TI Designs

Simple Open Real-Time Ethernet (SORTE) Master With PRU-ICSS Reference Design

Description

The TIDEP-0085 design implements a simple open real-time Ethernet (SORTE) master with the programmable real-time unit and industrial communication subsystem (PRU-ICSS). SORTE protocol enables customer applications to exchange process data between the master and devices in a 4-µs cycle time. The PRU firmware in source code to enable customers to differentiate their products.

The SORTE protocol includes device discovery, parameterization, PHY and cable delay measurement, synchronization, and process data exchange.

Resources

TIDEP-0085	Design Folder
AM3359	Product Folder
TLK110	Product Folder
DP83822I	Product Folder
TMDSICE359	Tool Folder

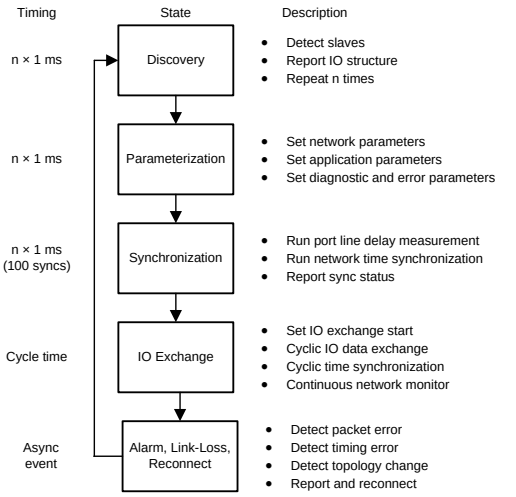
TI E2E™ Community — ASK Our E2E Experts

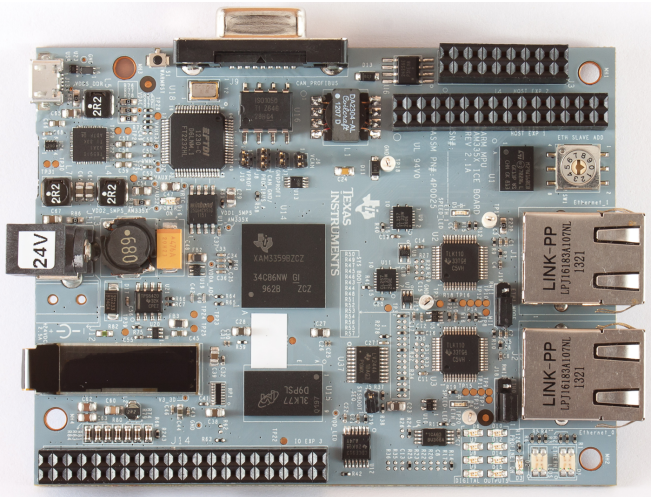
Features

- SORTE Master Reference Implementation
- Supports up to 254 Devices in One Network
- Enables 4-µs Cycle Time to Exchange Process Data
- PRU Firmware Provided in Source Code
- Fully-Customizable PRU Firmware

Applications

- Industrial Ethernet
- Servo Drives and Motion Control
- Programmable Logic Controllers (PLC)
- Industrial Communication Module
- Industrial Input-Output (IO) Modules
- Industrial Sensors and Actuators
- Stepper Drives
- Chip-to-Chip Communication Interface





An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

Copyright © 2017, Texas Instruments Incorporated

1 System Description

Production systems are organized in various levels to manage the production process. Optimization of the production process is only possible with a fully transparent and timely accurate view of the IO functions. Compared to consumer and office equipment, industrial communication must be deterministic and safe. There is cyclic exchange of IO data between devices on the field level of a manufacturing floor and the controller, which manages multiple IO devices organized in line or ring structure. Multiple IO controller (IOC) can be connected at the control level to exchange data between various machines or between machines and automation components, such as robots, conveyor belts, and tool magazines. Communication to office level bridges between the operational technology and the information technology (IT). This bridge function requires security for authentication and data while still maintaining the timing context of IO operation.

Figure 1 shows an overview of an industrial control system.

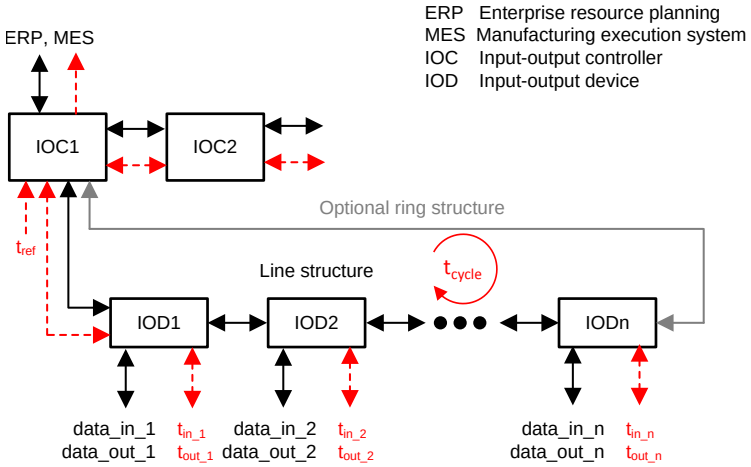


Figure 1. Industrial Control System

IO communication over industrial Ethernet is repeated with a preconfigured cycle time. There are different classes of cycle times in a production system ranging from 31.25 µs for motion control applications to more than 10 ms for a complete manufacturing site. Based on the communication cycle time (t_{cycle}), the IOC sends new data packets to IO devices (IOD). The IOC serves as a timing master with a local time reference (tref). The IOD synchronizes to the master time reference and all IODs in the network have same understanding of time. The time synchronization inside an IO communication network allows giving all input and output data at each node a reference time. For example IOD2 data_in2 is captured at a preconfigured time t_{in_2}. Same behavior for the output data, which is triggered with time t_{out_2}. The time synchronized control of input and output data over a network of many IO devices serves as a basis for PLCs and multi-axis motion controllers, which are used in machine tools and robotics applications.

There are more than 30 known industrial Ethernet standards driven from leading companies of the industrial automation market. The scope of these standards covers not just the physical layer and data link layer but also higher network layers and the interface to the application which programs the IO data over the network. Overall cycle time of the control loop is the sum of various hardware and software processing steps. For networks with more than one device the forwarding delay up to the last device must be considered.

Copyright © 2017, Texas Instruments Incorporated

Figure 2 shows the comparison of industrial Ethernet with processing steps in software on a host controller compared to SORTE using PRU-ICSS.

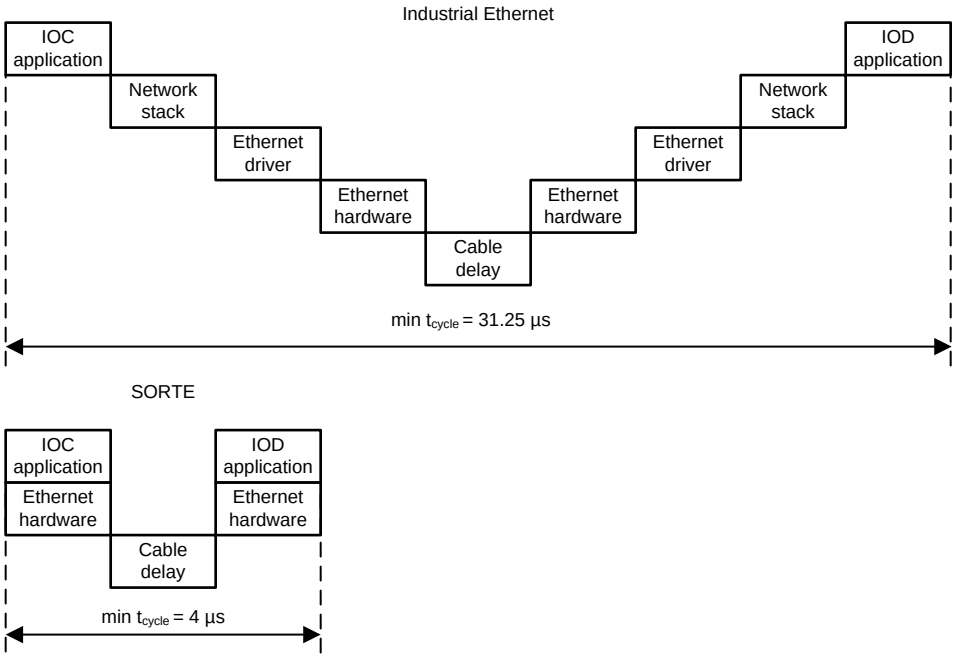


Figure 2. Cycle Time Optimization Through Processing Enhancements

PRU-ICSS is the programmable real-time core with real-time Ethernet peripheral, which allows combining application, networking, and Ethernet processing into a single-processing step. There are no additional variations in the signal chain to transfer data over interconnects, cache memories, and application CPU. The timing model in Figure 2 is a simplified view for a single device connection. When calculating the minimum cycle time, the number of IODs connected in the network must be considered. For each device there is an additional adder for cable delay and bridge delay in the Ethernet hardware.

Ethernet packets according to IEEE802.3 standard follow a defined frame structure and packet timing, which works for the internet at a worldwide level. Addressing scheme with 6 bytes for source address, 6 bytes of destination address, and additional network separation through virtual local area networks (VLANs) connect trillions of Internet protocol (IP) devices to a single network. In the case of controlling IODs with a single controller, the number of devices in the network can be scoped to 16 bit for a manufacturing site and 8 bit (256 devices) for a single control unit.

The scope of the SORTE protocol is to address single control unit networks, which spans no more than 8 bits of address. Another limit of standard Ethernet is the minimum frame length of 64 bytes. In many cases the size of IO data per control cycle is much smaller than 64 bytes. The minimum frame size of the SORTE protocol is 4 bytes.

Copyright © 2017, Texas Instruments Incorporated

Figure 3 shows the size-optimized frame structure. First optimization is before a packets starts with a shortened pre-amble. The pre-amble can be shortened to 1 byte as the physical layer no longer requires a long pre-amble for synchronization

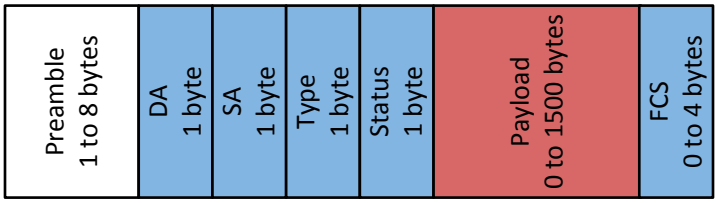


Figure 3. SORTE Frame Structure

DA: Destination address, 0 = master, 1..254 slaves, 255 = broadcast
SA: Source address, 0 = master, 1..254 slaves
Type: DISCOV_MC, PARAM_MC, SYNC_MC, LD_REQ_P2P, LD RESP_P2P,
OUT_MC, IN_UC, ALARM_UC, DIAG_UC, RECON_UC
Status: Bit 0: 0 = continuous, 1 = last
Bit 1: 0 = pending, 1 = done
Bit 2: 0 = no_error, 1 = error
Bit 3: 0 = no state change, 1 = state change in next cycle
Bit 4-7: master and slave states
M_DISC (0)
M_PARAM (1)
M_SYNC (2)
M_IO_EXCHANGE (3)
S_ALARM (4)
S_DIAG (5)
S_RECON (6)
S_RESET (7)

The gap between two packets is defined to 12 bytes by IEEE standard. Ethernet PHY can support less than 3 bytes of gap between two packets. The frame check sum (FCS) at the end of the packet is fixed to CRC32 (4 bytes) by the reference. SORTE allows for dynamic selection of error detection or even error correction codes. For the first implementation, a CRC8 polynomial is selected. Standard Ethernet expects the frame size within the range of 64 to 1518 bytes. SORTE protocol and Ethernet PHY support minimum-sized packets and have no upper bounds.

Table 1 shows the comparison of frame parameters between IEEE802.3 Ethernet, SORTE protocol, and industrial Ethernet physical layer device DP83822I.

