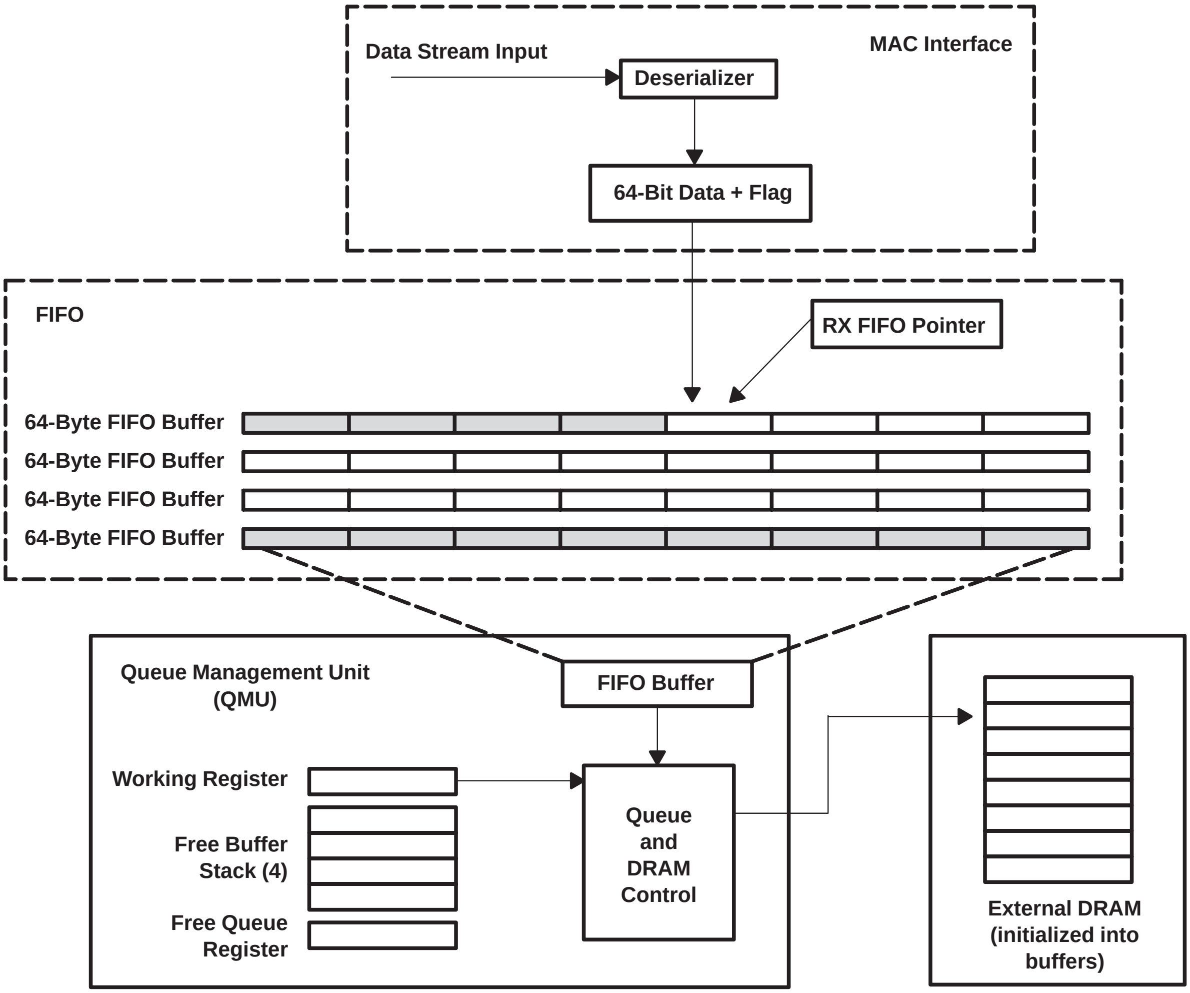


Figure 23. Data Flow From MAC Interface Through the TNETX3150/TNETX3150A to External DRAM

Each of the 31 FIFO blocks is subdivided into four buffers, each holding 64 bytes of data and eight bytes of flag information (see Figure 23). The flag byte records EOB information for the last buffer in a frame, where the buffer can be incompletely used.

FIFO (continued)

On data reception, when a FIFO buffer becomes full, the buffer is archived to the DRAM while the next buffer is received. Fast page access of the external DRAM enables efficient transfer. The queue manager uses the pointer from the working register to archive the buffer to external buffer memory. The working register value is then replaced by the next pointer in the free buffer stack. When all the pointers in the free buffer stack are used, the free queue register is loaded on demand with buffers from the free buffer queue.

If the FIFO becomes full and the external buffer memory also is full, subsequent frame data is lost and an error is logged. If this condition occurs, the health of the network at large is questionable (more data is entering than can leave the TNETX3150/TNETX3150A over a sustained period and the buffer depth is insufficient, resulting in storage overflow).

Diagrams showing the flow of normal frame data through the FIFO and the queue management unit (QMU) are shown in Figures 25 and 26.

QMU

The QMU comprises a number of tasks. At the top level, it provides an interface between the DRAM buffer memory and the on-chip FIFO. The queue manager uses internal 64-bit memory to maintain the status of all the queues. There are three queues associated with each port: The receive queue and transmit queue for store-and-forward operation, and the immediate queue for cut-through operation.

port structures

Internal registers are used to maintain the status of all the queues in external buffer memory. The internal register format is shown in Figure 24.

QUEUE STRUCTURE WORD FORMAT		
HEAD POINTER	TAIL POINTER	LENGTH
63 40	39 16	15 0

Figure 24. Queue Structure Word Format

The head pointer (bits 63–40) records the starting address of the queue in the buffer memory. The tail pointer (bits 39–16) records the last (or the tail) address of the queue. The TX length field (bits 15–0) is a residual length indicator and provides an indication of how many buffers are available to the queue. The number of buffers allocated to a queue at initialization depends on the size and configuration of the buffer memory. This can be stored in the EEPROM interface or written directly to the registers. For RX, the length recorded is the absolute number of buffers queued.

There are three queue types used by the TNETX3150/TNETX3150A per port:

- The receive queue collates buffer data for frames that cannot be cut-through to the destination port. All the frame data to be switched is collated on the appropriate receive queue. It is then concatenated to the end of the destination transmit queue. Concatenation entails the head pointer of the receive queue being placed in the forward pointer of the last buffer in the transmit queue. The length of the receive queue (number of buffers used) is subtracted from the number of free transmit queue buffers available. The tail pointer of the receive data becomes the new tail pointer for the transmit queue (there is one receive queue for every channel). If the destination port becomes idle and the frame (collated on the receive queue) can be cut-through, the receive queue is written to the immediate queue for transmission.
- The transmit queue stores complete frames that are ready for transmission. Once placed on the transmission queue, the data is transmitted. The transmit queues are not stalled pending the completion of receive data. The queues are stalled only if transmission cannot occur. There is one transmit queue for every channel.

port structures (continued)

- The immediate queue collates the cut-through-mode buffer information. If there is data queued to the immediate queue and the port is available, the data is transmitted. New frame data is placed onto the immediate queue only when the following conditions exist:
  - Data can cut through from source to destination.
  - The transmitter is currently idle on the destination port.
  - There is no existing frame transfer occurring on either transmit queue or immediate queue.

If the number of buffers in the buffer pool becomes less than or equal to zero, no further data is accepted. RX frame data is discarded until the free queue contains free buffers again. Additionally, individual queues can overflow, in particular the transmit queue. The transmit queue length is recorded as a residual figure (i.e., number of buffers remaining rather than number of buffers queued). If this becomes negative, no further frame data is queued and frames are discarded.

QM for cut-through operation (see Figure 25)

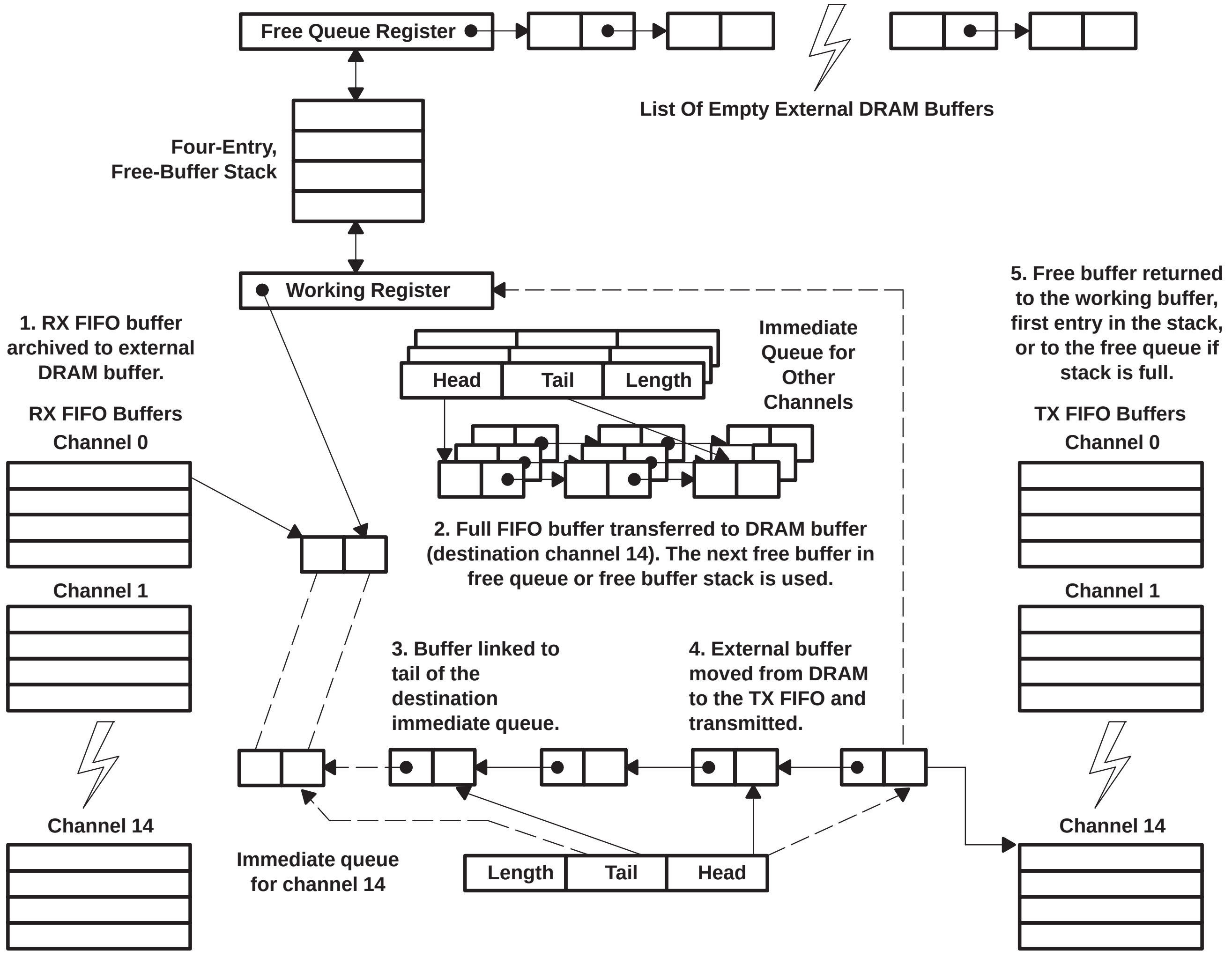

Free Queue Register
List Of Empty External DRAM Buffers
Four-Entry, Free-Buffer Stack
Working Register
5. Free buffer returned to the working buffer, first entry in the stack, or to the free queue if stack is full.
1. RX FIFO buffer archived to external DRAM buffer.
Immediate Queue for Other Channels
Head
Tail
Length
RX FIFO Buffers
Channel 0
TX FIFO Buffers
Channel 0
2. Full FIFO buffer transferred to DRAM buffer (destination channel 14). The next free buffer in free queue or free buffer stack is used.
Channel 1
Channel 1
3. Buffer linked to tail of the destination immediate queue.
4. External buffer moved from DRAM to the TX FIFO and transmitted.
Channel 14
Channel 14
Immediate queue for channel 14
Length
Tail
Head

Figure 25. QM for Cut-Through

port structures (continued)

QM for store-and-forward operation (see Figure 26)

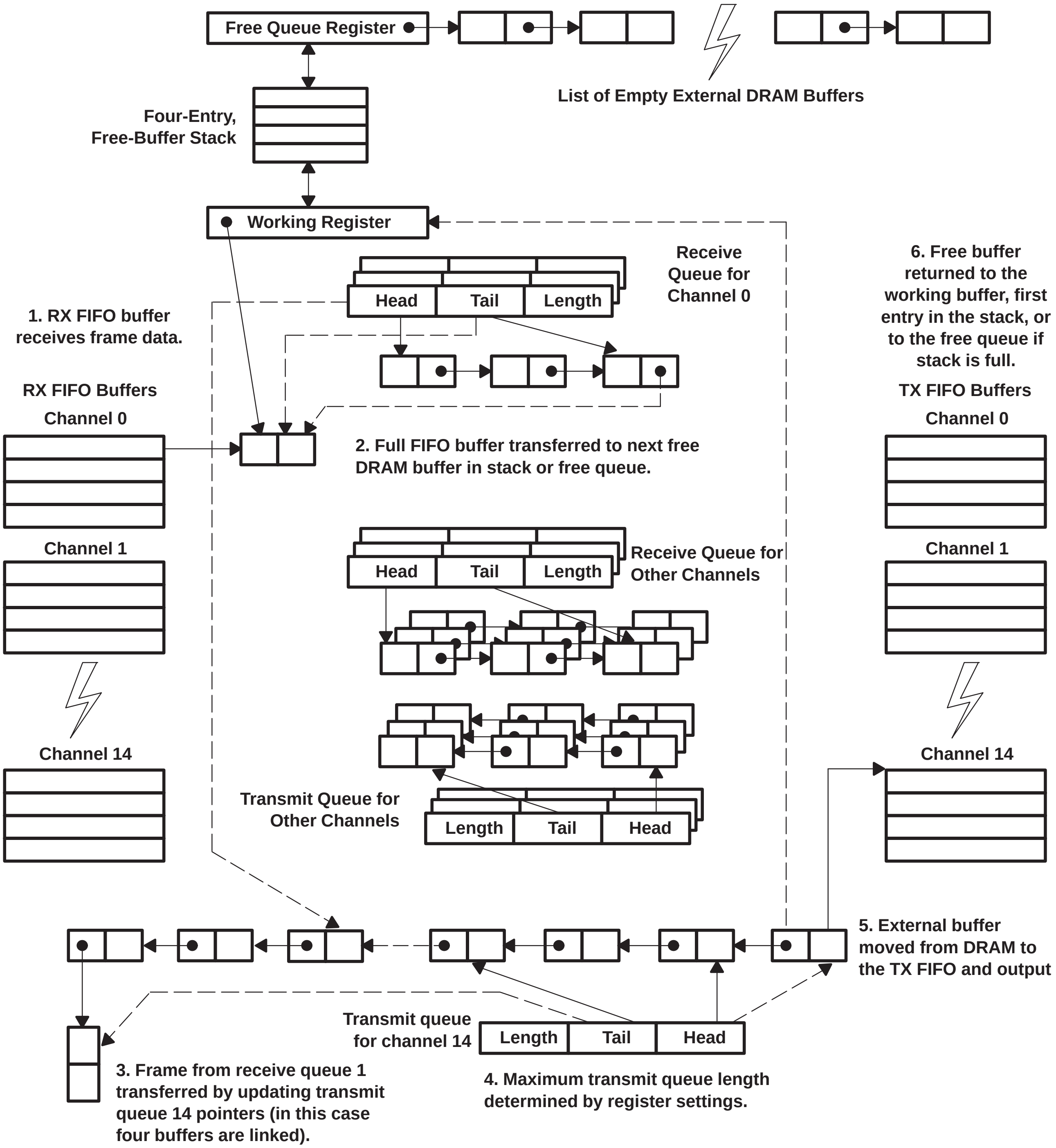


Figure 26. Queue Management for Store and Forward

buffer allocation

Each buffer is capable of holding the complete contents of one of the internal FIFO buffers (the minimum-size Ethernet frame). The buffers are aligned to fit within a DRAM page. No buffer crosses a page boundary, allowing consistent access times to be attained at the expense of losing four words per 1K page boundary (each word is 36 bits wide). The memory, organized in this way, permits fast data bursts between the internal FIFO and external buffer memory. This reduces the amount of intermediate data management, which reduces the internal bandwidth.

At initialization, the TNETX3150/TNETX3150A loads configuration information from the EEPROM (if present) or uses its default reset values to set the length field for each of the queues unless initialized by DIO access. This fixes the maximum number of buffers that a port can use for transmit queues. As buffers are used by these queues, the length field is adjusted to indicate the number of buffers that are still allocated for use by that particular queue (see Figure 27).

