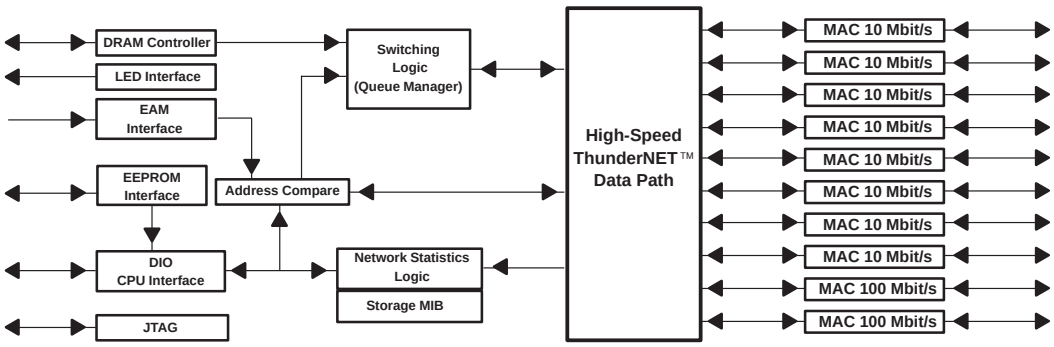
SPWS031D – JUNE 1997 – REVISED OCTOBER 1997

- **Highly Integrated Single-Chip Shared-Memory Switch**
- **Provides Eight 10-Mbit/s Ethernet Ports and Two 10-/100-Mbit/s Ethernet Ports**
- **Collision-Based Flow Control**
- **Supports Cut-Through and Store-and-Forward Modes of Switching Operation**
- **All Ports Are Full-Duplex Capable**
- **10-/100-Mbit/s Port Capable of Bidirectional Tagging**
- **Supports On-Chip Per-Port Storage for Etherstat™ and Remote Monitoring (RMON) for Simple-Network Management Protocol (SNMP)**
- **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**
- **Supports Multiple Media-Access Controller (MAC) Address Switching Modes**
- **Virtual-LAN (VLAN) Capable**
- **EEPROM Interface for Auto-Configuration, No CPU Needed, Enabling Lowest-Cost Switching Solution**
- **Provides External Address Match (EAM) Interface**
- **Packet Memory Utilizes Cost-Effective 60-ns Extended Data Out (EDO) DRAM**
- **Provides Direct Input/Output (DIO) Interface for Configuration and Statistics Information**
- **Provides Transmit Pacing**
- **LED Interface for Link Activity, Flow-Control Indication, and Port Status**
- **Fabricated in 3.3-V Low-Voltage Technology**
- **Packaged in 240-Pin Plastic Quad Flatpack**
- **5-V Tolerant I/Os**
- **JTAG Compliant**

description

The ThunderSWITCH TNETX3100 is a 10-port 10-/100-Mbit/s Ethernet switch. A low-cost solution is achieved by combining the TNETX3100 with physical interfaces and low-cost packet memory. The TNETX3100 provides 32 internal media access controller (MAC) address registers, collision-based flow control, virtual LAN (VLAN) support, and uplink support.

functional block diagram



description (continued)

The TNETX3100 provides two 100-Mbit/s interfaces and eight 10-Mbit/s interfaces. Half-duplex ports support collision-based flow control; the purpose of the flow control is to reduce the risk of data loss for a long burst of activity. With full-duplex capability, ports 02–09 can support 20-Mbit/s connections and ports 00–01 can support 200-Mbit/s connections to desktops or high-speed servers. The uplink (port 00) also can interface to an external routing engine. Besides the ten full-duplex ports, the TNETX3100 has three modes for making forwarding decisions.

The desktop switch internally supports multiple MAC addresses per port (32 total addresses). If more than 32 addresses are required, an external address match (EAM) interface is supplied to support multiple addresses per port. If an external routing engine is used, the uplink port supports frame tagging. The frame tagging provides source port information to an external routing engine. For more flexibility, the TNETX3100 can support VLAN for workgroup and segment-switching applications.

VLAN support is provided in two ways: 1) In the internal address 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.

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

The TNETX3100 utilizes an extremely low-cost 60-ns memory solution extended data out (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 TNETX3100 is fabricated with 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 a broad choice of interfacing-device options.

PGC PACKAGE
(TOP VIEW)

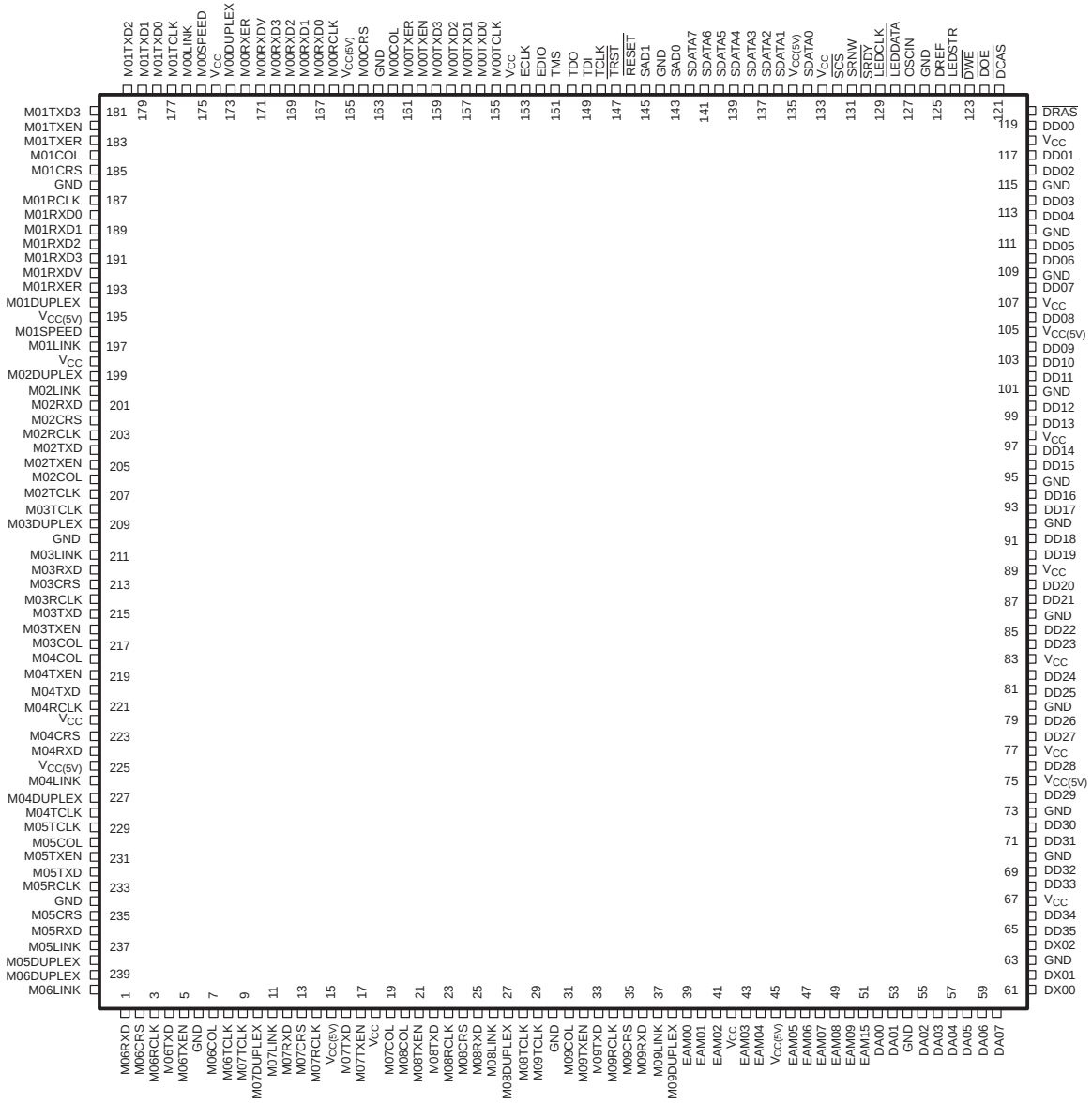


Table of Contents

Terminal Functions

10-Mbit/s MAC interface (ports 02–09)

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
M02COL	206	I	Collision sense. High = network collision and low = no network collision.
M03COL	217		
M04COL	218		
M05COL	230		
M06COL	7		
M07COL	19		
M08COL	20		
M09COL	31		
M02CRS	202	I	Carrier sense. High = frame carrier signal is received and low = no frame carrier signal is received.
M03CRS	213		
M04CRS	223		
M05CRS	235		
M06CRS	2		
M07CRS	13		
M08CRS	24		
M09CRS	35		
M02DUPLEX	199	I/O	Duplex selector. MXXDUPLEX switches the interface between full- and half-duplex operation (low= half duplex, high = full duplex). M02DUPLEX–M09DUPLEX inputs have active pulldown circuits used to force the inputs low when FORCEHD bit is set.
M03DUPLEX	209		
M04DUPLEX	227		
M05DUPLEX	238		
M06DUPLEX	239		
M07DUPLEX	10		
M08DUPLEX	27		
M09DUPLEX	38		
M02LINK	200	I	Link selector. MXXLINK indicates the condition of the port connection (low = no link, high = link).
M03LINK	211		
M04LINK	226		
M05LINK	237		
M06LINK	240		
M07LINK	11		
M08LINK	26		
M09LINK	37		

Terminal Functions (Continued)

10-Mbit/s MAC interface (ports 02–09) (continued)

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
M02RCLK M03RCLK M04RCLK M05RCLK M06RCLK M07RCLK M08RCLK M09RCLK	203 214 221 233 3 14 23 34	I	Receive clock. MXXRCLK is the receive clock from the attached PHY.
M02RXD M03RXD M04RXD M05RXD M06RXD M07RXD M08RXD M09RXD	201 212 224 236 1 12 25 36	I	Receive data. MXXRXD is the receive data from the PMD front end. Data is synchronous to MXXRCLK.
M02TCLK M03TCLK M04TCLK M05TCLK M06TCLK M07TCLK M08TCLK M09TCLK	207 208 228 229 8 9 28 29	I	Transmit clock. MXXTCLK is the transmit clock from the attached PHY.
M02TXD M03TXD M04TXD M05TXD M06TXD M07TXD M08TXD M09TXD	204 215 220 232 4 16 22 33	O	Transmit data. MXXTXD is the transmit data leaving port xx when MXXTXEN is asserted.
M02TXEN M03TXEN M04TXEN M05TXEN M06TXEN M07TXEN M08TXEN M09TXEN	205 216 219 231 5 17 21 32	O	Transmit enable. High = valid transmit data on MXXTXD and low = no valid transmit data on MXXTXD.

Terminal Functions (Continued)

10-/100-Mbit/s MAC interface (ports 00–01)

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
M00COL	162	I	Collision sense. High = network collision and low = no network collision.
M01COL	184		
M00CRS	164	I	Carrier sense. High = frame carrier signal is received and low = no frame carrier signal is received.
M01CRS	185		
M00DUPLEX	173	I/O	Duplex selector. MXXDUPLEX switches the interface between full- and half-duplex operation (low = half duplex, high = full duplex). The M00DUPLEX and M01DUPLEX inputs have active pulldown circuits used to force the inputs low when FORCEHD bit is set.
M01DUPLEX	194		
M00LINK	176	I	Link selector. MXXLINK indicates the condition of the port connection (low = no link, high = link).
M01LINK	197		
M00RCLK	166	I	Receive clock. MXXRCLK source is from the external PHY.
M01RCLK	187		
M00RXDV	171	I	Receive data valid. MXXRXDV indicates data on MXXRXD3–MXXRXD0 is valid.
M01RXDV	192		
M00RXER	172	I	Receive error. MXXRXER indicates reception of a coding error on received data.
M01RXER	193		
M00RXD0	167	I	Receive data. Nibble receive data MXXRXD3–MXXRXD0 is from the physical media-dependent front end. Data is synchronous to MXXRCLK. If MWIDTH bit in the port control register and M00SPEED terminal are set low, then data is received on M00RXD0 and M01RXD0 terminals.
M01RXD0	188		
M00RXD1	168		
M01RXD1	189		
M00RXD2	169		
M01RXD2	190		
M00RXD3	170		
M01RXD3	191		
M00SPEED	175	I	Bit-rate selection speed. The speed of the MAC interface is determined by the logic level on MXXSPEED (low = 10 Mbit/s, high = 100 Mbit/s).
M01SPEED	196		
M00TCLK	155	I	Transmit clock. Transmit clock source is from the external PHY.
M01TCLK	177		

Terminal Functions (Continued)

10-/100-Mbit/s MAC interface (ports 00–01) (continued)

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
M00TXD0	156	O	Transmit data. Nibble transmit data is from the TNETX3100. When MXXTXEN is asserted, these terminals carry transmit data. Data on these terminals is always synchronous to MXXTCLK. If MWIDTH bit in the port control register and M00SPEED terminal are set low, then data is transmitted on M00TXD0 and M01TXD0.
M01TXD0	178		
M00TXD1	157		
M01TXD1	179		
M00TXD2	158		
M01TXD2	180		
M00TXD3	159		
M01TXD3	181		
M00TXEN	160	O	Transmit enable. High = valid transmit data on MXXTXD and low = no valid transmit data on MXXTXD.
M01TXEN	182		
M00TXER	161	O	Transmit error. MXXTXER allows coding errors to propagate across the MII. This enables external logic to reconstruct and resend the frame. MXXTXER is asserted during an underrunning frame, enabling the TNETX3100 to force a coding error.
M01TXER	183		

DRAM interface

TERMINAL NAME†	TERMINAL NO.	I/O	DESCRIPTION
DA00	52	O	DRAM address bus. DA00–DA07 is time multiplexed with row and column address strobes.
DA01	53		
DA02	55		
DA03	56		
DA04	57		
DA05	58		
DA06	59		
DA07	60		
DX00	61	O	DRAM extended address lines. DX00–DX02 are time multiplexed with row and column address strobes.
DX01	62		
DX02	64		
$\overline{\text{DCAS}}$	121	O	DRAM column address select
$\overline{\text{DRAS}}$	120	O	DRAM row address select

† DD00 is the least-significant bit.

Terminal Functions (Continued)

DRAM interface (continued)

TERMINAL NAME†	TERMINAL NO.	I/O	DESCRIPTION
DD35	65	I/O	DRAM data bus. DD35–DD00 is bidirectional.
DD34	66		
DD33	68		
DD32	69		
DD31	71		
DD30	72		
DD29	74		
DD28	76		
DD27	78		
DD26	79		
DD25	81		
DD24	82		
DD23	84		
DD22	85		
DD21	87		
DD20	88		
DD19	90		
DD18	91		
DD17	93		
DD16	94		
DD15	96		
DD14	97		
DD13	99		
DD12	100		
DD11	102		
DD10	103		
DD09	104		
DD08	106		
DD07	108		
DD06	110		
DD05	111		
DD04	113		
DD03	114		
DD02	116		
DD01	117		
DD00	119		
$\overline{\text{DOE}}$	122	O	DRAM output enable
$\overline{\text{DWE}}$	123	O	DRAM write enable

† DD00 is the least-significant bit.

Terminal Functions (Continued)

EAM interface

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
EAM15 (mode select)	51	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	39	I	External address match. When EAM15 is low, a high on EAM00–EAM09 indicates the data frame is transmitted from that single-selected port. EAM00 is the least-significant bit and is associated with port 00.
EAM01	40		
EAM02	41		
EAM03	43		
EAM04	44		
EAM05	46		
EAM06	47		
EAM07	48		
EAM08	49		
EAM09	50		

DIO and statistics interface

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
SAD0	143	I	DIO address. SAD0–SAD1 selects the TNETX3100 host registers.
SAD1	145		
$\overline{\text{SCS}}$	132	I	DIO chip select. When low, $\overline{\text{SCS}}$ indicates a port access is valid.
SDATA0	134	I/O	DIO data. SDATA0–SDATA7 is bidirectional byte-wide data.
SDATA1	136		
SDATA2	137		
SDATA3	138		
SDATA4	139		
SDATA5	140		
SDATA6	141		
SDATA7	142		
$\overline{\text{SRDY}}$	130	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. TNETX3100 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 resistor.
SRNW	131	I	DIO read not write. When low, SRNW indicates a write cycle on the DIO port.

Terminal Functions (Continued)

EEPROM interface

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
ECLK	153	O	EEPROM data clock. ECLK is the serial EEPROM clock and requires an external pullup resistor.
EDIO	152	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. Reset is asynchronous, therefore, OSCIN needs to operate while RESET is active.

LED activity interface

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
LEDCLK	129	O	LED clock. LEDCLK is the serial-shift clock for the LED-status data.
$\overline{\text{LEDDATA}}$	128	O	LED data. $\overline{\text{LEDDATA}}$ is the active-low serial data signal for the LED-status data.
LEDSTR	124	O	LED strobe. LEDSTR pulses latch the contents of the LED shift registers.

JTAG interface

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
TCLK	148	I	Test clock. TCLK clocks state information and test data into and out of the device during operation of the test port.
TDI	149	I	Test data input. TDI serially shifts test data and test instructions into the device during operation of the test port.
TDO	150	O	Test data out. TDO serially shifts test data and test instructions out of the device during operation of the test port.
TMS	151	I	Test mode select. TMS controls the state of the TNETX3100 test-port controller.
$\overline{\text{TRST}}$	147	I	Test reset. $\overline{\text{TRST}}$ asynchronously resets the test-port controller.

miscellaneous functions

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
DREF	125	O	DRAM reference clock.
OSCIN	127	I	Oscillator input. OSCIN is the TNETX3100 clock input (50 MHz).
$\overline{\text{RESET}}$	146	I	Reset. $\overline{\text{RESET}}$ (active low) resets the TNETX3100. $\overline{\text{RESET}}$ should be held low for 25 ms after the power supplies and clocks have stabilized. This input is synchronous and therefore, the system clock is operational during reset.

