Using the Extended Port Awareness Feature of the TNETX4090 ThunderSWITCH

APPLICATION REPORT: SPWA018A

Enterprise Business Unit
Network and Computer Products
April 1998

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

TRADEMARKS

TI and ThunderSWITCH are trademarks of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

CONTACT INFORMATION

PIC Telephone	(972) 644-5580
PIC FAX	(972) 480-7800
Email	networks@ti.com

Contents

Abstract **7**
System Overview **8**
Crossbar Matrix/TNETX4090 Interface 9
Pre-Tag 10
Transmit Pre-Tag 11
Learning Format Receive Pre-Tag 11
Directed Format Receive Pre-Tag 12
TNETX4090 Configuration **14**
System Control Configuration 14
Port 08 Configuration 17
Extended Addresses 19
xroutecodes/xportcodes 23
Appendix A.Crossbar-Matrix Switch Design Considerations **26**
Crossbar-Matrix Switch 27
Crossbar-Matrix Port 27
Crossbar-Matrix Design Criteria 28
Appendix B.Product Support **29**
Related Documentation 29
World Wide Web 29
Email Support 29

Figures

Figure 1. High Port Density System 8
Figure 2. GMII Interface 9
Figure 3. Transmit Pre-Tag Timing Diagram 10
Figure 4. Receive Pre-Tag Timing Diagram 10
Figure 5. AddPort Register – Multicast Format 21
Figure 6. Crossbar-Matrix Switch Architecture 27

Tables

Table 1. SysControl Register 14
Table 2. FlowThreshold Register 17
Table 3. Port08Control Register 18
Table 4. Port8QTAG Register 19
Table 5. AddPort Register – Unicast Format 19
Table 6. XMultiGroup17 - 63 Registers 23
Table 7. xroute/xport Code Table 25

Using the Extended Port Awareness Feature of the TNETX4090 ThunderSWITCH

Abstract

This document describes how the extended port awareness feature of the Texas Instruments (TI™) ThunderSWITCH™ TNETX4090 enables system vendors to design a high port density system.

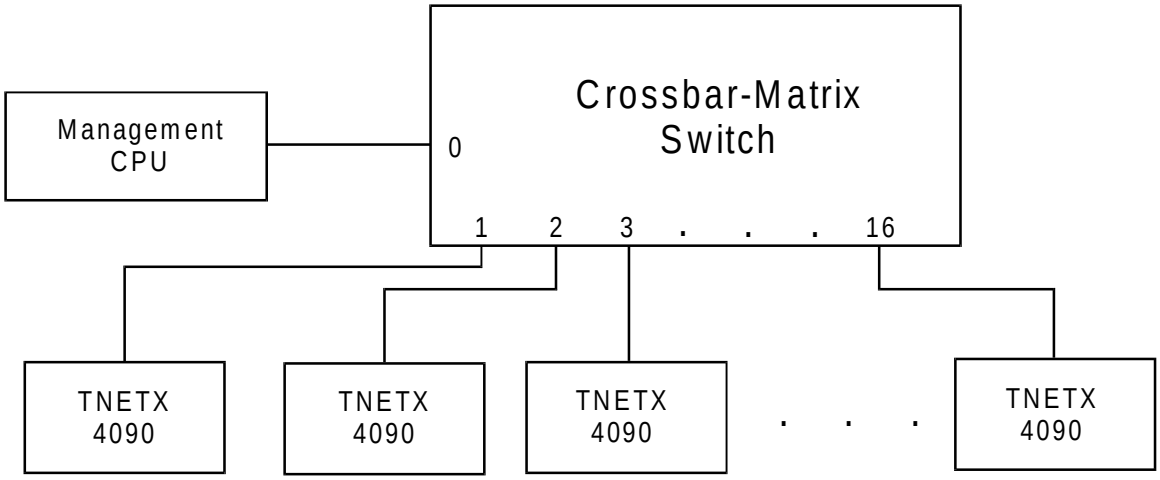
The TNETX4090 gigabit port (port 08) can be configured to support extended port awareness. This feature allows the TNETX4090 to comprehend a proprietary in-band-pre-tagging scheme, which replaces the preamble and start of frame (SOF) delimiter. The pre-tag contains, among other information, the forwarding decision to an extended port, which could be comprehended by a crossbar matrix switch. By designing a relatively simple and cheap crossbar matrix switch, a non-blocking, high port density system is achievable.

System Overview

The extended port awareness allows the TNETX4090 to be aware and perform forwarding and filtering decisions of up to 17 ports that would be residing on the crossbar matrix switch (see Figure 1). The proprietary in-band pre-tags contain the forwarding decision that allows the crossbar matrix to be relatively simple and inexpensive.

Figure 1 shows a block diagram of a high port density system:

Figure 1. High Port Density System



It is recommended that one port on the crossbar matrix be used as the network management port for the system. The management CPU receives frames forwarded by the TNETX4090 devices. Depending on the higher layer functionality the system supports, the TNETX4090 devices can be configured to forward BPDUs for the spanning tree algorithm, SNMP packets for network management, and IGMP frames for IP multicast to the management CPU. Note that if an external CPU is connected to the crossbar-matrix switch, the CPU must supply the correct CRC for the frames it will source.