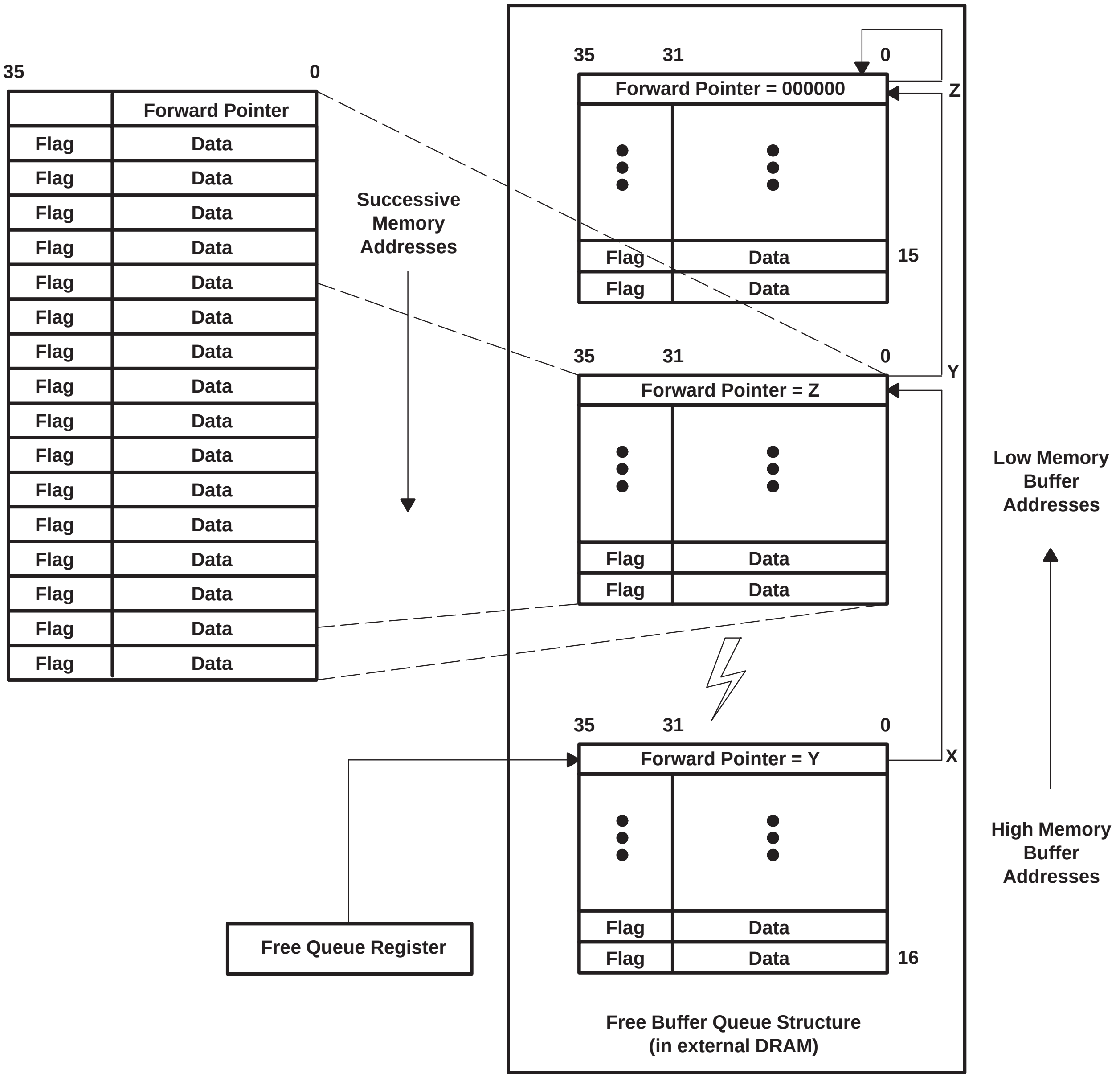

35
0
Forward Pointer
Flag
Data
Successive Memory Addresses
35
31
0
Forward Pointer = 000000
Z
Flag
Data
15
Forward Pointer = Z
Y
Low Memory Buffer Addresses
Forward Pointer = Y
X
High Memory Buffer Addresses
16
Free Queue Register
Free Buffer Queue Structure (in external DRAM)

Figure 27. Buffer Allocation

buffer allocation (continued)

The total number of buffers available to the TNETX3150/TNETX3150A is determined by the size of the external memory. The RSIZE (RAM size) field of the RAM size register is loaded from the EEPROM or from the DIO interface with the appropriate system RAM code. The TNETX3150/TNETX3150A uses this size information to modify the DRAM addressing limit when initializing the buffer data structures. The DRAM is initialized to contain a single list of data buffers (free buffer queue) available to all queues. The least-significant byte of the DRAM address is incremented in steps of 17 (decimal). During initialization, the normal TNETX3150/TNETX3150A operation is disabled. Once the buffer structure has been created in the DRAM, no further use is made of the sizing information.

The queue size for the transmit queues can be increased by adding a 2’s complement number (representing the number of buffers that need to be added to the queue) to the transmit queue-length field. Reducing the number of buffers allocated to the ports is done in the same way by adding a negative-length field. The length is updated after the transmission of a buffer. The update bit is cleared after the update occurs.

There is no checking between the number of free buffers physically available in memory and the number of buffers allocated to each queue. It is possible to oversubscribe the memory between the queues. If a frame is being buffered when the buffer ceiling is reached, all buffers constituting that incomplete queue of buffers are purged and replaced on the free buffer stack or queue. Thus, during memory limitation, large frames are inherently filtered in favor of smaller frames. When all buffers are subscribed and none are available for use, the TNETX3150/TNETX3150A accepts no new frames, but waits for buffers to be freed before continuing.

QMU – DRAM controller

Within the queue manager, the DRAM control block provides the interface to the external DRAM buffer memory. The interface control signals required are produced by the QMU, which controls the data transfer with the DRAM.

The interface relies on the use of EDO DRAM to minimize the access time, while maintaining RAM bandwidth. The TNETX3150/TNETX3150A requires EDO DRAM operating at 60 ns. The use of EDO DRAM permits the high data-transfer rates required by the TNETX3150/TNETX3150A.

The DRAM is accessed in a number of ways:

1. Single access – used during initialization and forward pointer writes. This is the slowest-access method and transfers a single 36-bit word. Each access takes seven 20-ns clock cycles (see Figures 2 and 4).
2. Page-mode burst access – used for fast data transfer of one 64-byte buffer from the FIFO RAM to the DRAMs. The locations used are located within the DRAMs page boundary, permitting fast burst accesses to be made. Each successive burst access requires only two clock cycles after the initial row address has been loaded (see Figure 9).

Dynamic memories must be refreshed periodically to prevent data loss. Each row refresh cycle requires a minimum of seven clock cycles and must be performed such that the whole device is refreshed every 16 ms. A normal read or write operation refreshes the whole row being accessed (see Figure 5).

The external DRAM data bus is 36 bits wide. Buffer data is accessed over two memory cycles from the DRAM before it is concatenated into an 8-byte data word and one byte of flag data. The format of the 36-bit data word used is shown in Figure 28.

DRAM DATA BUS FORMAT			
FLAG DATA		32-BIT DATA	
35	32	31	0

Figure 28. DRAM Data Bus Format

QMU – DRAM controller (continued)

The address lines are arranged to permit a wide range of memory sizes to be connected to a maximum of 22 address lines. The address lines are organized as shown in Table 13.

Table 13. Address-Line Organization

TERMINAL NAME	DX2	DX1	DX0	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Address bit valid during $\overline{\text{DRAS}}$	21	19	17	15	14	13	12	11	10	9	8
Address bit valid during $\overline{\text{DCAS}}$	20	18	16	7	6	5	4	3	2	1	0

This prohibits the alignment of buffers to cross a page boundary, reducing buffer management complexity and maintaining a high bandwidth.

address compare unit

The address compare unit provides the switching information required to route the data packets. The source and destination Ethernet addresses are examined. Source addresses are used to determine the port's address. Destination addresses are used to determine the packet destination. If a match is found, the appropriate destination channel address is generated and provided to the other internal components.

Each channel (except the uplink port) has an address compare register (see Figure 29). Each register holds a 48-bit Ethernet address. The Ethernet source address is taken from a received frame and assigned to the channel on which it was received. This occurs for each frame received. The destination address is compared to the address registers. If matched, the channel address for the port is assigned. If no match is found for the destination address, the frame is handled according to the ADRDIS and BRUN option bits. If BRUN (broadcast to unassigned) is set, an unmatched frame is broadcast to all ports with unassigned addresses (requires IOBMOD bit to be set). If BRUN is reset, the frame is handled according to the ADRDIS bit for the uplink (port 00). If the ADRDIS bit for port 00 is set, the unmatched frame is discarded. If reset, the unmatched frame is transmitted from the uplink.

The address compare registers learn their Ethernet address, used for comparison, from the source address of a received frame. The port address registers can be fully accessed via the DIO interface. This allows ports to be set up and secured under management control, or port address monitored. For further details, see the description of the TNETX3150/TNETX3150A registers.

As the frame is loaded the source address is compared against the source address attributed to that port. If the address has changed and the port address acquired by the TNETX3150/TNETX3150A was secured, the port may be suspended or disabled and an error logged. During this comparison, it is possible to detect multiple entries of the same address in the compare unit. This also is an error, and it is meaningless to have the same address applied to multiple ports.

If external address matching logic is not used, ports (1–14) must be confined to desktop (single address) and not multiple-address switching. The uplink (port 00) supports multiple-address switching.

address compare unit (continued)

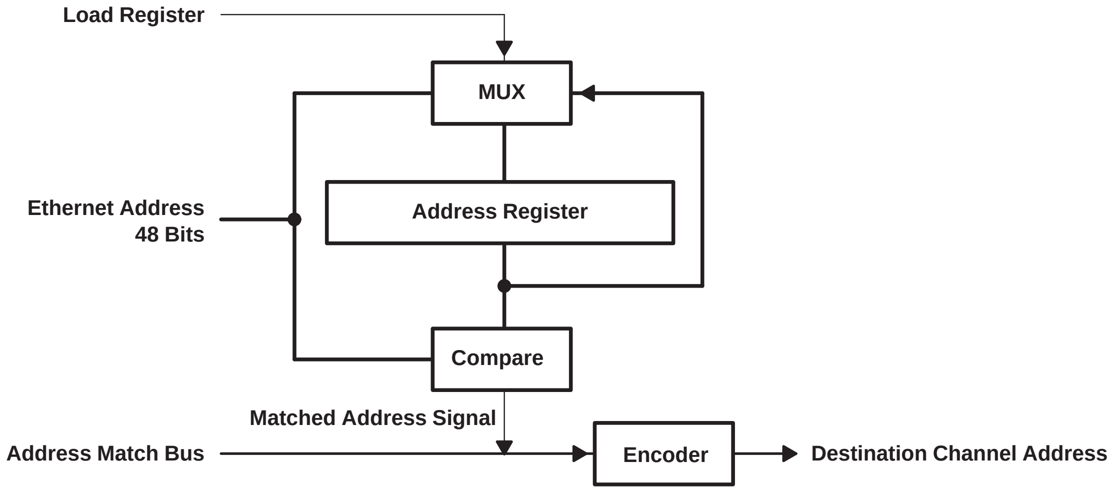


Figure 29. Address Compare Logic

port address register configuration

The port addresses can be assigned explicitly or dynamically. An address is explicitly assigned by writing it to the port address registers via the DIO interface. An address is assigned dynamically by the TNETX3150/TNETX3150A hardware, loading the register from the source address field of the received frames. If the port is in secured mode, the address is loaded only once from the first frame. In unsecured mode, the address is updated on every frame received.

The uplink port (port 00) does not have a port address register. If it receives a frame with a source address securely assigned to another port, the uplink either disables or continues to operate, depending on the SECDIS bit. When SECDIS is set, the port disables on receipt of a securely assigned address. When SECDIS is reset, the port continues to operate. Port suspension, due to address mismatch, is not supported on the uplink. Since no port address exists, suspension on a multiple-address port is not desirable.

IOB operation

This mode of handling broadcast traffic is selected by setting the IOBMOD bit in the system control register. IOB handling ensures that frames that are broadcast follow the strict order in which they were received.

When a multicast frame is detected in IOB mode, the reception continues in the same manner as for a normal store-and-forward frame. The buffers comprising the frame are linked in the receive queue.

When the end of frame is detected, an additional buffer is linked to the end of the receive queue. This buffer is similar in size to a normal data buffer, but contains indexed queue information rather than frame data. To distinguish between the types of buffers, bit 23 of the tag field is set.

IOB operation (continued)

When all multicast buffers have been linked onto the receive queue, the receive queue is linked to the transmit queues on which the multicast data is to be transmitted. The ports to which the data is sent can be defined two ways. If no external addressing logic is used, the multicast data is linked to all active ports. These are defined in the port bit map held in the VLAN register for the port on which the data was received. Alternatively, the port bit map presented on the EAM interface terminals is used. The data is linked to the active port subset of that defined on the terminals.

Having determined the list of transmit queues onto which the IOB data is linked, the forward pointer for each transmit queue is updated to point to the head of the receive queue (IOB data). In this way, the multicast data buffers appear linked to multiple queues without the program of replicating the multicast data. The index buffer is used to preserve the separate transmit queues as they form, following the IOB data frame. Each index buffer contains a forward pointer referencing the continuation of the transmit queue for every port. As new transmit queue data is queued, the forward pointers in the index buffer are updated to reflect the continuation of the independent transmit queues (see Figure 30).

The IOB frame buffers can be returned to the free buffer queue only when all ports have transmitted the IOB data. Since there can be a large discrepancy between the first port completing transmission and the last (due to a long transmit queue before the IOB data), a tag field is used to record which ports have transmitted the IOB data from the list of ports where the data was to be sent originally. The tag field also is stored in the index buffer. When the last port tag is cleared, all the buffers can be returned to the free pool of buffers.

The buffers can be freed only after the last transmission, by which stage the forward pointer pointing to the head of the IOB buffers is freed. The return address field of the index buffer is used to store the head address of the IOB buffers. Thus, even after the last IOB transmission, the head of the IOB buffers remains known. Freeing the buffers is done by writing the pointer at the top of the free queue to the last forward pointer of the IOB buffers and moving the return address to the top of the free queue, adding the used IOB buffers to the top of the free queue.

IOB operation (continued)

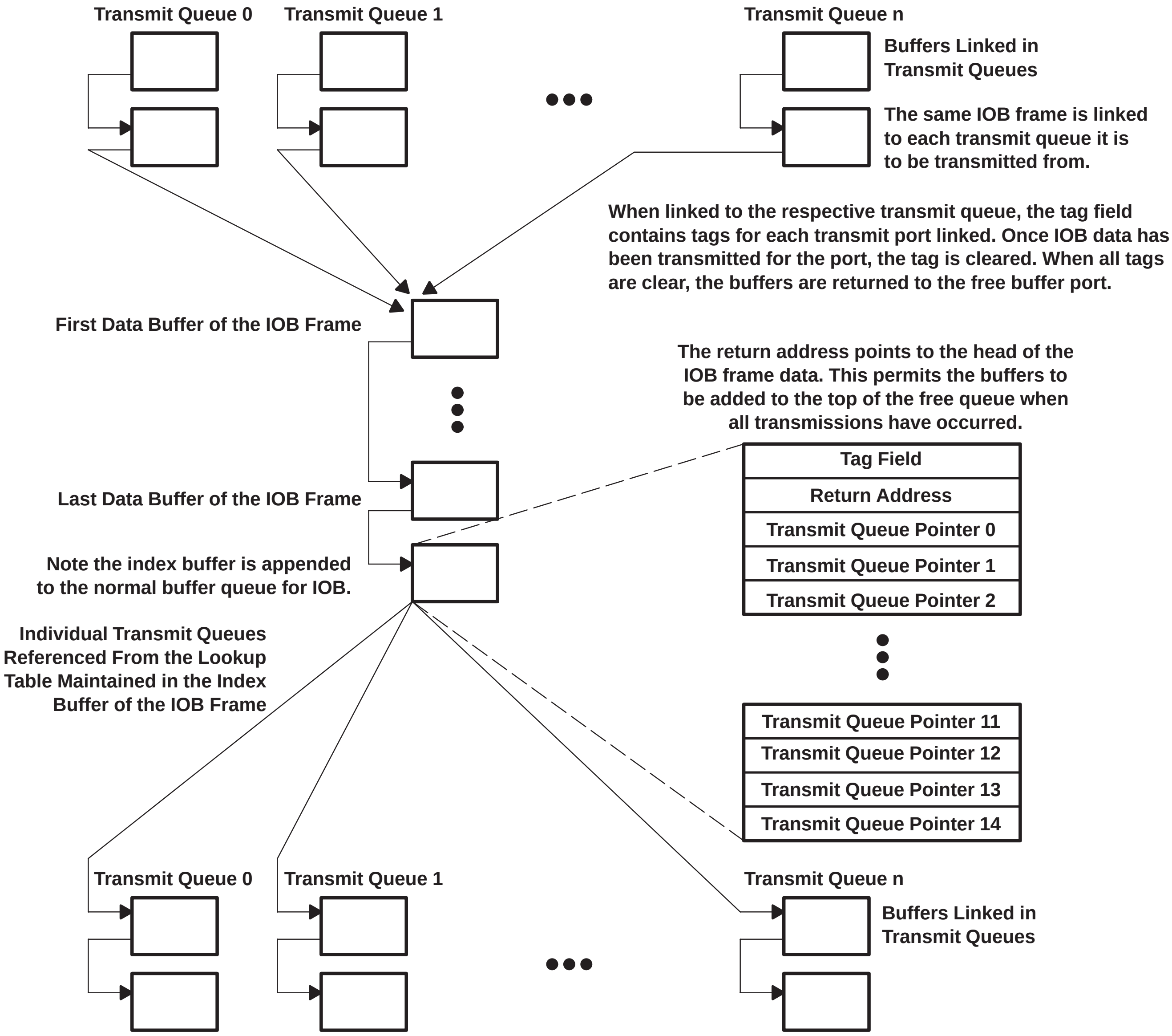

Transmit Queue 0
Transmit Queue 1
Transmit Queue n
Buffers Linked in Transmit Queues
The same IOB frame is linked to each transmit queue it is to be transmitted from.
When linked to the respective transmit queue, the tag field contains tags for each transmit port linked. Once IOB data has been transmitted for the port, the tag is cleared. When all tags are clear, the buffers are returned to the free buffer port.
First Data Buffer of the IOB Frame
The return address points to the head of the IOB frame data. This permits the buffers to be added to the top of the free queue when all transmissions have occurred.
Tag Field
Return Address
Transmit Queue Pointer 0
Transmit Queue Pointer 1
Transmit Queue Pointer 2
Last Data Buffer of the IOB Frame
Note the index buffer is appended to the normal buffer queue for IOB.
Individual Transmit Queues Referenced From the Lookup Table Maintained in the Index Buffer of the IOB Frame
Transmit Queue Pointer 11
Transmit Queue Pointer 12
Transmit Queue Pointer 13
Transmit Queue Pointer 14
Transmit Queue 0
Transmit Queue 1
Transmit Queue n
Buffers Linked in Transmit Queues