Terminal Functions (Continued)

power interface

TERMINAL NAME	TERMINAL NO.	DESCRIPTION
GND	6, 30, 54, 63, 70, 73, 80, 86, 92, 95, 101, 109, 112, 115, 126, 144, 163, 186, 210, 234	Ground. GND is the 0-V reference for the device.
V_{CC}	18, 42, 67, 77, 83, 89, 98, 107, 118, 133, 154, 174, 198, 222	Supply voltage V_{CC} = 3.3 ± 0.3 V
$V_{CC(5V)}$	15, 45, 75, 105, 135, 165, 195, 225	Supply voltage. $V_{CC(5V)}$ = 5 ± 0.5V. $V_{CC(5V)}$ is only for the I/O periphery and is used for the 5-V tolerant pads only.

summary of signal terminals by signal group function

PORT DESCRIPTION	NUMBER OF SIGNALS	MULTIPLIER	TOTAL
LED	3	1	3
10-Mbit/s port	9	8	72
10-/100-Mbit/s port	19	2	38
DIO	13	1	13
EAM port	11	1	11
EEPROM interface	2	1	2
DRAM interface	51	1	51
Miscellaneous	3	1	3
JTAG	5	1	5
Total design			198

SUMMARY	
Assigned terminals	198
Power terminals $V_{CC(5V)}$, 5 V	8
Power terminals V_{CC}, 3.3 V	14
Power terminals GND	20
Total number of terminals	240

10-Mbit/s and 10-/100-Mbit/s MAC interfaces (ports 00–09)

The following section illustrates the similarities between the 10-Mbit/s and 10-/100-Mbit/s MAC interfaces. All network data flows through either the 10-Mbit/s interface or 10-/100-Mbit/s interface.

data reception

receive arbitration

Receive arbitration raises the priority of received frames over pending transmit. Active transmissions are given highest priority. This makes it less likely that frames will be lost during heavy traffic load. For correct TNETX3100 operation, the RXARB bit (bit 5) must be set in the SIO register.

receive control

Receiver control is partitioned into two blocks:

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

TNETX3100 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 TNETX3100 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 TNETX3100 can force the PHY into half-duplex operation when desired (during testing, for example). See Figure 1.

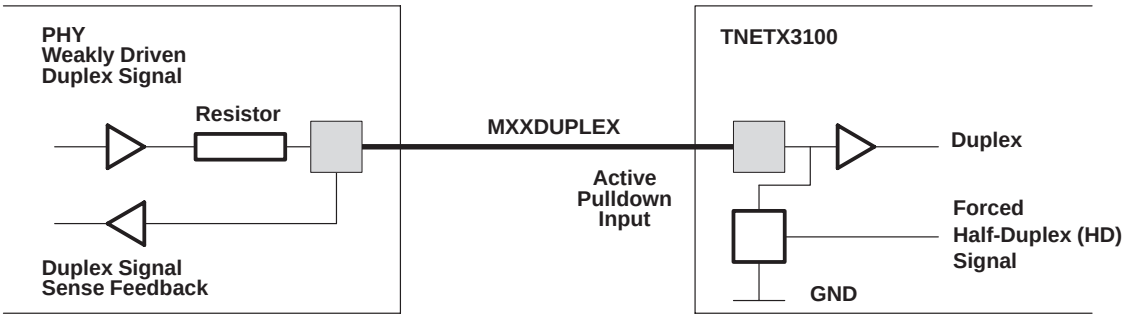


Figure 1. MXXDUPLEX Signal Connection

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 TNETX3100 forcing half duplex), a pullup resistor should be placed on the DUPLEX signal.

flow control

The TNETX3100 incorporates a collision-based flow-control mechanism; this mechanism is activated by setting the FLOW bit (bit 5) in the RAM size register to a 1. Flow control affects only the ports that are in half duplex.

The TNETX3100 detects that it is becoming congested by monitoring the size of the free-buffer queue. When there are fewer than 256 buffers on the free queue, the TNETX3100 prevents frames from entering the device by forcing a collision with all frames on all ports (except any frames that the TNETX3100 is transmitting). When the flow control is active, the FLOW LED is illuminated.

When the TNETX3100 has transmitted some of the backlog from the DRAM, and the free-buffer queue increases to 256 or more, forced collisions cease, and frames can again enter the device.

This simplified approach maximizes the use of the DRAM for the mix of port activity. The 256-buffer threshold ensures that all ports can receive and complete a maximum-length frame when there are 256 free buffers.

The purpose of flow control is to reduce the risk of data loss if a long burst of activity causes the TNETX3100 to backlog frames until the DRAM is full. Flow control has no mechanism to allow the TNETX3100 to operate on a sustained basis when the activity level exceeds the device internal bandwidth. Transmit queue length limiting is disabled if FLOW = 1.

giant (long) frames

The TNETX3100 can handle frames up to 1531 bytes to support IEEE Std 802.10. All frames longer than 1535 bytes are truncated by the TNETX3100. Frames greater than 1535 bytes received in cut-through mode are passed. Frames greater than 1535 bytes received in store-and-forward mode are discarded and not transmitted. The LONG option bit governs how the statistics for long frames are recorded. The byte count for long frames with good CRCs is 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 TNETX3100 within the RX FIFOs. They do not appear on the DRAM bus.

RX filtering of no-cut-through frames

The TNETX3100 filters RX frames that are in error (CRC, alignment, jabber, etc.) when operating in the 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.

data transmission

transmit control

Transmit control is partitioned into two blocks:

1. The frame control block handles the output of data into the PHYs, and 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.

transmit pacing

When transmit pacing is enabled, the ThunderSWITCH architecture is capable of altering its transmission routine during times of heavy network activity. The TNETX3100 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 stabilize before attempting transmission. If the delay was not added, the TNETX3100 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.

With pacing enabled, a frame is permitted (after one IPG) to immediately attempt transmission only if the pacing counter is zero. If the pacing counter is nonzero, the frame is delayed by the pacing delay (a delay of approximately four interframe gap delays).

interframe gap enforcement

The measurement reference for the interframe gap of 9.6 μs (when transmitting at 10 Mbit/s) is changed, depending on frame traffic conditions. If a frame is successfully transmitted (without collision), 9.6 μs is measured from MXXTXEN. If the frame suffered a collision, 9.6 μs is measured from MXXCRS. The TNETX3100 can receive frames with an interframe gap of less than 9.6 μs and it always transmits its frames with an interframe gap of 9.6 μs. The 100-Mbit/s ports can receive frames with an interframe gap of less than 0.96 μs and always transmit frames with a minimum interframe gap of 0.96 μs.

backoff

The TNETX3100 implements the IEEE Std 802.3 binary exponential backoff algorithm.

100-Mbit/s MAC interfaces (port 00)

The TNETX3100 allows port 00 to support pretagging and posttagging when forwarding frames to an external routing engine. Posttagging is used by an external routing engine to provide destination port information to the TNETX3100.

uplink TX pretagging to support an external routing engine

When forwarding frames to a routing engine, it is necessary to know which port received the frame. Multiple TNETX3100 devices can be incorporated into a larger switch fabric that is controlled by one routing engine. The TNETX3100 provides one nibble of information (to identify the source port) on the MII-interface data terminals prior to M00TXEN being asserted. See Figure 2.

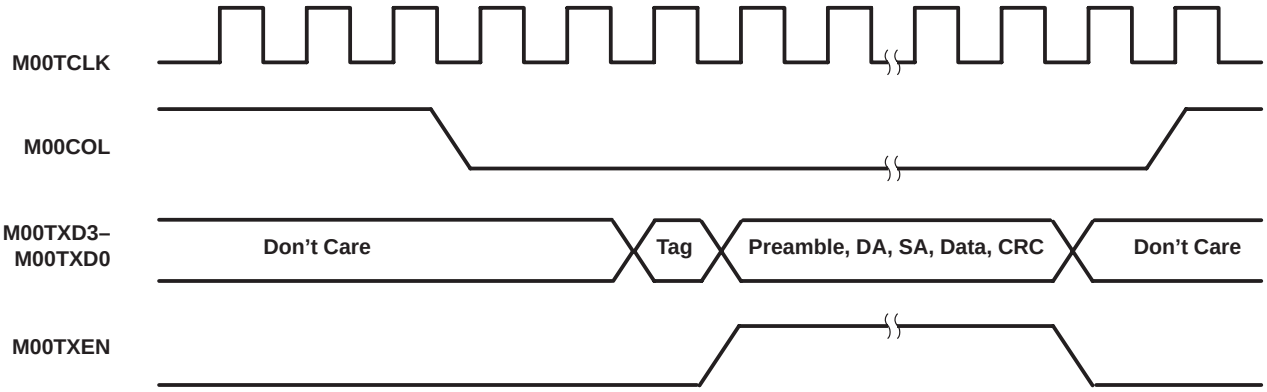


Figure 2. Multiplex Operation for Uplink-Port Transmit Frames

Port 00, when operated at 100-Mbit/s mode, provides a tag nibble one cycle prior to M00TXEN being asserted. The nibble format is shown in Table 1.

Table 1. Source Port Number Codes

SOURCE PORT NUMBER M00TXD3–M00TXD0	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	Reserved
1011	Reserved
1100	Reserved
1101	Reserved
1110	Reserved
1111	Reserved

uplink RX pretagging to support an external routing engine

When port 00 (uplink) is operated in 100-Mbit/s mode, a pretag is placed one cycle prior to MRXDV going high for the frame; tag data is applied to M00RXD3–M00RXD0. This pretag is not operated on internally by the TNETX3100 although it is used by an external address-lookup engine. The tag is passed to the DRAM interface, where it appears on bits 31–28 of the forward pointer; this provides the ability for an upstream device, connected to the uplink, to pass information through the TNETX3100 to the DRAM bus where an external address-lookup device (TNETX15VEPGE) operates on that information. The external address-lookup device makes a forwarding decision based on destination address and VLAN register value (see Figure 3).

The pretag is qualified with the port that received the frame. Only pretags received on port 00 are valid. A pretag received on the uplink (port 00) is retained for the duration of the frame.

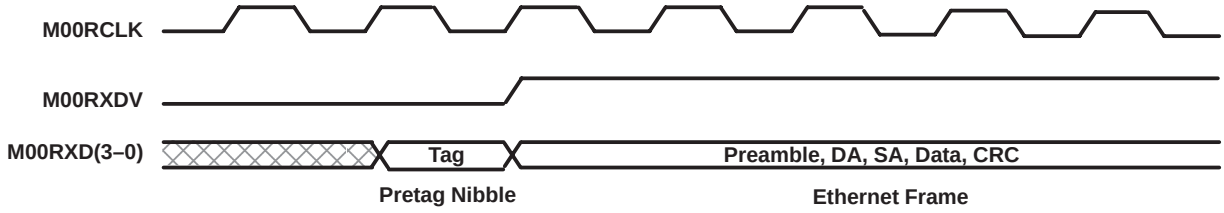


Figure 3. Data Format for Uplink RX Frames With Pretag

using TX pretagging and RX posttagging to support an external routing engine

If the TNETX3100 relies on the external routing engine to make switching decisions, no local switching is performed and port 00 forwards all frames to a routing engine; port 00 also provides a pretag on all forwarded frames. The pretag provides source-port information to the routing engine. The routing engine then makes forwarding decisions and transmits a frame and posttag to port 00; the posttag provides destination port information to the TNETX3100. To allow this, the TAGON bit (bit 4 of RSIZE register 0x00D2) must be a 1.

uplink RX posttagging to support an external routing engine

The external routing engine hardware must provide port 00 with an indication of the destination-port information; this information is used by the TNETX3100 to route frames to ports 01–09. This indication consists of three nibbles. When posttagging is enabled, the tag bytes overwrite the value received on the EAM interface (EAM interface has no effect). When posttagging is turned off (designed for local switching), the posttag 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 TNETX3100. If required, this must be implemented in upstream devices. No preamble is expected on data received by the uplink at 100 Mbit/s (see Figure 4). If RX posttagging is used, the receive side of port 00 must be configured for store-and-forward mode.

uplink RX posttagging to support an external routing engine (continued)

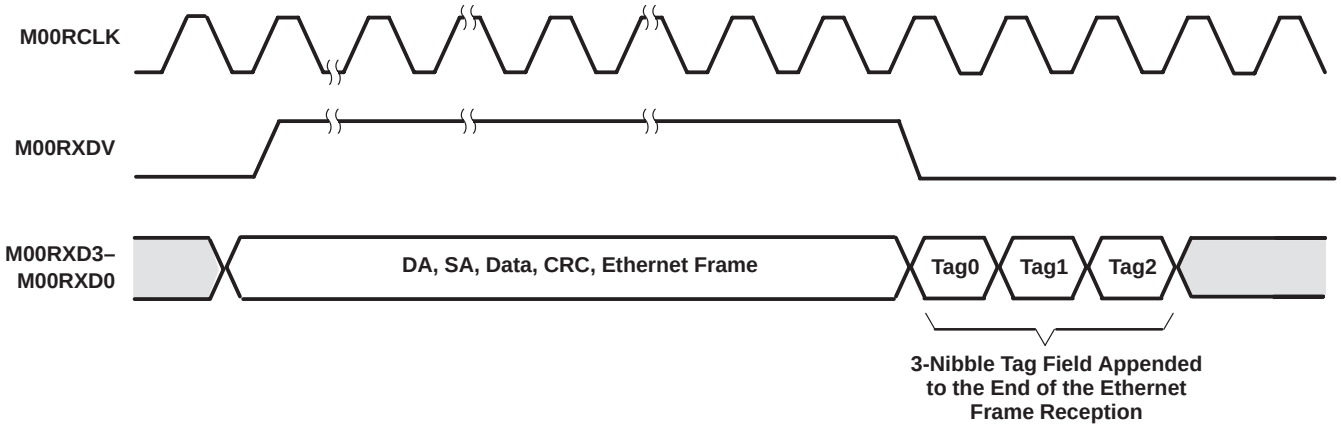


Figure 4. Data Format for Uplink RX Frames

uplink frame RX routing via posttagging

The tag fields are coded as shown in Figure 5:

tag 0

BIT 3	BIT 2	BIT 1	BIT 0
Destination Port 04 10 Mbit/s	Destination Port 03 10 Mbit/s	Destination Port 02 10 Mbit/s	Destination Port 01 10/100 Mbit/s

tag 1

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

tag 2

BIT 3	BIT 2	BIT 1	BIT 0
Reserved	Reserved	Reserved	Destination Port 09 10 Mbit/s

Figure 5. Tag Field Coding

If only one bit is set in the destination-port field, the packet is a unicast. With tag 0 = 0000, tag 1 = 0100, and tag 2 = 0000, the packet is unicast and destined for port 06.

If more than one bit is set, the packet is VLAN multicast. With tag 0 = 1010, and tag 1 = 1001, and tag 2 = 0001, the packet is transmitted from ports 02, 04, 05, 08, and 09. 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.

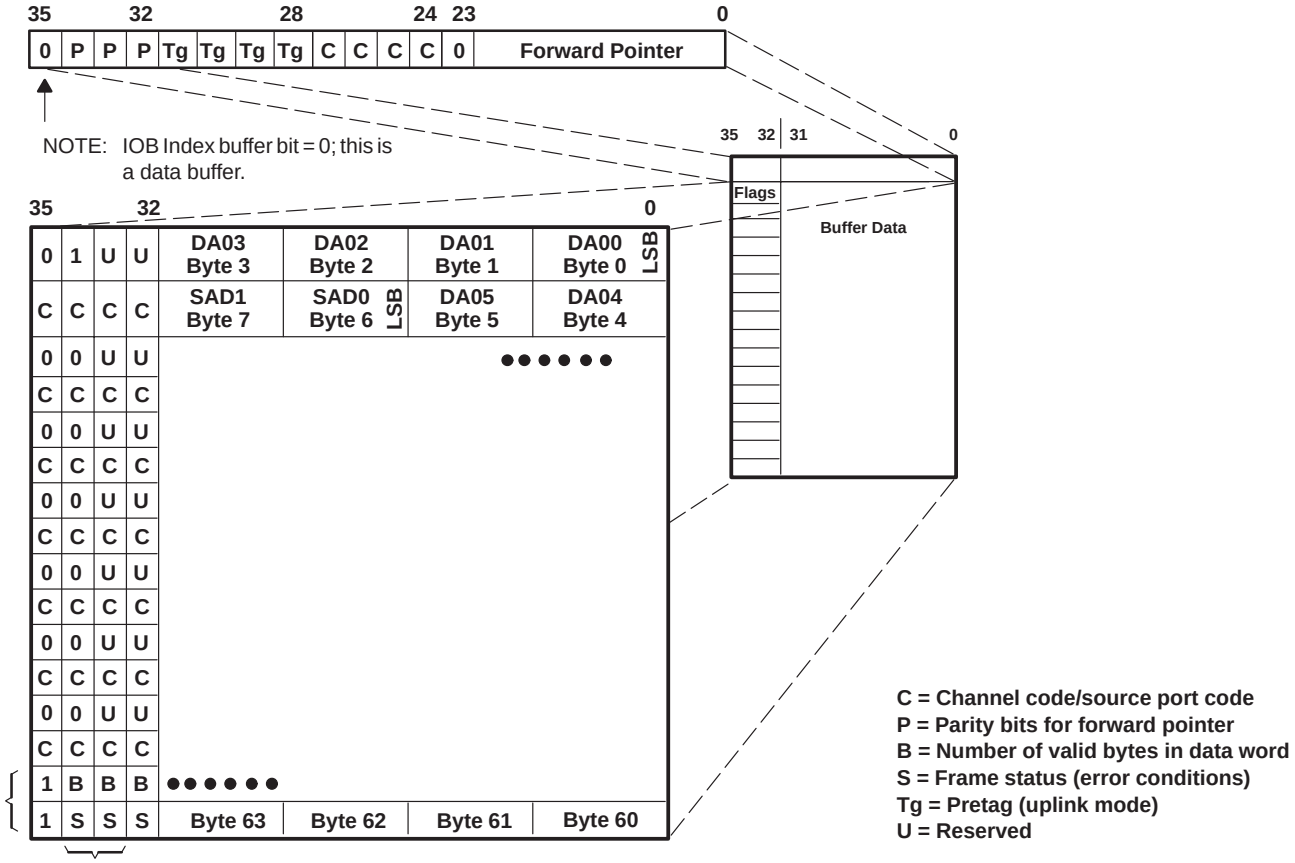
100-Mbit/s MAC interfaces (port 01)

This port is identical to the uplink port (port 00) except that it does not support posttagging. The port provides the pretag prior to TXEN going active. The tag is ignored by externally connected PHYs.

DRAM and EAM interfaces

All valid frames are passed across the DRAM interface (see Figure 8). 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 6). In conjunction with the DRAM column address strobe, external logic accesses the frame addresses and performs external address lookups 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 is zero to denote a start of frame. If it is high, the frame is an in-order broadcast (IOB) link buffer and not the start-of-data frame (bits 34, 33, and 32 contain parity information for the three forward pointers 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 21.

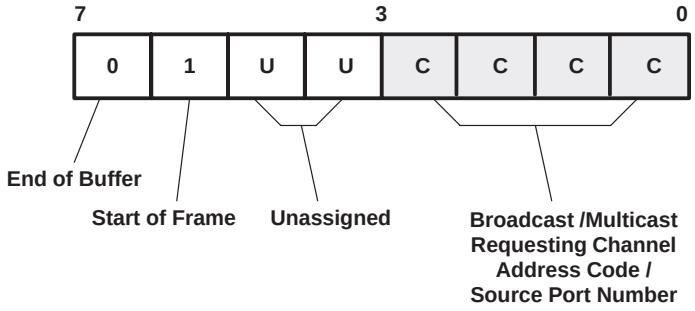


NOTE: Diagram shows end-of-frame flag format. Multiple buffer frames use this flag format only on the last buffer. (End-of buffer flags are used between a frame's buffers.)