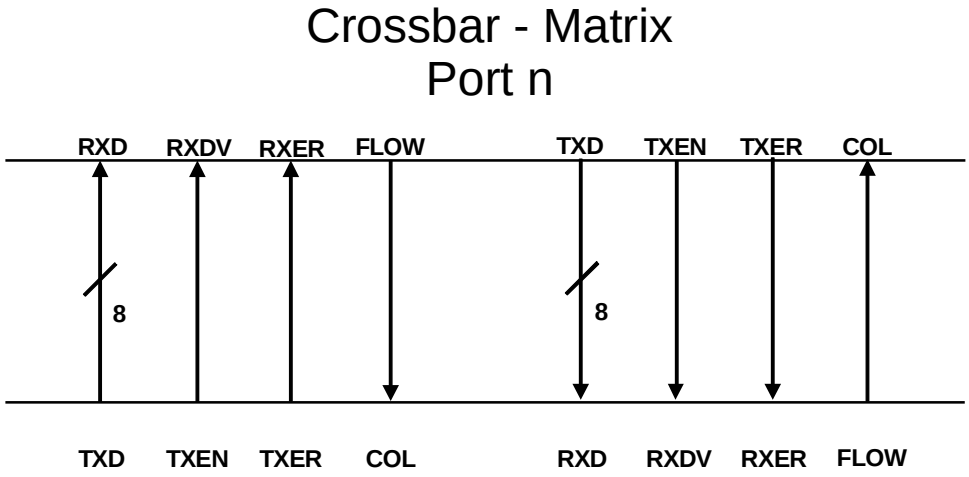
NOTE:

The TNETX4090 devices must be configured by either a CPU or an EEPROM. It is recommended that the Management CPU be used configure the TNETX4090 devices. To manage the entire system, the Management CPU should be able to perform all operations listed in the *TNETX4090 Programmer's Reference Guide* on each individual device.

Crossbar Matrix/TNETX4090 Interface

The crossbar matrix switch must interface with the TNETX4090 via the GMII interface (see Figure 2).

Figure 2. GMII Interface



To prevent the loss of data, it is recommended that the crossbar matrix switch support flow control. When the TNETX4090 experiences congestion (that is, the flow control threshold has been crossed), the **FLOW** pin is asserted. When the **FLOW** pin is asserted, the crossbar matrix switch must stop transmitting frames to the congested TNETX4090.

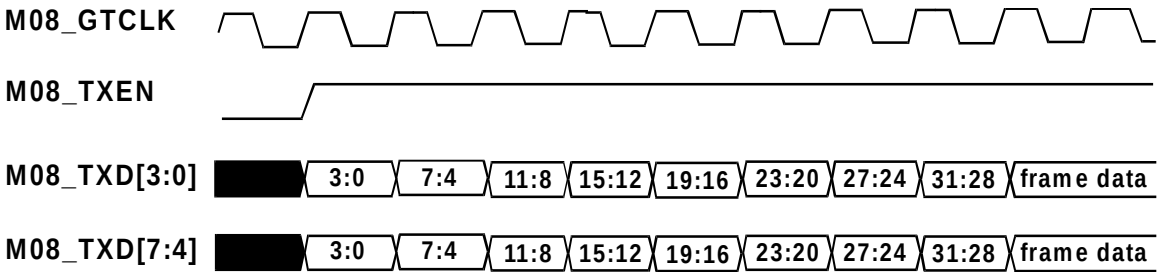
Likewise, when the crossbar matrix switch experiences congestion, it must assert its **FLOW** pin (which must be connected to the TNETX4090 **COL** pin). When the TNETX4090 samples **COL** as asserted, it prevents transmitting frames to the crossbar switch.

NOTE:
Refer to the *TNETX4090 Single-Chip 100/1000 Device* data sheet for timing on the GMII interface.

Pre-Tag

When the TNETX4090 is configured for extended port awareness, the in-band pre-tag, which contains the forwarding decision, replaces the preamble and the start of frame (SOF) delimiter on frames transmitted and received on port 08. The pre-tag is output nibble wide on the data lines and is 32 bits in length.

Figure 3. Transmit Pre-Tag Timing Diagram

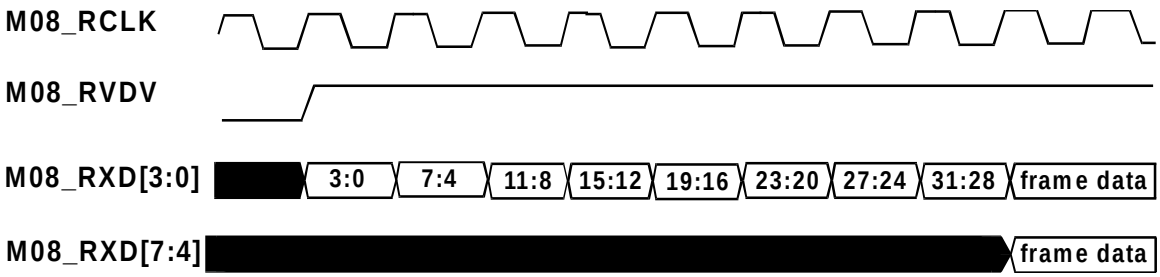


Note: Number ranges indicate which nibbles of the 32 bit pre-tag are output in each cycle.

When a frame is transmitted from port 08, the pre-tag is contained on **M08_TXD[3:0]** and mirrored on **M08_TXD[7:4]** (see Figure 3).

When a frame is received on port 08, the pre-tag is expected to be contained on **M08_RXD[3:0]** (see Figure 4). Note that no information is contained on **M08_RXD[7:4].**

Figure 4. Receive Pre-Tag Timing Diagram



Note: Number ranges indicate which nibbles of the 32 bit pre-tag are input in each cycle.