Figure 30. IOB Operation

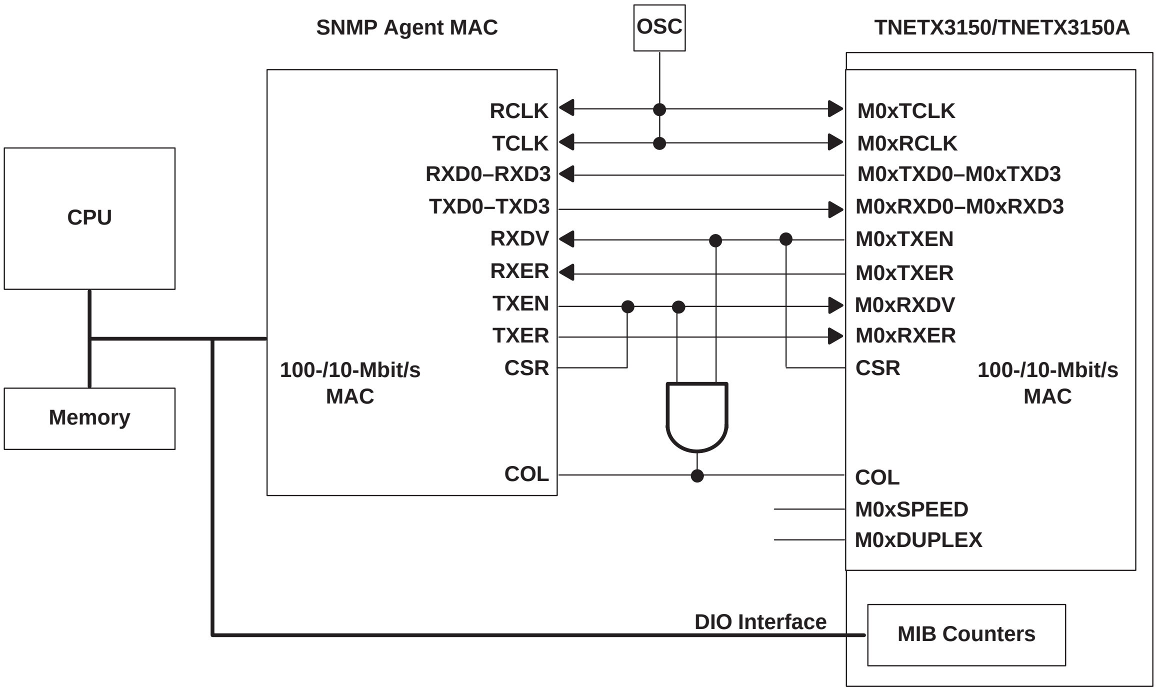
frame echoing

If a frame enters on a port whose address matches the destination address of the frame, the frame is not echoed back on the port. A general rule is that no frame is echoed back to the port on which it was received. If frame routing is being performed by the EAM interface, it is the user's responsibility to enforce this since it is not enforced by the TNETX3150/TNETX3150A.

SNMP

A CPU through an Ethernet MAC or suitable protocol translating device can be directly connected to one of the TNETX3150/TNETX3150A ports for use with SNMP. See Figure 31.

SNMP Support



MODE	OSC	SPEED	DUPLEX
100 Mbit/s	25 MHz	1	Dependent on external MAC
10 Mbit/s	2.5 MHz	0	Dependent on external MAC

Figure 31. SNMP Support Application

SNMP support requires an interface to a CPU and CPU-controlled software. The speed and duplex terminals must be configured correctly to ensure proper operation. This is a proposed schematic and has not been tested.

manufacturing test features

The methodology of these tests is considered before implementation. All tests are based on an incremental approach, building upon tested results before reaching the final goal. For tests using the DIO interface (for example), the DIO interface should be tested and then, if it is functioning correctly, the next depth of testing should be performed (i.e., internal to the TNETX3150/TNETX3150A testing). If a test fails using this methodology, the cause of the failure can be determined quickly and test/debug time reduced.

primary test access: DIO testing

The DIO registers can be written to and read directly from the terminal interface. This level of testing is trivial, but essential, before continuing to internally test the TNETX3150/TNETX3150A.

secondary test access: internal RAM access modes

When implementing an architecture that employs embedded RAM structures, it is necessary to ensure test access over and above JTAG connectivity testing through standard interfacing. The DIO interface used by the TNETX3150/TNETX3150A enables the user to interrogate the internal RAMS of the TNETX3150/TNETX3150A. User interrogation gives the required observability for the RAM and the data they contain.

RAM test access is desirable at all levels of testing:

- Silicon production level that enables detection of defective devices
- System production level that permits diagnostic testing
- Field level that permits diagnostics and debugging

FIFO RAM test access

FIFO RAM access for test is provided through the DIO interface. This allows full RAM access for RAM testing purposes. Access to the FIFO shall be allowed only following a soft reset and before the START bit is written (or after power up and before the START bit is written). The soft reset bit should be set, then immediately reset. If the soft reset bit is not cleared, the TNETX3150/TNETX3150A holds the DRAM refresh logic operation in reset and the contents of the external memory becomes invalid.

To access the FIFO RAM from the DIO, bytes are written to a holding latch that is the width of the RAM word (72 bits). When a byte is accessed, the whole word is updated in RAM. If the same pattern is to be loaded throughout the memory, it requires only a new FIFO RAM address for setup between accesses on a single byte within the word. The data in the latch does not change (i.e., a read /modify/write is not performed).

structure (statistics) RAM test access

Test access to the statistics RAM is provided through the DIO port after the TNETX3150/TNETX3150A is soft reset (or following power up before the start bit is set). In this mode, all locations of the RAM can be written to and read from. Once the start bit is set, only read access is permitted to the RAM. When asserting soft reset, it is important to reset the soft reset bit immediately after setting it. This ensures that the DRAM refresh logic operation is not held at reset. If held at reset, normal DRAM refreshes fail to occur, and the DRAM contents become invalid.

To access the structure RAM from the DIO, bytes are written to a holding latch that is the width of the RAM word (64 bits). When a byte is accessed, the whole word is updated in RAM. If the same pattern is loaded throughout the memory, it requires only a new structure RAM address for setup between accesses on a single byte within the word. The data in the latch does not change (i.e., a read/modify/write is not performed).

internal frame-wrap test mode

Frame-wrap mode allows the user to send a frame into a designated source port and selectively route the frame successively to and from ports involved in the test (or return the frame directly) before retransmitting the frame on the designated source port. By varying the number of ports between which the frame is forwarded, the potential fault-capture area can be expanded or constrained. Initially, it is desirable to send data to and from each port in turn, allowing the MAC-to-FIFO interface and MAC terminals to be tested for each port.

The TNETX3150/TNETX3150A provides an internal-loopback test mode. Internal loopback allows the frame data path to be tested, and is useful for individual die burn-in testing and system testing with minimal reliance on external parts. Internal loopback is selected by suitably setting the INTWRAP field of the DIATST register (see DIATST register at 0x00DD). Port 00 (uplink), port 02, or port 14 can be selected as the source port for injecting frames into the TNETX3150/TNETX3150A when internal wrap is selected. All other ports are set to internally wrap frames (see Figure 32).

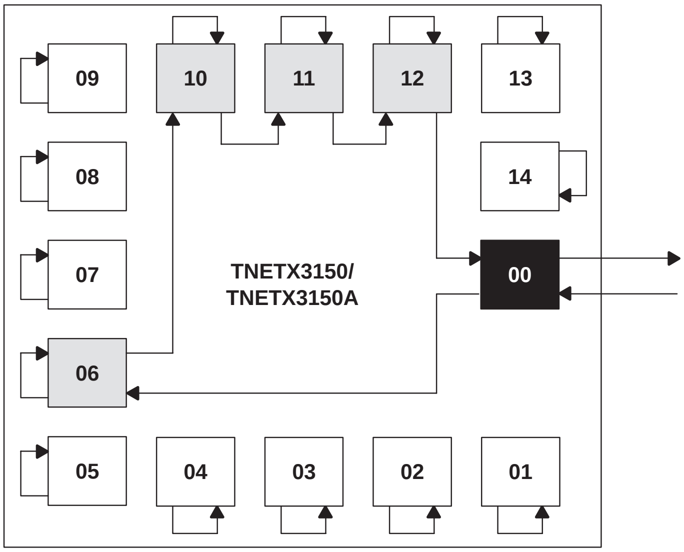


Figure 32. Internal Frame-Wrap Test (wrapped between multiple ports using broadcast frames and VLAN registers)

By injecting broadcast or multicast frames into the source port and suitably setting the VLAN registers, frames can be forwarded between internally wrapped ports before transmission of the frame on the source port.

The operational status of the PHY (or external connections to the TNETX3150/TNETX3150A) do not have to be considered or assumed good when in internal-loopback mode.

external frame-wrap test mode

Similar to internal frame-wrap mode, the ports can be set to accept frame data that is wrapped at the PHY. This permits network connections between the TNETX3150/TNETX3150A and the PHY to be verified. All frame data for external frame-wrap testing must be received by port 00 (uplink). By using multicast/broadcast frames, traffic is routed selectively between ports involved in the test (or return the frame directly) before retransmission on port 00 (uplink) (see Figure 33).

External frame-wrap test mode is selected by setting the DPWRAP bit (bit 3) of the DIATST register. When selected, the port is forced to receive the frame it transmits to the PHY (see DIATST register at 0x00DD).

By using broadcast or multicast frames and suitably setting the VLAN registers, frames are forwarded between internally frame-wrapped ports before transmission of the frame on the source port.

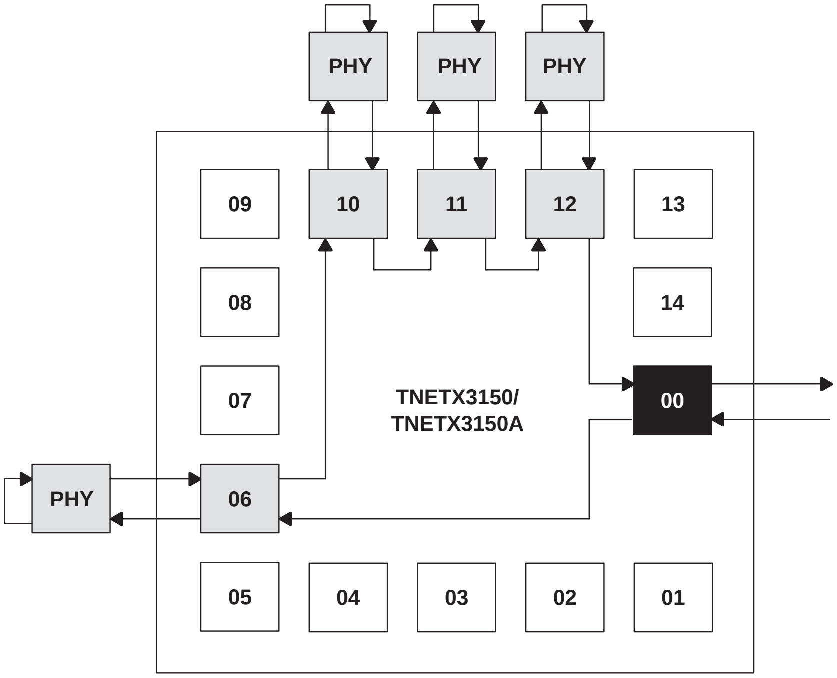


Figure 33. External Frame-Wrap Test (wrapped between multiple ports using broadcast frames and VLAN registers)

internal and external frame-wrap example

Perform the following steps to configure the TNETX3150/TNETX3150A for the internal frame-wrap configuration:

NOTE:
No traffic should be received by the TNETX3150/TNETX3150A at this time; internal address matching must be enabled and an external address matching device must be disabled.

Step 1. With power on, place the TNETX3150/TNETX3150A in a reset state by writing 0x81 at address 0x00C3.

internal and external frame-wrap example (continued)

Step 2. While the TNETX3150/TNETX3150A is in reset, configure the VLAN mask registers (0x00A4–0x00C1). For this example, all ports are in the wrap chain.

VLAN0 MASK	=	0x0002
VLAN1 MASK	=	0x0004
VLAN2 MASK	=	0x0008
VLAN3 MASK	=	0x0010
VLAN4 MASK	=	0x0020
VLAN5 MASK	=	0x0040
VLAN6 MASK	=	0x0080
VLAN7 MASK	=	0x0100
VLAN8 MASK	=	0x0200
VLAN9 MASK	=	0x0400
VLAN10 MASK	=	0x0800
VLAN11 MASK	=	0x1000
VLAN12 MASK	=	0x2000
VLAN13 MASK	=	0x4000
VLAN14 MASK	=	0x0001

Step 3. Start the TNETX3150/TNETX3150A by writing a 1 to the start bit and ensure that the IOBMOD bit is also 1 (0x21 at address 0x00C3).

Step 4. Set all ports into full duplex and enable all ports.

Step 5. Set the BRUN bit to zero (address 0x00A1).

Step 6. Configure the TNETX3150/TNETX3150A for test mode by writing a 1 to the MTEST bit (address 0x00C2).

Step 7. Configure the TNETX3150/TNETX3150A for internal wrap mode by writing a 1 to the INTWRAP. The port that is not wrapped (00, 02, or 14) is used to inject and observe test data frames from internally wrapped ports. For port 00, write 0x02 at 14, write 0x00DD. For port 02, write 0x04 at address 0x00DD. For port 14, write 0x06 at address 0x00DD.

Step 8. Transmit a broadcast/multicast frame into the TNETX3150/TNETX3150A port corresponding to the INTWRAP setting. Whatever is sent into the test port is received by the same port (if the VLAN register settings steered the packet back to that port). Compare the frame transmitted against the frame received and verify that the TNETX3150/TNETX3150A is working correctly.

For external wrap mode, replace steps 7 and 8 with the following:

Step 7. Configure the TNETX3150/TNETX3150A for external wrap mode by writing 0x08 at address 0x00DD.

Step 8. Configure all PHY-layer devices except port 00 for internal loopback.

Step 9. Transmit a broadcast/multicast frame into the TNETX3150/TNETX3150A on port 00. Whatever is sent into the test port is received by the same port (if the VLAN register settings steer the packet back to that port). Compare the frame transmitted against the frame received and verify that the TNETX3150/TNETX3150A is working correctly.

tertiary test access

The internal RAM access infers only that both DIO port and internal RAM structures are functioning correctly. It does not provide information on the TNETX3150/TNETX3150A data paths (to and from the RAMS) during normal frame operations or an indication of the control path functionality. To further assist with this, the proposed tests are as follows:

- DRAM access — this test proves that the data path between FIFO and DRAM is functioning, along with certain sections of the queue manager and FIFO logic operations.
- Frame forwarding — frame data is forwarded from one port to the next using a loop-back mode. This builds on the previous tests, and tests that the data path to and from the protocol handlers and control paths are operational. The number of ports that take part in forwarding is controlled using the VLAN registers, allowing any number of ports to be tested in this mode. Single connections can be tested allowing individual protocol-handler data paths-to-FIFO connections to be tested or multiple port testing that allows reduced system test time.

external DRAM test access

Using the incremental test approach (after the FIFO is tested and verified), the data path to and control of the external DRAM memory is verified.

DRAM writes are carried out by first constructing a buffer in the FIFO (64 bytes), and then initiating a buffer write from the FIFO to the DRAM. The buffer is transferred like a normal buffer transfer in a 17-write DRAM burst. The forward pointer field is mapped to the DRAM data register, and the flag data fields are mapped to the DRAM flag register.

Reading from the DRAM performs a buffer transfer to the FIFO from which individual bytes can be read (and tested) through the DIO interface. The flag bytes and forward-pointer bytes are transferred from the DRAM to the DRAM-data and DRAM-flag registers for reading.

The buffer transfer mechanism, when operated in DRAM test access mode, does not check the flag status. No actions are performed, depending on the status of the flags. The transfer is a test data transfer, with no attempt made to comprehend flag contents.

After completion of the DRAM testing, the TNETX3150/TNETX3150A should be reset before normal switching activity is resumed. This ensures that the TNETX3150/TNETX3150A is returned to a defined state before normal functionality is resumed. This mechanism is primarily intended for DRAM testing and is not a part of a breakpoint/debug mechanism.

For more information refer to the *test register* section.

glossary

address duplications

The number of address duplications between a securely assigned port address within the TNETX3150/TNETX3150A and a source address observed on this port. Occurrence of this causes the TNETX3150/TNETX3150A to suspend the port (see *port status register description*).

address mismatches/address changes

The sum of:

1. The number of mismatches seen on a port between a securely assigned port address and the source address observed on the port. Occurrence of this causes the TNETX3150/TNETX3150A to suspend the port (see *port status register description*).
2. The number of times the TNETX3150/TNETX3150A is required to assign or learn an address for a port.