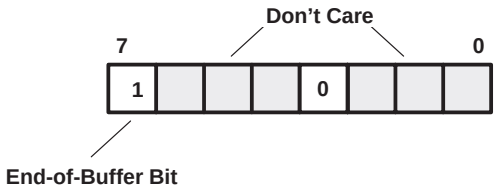
Figure 6. Buffer Diagram Showing Flag Format for a 64-byte Frame

flag byte

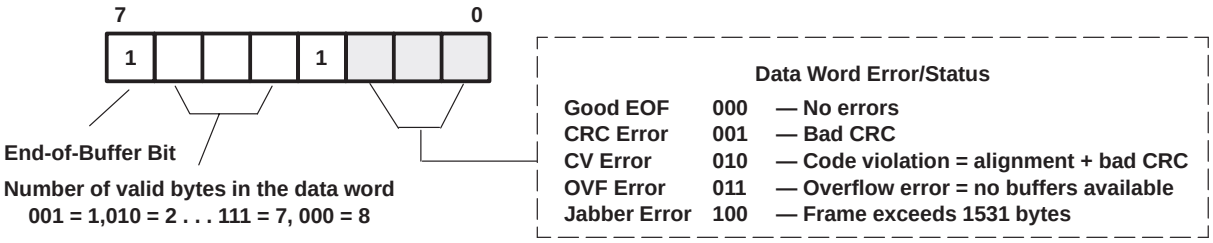
Flag attributes are assigned to the deserialized data word, identifying key attributes. The flags are used in later data handling. The flag field is assigned to every eight data bytes. The format of the subfields within the flag byte change, depending on the flag information. Previously, the start-of-frame format was described in the address compare text (see Figure 7). The formats shown in Figure 7 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



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

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

Figure 7. EOB, EOF, and SOF Flag Formats

DRAM and EAM interfaces (continued)

The TNETX3100 gives priority to the external channel address, over the internal-channel address-match information, to route the frame to the appropriate channel. To enable the EAM interface, the EAMON bit in the FEAGFG register must be set to a 1 (see Figure 8).

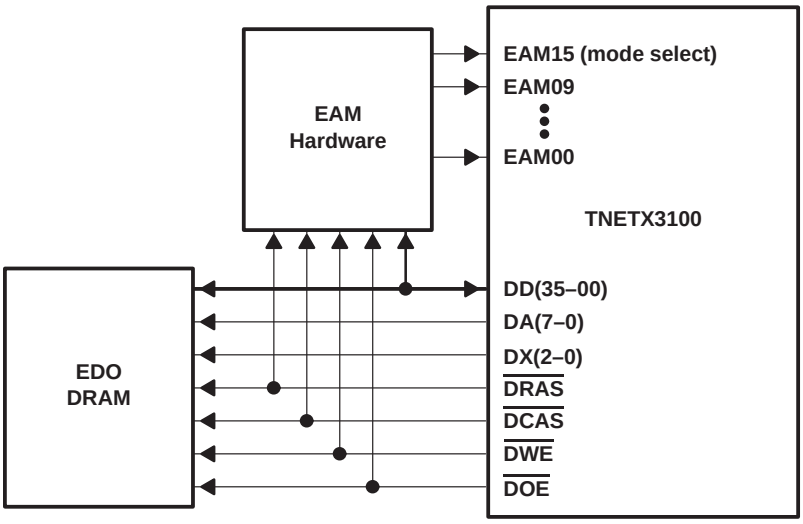


Figure 8. EAM Interface

Table 2 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 high, 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 are hardwired to 1 if no external device is present to disable this interface.
- With the EAM04 bit low and EAM15 bit (mode select) high, EAM03–EAM00 code is used to identify a single destination port.
- When the EAM15 bit (mode select) is low:
  - The other bits EAM09–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 09, the signal value is 0b1010000001.
  - To discard a frame, the external interface provides an all-zero code 0b0000000000).

All internal-address registers are disabled with the not-learn (NLEARN) bit (port-control register, bit 3), if the external device is used.

An external device has two choices of codes with which to forward a frame to a single port. For example, to forward a frame to port 3, one can enter 0x004 (multiport capable with only one port specified), or 0x403 (single-port capable only). While using the first method can make specifying unicast and multicast forwarding more unified, it forces the TNETX3100 to use internal cycles to build a broadcast table with only one entry, and thereby unicast traffic performance is reduced. Unicast forwarding is done with the EAM15 high, and the port specified in the lower four bits.

DRAM and EAM interfaces (continued)

Table 2. EAM Port Codes

TNETX3100 PORT	EAM15 MODE SELECT	EAM09–EAM05	EAM04	EAM03–EAM00
Port 00 (10/100 Mbit/s)	1	xxxxx	0	0000
Port 01 (10/100 Mbit/s)	1	xxxxx	0	0001
Port 02 (10 Mbit/s)	1	xxxxx	0	0010
Port 03 (10 Mbit/s)	1	xxxxx	0	0011
Port 04 (10 Mbit/s)	1	xxxxx	0	0100
Port 05 (10 Mbit/s)	1	xxxxx	0	0101
Port 06 (10 Mbit/s)	1	xxxxx	0	0110
Port 07 (10 Mbit/s)	1	xxxxx	0	0111
Port 08 (10 Mbit/s)	1	xxxxx	0	1000
Port 09 (10 Mbit/s)	1	xxxxx	0	1001
Reserved	1	xxxxx	0	1010
Reserved	1	xxxxx	0	1011
Reserved	1	xxxxx	0	1100
Reserved	1	xxxxx	0	1101
Reserved	1	xxxxx	0	1110
Broadcast channel	1	xxxxx	0	1111
No operation	1	xxxxx	1	xxxx
Frame discard	0	00000	0	0000
Bit-map mode	0	EAM9–EAM0 = port destination bit-map		

unmatched frames

When TNETX3100 receives a frame and the destination address is not matched by the address-compare unit, then it is processed in a manner that is dependent on the operating mode of the TNETX3100. This processing is defined in Table 3.

Table 3. Unmatched Frame Processing

EAM CODE	LIMLRN	PORT 00 NLEARN	BRUN	RECEIVE PORT†									
				00	01	02	03	04	05	06	07	08	09
Other	X	X	X	Per EAM code									
No operation	0	X	0	D	0	0	0	0	0	0	0	0	0
No operation	0	X	1	V	V	V	V	V	V	V	V	V	V
No operation	1	0	0	D	D	D	D	D	D	D	D	D	D
No operation	1	0	1	V	V	V	V	V	V	V	V	V	V
No operation	1	1	0	D	0	0	0	0	0	0	0	0	0
No operation	1	1	1	D	V	V	V	V	V	V	V	V	V

† 0 = Transmit on port 00
D = Discard
V = Transmit using VLAN register of receiving port

VLAN support

When EAM15 (mode select) is low, the EAM09–EAM00 inputs provide a mechanism for the EAM interface to specify which destination port or group of destination ports is used to transmit the frame. Each signal represents one destination port. Asserting one signal sends the frame to one destination port and asserting multiple signals transmits the same frame to multiple ports. When using an external address/VLAN engine, the external VLAN engine controls the EAM interface and the TNETX3100 forwards broadcast/multicast traffic to the configured VLAN ports. This mode of operation employs the broadcast mechanism and appends the frames to the transmit queues of the destination ports. The broadcast mechanism is an inefficient way to send frames to single ports when individual port codes can be used.

For the internal address mode, the TNETX3100 provides one 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 then sent only to the ports that are a part of the same VLAN.

DIO interface

The DIO interface allows host access to the TNETX3100. The DIO interface provides access to the on-chip registers and statistics.

The DIO provides the following information:

- Network statistics counters – provide access to the network statistics information compiled in the statistics RAM.
- System configuration registers – set or change the operation of the TNETX3100.
- RAM access – permits test access, allowing functional testing.
- Port registers – access port control, port status, and port address registers, permitting port management and status interrogation.

To reduce design overheads and to simplify interface logic, a byte-wide asynchronous interface is defined. Access to the internal TNETX3100 registers is available, indirectly, via the TNETX3100 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 4).

Table 4. SAD1 and 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

DIO interface (continued)

Data can be read or written to the address registers using the data lines SDATA7–SDATA0 under the control of chip select ($\overline{SCS}$), read not write (SRNW) and ready ($\overline{SRDY}$) signals. The physical interface is shown in Figure 9. The DIO interface timing is shown in Figure 19 and Figure 20. (See *DIO register* description.)

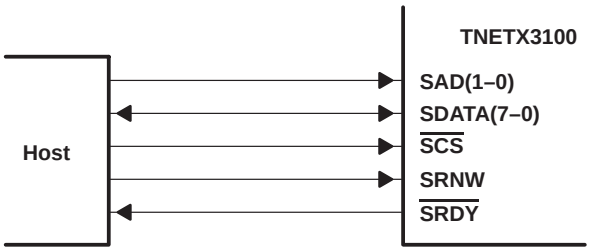


Figure 9. DIO Interface Signals

remote hardware reset

A hardware reset can be initiated by the DIO interface, permitting remote recovery of the TNETX3100 without requiring external glue logic to decode a DIO address and toggle the external reset signal of the TNETX3100. A hard DIO reset is initiated by writing to address 0x4000. This address is directly decoded by the TNETX3100 and forces a hardware reset.

TNETX3100 condensed DIO memory map

Table 5 shows the DIO memory map. Each section is expanded in detail.

Table 5. TNETX3100 Condensed DIO Memory Map

DIO ADDRESS (BITS 15–0) SHOWN IN HEX FORMAT	DESCRIPTION	AREA OF TNETX3100
0000–007F	Network address registers 16–31†	Network
0080–0093	Port 00–09 control and status registers†	Configuration
0094–009F	Reserved†	Reserved
00A0	Revision register†	Configuration
00A1	SIO/SCTRL register†	Configuration
00A2	Reserved†	Reserved
00A3	Device code†	Configuration
00A4–00B7	Port 00–09 transmit queue length†	Configuration
00B8–00CB	Port 00–09 VLAN registers†	Configuration
00CC–00CE	Free queue-length register†	Configuration
00CF	Reserved†	Configuration
00D0	FEACFG register†	Configuration
00D1	LED register†	Configuration
00D2	RAM size register†	Configuration
00D3	System-control register†	Configuration
00D4–00D7	DRAM data	Internal test
00D8	DRAM flag	Internal test
00D9–00DB	DRAM address	Reserved

† These registers are automatically loaded from the EEPROM.

TNETX3100 condensed DIO memory map (continued)

Table 5. TNETX3100 Condensed DIO Memory Map (Continued)

DIO ADDRESS (BITS 15–0) SHOWN IN HEX FORMAT	DESCRIPTION	AREA OF TNETX3100
00DC	Reserved	Internal test
00DD	DIATST	Internal test
00DE	PACTST	Internal test
00DF–00FF	Reserved	Reserved
0100–017F	Network address registers 00–15	Network
0180–3FFF	Reserved	Reserved
4000–7FFF	Hardware reset if written to	Internal test
8000–84FF	Port 00–09 statistics	Statistics RAM
8500–854F	Port 00–09 collisions and RX over runs	Statistics RAM
8550–85BF	Reserved	Statistics RAM
85C0–85D7	Port 09 TXQ, IMQ, and RXQ registers	Statistics RAM
85D8–85DF	Reserved	Statistics RAM
85E0–85F7	Port 08 TXQ, IMQ, and RXQ registers	Statistics RAM
85F8–85FF	Reserved	Statistics RAM
8600–8617	Port 07 TXQ, IMQ, and RXQ registers	Statistics RAM
8618–861F	Reserved	Statistics RAM
8620–8637	Port 06 TXQ, IMQ, and RXQ registers	Statistics RAM
8638–863F	Reserved	Statistics RAM
8640–8657	Port 05 TXQ, IMQ, and RXQ registers	Statistics RAM
8658–865F	Reserved	Statistics RAM
8660–8677	Port 04 TXQ, IMQ, and RXQ registers	Statistics RAM
8678–867F	Reserved	Statistics RAM
8680–8697	Port 03 TXQ, IMQ, and RXQ registers	Statistics RAM
8698–869F	Reserved	Statistics RAM
86A0–86B7	Port 02 TXQ, IMQ, and RXQ registers	Statistics RAM
86B8–86BF	Reserved	Statistics RAM
86C0–86D7	Port 01 TXQ, IMQ, and RXQ registers	Statistics RAM
86D8–86DF	Reserved	Statistics RAM
86E0–86F7	Port 00 TXQ, IMQ, and RXQ registers	Statistics RAM
86F8–BFFF	Reserved	Statistics RAM
C000–C1FF	Port 00 RX FIFO	FIFO RAM
C200–C3FF	Port 00 TX FIFO	FIFO RAM
C400–C5FF	Port 01 RX FIFO	FIFO RAM
C600–C7FF	Port 01 TX FIFO	FIFO RAM
C800–C9FF	Port 02 RX FIFO	FIFO RAM
CA00–CBFF	Port 02 TX FIFO	FIFO RAM
CC00–CDFF	Port 03 RX FIFO	FIFO RAM
CE00–CFFF	Port 03 TX FIFO	FIFO RAM
D000–D1FF	Port 04 RX FIFO	FIFO RAM
D200–D3FF	Port 04 TX FIFO	FIFO RAM

TNETX3100 condensed DIO memory map (continued)

Table 5. TNETX3100 Condensed DIO Memory Map (Continued)

DIO ADDRESS (BITS 15–0) SHOWN IN HEX FORMAT	DESCRIPTION	AREA OF TNETX3100
D400–D5FF	Port 05 RX FIFO	FIFO RAM
D600–D7FF	Port 05 TX FIFO	FIFO RAM
D800–D9FF	Port 06 RX FIFO	FIFO RAM
DA00–DBFF	Port 06 TX FIFO	FIFO RAM
DC00–DDFF	Port 07 RX FIFO	FIFO RAM
DE00–DFFF	Port 07 TX FIFO	FIFO RAM
E000–E1FF	Port 08 RX FIFO	FIFO RAM
E200–E3FF	Port 08 TX FIFO	FIFO RAM
E400–E5FF	Port 09 RX FIFO	FIFO RAM
E600–E7FF	Port 09 TX FIFO	FIFO RAM
E800–FFFF	Reserved (not implemented)	

address registers at [0x08*(A–16)] + 0x0000 through [0x08*(A–16)] + 0x0007 (A = address register in hex, legal values are 16–31)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3 (MSB)	BIT 2	BIT 1	BIT 0 (LSB)	DIO ADDRESS
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	LOCK†	0x08(A-16)
Reserved	Reserved	Reserved	Reserved	Port Number	Port Number	Port Number	Port Number	0x08(A–16)+0x01
Address Bit 47	Address Bit 46	Address Bit 45	Address Bit 44	Address Bit 43	Address Bit 42	Address Bit 41	Address Bit 40	0x08(A–16)+0x02
Address Bit 39	Address Bit 38	Address Bit 37	Address Bit 36	Address Bit 35	Address Bit 34	Address Bit 33	Address Bit 32	0x08(A–16)+0x03
Address Bit 31	Address Bit 30	Address Bit 29	Address Bit 28	Address Bit 27	Address Bit 26	Address Bit 25	Address Bit 24	0x08(A–16)+0x04
Address Bit 23	Address Bit 22	Address Bit 21	Address Bit 20	Address Bit 19	Address Bit 18	Address Bit 17	Address Bit 16	0x08(A–16)+0x05
Address Bit 15	Address Bit 14	Address Bit 13	Address Bit 12	Address Bit 11	Address Bit 10	Address Bit 9	Address Bit 8	0x08(A–16)+0x06
Address Bit 7	Address Bit 6	Address Bit 5	Address Bit 4	Address Bit 3	Address Bit 2	Address Bit 1	Address Bit 0	0x08(A–16)+0x07

† LOCK is present for network address register 0; for all other network address registers (1–15), LOCK is reserved.

address register description

The 16 network registers shown are combined with the other 16 registers (address 0100 through 017F) for a total of 32 network address registers; they form a 32-deep push-down stack, and this stack holds the last 32 source addresses seen by the TNETX3100. When a frame is received by the TNETX3100, the source address is extracted and pushed onto the top of the stack (register 0) along with the source port number, and the current contents of the stack, pushed down one location. The oldest address that was at the bottom of the stack (register 31) is discarded.

address register description (continued)

The address-learning operation is disabled on a per-port basis by setting the NLEARN bit in the port control register. When a frame is received by a port that has its NLEARN bit set, the source address is not pushed onto the stack.

To prevent multiple copies of the same address filling the stack, the push-down mechanism is modified; if the stack already contains an entry with the same network address (possibly with a different port number), then the registers below that location are protected from being overwritten.

Addresses that are seen repeatedly remain near the top of the stack and those that are seen infrequently work their way toward the bottom of the stack as new addresses are required, until they are finally aged out by being pushed off the end of the stack.

When the TNETX3100 is reset, the stack is initialized such that all entries have a network address of zero and a port number of 15; this identifies the entry as invalid and no address matching can be performed with such an entry.

The address-learning capability of the uplink (port 00) is restricted by setting the LIMLRN bit of the FEACFG register to a 1. In this mode, port 00 learns addresses only from frames it receives if the destination address has already been learned (currently in the address table). The learning capability of the port is still disabled if the ports NLEARN bit is set.

secured address

The TNETX3100 can secure addresses to specified ports. When an address is secured to a specified port, that source address is used only with that port. Use of a secured address on a port to which it is not assigned causes the received frame to be discarded.

An address is considered secure if the least-significant bit of that address is a 1. When an address is pushed onto the stack, the least-significant bit (G/S bit) is discarded and replaced by a zero. This causes the learned address to be unsecured.

A secured address entry in the stack modifies the push-down mechanism such that the secured address entry and all those below it are protected from being overwritten. When secured addresses are at the bottom end of the stack, this decreases the depth of the stack of learned addresses and prevents the secured addresses from being aged out.

A secured address prevents that source address from being used on another port. If the source address is secured to another port, then the frame is discarded and if the SECDIS bit is set, the receiving port enters the *disabled due to address duplication* state.

DIO access to address registers

The network registers must be locked to prevent the registers from changing. LOCK is set before reading or writing to the address registers. When LOCK = 1, no new addresses become learned addresses by the TNETX3100, but the TNETX3100 continues to route frames based upon the network registers.