Table 1. Ethernet Frame Parameter

PARAMETER	IEEE802.3 100Mb	SORTE	DP83822I PHY
InterPacketGap	960 ns	260 ns	<200 ns
Preamble + SFD	8 bytes	1 to 8 bytes	1 byte
FCS	CRC32 (4 bytes)	0 to 4 bytes	—
SA, DA	6 + 6 bytes	1 + 1 bytes	—
MinFrameSize	64 bytes	4 bytes	1 byte
MaxFrameSize	1518 bytes	Endless	Endless

The SORTE protocol enhancements compared to standard Ethernet are grouped into:

- Optimization of processing steps between application layer and Ethernet hardware
- Optimization of frame format, frame pre-amble, and interframe gap
- Optimization of frame processing for two-port Ethernet hardware
- Dynamic change of frame parameters and processing based on protocol states

Copyright © 2017, Texas Instruments Incorporated

Section 1.1 describes the key features of PRU-ICSS hardware, which supports all four optimizations previously listed.

1.1 Key System Specifications

Texas Instruments has a family of ARM®-based processor with integrated industrial communication peripheral also referred as PRU-ICSS. The entry level device is AM335x processor with one instance of PRU-ICSS. Processor families like AM437x and AM57xx have two instances of the flexible real-time Ethernet peripheral. This design describes the implementation of SORTE protocol based on AM3359 device on the industrial communication engine (ICE).

Complete documentation of the processor, ICE board, and PRU_ICSS can be found with following links:

- AM3359 Technical Documents Product Folder
- *AM335x Sitara™ Processors* Datasheet[3]
- *AM335x and AMIC110 Sitara Processors Technical Reference Manual*[4]
- AM3359 ICE Tool Folder
- PRU-ICSS Wiki Page

The PRU register file serves different operating modes, which are configured through CPCFG0[PRU0_GPI_MODE] register. Setting this bit field to three configures MII_RT mode. Other modes are direct input, 16-bit parallel capture, and 28-bit shift mode. In Ethernet mode the MII RX ports can go to either PRU or directly from RX L1 FIFO to one or both MII TX ports. This flexible configuration of port assignment, PRU assignment, and RX FIFO modes supports many real-time Ethernet functions.

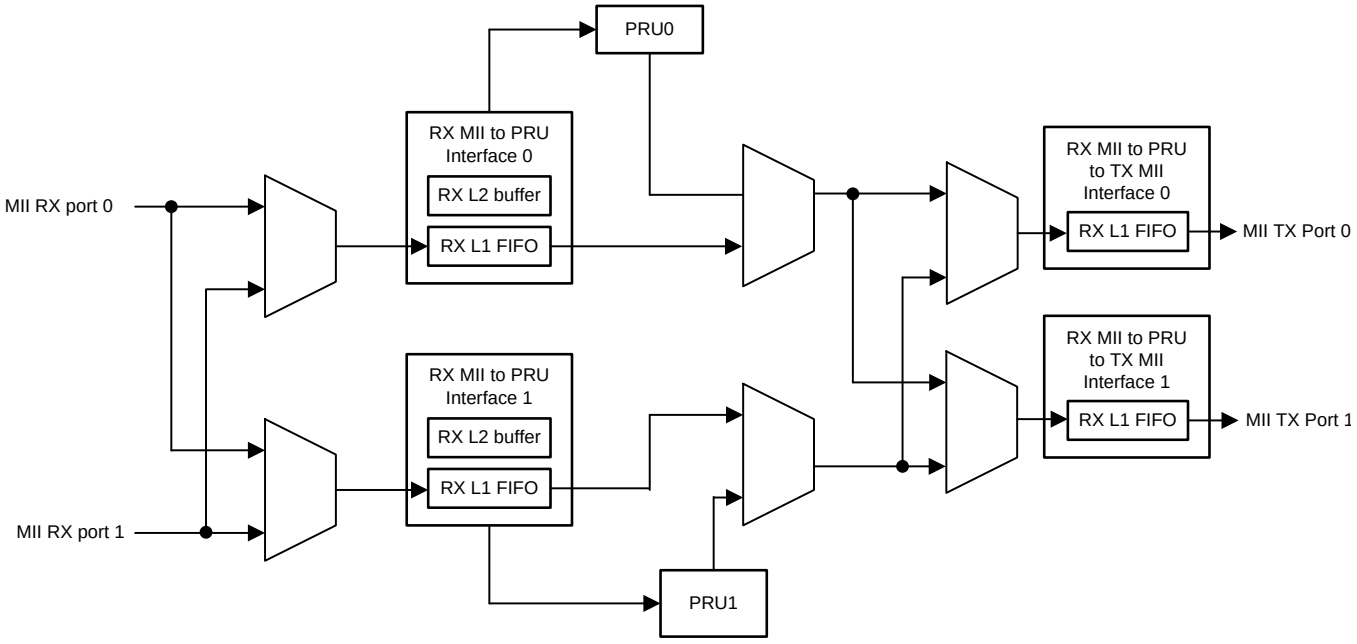


Figure 4. Ethernet Port Configuration

Figure 4 shows the multiplexing options in MII_RT mode. MII RX port 0 can feed its data into either interface, which is dedicated to PRU0 and PRU1. Only one RX port can be mapped to one interface at a time. In auto-forwarding mode the data from RX port goes to RX L1 FIFO and then directly to the configured MII TX port. In addition, PRU can receive Ethernet frame through RX L1 FIFO, which is typically used for *on-the-fly* processing, or RX L2 FIFO, which is used for cut-through processing.

For Ethernet MAC mode with one PRU same, RX and TX port numbers are mapped. The various Ethernet packet processing modes are described in Section 4.4.6 MII_RT of the *AM335x and AMIC110 Sitara Processors Technical Reference Manual*[4].

Copyright © 2017, Texas Instruments Incorporated

For real-time Ethernet the time to transfer packets and process packets are key performance parameter. With the programmable approach there is still the requirement of minimum latency and low jitter. The RX_L1 interface is used for on-the-fly processing of Ethernet packets. In this mode PRU reads the data and status through register R31 and writes data through register R30 and R31. The frame data does not leave the PRU core which enables minimum latency. To remove any PRU firmware jitter between MII receive time and MII transmit time, the forwarding time is controlled using a hardware timer. TX_START_DELAY bit field in TXCFG0 and TXCFG1 registers configures this forwarding time in the range of 0 to 5 µs in 40-ns increments.

Real-time Ethernet, which is switch-based, that is conditional forwarding of packets based on address information, requires a more complex data flow model, which includes memory transfer to and from data memory. There are two different modes for switch-based packet processing. For industrial field bus type of applications, the benefit of low latency supports a larger network (> 7 hops) in ring or line structure. Low latency can only be accomplished with cut-through processing of Ethernet packets. The switching decision time for cut-through is always after the address information at the beginning of the packet and, therefore, constant – typically < 3 µs. In store and forward mode, the switching time varies a lot as large packets cause a delay of 120 µs per switch, whereas small packets can be forwarded in less than 7 µs.

Figure 5 shows the data flow model with respective timing parameters to transfer Ethernet frames in PRU-ICSS system. PRU can use three different instructions to move data:

- Broadside move instructions XIN and XOUT (51.2 Gbit/s for 32 bytes)
- Burst data commands LBCO and SBCO (5.12 Gbit/s read and 5.7 Gbit/s write for 32 bytes)
- MOV instruction for RX_L1 and TX_L1 (1.6 Gbit/s rad and 1.07 Gbit/s write for 32 bytes)

Switch-based processing without 100% time division multiplexing of packets, that is engineered traffic, can run into port collision situations. For example, the local interface starts to transmit a large packet on port 1. Shortly after the start of transmission on port 1, another frame is received on port 0, which must get send out to port 1 as well. This is a collision case that requires the PRU to buffer incoming frames from port 0 for up to 1500 bytes and 24 small-sized packets. This buffer or collision queue no longer fits into PRU register bank or additional scratch pad register banks, which can be reached through broadside instructions XIN and XOUT with highest throughput. The local data memory inside PRU-ICSS can be used for collision packets. For more complex switching architectures (IEEE 802.1Q learning bridge) with up to eight queues per physical port, the memory model can be extended using L3 memory of the Sitara processor. The read latency of the first 32-bit word from L3 memory especially comes at a high cost. The 32-byte block read from L3 memory runs with 1.6 Gbit/s data rate.

Another timing parameter of switched Ethernet transfers is the gap between two packets also called inter-packet gap (IPG). To ensure minimum IPG time on a transmit port a hardware timer enforces the gap time based on previous packets. This hardware timer can be programmed to support different setting of IPG. In addition PRU shall support minimum gap time between packets and for this PRU requires real-time view of packet start and end time. Start of frame (SOF) and end of frame (EOF) for both receive and transmit port are important real-time triggers, which PRU can react on using event mapping in interrupt controller (INTC).

Copyright © 2017, Texas Instruments Incorporated

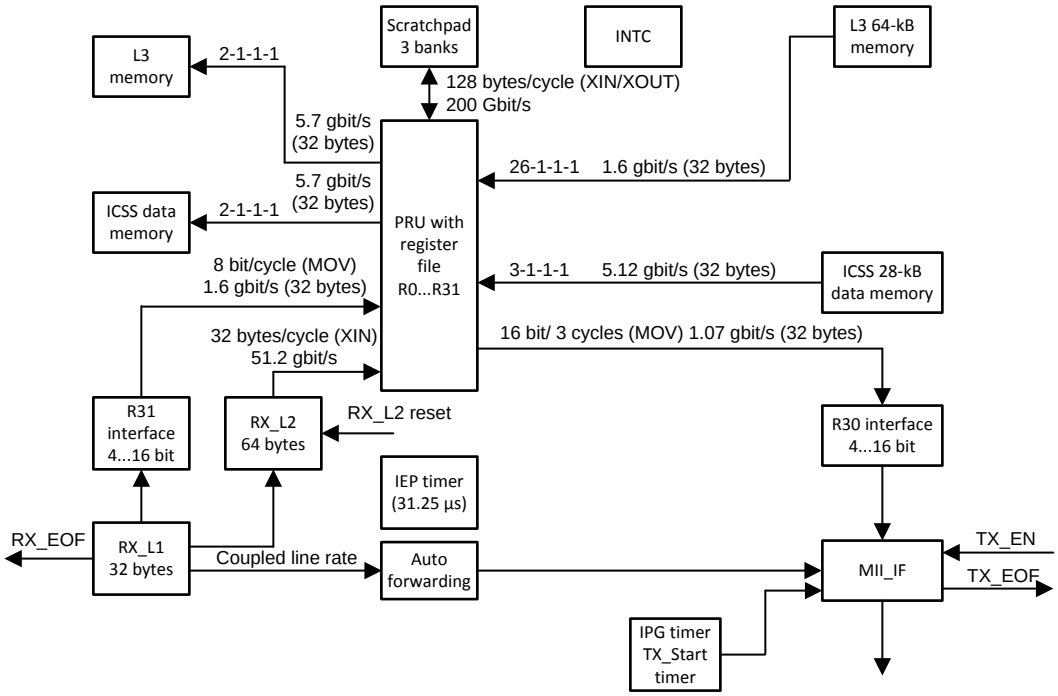


Figure 5. PRU Ethernet Switch Data Flow – One Slice

Real-time control of Ethernet packets requires the connection of a timer module with MII_RT interface triggers. Figure 5 shows the integration of a tunable counter with multiple capture and compare units, external hardware synchronization pins (SYNC, LATCH), low-latency IO (EDIO) pins, watchdog, and monitor of IPG time.

