LAN9512/LAN9512i

USB 2.0 Hub and 10/100 Ethernet Controller

PRODUCT FEATURES

Databook

Highlights

- Two downstream ports, one upstream port
  - Two integrated downstream USB 2.0 PHYs
  - One integrated upstream USB 2.0 PHY
- Integrated 10/100 Ethernet MAC with full-duplex support
- Integrated 10/100 Ethernet PHY with HP Auto-MDIX
- Implements Reduced Power Operating Modes
- Minimized BOM Cost
  - Single 25 MHz crystal (Eliminates cost of separate crystals for USB and Ethernet)
  - Built-in Power-On-Reset (POR) circuit (Eliminates requirement for external passive or active reset)

Target Applications

- Desktop PCs
- Notebook PCs
- Printers
- Game Consoles
- Embedded Systems
- Docking Stations

Key Features

- USB Hub
  - Fully compliant with Universal Serial Bus Specification Revision 2.0
  - HS (480 Mbps), FS (12 Mbps), and LS (1.5 Mbps) compatible
  - Two downstream ports, one upstream port
  - Port mapping and disable support
  - Port Swap: Programmable USB diff-pair pin location
  - PHY Boost: Programmable USB signal drive strength
  - Select presence of a permanently hardwired USB peripheral device on a port by port basis
  - Advanced power saving features
  - Downstream PHY goes into low power mode when port power to the port is disabled
  - Full Power Management with individual or ganged power control of each downstream port.
  - Integrated USB termination Pull-up/Pull-down resistors
  - Internal short circuit protection of USB differential signal pins
- High-Performance 10/100 Ethernet Controller
  - Fully compliant with IEEE802.3/802.3u
  - Integrated Ethernet MAC and PHY
  - 10BASE-T and 100BASE-TX support
  - Full- and half-duplex support with flow control
  - Preamble generation and removal
  - Automatic 32-bit CRC generation and checking
  - Automatic payload padding and pad removal
  - Loop-back modes
  - TCP/UDP checksum offload support
  - Flexible address filtering modes
    - One 48-bit perfect address
    - 64 hash-filtered multicast addresses
    - Pass all multicast
    - Promiscuous mode
    - Inverse filtering
    - Pass all incoming with status report
  - Wakeup packet support
  - Integrated Ethernet PHY
    - Auto-negotiation, HP Auto-MDIX
    - Automatic polarity detection and correction
    - Energy Detect
- Power and I/Os
  - Three PHY LEDs
  - Eight GPIOs
  - Supports bus-powered and self-powered operation
  - Internal 1.8v core supply regulator
  - External 3.3v I/O supply
- Miscellaneous features
  - Optional EEPROM
  - Optional 24MHz reference clock output for partner hub
  - IEEE 1149.1 (JTAG) Boundary Scan
- Software
  - Windows 2000/XP/Vista Driver
  - Linux Driver
  - Win CE Driver
  - MAC OS Driver
  - EEPROM Utility
- Packaging
  - 64-pin QFN, lead-free RoHS compliant
- Environmental
  - Commercial Temperature Range (0°C to +70°C)
  - Industrial Temperature Range (-40°C to +85°C)
  - ±8kV HBM without External Protection Devices
  - ±8kV contact mode (IEC61000-4-2)
  - ±15kV air-gap discharge mode (IEC61000-4-2)

Order Numbers:

LAN9512-JZX for 64-pin, QFN lead-free RoHS compliant package (0 to +70°C temp range)

LAN9512i-JZX for 64-pin, QFN lead-free RoHS compliant package (-40 to +85°C temp range)

This product meets the halogen maximum concentration values per IEC61249-2-21

For RoHS compliance and environmental information, please visit www.smsc.com/rohs

Circuit diagrams and other information relating to SMSC products are included as a means of illustrating typical applications. Consequently, complete information sufficient for construction purposes is not necessarily given. Although the information has been checked and is believed to be accurate, no responsibility is assumed for inaccuracies. SMSC reserves the right to make changes to specifications and product descriptions at any time without notice. Contact your local SMSC sales office to obtain the latest specifications before placing your product order. The provision of this information does not convey to the purchaser of the described semiconductor devices any licenses under any patent rights or other intellectual property rights of SMSC or others. All sales are expressly conditional on your agreement to the terms and conditions of the most recently dated version of SMSC's standard Terms of Sale Agreement dated before the date of your order (the "Terms of Sale Agreement"). The product may contain design defects or errors known as anomalies which may cause the product's functions to deviate from published specifications. Anomaly sheets are available upon request. SMSC products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an Officer of SMSC and further testing and/or modification will be fully at the risk of the customer. Copies of this document or other SMSC literature, as well as the Terms of Sale Agreement, may be obtained by visiting SMSC's website at http://www.smsc.com. SMSC is a registered trademark of Standard Microsystems Corporation ("SMSC"). Product names and company names are the trademarks of their respective holders.

SMSC DISCLAIMS AND EXCLUDES ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY AND ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND AGAINST INFRINGEMENT AND THE LIKE, AND ANY AND ALL WARRANTIES ARISING FROM ANY COURSE OF DEALING OR USAGE OF TRADE. IN NO EVENT SHALL SMSC BE LIABLE FOR ANY DIRECT, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES; OR FOR LOST DATA, PROFITS, SAVINGS OR REVENUES OF ANY KIND; REGARDLESS OF THE FORM OF ACTION, WHETHER BASED ON CONTRACT; TORT; NEGLIGENCE OF SMSC OR OTHERS; STRICT LIABILITY; BREACH OF WARRANTY; OR OTHERWISE; WHETHER OR NOT ANY REMEDY OF BUYER IS HELD TO HAVE FAILED OF ITS ESSENTIAL PURPOSE, AND WHETHER OR NOT SMSC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Table of Contents

List of Figures

List Of Tables

Chapter 1 Introduction

1.1 General Terms and Conventions

The following is list of the general terms used in this document:

BYTE	8-bits
CSR	Control and Status Registers
DWORD	32-bits
FIFO	First In First Out buffer
Frame	In the context of this document, a frame refers to transfers on the Ethernet interface.
FSM	Finite State Machine
GPIO	General Purpose I/O
HOST	External system (Includes processor, application software, etc.)
Level-Triggered Sticky Bit	This type of status bit is set whenever the condition that it represents is asserted. The bit remains set until the condition is no longer true, and the status bit is cleared by writing a zero.
LFSR	Linear Feedback Shift Register
MAC	Media Access Controller
MII	Media Independent Interface
N/A	Not Applicable
Packet	In the context of this document, a packet refers to transfers on the USB interface.
POR	Power on Reset.
RESERVED	Refers to a reserved bit field or address. Unless otherwise noted, reserved bits must always be zero for write operations. Unless otherwise noted, values are not guaranteed when reading reserved bits. Unless otherwise noted, do not read or write to reserved addresses.
SCSR	System Control and Status Registers
SMI	Serial Management Interface
TLI	Transaction Layer Interface
URX	USB Bulk Out Packet Receiver
UTX	USB Bulk In Packet Transmitter
WORD	16-bits
ZLP	Zero Length USB Packet