After reset, the address registers are zero. LOCK is present only for network address register 0 and register 16; LOCK is reserved for all other network registers (1–15, 17–31).

port-N control register at 0x2*N + 0080 (N = port number in hex)

RESET CONDITIONS	INITIAL VALUES AFTER RESET†							
	BIT							
	7	6	5	4	3	2	1	0
	DISABLE	ENABLE	STFORTX	STFORRX	NLEARN	MWIDTH	TXPACE	FORCEHD
Port 00 (initial value after reset)	0	0	0	0	1	0	0	0
Ports 01–09 (initial value after reset)	0	0	0	0	0	0	0	0

† This register is reset to initial values after any reset condition and must be reloaded.

BIT	NAME	FUNCTION
7	DISABLE	Port disable. DISABLE = 1 disables the port. Frames are not forwarded from or to a disabled port, but the port attempts to transmit any previously queued frames. Latched bit. DISABLE is set by both hard and soft reset (default state is port disabled). DISABLE is cleared by writing ENABLE = 1.
6	ENABLE	Port enable. ENABLE = 1 enables the port, provided DISABLE = 1 is not written at the same time. 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	NLEARN	Not learn. When NLEARN = 1, the port does not take part in address-learning activity. Any stored addresses remain valid until they are aged out, and secured addresses are never aged out. When NLEARN = 0, the port takes part in address-learning activity.
2	MWIDTH	MII width selection. MWIDTH is only valid on 10-/100-Mbit/s capable ports (ports 00, 01). When MWIDTH = 1, and the port is operated in the 10-Mbit/s mode, the interface is operated in nibble-serial mode. When MWIDTH = 0, the interface is operated in the 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 0x2*N + 0x0081 (read only) (N = port number in hex)

INITIAL VALUES AFTER RESET							
BIT							
7	6	5	4	3	2	1	0
UPDATE	NLINK	Reserved	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 is 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 (after this port has transmitted a frame). UPDATE is maintained only for ports with the LINK signal active. Any port with link down is not updated.
6	NLINK	Not link. NLINK indicates that the port's link is inactive. NLINK = 1, when the port does not have a valid link. NLINK = 0, when the port has a valid link. NLINK reports the inverse of the state of the port's MXXLINK terminal.
5	Reserved	Reserved. Not implemented (this bit is always read as zero).
4	SPEED	Network speed. SPEED indicates the speed of a network port. When SPEED = 1, it indicates 100 Mbit/s. When SPEED = 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 DUPLX = 1, it indicates full duplex. When DUPLX = 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 – Reserved 011 – Reserved 100 – Disabled by management 101 – Reserved 110 – Disabled due to address duplication 111 – Reserved 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 the port DISABLE bit is set.

port-N status register at 0x2*N + 0x0081 (read only) (N = port number in hex) (continued)

PORT STATE	FUNCTION
000	Enabled. This is the normal state of a port. This is the only port state during which frames are forwarded to the port. In all other states, no new frames are forwarded to 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 indicate that there is no station attached to the port. The port is reenabled when link activity is detected at the port, indicated by an active (1) state of the port's MXXLINK terminal. If link is lost during transmission of a frame, transmission continues until the start of the next frame (the transmitted frame is lost). If link is lost during transmit, all enqueued frames are transmitted.
010	Reserved
011	Reserved
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 reenabled 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 is reenabled only by writing a 1 to the ENABLE control bit.
111	Reserved

revision register at 0x00A0

BIT							
7	6	5	4	3	2	1	0
Revision							

BIT	NAME	FUNCTION
7–4	Revision	Hardware revision code for this device. This field is read only. Revision 0.0 devices have a revision code of 0x00. Bits 7–4 denote the major digit.
3–0	Revision	Hardware revision code for this device. This field is read only. Revision 0.0 devices have a revision code of 0x00. Bits 3–0 denote the minor digit.

SIO/XCTRL register at 0x00A1

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

BIT	NAME	FUNCTION
7	Reserved	Reserved. Not implemented (reads as zero).
6	CUT100	Single-buffer cut-through operation on 100-Mbit/s ports only. CUT100 = 1 disables single-buffer cut-through operation for frames received on 10-Mbit/s source ports. A frame is transmitted only when two buffers are transferred to the transmit FIFO or an end of frame (prior to two buffers) is received. Although it increases latency on 10-Mbit/s ports, enabling CUT100 reduces the probability of dropping frames due to FIFO underrun on 100-Mbit/s ports. When CUT100 = 0, all ports operate in single-buffer cut-through mode.
5	RXARB	Receive arbitration mode. RXARB = 1 ensures that no frames are dropped during bursty conditions. When RXARB = 0, errored frames can be transmitted. To ensure proper operation, set RXARB to 1.
4	BRUN	Broadcast to unassigned ports. If no port address is matched, and BRUN = 1, the TNETX3100 broadcasts a unicast frame to all ports with unassigned addresses (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 provides the function by itself.
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, ECLOK is always read as a 0, even if a value of 1 is written to the bit. The TNETX3100 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), 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.

device code register at 0x00A3 (DIO)

BIT							
7	6	5	4	3	2	1	0
Device Identification Code							
0	0	0	0	0	0	0	1

BIT	NAME	FUNCTION
7–0	DEVCODE	Device type code for the device. The DEVCODE field is read only and identifies the type of ThunderSWITCH device. The standard ThunderSWITCH (TNETX3150) has a code of 0x00. Tiny ThunderSWITCH (TNETX3100) has a code of 0x01.

transmit queue-length registers at 0x2*N + 0x00A4 through 0x2*N + 0x00A5

+1	+0	DIO ADDRESS
Transmit queue N length		0x2*N + 0x00A4

Transmit queues use a residual queue length to control their behavior. The 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 is read from the EEPROM. The first value written to the transmit queue-length registers is the initial value. When you write 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. If the transmit queue length becomes negative (most-significant bit is 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 consumes 23 buffers more than the initial residual length (i.e., if the transmit queue is set to length = 1, a full-size Ethernet frame can still 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 completed. The operation is a 16-bit addition, allowing the current queue length to increase or decrease. The update operation is enabled only when the most-significant byte of the register (bits 15–8) is written to prevent possible length corruption. Length bytes are always written with the least-significant byte first.

port VLAN registers at 0x2*N + 0x00B8 through 0x2*N + 0x00B9 (N = port number in hex)

The VLAN registers hold broadcast destination masks for each source port. Each bit in the VLAN register (except bits 15–10) directly corresponds to a port (bit 9 = port 09 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 6).

Table 6. VLAN Register Initial Values

REGISTER NAME	INITIAL VALUE		DIO ADDRESS
	+1	+0	
VLAN 0 MASK	00000011	11111110	0x00B8
VLAN 1 MASK	00000011	01111101	0x00BA
VLAN 2 MASK	00000011	11111011	0x00BC
VLAN 3 MASK	00000011	11110111	0x00BE
VLAN 4 MASK	00000011	11101111	0x00C0
VLAN 5 MASK	00000011	11011111	0x00C2
VLAN 6 MASK	00000011	10111111	0x00C4
VLAN 7 MASK	00000011	01111111	0x00C6
VLAN 8 MASK	00000010	11111111	0x00C8
VLAN 9 MASK	00000001	11111111	0x00CA

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

free queue-length register at 0x00CC–0x00CE

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	DIO ADDRESS
Free queue length (7–0)								0x0CC
Free queue length (15–8)								0x0CD
Reserved						Free queue length (17–16)		0x0CE

This register contains the number of buffers currently on the free queue and is set to the total number of buffers in the system at device initialization. The user should always read 0x0CC first; reading this location causes the register contents to transfer to a holding latch to prevent the value from changing while being read. Writes to this register have no effect.

feature-configuration register (FEACFG) at 0x00D0

BIT							
7	6	5	4	3	2	1	0
Reserved				TXAIS	LIMLRN	EAMON	TPHY
Initial Values After Reset							
X	X	X	X	0	0	0	0

BIT	NAME	FUNCTION
7–4	Reserved	Reserved
3	TXAIS	Transmit activity indication select. When TXAIS = 1, the port LED indicates all RX and all TX activity. When TXAIS = 0, the LEDs indicate unicast TX-frame activity and all RX-frame activity.
2	LIMLRN	Limit learning. When LIMLRN = 1, the address-learning capability of the uplink (port 00) is restricted. In this mode, port 00 learns addresses from frames it receives if the destination address already has been learned (currently in the address table). If NLEARN = 0 in the port 00 control register, LIMLRN has no effect.
1	EAMON	EAM enable bit. When EAMON = 1, the TNETX3100 uses EAM hardware; when EAMON = 0, the EAMXX terminals are ignored.
0	Reserved	Reserved

LED control register at 0x00D1

BIT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		FLTLED	PONLY	S4LED	S3LED	S2LED	S1LED
Initial Values After Reset							
0	0	0	0	0	0	0	0

BIT	NAME	FUNCTION
7–6	Reserved	Reserved
5	FLTLED	Fault LED. When FLTLED = 1, the fault LED is turned on. When FLTLED = 0, the fault LED turns on if an invalid CRC is detected during the EEPROM load.
4	PONLY	Partition only. If PONLY = 1, the port LEDs display the enable/disable status of the ports; the display of port activity and link status is disabled.
3–0	S4LED S3LED S2LED S1LED	Software-controlled LEDs. S4LED–S1LED control the state of the external software-controlled LEDs; when SxLED = 1, the corresponding LED is turned on, when SxLED = 0, the LED is off. The TNETX3100 makes no use of the value of these bits other than for controlling the external LEDs.

RAM size register at 0x00D2

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

BIT	NAME	FUNCTION
7	MTEST	Test access bit. Before DIO accesses are made to the DIO test registers (address range 0x00DC–0x00FF), the user must write MTEST = 1. When an EEPROM is used to initialize registers 0x00–0xC3, MTEST is initialized to 0. MTEST allows access to the DIO registers to enable frame-wrap test modes.
6	Reserved	Reserved. Set to 0.
5	FLOW	Flow control enable bit. When FLOW = 1, the TNETX3100 implements collision-based flow control. When FLOW = 0, no flow control is implemented.
4	TAGON	Post-frame tag enable bit. TAGON enables the requirement for post-frame tagging on the port 00 uplink. When TAGON = 1, the TNETX3100 uses the posttag to route frames received on the uplink. The other 100 Mbit/s (port 01) does not support post-frame tagging. When TAGON = 0, the TNETX3100 uses either the EAM interface or the internal address-lookup registers to route frames received on the uplink.
3–0	RSIZE	RAM size select. RSIZE indicates the size of the external DRAM and the number of 64-byte data buffers available. This field is used by the TNETX3100 to determine how many buffers to initialize. The code values are:

BIT 3 0	CODE VALUES	NUMBER OF BUFFERS
0100	4K × 36	240
0101	8K × 36	480
0110	16K × 36	960
0111	32K × 36	1920
1000	64K × 36	3840
1001	128K × 36	7680
1010	256K × 36	15360
1011	512K × 36	30720
1100	1M × 36	61440
1101	2M × 36	122880
1110	4M × 36	245760
1111	Reserved	—

system control register at 0x00D3

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	1

BIT	NAME	FUNCTION
7	RESET	RESET = 1 places the TNETX3100 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 register. Configuration registers are cleared by a hardware reset (write to DIO address 4000–7FFF).
6	LOAD	Load system. Writing LOAD = 1 causes the TNETX3100 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 LOAD = 0 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 START = 1 causes the TNETX3100 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 START = 0 has no effect.
4	CLRSTS	Clear statistics. Writing CLRSTS = 1 causes the TNETX3100 to clear all its statistics counters. The TNETX3100 repeats clearing the statistics counters until CLRSTS is cleared.
3	STMAP	Statistic mapping. STMAP selects which statistic is recorded in multiple-function statistic counters. When STMAP = 1, the number of TX frames discarded on TX, due to lack of resources, is recorded. If STMAP = 0, the number of data errors at TX is recorded.
2	SECDIS	Disable ports on security violations. When SECDIS = 1, address security violations cause a port to be disabled. When SECDIS = 0, address security violations cause a port to be suspended. Suspended ports are reenabled when the offending condition is removed. Disabled ports are only reenabled by management (by setting port ENABLE bit).
1	LONG	Long-frame handling. When LONG = 1, the RX and TX frame 1024–1518 bucket counter is redefined to become RX and TX frames 1024–1531. When LONG = 0, the RX and TX frame 1024–1518 bucket counter is operating normally. Frames 1536 bytes or larger are truncated.
0	IOBMOD	In-order broadcast mode. Always read as IOBMOD = 1; this is a read-only bit.

DRAM data register at 0x00D4–0x00D7

BIT	
31	0
DRAMDATA	

BIT	NAME	FUNCTION
31–0	DRAMDATA	DRAM data. DRAMDATA 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 activity. 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 is completed. After a DRAM test-access buffer is read, the user detects a falling edge on DRAMACT before proceeding to use the accessed data.
6–4	Reserved	Reserved
3–0	DRAMFLAG	DRAM flag. DRAMFLAG holds a value that maps to the flag field of a DRAM buffer when accessed in DRAM test access mode.

DRAM address 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, DRAMDATA, 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	RAS								CAS								

BIT	NAME	FUNCTION
22	Reserved	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 (do not set the start bit). Perform the following:
  DIO write (address 00D3) = 80
  DIO write (address 00D3) = 00
- Write to the port 00 TX FIFO block (72 bytes total):
  DIO write (address C200) = A5
  DIO write (address C201) = A5
  DIO write (address C202) = A5
  DIO write (address C203) = A5
  DIO write (address C204) = A5
  DIO write (address C205) = A5
  DIO write (address C206) = A5
  DIO write (address C207) = A5
  DIO write (address C208) = 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
- 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

DRAM test-access operation (continued)