Copyright © 2017, Texas Instruments Incorporated

2 System Overview

2.1 Block Diagram

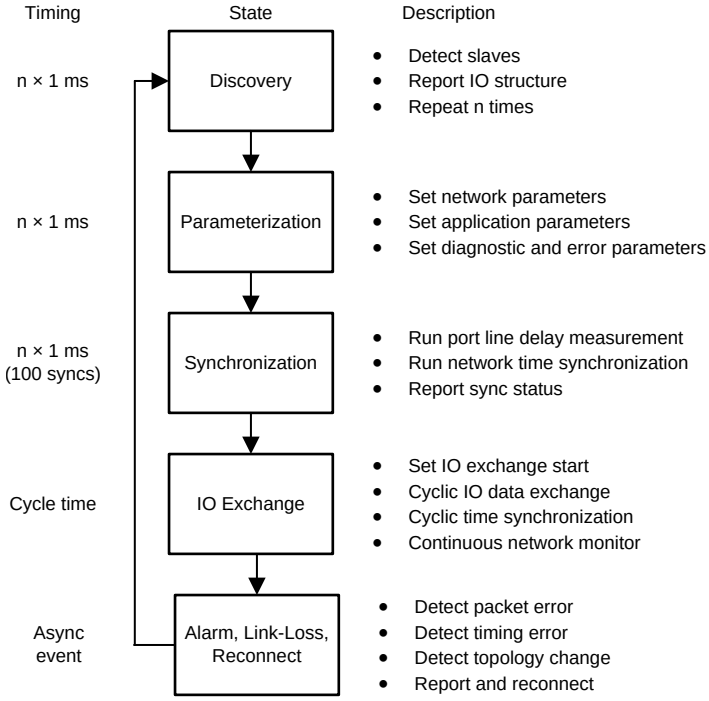


Figure 6. Block Diagram

Copyright © 2017, Texas Instruments Incorporated

2.2 Design Considerations

2.2.1 Industrial Ethernet Peripheral

The industrial Ethernet peripheral (IEP) hardware is clocked with either a dedicated iep_clk or same clock as PRU. Switching from iep_clk to ocp_clk is done by writing 1 to the IEPCLK.OCP_EN bit. IEP clock can also be stopped through CGR.IEP_CLK_EN bit. The asynchronous clock option of IEP timer comes at an overhead of additional wait states to read from IEP register space. PRU read commands with LBCO instructions take 12 PRU cycles versus three cycles reading other PRU-ICSS register and memory.

For real-time Ethernet, incoming and outgoing packets generate capture events to take time stamps with reference to MII_RT. These time stamps can be used to synchronize time in a network through line delay measurement and bridge delay measurement of Synchronization packets. The compare units generate events to time trigger outgoing packets, send interrupt to PRU through INTC, and send event to SYNC hardware for external signaling. EDIO function is typically not time triggered but PRU firmware triggered. The watchdog timer module can be used to make sure PRU is operational or ARM is operational. The event is than either mapped to PRU event register or ARM INTC.

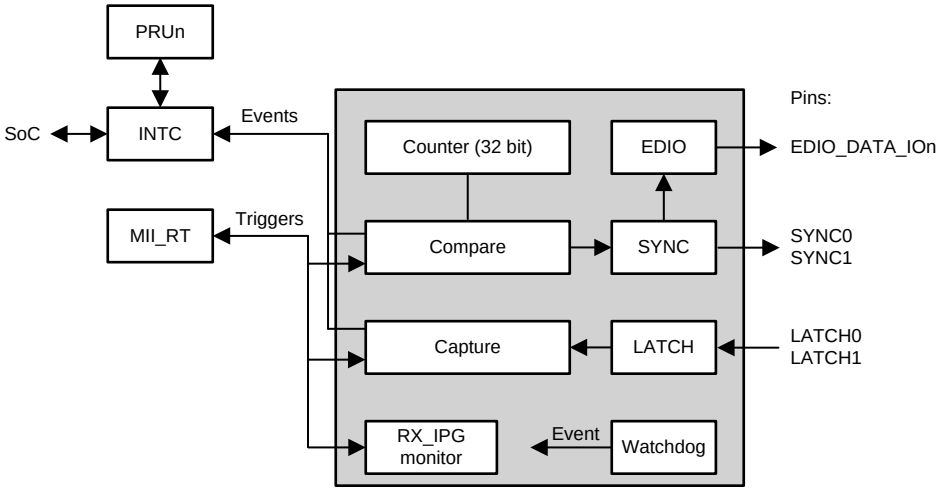


Figure 7. Industrial Ethernet Peripheral (IEP)

The counter in IEP block, also referred as IEP timer, can be rate compensated and offset adjusted. Default increment of the counter is with every IEP hardware clock. Figure 8 shows the IEP timer. The 32-bit COUNTER_REG is clocked with 200 MHz when cnt_enable bit is set in IEP_TMR_GLB_CFG register. A wrap-around of the 32-bit counter at 200 MHz occurs every 4.2 seconds also indicated by cnt_ovl status bit. Alternative mode for wrap-around is using compare 0 register through cmp0_cnt_rst_enable bit. For rate compensation the increment of the counter is varied from the default value of five. The alternative increment can be pre-programmed between 1 and 16 ns. The number of IEP updates with alternative increment is set by compen_cnt bit field in IEP_TMR_COMPEN register. The compare unit and capture unit use the COUNTER_REG for event generation.

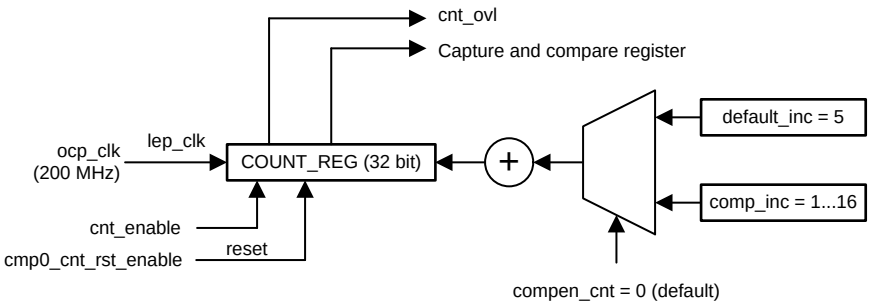


Figure 8. IEP Timer With Rate Compensation

Copyright © 2017, Texas Instruments Incorporated

Figure 9 shows the IEP compare unit with eight compare registers. Each compare register has separate a enable bit. Compare 0 register can be used to reset the IEP counter. For real-time Ethernet application, the compare 0 value can be set to the communication cycle time, for example 250-µs cycle time. In comparison to the IEP counter, the compare register is greater or equal. For example, if counter value is at 200 and IEP compare register is programmed to 150, there will be an immediate hit. A compare register can also be reprogrammed within the same cycle. For example if the counter is at 200, the compare register triggered before at 150 and is re-programmed to 250. Before a new compare hit event is generated, the new hit must be cleared from the previous hit.

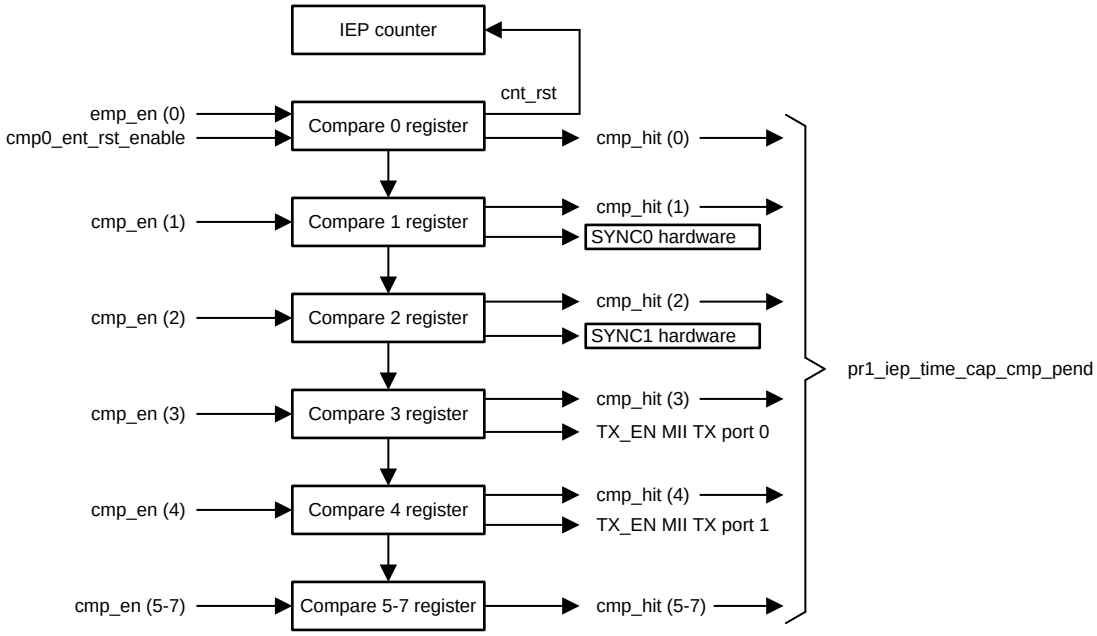


Figure 9. Compare Unit With Ethernet Time Trigger and SYNC Hardware

Compare registers 1 to 4 have additional hardware triggers for:

- External SYNC0 signal (compare 1)
- External SYNC1 signal (compare 2)
- Enable packet transfer on MII TX port 0 (compare 3)
- Enable packet transfer on MII TX port 1 (compare 4)

Compare 3 and compare 4 register can be used to schedule packets on MII TX ports 0 and 1 with 5-ns resolution. Additional jitter between internal MII_RT hardware and external Ethernet PHY can be removed by running both devices from the same clock source. Compare register 5 through 7 are general purpose with no direct hardware function. These registers generate a compare hit (cmp_hit) event, which can be read and cleared through CMP_STATUS_REG register. All events out of compare registers are combined to a single event into the INTC. The interrupts controller supports event generation to both PRU cores or other masters in the system like ARM CPU and enhanced direct memory access (EDMA). The PRU first must find out which compare or capture event occurred. The read latency to access IEP register takes 12 cycles. IEP timer event driven EDMA transfer of IEP register space into PRU-ICSS data memory can reduce this latency to three cycles. The chaining capability of EDMA would also allow to clear the flags after they are transferred into memory.

Copyright © 2017, Texas Instruments Incorporated

The IEP capture unit takes time stamps when packets are transferred over MII interface or external signal connected to LATCH 0 and LATCH1 pin have a positive or negative edge. Figure 10 shows the block diagram of the capture unit. The value captured through external events is taken from CONTER_REG at the time the event occurs. As external events may not be synchronized with internals iep_clk a double flop synchronization with the iep_clk is enabled by default configuration of cap_async_en[0..9] configuration bits. The capture mechanism supports two modes controlled with cap_1st_event_en[0..9] bits. When this control bit is set for a certain capture register, the captured time value will be kept even if there are more events happening. Only by reading the capture register will enable a new time capture. In continuous mode the time is captured with every event.

CAPTURE0-5 registers are dedicated to packet boundaries on receive and transmit over MII_RT interface. Receiving a packet on one port has two options for time stamps. One is start of frame (RX_SOF) indicated by RX_DV signal going active on MII_RT interface. The second option is after the MII_RT interface detected sync frame delimiter (RX_SFD) data pattern, which is typically the last data nibble of the pre-amble. For transmit the time stamp is taken when MII_RT interface starts sending a packet indicated through TX_EN signal.