1.2 Block Diagram

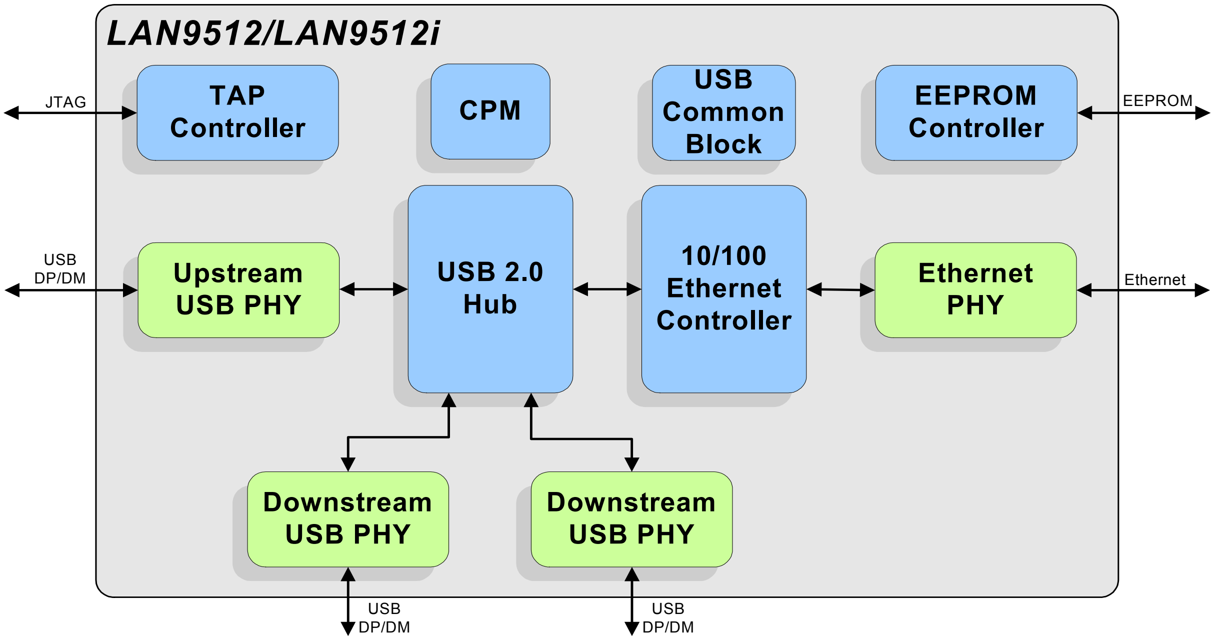


Figure 1.1 Internal Block Diagram

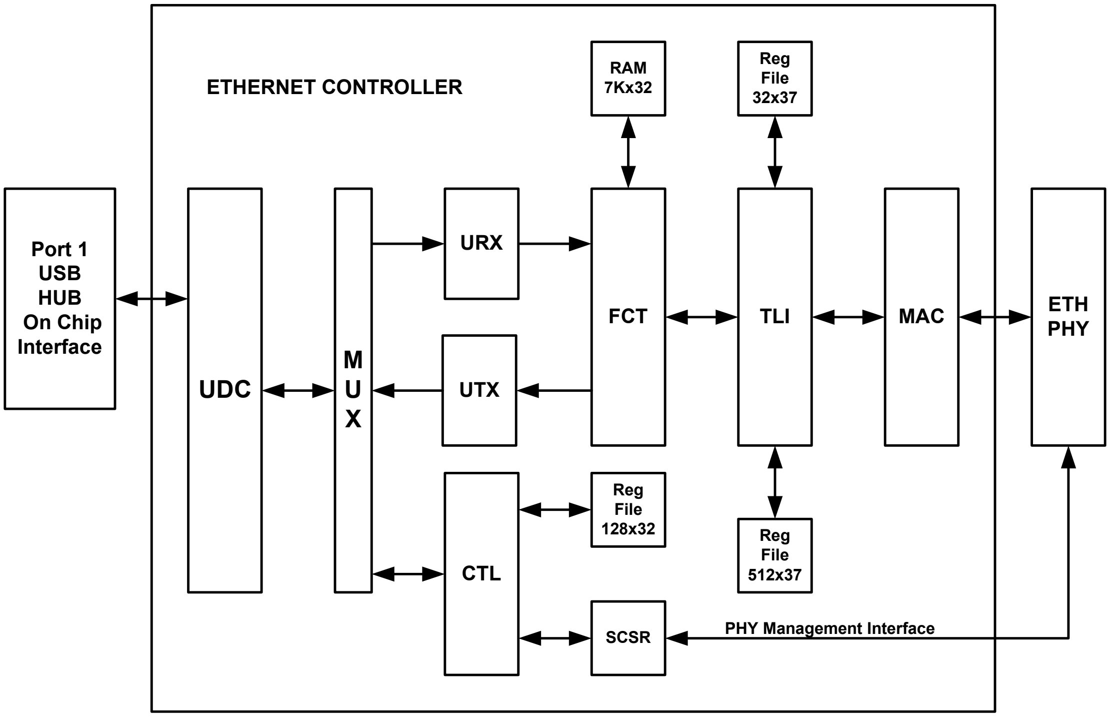


Figure 1.2 Ethernet Controller Block Diagram

1.2.1 Overview

The LAN9512/LAN9512i is a high performance Hi-Speed USB 2.0 hub with a 10/100 Ethernet controller. With applications ranging from embedded systems, desktop PCs, notebook PCs, printers, game consoles, and docking stations, the LAN9512/LAN9512i is targeted as a high performance, low cost USB/Ethernet and USB/USB connectivity solution.

The LAN9512/LAN9512i contains an integrated USB 2.0 hub, two integrated downstream USB 2.0 PHYs, an integrated upstream USB 2.0 PHY, a 10/100 Ethernet PHY, a 10/100 Ethernet Controller, a TAP controller, and a EEPROM controller. A block diagram of the LAN9512/LAN9512i is provided in Figure 1.1.

The LAN9512/LAN9512i hub provides over 30 programmable features, including:

PortMap (also referred to as port remap) which provides flexible port mapping and disabling sequences. The downstream ports of the LAN9512/LAN9512i Hub can be reordered or disabled in any sequence to support multiple platform designs' with minimum effort. For any port that is disabled, the LAN9512/LAN9512i automatically reorders the remaining ports to match the USB Host controller's port numbering scheme.

PortSwap which adds per-port programmability to USB differential-pair pin locations. PortSwap allows direct alignment of USB signals (USBDPn/USBDMn pins) to connectors avoiding uneven trace length or crossing of the USB differential signals on the PCB.

PHYBoost which enables four programmable levels of USB signal drive strength in USB port transceivers. PHYBoost attempts to restore USB signal integrity that has been compromised by system level variables such as poor PCB layout, long cables, etc.

1.2.2 USB Hub

The integrated USB Hub is fully compliant with the USB 2.0 Specification and will attach to a USB Host as a Full-Speed Hub or as a Full-/High-Speed Hub. The Hub supports Low-Speed, Full-Speed, and High-Speed (if operating as a High-Speed hub) downstream devices on all of the enabled downstream ports.

A dedicated Transaction Translator (TT) is available for each downstream facing port. This architecture ensures maximum USB throughput for each connected device when operating with mixed-speed peripherals.

The Hub works with an external USB power distributed switch device to control V_{BUS} switching to downstream ports, and to limit current and sense over-current conditions.

All required resistors on the USB ports are integrated into the Hub. This includes all series termination resistors on USBDPn and USBDMn pins and all required pull-down and pull-up resistors on USBDPn and USBDMn pins. The over-current sense inputs for the downstream facing ports have internal pull-up resistors.

Two external ports are available for general USB device connectivity.

1.2.3 Ethernet Controller

The Ethernet controller contains a USB Subsystem, a FIFO controller with a total of 30KB of internal packet buffering, a Transaction Layer Interface, and a 10/100 Ethernet Media Access Controller (MAC). The following sections provide detailed information for each of these components.

1.2.3.1 USB Subsystem

The USB portion of the Ethernet Controller consists of the USB 2.0 Hi-Speed compliant Device Controller (UDC), USB Bulk Out Packet Receiver (URX), USB Bulk In Packet Transmitter (UTX), Control Block (CTL), and System Control and Status Registers (SCSR). The Ethernet Controller is permanently connected to Port 1 of the USB Hub.

The USB Subsystem implements four USB endpoints: Control, Interrupt, Bulk-in, and Bulk-out. The Bulk-in and Bulk-out Endpoints allow for Ethernet reception and transmission respectively. Implementation of vendor-specific commands allows for efficient statistics gathering and access to the Ethernet controller’s system control and status registers.