- 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 C0D8)
DIO read [address C000 + (10 * n)] through [address C008 + (10 * n)]     where n = 0–7
```

DIATST register at 0x00DD

BIT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				DPWRAP	INTWRAP		Reserved
Initial Values After Reset							
0	X	X	0	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. When DPWRAP = 0, all ports function normally.
2–1	INTWRAP	Internal wrap mode. Ports 01–09 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 09 The port that is not wrapped (00, 02, or 09) should be used to inject and observe test data frames from the internally wrapped ports.
0	Reserved	Reserved. Should always be written as a 0.

PACTST register at 0x00DE

BIT							
7	6	5	4	3	2	1	0
FLAG100	FLAG10	Reserved	INITPACE				
Initial Values After Reset							
—	—	X	1	1	1	1	1

BIT	NAME	FUNCTION
7	FLAG100	Pacing flag comparison for all 100-Mbit/s. FLAG100 is the OR of all the 100-Mbit/s port compare signals resulting from the comparison between the pacing register and the initial pace value. When FLAG100 = 1, an error has occurred if all ports are involved in pacing and have experienced similar traffic. When an error is detected, no information is given as to which port signal is in error. When FLAG100 = 0, the TNETX3100 is functioning normally.
6	FLAG10	Pacing flag comparison for all 10-Mbit/s ports. Flag10 is the OR of all the 10-Mbits ports.
5	Reserved	Reserved
4–0	INITPACE	Pacing register initial value. At reset, bits 4–0 are inverted and loaded into the pacing register (the default value for the register is 00000 and the default loaded value after reset is 11111). Following reset, bits 4–0 are used to compare the contents of the pacing register. The result of the comparison is returned and ORed to form bits 6 and 7 in the PACTST register.

address registers at 0x08*A + 0x0100 through 0x08*A + 0x0107 (A = address register in hex, legal values are 1–15)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3 (MSB)	BIT 2	BIT 1	BIT 0 (LSB)	DIO ADDRESS
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	LOCK†	0x08(A–16)
Reserved	Reserved	Reserved	Reserved	Port number	Port number	Port number	Port number	0x08(A–16)+0x01
Address Bit 47	Address Bit 46	Address Bit 45	Address Bit 44	Address Bit 43	Address Bit 42	Address Bit 41	Address Bit 40	0x08(A–16)+0x02
Address Bit 39	Address Bit 38	Address Bit 37	Address Bit 36	Address Bit 35	Address Bit 34	Address Bit 33	Address Bit 32	0x08(A–16)+0x03
Address Bit 31	Address Bit 30	Address Bit 29	Address Bit 28	Address Bit 27	Address Bit 26	Address Bit 25	Address Bit 24	0x08(A–16)+0x04
Address Bit 23	Address Bit 22	Address Bit 21	Address Bit 20	Address Bit 19	Address Bit 18	Address Bit 17	Address Bit 16	0x08(A–16)+0x05
Address Bit 15	Address Bit 14	Address Bit 13	Address Bit 12	Address Bit 11	Address Bit 10	Address Bit 9	Address Bit 8	0x08(A–16)+0x06
Address Bit 7	Address Bit 6	Address Bit 5	Address Bit 4	Address Bit 3	Address Bit 2	Address Bit 1	Address Bit 0	0x08(A–16)+0x07

† LOCK is present for network address register 16; for all other network address registers (17–31), LOCK is reserved.

address register description

The 16 network registers shown in the following tables are combined with the other 16 registers (address 0000 through 007F) for a total of 32 network address registers; they form a 32-deep push-down stack, and this stack holds the last 32 source addresses.

port-N statistics 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
Multicast RX frames				Broadcast RX frames				0x80*N + 0x8008
RX align/code errors				RX CRC errors				0x80*N + 0x8010
RX jabbers				Oversize RX frames				0x80*N + 0x8018
RX fragments				Undersize RX frames				0x80*N + 0x8020
Frames 65–127				Frame 64				0x80*N + 0x8028
Frames 256–511				Frames 128–255				0x80*N + 0x8030
Frames 1024–1518				Frames 512–1023				0x80*N + 0x8038
SQE test errors				Net octets				0x80*N + 0x8040
Good TX frames				TX octets				0x80*N + 0x8048
Multicollision TX frames				Single-collision TX frames				0x80*N + 0x8050
Deferred TX frames				Carrier sense errors				0x80*N + 0x8058
Excessive collisions				Late collisions				0x80*N + 0x8060
Multicast TX frames				Broadcast TX frames				0x80*N + 0x8068
TX data errors†				Filtered RX frames				0x80*N + 0x8070
Address changes/mismatches				Address duplications				0x80*N + 0x8078

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

port-N RX overruns and collisions at 0x80*N + 0x8500 through 0x80*N + 0x8507 (N = port number in hex)

+7	+6	+5	+4	+3	+2	+1	+0	ADDRESS
RX overrun port 00				Collision port 00				0x80*N + 0x8500
RX overrun port 01				Collision port 01				0x80*N + 0x8508
RX overrun port 02				Collision port 02				0x80*N + 0x8510
RX overrun port 03				Collision port 03				0x80*N + 0x8518
RX overrun port 04				Collision port 04				0x80*N + 0x8520
RX overrun port 05				Collision port 05				0x80*N + 0x8528
RX overrun port 06				Collision port 06				0x80*N + 0x8530
RX overrun port 07				Collision port 07				0x80*N + 0x8538
RX overrun port 08				Collision port 08				0x80*N + 0x8540
RX overrun port 09				Collision port 09				0x80*N + 0x8548

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 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 counter statistics 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 rollover indication is not given.

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). The state of this bit is latched. When set, the next statistics update cycle writes 0 to all counters in the statistics RAM before resetting the latched bit. If the CLRSTS bit has not subsequently been reset by the user, the latched bit is set again, causing the TNETX3100 to load 0 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 0. Port statistics cannot be cleared on a per-port basis.

queue structures address map at 0x86E0 – 0x20*N + 0x86FF – 0x20*N (N = port number in hex)

+7	+6	+5	+4	+3	+2	+1	+0	ADDRESS
Transmit queue N head			Transmit queue N tail			Transmit queue N length		0x86E0–(0x20*N)
Immediate queue N head			Immediate queue N tail			Immediate queue N length		0x86E8–(0x20*N)
Receive queue N head			Receive queue N tail			Receive queue N length		0x86F0–(0x20*N)
Reserved			Reserved			Reserved		0x86F8–(0x20*N)

EEPROM interface

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

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

The TNETX3100 uses the 24C02 serial EEPROM device (2048 bits organized as 256 bits × 8). See Table 7. The 24C02 device uses a two-wire serial interface for communication and is available in a small-footprint package.

The organization of the EEPROM data is the same format as the TNETX3100 registers 0x00–0xD7 (see section on internal registers). This allows a complete initialization to be performed by downloading the contents of the EEPROM into the TNETX3100. 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.

The TNETX3100 detects the presence or 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 TNETX3100 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 Figure 23 and Figure 24.

If EEPROM is present, but has a bad CRC, the fault LED is illuminated.

EEPROM auto-configuration from an external x24C02 EEPROM

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

The organization of the EEPROM data is shown in Table 7. The last register loaded is the control register. This allows a complete initialization by downloading the contents of the EEPROM into the address-lookup table TNETX15VEPGE. 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 TNETX3100 detects the presence or absence of the EEPROM. If it is not installed, the EDIO terminal is tied low. For EEPROM operation, the terminal requires an external pullup (see EEPROM data sheet). When no EEPROM is detected, the TNETX3100 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 bit 7. Therefore, writing the address 0xC0 is accomplished by writing a 1, and then 1, 0, 0, 0, 0, 0, 0.

The TNETX3100 expects data to be stored in the EEPROM in a specific format. The range from 0x00 to 0x0D3 in the EEPROM is reserved for use by the adapter. The contents of the remaining bytes are undefined. The EEPROM is read/written by a software driver through the SIO register.

A 32-bit CRC value is calculated from the EEPROM data and placed in the EEPROM in the location following the bytes loaded into the internal register. The TNETX3100 uses this 32-bit CRC to validate the EEPROM data. If the CRC fails, TNETX3100 registers are set to their default (hardwired) values. The TNETX3100 is then placed in a post-reset halted state. The TNETX3100 is started through the DIO interface control register START bit.

The EEPROM algorithm, which is the same as the algorithm for the TNETX3100 EEPROM CRC, is used by IEEE Std 802.3 for the packet CRC calculation. EEPROM bytes are processed by the internal logic in the same order as the 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 7. EEPROM Address Map

EEPROM PHYSICAL ADDRESS	DESCRIPTION
00–07	Network address register 16
08–0F	Network address register 17
10–17	Network address register 18
18–1F	Network address register 19
20–27	Network address register 20
28–2F	Network address register 21
30–37	Network address register 22
38–3F	Network address register 23
40–47	Network address register 24
48–4F	Network address register 25
50–57	Network address register 26
58–5F	Network address register 27
60–67	Network address register 28
68–6F	Network address register 29
70–77	Network address register 30
78–7F	Network address register 31
80	Port 00 control
81	Port 00 status
82	Port 01 control
83	Port 01 status
84	Port 02 control
85	Port 02 status
86	Port 03 control
87	Port 03 status
88	Port 04 control
89	Port 04 status
8A	Port 05 control
8B	Port 05 status
8C	Port 06 control
8D	Port 06 status
8E	Port 07 control
8F	Port 07 status
90	Port 08 control
91	Port 08 status
92	Port 09 control
93	Port 09 status
94–9F	Reserved

EEPROM PHYSICAL ADDRESS	DESCRIPTION
A0	Revision register
A1	SIO/SCTRL register
A2	Reserved