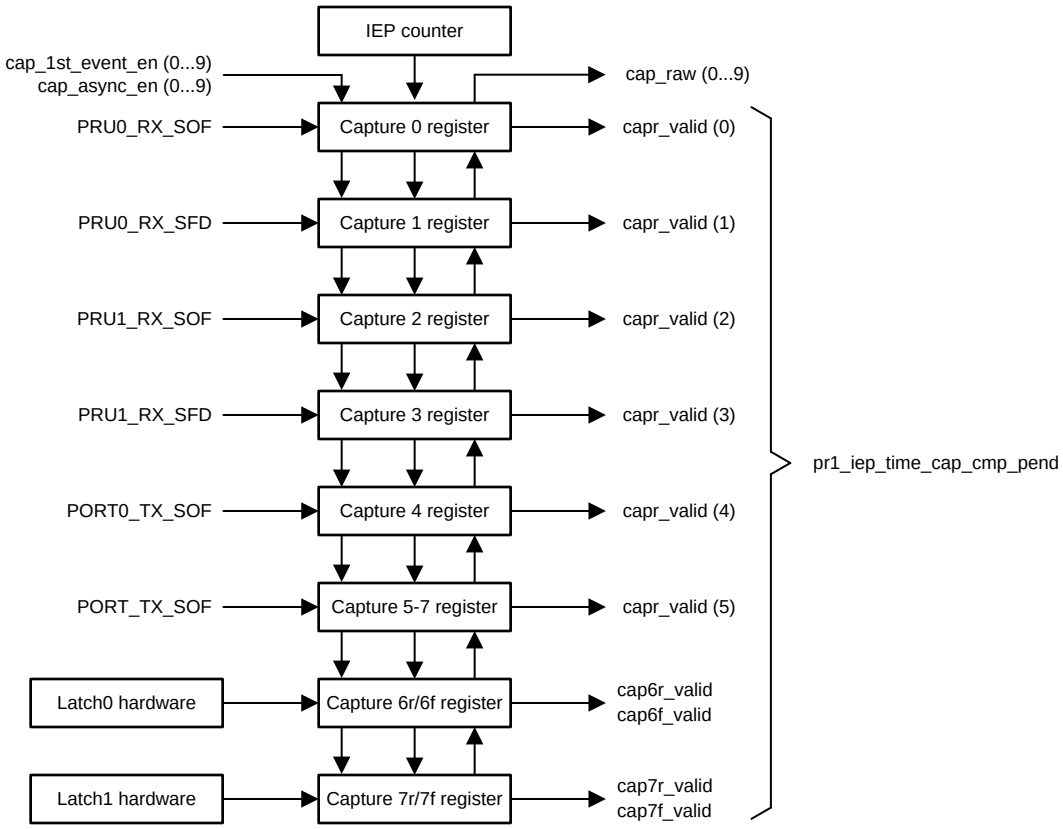


Figure 10. IEP Capture Unit for Ethernet Time Stamps

Detailed information about LATCH, SYNC, watchdog, and EDIO can be find in the *AM335x and AMIC110 Sitara Processors Technical Reference Manual*[4].

Copyright © 2017, Texas Instruments Incorporated

2.2.2 MII_RT Interface

MII_RT interface has additional features to support real-time Ethernet compared to standard MII interface and Ethernet media access layer (MAC). The full documentation of the interface can be found in Section 4.4.6 of the *AM335x and AMIC110 Sitara Processors Technical Reference Manual*[4]. This chapter summarizes the key features of MII_RT for real-time processing of Ethernet packets.

The interface has 16 memory mapped configuration registers.

Registers RXCFG0 and RXCFG1 configure the receive port functions:

- Enable or disable receive traffic
- Cut pre-amble or provide all data to PRU register
- Select MII RX data from port 0 or 1
- Enable RX_L2 buffer (broadside) instead of RX_L1 (16-bit register)
- Define the byte order into receive fifos
- Enable or disable auto-forward of received pre-amble

Register TXCFG0 and TXCFG1 configures the transmit port functions:

- Enable or disable TX port
- Automatically insert pre-amble by TX FIFO and not by PRU
- Enable transmit self-clear on TX_EOF event
- Byte order in R30 register interface to TX L1 FIFO
- Set MII TX port for PRU
- One PRU can send same packet to both ports
- Select transmit auto sequence which takes the data directly from RX L1 FIFO
- Set delay between received packet and transmit in the range of 80 ns up to 5 µs. Configure TX_EN and TX_DV setup time signal from TX_CLK.

Registers TXCRC0 and TXCRC1 allow PRU to read back CRC32 for diagnostics.

Registers TXIPG0 and TXIPG1 configure the minimum time for IPG

Registers PRS0 and PRS1 provide current state of MII CRS and COL signals. If not used for half duplex, these pins could be used as digital inputs with low-read latency compared to system GPIOs.

Registers RXFRMS0 and RXFRMS1 configure minimum and maximum frame size to trigger RX_MIN or MAX_FRM_ERR. These are 16-bit values, which support frame length outside IEEE 802.3 definitions.

Registers RXPCNT0 and RXPCNT1 configure minimum and maximum number of nibbles before SFD to trigger RX_MIN or MAX_PRE_COUNT_ERR.

Registers RXERR0 and RXERR1 indicate errors for frame size and pre-amble size.

PRU_ICSS can reprogram the MII_RT interface for every packet. Some of the configurations are only allowed with disabled receive or transmit mode or only during IPG time. In order to simplify MII_RT interface configuration, Table 2 summarizes the settings for different packet processing mode.

Copyright © 2017, Texas Instruments Incorporated

Table 2. MII_RT Configurations

	MII CONFIGURATION	CUT-THROUGH (CT)	AUTO-FORWARD (AF)	DELAY MASTER (DM)	DELAY SLAVE (DS)	HOST RECEIVE (HR)	HOST SEND (HS)	TIME TRIGGERED SEND (TSS)	PEER TO PEER (P2P)
RXCFG0	RX_ENABLE	1	1	1	1	1	—	—	1
	Reserved	0	0	0	0	0	—	—	0
	RX_CUT_PREAMBLE	1	1	1	1	1	—	—	1
	RX_MUX_SELECT	0	0	0	0	0	—	—	0
	RX_L2_ENABLE	1	1	1	1	1	—	—	1
	RX_BYTE_SWAP	0	0	0	0	0	—	—	0
	RX_AUT_FWD_PRE	1	1	0	1	0	—	—	1
	Bit 7...31 Reserved	0	0	0	0	0	—	—	0
RXCFG1	RX_ENABLE	1	1	1	1	1	—	—	1
	Reserved	0	0	0	0	0	—	—	0
	RX_CUT_PREAMBLE	1	1	1	1	1	—	—	1
	RX_MUX_SELECT	1	1	1	1	1	—	—	1
	RX_L2_ENABLE	1	1	1	1	1	—	—	1
	RX_BYTE_SWAP	0	0	0	0	0	—	—	0
	RX_AUT_FWD_PRE	1	1	0	1	0	—	—	1
	Bit 7...31 Reserved	0	0	0	0	0	—	—	0
TXCFG0	TX_ENABLE	1	1	1	1	—	1	1	1
	TX_AUTO_PREAMBLE	0	0	0	0	—	0	0	0
	TX_EN_MODE	0	0	0	0	—	0	1	0
	TX_BYTE_SWAP	0	0	0	0	—	0	0	0
	Reserved bit 7..4	0	0	0	0	—	0	0	0
	TX_MUX_SEL	1	1	0	0	—	0	0	0
	TX_AUTO_SEQUENCE	0	1	0	1	—	0	0	0
	Reserved	0	0	0	0	—	0	0	0
	TX_START_DELAY	0x40 (320 ns)	0x40 (320 ns)	0x40 (320 ns)	0x40 (320 ns)	—	Default	Default	0x40 (320 ns)
	TX_CLOCK_DELAY	0	0	0	0	—	0	0	0

Copyright © 2017, Texas Instruments Incorporated

Table 2. MII_RT Configurations (continued)

	MII CONFIGURATION	CUT-THROUGH (CT)	AUTO-FORWARD (AF)	DELAY MASTER (DM)	DELAY SLAVE (DS)	HOST RECEIVE (HR)	HOST SEND (HS)	TIME TRIGGERED SEND (TSS)	PEER TO PEER (P2P)
TXCFG1	TX_ENABLE	1	1	1	1	—	1	1	1
	TX_AUTO_PREAMBLE	0	0	0	0	—	0	0	0
	TX_EN_MODE	0	0	0	0	—	0	1	0
	TX_BYTE_SWAP	0	0	0	0	—	0	0	0
	Reserved bit 7..4	0	0	0	0	—	0	0	0
	TX_MUX_SEL	0	0	1	1	—	1	1	1
	TX_AUTO_SEQUENCE	0	1	0	1	—	0	0	0
	Reserved	0	0	0	0	—	0	0	0
	TX_START_DELAY	0x40 (320 ns)	0x40 (320 ns)	0x40 (320 ns)	0x40 (320 ns)	—	Default	Default	0x40 (320 ns)
	TX_CLOCK_DELAY	6	6	6	6	6	6	6	6
	RXCFG0	0x55	0x55	0x15	0x55	0x15	—	—	0x55
	RXCFG1	0x5d	0x5d	0x1d	0x5d	0x1d	—	—	0x5d
	TXCFG0	0x60400101	0x60400301	0x60400001	0x60400201	—	0x60400001	0x60400005	0x60400001
	TXCFG1	0x60400001	0x60400201	0x60400101	0x60400301	—	0x60400101	0x60400105	0x60400101

Copyright © 2017, Texas Instruments Incorporated

2.2.3 System Design

The SORTE protocol uses programmable approach to real-time communication for IO networks with a maximum of 254 devices. This protocol is fully documented and released in source code. The protocol is open to customers to learn, adapt, and enhance the protocol for their application requirements. SORTE is not bound to a given communication standard and breaks some of the limits of existing standards, such as minimum Ethernet frame size, addressing, and error detection. The primary use case of the protocol is to connect different end-equipment using Ethernet physical layer devices and 100-Mbit Ethernet cable.

In applications such as PLC backplane and modular drives, the protocol can be used to communicate inside the end-equipment using serial communication interfaces at different speeds. The protocol can be enhanced in future to support:

- Different data rates up to Gb
- LVDS physical layer with additional error correction code
- Non real-time channel for standard Ethernet traffic
- Chip-to-chip interface with external master port

The version 1.0 of the protocol describes the 100-Mbit Ethernet physical layer configuration supported by TI Sitara processor families – AM335x, AM437x, AM57xx, and 66AK2G. These processors have PRU-ICSS peripheral integrated. The PRU-ICSS firmware handles the networking layers of the protocol to maximize bandwidth of the host CPU to generate and consume IO data.

2.2.3.1 Packet Format

SORTE uses an optimized packet format to minimize processing delay and forwarding delay in the network. There are two different modes a received packet is forwarded. Packets which are not modified by a device are forwarded using auto-forward hardware in PRU-ICSS. In this mode the PRU can still read the content of the packet and extract input data or verify content of a packet. In the second mode, incoming packet is stored in receive FIFO; PRU modifies the packet and sends it to the transmit FIFO. The time when content of transmit FIFO is sent to the wire is determined by either PRU or auto-forward timer. The auto-forward timer can be programmed in a wide range from few ns up to 5 µs in steps of 5 ns.

Figure 3 shows the common format for all packet types. Addressing of source and destination is limited to one byte, that is 254 device network with one master. Value 255 in DA is reserved for broadcast message. Type field describes the packet type, which is unique for various states of the protocol. There are in total nine different packet types defined. Status field gives status information of current operation and state in which the master and device are currently operating in.

Payload field can be empty for some packet types or go for maximum length of 1500 bytes. The hardware supports packets larger than 1500 bytes and extending the maximum size to larger packets can be defined. There is the standard FCS from Ethernet, which uses 32-bit CRC code. For shorter packets this checksum can be configured to be CRC8 or CRC16. For minimum packet size and more granular error detection, the RX_ERR signal from Ethernet interface can be used instead of FCS. The IPG for 100-Mbit Ethernet is 960 ns. ICSS hardware has a programmable timer to enforce the IPG for back-to-back packets. For smaller networks, this parameter can be set to a shorter period like 640 ns.

2.2.3.2 Topology

The protocol supports one master and up to 254 devices. There are two modes supported. Mode one is line topology in which one master port sends and receives all the packets connected to the line. Mode two is a ring topology in which the master sends redundant packets on two ports to allow for one ring break without disturbing cyclic IO communication. Ring mode also allows shortening cycle time for the same amount of IO devices compared to line mode. During start-up phase the master explores the topology and mode of the devices. Operating mode can be redundant or normal if ring mode is discovered. With line topology only normal mode is supported. There are no multiple masters allowed in the network. The last device in a line topology has only one port connected. The other port of last device in a line remains open and cannot be used as an edge port for other Ethernet traffic.