Port statistics memory region 0x780–0x7FF.

broadcast RX frames

The total number of good packets received that were directed to the broadcast address. This does not include multicast packets.

broadcast TX frames

The total number of packets transmitted that were directed to the broadcast address. This does not include multicast packets. See IEEE Std 802.3 Supplements, Layer Management (section 5) 5.2.2.1.1, number 6.

carrier sense errors

The number of times the carrier sense condition was lost or never asserted when attempting to transmit a frame on a particular interface. The count is incremented a maximum of once per transmission attempt, even if the carrier sense condition fluctuates during a transmission attempt.

collisions port (03–14)

The number of times the port’s transmitter was required to send a jam sequence

The following counters are implemented in previously described counters.

deferred TX frames

A count of the frames for which the first transmission attempt on a particular interface was delayed because the medium was busy

excessive collisions

A count of frames for which transmission on a particular interface fails due to excessive collisions

filtered RX frames

The count of frames received but discarded due to lack of resources (TXQ full, destination disabled, or RX errors). The number of frames sent to the TNETX3150/TNETX3150A discard channel.

good RX frames

The total number of good packets (including unicast, broadcast packets and multicast packets) received

good TX frames

The total number of packets (including bad packets, broadcast packets, and multicast packets) transmitted successfully. See IEEE Std 802.3 Supplements, Layer Management (section 5) 5.2.2.1.1, number 1.

glossary (continued)

late collisions

The number of times that a collision is detected on a particular interface later than 512-bit times into the transmission of a packet

multicast RX frames

The total number of good packets received that were directed to the multicast address. This does not include packets directed to the broadcast address.

For the 100-Mbit/s ports, the counter records the sum of alignment errors and code errors (frame received with RX error signal).

multicast TX frames

The total number of packets transmitted that were directed to a multicast address. This number does not include packets directed to the broadcast address. See IEEE Std 802.3 Supplements, Layer Management (section 5) 5.2.2.1.1, number 7.

multiple-collision TX frames

A count of successfully transmitted frames on a particular interface for which transmission is inhibited by more than one collision

net octets

The total number of octets of data (including those in bad packets) received on the network (excluding framing bits but including frame check sequence (FCS) octets). This object can be used as a reasonable indication of Ethernet utilization.

oversize RX frames

The total number of packets received that were longer than 1518 octets (excluding framing bits, but including FCS octets) and were otherwise well formed. If SYSCTRL option bit LONG is set, the number is 1536 octets.

RX align/code errors

For the 10-Mbit/s ports, the counter records alignment errors.

RX and TX frame 64

The total number of packets (including bad packets) received and transmitted that were 64 octets in length (excluding framing bits but including FCS octets)

RX and TX frames 65–127

The total number of packets (including bad packets) received and transmitted that were between 65 and 127 octets in length inclusive (excluding framing bits but including FCS octets)

RX and TX frames 128–255

The total number of packets (including bad packets) received and transmitted that were between 128 and 255 octets in length inclusive (excluding framing bits but including FCS octets)

RX and TX frames 256–511

The total number of packets (including bad packets) received and transmitted that were between 256 and 511 octets in length inclusive (excluding framing bits but including FCS octets)

RX and TX frames 512–1023

The total number of packets (including bad packets) received and transmitted that were between 512 and 1023 octets in length inclusive (excluding framing bits but including FCS octets)

glossary (continued)

RX and TX frames 1024–1518

The total number of packets (including bad packets) received and transmitted that were between 1024 and 1518 octets in length inclusive (excluding framing bits but including FCS octets). If the LONG bit is set, this statistic counts frames that are between 1024 and 1536 octets in length inclusive (excluding framing bits but including FCS octets).

RX CRC errors

A count of frames received on a particular interface that is an integral number of octets in length but does not pass the FCS check

RX H/W errors

The function of this counter is performed by the filtered RX frames counter.

RX fragments

The total number of packets received that were less than 64 octets in length (excluding framing bits, but including FCS octets) and had either a bad FCS with an integral number of octets (FCS error) or a bad FCS with a non-integral number of octets (alignment error).

RX jabbers

The total number of packets received that were longer than 1518 octets (excluding framing bits, but including FCS octets), and had either a bad FCS with an integral number of octets (FCS error) or a bad FCS with a non-integral number of octets (alignment error); 1536 octets if SYSCTRL option bit LONG is set.

RX octets

This contains a count of data and padding octets in frames that were successfully received. This does not include octets in frames received with frame-too-long, FCS, length or alignment errors. (IEEE Std 802.3 Supplements, Layer Management (section 5) 5.2.2.1.3, number 2).

RX overruns port (00–14)

The number of frames lost due to a lack of resources during frame reception. This counter is incremented when frame data cannot enter the RX FIFO for whatever reason. Frames that overrun after entering the FIFO also can be counted as RX discards if they are not cut-through.

single-collision TX frames

A count of successfully transmitted frames on a particular interface for which transmission is inhibited by exactly one collision

signal quality error (SQE) test errors

A count of times that the SQE test error is generated by the PHY layer for a particular interface. The SQE test error is defined in section 7.2.2.2.4 of ANSI/IEEE Std 802.3-1985 and its generation in 7.2.4.6 of the same document.

glossary (continued)

TX data errors

This statistic is switchable between:

1. The number of transmit frames discarded on transmission due to lack of resources (i.e., the transmit queue is full). This allows queue monitoring for dynamic queue sizing and buffer allocation.
2. The number of data errors at transmission. This is incremented when a mismatch is seen between a received good CRC and a checked CRC at transmission or when a partial frame is transmitted due to a receive underrun.

The function this counter performs is selected by the STMAP bit (bit 3) of the system control register.

TX H/W errors

The function of this counter is performed by the TX data errors counter.

TX octets

This contains a count of data and padding octets of frames that were successfully transmitted. See IEEE Std 802.3 Supplements, Layer Management (section 5) 5.2.2.1.1, number 5.

unallocated memory

Due to the memory configuration, an additional 16 words of memory are created. Address 0x980–0x9FF are reserved.

undersize RX frames

The total number of packets received that were less than 64 octets long (excluding framing bits, but including FCS octets) and were otherwise well formed.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, V_{CC} (see Notes 1 and 2)	–0.5 V to 4 V
Supply voltage range, $V_{CC\ (5V)}$ (see Notes 1 and 2)	–0.5 V to 5.5 V
Input voltage range, V_I	–0.5 V to $V_{CC(5\ V)}$ + 0.5 V
Output voltage range, V_O	–0.5 V to V_{CC}
Thermal impedance, junction-to-ambient package, airflow = 0, $Z_{\theta JA}$	11.11°C/W
Thermal impedance, junction-to-ambient package, airflow = 100 ft/min, $Z_{\theta JA}$	9.61°C/W
Thermal impedance, junction-to-case package, $Z_{\theta JC}$	0.94°C/W
Operating case temperature range, T_C	0°C to 95°C
Storage temperature range, T_{stg}	–65°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” can cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods can affect device reliability.

NOTES: 1. All voltage values are with respect to GND.

2. Turning power supplies on and off (cycling sequence) within a mixed 5-V/3.3-V system is an important consideration. The designer must observe a few rules to avoid damaging the TNETX3150/TNETX3150A. Check with the manufacturers of all components used in the 3.3-V to 5-V interface to ensure that no unique device characteristics exist that would lead to rules more restrictive than the TNETX3150/TNETX3150A requires.

The optimum solution to power-supply sequencing in a mixed-voltage system is to ramp up the 3.3-V supply first. A power-on reset component operating from this supply forces all 5-V tolerant outputs into the high-impedance state. Then, the 5-V supply is ramped up. On power down, the 5-V rail deenergizes first, followed by the 3.3-V rail.

The second-best solution is to ramp both the 3.3-V and 5-V rails at the same time, making sure that no more than 3.6 V exists between these two rails during the ramp up or down. If the 3.3 V is derived from the 5 V, then the 3.3 V rises as the 5 V rises so that the 5-V rail never exceeds the 3.3-V rail by more than 3.6 V. Both the optimum and second-choice algorithms for power up prevent device damage. If it is impractical to implement ramping, follow these rules:

- When turning on the power supply, all 3.3-V and 5-V supplies should start ramping from 0 V and reach 95 percent of their end-point values within 25 ms. All bus contention between the device and external devices is eliminated by the end of 25 ms.
- When turning off the power supply, 3.5-V and 5-V supplies should start ramping from steady-state values and reach 5 percent of these values within 25 ms. All bus contention between the device and external devices is eliminated by the end of 25 ms. There is a 250-second lifetime maximum at greater than 3.6 V between the supply rails. Holding this period to 25 ms per power-on/off cycle should not significantly contribute to mean time between failure (MTBF) shifts during product lifetimes.

recommended operating conditions

		TNETX3150			TNETX3150A			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	3	3.3	3.6	3	3.3	3.6	V
$V_{CC(5V)}$	Supply voltage	4.5	5	5.5	4.5	5	5.5	V
V_{IH}	High-level input voltage	2		$V_{CC(5V)}$	2		$V_{CC(5V)}$	V
V_{IL}	Low-level input voltage (see Note 3)	0		0.8	0		0.8	V
I_{OH}	High-level output current			–4			–2	mA
I_{OL}	Low-level output current			4			2	mA

NOTE 3: The algebraic convention, where the more negative (less positive) limit is designated as a minimum, is used for logic voltage levels only.

electrical characteristics over recommended operating conditions (unless otherwise noted)

PARAMETER		TNETX3150				TNETX3150A				UNIT
		TEST CONDITIONS	MIN	TYP	MAX	TEST CONDITIONS	MIN	TYP	MAX	
V_{OH}	High-level output voltage	I_{OH} = –4mA	2.4			I_{OH} = –2mA	2.4			V
V_{OL}	Low-level output voltage	I_{OL} = 4mA			0.4	I_{OL} = 2mA			0.4	V
I_{OZ}	High-impedance-state output current	$V_O = V_{CC}$			20	$V_O = V_{CC}$			20	µA
		V_O = 0			–20	V_O = 0			–20	
I_{IH}	High-level input current	$V_I = V_{I(MAX)}$			–10	$V_I = V_{I(MAX)}$			–20	µA
I_{IL}	Low-level input current	V_I = GND			+10	V_I = GND			+20	µA
I_{CC}	Supply current, 3.6 V				1500				550	mA

timing requirements (see Note 4 and Figure 34)
DRAM read cycle

NO.			TNETX3150			TNETX3150A			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
1	$t_{c(OSCIN)}$	Cycle time, OSCIN clock		20			20		ns
2	$t_{w(OSCINH)}$	Pulse duration, OSCIN high	8			8			ns
3	$t_{w(OSCINL)}$	Pulse duration, OSCIN low	8			8			ns
		Frequency drift, OSCIN clock			±50			±50	ppm

NOTE 4: All DRAM output signals are synchronous to the DREF clock. Figure 14 shows a single DRAM read (TNETX3150/TNETX3150A forward pointer update).

operating characteristics over recommended operating conditions (see Note 4 and Figure 34)
DRAM read cycle

NO.	PARAMETER		TNETX3150			TNETX3150A			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
4	$t_{c(DREF)}$	Cycle time, DREF clock		20			20		ns
5	$t_{d(DD)1}$	Delay time, from DREF↑ to DD35–DD0 valid			24			25	ns
6	$t_{d(DD)2}$	Delay time, from DREF↑ to DD35–DD0 invalid	3			0			ns
7	$t_{d(DA)}$	Delay time, from DREF↑ to DA7–DA0 valid/invalid	–4		3	–4		3	ns
7	$t_{d(DX)}$	Delay time, from DREF↑ to DX2–DX0 valid/invalid	–4		3	–4		3	ns
8	$t_{d(DRAS)}$	Delay time, from DREF↑ to $\overline{\text{DRAS}}$ transition	–4		5	–4		3	ns
9	$t_{d(DWE)}$	Delay time, from DREF↑ to $\overline{\text{DWE}}$ transition	–4		4	–4		3	ns
10	$t_{d(DCAS)}$	Delay time, from DREF↑ to $\overline{\text{DCAS}}$ transition	–4		4	–4		3	ns
11	$t_{d(DOE)}$	Delay time, from DREF↑ to $\overline{\text{DOE}}$ transition	–4		5	–4		3	ns

NOTE 4: All DRAM output signals are synchronous to the DREF clock. Figure 14 shows a single DRAM read (TNETX3150/TNETX3150A forward pointer update).

operating characteristics over recommended operating conditions (see Note 5 and Figure 34)
DRAM read cycle

NO.	PARAMETER		TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
12	$t_{d(DA)}$	Delay time, from OSCIN↑ to DA7–DA0 valid/invalid	3	15	3	15	ns
12	$t_{d(DX)}$	Delay time, from OSCIN↑ to DX2–DX0 valid/invalid	3	15	3	15	ns
13	$t_{d(DRAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DRAS}}$ transition	3	15	3	15	ns
14	$t_{d(DWE)}$	Delay time, from OSCIN↑ to $\overline{\text{DWE}}$ transition	3	15	3	15	ns
15	$t_{d(DCAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DCAS}}$ transition	3	15	3	15	ns
16	$t_{d(DOE)}$	Delay time, from OSCIN↑ to $\overline{\text{DOE}}$ transition	3	15	3	15	ns

NOTE 5: All DRAM output signals are synchronous to the OSCIN clock. Figure 14 shows a single DRAM read (TNETX3150/TNETX3150A forward pointer update).

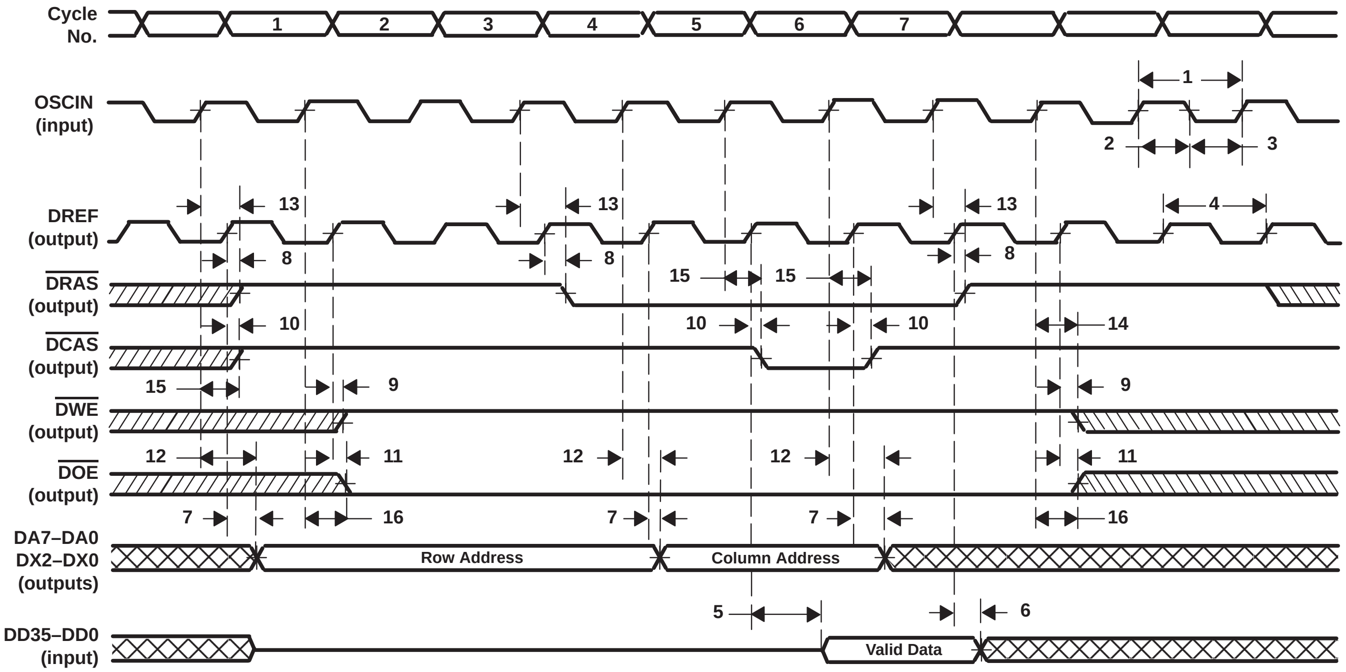


Figure 34. DRAM Read Cycle

timing requirements (see Note 6 and Figure 35) DRAM write cycle

NO.			MIN	NOM	MAX	UNIT
1	$t_{c(OSCIN)}$	Cycle time, OSCIN clock		20		ns
2	$t_{w(OSCINH)}$	Pulse duration, OSCIN high	8			ns
3	$t_{w(OSCINL)}$	Pulse duration, OSCIN low	8			ns
		Frequency drift, OSCIN clock			±50	ppm

NOTE 6: All DRAM output signals are synchronous to the DREF clock. Figure 15 shows a single DRAM write (TNETX3150/TNETX3150A forward pointer update).

operating characteristics over recommended operating conditions (see Note 6 and Figure 35) DRAM write cycle (with DREF clock)

NO.	PARAMETER		TNETX3150			TNETX3150A			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
4	$t_{c(DREF)}$	Cycle time, DREF clock		20			20		ns
5	$t_{d(DA)}$	Delay time, from DREF↑ to DA7–DA0 valid/invalid	–4		3	–4		3	ns
5	$t_{d(DX)}$	Delay time, from DREF↑ to DX2–DX0 valid/invalid	–4		3	–4		3	ns
6	$t_{d(DRAS)}$	Delay time, from DREF↑ to $\overline{\text{DRAS}}$ transition	–4		5	–4		3	ns
7	$t_{d(DD)}$	Delay time, from DREF↑ to DD35–DD0 valid/invalid	–4		8	–4		3	ns
8	$t_{d(DOE)}$	Delay time, from DREF↑ to $\overline{\text{DOE}}$ transition	–4		5	–4		3	ns
9	$t_{d(DWE)}$	Delay time, from DREF↑ to $\overline{\text{DWE}}$ transition	–4		4	–4		3	ns
10	$t_{d(DCAS)}$	Delay time, from DREF↑ to $\overline{\text{DCAS}}$ transition	–4		4	–4		3	ns

NOTE 6: All DRAM output signals are synchronous to the DREF clock. Figure 15 shows a single DRAM write (TNETX3150/TNETX3150A forward pointer update).

operating characteristics over recommended operating conditions (see Note 7 and Figure 35) DRAM write cycle (with OSCIN clock)

NO.	PARAMETER		TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
11	$t_{d(DA)}$	Delay time, from OSCIN↑ to DA7–DA0 valid/invalid	3	15	3	15	ns
11	$t_{d(DX)}$	Delay time, from OSCIN↑ to DX2–DX0 valid/invalid	3	15	3	15	ns
12	$t_{d(DRAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DRAS}}$ transition	3	15	3	15	ns
13	$t_{d(DD)}$	Delay time, from OSCIN↑ to DD35–DD0 valid/invalid	3	17	3	15	ns
14	$t_{d(DOE)}$	Delay time, from OSCIN↑ to $\overline{\text{DOE}}$ transition	3	15	3	15	ns
15	$t_{d(DWE)}$	Delay time, from OSCIN↑ to $\overline{\text{DWE}}$ transition	3	15	3	15	ns
16	$t_{d(DCAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DCAS}}$ transition	3	15	3	15	ns

NOTE 7: All DRAM output signals are synchronous to the OSCIN clock. Figure 15 shows a single DRAM write (TNETX3150/TNETX3150A forward pointer update).