A3	Device code register
A4–A5	Port 00 transmit queue-length register
A6–A7	Port 01 transmit queue-length register
A8–A9	Port 02 transmit queue-length register
AA–AB	Port 03 transmit queue-length register
AC–AD	Port 04 transmit queue-length register
AE–AF	Port 05 transmit queue-length register
B0–B1	Port 06 transmit queue-length register
B2–B3	Port 07 transmit queue-length register
B4–B5	Port 08 transmit queue-length register
B6–B7	Port 09 transmit queue-length register
B8–B9	Port 00 VLAN register
BA–BB	Port 01 VLAN register
BC–BD	Port 02 VLAN register
BE–BF	Port 03 VLAN register
C0–C1	Port 04 VLAN register
C2–C3	Port 05 VLAN register
C4–C5	Port 06 VLAN register
C6–C7	Port 07 VLAN register
C8–C9	Port 08 VLAN register
CA–CB	Port 09 VLAN register
CC–CE	Free queue-length register
CF	Reserved
D0	Feature-configuration register
D1	LED register
D2	RAM size register
D3	System-control register
D4	CRC byte 3
D5	CRC byte 2
D6	CRC byte 1
D7	CRC byte 0
D8–FF	Reserved

LED interface

An LED interface allows a visual status for each port to be displayed. Each port has a single LED; the LED display conveys information about the port state.

PORT STATE	LED DISPLAY
No link	Off
Link, but no activity	On
Occasional activity	On, with occasional off periods (1/16th second)
High activity	Fast flashing (8 Hz)
Partitioned	On for 0.5 seconds, then off for 1.5 seconds

In addition to the port status information, the LED interface also enables the state of the internal flow control and fault functions to be displayed along with four software-controllable LEDs. The display can be configured in a mode that displays only the partitioned status by setting the PONLY bit in the LED register. The $\overline{\text{LEDDATA}}$ signal is active low since TTL is more efficient at driving low than high (see Figure 10).

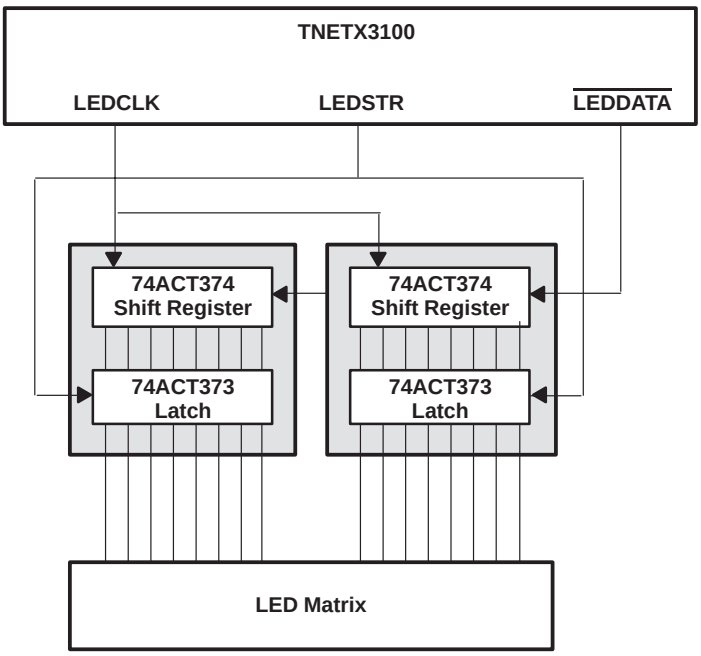


Figure 10. LED Interface

Sixteen status bits are shifted serially out and then LEDSTR is pulsed to latch the data; this process is repeated every 1/16th of a second. The shift register can be shortened if a system is not required to display all of the LEDs; a very low-cost system that implements only the FAULT LED can be readily implemented by using LEDSTR to clock a single external latch.

LED functions in shift-register order

NAME	FUNCTION
S1	Software LED 1. S1 is the first bit shifted out of the $\overline{\text{LEDDATA}}$ terminal and is controlled by the S1LED bit of the LED register. The function of this indicator is determined by the system designer.
S2	Software LED 2. S2 is controlled by the S2LED bit of the LED register. The function of this indicator is determined by the system designer.
S3	Software LED 3. S3 is controlled by the S3LED bit of the LED register. The function of this indicator is determined by the system designer.
S4	Software LED 4. S4 is controlled by the S4LED bit of the LED register. The function of this indicator is determined by the system designer.
FLOW	Flow control. FLOW is on when the internal flow control is enabled and active.
PORT00	Port 00 status. PORT00 indicates the status of port 00.
PORT01	Port 01 status. PORT01 indicates the status of port 01.
PORT02	Port 02 status. PORT02 indicates the status of port 02.
PORT03	Port 03 status. PORT03 indicates the status of port 03.
PORT04	Port 04 status. PORT04 indicates the status of port 04.
PORT05	Port 05 status. PORT05 indicates the status of port 05.
PORT06	Port 06 status. PORT06 indicates the status of port 06.
PORT07	Port 07 status. PORT07 indicates the status of port 07.
PORT08	Port 08 status. PORT08 indicates the status of port 08.
PORT09	Port 09 status. PORT09 indicates the status of port 09.
FAULT	Fault. FAULT indicates that the CRC of the EEPROM was invalid; this is the last bit shifted out of the $\overline{\text{LEDDATA}}$ terminal.

lamp test

When TNETX3100 is in the reset state, $\overline{\text{LEDDATA}}$ line is driven low; this causes all LEDs to illuminate and serves as a lamp-test function. In this mode, LEDCLK and LEDSTR run at a higher frequency to shorten the time it takes to illuminate all of the LEDs.

JTAG interface

The TNETX3100 is fully JTAG compliant, with the exception of requiring external pullup resistors on the TDI, TMS, and $\overline{\text{TRST}}$ terminals.

external pullup resistors

To implement internal pullup resistors, TNETX3100 requires 5-V-nontolerant input pads. The use of 5-V tolerant pads is considered more important for mixed-voltage system boards than to integrate the required pullup resistors and 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 (6-bit instruction register) are listed in Table 8.

supported JTAG instructions (continued)

Table 8. Opcodes for JTAG Instructions

INSTRUCTION TYPE	INSTRUCTION NAME	JTAG OPCODE
Mandatory	EXTEST	000000
Mandatory	SAMPLE/PRELOAD	000001
Optional	IDCODE	000100
Optional	HIGHZ	000101
Mandatory	BYPASS	111111

The IDCODE for the TNETX3100 is given in Table 9.

Table 9. IDCODE Code

VARIANT	PART NUMBER	MANUFACTURE	LEAST-SIGNIFICANT BIT
Bit 31 • • • Bit 28	Bit 27 • • • Bit 12	Bit 11 • • • Bit 1	Bit 0
0000	0000000001000110	00000010111	1

manufacturing test features

The methodology of these tests is considered before implementation. All tests are based on an incremental approach; for tests using the DIO interface (for example), only the DIO interface should be tested and if it is functioning correctly, the next depth of testing is performed (i.e., internal TNETX3100 testing). If a test fails using this methodology, the cause of the failure can be determined quickly and test/debug time is 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 test the TNETX3100 internally.

secondary test access: internal RAM access modes

When implementing an architecture that employs embedded RAM structures, it is necessary to ensure test access greater than JTAG-connectivity testing through standard interfacing. The DIO interface used by TNETX3100 enables the user to interrogate the internal RAMS of the TNETX3100. User interrogation gives the required observability for the RAMs 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 allows 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. Access to the FIFO is allowed 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 TNETX3100 holds the DRAM refresh logic operation in reset and the contents of the external memory become 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 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 TNETX3100 is soft reset (or following power up before the start bit is set). In this mode, all locations of the RAM are 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, resulting in the DRAM contents becoming 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 only requires a new structure RAM address for set up 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

The frame-wrap test mode allows the user to send a frame into a designated source port and selectively route the frame 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 TNETX3100 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) or port 09 can be selected as the source port for injecting frames into the TNETX3100 when internal wrap is selected. All other ports are set to internally wrap frames (see Figure 11).

internal frame-wrap test mode (continued)

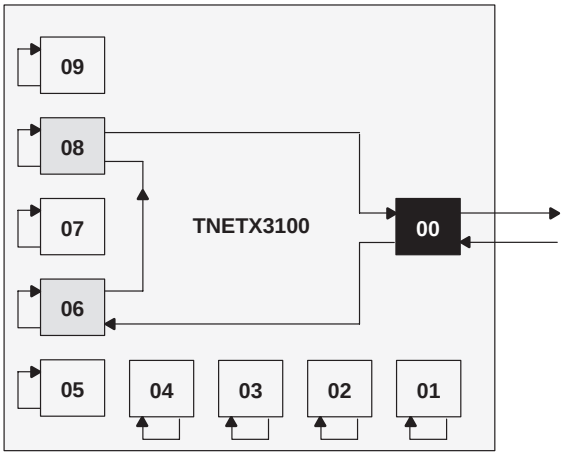


Figure 11. Internal Frame-Wrap Test (Wrapped Between Multiple Ports Using Broadcast Frames and VLAN Registers)

By injecting broadcast or multicast frames into the source port and 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 TNETX3100) 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 TNETX3100 and the PHY to be verified. 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 12).

External frame-wrap test mode is selected by setting the DPWRAP bit (bit 3) of the DIATST register. When selected, the port is forced into full duplex, allowing it 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.

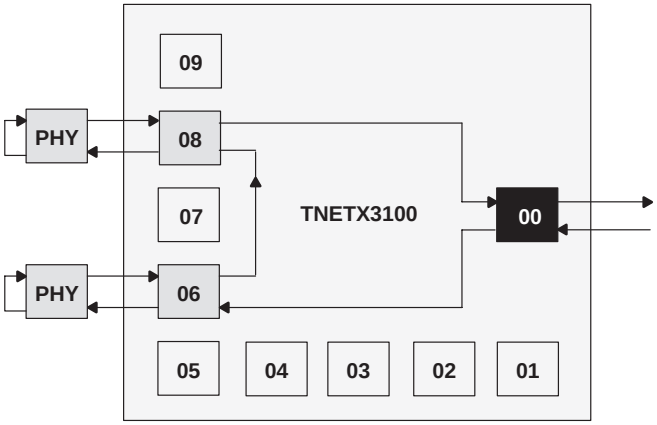


Figure 12. External Frame-Wrap Test (Wrapped Between Multiple Ports Using Broadcast Frames and VLAN Registers)

internal and external frame-wrap example

Perform the following steps to configure the TNETX3100 for the internal frame-wrap configuration:

Step 1. Disable any enternal address-matching hardware. (No traffic should be received by the TNETX3100 at this time.)

Step 2. With power on, place TNETX3100 in a reset state by writing 0x81 at address 0x00D3.

Step 3. While the TNETX3100 is in reset, configure the VLAN mask registers (0x00B8–0x00CB). 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	=	0x0001

Step 4. Start the TNETX3100 by writing a 1 to the start bit and make sure the IOBMOD bit also is a 1 (0x21 at address 0x00D3).

Step 5. Set all ports to full duplex and enable all ports.

Step 6. Set the broadcast unknown (BRUN) bit to zero (address 0x00A1).

Step 7. Configure the TNETX3100 for test mode by writing a 1 to the MTEST bit (address 0x00D2).

Step 8. Configure the TNETX3100 for internal wrap mode by writing a 1 to the INTWRAP. See the data sheet for INTWRAP details. The port that is not wrapped (00, 01, or 09) is used to inject and observe test data frames from internally wrapped ports. For port 00, write 0x02 at address 0x00DD. For port 01, write 0x40 at address 0x00DD. For port 09, write 0x06 at address 0x00DD.

Step 9. Transmit a broadcast/multicast frame into the TNETX3100 port corresponding to the INTWRAP setting. The frame that is sent into the test port is received by the same port (provided the VLAN register settings steered the packet back to that port). Compare the frame transmitted against the frame received and verify that the TNETX3100 is working correctly.

For external wrap mode, replace steps 8 and 9 with the following:

Step 8. Configure the TNETX3100 for external wrap mode by writing 0x08 at address 0x00DD.

Step 9. Configure all PHY – layer devices except port 00 for internal loopback.