Copyright © 2017, Texas Instruments Incorporated

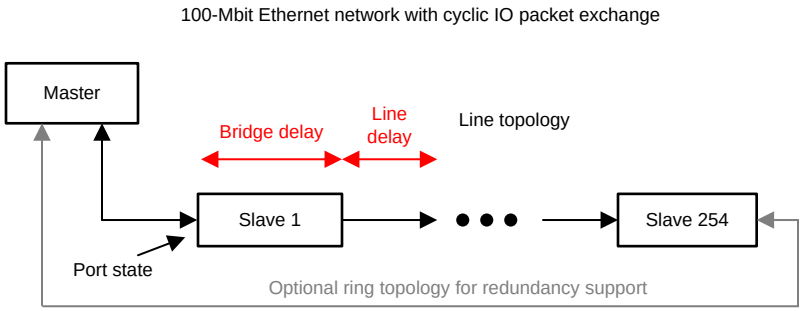


Figure 11. Topology

2.2.3.3 Start-up Phase

After power-up the master starts sending a Discovery packet to the network. Each device increments the node counter and inserts the number of input bytes and the number of output bytes in the payload field. There is a 16-bit wide field that contains the number of input data and another 16-bit field for the number of output data bytes per device. As long as there is another device connected on the other port of a device the packet is forwarded with STATUS byte set to *continuous*. Only last device in the line changes the STATUS to *last* and returns the packet back to the master.

The Discovery packet repeats n times at start-up until a stable network configuration is reached. At this time all input and output parameters of the network are known by the master. The length of a Discovery packet is 1030 bytes for a system, which supports 254 devices. Assuming a maximum line delay of 500 ns (100-m cable × 5 ns) + physical layer delay of 300 ns + bridge delay of 320 ns, there is a total delay of 1120 ns per device or 285.6 µs for a complete line in one direction. The total round trip time with 255 devices is 571.2 µs. With additional margin for the master to store and compare the results of the Discovery packet a cycle time of 1 ms is configured. Typical repeat count is in the range of 10 to 100.

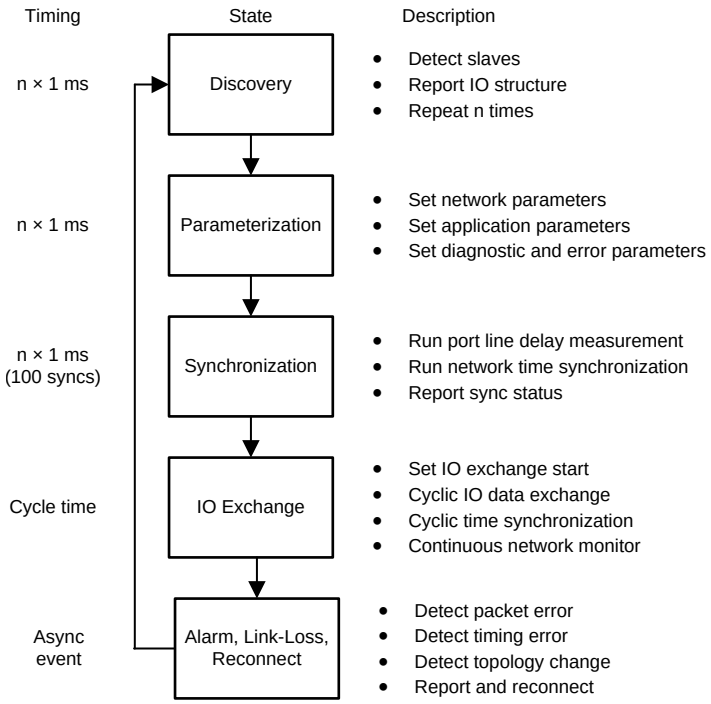


Figure 12. State Machine

Copyright © 2017, Texas Instruments Incorporated

2.2.3.4 Discovery

Discovery of the network is completed only once during start-up or in the case the topology changes. The network devices report a topology change in case a connection to one port is lost. In this case the start-up phase is initiated by the master, a link-loss message from the device to the master triggers the restart of the network. Figure 12 shows the state machine of the real-time Ethernet protocol. IO data exchange can be as fast as 3 ms after power up of the master. For a more reliable start-up sequence supporting networks with more than ten devices, 50 to 100 ms can be chosen for start-up timing. For example, ten repeats for Discovery, ten repeats for Parameterization, and 50 repeats for Synchronization states.

Table 3. Discovery Packet Format

FIELD	LENGTH	R/W	DEFAULT	DESCRIPTION
DA	1 byte	w (master) r (device)	0xff	Broadcast message to all devices
SA	1 byte	w (master)	0x00	Master is source of packet. Not required to check address by devices
Type	1 byte	w (master) r (device)	0x01	0x01 = DISCOV. Discovery packet type. Devices check the type on receives.
Status	1 byte	w (master) r/w (device)	0x00	0 = continuous, 1 = last On continuous the device inserts the IO parameter. On last state the device only forwards the packet without modification.
Payload	6 + 254 × 4 bytes	r/w (master) r/w (device)	0x00	Word 0: CNT – counter incremented +1 by each device also used as 32 bit pointer into data field Word 1: read position offset used in output data exchange, updated by each device Word 2: 16 bit number of input data for device 1 Word 3: 16 bit number of output data for device 1 … Word 509: number of inputs bytes for device 254 Word 510: number of output bytes for device 254 Each device inserts its own number of input and output date using byte 0 as a 32 bit pointer to the position.
FCS	4 bytes	w (master) r (device)	CRC32	Default Ethernet FCS

For advanced diagnostics the Ethernet PHYs may report line quality parameter and line break parameter. The line break parameter is typically the length of the broken wire, which is included in the LINK_LOSS packet. Line quality parameter is part of diagnostic data, which can be integrated into payload input data field or reported through separate diagnostic packet. Diagnostic packets can be sent by each device in the corresponding cycle number, which matches the device number. This ensures that there is only one diagnostic packet per cycle. Maximum latency to report diagnostic data back to the master is 254 IO cycles.

Copyright © 2017, Texas Instruments Incorporated

2.2.3.5 Parameterization

After Discovery phase the master distributes network parameters to the network. The Parameterization packet includes the following information:

- IO communication cycle time in ns (32 bit)
- Cycle time of time synchronization in ns (32 bit)
- Period of line delay measurements in ms (16 bit)
- Burst count of line delay measurement (8 bit)
- Gap time for line delay burst in ms (8 bit)
- Delay forward time in ns (16 bit)
- Application input time to relative to cycle start time in ns (32 bit)
- Application output time to relative to cycle start time in ns (32 bit)
- Topology mode: 0 = normal, 1 = ring, 2 ring redundant (8 bit)
- Diagnostic mode: 0 = part of payload, 1 = separate diagnostic packet per device
- Alarm mode: 0 = no immediate network reset on alarms, 1 = network reset on alarms
- CRC mode to choose from no, 8-, 16-, or 32- bit CRC code word

Parameterization packet is sent to all devices and returned by last device in network to the master. The status *LAST* is set by the device which returns the packet. Like with Discovery packet the Parameterization packet can be sent multiple times to ensure all devices are ready to take the parameters. Only if all devices are returning the packet with status LAST the parameters configured by the master are accepted. If a device does not accept the parameters, the device sets the status to *FAIL*. A typical repeat value for Parameterization packets is 1 to 10 with a cycle time of 1 ms.

Copyright © 2017, Texas Instruments Incorporated

2.2.3.6 Time Synchronization

There are two types of operation with respect to time synchronization of the network. First, each device runs line delay measurement to the neighbor ports and repeats it according to parameter provided by master. Figure 13 shows the principal of line delay measurement. LD_REQ packet does not contain any payload bytes. The device that sends out the LD_REQ packet saves the transmit time stamp of the packet as time T1. The device that receives the request packet takes a receive time stamp T3 and prepares a LD_RESP packet in which the payload contains the local delay between receiving LD_REQ and transmitting LD_RESP packet. Time to T4 is taken when the LD_RESP is sent to the wire. While the packet starts transmission, the local delay (T4-T3) is inserted into the payload field. Using PRU-ICSS with programmable delay between packet receive and packet transmit, the line delay measurement can be simplified. In this case the local delay (T4-T3) is already known by the delay master. The device, which receives LD_REQ, does a loopback on same port with a preprogrammed local delay. The LD_REQ and LD_RESP packets are identical. On the last LD_REQ packet the status bit indicates a state transition.

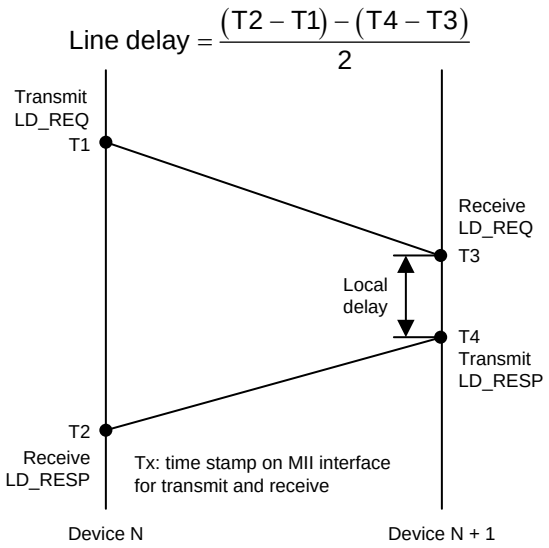


Figure 13. Line Delay Measurement

The sequence of line delay measurements is triggered from the master port. First the master sends LD_REQ to the first device, which triggers a line delay measurement on the other port of the device and so on.

The time synchronization function, as shown in Figure 14, starts after line delay is measured. The master sends out a SYNC packet with the master time (TM), which represents the time at MII interface when the packet leaves the master device. The first device receives the SYNC packet and updates the master time with line delay towards the master port and local bridge delay. With preprogrammed bridge delay of 320 ns the TM field can updated without reading time stamp T1 and T2. Subsequent devices perform the same update to TM field. Last device in the line does not update and forward the SYNC packet.

After the SYNC packet is processed, the device adjusts the local time to be equal with the master time. As the hardware of master and devices run from different oscillators, the local time adjustment can be split into drift compensation, —for example long term variation over temperature— and offset compensation coming from frequency offsets of the oscillators. Using a filtered adaptation of local timer will compensate short time jitter induced from interface clock or external influence like shock and vibration. The timer adjustment is performed right after SYNC packet is received by the forwarding PRU.

Copyright © 2017, Texas Instruments Incorporated

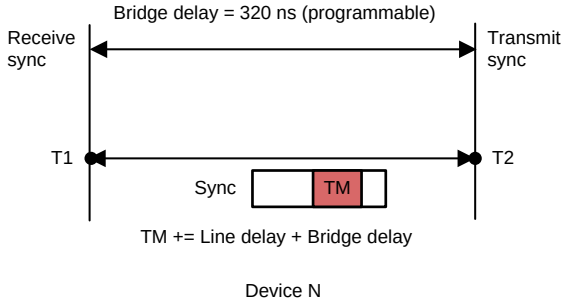


Figure 14. Time Synchronization

During IO Exchange state the master sends output packet at a known time. The devices use this time reference to adjust local time. In case there is a long time drift of line delay, the master time field can be appended to the output packet and updated by the devices in the same way as the sync packet.

2.2.3.7 IO Exchange

After start-up phase with Discovery, Parameterization, and time synchronization, the master and devices transition to cyclic IO Exchange state. The state change is indicated in last sync packet. Last sync packet is sent out 10 µs before state change to IO state.

In IO state, the device and master operate in full duplex mode. The first device starts sending its data towards the master after 100 ns. The master begins sending the multicast packet, which contains all output data after 500 ns. This out packet is also used for time synchronization at each device. The difference to sync packet is that cycle time is not 10 µs but cycle time of IO Exchange state. The minimum cycle time of IO Exchange is 4 µs when up to four devices are connected to the master.