The TNETX4090 supports three pre-tag formats when configured for extended port awareness (that is, the ***pretag*** bit is set to 1 in the ***Port08Control*** register):

- Transmit pre-tag format
- Learning format receive pre-tag
- Directed format receive pre-tag

Transmit Pre-Tag

Transmit pre-tag format is used when the frame is transmitted from the TNETX4090 to the crossbar matrix switch. The pre-tag contains the forwarding decision (that is, ***xportvector*** indicates the destination portvect), an indication of whether an 802.1Q-like header was inserted (that is, the ***rxheader*** bit), and which port on the TNETX4090 received the frame (that is, ***portcode***).

Bit	Name	Function
31:28	Reserved	These bits will always be zero.
27	***rxheader***	This bit indicates whether the receiving MAC added an 802.1Q-like header (i.e. TI header) to the frame. See note below for detail. ***rxheader*** = 1, an 802.1Q-like header was added to the frame ***rxheader*** = 0, an 802.1Q-like header was not added to the frame
26:25	Reserved	These bits will always be zero.
24:20	***portcode***	This bit field indicates which port on the TNETX4090 received the frame. Valid ***portcodes*** are 0b00000 – 0b01001, which indicate port 00 through port 09 (port 09 is the network management DIO port). All other codes are reserved and will not be generated.
19:17	Reserved	These bits will always be zero.
16:0	***xportvector***	A bit position set to 1 indicates that the frame must be forwarded to that port on the crossbar matrix switch. A multicast may have several bit set in the vector. The TNETX4090 never forwards a frame with a ***xportvector*** of all 0s.

The ***rxheader*** bit indicates to the Management CPU whether the port that received the frame added an 802.1Q-like header. The Management CPU must use this information and create a frame with or without the 802.1Q-like header when sending a frame to a TNETX4090 port. (For detailed information on the 802.1Q-like header (TI header), refer to the *TNETX4090 Programmer's Reference Guide*.)

Learning Format Receive Pre-Tag

Learning format receive pre-tag is used when the TNETX4090 receives a frame sourced from a data port on the crossbar matrix switch (that is, not sourced from the management CPU port). It is the responsibility of the crossbar matrix switch to alter the transmit pre-tag to a learning format receive pre-tag.

Bit	Name	Function
31	***type***	Receive frame indicator. When bit 31 equals 0, the receive pre-tag is learning format.
30:5	Reserved	These bits will be ignored
4:0	***cportcode***	Indicates which port on the crossbar matrix switch received the frame. Valid ***cportcodes*** are 0b00000 – 0b10000 which indicate crossbar matrix switch ports 0 through 16. All other codes are reserved and will be ignored.

The source address of a frame received on the TNETX4090 with a learning format receive pre-tag will be learned. The port on which this node resides (according to the TNETX4090) is the ***cportcode*** of the frames pre-tag. The frame is forwarded according to the frame routing algorithm using the destination address of the frame.

It must be pointed out that the TNETX4090 converts the ***cportcode*** into a ***xportcode***. The ***xportcode*** is stored in the TNETX4090's address table.

NOTE:
The TNETX4090 uses the ***xportcode*** to create the transmit pre-tag when forwarding a frame to the particular node on an extended port.

Directed Format Receive Pre-Tag

The directed format receive pre-tag is used when the frame is sourced from the management port on the crossbar matrix switch.

Bit	Name	Function
31	***type***	Receive frame indicator. When bit 31 equals 1, the receive pre-tag is directed format.
30:10	Reserved	These bits will be ignored
9:0	***portvector***	A bit position set to 1 indicates that the frame must be forwarded to that port on the TNETX4090 (bit 9 set to 1 indicates the destination is the NM port). Several bits may be set in the ***portvector*** to send the same frame to multiple ports.

Frames in directed format bypass T-Bird's frame routing algorithm (see the TI *TNETX4090 Programmer's Reference Guide* for information on the frame routing algorithm). It does not matter if the frame is multicast or unicast, the destination address known or unknown; the frame is forwarded only to the ports as specified in the ***portvector*** of the pre-tag. (If the ***portvector*** is set to all 0s, the frame will be discarded by theTNETX4090). If port 08 (that is, the source port) is set in the ***portvector***, the frame is forwarded back out on port 08.

To support the spanning tree algorithm on the external CPU (the CPU connected to the crossbar-matrix switch), the ***TxBlockPorts***, ***RxUniBlockPorts***, and ***RxMultiBlockPorts*** registers have no effect on the reception or transmission of a frame in direct format.

TNETX4090 Configuration

This section focuses on the configuration of the TNETX4090 when connected to a crossbar-matrix switch, including:

- How to configure the ***SysControl*** register and port 08 to interface correctly with the crossbar-matrix switch
- How nodes on the crossbar-matrix switch ports are addressed
- How the xroute codes work in conjunction with features such as mirroring, unknown destination and VLAN, and unknown source

Please refer to the *TNETX4090 Programmer's Reference Guide* regarding all other aspects of configuring the TNETX4090.

System Control Configuration

The system must prevent data from being lost due to congestion by utilizing flow control. For the system to support flow control, the ***flow*** bit in the ***SysControl*** register must be set to 1.

Table 1. SysControl Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field Name	***stop***	***load***	***start***	***inltd***	Reserved[11:9]			***cnotify***	***nage***	***dmalnc***	***unvlan***	***portcode***	***mlrr***	***nauto***	***ncrc***	***flow***	Byte Address Offset 0x0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Reset type	h	h	h	hs	–	–	–	h	h	h	h	h	h	h	h	h	
Field type	rw	rw	rwl	r	r	r	r	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	

Bit	Name	Function
15	***stop***	*Stop system.* When ***stop*** = 1, device operation is halted and will remain so until this bit is cleared to zero. All internal state machines, FIFOs, and MACs are reset. Register bits are cleared as per the register definition tables in this section. The HOST registers are not affected. Any data in the device Is lost. ❑ This bit will become set to one if an EEPROM load fails because of a bad CRC word. ❑ This bit will become set to one if a parity error is detected on the external memory interface.