Step 10. Transmit a broadcast/multicast frame into the TNETX3100 on port 00. The frame that is sent into the test port is received by the same port (provided the VLAN register settings steer the packet back to that port). Compare the frame transmitted against the frame received and verify that the TNETX3100 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 TNETX3100 data paths (to and from the RAMS) during normal frame operations or an indication of the control-path functionality. To assist with this further, 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 loopback mode. This builds on the previous tests, and ensures 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 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 TNETX3100 should be reset before normal switching activity is resumed. This ensures that the TNETX3100 is returned to a defined state before normal functionality is resumed. This mechanism is intended primarily for DRAM testing and is not a part of a breakpoint/debug mechanism.

For more information refer, to the *DRAM test-access operation register* section in the TNETX3100 register descriptions.

glossary

address duplications

The number of address duplications between a securely assigned port address within the TNETX3100 and a source address observed on this port. Occurrence of this causes the TNETX3100 to suspend the port (see *port-N status register* section for 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 TNETX3100 to suspend the port (see *port-N status register* section for description).
2. The number of times the TNETX3100 is required to assign or learn an address for a port

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 that the carrier sense condition was lost or never asserted when attempting to transmit a frame on a particular interface. The count is incremented, at most, once for every transmission attempt, even if the carrier sense condition fluctuates during a transmission attempt.

collisions port (02–09)

The number of times that 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 is 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 TNETX3100 discard channel for any reason.

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 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 the LONG bit in the system-control register is set, oversize RX frames are redefined as being longer than 1535 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 1535 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 nonintegral 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 frame-check sequence (FCS) with an integral number of octets (FCS error) or a bad FCS with a nonintegral number of octets (alignment error). If the LONG bit in the system control register is set, RX jabber is redefined as being longer than 1535 octets.

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 (00–09)

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 any 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 sublayer for a particular interface. The SQE test error is defined in section 7.2.2.2.4 of ANS/IEEE Std 802.3-1985 and its generation is described in section 7.2.4.6 of the same document.

glossary (continued)

TX data errors

This statistic is switchable between:

- The number of transmit frames discarded on transmission due to lack of resources (i.e., the transmit queue was full). This allows queue monitoring for dynamic queue sizing and buffer allocation.
- 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.

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, $Z_{\theta JA}$	31°C/W
Thermal impedance, junction-to-case package, $Z_{\theta JC}$	8°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. 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 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

		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	3	3.3	3.6	V
$V_{CC(5V)}$	Supply voltage	4.5	5	5.5	V
V_{IH}	High-level input voltage	2			V
V_{IL}	Low-level input voltage (see Note 3)			0.8	V
I_{OH}	High-level output current			–2	mA
I_{OL}	Low-level output current			–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		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	V_{CC} = MIN, I_{OH} = –2mA	2.4			V
V_{OL}	Low-level output voltage	V_{CC} = MIN, I_{OL} = 2mA			0.4	V
I_{OZ}	High-impedance-state output current	$V_O = V_{CC}$			20	μA
		V_O = 0			–20	
I_{IH}	High-level input current	$V_I = V_{I(MAX)}$			–20	μA
I_{IL}	Low-level input current	V_I = GNDsp			20	μA
I_{CC}	Supply current	V_{CC} = MAX, f = 50 MHz			390	mA
$I_{CC(5V)}$	Supply current	$V_{CC(5V)}$ = MAX			1	mA

PARAMETER MEASUREMENT INFORMATION

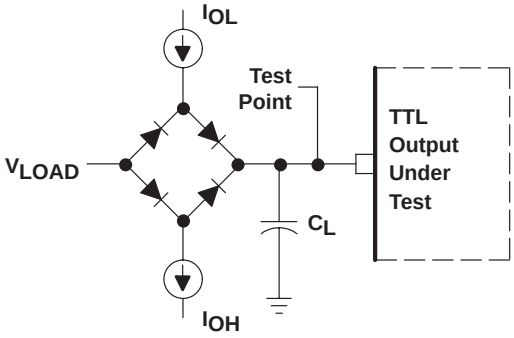
Outputs are driven to a minimum high-logic level of 3.3 V and to a maximum low-logic level of 0 V.

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 1.4 V and the level at which the signal is said to be low is 1.4 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-and-load circuit shown in Figure 13 represents the programmable load of the tester-pin electronics that is used to verify timing parameters of the TNETX3100 output signals.



TTL OUTPUT TEST LOAD

Where:	I_{OL}	=	**Refer to I_{OL} in recommended operating conditions.**
	I_{OH}	=	**Refer to I_{OH} in recommended operating conditions.**
	V_{LOAD}	=	**1.5 V, typical dc-level verification or 1.5 V, typical timing verification**
	C_L	=	**45 pF, typical load-circuit capacitance**

Figure 13. Test-and-Load Circuit

timing requirements (see Note 4 and Figure 14)
DRAM read 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 4: All DRAM output signals are synchronous to the DREF clock. Figure 14 shows a single DRAM read (TNETX3100 forward pointer update).

operating characteristics over recommended operating conditions (see Note 4 and Figure 14)
DRAM read cycle (with DREF clock)

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

† All DRAM terminals are specified to have less than a 7-ns variance at any temperature and voltage.

NOTE 4: All DRAM output signals are synchronous to the DREF clock. Figure 14 shows a single DRAM read (TNETX3100 forward pointer update).

operating characteristics over recommended operating conditions (see Note 5 and Figure 14)
DRAM read cycle (with OSCIN clock)

NO.	PARAMETER		MIN	MAX	UNIT
12	$t_{d(DA)}$	Delay time, from OSCIN↑ to DA7–DA0 valid/invalid	3	18	ns
12	$t_{d(DX)}$	Delay time, from OSCIN↑ to DX2–DX0 valid/invalid	3	18	ns
13	$t_{d(DRAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DRAS}}$ transition	3	16	ns
14	$t_{d(DWE)}$	Delay time, from OSCIN↑ to $\overline{\text{DWE}}$ transition	3	16	ns
15	$t_{d(DCAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DCAS}}$ transition	3	15	ns
16	$t_{d(DOE)}$	Delay time, from OSCIN↑ to $\overline{\text{DOE}}$ transition	3	18	ns

NOTE 5: All DRAM output signals are synchronous to the OSCIN clock. Figure 14 shows a single DRAM read (TNETX3100 forward pointer update).

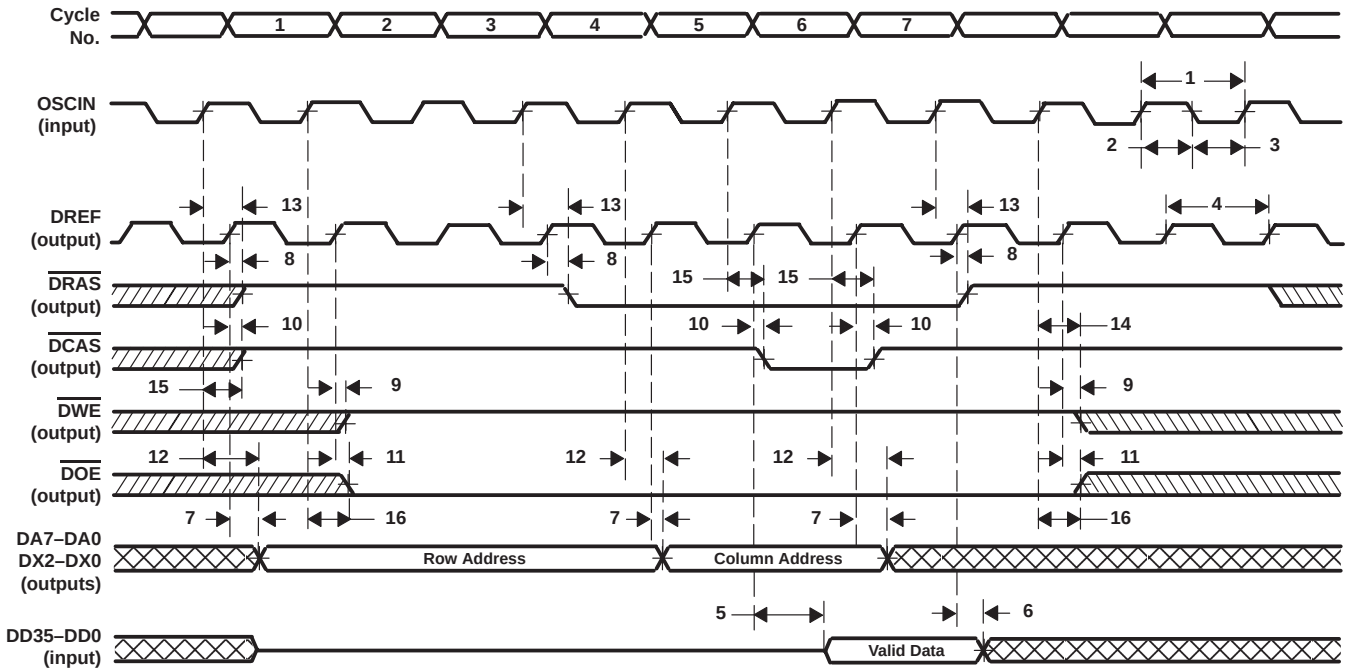


Figure 14. DRAM Read Cycle

timing requirements (see Note 6 and Figure 15)
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 (TNETX3100 forward pointer update).

operating characteristics over recommended operating conditions (see Note 6 and Figure 15)
DRAM write cycle (with DREF clock)

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

† All DRAM terminals are specified to have less than a 7-ns variance at any temperature and voltage.

NOTE 6: All DRAM output signals are synchronous to the DREF clock. Figure 15 shows a single DRAM write (TNETX3100 forward pointer update).

operating characteristics over recommended operating conditions (see Note 7 and Figure 15)
DRAM write cycle (with OSCIN clock)

NO.	PARAMETER		MIN	MAX	UNIT
11	$t_{d(DA)}$	Delay time, from OSCIN↑ to DA7–DA0 valid/invalid	3	18	ns
11	$t_{d(DX)}$	Delay time, from OSCIN↑ to DX2–DX0 valid/invalid	3	18	ns
12	$t_{d(DRAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DRAS}}$ transition	3	16	ns
13	$t_{d(DD)}$	Delay time, from OSCIN↑ to DD35–DD0 valid/invalid	3	18	ns
14	$t_{d(DOE)}$	Delay time, from OSCIN↑ to $\overline{\text{DOE}}$ transition	3	18	ns
15	$t_{d(DWE)}$	Delay time, from OSCIN↑ to $\overline{\text{DWE}}$ transition	3	16	ns
16	$t_{d(DCAS)}$	Delay time, from OSCIN↑ to $\overline{\text{DCAS}}$ transition	3	15	ns

NOTE 7: All DRAM output signals are synchronous to the OSCIN clock. Figure 15 shows a single DRAM write (TNETX3100 forward pointer update).

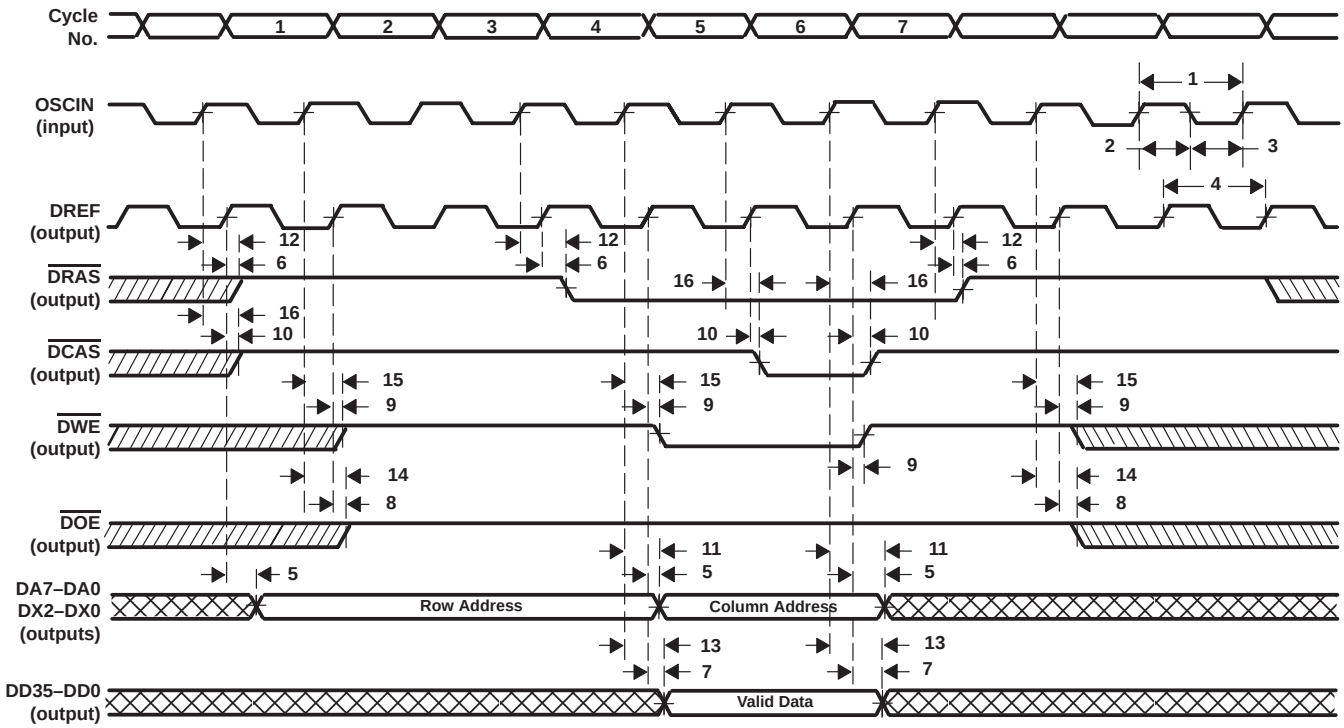


Figure 15. DRAM Write Cycle

timing requirements (see Notes 8 and 9 and Figure 16)
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 TNETX3100 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 16)
DRAM CAS-before-RAS (CBR) refresh cycle (with DREF clock)

NO.	PARAMETER		MIN	TYP	MAX	UNIT
4	$t_{c(DREF)}$†	Cycle time, DREF clock		20		ns
5	$t_{d(DCAS)}$†	Delay time, from DREF↑ to $\overline{DCAS}$ transition		0		ns
6	$t_{d(DRAS)}$†	Delay time, from DREF↑ to $\overline{DRAS}$ transition		0		ns
7	$t_{d(DOE)}$†	Delay time, from DREF↑ to $\overline{DOE}$ transition		0		ns
8	$t_{d(DWE)}$†	Delay time, from DREF↑ to $\overline{DWE}$ transition		0		ns

† All DRAM terminals are specified to have less than a 7-ns variance at any temperature and voltage.

NOTES: 8. All DRAM output signals are synchronous to the DREF clock.
9. The TNETX3100 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 16)
DRAM CAS-before-RAS (CBR) refresh cycle (with OSCIN clock)

NO.	PARAMETER		MIN	MAX	UNIT
9	$t_{d(DCAS)}$	Delay time, from OSCIN↑ to $\overline{DCAS}$ transition	3	15	ns
10	$t_{d(DRAS)}$	Delay time, from OSCIN↑ to $\overline{DRAS}$ transition	3	16	ns
11	$t_{d(DOE)}$	Delay time, from OSCIN↑ to $\overline{DOE}$ transition	3	18	ns
12	$t_{d(DWE)}$	Delay time, from OSCIN↑ to $\overline{DWE}$ transition	3	16	ns

NOTES: 9. The TNETX3100 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 (TNETX3100 forward pointer update).

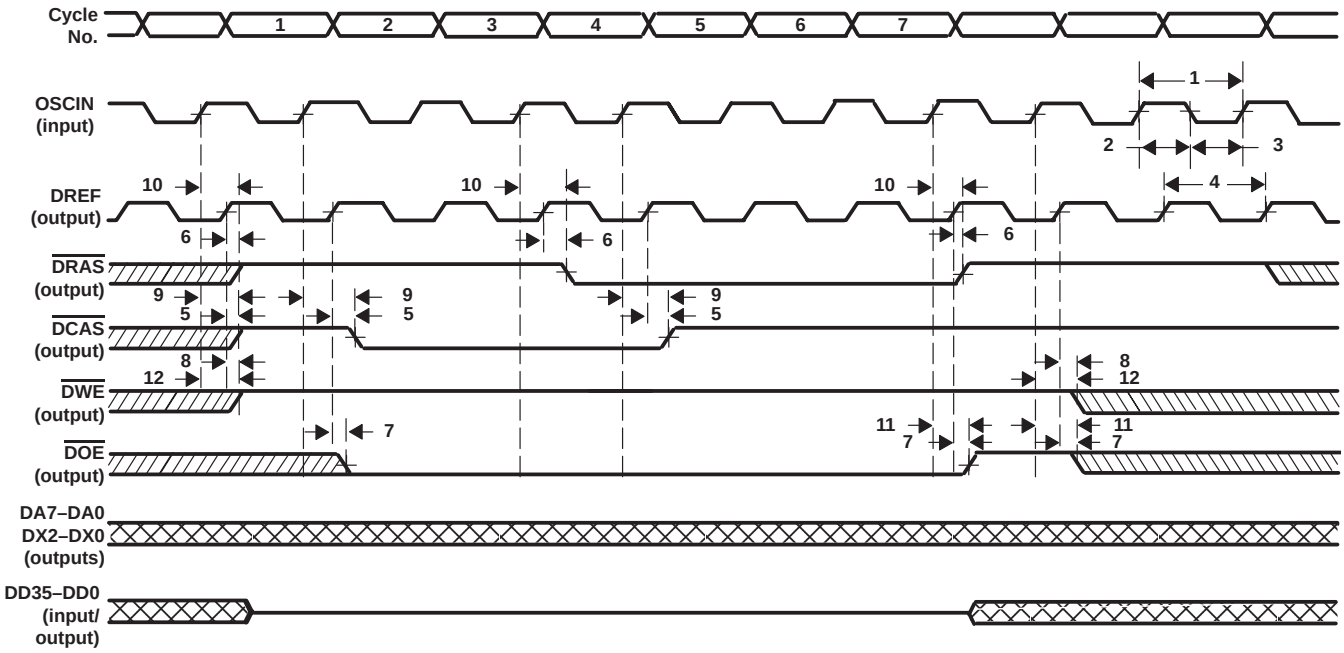


Figure 16. CBR Refresh Cycle

timing requirements (see Figure 17)