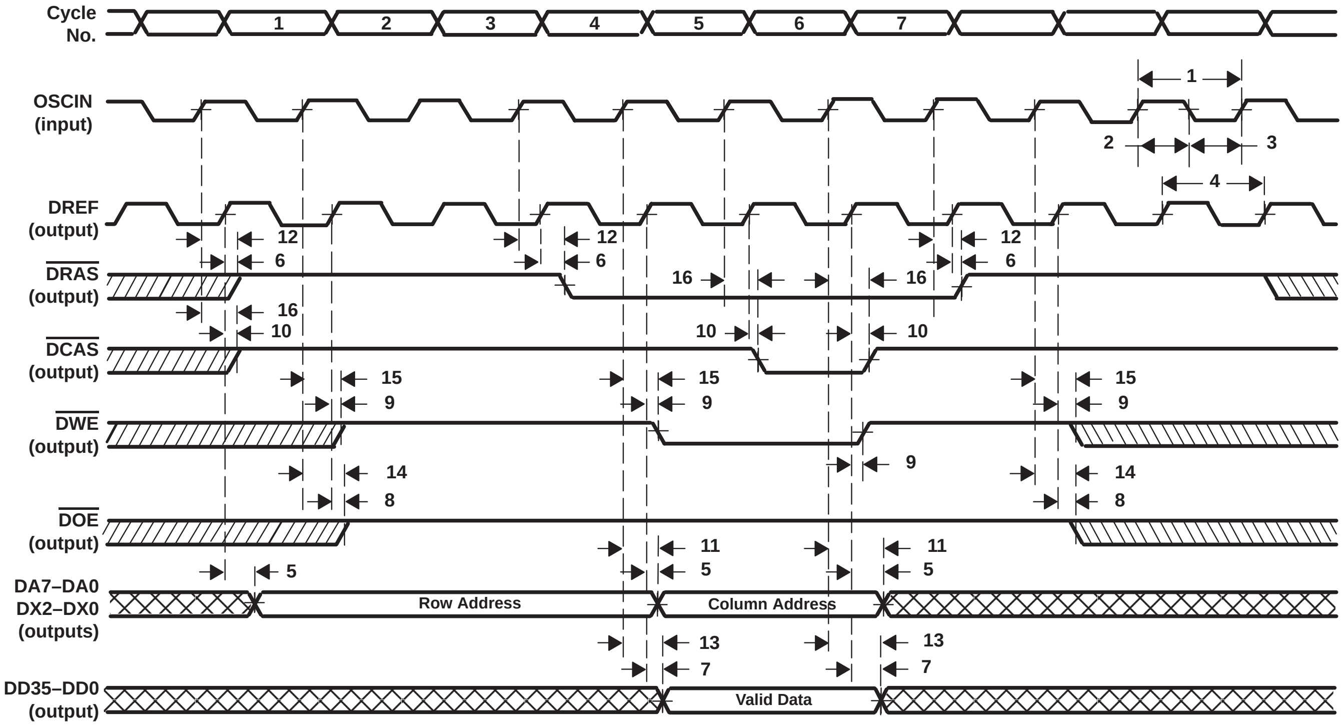


Figure 35. DRAM Write Cycle

timing requirements (see Notes 8 and 9 and Figure 36) DRAM CAS-before-RAS (CBR) refresh cycle

NO.			MIN	NOM	MAX	UNIT
1	$t_{c(OSCIN)}$	Cycle time, OSCIN clock		20		ns
2	$t_{w(OSCINH)}$	Pulse duration, OSCIN high	8			ns
3	$t_{w(OSCINL)}$	Pulse duration, OSCIN low	8			ns
		Frequency drift, OSCIN clock			±50	ppm

NOTES: 8. All DRAM output signals are synchronous to the DREF clock.

9. The TNETX3150/TNETX3150A produces a refresh request at a fixed rate of once every 10.22 µs. If a block transfer is underway, the refresh is deferred.

operating characteristics over recommended operating conditions (see Notes 8 and 9 and Figure 36) DRAM CAS-before-RAS (CBR) refresh cycle (with DREF clock)

NO.	PARAMETER		TNETX3150			TNETX3150A			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
4	$t_{c(DREF)}$	Cycle time, DREF clock		20			20		ns
5	$t_{d(DCAS)}$	Delay time, from DREF↑ to $\overline{\text{DCAS}}$ transition	–4		4	–4		3	ns
6	$t_{d(DRAS)}$	Delay time, from DREF↑ to $\overline{\text{DRAS}}$ transition	-4		5	-4		3	ns
7	$t_{d(DOE)}$	Delay time, from DREF↑ to $\overline{\text{DOE}}$ transition	–4		5	–4		3	ns
8	$t_{d(DWE)}$	Delay time, from DREF↑ to $\overline{\text{DWE}}$ transition	–4		4	–4		3	ns

NOTES: 8. All DRAM output signals are synchronous to the DREF clock.

9. The TNETX3150/TNETX3150A produces a refresh request at a fixed rate of once every 10.22 µs. If a block transfer is underway, the refresh is deferred.

operating characteristics over recommended operating conditions (see Notes 9 and 10 and Figure 36) DRAM CAS-before-RAS (CBR) refresh cycle (with OSCIN clock)

NO.	PARAMETER		TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
9	$t_{d(DCAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DCAS}}$ transition	3	15	3	15	ns
10	$t_{d(DRAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DRAS}}$ transition	3	15	3	15	ns
11	$t_{d(DOE)}$	Delay time, from OSCIN↑ to $\overline{\text{DOE}}$ transition	3	15	3	15	ns
12	$t_{d(DWE)}$	Delay time, from OSCIN↑ to $\overline{\text{DWE}}$ transition	3	15	3	15	ns

NOTES: 9. The TNETX3150/TNETX3150A produces a refresh request at a fixed rate of once every 10.22 µs. If a block transfer is underway, the refresh is deferred.

10. All DRAM output signals are synchronous to the OSCIN clock. Figure 15 shows a single DRAM write (TNETX3150/TNETX3150A forward pointer update).

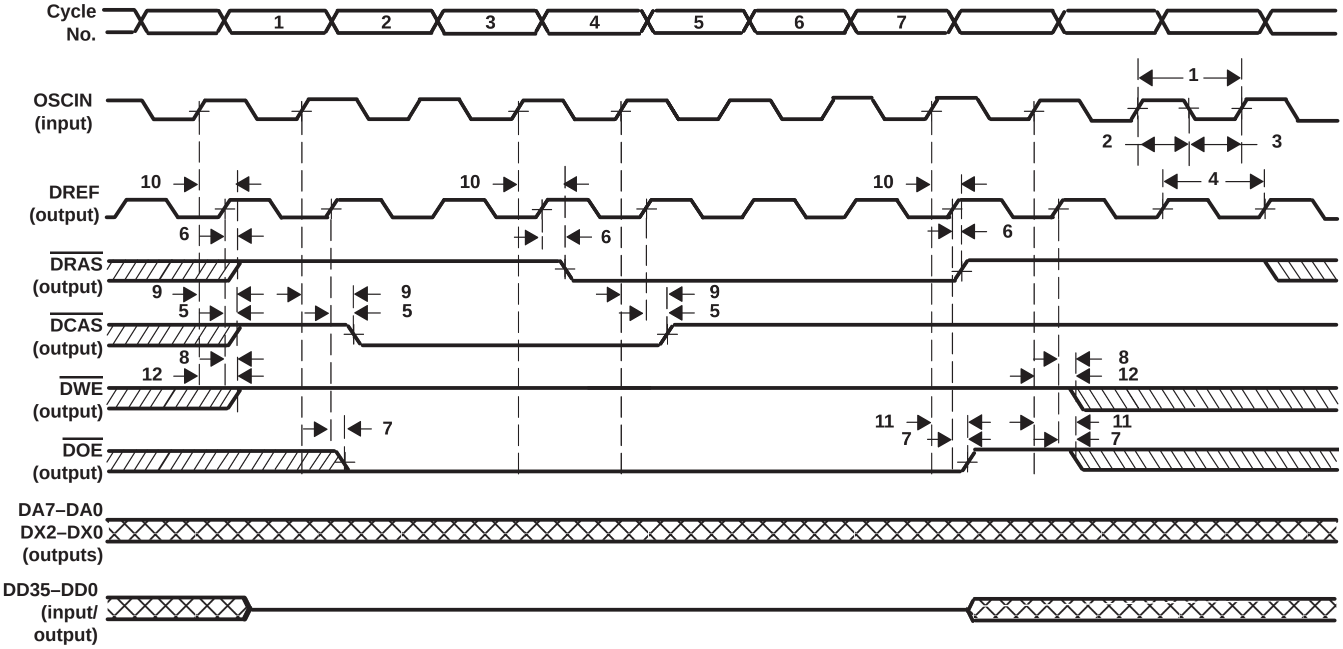


Figure 36. CBR Refresh Cycle

timing requirements (see Figure 37)
DRAM burst write cycle (TNETX3150/TNETX3150A frame buffer write)

NO.			MIN	NOM	MAX	UNIT
1	$t_{c(OSCIN)}$	Cycle time, OSCIN clock		20		ns
2	$t_{w(OSCINH)}$	Pulse duration, OSCIN high	8			ns
3	$t_{w(OSCINL)}$	Pulse duration, OSCIN low	8			ns
		Frequency drift, OSCIN clock			±50	ppm

operating characteristics over recommended operating conditions (see Figure 37)
DRAM burst write cycle (TNETX3150/TNETX3150A frame buffer write) (with DREF clock)

NO.	PARAMETER		TNETX3150			TNETX3150A			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
4	$t_{c(DREF)}$	Cycle time, DREF clock		20			20		ns
5	$t_{d(DA)}$	Delay time, from DREF↑ to DA7–DA0 valid/invalid	–4		3	–4		3	ns
5	$t_{d(DX)}$	Delay time, from DREF↑ to DX2–DX0 valid/invalid	–4		3	–4		3	ns
6	$t_{d(DRAS)}$	Delay time, from DREF↑ to $\overline{\text{DRAS}}$ transition	–4		5	–4		3	ns
7	$t_{d(DD)}$	Delay time, from DREF↑ to DD35–DD0 valid/invalid	–4		8	–4		3	ns
8	$t_{d(DOE)}$	Delay time, from DREF↑ to $\overline{\text{DOE}}$ transition	–4		5	–4		3	ns
9	$t_{d(DWE)}$	Delay time, from DREF↑ to $\overline{\text{DWE}}$ transition	–4		4	–4		3	ns
10	$t_{d(DCAS)}$	Delay time, from DREF↑ to $\overline{\text{DCAS}}$ transition	–4		4	–4		3	ns

operating characteristics over recommended operating conditions (see Figure 37)
DRAM burst write cycle (TNETX3150/TNETX3150A frame buffer write) (with OSCIN clock)

NO.	PARAMETER		TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
11	$t_{d(DA)}$	Delay time, from OSCIN↑ to DA7–DA0 valid/invalid	3	15	3	15	ns
11	$t_{d(DX)}$	Delay time, from OSCIN↑ to DX2–DX0 valid/invalid	3	15	3	15	ns
12	$t_{d(DRAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DRAS}}$ transition	3	15	3	15	ns
13	$t_{d(DD)}$	Delay time, from OSCIN↑ to DD35–DD0 valid/invalid	3	17	3	15	ns
14	$t_{d(DOE)}$	Delay time, from OSCIN↑ to $\overline{\text{DOE}}$ transition	3	15	3	15	ns
15	$t_{d(DWE)}$	Delay time, from OSCIN↑ to $\overline{\text{DWE}}$ transition	3	15	3	15	ns
16	$t_{d(DCAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DCAS}}$ transition	3	15	3	15	ns

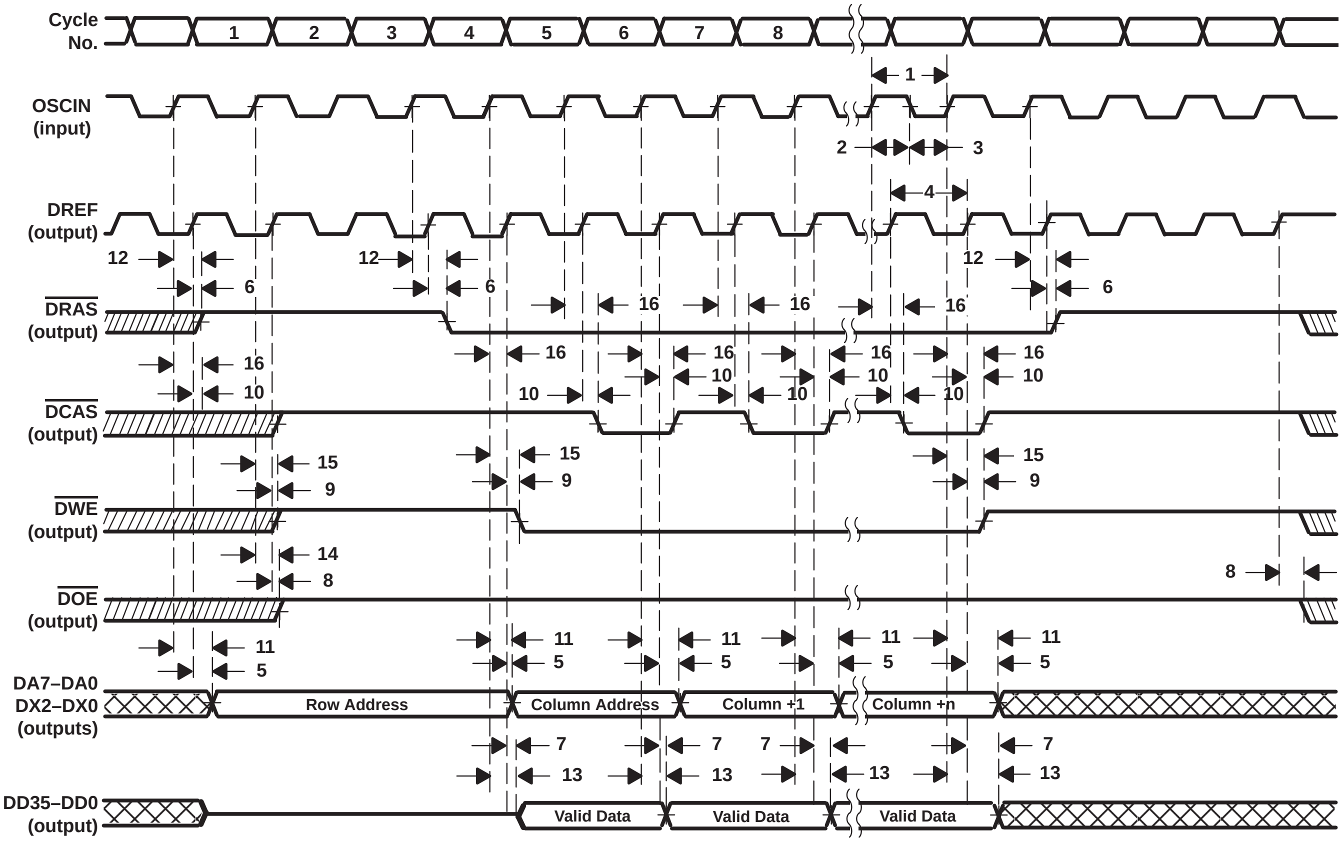


Figure 37. DRAM Burst Write Cycle

timing requirements (see Figure 38)
DRAM burst read cycle (TNETX3150/TNETX3150A frame buffer read)

NO.			MIN	NOM	MAX	UNIT
1	$t_{c(OSCIN)}$	Cycle time, OSCIN clock		20		ns
2	$t_{w(OSCINH)}$	Pulse duration, OSCIN high	8			ns
3	$t_{w(OSCINL)}$	Pulse duration, OSCIN low	8			ns
		Frequency drift, OSCIN clock			±50	ppm

operating characteristics over recommended operating conditions (see Figure 38)
DRAM burst read cycle (TNETX3150/TNETX3150A frame buffer read) (with DREF clock)