The master always sends the multicast out packet at the beginning of a cycle. The devices have variable start time for sending the in packet towards the master. This time depends on the line delay, bridge delay and in packet size of other devices which are between the master and the current device.

Figure 15 shows the principle operation of the IO Exchange. The master sends single packet, which contains all output data of each device. The location inside the out packet for each device was provided during Discovery phase. Input data is sent by each device in a separate packet. To allow for maximum bandwidth in short cycle times, each device calculates the optimum time triggered send value (TTSx). An error condition occurs when transmission of local in packet is not finished before sending packet from other devices.

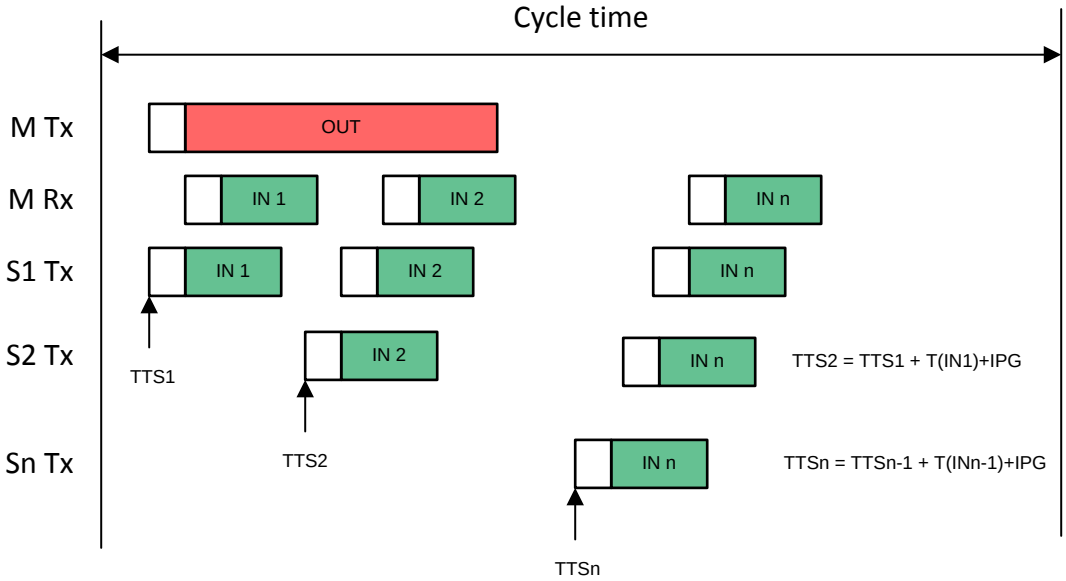


Figure 15. IO Cycle Principle

Copyright © 2017, Texas Instruments Incorporated

Figure 16 gives an overview of an implementation using 4 µs of cycle time. Each device has four input and three output bytes. The packet format and packet timing is highly optimized to allow such short cycle time over Ethernet physical layer. IPG time is set to 260 ns and pre-amble size of the packet is reduced to 2 bytes.

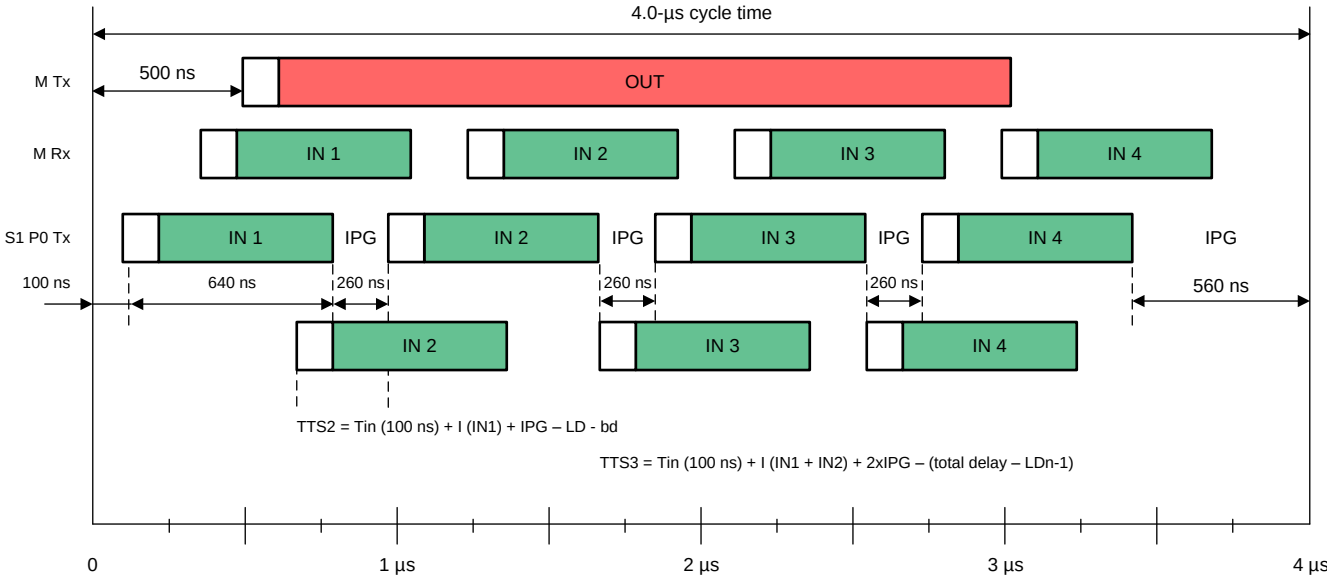


Figure 16. IO Distribution in 4-µs Cycle Time

The input packet format is identical for each device in this example. Source address (SA) is redundant information as the time when a packet received at the master provides the same information.

Table 4. Input Packet Format

BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	BYTE 8
0x55	0xd5	SA	STATUS	DATA 1	DATA 2	DATA 3	CRC8

The output packet format contains 16 data bytes, which are evenly distributed over four devices. Additional 2 bytes of tx time stamp (TX_TS1 and TX_TS2) are added to use for time synchronization during IO Exchange. CRC8 polynomial spans over bytes 3 through 20. Transmit time stamp (TX_TS) is 16 bit with MSB in byte 21 and LSB in byte 22.

Table 5. Output Packet Format

BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	...	BYTE 20	BYTE 21	BYTE 22	BYTE 23
0x55	0xd5	SA	STATUS	DATA 1	...	DATA 16	TX_TS1	TX_TS2	CRC8

Copyright © 2017, Texas Instruments Incorporated

2.2.3.8 Error and Diagnostic

There are several ways to detect and handle error conditions in the network. In order to exchange packets a physical link must be available. PRU-ICSS has two options to detect whether link is available. The MDIO interface can be used to read Ethernet physical layer status. A faster way to detect a link loss provides RX_ERR counter in MII_RT interface. In case there are 32 or more continuous RX_ERR events within a 10-µs sliding window, an RX_ERR32 event is sent to the INTC.

During protocol start-up a certain device may report an error through STATUS bit field. Bit 2 indicates an error condition from a slave towards the master, such as the Parameterization parameters are not accepted by certain devices. Frame errors are indicated by FCS at the end of the packet. SORTE protocol provides options of various FCS code words. During start-up phase with larger packets for Discovery and Parameterization, a 32-bit CRC code word is used and verified by MII_RT hardware. During IO Exchange phase with smaller packets, CRC8 polynomial is generated and verified by PRU firmware.

More error identification can be completed through timing analysis of received packets and IPG time. In case time synchronization is lost in the network, the timing of packets sent and received will start to drift. This drift can be measured by each network node using RX and TX time stamps of received and sent packets during IO Exchange phase.

There are three packet types defined to report back to the master. If a certain device in the network must enforce a reset condition and tell the master which device has issued, the request a RESET packet is defined.

Application on the devices may see error conditions, which are relevant for the application on the master device. ALARM packet provides the possibility to report alarm coder word with associated alarm data. Engineering of the network must reserve extra space for ALARM packets after IN packets are sent.

Application and network interface can provide additional diagnostic data to the master. The DIAG packet has additional time stamp field to report when diagnostic data was captured. As with ALARM packets the DIAG packets required extra bandwidth during IO Exchange.

Bandwidth allocation can be for single packet per cycle or all packets per cycle. In single cycle packet mode each device in the network maintains a cycle counter in IO Exchange state and takes the slot of the cycle counter modulo device number.

The master may not immediately retrain the network on a failing condition reported by the devices in the network. The interface described in Section 2.2.3.10 provides the option to not retrain, retrain after first error, or retrain after a number of errors.

The interface also defines a functional error test. When set, the PRU will force a predefined number of CRC errors in the packets. In addition a timing error can be enforced through this interface.

Copyright © 2017, Texas Instruments Incorporated

2.2.3.9 Packet Format Summary

Table 6. Packet Format Summary(1)

PACKET	DA	SA	TYPE	STATUS	PAYLOAD	SIZE [BYTES]
DISC	FF	0	DISK (1)	CONT DONE	count + reserved + (2 bytes) in_ptr + out_ptr (4) + 254 × 4	1026
PARAM	FF	0	PARAM (2)	CONT DONE	LD_PERIOD (8) , LD_BURST (8) SYNC_CNT (16), CYC_TIME (32) T_IN (32), T_OUT (32) SYNC_MODE (8), LD_MODE (8) ERROR_MODE (8), DIAG_LEVEL(8), IPG(16), BD(16), TTS_IN (N*32) RESERVED (16 bytes)	1062
LD_REQ	N-1	N	LD_REQ (3)	LD_PENDING LD_DONE	-	8
SYNC	FF	0	SYNC (5)	—	32-bit master time TM, 1 ns field	12
OUT	FF	0	OUT	CONT	Dynamic, 2 bytes of TX_TS before CRC	8+ variable
IN	0	N	IN	—	Dynamic	8+ variable
RESET	0	N	RES	RESET	N - device number which reports error	9
ALARM	0	N	ALARM	ALARM	ALARM_CODE (8), ALARM_DATA (32)	13
DIAG	0	N	DIAG	—	DIAG_CODE (8), DIAG_TS (32), DIAG_DATA (64)	21

(1) Abbreviation definitions available in Section 7.

Copyright © 2017, Texas Instruments Incorporated

2.2.3.10 Memory Map

PRU-ICSS has three internal memory blocks, which are low latency for PRU to read in 3-1-1-1 cycle bursts. First 32 bit takes three cycles and consecutive reads of single burst command (LBBO or LBCO) take one PRU cycle per 32 bits. The memory blocks are PRU0 data memory, PRU 1 data memory, and ICSS shared memory. PRU0 detects PRU0 data memory at offset 0 and PRU1 data memory at offset 0x2000. PRU1 detect PRU0 data memory at offset 0x2000 and PRU1 data memory at offset 0. Both PRUs see ICSS shared RAM at offset 0x10000.

The SORTE master protocol example runs on PRU0 and uses PRU0 data memory for control interface between ARM and PRU_ICSS, look-up tables, packet storage, and IO data. Table 7 lists all the entries of PRU0 data memory.

Table 7. Memory Map PRU0 Data Memory

PRU0 DATA MEMORY	OFFSET	LENGTH	DATA	COMMENT
SORTE_REG_INTERFACE	0x000	256	See Table 8	—
CRC8_TABLE	0x100	256	Look-up table for CRC8	—
DISCOVERY_RX	0x200	1033	Received Discovery packet	Used for engineering
OUT_DATA	0x700	Variable	Data field of OUT packet	16 bytes for 4-µs cycle time
IN_DATA	0x800	Variable	All in packets from IO Exchange	—

Table 8. Memory Map SORTE Register Interface

PRU0 DATA MEMORY	OFFSET	LENGTH	DATA	COMMENT
Control register	0x000	32 bits	Bit field set by host	—
Enable	—	—	Bit 0: 0 =disable 1 enable	Global start of protocol
Engineering	—	—	Bit 1: 0 = offline (host) 1 = online (PRU)	Engineering of cycle time and scheduling of packets Offline uses additional register settings.
Debug	—	—	Bit 2: 0 = off 1 = on	Debug trace buffer active, must be configured when enabled
Diagnostic	—	—	Bit 3: 0 = off 1 = on	Diagnostic trace buffer, must be configured when enabled
CRC error mode	—	—	Bit 4 and 5: 00: no retrain on error 01: retrain on single crc 10: retrain on n crc 11: not defined	Error handling when master receives CRC error during IO Exchange. Must specify number of CRC errors with setting *10*.
Timing error mode	—	—	Bit 6 and 7: 00: no retrain on error 01: retrain on single error 10: retrain on n errors 11: not defined	Error handling when master receives timing error of IN packets during exchange. Must specify number of errors with setting *10*.
Diagnostic mode	—	—	Bit 8: 00 : disabled 01: enable_single 10: enable_all 11: enable_shared	When enabled, the engineering leaves space in a cycle to send and receive a separate diagnostic packet. Mode 01 allows one diagnostic packet of specified size per cycle.