DRAM burst write cycle (TNETX3100 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 17)
DRAM burst write cycle (TNETX3100 frame buffer write) (with DREF clock)

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

† All DRAM terminals are specified to have less than a 7-ns variance at any temperature and voltage.

operating characteristics over recommended operating conditions (see Figure 17)
DRAM burst write cycle (TNETX3100 frame buffer write) (with OSCIN clock)

NO.	PARAMETER		MIN	MAX	UNIT
11	$t_{d(DA)}$	Delay time, from OSCIN↑ to DA7–DA0 valid/invalid	3	18	ns
11	$t_{d(DX)}$	Delay time, from OSCIN↑ to DX2–DX0 valid/invalid	3	18	ns
12	$t_{d(DRAS)}$	Delay time, from OSCIN↑ to $\overline{DRAS}$ transition	3	16	ns
13	$t_{d(DD)}$	Delay time, from OSCIN↑ to DD35–DD0 valid/invalid	3	18	ns
14	$t_{d(DOE)}$	Delay time, from OSCIN↑ to $\overline{DOE}$ transition	3	18	ns
15	$t_{d(DWE)}$	Delay time, from OSCIN↑ to $\overline{DWE}$ transition	3	16	ns
16	$t_{d(DCAS)}$	Delay time, from OSCIN↑ to $\overline{DCAS}$ transition	3	15	ns

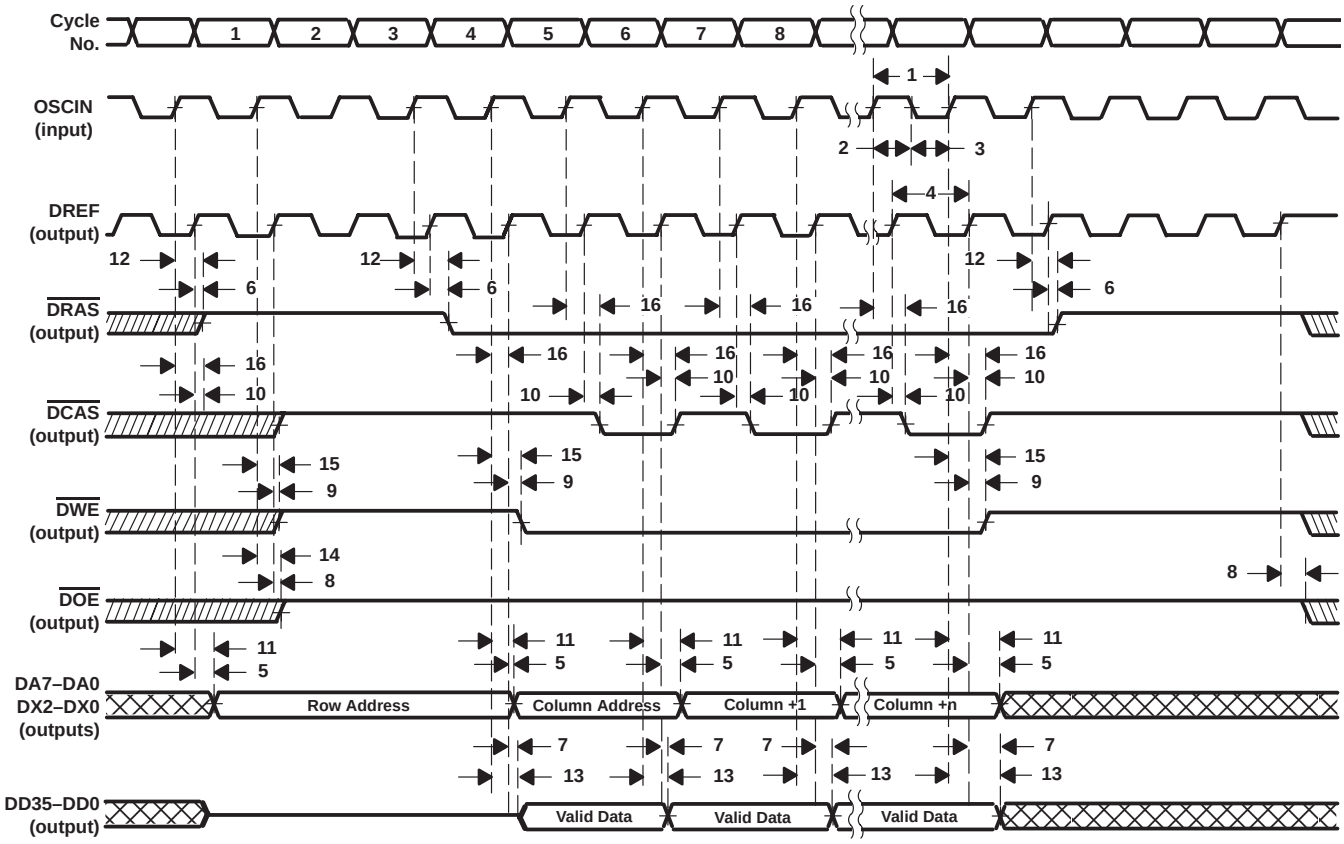


Figure 17. DRAM Burst Write Cycle

timing requirements (see Figure 18)
DRAM burst read cycle (TNETX3100 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 18)
DRAM burst read cycle (TNETX3100 frame buffer read) (with DREF clock)

NO.	PARAMETER		TNETX3150A			UNIT
			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			25	ns
6	$t_{d(DD)2}$	Delay time, from DREF↑ to DD35–DD0 invalid	0			ns
7	$t_{d(DA)}$†	Delay time, from DREF↑ to DA7–DA0 valid/invalid		0		ns
7	$t_{d(DX)}$†	Delay time, from DREF↑ to DX2–DX0 valid/invalid		0		ns
8	$t_{d(DRAS)}$†	Delay time, from DREF↑ to $\overline{DRAS}$ transition		0		ns
9	$t_{d(DWE)}$†	Delay time, from DREF↑ to $\overline{DWE}$ transition		0		ns
10	$t_{d(DCAS)}$†	Delay time, from DREF↑ to $\overline{DCAS}$ transition		0		ns
11	$t_{d(DOE)}$†	Delay time, from DREF↑ to $\overline{DOE}$ transition		0		ns

† All DRAM terminals are specified to have less than a 7-ns variance at any temperature and voltage.

operating characteristics over recommended operating conditions (see Figure 18)
DRAM burst read cycle (TNETX3100 frame buffer read) (with OSCIN clock)

NO.	PARAMETER		TNETX3150A		UNIT
			MIN	MAX	
12	$t_{d(DA)}$	Delay time, from OSCIN↑ to DA7–DA0 valid/invalid	3	18	ns
12	$t_{d(DX)}$	Delay time, from OSCIN↑ to DX2–DX0 valid/invalid	3	18	ns
13	$t_{d(DRAS)}$	Delay time, from OSCIN↑ to $\overline{DRAS}$ transition	3	16	ns
14	$t_{d(DWE)}$	Delay time, from OSCIN↑ to $\overline{DWE}$ transition	3	16	ns
15	$t_{d(DCAS)}$	Delay time, from OSCIN↑ to $\overline{DCAS}$ transition	3	15	ns
16	$t_{d(DOE)}$	Delay time, from OSCIN↑ to $\overline{DOE}$ transition	3	18	ns

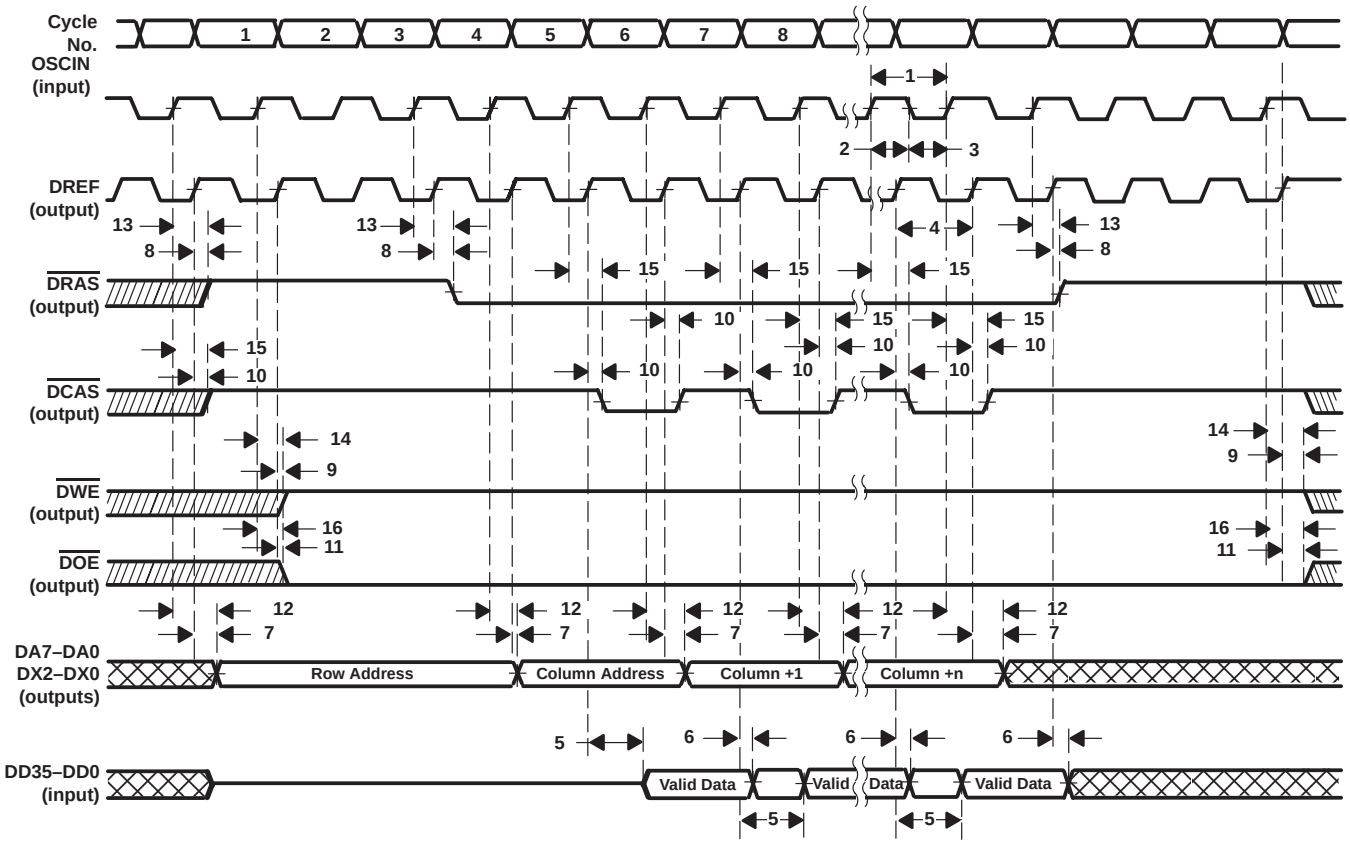


Figure 18. DRAM Burst Read Cycle

DIO write cycle

- TNETX3100 host-register address data SAD1–SAD0 and SDATA7–SDATA0 are asserted and SRNW is taken low.
- After the setup time, $\overline{SCS}$ is taken low, initiating a write cycle.
- The TNET3100 pulls $\overline{SRDY}$ low as the data is accepted and SDATA7–SDATA0, SAD1–SAD0, and SRNW are deasserted after the hold time is satisfied.
- $\overline{SCS}$ is taken high by the host to complete the cycle, causing $\overline{SRDY}$ to be deasserted, and $\overline{SRDY}$ is driven high for one cycle before going to the high-impedance (Z) state.

timing requirements (see Note 11 and Figure 19) write cycle DIO interface

NO.			MIN	MAX	UNIT
1	$t_{w(SCS)}$	Pulse duration, $\overline{SCS}$ low	40		ns
2	$t_{su(SAD)}$	Setup time, SAD1–SAD0 valid before $\overline{SCS}\downarrow$	0		ns
3	$t_{su(SDATA)}$	Setup time, SDATA7–SDATA0 valid before $\overline{SCS}\downarrow$	0		ns
4	$t_{su(SRNW)}$	Setup time, SRNW low before $\overline{SCS}\downarrow$	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 19) write cycle DIO interface

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

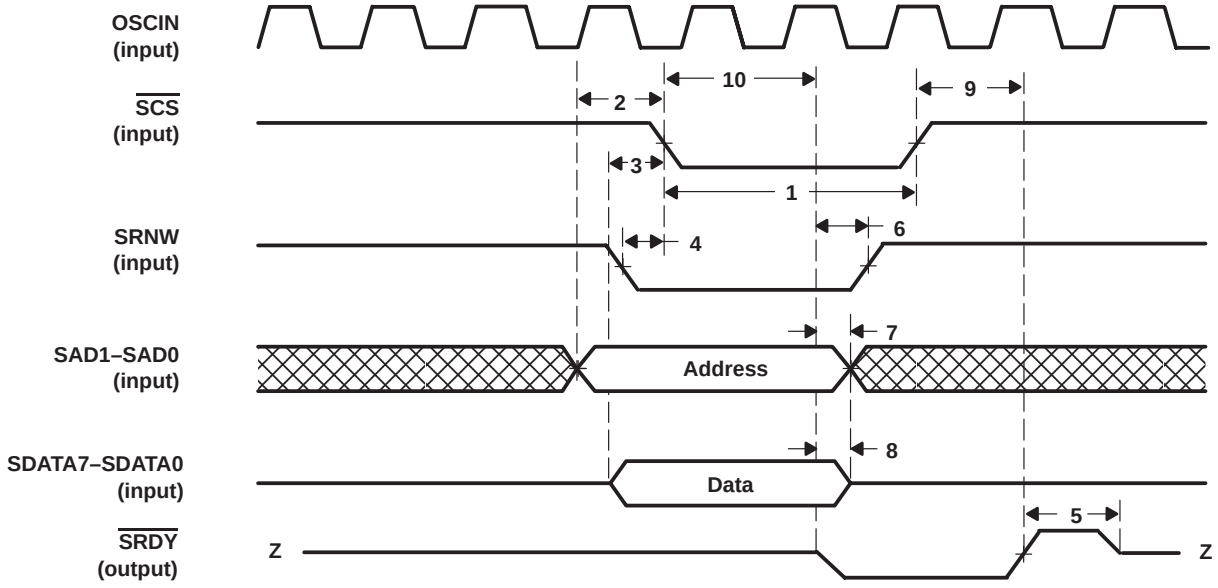


Figure 19. DIO Write Cycle

DIO read cycle

- TNETX3100 host-register address data is placed on address terminals SAD1–SAD0 and SRNW is held high.
- After setup time, $\overline{SCS}$ is taken low, initiating the read cycle.
- After delay following $\overline{SCS}$ low, SDATA7–SDATA0 is released from the Z state.
- After delay following $\overline{SCS}$ low, SDATA7–SDATA0 is driven with valid data and $\overline{SRDY}$ is pulled low. The host can access the data.
- $\overline{SCS}$ is taken high by the host upon completion of the cycle, causing $\overline{SRDY}$ to deassert, and $\overline{SRDY}$ is driven high for one clock cycle before going to the Z state. SDATA7–SDATA0 also goes to the Z state.

timing requirements (see Note 12 and Figure 20) read cycle DIO interface

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

NOTE 12: $\overline{SRDY}$ 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 20) read cycle DIO interface

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

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

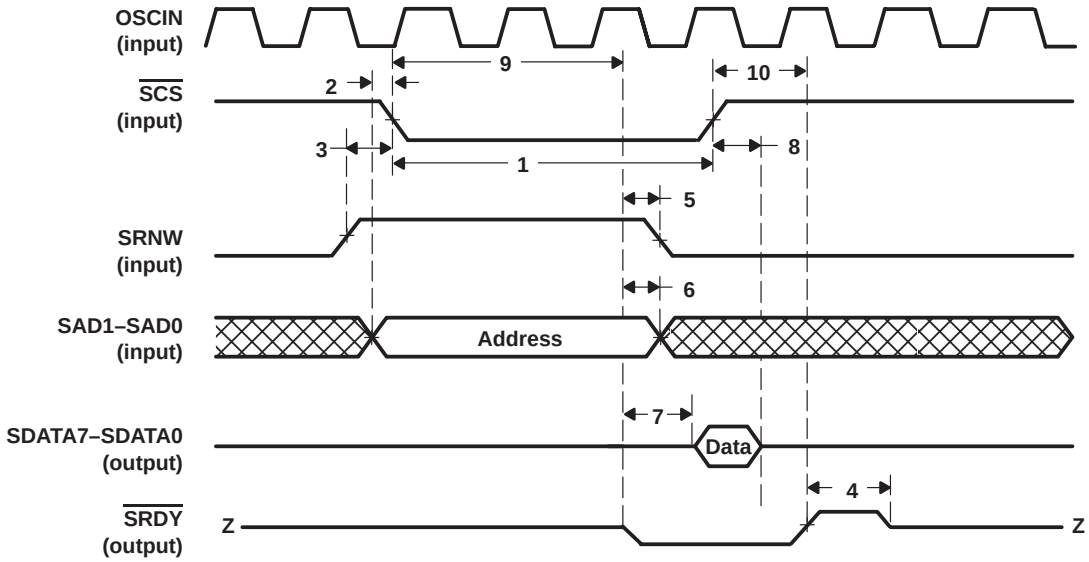


Figure 20. DIO Read Cycle

timing requirements (see Note 13 and Figure 21)
EAM interface

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

NOTE 13: To determine the start of frame, the external address hardware tests bit 35 of the forward pointer and decodes the first flag nibble placed on the DRAM data bus. Bit 35 should be 0, indicating a valid data-frame start as opposed to a link buffer transfer. By using the $\overline{\text{DCAS}}$ signal, the destination address and source address of the frame is extracted for external processing.

The channel destination is returned in one of two methods. If only one port address is specified (effectively a unicast), the EAM15 (MODE SELECT) signal is asserted, and a 5-bit port code is placed on EAM04–EAM00. If a group multicast is required, the channel bit map is applied directly to the EAM interface with EAM15 (MODE SELECT) low. The EAM15–EAM0 signals must be valid by the start of the 15th memory access.

All signals in the external address checking interface are synchronous with the DREF clock.

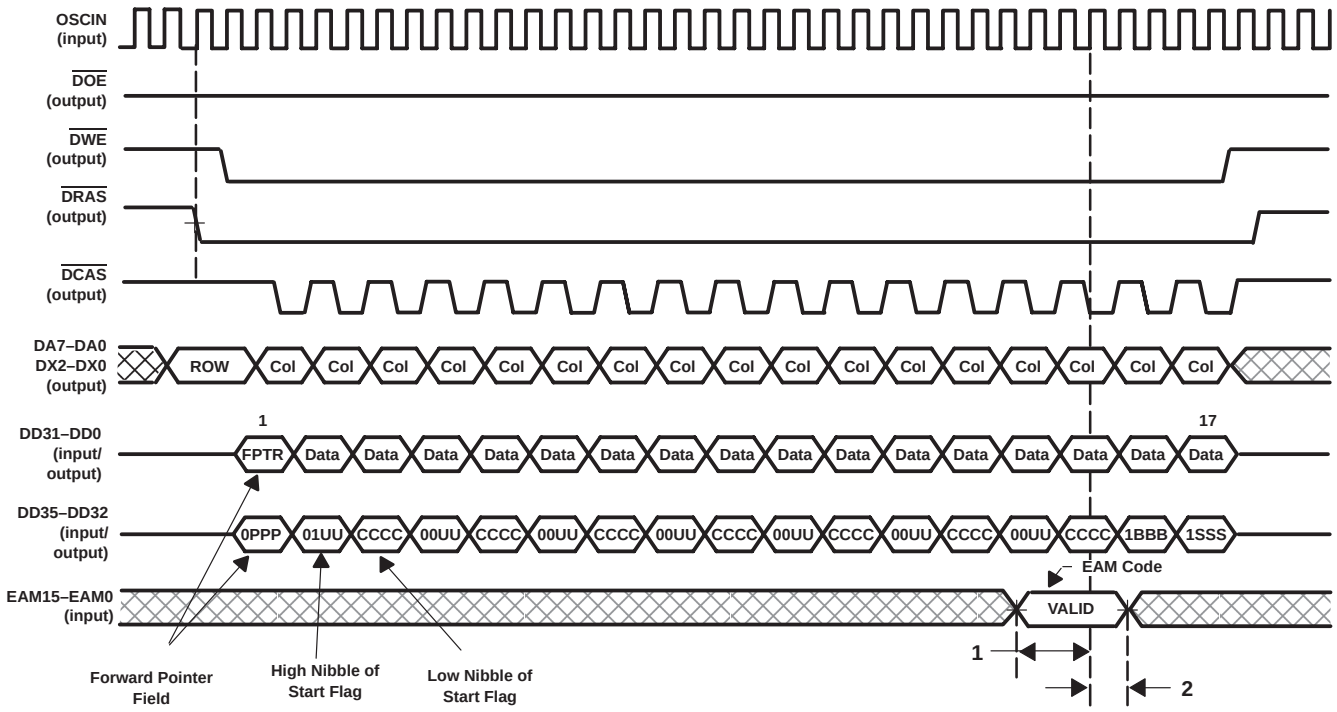


Figure 21. EAM Interface Timing

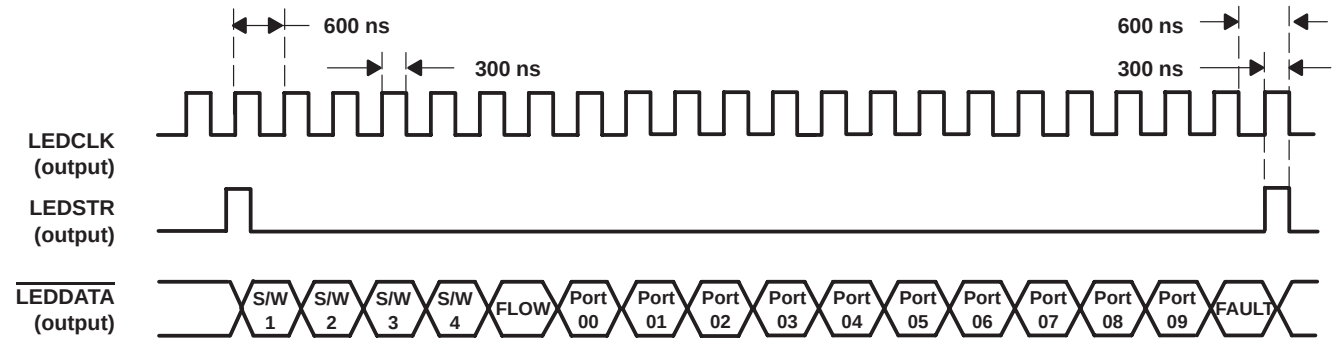


Figure 22. LED Interface Timing

operating characteristics over recommended operating conditions (see Figure 23)
EEPROM interface

NO.	PARAMETER		MIN	MAX	UNIT
	f_{clock} (ECLK)	Clock frequency, ECLK		98	kHz
1	t_d (ECLKH–EDIOL)	Delay time, from ECLK↑ to EDIO↓ (see Note 14)	5		µs
2	t_d (EDIOL–ECLKL)	Delay time, from EDIO↓ to ECLK↓ (see Note 14)	5		µs
3	t_d (ECLKL–EDIOX)	Delay time, from ECLK↓ to EDIO changing (see Note 15)	0		µs
4	t_d (EDIOV–ECLKH)	Delay time, from EDIO valid output to ECLK↑	0		µs
5	t_d (ECLKL–EDIOV)	Delay time, from ECLK↓ to EDIO valid	0		µs
6	t_d (ECLKL–EDIOX)	Delay time, from ECLK↓ to EDIO changing (see Note 16)	0		µs
7	t_d (ECLKH–EDIOX)	Delay time, from ECLK↑ to EDIO invalid	5		µs
8	t_d (EDIOV–ECLKH)	Delay time, from EDIO valid input to ECLK↑	10		µs

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

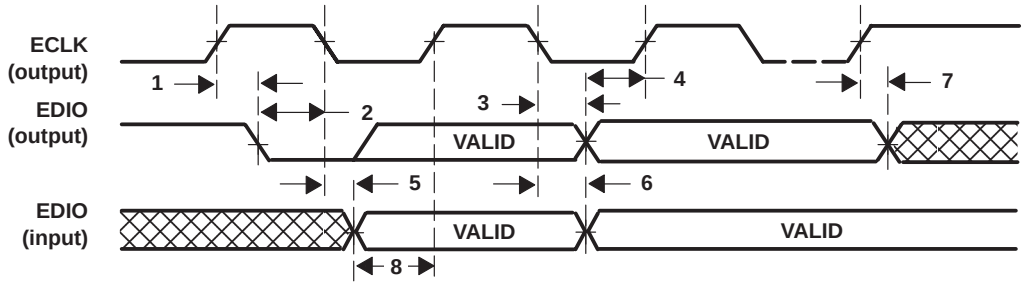


Figure 23. EEPROM Interface Timing

timing requirements (see Figure 24)
MII receive 100 Mbit/s (xx=ports 00–01)

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

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 = 00–01).

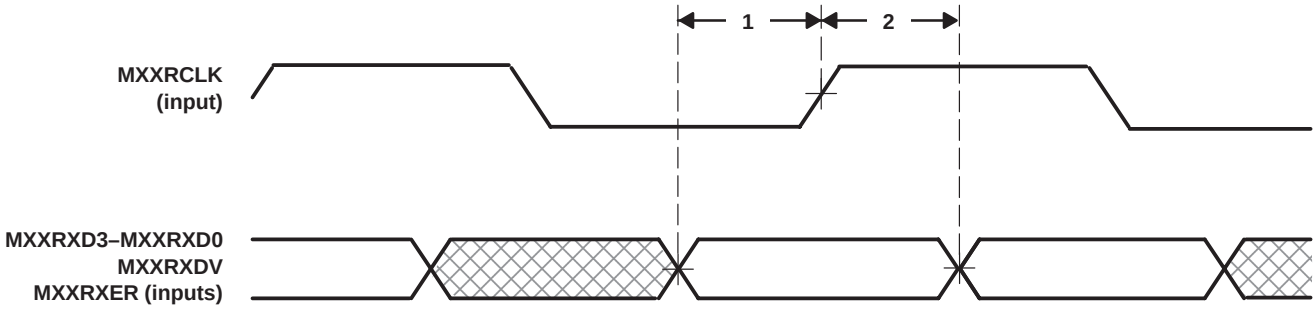


Figure 24. Ports (00–01) Receive Interface Timing

operating characteristics (see Figure 25)
MII transmit 100 Mbit/s (xx=ports 00–01)

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

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 = 00–01).

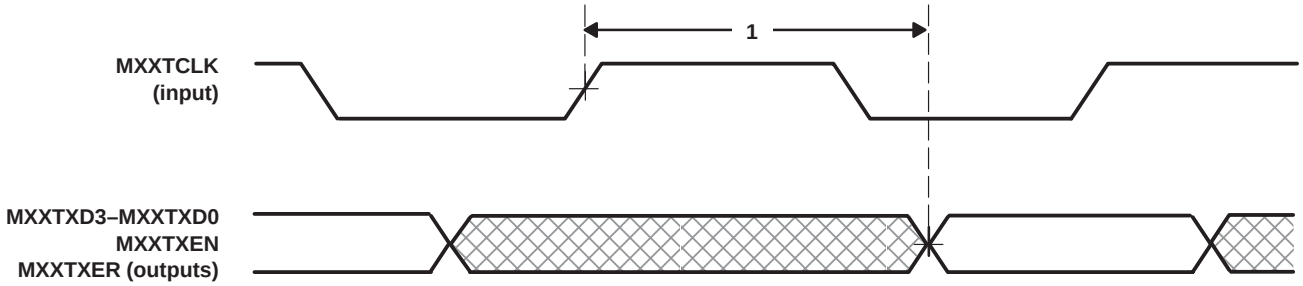


Figure 25. Ports (00–01) Transmit Interface Timing

timing requirements (see Figure 26)
MII receive 10 Mbit/s (xx = ports 02–09)

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

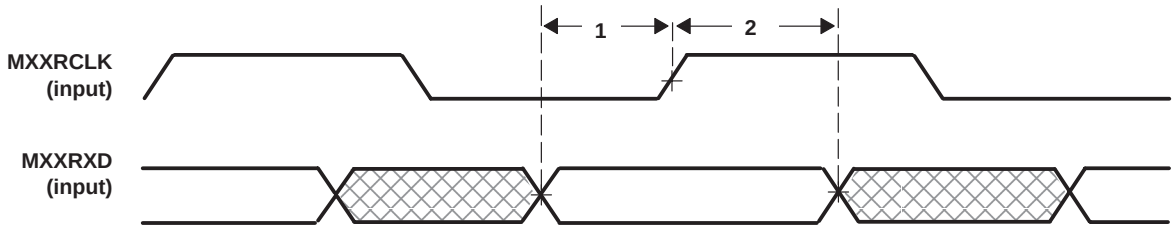


Figure 26. Ports (02–09) Receive Interface Timing

operating characteristics over recommended operating conditions (see Figure 27)
MII transmit 10 Mbit/s (xx = ports 02–09)

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

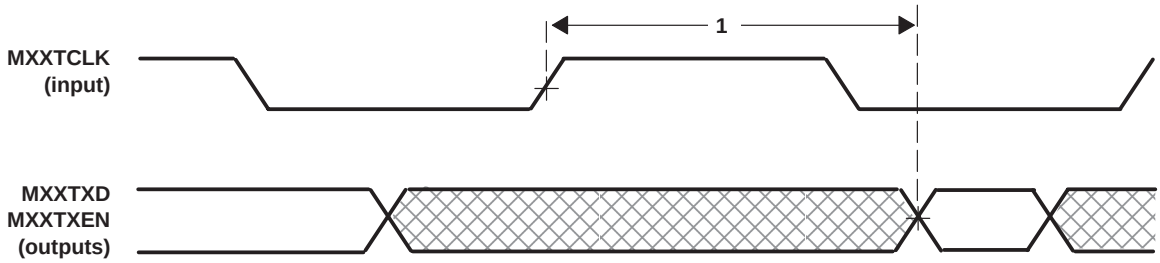
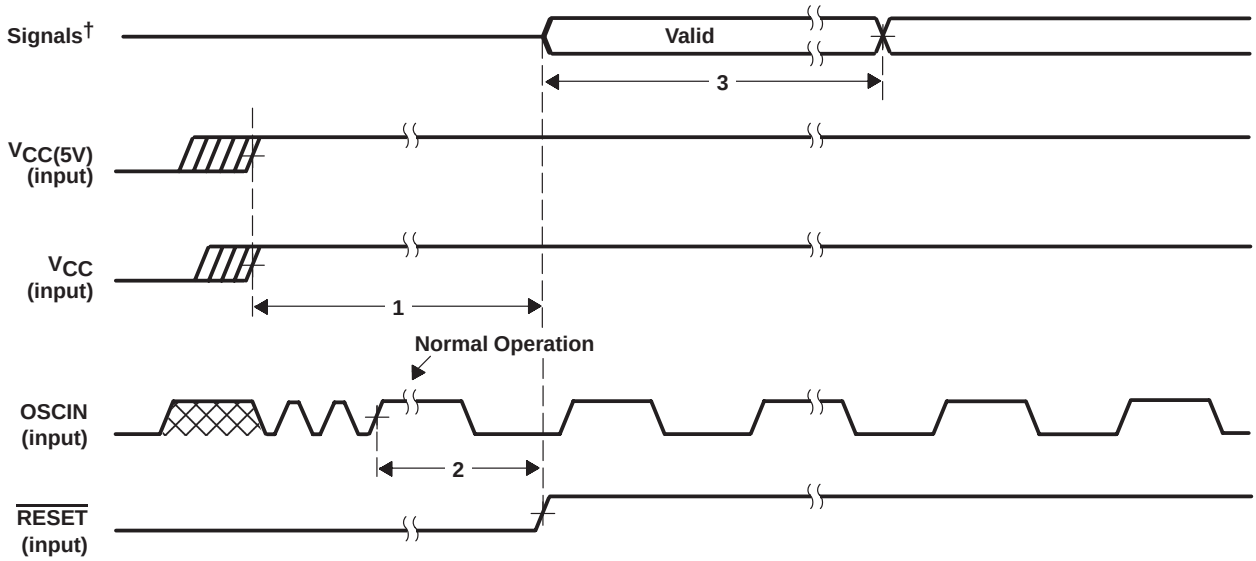


Figure 27. Ports (02–09) Transmit Interface Timing

timing requirements (see Figure 28)
power-on reset

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

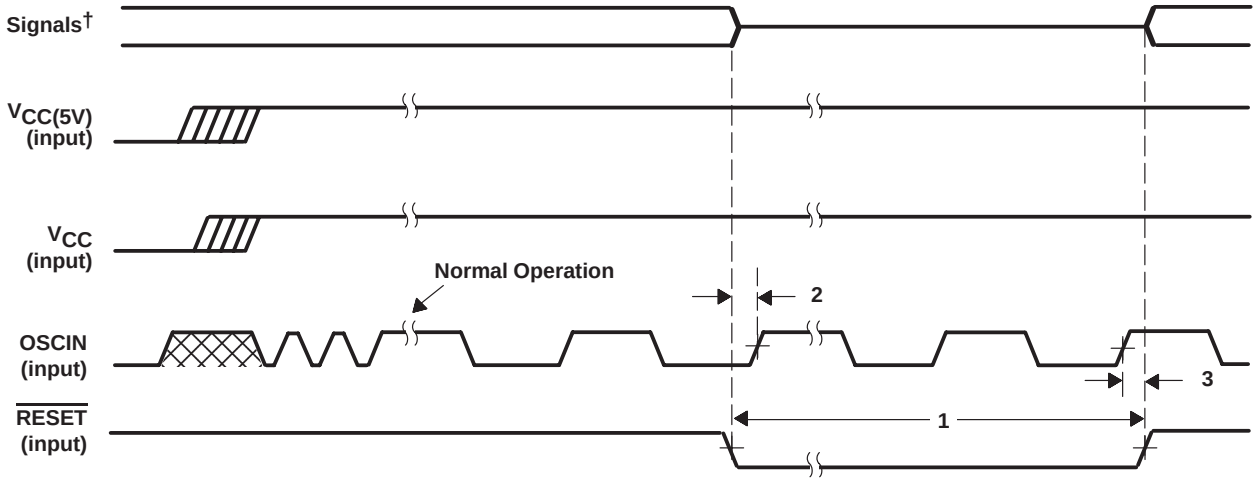


† MXXCOL, MXXCRS, MXXPROTOCOL, MXXDUPLEX, MXXLINK, MXXRCLK, MXXRXD0, MXXRXD1, MXXRXD2, MXXRXD3, MXXRXD4, M00RXD5, M00RXD6, M00RXD7, M00RXDV, M00RXDVX, MXXRXER, MXXSPEED, MXXTCLK, M00UPLINK, DD35–DD0, EAM15–EAM00, SDATA7–SDATA0, SAD1, SAD0, SCS, SRNW, EDIO, TDI, TMS, TRST, TCLK, OSCIN

Figure 28. Power-On Reset Timing

timing requirements over recommended operating conditions (see Figure 29)
RESET timing

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

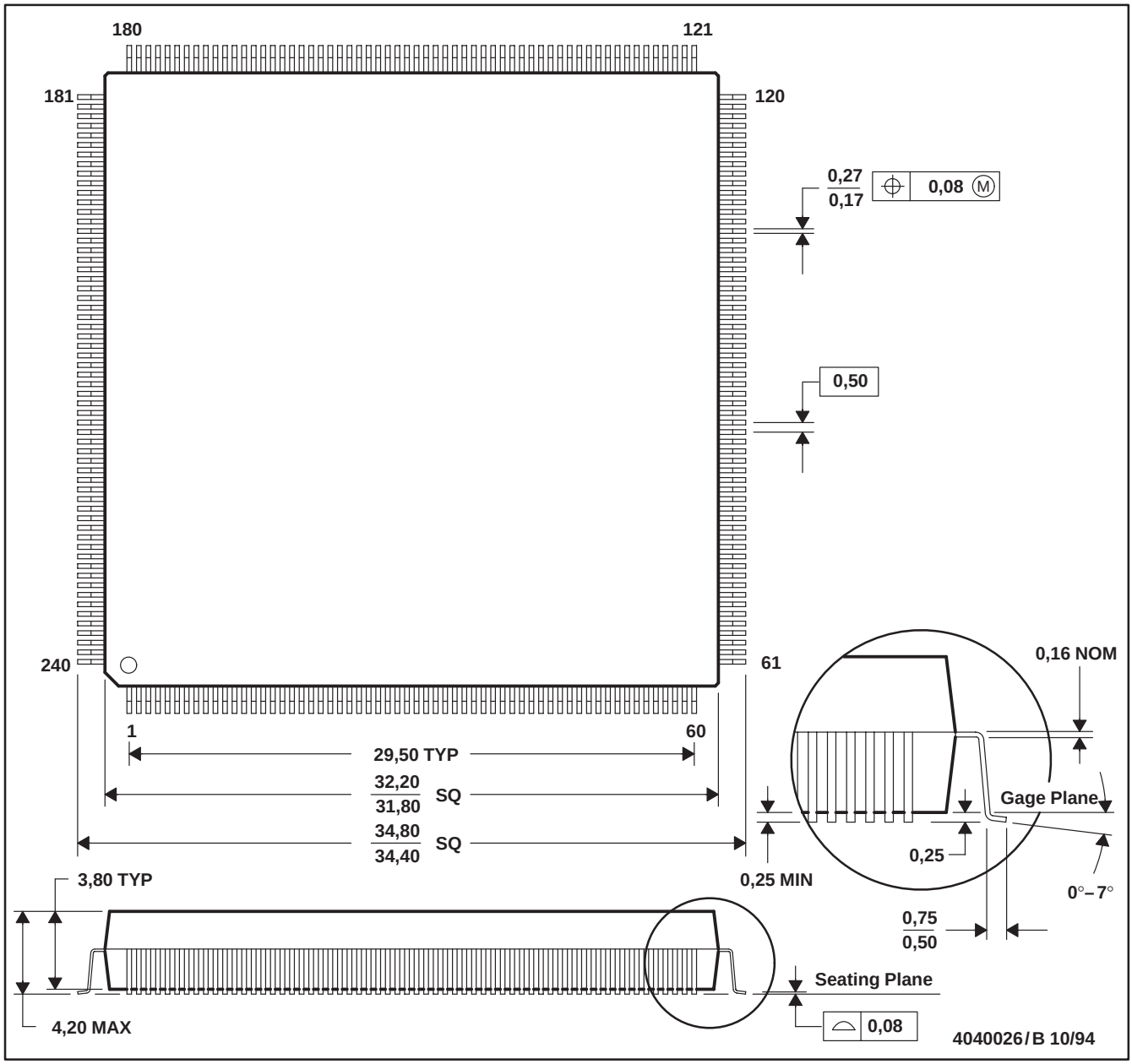


† MXXCOL, MXXCRS, MXXPROTOCOL, MXXDUPLEX, MXXLINK, MXXRCLK, MXXRXD0, MXXRXD1, MXXRXD2, MXXRXD3, MXXRXD4, M00RXD5, M00RXD6, M00RXD7, M00RXDV, M00RXDVX, MXXRXER, MXXSPEED, MXXTCLK, M00UPLINK, DD35–DD0, EAM15–EAM00, SDATA7–SDATA0, SAD1, SAD0, SCS, SRNW, EDIO, TDI, TMS, TRST, TCLK, OSCIN

Figure 29. RESET Timing

MECHANICAL DATA

PGC (S-PQFP-G240) **PLASTIC QUAD FLATPACK**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish	MSL Peak Temp (3)
TNETX3100PGC	OBSOLETE	QFP	PGC	240		TBD	Call TI	Call TI
TNETX3100PGW	OBSOLETE	HQFP	PGY	240		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), Pb-Free (RoHS Exempt), 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.
Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.
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 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 TI 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. Information of third parties may be subject to additional restrictions.

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.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

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