The USB device controller (UDC) contains a USB low-level protocol interpreter that controls the USB bus protocol, packet generation/extraction, PID/Device ID parsing, and CRC coding/decoding with autonomous error handling. It is capable of operating either in USB 1.1 or 2.0 compliant modes. It has autonomous protocol handling functions like stall condition clearing on setup packets, suspend/resume/reset conditions, and remote wakeup. It also autonomously handles contingency operations for error conditions such as retry for CRC errors, Data toggle errors, and generation of NYET, STALL, ACK and NACK depending on the endpoint buffer status. The UDC implements four USB endpoints: Control, Interrupt, Bulk-In, and Bulk-Out.

The Control block (CTL) manages traffic to/from the control endpoint that is not handled by the UDC and constructs the packets used by the interrupt endpoint. The CTL is responsible for handling some USB standard commands and all vendor specific commands. The vendor specific commands allow for efficient statistics collection and access to the SCSR.

The URX and UTX implement the bulk-out and bulk-in pipes, respectively, which connect the USB Host and the UDC. They perform the following functions:

The URX passes USB Bulk-Out packets to the FIFO Controller (FCT). It tracks whether or not a USB packet is erroneous. It instructs the FCT to flush erroneous packets by rewinding its write pointer.

The UTX retrieves Ethernet frames from the FCT and constructs USB Bulk-In packets from them. If the handshake for a transmitted Bulk-In packet does not complete, the UTX is capable of retransmitting the packet. The UTX will not instruct the FCT to advance its read head pointer until the current USB packet has been successfully transmitted to the USB Host.

Both the URX and UTX are responsible for handling Ethernet frames encapsulated over USB by one of the following methods.

- Multiple Ethernet frames per USB Bulk packet
- Single Ethernet frame per USB Bulk packet

The UDC also implements the System Control and Status Register (SCSR) space used by the Host to obtain status and control overall system operation.

1.2.3.2 FIFO Controller (FCT)

The FIFO controller uses a 28 KB internal SRAM to buffer RX and TX traffic. 20 KB is allocated for received Ethernet-USB traffic (RX buffer), while 8 KB is allocated for USB-Ethernet traffic (TX buffer). Bulk-Out packets from the USB controller are directly stored into the TX buffer. The FCT is responsible for extracting Ethernet frames from the USB packet data and passing the frames to the MAC.Ethernet Frames are directly stored into the RX buffer and become the basis for bulk-in packets. The FCT passes the stored data to the UTX in blocks typically 512 or 64 bytes in size, depending on the current HS/FS USB operating speed.

1.2.3.3 Transaction Layer Interface (TLI)

The TLI interfaces the MAC with the FCT. It is a conduit between these two modules through which all transmitted and received data, along with status information, is passed. It has separate receive and transmit data paths. The TLI contains a 2KB transmit FIFO and a 128-byte receive FIFO. The transmit FIFO operates in store and forward mode and is capable of storing up to two Ethernet frames.

1.2.3.4 Ethernet MAC

The Ethernet MAC is fully IEEE 802.3 10BASE-T and 802.3u 100BASE-TX compliant. The transmit and receive data paths within the 10/100 Ethernet MAC are independent, allowing for the highest performance possible, particularly in full-duplex mode. The Ethernet MAC operates in store and forward mode, utilizing an independent 2KB buffer for transmitted frames, and a smaller 128 byte buffer for received frames. The Ethernet MAC data paths connect to the FIFO controller. The MAC also implements a Control and Status Register (CSR) space used by the Host to obtain status and control its operation.

Multiple power management features are provided, including various low power modes and "Magic Packet", "Wake On LAN", and "Link Status Change" wake events. These wake events can be programmed to initiate a USB remote wakeup.

1.2.3.5 Control and Status Registers (CSR)

The Ethernet Controller’s functions are controlled and monitored by the Host via the Control and Status Registers (CSR). This register space includes registers that control and monitor the USB controller, as well as elements of overall Ethernet Controller operation (System Control and Status Registers - SCSR), the MAC (MAC Control and Status Registers - MCSR), and the PHY (accessed indirectly through the MAC via the MII_ACCESS and MII_DATA registers). The CSR may be accessed via the USB Vendor Commands (REGISTER READ/REGISTER WRITE). Please refer to Section 3.6.6, "USB Vendor Commands," on page 81 for more information.

1.2.4 Ethernet PHY

The 10/100 Ethernet PHY integrates an IEEE 802.3 physical layer for twisted pair Ethernet applications. The PHY block includes support for auto-negotiation, full or half-duplex configuration, auto-polarity correction and Auto-MDIX. Minimal external components are required for the utilization of the integrated PHY.

1.2.5 EEPROM Controller (EPC)

The LAN9512/LAN9512i contains an EEPROM controller for connection to an external EEPROM. This allows for the automatic loading of static configuration data upon power-on reset, pin reset, or software reset. The EEPROM can be configured to load USB descriptors, USB device configuration, and the MAC address.

1.2.6 Peripherals

The LAN9512/LAN9512i also contains a TAP controller, and provides three PHY LED indicators, as well as eight general purpose I/O pins. All GPIOs can serve as remote wakeup events when LAN9512/LAN9512i is in a suspended state.

The integrated IEEE 1149.1 compliant TAP controller provides boundary scan and various test modes via JTAG.

1.2.7 Power Management

The LAN9512/LAN9512i features three variations of USB suspend: SUSPEND0, SUSPEND1, and SUSPEND2. These modes allow the application to select the ideal balance of remote wakeup functionality and power consumption for the Ethernet Controller.

- **SUSPEND0:** Supports GPIO, “Wake On LAN”, and “Magic Packet” remote wakeup events. This suspend state reduces power by stopping the clocks of the MAC and other internal Ethernet Controller modules.
- **SUSPEND1:** Supports GPIO and “Link Status Change” for remote wakeup events. This suspend state consumes less power than SUSPEND0.
- **SUSPEND2:** Supports only GPIO assertion for a remote wakeup event. This is the default suspend mode for the Ethernet Controller.

Please refer to Section 3.14, "Wake Events," on page 140 for more information on the USB suspend states and the wake events supported in each state.

1.2.8 Resets

LAN9512/LAN9512i supports the following system reset events:

- Power on Reset (POR)
- Hardware Reset Input Pin Reset (nRESET)
- Lite Reset (LRST)
- Software Reset (SRST)
- USB Reset
- VBUS Reset

LAN9512/LAN9512i supports the following module level reset events:

- Ethernet PHY Software Reset (PHY_RST)
- nTRST Pin Reset for Tap Controller

1.2.9 Test Features

Read/Write access to internal SRAMs is provided via the CSR’s. JTAG based USB BIST is available. Full internal scan and At Speed scan are supported.