Copyright © 2017, Texas Instruments Incorporated

Table 8. Memory Map SORTE Register Interface (continued)

PRU0 DATA MEMORY	OFFSET	LENGTH	DATA	COMMENT
Functional error test	—	—	Bit 9 to 11: 000 – no error test, test competed 001 – force crc error with one packet 010 – force timing error with one packet 011 – force 3 crc errors 100 – force 3 timing errors 101 – force 10 crc errors 110 – force 10 timing errors 111 - reserved	Host sets the error tests. PRU clears the test when completed.
Status register	0x0004	32 bits	Bit field set by host	Status of SORTE protocol
Protocol state	—	—	Bit 0 to 2: 000: idle 001: discovery 010: parameterization 011: line delay 100: sync 101: IO exchange 110: error 111: ----	PRU indicates which state it is currently in.
Port 0 state	—	—	Bit 3: 0=no link 1 = link	PRU indicates whether port is active.
Port 1 state	—	—	Bit 4: 0 = no link 1 = link	PRU indicates whether port is active.
SYNC state	—	—	Bit 5: 0 = no sync 1 = sync	Device is not in sync state if the monitored jitter is out of range.
Communicatio n state	0x0008	32 bits	—	—
CRC error count	—	—	Bit 0 to 7: number of CRC errors	CRC error threshold for retrain
Timing violation count	—	—	Bit 8 to 15: number timing error count	Timing error threshold for retrain
Size of diagnostic Packets	—	—	Bit 16 to 23: number of bytes in extra diagnostic packet	Diagnostic packet sent
Reserved	—	—	Bit 23 to 31	—
Firmware revision 1	0x000C	32 bits	Bit 0 to 7: interim version Bit 8 to 15: minor version Bit 16 to 23: major version Bit 24: 0 = master, 1 = slave Bit 25 to 31: reserved	PRU firmware revision of SORTE protocol written by PRU
Interrupt control	0x0010	32 bits	Bit 0 to 7: timing interrupts Bit 8 to 15: packet interrupts Bit 16 to 23: state interrupts Bit 24 to 31: reserved	Host can select interrupt sources.
Interrupt status	0x0014	32 bits	Bit 0 to 7: timing interrupts Bit 8 to 15: packet interrupts Bit 16 to 23: state interrupts Bit 24 to 31: reserved	Interrupt status set by PRU.
Interrupt clear	0x0018	32 bits	Bit 0 to 7: timing interrupts Bit 8 to 15: packet interrupts Bit 16 to 23: state interrupts Bit 24 to 31: reserved	Host must clear interrupt to receive new one.
Reserved	0x001C	32 bits	—	—
PARAM DATA	0x0020	32 bytes	32 bytes of parameters sent by the master	—
Cycle_time	—	32 bits	Cycle time in ns	Minimum cycle time is 4 µs Maximum cycle time 100 ms

Copyright © 2017, Texas Instruments Incorporated

Table 8. Memory Map SORTE Register Interface (continued)

PRU0 DATA MEMORY	OFFSET	LENGTH	DATA	COMMENT
Sync_time	—	32 bits	Cycle time of sync packets in ns	If none is defined the master out packet is used as timing reference which has time stamp appended
Delay_period	—	32 bits	Period of line delay measurements in ms 0: no additional line delay 1-n: add line delay to cycle	Line delay is measured during start-up sequence. For period >0 a window of 2 µs is appended to IO Exchange to run line delay with period specified in this field.
Delay_burst	—	8 bits	Number of back to back delay measurements. Default = 16.	—
Delay_gap	—	8 bits	Delay gap time in ms	Gap time for line delay bursts in ms
Delay_forward_time	—	16 bits	Local forward time in ns	Delay forwards time of device, which responds after given time on same port
Inter_packet_gap	—	16 bits	Inter packet gap in ns	Minimum delay for back to back packets enforced by MII interface. Must respect Ethernet PHY min IPG.
T_in	—	16 bits	Offset time in ns	Input time relative to cycle start time
T_out	—	16 bits	Offset time in ns	Output time relative to cycle start time
Topology	—	8 bits	0: line 1: ring 2: ring redundant 3 to 255: reserved	Topology mode
Diagnistics	—	8 bits	0 = part of payload 1 = separate diagnostic packet per device	Diagnostic mode during IO Exchange state
Alarm	—	8 bits	0: no immediate reset on alarm 1: immediate reset on alarm	Master indicates to device how to behave in error conditions.
Crc_mode	—	8 bits	0: NO_CRC 1: CRC32 2: CRC16 3: CRC8 4: CASTAGNOLI 5 to 255: reserved	Crc code used during exchange state for IN and OUT packet
Reserved	0x0040	16 bytes	—	—
Statistic register port 0	0x0050	16 bytes	Port specific statistic	—
Cycle count	—	32 bits	IO cycle counter	Min 4 µs
CRC errors	—	32 bits	CRC error counter	—
Timing errors	—	32 bits	Timing error counter	—
Format errors	—	32 bits	Format error counter	Can be length or data structure For example, unknown command or address
Protocol retrain	0x0070	32 bits	Retrain counter	Number of retrains since power up
Reserved	0x0074	12 bytes	—	—
Network timing	0x0080	16 bytes	Packet timing for cyclic IO Exchange	Used to monitor any drift in network

Copyright © 2017, Texas Instruments Incorporated

2.3 Highlighted Products

2.3.1 AM3359

- Up to 1-GHz Sitara ARM Cortex-A8 32-bit RISC processor
- NEON™ SIMD coprocessor
- 32KB of L1 Instruction and 32KB of data cache with single-error detection (parity)
- 256KB of L2 cache with error correcting code (ECC)
- 176KB of on-chip boot ROM
- 64KB of dedicated RAM
- Emulation and debug - JTAG
- Interrupt controller (up to 128 interrupt requests)

PRU-ICSS:

- Supports protocols such as EtherCAT®, PROFIBUS®, PROFINET®, EtherNet/IP™, and more
- Two PRUs 32-bit load and store RISC processor capable of running at 200 MHz
- 8KB of instruction RAM with single-error detection (parity)
- 8KB of data RAM with single-error detection (parity)
- Single-cycle, 32-bit multiplier with 64-bit accumulator
- Enhanced GPIO module provides shift-in or shift-out support and parallel latch on external signal
- 12KB of shared RAM with single-error detection (parity)
- Three 120-byte register banks accessible by each PRU INTC for handling system input events
- Local interconnect bus for connecting internal and external masters to the resources inside the PRU-ICSS
- Peripherals inside the PRU-ICSS:
  - One universal asynchronous receiver and transmitter (UART) port with flow control pins that supports up to 12 Mbps
  - One enhanced capture (eCAP) module
  - Two MII Ethernet ports that support industrial Ethernet, such as EtherCAT
  - One management data input and output (MDIO) port

On-chip memory (shared L3 RAM):

- 64KB of general-purpose on-chip memory controller (OCMC) RAM
- Accessible to all masters

External memory interfaces (EMIF):

- mDDR(LPDDR), DDR2, DDR3, and DDR3L controller:
  - mDDR: 200-MHz clock (400-MHz data rate)
  - DDR2: 266-MHz clock (532-MHz data rate)
  - DDR3: 400-MHz clock (800-MHz data rate)
  - DDR3L: 400-MHz clock (800-MHz data rate)
  - 16-bit data bus
  - 1GB of total addressable space
  - Supports one x16 or two x8 memory device configurations
- General-purpose memory controller (GPMC)
- Flexible 8-bit and 16-bit asynchronous memory interface with up to seven chip selects (NAND, NOR, Muxed-NOR, or SRAM)
- Uses BCH code to support 4-, 8-, or 16-bit ECC
- Uses hamming code to support 1-bit ECC

See the *AM335x Sitara Processors*[3] datasheet for a complete list of features.

Copyright © 2017, Texas Instruments Incorporated

2.3.2 DP83822I

- IEEE 802.3u compliant: 100BASE-FX, 100BASETX and 10BASE-Te
- MII, RMII, and RGMII MAC Interfaces
- Low-power single supply options:
  - 1.8-V average (AVD) < 120 mW
  - 3.3-V AVD < 220 mW
- ±16-kV HBM ESD Protection
- ±8-kV IEC 61000-4-2 ESD Protection
- Start of frame detect for IEEE 1588 time stamp
- Fast link-down timing
- Auto-crossover in force modes
- Operating temperature: –40°C to 125°C
- IO voltages: 3.3 V, 2.5 V, and 1.8 V
- Power savings features:
  - Energy efficient Ethernet (EEE) IEEE 802.3az
  - Wake-on-LAN (WoL) support with magic packet detection
  - Programmable energy savings modes
- Cable diagnostics
- BIST
- Management data clock (MDC) and MDIO interface

See the *DP83822 Robust, Low Power 10/100 Mbps Ethernet Physical Layer Transceiver*[5] datasheet for a complete list of features.

Copyright © 2017, Texas Instruments Incorporated

2.3.3 TMDSICE3359 ICE EVM

Hardware specifications:

- AM3359 ARM Cortex-A8
- DDR3, NOR flash, and SPI flash
- Organize light-emitting diode (OLED) display
- TPS65910 power management 24-V power supply
- USB cable for JTAG interface and serial console

Software and tools:

- SYSBIOS real-time operating system (OS)
- Starterware base port
- TI's Code Composer Studio™ (CCS) integrated development environment (IDE)
- Application stack for industrial communication protocols
- Sample industrial applications

Connectivity:

- PROFIBUS interface
- CANOpen
- EtherNet/IP
- PROFINET
- Sercos III
- Digital IO
- SPI
- UART
- JTAG

Copyright © 2017, Texas Instruments Incorporated

3 Hardware, Software, Testing Requirements, and Test Results

3.1 Required Hardware and Software

3.1.1 Hardware

The following boards are required for this TI Design:

- Between two to five TMDSICE3359 ICE EVM boards

Connect an Ethernet cable from the master's RJ45 J2 port to the device's RJ45 J2 port. With multiple devices, connect an Ethernet cable from the first device's RJ45 J1 port to the second device's RJ45 J2 port. Follow the same procedure for additional devices. Power up all SORTE devices before powering up the SORTE master. The SORTE master will detect the connected SORTE devices and will proceed to IO Exchange state.

3.1.2 Software

Download the software package, and import the software projects into CCS. There are two CCS projects:

- SORTE_app - ARM application
- SORTE_MASTER - Master PRU firmware

Compile each project individually. Compile the SORTE_master project first, as it creates PRU C-header for the SORTE_app project. Download the SORTE_app.out file through JTAG to each TMDSICE3359 EVM. Alternatively, copy the file SORTE_app_ti.app to a bootable micro SD card and rename the file to app. Insert the micro SD card into the micro SD card interface of the TMDSICE3359 EVM, and press the reset button or power cycle the board. The application software is getting executed.

3.1.2.1 SORTE Master Firmware

The master protocol is a PRU firmware project inside CCS. Download and start of master protocol is completed by ARM driver. The SORTE master protocol is integrated into processor software development kit, which contains build instructions and all sources for ARM and PRU.