NO.	PARAMETER		TNETX3150			TNETX3150A			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
4	$t_{c(DREF)}$	Cycle time, DREF clock					20		ns
5	$t_{d(DD)1}$	Delay time, from DREF↑ to DD35–DD0 valid			24			25	ns
6	$t_{d(DD)2}$	Delay time, from DREF↑ to DD35–DD0 invalid	3			0			ns
7	$t_{d(DA)}$	Delay time, from DREF↑ to DA7–DA0 valid/invalid	–4		3	–4		3	ns
7	$t_{d(DX)}$	Delay time, from DREF↑ to DX2–DX0 valid/invalid	–4		3	–4		3	ns
8	$t_{d(DRAS)}$	Delay time, from DREF↑ to $\overline{\text{DRAS}}$ transition	–4		5	–4		3	ns
9	$t_{d(DWE)}$	Delay time, from DREF↑ to $\overline{\text{DWE}}$ transition	–4		4	–4		3	ns
10	$t_{d(DCAS)}$	Delay time, from DREF↑ to $\overline{\text{DCAS}}$ transition	–4		4	–4		3	ns
11	$t_{d(DOE)}$	Delay time, from DREF↑ to $\overline{\text{DOE}}$ transition	–4		5	–4		3	ns

operating characteristics over recommended operating conditions (see Figure 38)
DRAM burst read cycle (TNETX3150/TNETX3150A frame buffer read) (with OSCIN clock)

NO.	PARAMETER		TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
12	$t_{d(DA)}$	Delay time, from OSCIN↑ to DA7–DA0 valid/invalid	3	15	3	15	ns
12	$t_{d(DX)}$	Delay time, from OSCIN↑ to DX2–DX0 valid/invalid	3	15	3	15	ns
13	$t_{d(DRAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DRAS}}$ transition	3	15	3	15	ns
14	$t_{d(DWE)}$	Delay time, from OSCIN↑ to $\overline{\text{DWE}}$ transition	3	15	3	15	ns
15	$t_{d(DCAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DCAS}}$ transition	3	15	3	15	ns
16	$t_{d(DOE)}$	Delay time, from OSCIN↑ to $\overline{\text{DOE}}$ transition	3	15	3	15	ns

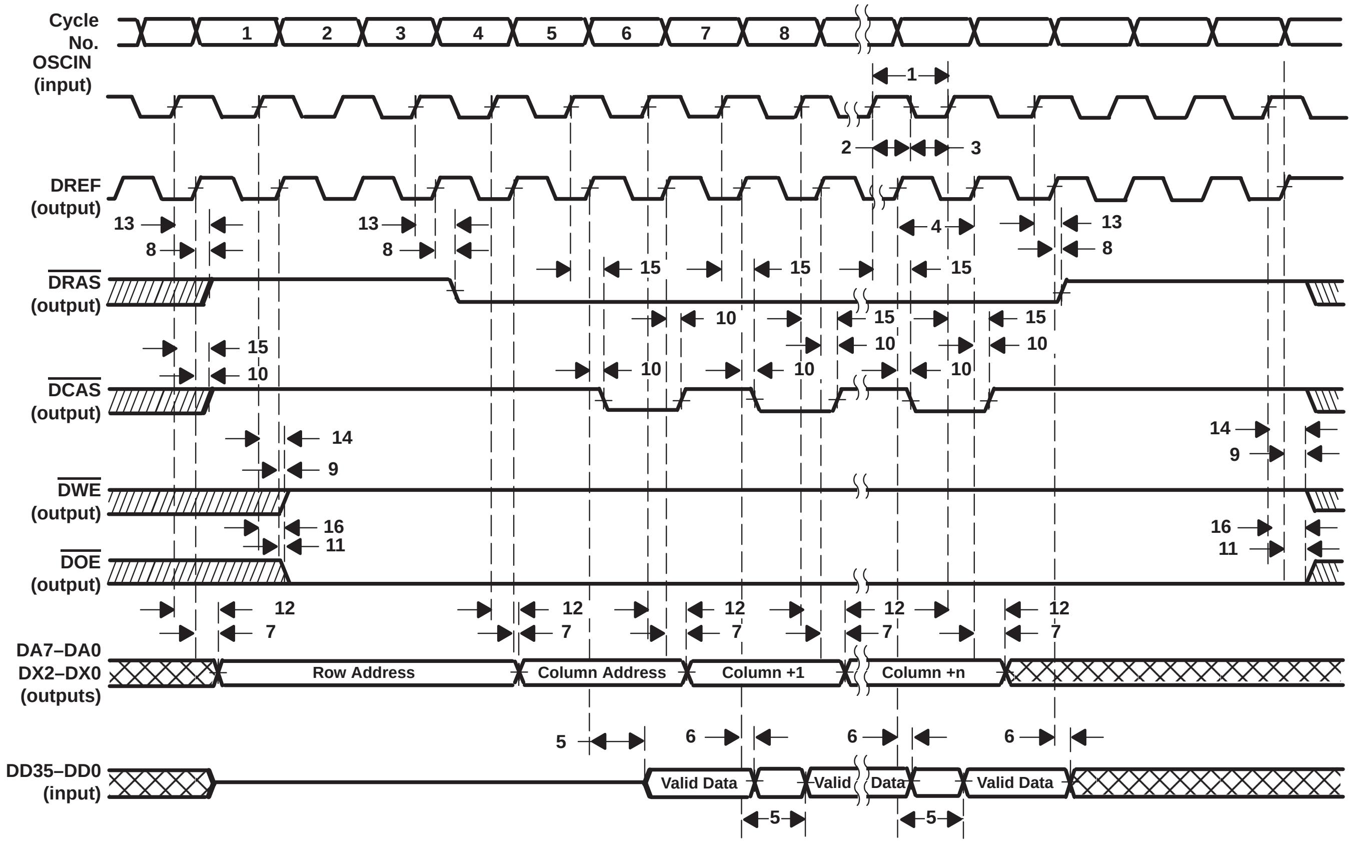


Figure 38. DRAM Burst Read Cycle

DIO write cycle

- TNETX3150/TNETX3150A host-register address data SAD1–SAD0 and SDATA7–SDATA0 are asserted and SRNW is taken low.
- After the setup time, $\overline{\text{SCS}}$ is taken low, initiating a write cycle.
- The TNETX3150/TNETX3150A pulls $\overline{\text{SRDY}}$ low as the data is accepted and SDATA7–SDATA0, SAD1–SAD0, and SRNW are deasserted after the hold time is satisfied.
- $\overline{\text{SCS}}$ is taken high by the host to complete the cycle, causing $\overline{\text{SRDY}}$ to be deasserted and $\overline{\text{SRDY}}$ is driven high for one cycle before being placed in the high-impedance (Z) state.

timing requirements over recommended operating conditions (see Note 11 and Figure 39) write cycle DIO interface

NO.			TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{w(SCS)}$	Pulse duration, $\overline{\text{SCS}}$ low	40		40		ns
2	$t_{su(SAD)}$	Setup time, SAD1–SAD0 valid before $\overline{\text{SCS}}\downarrow$	0		0		ns
3	$t_{su(SDATA)}$	Setup time, SDATA7–SDATA0 valid before $\overline{\text{SCS}}\downarrow$	0		0		ns
4	$t_{su(SRNW)}$	Setup time, SRNW low before $\overline{\text{SCS}}\downarrow$	0		0		ns

NOTE 11: The DIO interface/DIO write cycle is asynchronous, allowing easy adaptation to a range of microprocessor devices and computer system interfaces.

operating characteristics over recommended operating conditions (see Note 11 and Figure 39) write cycle DIO interface

NO.	PARAMETER		TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
5	$t_{w(SRDY)}$	Pulse duration, $\overline{\text{SRDY}}$ high		20		20	ns
6	$t_{d(SRNW)}$	Delay time, from $\overline{\text{SRDY}}\downarrow$ to SRNW$\uparrow$	0		0		ns
7	$t_{d(SAD)}$	Delay time, from $\overline{\text{SRDY}}\downarrow$ to SAD1–SAD0 invalid	0		0		ns
8	$t_{d(SDATA)}$	Delay time, from $\overline{\text{SRDY}}\downarrow$ to SDATA7–SDATA0 invalid	0		0		ns
9	$t_{d(SCS\text{-}SRDY)1}$	Delay time from $\overline{\text{SCS}}\uparrow$ to $\overline{\text{SRDY}}\uparrow$	0	20	0	60	ns
10	$t_{d(SCS\text{-}SRDY)2}$	Delay time from $\overline{\text{SCS}}\downarrow$ to $\overline{\text{SRDY}}\downarrow$	0		0		ns

NOTE 11: The DIO interface/DIO write cycle is asynchronous, allowing easy adaptation to a range of microprocessor devices and computer system interfaces.

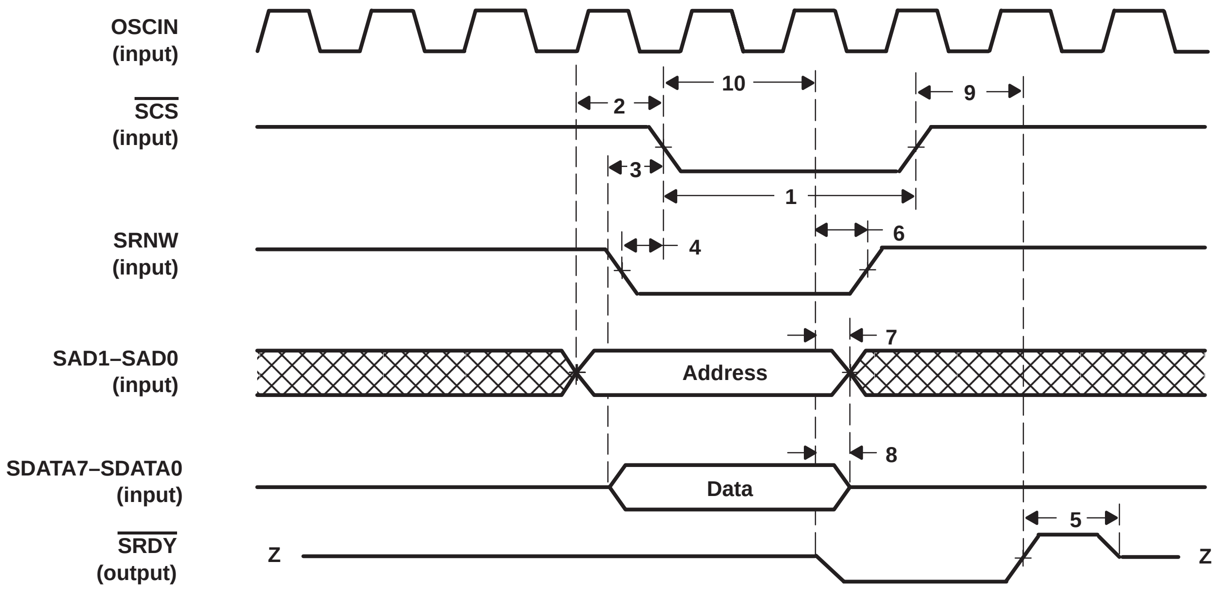


Figure 39. DIO Write Cycle

DIO read cycle

- The TNETX3150/TNETX3150A host-register address data is placed on address terminals SAD1–SAD0 and SRNW is held high.
- After setup time, $\overline{\text{SCS}}$ is taken low, initiating the read cycle.
- After delay following $\overline{\text{SCS}}$ low, SDATA7–SDATA0 is released from the Z state.
- After delay following $\overline{\text{SCS}}$ low, SDATA7–SDATA0 is driven with valid data and $\overline{\text{SRDY}}$ is pulled low. The host can access the data.
- $\overline{\text{SCS}}$ is taken high by the host upon completion of the cycle, causing $\overline{\text{SRDY}}$ to be deasserted and $\overline{\text{SRDY}}$ is driven high for one clock cycle before being placed in the Z state. SDATA7–SDATA0 is also placed in the Z state.

timing requirements over recommended operating conditions (see Note 12 and Figure 40) read cycle DIO interface

NO.			TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{w(SCS)}$	Pulse duration, $\overline{\text{SCS}}$ low	40		40		ns
2	$t_{su(SAD)}$	Setup time, SAD1–SAD0 valid before $\overline{\text{SCS}}$↓	0		0		ns
3	$t_{su(SRNW)}$	Setup time, SRNW high before $\overline{\text{SCS}}$↓	0		0		ns

NOTE 12: $\overline{\text{SRDY}}$ should be held high using an external pullup resistor to ensure correct system operation.

operating characteristics over recommended operating conditions (see Note 12 and Figure 40) read cycle DIO interface

NO.	PARAMETER		TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
4	$t_{w(SRDY)}$	Pulse duration, $\overline{\text{SRDY}}$ high		20		20	ns
5	$t_{d(SRNW)}$	Delay time, from $\overline{\text{SRDY}}$↓ to SRNW↓	0		0		ns
6	$t_{d(SAD)}$	Delay time, from $\overline{\text{SRDY}}$↓ to SAD1–SAD0 invalid	0		0		ns
7	$t_{d(SDATA)1}$	Delay time from $\overline{\text{SRDY}}$↓ to SDATA7–SDATA0 valid		3		2	ns
8	$t_{d(SDATA)2}$	Delay time from $\overline{\text{SCS}}$↑ to SDATA7–SDATA0 invalid	0	10	0	60	ns
9	$t_{d(SCS–SRDY)1}$	Delay time, $\overline{\text{SCS}}$↓ to $\overline{\text{SRDY}}$↓	0		0		ns
10	$t_{d(SCS–SRDY)2}$	Delay time, $\overline{\text{SCS}}$↑ to $\overline{\text{SRDY}}$↑	0	20	0	60	ns

NOTE 12: $\overline{\text{SRDY}}$ should be held high using an external pullup resistor to ensure correct system operation.

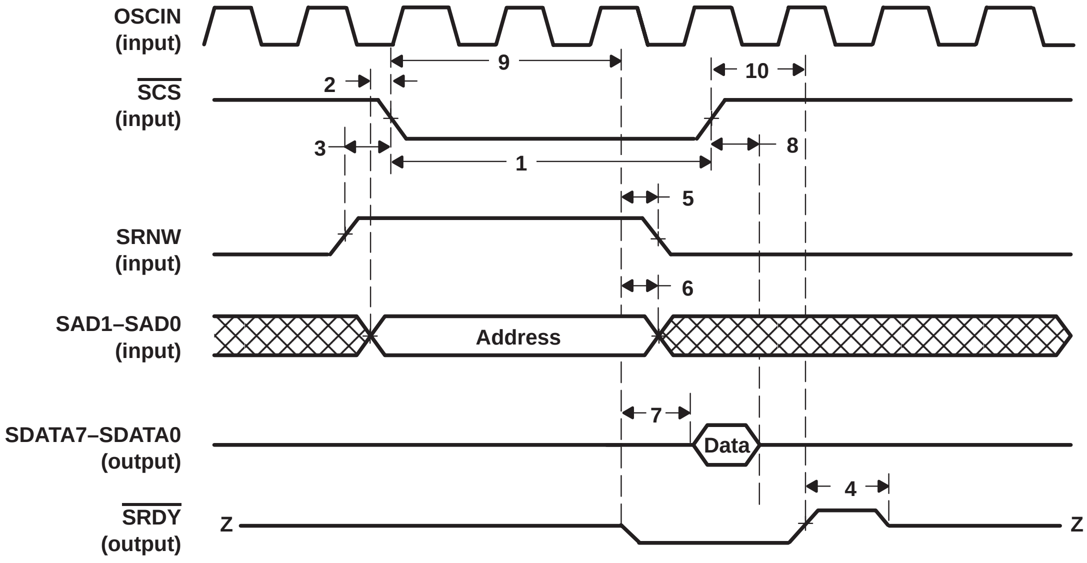


Figure 40. DIO Read Cycle

timing requirements over recommended operating conditions (see Figure 41) EAM interface

NO.			TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{su(EAM)}$	Setup time, EAM15–EAM0 valid before OSCIN↑	8		8		ns
2	$t_{h(EAM)}$	Hold time, EAM15–EAM0 valid after OSCIN↑	8		8		ns