Bit	Name	Function
14	***load***	*Load System.* Writing a 1 to this bit causes the DIO registers to be auto-loaded from an external EEPROM (if present). Writing a zero to this bit has no effect. Writing a one to this bit also clears the CRC-error indication from the fault-LED. All DIO operations are inhibited ($\overline{\textbf{SRDY}}$ will be held inactive high) while ***load*** is a one once the EEPROM state machine has finished checking that **EDIO** is not held at zero, and begins to download data. ***Load*** will clear to zero once the download is completed. When this bit is written with a one the ***stop*** bit should simultaneously be written with a zero.
13	***start***	*Start system.* Writing a one to this bit causes the device to begin operation following a reset or stop. This bit will be read as a one until buffer and address lookup memory initialization is complete. This will result in any previous frames or address records being erased. While the memory is being initialized all ports are disabled. All DIO writes are inhibited ($\overline{\textbf{SRDY}}$ inactive high) while ***start*** is a one. Writing a one to this bit also clears the parity-error indication from the fault-LED. Writing a zero to this bit has no effect, as does writing a one when ***initd*** – 1. When this bit is written with a one the ***stop*** bit should simultaneously be written with a zero.
12	***Initd***	*Initialization Done.* This bit becomes a one after ***start*** has been set, and buffer and address lookup memory initialization is complete (i.e. at the same time ***start*** is cleared). Once set, it remains so until a hard reset (pin or DIO), or until ***stop*** is set to one. Until this bit becomes a one the ports will ignore any frames they receive.
11:9	**reserved**	*Reserved.* Writes to these bits have no effect. They always read as zero.
8	***cnotify***	*Change Notification.* Determines what action is taken when a known source address is received on a port other than the one currently associated with the address: ❑ If ***cnotify*** = 0, change notification is disabled. ❑ If ***cnotify =*** 1, change notification is enabled. The frame is additionally forwarded to the ports specified by ***UnkSrcPorts*** ANDed with the appropriate ***VLANnPorts*** register, provided that address learning is enabled for the port the address has moved to (appropriate bit in ***NlearnPorts*** = 0). This allows the address lookup tables of other switches to be updated in accordance with the change.
7	***nage***	*No aging:* Allows automatic address-record aging to be disabled independent of learning. (***nauto*** – 1 disables both learning and aging). ❑ ***Nage*** = 1 aging is disabled even if ***nauto***=0. ❑ ***Nage*** = 0 aging is enabled unless ***nauto***=1. If ***nauto***=1 then the value of ***nage*** is don't care – aging will be disabled. This bit has no effect on address learning or time-stamp updating.
6	***dmainc***	*DMA Address Autoincrement.* When accessing the DIO interface with a DMA controller ($\overline{\textbf{SDMA}}$ pin low), this bit determines whether the address held in ***dmaadress*** should be incremented between successive accesses or not: ❑ ***Dmainc*** = 1, the address will increment between accesses. ❑ ***Dmainc*** = 0, the address will not increment between accesses.

Bit	Name	Function
5	***lshare***	*Load Sharing.* If a frame is destined for a port which is part of one of the trunk groups defined by the ***Trunk0Ports – Trunk3Ports*** registers, the eight entry lookup table formed by the ***TrunkMap0 – TrunkMap7*** registers is used to determine which member of the trunk group the frame will be transmitted on. This bit determines how to index into the lookup table: ❑ ***Lshare*** = 1, if the destination address is unknown, the index is computed using the source address only by exclusive-ORing bits 47:32, 31:16 and 15.0 that address. If the destination address is known, the frame will be routed according to the port associated with the address. ❑ ***Lshare*** = 0, the index is computed using both the source and destination addresses, by bitwise exclusive-ORing bits 3:1 of the destination address with bits 3:1 of the source address.
4	***unkvlan***	*Unknown VLAN Forwarding.* This bit enables forwarding of multicast frames that have an unknown IEEE802.1Q VLAN Identifier (i.e. one that matches none of the VLAN Ids registered in ***VLANOQID – VLAN63QID***). ❑ ***Unkvlan*** = 1, all multicast frames belonging to unknown VLANs will be forwarded to the port indicated in ***UnkVLANPort.*** ❑ ***Unkvlan*** = 0, all multicast frames belonging to unknown VLANs will be discarded.
3	***mirr***	*Port Mirroring.* This bit enables port mirroring. ❑ ***Mirr*** *= 1,* all frames received on or sent to the port specified in ***MirrorPort*** will be copied to the port specified in ***UplinkPort.*** ❑ ***Mirr*** = 0, no mirroring is performed.
2	***nauto***	*NOT Automatically Add Address Mode.* This bit selects the manner in which addresses will be added to the address records: ❑ ***Nauto*** = 1, addresses can only be added to the records using DIO adds. The aging state machine will be disabled; it is management's responsibility to manage the address records. ❑ ***Nauto*** = 0, new addresses will be learned from the wire and added to the address tables automatically. (They can also be added using DIO adds). If however ***nage***=1 and the address tables become full then further addresses will not be added unless space is first created usimg DIO deletes.
1	***ncrc***	*No CRC Check.* This bit determines whether addresses are learned if the frame containing them contains an invalid CRC: ❑ ***Ncrc*** = 1, new addresses are added to the lookup table without regard for the validity of the CRC. Unknown addresses will be added to the lookup table automatically. ❑ ***Ncrc*** = 0, new addresses can only be added to the lookup table if the frame containing the address has a valid CRC.
0	***flow***	*Flow Control Enable.* This bit enables flow control: ❑ ***Flow*** = 1, the device will implement flow control. Ports configured for IEEE 802.3X will use pause frames. Ports thatare not will use collisions. ❑ ***Flow*** = 0, no flow control will be implemented.