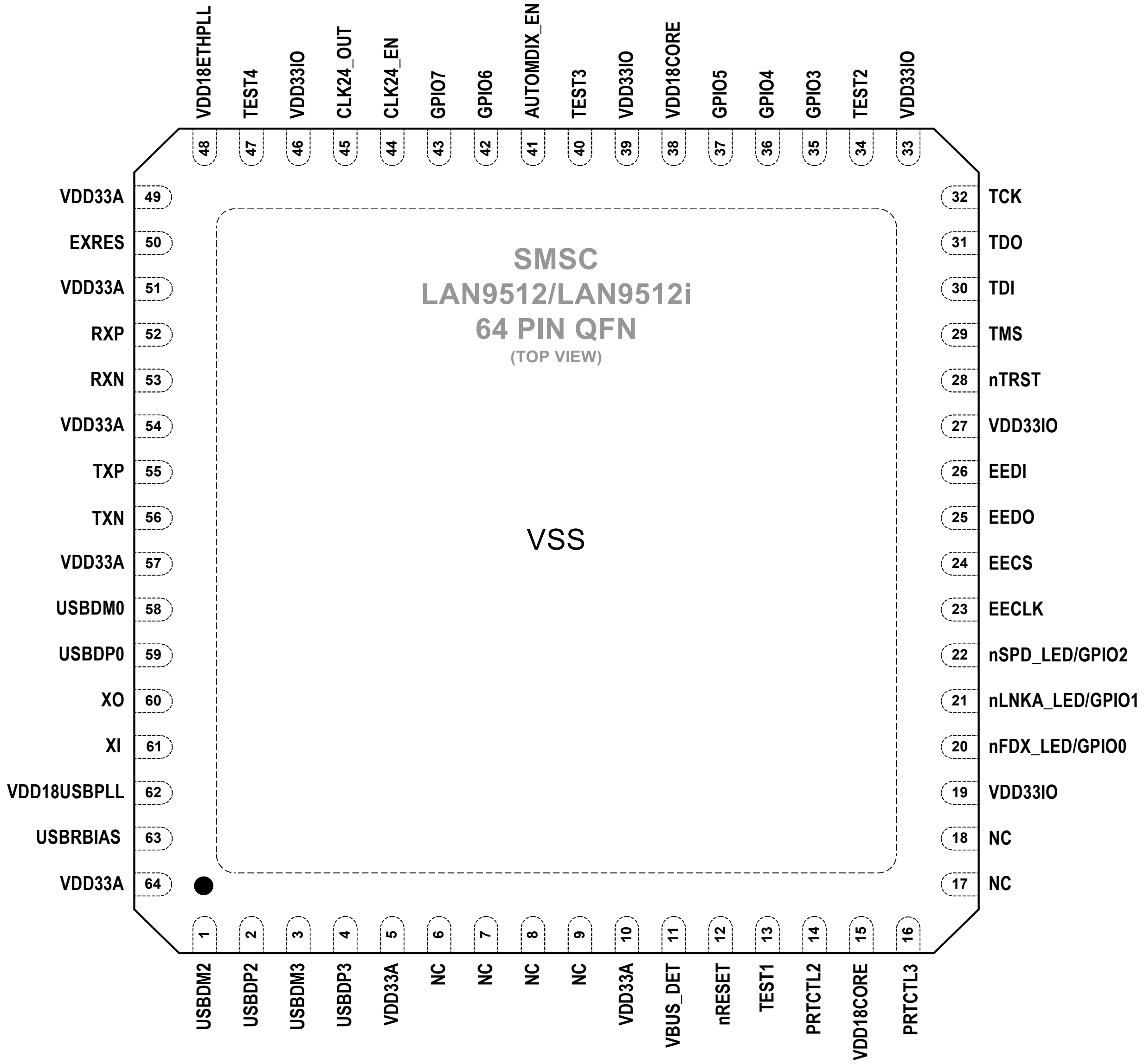
1.2.10 System Software

LAN9512/LAN9512i software drivers are available for the following operating systems:

- Windows XP
- Windows Vista
- Linux
- Win CE
- MAC OS

In addition, an EEPROM programming utility is available for configuring the external EEPROM.

Chapter 2 Pin Description and Configuration



NOTE: When HP Auto-MDIX is activated, the TXN/TXP pins can function as RXN/RXP and vice-versa

NOTE: Exposed pad (VSS) on bottom of package must be connected to ground

Figure 2.1 LAN9512/LAN9512i 64-QFN Pin Assignments (TOP VIEW)

Table 2.1 EEPROM Pins

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	EEPROM Data In	EEDI	IS (PD)	This pin is driven by the EEDO output of the external EEPROM.
1	EEPROM Data Out	EEDO	O8	This pin drives the EEDI input of the external EEPROM.
1	EEPROM Chip Select	EECS	O8	This pin drives the chip select output of the external EEPROM.
1	EEPROM Clock	EECLK	O8	This pin drives the EEPROM clock of the external EEPROM.

Table 2.2 JTAG Pins

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	JTAG Test Port Reset	nTRST	IS	This active low pin functions as the JTAG test port reset input. **Note:** This pin should be tied high if it is not used.
1	JTAG Test Mode Select	TMS	IS	This pin functions as the JTAG test mode select.
1	JTAG Test Data Input	TDI	IS	This pin functions as the JTAG data input.
1	JTAG Test Data Out	TDO	O12	This pin functions as the JTAG data output.
1	JTAG Test Clock	TCK	IS	This pin functions as the JTAG test clock.

Table 2.3 Miscellaneous Pins

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	System Reset	nRESET	IS	This active low pin allows external hardware to reset the device. **Note:** This pin should be tied high if it is not used.
1	Ethernet Full-Duplex Indicator LED	nFDX_LED	OD12 (PU)	This pin is driven low (LED on) when the Ethernet link is operating in full-duplex mode.
	General Purpose I/O 0	GPIO0	IS/O12/ OD12 (PU)	This General Purpose I/O pin is fully programmable as either a push-pull output, an open-drain output, or a Schmitt-triggered input.

Table 2.3 Miscellaneous Pins (continued)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Ethernet Link Activity Indicator LED	nLNKA_LED	OD12 (PU)	This pin is driven low (LED on) when a valid link is detected. This pin is pulsed high (LED off) for 80mS whenever transmit or receive activity is detected. This pin is then driven low again for a minimum of 80mS, after which time it will repeat the process if TX or RX activity is detected. Effectively, LED2 is activated solid for a link. When transmit or receive activity is sensed, LED2 will function as an activity indicator.
	General Purpose I/O 1	GPIO1	IS/O12/ OD12 (PU)	This General Purpose I/O pin is fully programmable as either a push-pull output, an open-drain output, or a Schmitt-triggered input.
1	Ethernet Speed Indicator LED	nSPD_LED	OD12 (PU)	This pin is driven low (LED on) when the Ethernet operating speed is 100Mbs, or during auto-negotiation. This pin is driven high during 10Mbs operation, or during line isolation.
	General Purpose I/O 2	GPIO2	IS/O12/ OD12 (PU)	This General Purpose I/O pin is fully programmable as either a push-pull output, an open-drain output, or a Schmitt-triggered input.
1	General Purpose I/O 3	GPIO3	IS/O8/ OD8 (PU)	This General Purpose I/O pin is fully programmable as either a push-pull output, an open-drain output, or a Schmitt-triggered input.
1	General Purpose I/O 4	GPIO4	IS/O8/ OD8 (PU)	This General Purpose I/O pin is fully programmable as either a push-pull output, an open-drain output, or a Schmitt-triggered input.
1	General Purpose I/O 5	GPIO5	IS/O8/ OD8 (PU)	This General Purpose I/O pin is fully programmable as either a push-pull output, an open-drain output, or a Schmitt-triggered input.
1	General Purpose I/O 6	GPIO6	IS/O8/ OD8 (PU)	This General Purpose I/O pin is fully programmable as either a push-pull output, an open-drain output, or a Schmitt-triggered input.
1	General Purpose I/O 7	GPIO7	IS/O8/ OD8 (PU)	This General Purpose I/O pin is fully programmable as either a push-pull output, an open-drain output, or a Schmitt-triggered input.
1	Detect Upstream VBUS Power	VBUS_DET	IS_5V	This pin detects the state of the upstream bus power. The Hub monitors VBUS_DET to determine when to assert the USBDP0 pin's internal pull-up resistor (signaling a connect event). For bus powered hubs, this pin must be tied to VDD33IO. For self powered hubs where the device is permanently attached to a host, VBUS_DET should be pulled to VDD33IO. For other self powered applications, refer to the device reference schematic for additional connection information.
1	Auto-MDIX Enable	AUTOMDIX_EN	IS	Determines the default Auto-MDIX setting. 0 = Auto-MDIX is disabled. 1 = Auto-MDIX is enabled.