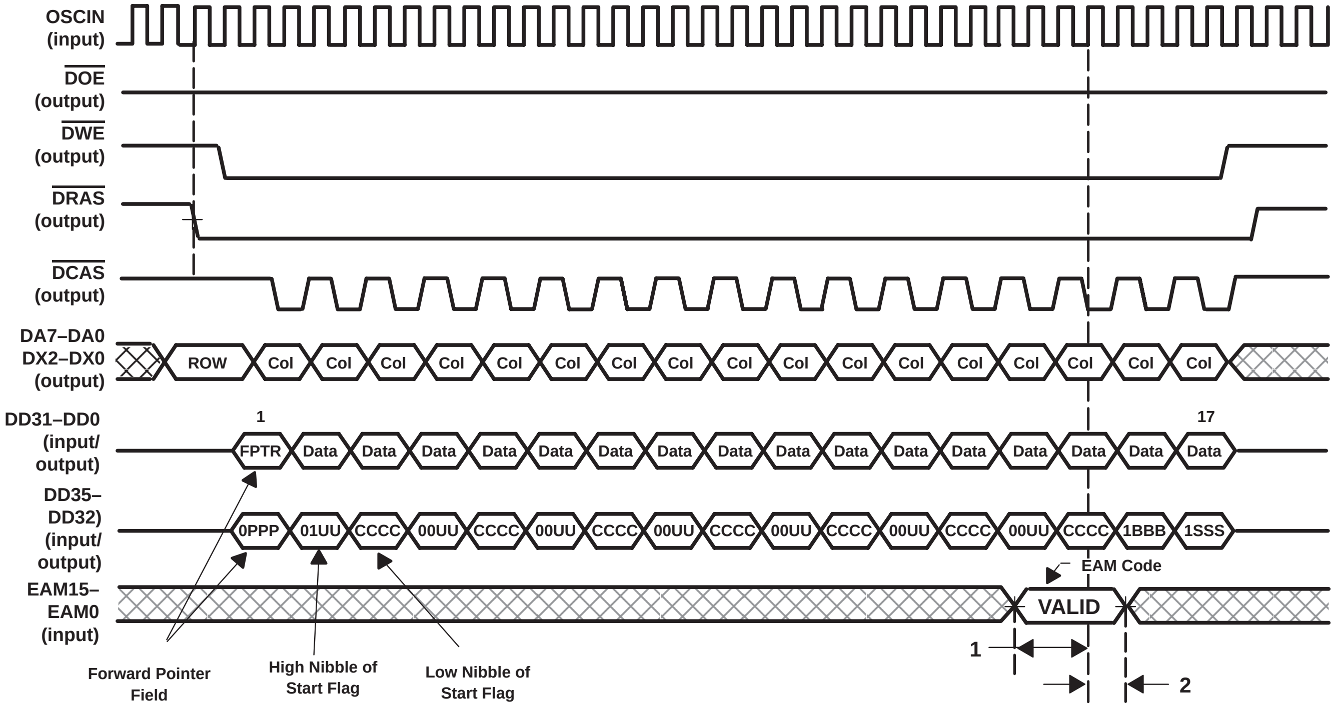
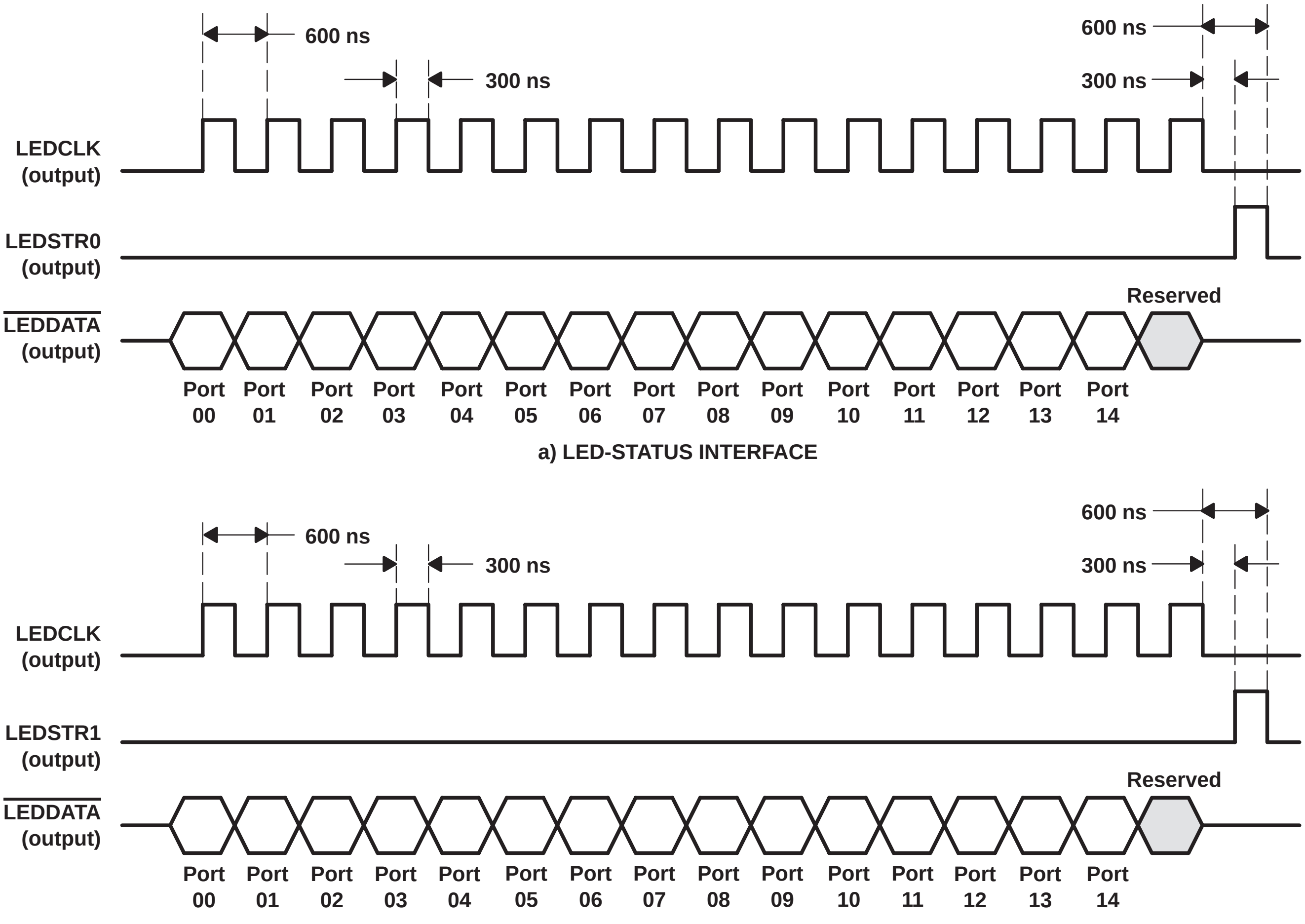


Figure 41. EAM Interface Timing

LED timing (see Figure 42)



a) LED-STATUS INTERFACE

b) TRANSMIT-QUEUE-STATUS INTERFACE

Figure 42. LED- and Transmit-Queue-Status Interface Timing

The LEDSTR1 signal is pulsed only when there has been a change in status for any of the transmit queues. An external system monitoring this signal can use it as a trigger to investigate which transmit queue is congested or has recovered from congestion.

During hardware reset, the TNETX3150/TNETX3150A turns on the LEDs attached to it to indicate that the TNETX3150/TNETX3150A is in reset. The LEDCLK is free running, LEDSTR0 and LEDSTR1 are held high, and $\overline{\text{LEDDATA}}$ is taken low.

With LEDSTR0 and LEDSTR1 high, the LED latch is transparent and the LEDs are illuminated.

When RESET is released, a normal LED cycle is initiated with $\overline{\text{LEDDATA}}$ held low and LEDSTR0 pulsed. The LEDs remain on until the start bit is set (an LED cycle updating LEDSTR1 is not performed).

When the START option bit is set, the TNETX3150/TNETX3150A initiates another LED cycle with $\overline{\text{LEDDATA}}$ held high, extinguishing the LEDs.

operating characteristics over recommended operating conditions (see Figure 43) EEPROM interface

NO.	PARAMETER		TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
	f_{clock} (ECLK)	Clock frequency, ECLK		98		98	kHz
1	t_d (ECLKH–EDIOL)	Delay time, from ECLK↑ to EDIO↓ (see Note 13)	5		5		µs
2	t_d (EDIOL–ECLKL)	Delay time, from EDIO↓ to ECLK↓ (see Note 13)	5		5		µs
3	t_d (ECLKL–EDIOX)	Delay time, from ECLK↓ to EDIO changing (see Note 14)	0		0		µs
4	t_d (EDIOV–ECLKH)	Delay time, from EDIO valid output to ECLK↑	0		0		µs
5	t_d (ECLKL–EDIOV)	Delay time, from ECLK↓ to EDIO valid	0		0		µs
6	t_d (ECLKL–EDIOX)	Delay time, from ECLK↓ to EDIO changing (see Note 15)	0		0		µs
7	t_d (ECLKH–EDIOX)	Delay time, from ECLK↑ to EDIO invalid	5		5		µs
8	t_d (EDIOV–ECLKH)	Delay time, from EDIO valid input to ECLK↑	10		10		µs

NOTES: 13. This is a start condition delay time during ECLK high.
14. This is a changing-data condition delay time for output EDIO.
15. This is a changing-data condition delay time for input EDIO.

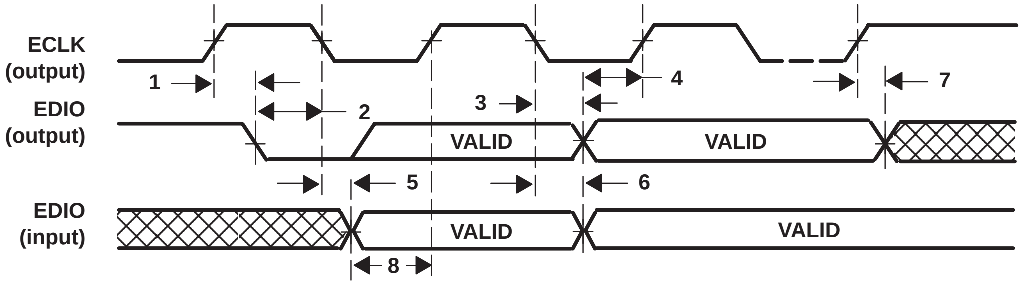


Figure 43. EEPROM Interface Timing

timing requirements over recommended operating conditions (see Figure 44) MII receive 100/200 Mbit/s (XX = ports 00–02)

NO.			TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{su(MXXRXD)1}$	Setup time, MXXRXD3–MXXRXD0 valid before MXXRCLK↑	8		8		ns
1[†]	$t_{su(MXXRXD)2}$	Setup time, MXXRXD7–MXXRXD0 valid before MXXRCLK↑	8		8		ns
1	$t_{su(MXXRXDV)}$	Setup time, MXXRXDV valid before MXXRCLK↑	8		8		ns
1[†]	$t_{su(MXXRXDVX)}$	Setup time, MXXRXDVX valid before MXXRCLK↑	8		8		ns
1	$t_{su(MXXRXER)}$	Setup time, MXXRXER valid before MXXRCLK↑	8		8		ns
2	$t_{h(MXXRXD)1}$	Hold time, MXXRXD3–MXXRXD0 valid after MXXRCLK↑	8		8		ns
2[†]	$t_{h(MXXRXD)2}$	Hold time, MXXRXD7–MXXRXD0 valid after MXXRCLK↑	8		8		ns
2	$t_{h(MXXRXDV)}$	Hold time, MXXRXDV valid after MXXRCLK↑	8		8		ns
2[†]	$t_{h(MXXRXDVX)}$	Hold time, MXXRXDVX valid after MXXRCLK↑	8		8		ns
2	$t_{h(MXXRXER)}$	Hold time, MXXRXER valid after MXXRCLK↑	8		8		ns

[†] Also applies to the uplink (port 00) when operating in the 200-Mbit/s mode, except that MXXRXD3–MXXRXD0 is replaced with MXXRXD7–MXXRXD0 and MXXRXDVX is used.

MXXRXD3–MXXRXD0 is driven by the PHY on the falling edge of MXXRCLK. MXXRXD3–MXXRXD0 timing must be met during clock periods when MXXRXDV is asserted. MXXRXDV is asserted and deasserted by the PHY on the falling edge of MXXRCLK. MXXRXER is driven by the PHY on the falling edge of MXXRCLK (XX = ports 00–02).

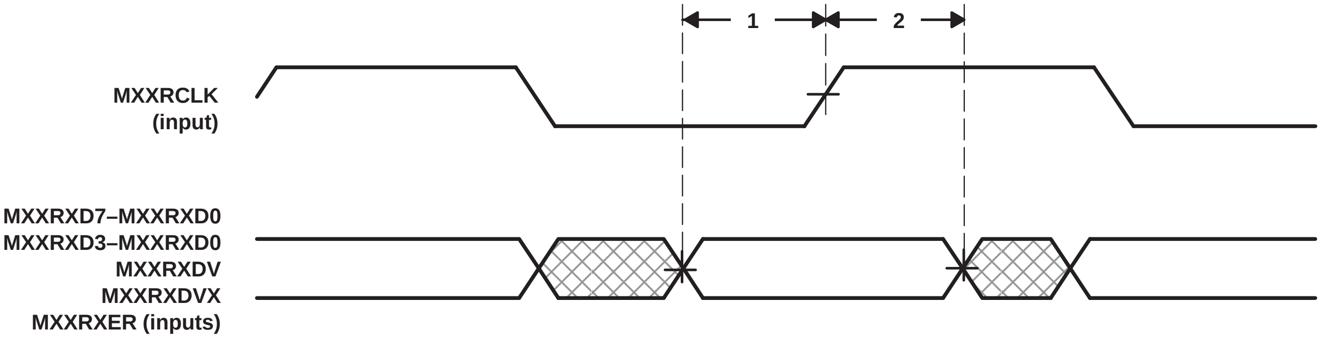


Figure 44. Ports (00–02) Receive Interface Timing

operating characteristics over recommended operating conditions (see Figure 45) MII transmit 100/200 Mbit/s (XX = ports 00–02)

NO.	PARAMETER		TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{d(MXXTXD)1}$	Delay time, from MXXTCLK↑ to MXXTXD3–MXXTXD0 valid	5	25	2.5	25	ns
1[†]	$t_{d(MXXTXD)2}$	Delay time, from MXXTCLK↑ to MXXTXD7–MXXTXD0 valid	5	25	2.5	25	ns
1	$t_{d(MXXTXEN)}$	Delay time, from MXXTCLK↑ to MXXTXEN valid	5	25	2.5	25	ns
1	$t_{d(MXXTXER)}$	Delay time, from MXXTCLK↑ to MXXTXER valid	5	25	2.5	25	ns

[†] Also applies to the uplink (port 00) when operating in the 200 Mbit/s mode, except that MXXRXD3–MXXRXD0 is replaced with MXXRXD7–MXXRXD0 and MXXRXDVX is used.

MXXTXD3–MXXTXD0 is driven by the reconciliation sublayer synchronous to the MXXTCLK. MXXTXEN is asserted and deasserted by the reconciliation sublayer synchronous to the MXXTCLK rising edge. MXXTXER is driven synchronous to the rising edge of MXXTCLK (XX = ports 00–02).

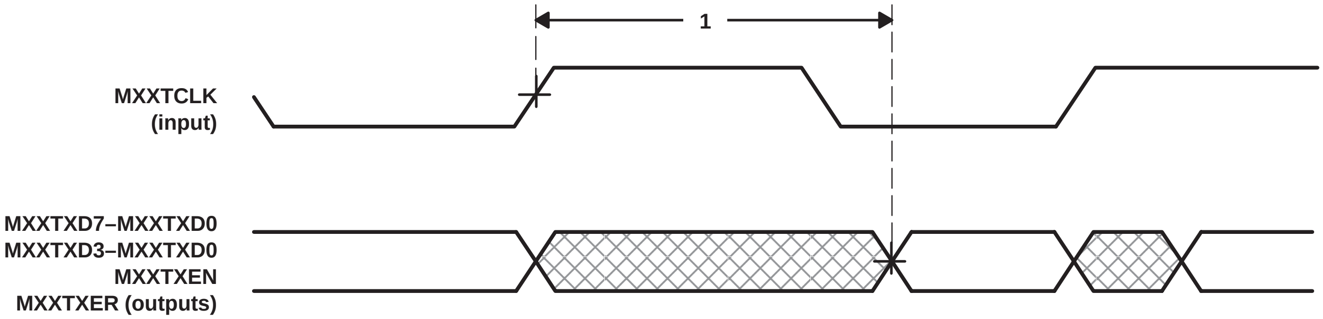


Figure 45. Ports (00–02) Transmit Interface Timing

timing requirements over recommended operating conditions (see Figure 46) MII receive 10 Mbit/s (XX = ports 03–14)

NO.			TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{su(MXXRXD)}$	Setup time, MXXRXD valid before MXXRCLK↑	8		8		ns
2	$t_{h(MXXRXD)}$	Hold time, MXXRXD valid after MXXRCLK↑	8		8		ns

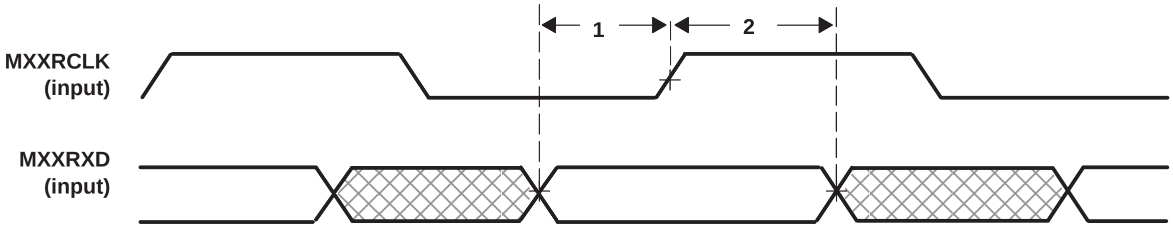


Figure 46. Ports (03–14) Receive Interface Timing

operating characteristics over recommended operating conditions (see Figure 47) MII transmit 10 Mbit/s (XX = ports 03–14)

NO.	PARAMETER		TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{d(MXXTXD)}$	Delay time, from MXXTCLK↑ to MXXTXD valid	5	35	2.5	25	ns
1	$t_{d(MXXTXEN)}$	Delay time, from MXXTCLK↑ to MXXTXEN valid	5	35	2.5	25	ns

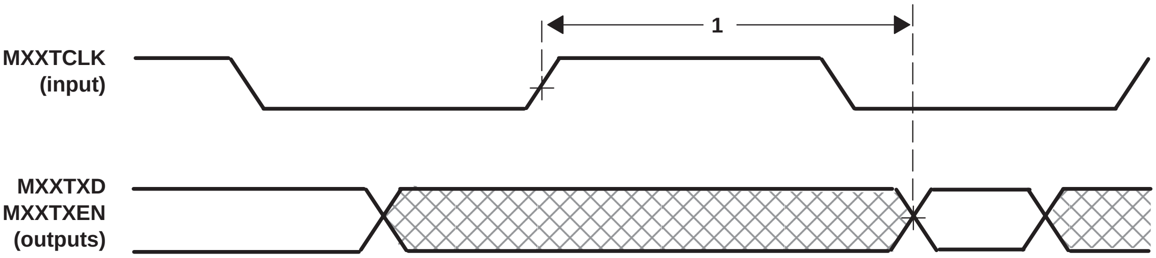


Figure 47. Ports (03–14) Transmit Interface Timing

operating characteristics over recommended operating conditions (see Notes 16 and 17 and Figure 48)

NMON interface (100-Mbit/s mode)

NO.	PARAMETER		TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{d(NMONXX)}$	Delay time, from NMON05↓ to NMONXX valid	–8	8	–8	8	ns

NOTE 16: The worst-case MII clock oscillator duty cycle specification is 65:35 (implying a minimum pulse duration of 14 ns). This timing delay produces a minimum system delay time (data stable before NMON05 rises) of 6 ns.