When the TNETX4090's buffer threshold has been broken (that is, the TNETX4090 is considered to be congested) the **FLOW** pin will be asserted. The **FLOW** pin holds off the crossbar-matrix switch from transmitting frames to the TNETX4090. (See Appendix A for details.)

The TNETX4090's buffer threshold is configurable via the ***FlowThreshold*** register. When the external RAM contains fewer free buffers then the threshold value then the TNETX4090 is considered congested.

Table 2. FlowThreshold Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Byte Address Offset0 0x0
Field Name	***Flowthreshold[15:0]***																
Reset value	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	
Reset type	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	
Field type	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	

Bit	23	22	21	20	19	18	17	16	Byte Address Offset0 0x0
Field Name	reserved[23:16]								
Reset value	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	
Reset type	–	–	–	–	–	–	–	–	
Field type	r	r	r	r	r	r	r	r	

Bit	Name	Function
23:16	reserved	*Reserved*. Writes to these bits have no effect. They always read a 0.
15:0	***flowthreshold***	*Flow-control threshold value.* When the number of free buffers in external memory that are available for frame reception drops below this value then flow control is initiated (providing *flow* is set to one in *SysControl*). Flow-control ceases when the free-buffer count equals or exceeds the value. Note that depending on the amount of memory in the system it may be possible to program this value to be greater than the total number of buffers in the system, thereby permanently inducing flow control.

Port 08 Configuration

For the TNETX4090 to interface correctly to the crossbar-matrix switch, the Gigabit port (port 08) must be configured correctly. This is accomplished by configuring both the ***Port08Control*** and ***Port8QTag*** registers.

Table 3. Port08Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Field Name	reserved			***maxlen***	***txacc***	***rxacc***	***pretag***	***neg***	***disable***	***reqntxp***	***lckref***	***reqnrxp***	***reqpma***	***req100***	***txpace***	***reqhd***	DIO Address 0x0000 +0x2★x
Reset value	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	
Reset type				h	h	h	h	h	h	H	h	h	h	h	h	h	
Field type	r	r	r	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	

The maxlen bit is the only bit of the Port8Control register that is up to the user to configure. All other bits in this register must be set to the values listed in the following table to support the extended port awareness feature.

Bit	Name	Value	Explanation
12	***maxlen***	0 or 1	User configurable to support 1518-byte or 1535-byte frames.
11	***txacc***	0	Prevent the Gigabit MAC from stripping the VLAN tag.
10	***rxacc***	0	Prevent the Gigabit MAC from adding a VLAN tag.
9	***pretag***	1	Configures the Gigabit port to comprehend extended port pretags.
8	***neg***	0	Prevents the Gigabit MAC from advertising its capability.
7	***disable***	0	Port 08 must be enabled.
6	***reqntxp***	1	Prevents the Gigabit MAC from generating 802.3X pause frames. Flow control is accomplished via the **FLOW** pin.
5	***lckref***	0	Lock to reference feature is not used when port 08 is connected to a crossbar-matrix switch.
4	***reqnrxp***	1	Prevents the Gigabit MAC from comprehending 802.3X pause frames. Flow control is accomplished via the **FLOW** pin.
3	***reqpma***	0	The Gigabit port must interface to the crossbar-matrix switch via the GMII. Port 08 must not be configured in PCS mode.
2	***req100***	0	Port 08 must be configured for 1000Mbps mode.
1	***txpace***	0	Prevents the Gigabit MAC from using ***txpace*** (mode can only be used when connecting to other TI devices which supports txpace).
0	***reqhd***	0	Port 08 must be configured for full duplex.

The ***Port8QTag*** register must be configured to 0x000 because the Gigabit port is used as a backplane for the system. As such, the port must be transparent to the system. This means that the Gigabit port must not add or strip a VLAN tag off a frame. This is accomplished by configuring both the ***txacc*** and ***rxacc*** bits in the ***Port08Control*** register to 0 and the ***Port8QTag*** register to 0x000.

Table 4. Port8QTAG Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	DIO Address 0x0000 +0x2★x
Field Name	reserved[15:12]				***Vlanqid[11:0]***												
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Reset type	–	–	–	–	h	h	h	h	h	h	h	h	h	h	h	h	
Field type	r	r	r	r	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	rwl	

Bit	Name	Function
15:12	**reserved**	*Reserved*. Writes to these bits have no effect. They always read a 0.
15:0	***flowthreshold***	*IEEE 802.1Q VLAN Identifier*. Identifies the default IEEE 802.1Q VLAN ID associated with each port. How it is used depends on the value of ***rxacc*** in ***PortxControl***: ❑ ***rxacc*** = 1, an IEEE 802.1Q tag header is always added to the frame (even if the frame is already tagged). The default source port VLAN ID ccontained in this field is used. ❑ ***Rxacc*** = 0, if a received frame contains no IEEE 802.1Q tag header, one is added using the default source port VLAN ID contained in this field. If the frame has a tag header with an ID of 0x000, it is replaced by this field.