Table 2.3 Miscellaneous Pins (continued)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Test 1	TEST1	-	Used for factory testing, this pin must always be left unconnected.
1	Test 2	TEST2	-	Used for factory testing, this pin must always be connected to VSS for proper operation.
1	Test 3	TEST3	-	Used for factory testing, this pin must always be connected to VDD33IO for proper operation.
1	24 MHz Clock Enable	CLK24_EN	IS	This pin enables the generation of the 24 MHz clock on the CLK_24_OUT pin.
1	24 MHz Clock	CLK24_OUT	08	This pin outputs a 24 MHz clock that can be used a reference clock for a partner hub.
1	Test 4	TEST4	-	Used for factory testing, this pin must always be left unconnected.

Table 2.4 USB Pins

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Upstream USB DMINUS 0	USBDM0	AIO	Upstream USB DMINUS signal.
1	Upstream USB DPLUS 0	USBDP0	AIO	Upstream USB DPLUS signal.
1	Downstream USB DMINUS 2	USBDM2	AIO	Downstream USB peripheral 2 DMINUS signal.
1	Downstream USB DPLUS 2	USBDP2	AIO	Downstream USB peripheral 2 DPLUS signal.
1	Downstream USB DMINUS 3	USBDM3	AIO	Downstream USB peripheral 3 DMINUS signal.
1	Downstream USB DPLUS 3	USBDP3	AIO	Downstream USB peripheral 3 DPLUS signal.
1	USB Port Power Control 2	PRTCTL2	IS/OD12 (PU)	When used as an output, this pin enables power to downstream USB peripheral 2. When used as an input, this pin is used to sample the output signal from an external current monitor for downstream USB peripheral 2. An overcurrent condition is indicated when the signal is low. Refer to Section 2.2 for additional information.

Table 2.4 USB Pins (continued)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	USB Port Power Control 3	PRTCTL3	IS/OD12 (PU)	When used as an output, this pin enables power to downstream USB peripheral 3. When used as an input, this pin is used to sample the output signal from an external current monitor for downstream USB peripheral 3. An overcurrent condition is indicated when the signal is low. Refer to Section 2.2 for additional information.
1	External USB Bias Resistor	USBRBIAS	AI	Used for setting HS transmit current level and on-chip termination impedance. Connect to an external 12K 1.0% resistor to ground.
1	USB PLL +1.8V Power Supply	VDD18USBPLL	P	Refer to the LAN9512/LAN9512i reference schematics for additional connection information.
1	Crystal Input	XI	ICLK	External 25 MHz crystal input. **Note:** This pin can also be driven by a single-ended clock oscillator. When this method is used, XO should be left unconnected
1	Crystal Output	XO	OCLK	External 25 MHz crystal output.

Table 2.5 Ethernet PHY Pins

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Ethernet TX Data Out Negative	TXN	AIO	Negative output of the Ethernet transmitter. The transmit data outputs may be swapped internally with receive data inputs when Auto-MDIX is enabled.
1	Ethernet TX Data Out Positive	TXP	AIO	Positive output of the Ethernet transmitter. The transmit data outputs may be swapped internally with receive data inputs when Auto-MDIX is enabled.
1	Ethernet RX Data In Negative	RXN	AIO	Negative input of the Ethernet receiver. The receive data inputs may be swapped internally with transmit data outputs when Auto-MDIX is enabled.
1	Ethernet RX Data In Positive	RXP	AIO	Positive input of the Ethernet receiver. The receive data inputs may be swapped internally with transmit data outputs when Auto-MDIX is enabled.
7	+3.3V Analog Power Supply	VDD33A	P	Refer to the LAN9512/LAN9512i reference schematics for connection information.

Table 2.5 Ethernet PHY Pins (continued)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	External PHY Bias Resistor	EXRES	AI	Used for the internal bias circuits. Connect to an external 12.4K 1.0% resistor to ground.
1	Ethernet PLL +1.8V Power Supply	VDD18ETHPLL	P	Refer to the LAN9512/LAN9512i reference schematics for additional connection information.

Table 2.6 I/O Power Pins, Core Power Pins, and Ground Pad

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
5	+3.3V I/O Power	VDD33IO	P	+3.3V Power Supply for I/O Pins. Refer to the LAN9512/LAN9512i reference schematics for connection information.
2	Digital Core +1.8V Power Supply Output	VDD18CORE	P	+1.8 V power from the internal core voltage regulator. All VDD18CORE pins must be tied together for proper operation. Refer to the LAN9512/LAN9512i reference schematics for connection information.
1 Note 2.1	Ground	VSS	P	Ground

Note 2.1 Exposed pad on package bottom (Figure 2.1).

Table 2.7 No-Connect Pins

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
6	No Connect	NC	-	These pins must be left floating for normal device operation

Table 2.8 64-QFN Package Pin Assignments

PIN NUM	PIN NAME	PIN NUM	PIN NAME	PIN NUM	PIN NAME	PIN NUM	PIN NAME
1	USBDM2	17	NC	33	VDD33IO	49	VDD33A
2	USBDP2	18	NC	34	TEST2	50	EXRES
3	USBDM3	19	VDD33IO	35	GPIO3	51	VDD33A
4	USBDP3	20	nFDX_LED/ GPIO0	36	GPIO4	52	RXP
5	VDD33A	21	nLNKA_LED/ GPIO1	37	GPIO5	53	RXN
6	NC	22	nSPD_LED/ GPIO2	38	VDD18CORE	54	VDD33A
7	NC	23	EECLK	39	VDD33IO	55	TXP
8	NC	24	EECS	40	TEST3	56	TXN
9	NC	25	EEDO	41	AUTOMDIX_EN	57	VDD33A
10	VDD33A	26	EEDI	42	GPIO6	58	USBDM0
11	VBUS_DET	27	VDD33IO	43	GPIO7	59	USBDP0
12	nRESET	28	nTRST	44	CLK24_EN	60	XO
13	TEST1	29	TMS	45	CLK24_OUT	61	XI
14	PRTCTL2	30	TDI	46	VDD33IO	62	VDD18USBPLL
15	VDD18CORE	31	TDO	47	TEST4	63	USBRBIAS
16	PRTCTL3	32	TCK	48	VDD18ETHPLL	64	VDD33A
EXPOSED PAD MUST BE CONNECTED TO VSS							

2.1 Power Connections

Figure 2.2 illustrates the power connections for LAN9512/LAN9512i.