The application of digital IO runs inside PRU core and no ARM interaction is necessary once the protocol is started. This chapter describes only the PRU firmware project. Some of the files are shared between ARM and PRU to use the same definitions for the interface.

Directory structure:

- SORTE\firmware\include SORTE\firmware\include\icss
  - ICSS register definitions
- SORTE\firmware\include\macros
  - Macro definitions
- SORTE\firmware\include\protocol
  - Protocol-specific definitions
- SORTE\firmware\master
  - Master source file
- SORTE\firmware\master\header_gen
  - Header generation scripts
- SORTE\firmware\master\PRU0
  - Build directory with memory map files

The follow are brief descriptions of SORTE specific includes files in the project:

- sorte_macros.inc
  - Macro to change SORTE protocol state
- utils.inc
  - Macro to calculate CRC8

Copyright © 2017, Texas Instruments Incorporated

- sorte_host_interface.inc
  - SORTE host interface definitions (slave)
- sorte_master_configuration.inc
  - SORTE host interface definitions for master
- sorte_packet_defintion.inc
  - SORTE packet definitions
- sorte_pru_master_register.inc
  - PRU register definition for master protocol
- sorte_state_definition.inc
  - Protocol state and MII_RT configuration

Assembly source file for the master protocol is split into two files:

- main.asm
  - Includes setup-code, SORTE state machine, and simple IO application with four devices
- discovery.asm
  - Includes receive functions for Discovery packet

3.1.2.1.1 main.asm

Main.asm contains the SORTE protocol master for PRU0, which starts with include section of all required header files. Next are defines for which resources to use, which trace code to use, and which application to run on top of SORTE. For example:

- PRU0 .set 0
- PRU1 .set 1
- MASTER .set PRU0
- FULL_PREAMBLE .set 0
- ; trace settings
- trace_RXL2_length .set 1
- ; application setting
- application_io .set 1
- ; application_cnc .set 1

Before the source code starts, there are CCS specific assembler directives required for building .out file with assembly files. The directives include global name definitions to be used across source files or for far distance calls within same source file, like .global IDLE_WAIT_FOR_LINK_ACTIVE.

At main: label all PRU register are cleared and CRC8 look-up table is generated into PRU0 data memory. Then PRU configures SORTE registers and states. PRU R30 and R31 interface is set o MII_RT and frame error counters are adapted to SORTE packet size. Only frames less than 6 bytes cause including pre-amble are counter as length error. Ethernet port 1 is disabled and MII_RT register is set to host send and host receive mode using macros M_SET_MIIRT_HR and M_SET_MIIRT_HS. The IEP time is initialized and CMP0 wrap around is disabled.

The outer loop of SORTE master protocol starts at label IDLE_WAIT_FOR_LINK_ACTIVE. Only with active LINK indication from MDIO interface and interface enable from SORTE control register, the state changes to Discovery state.

At the beginning of Discovery state, PRU sets state to S_DICOV, MII_RT mode to host send, and resets the MII receive and transmit FIFOs. The Discovery packet sent by master is repeated ten times. Each packet has a length of 1026 bytes or equivalent transmission time at 100 Mbit/s of 82 µs. Discovery packet is transferred in the network in cut-through mode with latency of less than 1 µs per device. In a configuration with four devices, the Discovery packet takes about 10 µs between transmit and receive on master port, which means that there is significant overlap between transmit and receive of Discovery packet at the master.

Copyright © 2017, Texas Instruments Incorporated

The PRU must manage both transmit FIFO and receive FIFO concurrently with an arbitrary offset between the start time of both FIFO operation. Transmit FIFO supports only 16-bit interface, whereas receive supports 32-byte block transfer using broadside interface. The first 32 bytes of an outgoing Discovery packet is time triggered by PRU cycle counter with a period of 1 ms. PRU fills the header of the packet and first 32 bytes of payload. PRU also keeps the number of bytes written into the FIFO and the time from when PRU started to send the first byte. With these two numbers, the TX FIFO fill level can be calculated at any time as:

TX_FIFO_FILL_LEVEL = bytes inserted – bytes sent on the wire

At a minimum it takes PRU 240 ns to fill 32 bytes into TX FIFO. Sending 32 bytes on the wire takes 2.56 µs, which means the PRU is ten times faster filling the FIFO compared to 100-Mbit wire rate. The overhead to check TX FIFO level is 35 ns. These numbers are the basis for scheduling decision. Assuming PRU just missed the 32 byte threshold with a current FIFO level check, PRU now can go off doing other tasks for maximum 2.50 µs before the next check on FIFO fill level, which ensures that FIFO will not run empty because PRU missed to fill next 32 byte block in time. In the source code for Discovery packet, PRU calls function FN_DISC_RCV after TX FIFO is filled with minimum 32 bytes of data. FN_DISC_RCV is the receive function for Discovery packet returned from the network. The function resides in a different file and is described in Section 3.1.2.1.2.

On last block of Discovery packet, PRU indicates end of packet with CRC32 added by MII_RT interface. Now PRU waits for indication that Discovery packet is received before next iteration. On last iteration status information in header indicates state change to the network. After ten successful Discovery packets transmitted and received the state changes to Parameterization.

In S_PARAM state, all parameters are configured in PRU register space before transmission to the wire starts. Like with Discovery packet there is a 1-ms time tick to schedule Parameterization packet. After Parameterization packet is received, PRU checks whether all devices accepted the parameters. After ten iterations the program switches to line delay measurement state.

In state S_LINED the IEP timer is used to measure line delay between master and first device in the network. After line delay packet is sent and received PRU reads time stamps and calculates line delay by subtracting 320 ns of loopback delay of first device and divide by 2 to get delay time for one direction. The measurement repeats 16 times, and PRU keeps the minimum, maximum, and average value for line delay.

Line delay measurement is followed by network synchronization using SYNC packets. PRU schedules 1024 SYNC packets every 10 µs. After PRU starts sending SYNC packet, PRU inserts transmit time plus line delay into the outgoing packet. This time is used by next device to compare with own IEP timer and to adjust the timer accordingly. Now the network is configured and time synchronized. Final state change is to state IO_EXCHANGE.

The control network exchanges IO data with fixed cycle time. PRU on the master side transmits single output packet at a fixed offset of 500 ns. Scheduling of the cycle time and outgoing packet is done with IEP timer. As timing master there is no requirement to adjust IEP timer. Like with sync packet, the PRU appends time stamp plus line delay to the out packet, which is used by the devices in the network to keep exact synchronization during IO_EXCHANGE state. The output data is taken from shared memory.

After PRU has sent output packet there is enough time to work on application of IO data. The example in this project programs a running light for four devices with different update rates between 32 ms and 256 ms. The generation of new output data for four devices take 155 ns, which is only 4% of cycle budget with 4-µs cycle time.

After IO application PRU waits for all packets from the devices and stores them in PRU data memory. In addition PRU has setup SYNC0 hardware output to signal a new pulse every cycle time. This hardware sync signal can be used to measure network time synchronization accuracy. A second signal (SYNC1) may be used for local application synchronization.

In case there are no errors and alarms in current cycle then PRU repeats the IO_EXCHANGE cycle.

Copyright © 2017, Texas Instruments Incorporated

3.1.2.1.2 discovery.asm

The receive function for Discovery packet is implemented in a separate source file. This file includes all necessary header files and defines the global names for multi-source file projects. First check in the receive function to see whether all 32-byte blocks of packet are received to jump to last block processing. The function exits in case there are not enough bytes received in the FIFO. The RX L2 FIFO works in a ping-pong mode of 32-byte blocks. On the first block, PRU checks whether more than 32 bytes are received. In this case the block of data is stored into PRU0 data memory. The next time the function is called, the function checks the second bank and writes it into memory if complete.

On last block the RX L2 FIFO status is tested against RX_EOF. Another check is on packet length error. In case of error conditions are detected the statistic counter in host interface for CRC_ERR and FORMAT_ERR are incremented. PRU than jumps back to retrain location of the protrocol.

3.2 Testing and Results

3.2.1 Test Setup

The setup consists of one SORTE master connected to four SORTE devices, which are connected through Ethernet cable in line topology.

3.2.2 Test Results

Figure 17 shows the measurements on the first SORTE device of the Ethernet PHY with 4-µs cycle time. Port 0 is connected through the Ethernet cable to the master, and Port 1 is connected to the second device. The network is in IO Exchange state.

- Port 0 RX_DV shows the reception of the master frame.
- Port 0 TX_EN shows the forward transmission of master frame.
- Port 1 RX_DV shows the received device frames; the three frames are forwarded to the master through PORT 0 TX_EN.

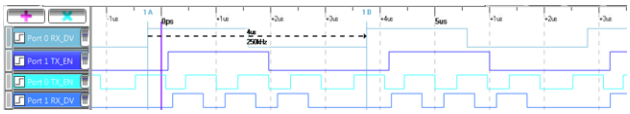


Figure 17. Industrial Ethernet With 4-µs Cycle Time

Copyright © 2017, Texas Instruments Incorporated

4 Design Files

4.1 Schematics

To download the schematics, see the design files at TIDEP-0085.

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDEP-0085.

4.3 PCB Layout Recommendations

4.3.1 Layout Prints

To download the layer plots, see the design files at TIDEP-0085.

4.4 Altium Project

To download the Altium project files, see the design files at TIDEP-0085.

4.5 Gerber Files

To download the Gerber files, see the design files at TIDEP-0085.

4.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDEP-0085.

5 Software Files

To download the software files, see the design files at TIDEP-0085.

6 Related Documentation

1. Texas Instruments, *Simple Open Real-Time Ethernet Device with PRU-ICSS Reference Design*, TIDEP-0086 TI Design (TIDUCK5)
2. Texas Instruments, *4-Axis CNC Router With 250-kHz Control Loop Reference Design*, TIDEP0061 TI Design (TIDUBY0)
3. Texas Instruments, *AM335x Sitara Processors*, AM3359, AM3358, AM3357, AM3356, AM3354, AM3352, and AM3351 Datasheet (SPRS717)
4. Texas Instruments, Processor SDK for AM33x Sitara Processor - TI-RTOS support
5. Texas Instruments, *AM335x and AMIC110 Sitara Processors Technical Reference Manual*, User's Guide (SPRUH73)
6. Texas Instruments, *DP83822 Robust, Low Power 10/100 Mbps Ethernet Physical Layer Transceiver*, DP83822HF, DP83822IF, DP83822H, DP83822I Datasheet (SNLS505)

6.1 Trademarks

Sitara, Code Composer Studio are trademarks of Texas Instruments, Inc..
NEON is a trademark of ARM Limited.
ARM is a registered trademark of ARM Limited.
EtherCAT is a registered trademark of Beckhoff Automation GmbH, Germany.
EtherNet/IP is a trademark of ODVA, Inc..
PROFIBUS, PROFINET are registered trademarks of PROFIBUS and PROFINET International (PI).
All other trademarks are the property of their respective owners.

Copyright © 2017, Texas Instruments Incorporated

7 Terminology

- CCS - Code Composer Studio
- CONT - Continue status
- DA - Destination address
- DISC - Discovery packet
- ICSS - Industrial communication system
- INTC - interrupt controller inside PRU-ICSS
- LD_REQ - Request line delay
- PARAM - Parameterization packet
- PRU - Programmable real-time unit
- SA - Source address
- SORTE - Simple open real-time Ethernet
- TM - Master time

8 About the Author

THOMAS LEYRER is a Systems Architect at Texas Instruments responsible for Industrial Communication solutions.

THOMAS MAUER is a System Engineer in the Factory Automation and Control Team at Texas Instruments Freising. He is responsible for developing reference design solutions for the industrial segment. Thomas brings his extensive experience in industrial communications like Industrial Ethernet and fieldbuses and industrial applications to this role. Thomas earned his degree in Electrical Engineering (Dipl. Ing. (FH)) at the University of Applied Sciences in Wiesbaden, Germany.

Copyright © 2017, Texas Instruments Incorporated

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2017, Texas Instruments Incorporated