Extended Addresses

Nodes residing on extended crossbar matrix ports can be learned from the wire (the frame received on port 08 must contain a pre-tag with learning format) or via the DIO. Multicasts can only be learned via the DIO interface. (For detailed information on how to add and delete address records as well as search the address records table, see the *TNETX4090 Programmer's Reference Guide*.)

Table 5. AddPort Register – Unicast Format

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Byte Address Offset0 0x0
Field Name	reserved[15:8]								*reserved*		***Xportcode[5:0]***						
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Reset type	–	–	–	–	–	–	h	h	h	h	h	h	h	h	h	h	
Field type	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Byte Address Offset0 0x2
Field Name	*nblck*	*secure*	*locked*	*cuplnk*	*new*	Reserved[26:16]											
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Reset type	h	h	h	h	h	h	h	h	–	–	–	–	–	–	–	–	
Field type	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r	

Bit	Name	Function
31	***nblck***	*Not Blocked Flag*. This bit sets the ***RxUniBlockPorts*** override function. ❑ ***Nblck*** = 1, ***RxUniBlockPorts*** will not be used to filter frames. ***Nblck*** = 0, frames will be forwarded only if the appropriate ***RxUniBlockPorts*** bit for the source port is not set. Locked against modification while an Add is in progress – writes have no effect.
30	***secure***	*Secured Address Flag*. This bit determines the security level for the address contained in ***AddNode***. Locked against modification while an Add is in progress – writes have no effect.
29	***locked***	*Locked Address Flag*. This bit determines the lock status for the address contained in ***AddNode***. Any frames received from a locked source will be discarded. Locked against modification while an Add is in progress – writes will have no effect.
28	***cuplnk***	*Copy Frames to Uplink Flag*. This bit determines the Copy Uplink status for the address contained in ***AddNode***. Addresses tagged with this bit will add the port specified in the ***UplinkPort*** register to the routing code. Locked against modification wile an Add is in progress – writes have no effect.
27	***new***	*New Address Flag*. This bit determines the new status for the address contained in ***AddNode***. Having set ***new*** to 1, this address can be easily found by a *Find* command which has ***new*** in ***FindControl*** = 1. Locked against modification while an Add is in progress – writes have no effect.
26:8	**reserved**	*Reserved*. Writes to these bits have no effect. They always read as zero.
7:6	**reserved**	*Reserved*. Writes to these bits have no effect. They always read as zero.
5:0	***xportcode***	*Current Extended Port for Node*. This filed determines the destination port for the unicast address shown in ***AddNode***. If a nonexistent port is specified, no address record will be entered when the add is performed. (The add will complete in the normal manner but no record will exist.) Locked against modification while an Add is in progress – writes have no effect.

Figure 5. AddPort Register – Multicast Format

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Byte Address Offset0 0x0
Field Name	reserved[15:10]						***portvector[5:0]***										
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Reset type	–	–	–	–	–	–	h	h	h	h	h	h	h	h	h	h	
Field type	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Byte Address Offset0 0x2
Field Name	*nblck*	*reserved*	*Xroutecode[29:24]*						Reserved[23:16]								
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Reset type	h	h	h	h	h	h	h	h	–	–	–	–	–	–	–	–	
Field type	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r	

Bit	Name	Function
31	***nblck***	*Not Blocked Flag*. This bit sets the ***RxMultiBlockPorts*** override function. ❑ ***Nblck*** = 1, ***RxMultiBlockPorts*** will not be used to filter frames. ***Nblck*** = 0, frames will be forwarded only if the appropriate ***RxMultiBlockPorts*** bit for the source port is not set. Locked against modification while an Add is in progress – writes have no effect.
30	**reserved**	*Reserved*. Writes to these bits have no effect. They always read as zero.
29:24	***xroutecode***	*Extended Route Code for Multicast*. This field determines the route code used to generate the pre-frame tag added to frames routed to port 08 when ***pretag*** in ***Port8Control*** = 1. It is ignored if ***pretag*** or bit 8 of ***portvector*** is 0. Locked against modification while an Add is in progress – writes will have no effect.
23:10	**reserved**	*Reserved*. Writes to these bits have no effect. They always read as zero.
9:0	***portvector***	*Port Bit Vector for Multicast*. This field determines the port bit vector for the multicast address contained in ***AddNode***. The bit values in this field correspond one to one with the port assignment. Locked against modification while an Add is in progress – writes have no effect.

When a frame with a learning format pre-tag is received on port 08, the internal address lookup engine (IALE) will learn/update the address, node, and VLAN if the frame is a unicast. When learning the address, the IALE takes the ***xportvector*** of the pre-tag and converts it to a ***xportcode*** (the xportcodes for crossbar matrix switch ports range from 0b100000 to 0b110000, where 0b100000 represents crossbar matrix port 0 and 0b110000 represents crossbar-matrix port 16).

The ***xportcode***, along with the source address and the VLAN information, is added to the TNETX4090 address record table. The same information can be learned via the DIO interface.

The ***xroutecode*** of the ***AddPort*** register and the ***XMultiGroup*** registers allows the system to include crossbar-matrix ports in a multicast address. The following conditions must be met for a crossbar-matrix port to be included in the multicast:

- The ***pretag*** bit in the Port08Control register must be set to 1.
- Bit 8 (indicating port 08) of ***portvector*** in the ***AddPort*** register must be set to 1.

When the conditions above are met, the multicast frame will be forwarded to all ports on the TNETX4090 as specified by the ***portvector*** in the ***AddPort*** register. Furthermore, the multicast frame will also be forwarded to the crossbar-matrix port(s) as specified in the ***xroutecode*** in the ***AddPort*** register.