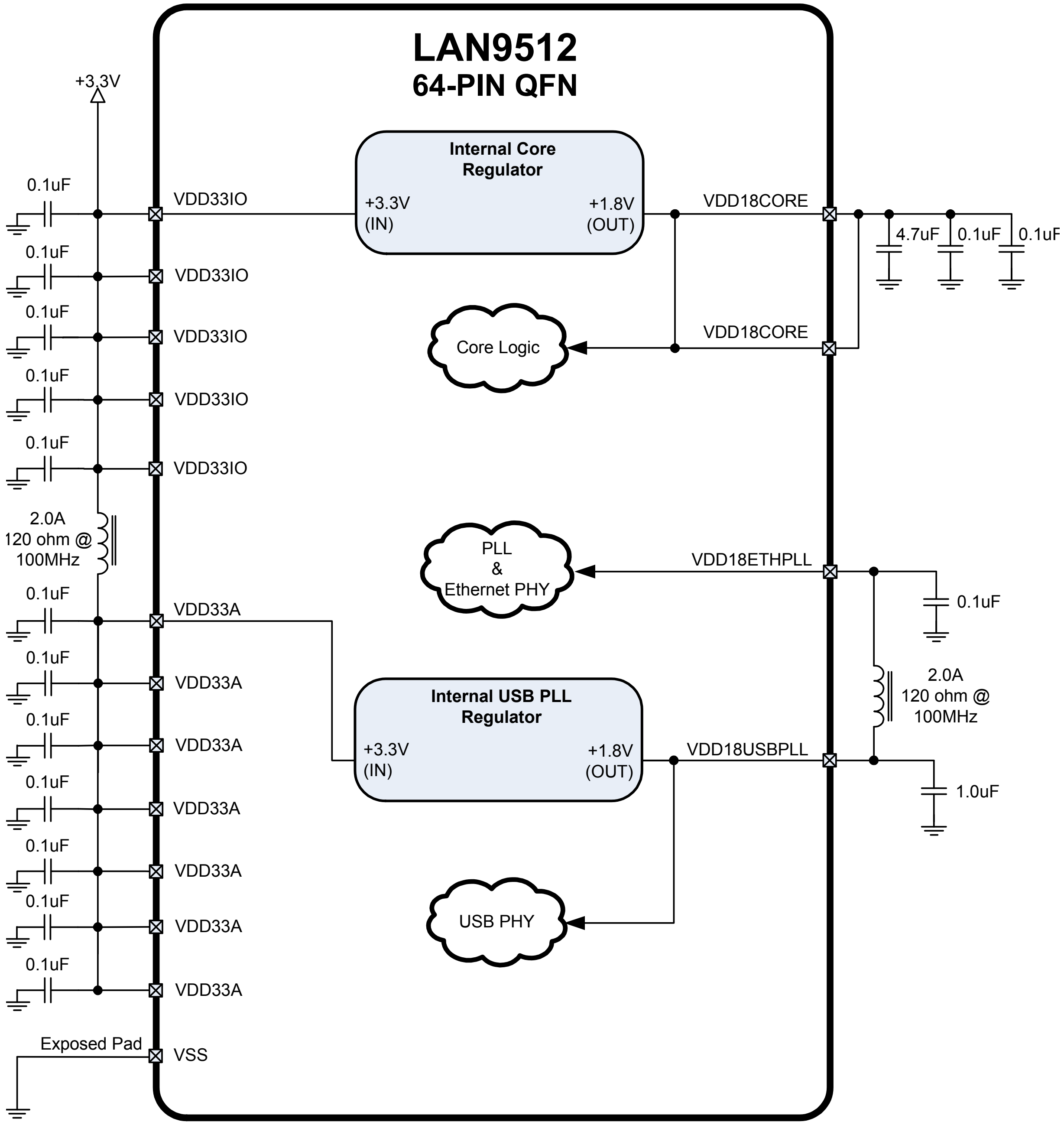


Figure 2.2 Power Connections

2.2 Port Power Control

This section details the usage of the port power control pins PRTCTL[3:2].

2.2.1 Port Power Control Using a USB Power Switch

The LAN9512/LAN9512i has a single port power control and over-current sense signal for each downstream port. When disabling port power the driver will actively drive a ‘0’. To avoid unnecessary power dissipation, the internal pull-up resistor will be disabled at that time. When port power is enabled, the output driver is disabled and the pull-up resistor is enabled, creating an open drain output. If there is an over-current situation, the USB Power Switch will assert the open drain OCS signal. The schmitt trigger input will recognize this situation as a low. The open drain output does not interfere. The overcurrent sense filter handles the transient conditions, such as low voltage, while the device is powering up.

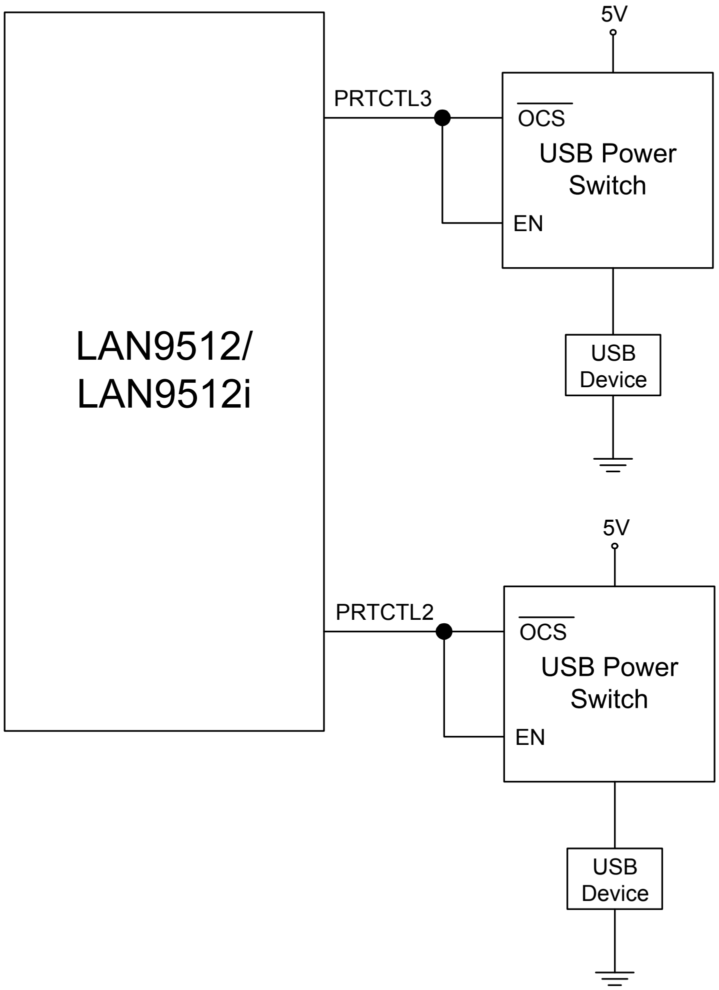


Figure 2.3 Port Power Control with USB Power Switch

2.2.2 Port Power Control Using a Poly Fuse

When using theLAN9512/LAN9512i with a poly fuse, an external diode must be used (See Figure 2.4). When disabling port power, the driver will drive a ‘0’. This procedure will have no effect since the external diode will isolate the pin from the load. When port power is enabled, the output driver is disabled and the pull-up resistor is enabled, which creates an open drain output. This means that the pull-up resistor is providing 3.3 volts to the anode of the diode. If there is an over-current situation, the poly fuse will open. This will cause the cathode of the diode to go to 0 volts. The anode of the diode will be at 0.7 volts, and the Schmidt trigger input will register this as a low, resulting in an overcurrent detection. The open drain output does not interfere.

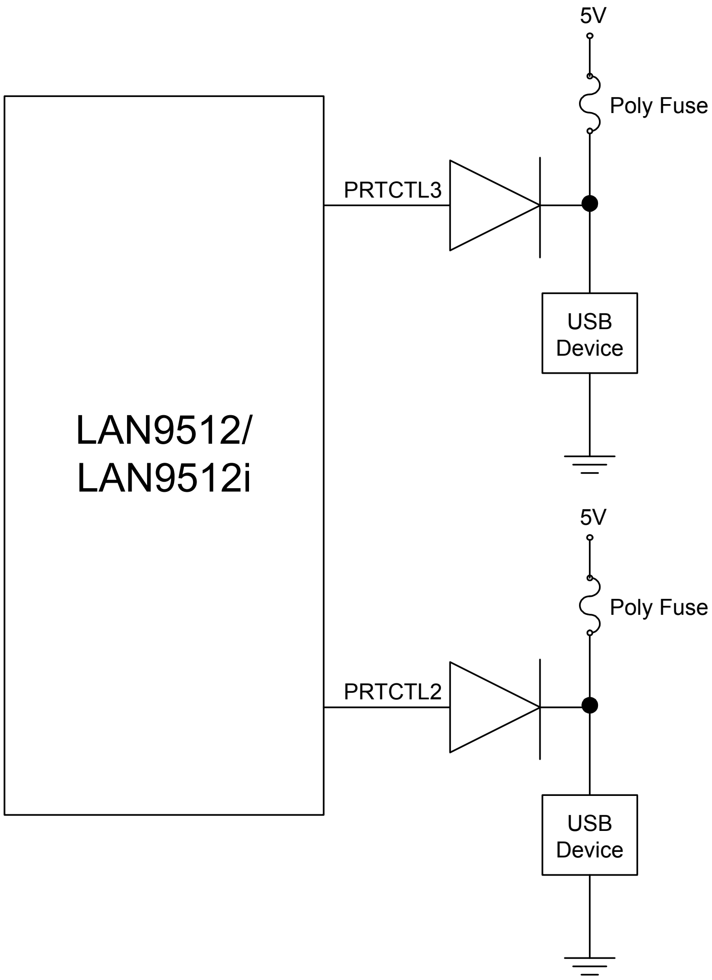


Figure 2.4 Port Power Control with Poly Fuse

Many customers use a single poly fuse to power all their devices. For the ganged situation, all power control pins must be tied together.