NOTE 17: Option bits: MONWIDE = 1
MONRXTX = 0/1 (same timing for both RX and TX wide formats)
XX = ports 00–04

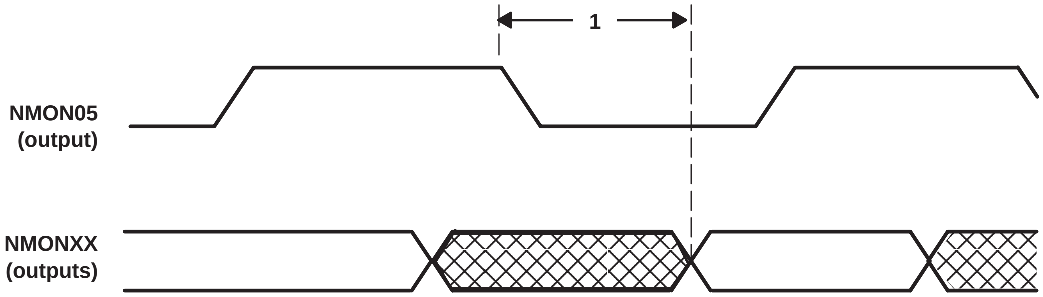


Figure 48. Ports 00–02 NMON Interface Timing (100-Mbit/s Mode)

operating characteristics over recommended operating conditions (see Note 18 and Figure 49) NMON interface (10-Mbit/s mode)

NO.	PARAMETER		TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{d(NMONXX)}$	Delay time, from NMON05↓ to NMONXX valid	–8	25	–8	25	ns
2	$t_{d(NMONYY)}$	Delay time, from NMON02↓ to NMONYY valid	–8	25	–8	25	ns

NOTE 18: Option bits: MONWIDE = 0
MONRXTX = Don’t care
XX = ports 03, 04
YY = ports 00, 01

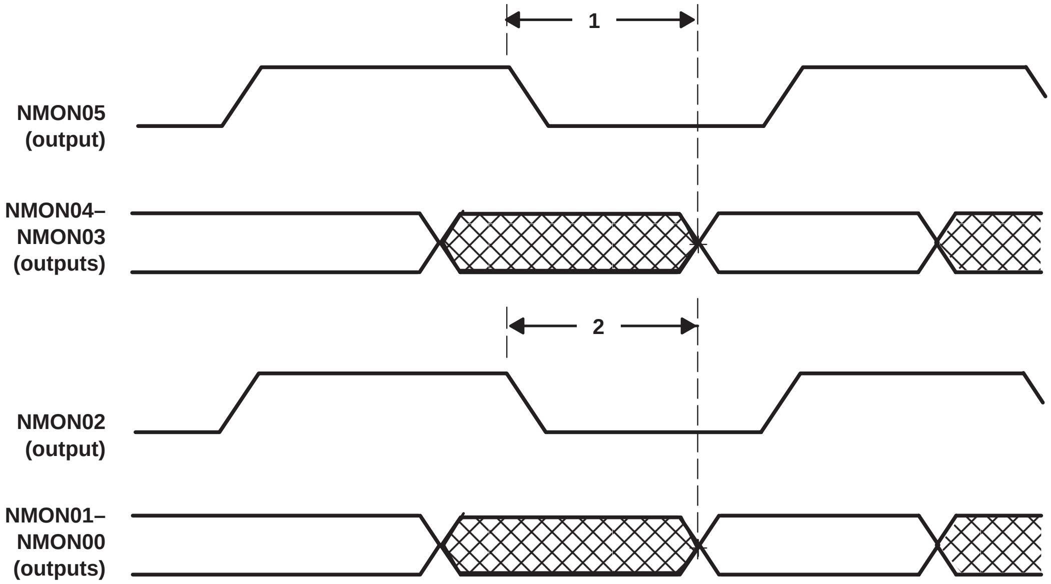
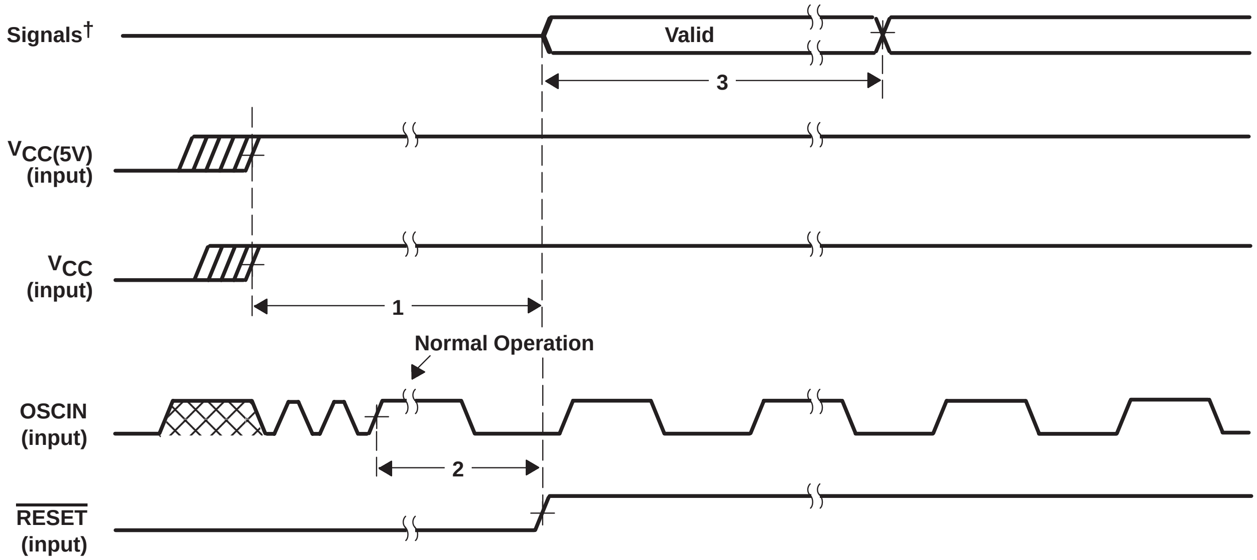


Figure 49. Ports 00–14 SNI NMON Interface Timing (10-Mbit/s mode)

timing requirements over recommended operating conditions (see Figure 50) power-on reset

NO.			TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{d(OSCIN)}$	Delay time, from V_{CC}↑ to $\overline{\text{RESET}}$↑	25		25		ms
2	$t_{d(RESET)}$	Delay time, from OSCIN↑ to $\overline{\text{RESET}}$↑	25		25		ms
3	$t_{d(AUTO)}$	Delay time, from $\overline{\text{RESET}}$↑ to EEPROM autoload complete		50		50	ms

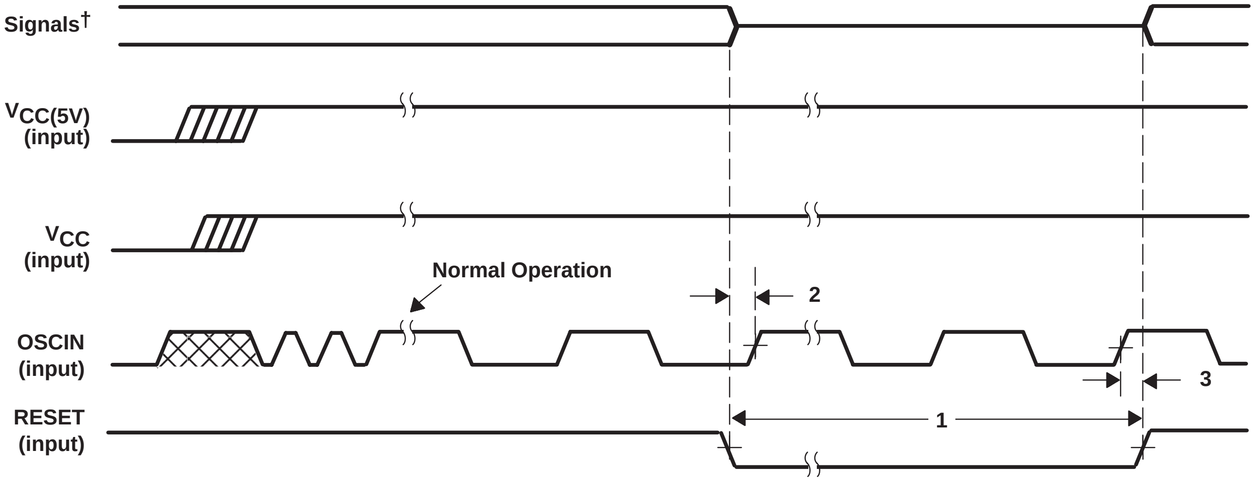


† MXXCOL, MXXCRS, MXXPROTOCOL, MXXDUPLEX, MXXLINK, MXXRCLK, MXXRXD0, MXXRXD1, MXXRXD2, MXXRXD3, MXXRXD4, M00RXD5, M00RXD6, M00RXD7, M00RXDV, M00RXDVX, MXXRXER, MXXSPEED, MXXTCLK, M00UPLINK, DD00–DD35, EAM15–EAM00, SDATA7–SDATA0, SAD1, SAD0, $\overline{\text{SCS}}$, SRNW, EDIO, TDI, TMS, $\overline{\text{TRST}}$, TCLK, OSCIN

Figure 50. Power-On Reset Timing

timing requirements over recommended operating conditions (see Figure 51) RESET (software) timing

NO.			TNETX3150		TNETX3150A		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{w(RESET)}$	Pulse duration, RESET low	3		3		µs
2	$t_{su(RESET)}$	Setup time, RESET low before OSCIN↑		7		7	ns
3	$t_{h(RESET)}$	Hold time, RESET low after OSCIN↑		10		10	ns



† MXXCOL, MXXCRS, MXXPROTOCOL, MXXDUPLEX, MXXLINK, MXXRCLK, MXXRXD0, MXXRXD1, MXXRXD2, MXXRXD3, MXXRXD4, M00RXD5, M00RXD6, M00RXD7, M00RXDV, M00RXDVX, MXXRXER, MXXSPEED, MXXTCLK, M00UPLINK, DD00–DD35, EAM15–EAM00, SDATA7–SDATA0, SAD1, SAD0, $\overline{\text{SCS}}$, SRNW, EDIO, TDI, TMS, $\overline{\text{TRST}}$, TCLK, OSCIN

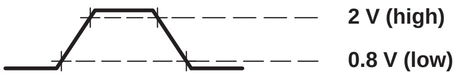
Figure 51. Reset (Software) Timing

PARAMETER MEASUREMENT INFORMATION

Outputs are driven to a minimum high-logic level of 2.4 V and to a maximum low-logic level of 0.6 V. These levels are compatible with TTL devices.

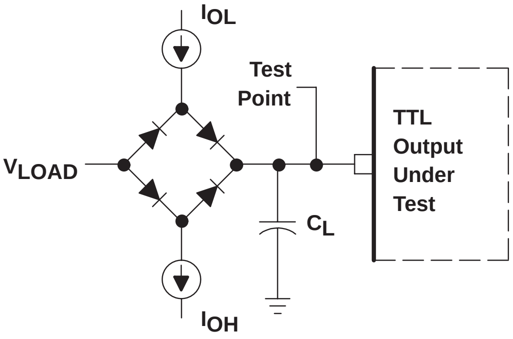
Output transition times are specified as follows: For a high-to-low transition on either an input or output signal, the level at which the signal is said to be no longer high is 2 V and the level at which the signal is said to be low is 0.8 V. For a low-to-high transition, the level at which the signal is said to be no longer low is 0.8 V and the level at which the signal is said to be high is 2 V, as shown in the following.

The rise and fall times are not specified but are assumed to be those of standard TTL devices, which are typically 1.5 ns.



test measurement

The test-load circuit shown in Figure 52 represents the programmable load of the tester pin electronics that is used to verify timing parameters of the TNETX3150/TNETX3150A output signals.



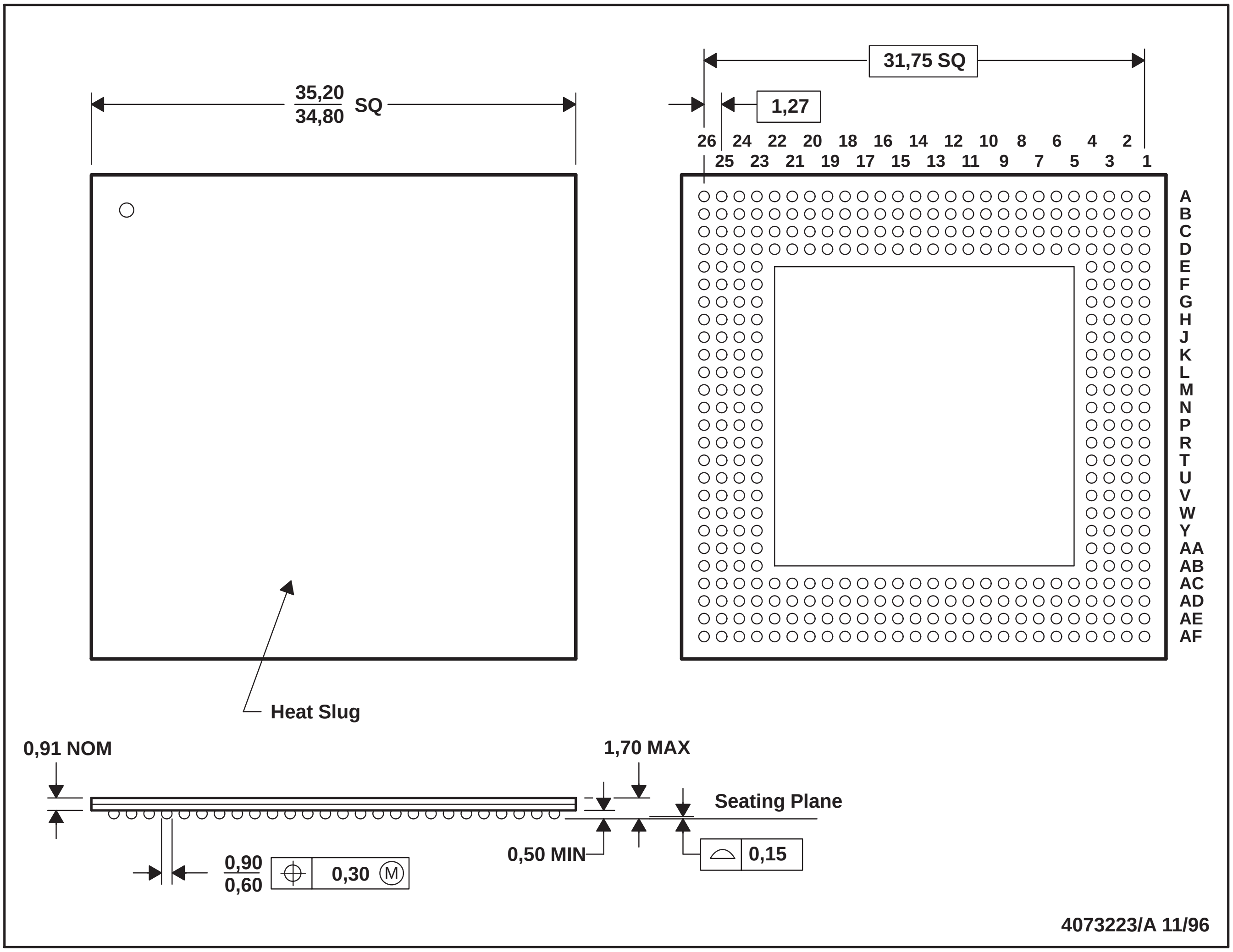
TTL OUTPUT TEST LOAD

Where:	I_{OH}	=	Refer to I_{OH} in recommended operating conditions.
	I_{OL}	=	Refer to I_{OL} in recommended operating conditions.
	V_{LOAD}	=	1.5 V, typical dc-level verification or 0.7 V, typical timing verification
	C_L	=	18 pF, typical load-circuit capacitance

Figure 52. Test and Load Circuit

MECHANICAL DATA

GGP (S-PBGA-N352) PLASTIC BALL GRID ARRAY (CAVITY DOWN) PACKAGE



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Thermally enhanced (die down) plastic package with top surface metal heat slug.

PACKAGING INFORMATION

Orderable Device	Status [1]	Package Type	Package Drawing	Pins	Package Qty	Eco Plan [2]	Lead/Ball Finish	MSL Peak Temp [3]
TNETX3150AGGP	OBSOLETE	BGA	GGP	352		TBD	Call TI	Call TI
TNETX3150GGP	OBSOLETE	BGA	GGP	352		TBD	Call TI	Call TI

[1] The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

[2] Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS) or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
TBD: The Pb-Free/Green conversion plan has not been defined.
Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

[3] MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265