If the value of the ***xroutecode*** is in the range of 0b000000 to 0b010000 (where 0b000000 indicates crossbar-matrix port 0 and 0b010000 indicates crossbar-matrix port 16), the multicast frame will be forwarded only to the specified crossbar-matrix port. If the value of the xroutecode is in the range of 0b010001 to 0b111111 (where 0b010001 indicates ***XMulitGroup17*** register and 0b111111 indicates ***XMulitGroup63*** register), the multicast frame will be forwarded to the crossbar-matrix ports specified in the ***XMultiGroup*** register the ***xroutecode*** indexes.

Table 6. XMultiGroup17 - 63 Registers

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	DIO Address Offset 0x0500 +4★n
Field Name	***xportvector15:0]***																
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Reset type	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	
Field type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	DIO Address Offset 0x0500 +4★n
Field Name	Reserved[31:17]															***xportvector***	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Reset type	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	h	
Field type	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	

Bit	Name	Function
31:17	**reserved**	*Reserved*. Writes to these bits have no effect. They always read a 0.
16:0	***xportvector***	*Extended Multicast Group Port Vector*. This field determines the extended port vector to be inserted into the pre-frame tag added to frames routed to port 08 when ***pretag*** in ***Port8Control*** =1. ***XMultiGroupn*** is associated with ***xroutecode*** = *n*, where *n* is in the range 17 – 63. If ***xroutecode*** is less than 17, it is interrupted as a unicast extended port vector, and only bit *n* of the vector is set.

The ***xportvector*** in the ***XMultiGroupn*** registers determine which crossbar-matrix ports must forward the multicast frame. The bit position in the ***xportvector*** indicates the crossbar-matrix port (for example, bit 0 set to one indicates the frame must be forwarded on the crossbar-matrix port 1, bit 1 set to one indicates crossbar-matrix port 1, and so on).

xroutecodes/xportcodes

Besides generating ***xroutcodes*** or ***xportcodes*** from the IALE performing a lookup on the destination address in the address record table, ***xroutecodes*** and ***xportcodes*** can be generated when utilizing the following features (refer to the *TNETX4090 Programmer's Reference Guide* for detailed information on the features listed below):

- **Unknown VLAN** – If a multicast address is received with an unknown VLAN Identifier, the ***xportcode*** contained in the ***UnkVLANPort*** register will determine which port the frame is forwarded to. This feature is listed as **UnkVLAN** in Table 7.

- **Unknown Source Address** – When a frame is received with an unknown source address, the frame will be forwarded to the port according to the ***portvector*** and ***xroutecode*** in the ***UnkSrcPorts*** register. This feature is listed as **UnkSrc** in Table 7.

- **Unknown Unicast Address** - When a frame is received with an unknown unicast address, the frame will be forwarded to the port according to the ***portvector*** and ***xroutecode*** in the ***UnkUniPorts*** register. This feature is listed as **UnkUni** in Table 7.

- **Unknown Multicast Address** - When a frame is received with an unknown multicast address, the frame will be forwarded to the port according to the ***portvector*** and ***xroutecode*** in the ***UnkMultiPorts*** register. This feature is listed as **UnkMulti** in Table 7.

- **Destination Address Mirroring –** If a frame is received and the ***cuplnk*** bit is set for that destination address, the frame will be mirrored on the port specified in the ***xportcode*** of the ***UplinkPort*** register. This feature is listed as **Cuplink (cuplink)** in Table 7.

- **Port Mirroring –** If the ***mirr*** bit is set to one in the ***SysControl*** register and a frame is sourced from or destined to the mirror port (that is, the mirror port is configured in the ***MirrorPort*** register), the frame will be mirrored on the port specified in the ***xportcode*** of the ***UplinkPort*** register. This feature is listed as **Cuplink (mirror)** in Table 7.

The ***xroutecode*** generated from address record lookup and the ***xroutecode***/***xportcode*** generated in the features listed above are not mutually exclusive. Table 7 depicts how the TNETX4090 resolves ***xroutecode***/***xportcode*** when a conflict occurs.

Table 7. xroute/xport Code Table

UnkVLAN	UnkSrc	UnkUni	UnkMulti	Cuplink (cuplink)	Cuplink (mirror)	Lookup Table	Select
√					√		UnkVLAN
	√				√		Cuplink
	√	√					UnkUni
	√	√			√		Cuplink
		√			√		Cuplink
	√		√				UnkMulti
	√		√		√		Cuplink
			√		√		Cuplink
	√					√	UnkSrc
	√			√			Cuplink
	√				√	√	Cuplink
	√			√	√		Cuplink
	√			√		√	Cuplink
	√			√	√	√	Cuplink
				√		√	Cuplink
				√	√		Cuplink
				√	√	√	Cuplink
					√	√	Cuplink

Appendix A. Crossbar-Matrix Switch Design Considerations

The crossbar-matrix switch can be relatively simple and inexpensive because of the following reasons:

- Minimal frame processing.
- The crossbar-matrix switch does not perform address learning, address lookup, or forwarding decisions.
- The crossbar-matrix switch does not perform any filtering decisions.
- Pin based flow control allows the on-chip buffering to be kept to a minimum.

The crossbar-matrix switch does not have to contain an Ethernet MAC. The only frame processing the crossbar-matrix switch must perform is to change the transmit pre-tag into a receive pre-tag.

The TNETX4090 devices perform the forwarding decisions of frames on the crossbar-matrix switch. The forwarding decision is contained in the transmit pre-tag. Hence, the crossbar-matrix switch does not learn addresses, keep an address table, or perform forwarding decisions. The crossbar-matrix switch must only forward frames according to the ***xportvector*** in the transmit pre-tag. (See the *Transmit Pre-Tag* section for details.)