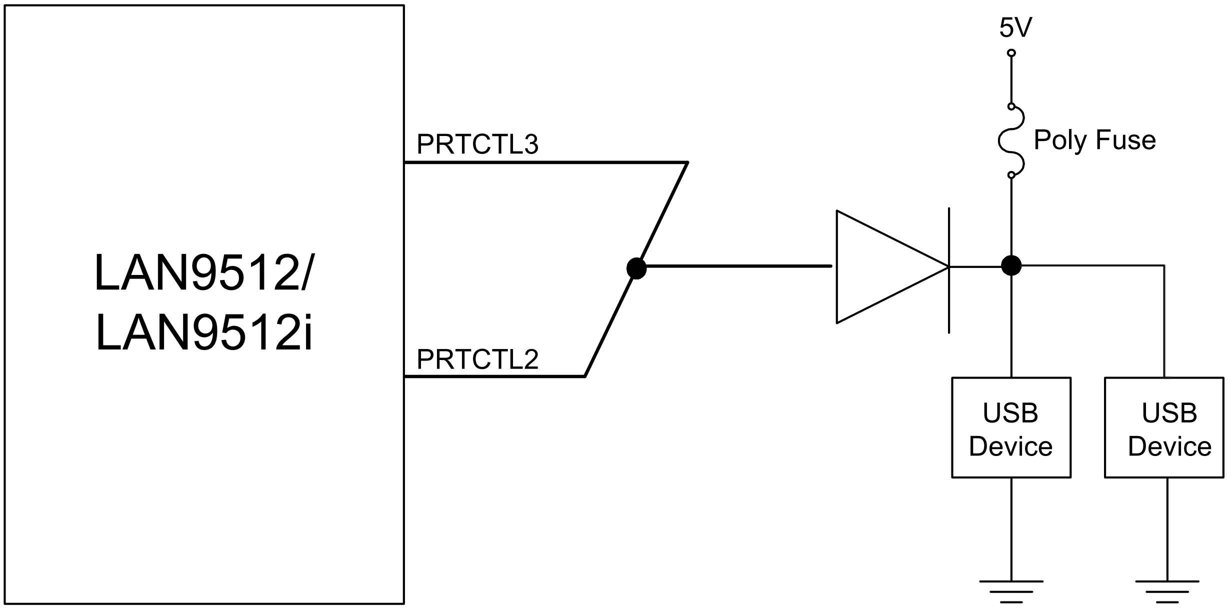


Figure 2.5 Port Power with Ganged Control with Poly Fuse

2.3 Buffer Types

Table 2.9 Buffer Types

BUFFER TYPE	DESCRIPTION
IS	Schmitt-triggered Input
IS_5V	5V Tolerant Schmitt-triggered Input
O8	Output with 8mA sink and 8mA source
OD8	Open-drain output with 8mA sink
O12	Output with 12mA sink and 12mA source
OD12	Open-drain output with 12mA sink
PU	50uA (typical) internal pull-up. Unless otherwise noted in the pin description, internal pull-ups are always enabled. **Note:** Internal pull-up resistors prevent unconnected inputs from floating. Do not rely on internal resistors to drive signals external to LAN9512/LAN9512i. When connected to a load that must be pulled high, an external resistor must be added.
PD	50uA (typical) internal pull-down. Unless otherwise noted in the pin description, internal pull-downs are always enabled. **Note:** Internal pull-down resistors prevent unconnected inputs from floating. Do not rely on internal resistors to drive signals external to LAN9512/LAN9512i. When connected to a load that must be pulled low, an external resistor must be added.
AI	Analog input
AIO	Analog bi-directional
ICLK	Crystal oscillator input pin
OCLK	Crystal oscillator output pin
P	Power pin

Chapter 3 Functional Description

3.1 Functional Overview

The LAN9512/LAN9512i USB Hub with integrated Ethernet Controller consists of the following major functional blocks:

- USB Common Block (UCB)
- Clock Generator and Power Management Controller (CPM)
- USB PHYs
- USB Hub
- Ethernet Controller consisting of the following major components:
    - USB 2.0 Device Controller (UDC)
    - FIFO Controller (FCT) and Associated SRAM
    - 10/100 Ethernet MAC
- 10/100 Ethernet PHY
- IEEE 1149.1 Tap Controller
- EEPROM Controller (EPC)

The following sections discuss the features of each block. A block diagram of LAN9512/LAN9512i is shown in Figure 1.1 Internal Block Diagram on page 14.

3.2 USB Common Block (UCB)

The UCB manages the device’s PLL and generates various clocks, including the system clocks, of the device.

3.3 Clock Generator and Power Management Controller (CPM)

The CPM generates clocks for use by various components of the device and manages the device’s power states.

3.4 USB PHYs

There are a total of 3 USB PHYs - 2 Downstream and 1 Upstream. Each USB PHY has the USB interface on one end, and connects to the USB Hub on the other. The Parallel-to-serial/serial-to-parallel conversion, bit stuffing, and NRZI coding / decoding are handled in the PHY block. The PHY is capable of operating in the USB 1.1 and 2.0 modes.

3.5 USB Hub

The USB 2.0 Hub is fully specification compliant to the Universal Serial Bus Specification Revision 2.0 April 27,2000 (12/7/2000 and 5/28/2002 Errata). The Hub will attach to an upstream port as a Full-Speed Hub or as a Full-/Hi-Speed Hub. The Hub supports Low-Speed, Full-Speed, and Hi-Speed (if operating as a Hi-Speed Hub) downstream devices on all of the enabled downstream ports.

For performance reasons, the Hub provides one Transaction Translator (TT) per port (defined as Multi-TT configuration), and each TT has 1512 bytes of periodic buffer space and 272 Bytes of non- periodic buffer space (divided into 4 non-periodic buffers per TT), for a total of 1784 bytes of buffer space for each Transaction Translator.

When configured as a Single-TT Hub (required by USB 2.0 Specification), the Single Transaction Translator will have 1512 bytes of periodic buffer space and 272 bytes of non-periodic buffer space (divided into 4 non-periodic buffers per TT), for a total of 1784 bytes of buffer space for the entire Transaction Translator. Each Transaction Translator's buffer is divided as shown in Table 3.1. Note the table does not reflect actual RAM size.

Table 3.1 Transaction Translator Buffer Chart

Periodic Start-Split Descriptors	256 Bytes
Periodic Start-Split Data	752 Bytes
Periodic Complete-Split Descriptors	128 Bytes
Periodic Complete-Split Data	376 Bytes
Non-Periodic Descriptors	16 Bytes
Non-Periodic Data	256 Bytes
Total for each Transaction Translator	1784 Bytes

3.5.1 Hub Configuration Options

The Hub supports a large number of features (some are mutually exclusive), and must be configured in order to correctly function when attached to a USB Host controller. There are two principal ways to configure the Hub: via EEPROM load or by internal default settings. In all cases, the configuration method used will be determined by the EEPROM controller, upon determination of the presence or absence of a properly formatted EEPROM. Refer to Section 3.10, "EEPROM Controller (EPC)," on page 123 for details.

3.5.2 Power Switching Polarity

The Hub only supports "active high" port power controllers.