Filtering, including CRC checking, is performed on both the ingress and egress ports of the TNETX4090 devices. Any frame forwarded to the crossbar-matrix switch should be considered a valid frame. Therefore, the crossbar-matrix switch does not have to provide any filtering decisions. This allows the crossbar-matrix switch to be designed to operate in cut-through mode rather than in store-and-forward mode.

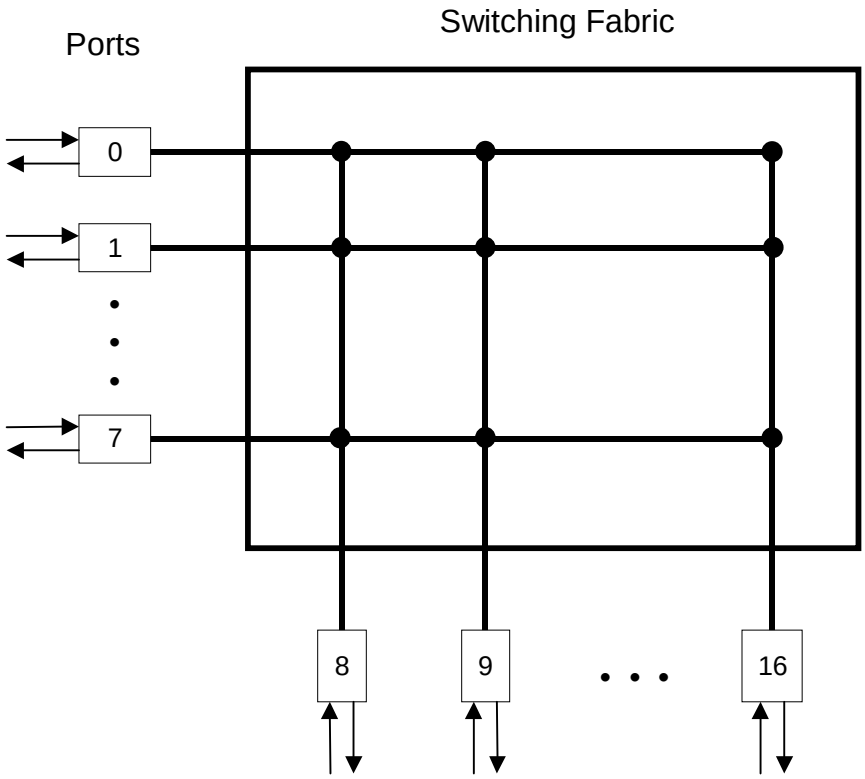
During congestion of the crossbar-matrix switch, the pin-based flow control prevents the TNETX4090 from transmitting frames to crossbar-matrix switch. All frames destined for the crossbar-matrix switch will be backlogged in the TNETX4090's packet memory. This allows the crossbar-matrix switch to keep the on-chip buffering to a minimum.

NOTE:
The aggregate bandwidth of the switching fabric should ultimately determine the size of the on-chip buffering.

Crossbar-Matrix Switch

Figure 3 shows the maximum number of ports that TNETX4090's extended port awareness feature supports. It is recommended that one port be system management (that is, by connecting the Management CPU to this port).

Figure 6. Crossbar-Matrix Switch Architecture



Only the basic crossbar-matrix switching fabric is shown. No recommendation is made regarding which buffer scheme is the superior solution (that is, input-buffer, output-buffer, etc.). Implementation of the switching fabric is up to the user to design.

Crossbar-Matrix Port

The frame processing performed at the ports is minimized since the forwarding decision is performed by the TNETX4090 devices. The only frame processing the port needs to perform is pre-tag manipulation.

The pre-tag (transmit format) of the frame that the port receives contains the forwarding decision. The source port (that is, the port that received the frame) must ensure that the frame is forwarded to the destination ports according to the pre-tag. The port must also alter the pre-tag of the frame into one of the receive formats.

The only other design criteria for the ports is that they must interface to the GMII of the TNETX4090. This includes supporting flow control. The port must stop transmitting to a TNETX4090 device, if that device indicates a state of congestion. Likewise, the port must be able to prevent the transmission from the TNETX4090 to the crossbar-matrix during congestion.

Crossbar-Matrix Design Criteria

The crossbar-matrix switch must be designed to the following criteria:

- Support full duplex over the interface with the TNETX4090 (GMII)
- Support a pin-based flow control. The crossbar-matrix must be able to prevent the TNETX4090 from transmitting new frames over the GMII interface by asserting a "**FLOW**" - equivalent pin. The crossbar-matrix must also stop transmitting new frames when the TNETX4090 asserts its **FLOW** pin on the GMII.
- The crossbar-matrix must forward the received frames to the proper port(s) according to the forwarding decision in the transmit pre-tag.
- The crossbar-matrix must convert the transmit pre-tag into a receive pre-tag.

Appendix B. Product Support

Related Documentation

The following list specifies product names, part numbers, and literature numbers of corresponding TI documentation.

- Data sheet, *TNETX4090 ThunderSWITCH II 9-Port 100-/1000-Mbit/s Ethernet Switch*, March 1998, Literature number SPWS044B
- *TNETX4090 Programmer's Reference Guide*, Literature number SPAU003

World Wide Web

Our World Wide Web site at **www.ti.com** contains the most up to date product information, revisions, and additions. Users registering with TI&ME can build custom information pages and receive new product updates automatically via email.

Email Support

For technical issues or clarification on switching products, please send a detailed email to **networks@ti.com**. Questions receive prompt attention and are usually answered within one business day.