3.5.3 VBus Detect

According to Section 7.2.1 of the USB2.0 Specification, a downstream port can never provide power to its USBDPn or USBDMn pin pull-up resistors unless the upstream port's VBUS is in the asserted (powered) state. The VBUS_DET pin on the device monitors the state of the upstream VBUS signal and will not pull-up the USBDPn pin's resistor if VBUS is not active. If VBUS goes from an active to an inactive state (Not Powered), the Hub will remove power from the USBDPn pin's pull-up resistor within 10 seconds.

3.5.4 Hub USB Descriptors

There are two methods for configuring the Hub descriptors: Internal Default Configuration or EEPROM load. The SMSC Hub will not electrically attach to the USB until after it has loaded valid data for all user-defined descriptor fields (either through EEPROM or Internal Defaults).

Note: Unlike the Ethernet Controller USB descriptors, whose images may be loaded directly from EEPROM into internal device memory, the Hub USB descriptors are internally generated according to the Hub registers, whose values are specified in EEPROM addresses 20h through 39h. If a valid EEPROM is not present, the Hub USB descriptors are generated according to the Internal Defaults of the Hub registers.

3.5.4.1 Hub Attached as a Full-Speed Device, (High-Speed Disabled)

When High-Speed capability is disabled via OEM configuration options, the Hub will only be able to attach as a Full-Speed device, and the following descriptor information applies.

3.5.4.1.1 STANDARD DEVICE DESCRIPTOR

Table 3.2 provides the descriptor values for Full-Speed operation.

Table 3.2 Device Descriptor (Full Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	12h	No	Size of the Descriptor in Bytes (18 bytes)
01h	bDescriptorType	1	01h	No	Device Descriptor (0x01)
02h	bcdUSB	2	0200h	No	USB Specification Number which device complies to.
04h	bDeviceClass	1	09h	No	Class Code assigned by USB-IF for Hubs
05h	bDeviceSubClass	1	00h	No	Subclass Code assigned by USB-IF for Hubs
06h	bDeviceProtocol	1	00h	No	Protocol Code
07h	bMaxPacketSize	1	40h	No	Maximum Packet Size for Endpoint 0 (64 bytes)
08h	IdVendor	2	0424h	Yes	Vendor ID - OEM value defined in EEPROM
0Ah	IdProduct	2	9512h	Yes	Product ID - OEM value defined in EEPROM
0Ch	bcdDevice	2	Note 3.1	Yes	Device Release Number - OEM value defined in EEPROM
0Eh	iManufacturer	1	00h	No	This optional string is not supported
0Fh	iProduct	1	00h	No	This optional string is not supported
10h	iSerialNumber	1	00h	No	This optional string is not supported
11h	bNumConfigurations	1	01h	No	Supports one configuration

Note 3.1 Default value is dependent on device revision. MSB matches the device revision and LSB hardcoded to 00h.

3.5.4.1.2 CONFIGURATION DESCRIPTOR

Table 3.3 provides the Configuration Descriptor values for Full-Speed operation.

Table 3.3 Configuration Descriptor (Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of the Configuration Descriptor in bytes (9 bytes)
01h	bDescriptorType	1	02h	No	Configuration Descriptor (0x02)
02h	wTotalLength	2	yyyyyh	No	Total combined length of all descriptors for this configuration (configuration, interface, endpoint, and class- or vendor-specific). yyyyh = 0019h
04h	bNumInterfaces	1	01h	No	Number of Interfaces supported by this configuration
05h	bConfigurationValue	1	01h	No	Value to use as an argument to the SetConfiguration() request to select this configuration
06h	iConfiguration	1	00h	No	Index of String Descriptor describing this configuration (string not supported)
07h	bmAttributes	1	E0h	Yes	Configuration characteristics Note 3.2
08h	bMaxPower	1	01h	Yes	Maximum Power Consumption of the Hub From VBUS when fully operational Note 3.3

Note 3.2 Communicates the capabilities of the Hub regarding Remote Wakeup capability, and also reports the self-power status. The value reported to the Host always indicates that the Hub supports Remote Wakeup.

The value reported to the Host is dependent upon the Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register:

= A0h for Bus-Powered (Self or Bus Power (SELF_BUS_PWR) = 0)
= E0h for Self-Powered (Self or Bus Power (SELF_BUS_PWR) = 1)

All other values are reserved.

Note 3.3 This value includes all support circuitry associated with the Hub (including an attached “embedded” peripheral if the Hub is part of a compound device), and is in 2 mA increments. The Hub supports Self-Powered and Bus-Powered operation. The Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register is used to determine which of the values below are reported. The value reported to the Host must coincide with the current operating mode, and will be determined by the following rules:

Max Power Bus-Powered (MAX_PWR_BP) if Self or Bus Power (SELF_BUS_PWR) = 0
Max Power Self-Powered (MAX_PWR_SP) if Self or Bus Power (SELF_BUS_PWR) = 1

In all cases, the reported value is sourced from the MAX POWER data field (for Self or Bus power) that was loaded from EEPROM.

3.5.4.1.3 INTERFACE DESCRIPTOR

Table 3.4 provides the Interface Descriptor values for Full-Speed operation.

Table 3.4 Interface Descriptor (Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in Bytes (9 Bytes
01h	bDescriptorType	1	04h	No	Interface Descriptor (0x04)
02h	bInterfaceNumber	1	00h	No	Number identifying this Interface
03h	bAlternateSetting	1	00h	No	Value used to select this alternative setting for the interface
04h	bNumEndpoints	1	01h	No	Number of Endpoints used for this interface (Less endpoint 0)
05h	bInterfaceClass	1	09h	No	Hub Class Code
06h	bInterfaceSubClass	1	00h	No	Subclass Code
07h	bInterfaceProtocol	1	00h	No	Protocol Code
08h	iInterface	1	00h	No	Index of String Descriptor Describing this interface (strings not supported)

3.5.4.1.4 ENDPOINT DESCRIPTOR

Table 3.5 provides the Endpoint Descriptor values for Full-Speed operation.

Table 3.5 Endpoint Descriptor (For Status Change Endpoint, Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	07h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	05h	No	Endpoint Descriptor
02h	bEndpointAddress	1	81h	No	The address of the endpoint on the USB device
03h	bmAttributes	1	03h	No	Describes the endpoint’s attributes (interrupt only, no synchronization, data endpoint)
04h	wMaxPacketSize	2	0001h	No	Maximum Packet Size this endpoint is capable of sending
06h	bInterval	1	FFh	No	Interval for polling endpoint for data transfers (Maximum possible)

3.5.4.2 Hub Attached as a Full-Speed Device, But is High-Speed Capable

When attached as a Full-Speed device (most likely due to being connected to a Host Controller or Operating System that is not High-Speed capable), the following descriptor information applies.

3.5.4.2.1 STANDARD DEVICE DESCRIPTOR

Table 3.6 provides the descriptor values for Full-Speed operation.

Table 3.6 Device Descriptor (Full Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	12h	No	Size of the Descriptor in Bytes (18 bytes)
01h	bDescriptorType	1	01h	No	Device Descriptor (0x01)
02h	bcdUSB	2	0200h	No	USB Specification Number which device complies to.
04h	bDeviceClass	1	09h	No	Class Code assigned by USB-IF for Hubs
05h	bDeviceSubClass	1	00h	No	Subclass Code assigned by USB-IF for Hubs
06h	bDeviceProtocol	1	00h	No	Protocol Code
07h	bMaxPacketSize	1	40h	No	Maximum Packet Size for Endpoint 0
08h	IdVendor	2	0424h	Yes	Vendor ID - OEM value defined in EEPROM
0Ah	IdProduct	2	9512h	Yes	Product ID - OEM value defined in EEPROM
0Ch	bcdDevice	2	Note 3.4	Yes	Device Release Number - OEM value defined in EEPROM
0Eh	iManufacturer	1	00h	No	This optional string is not supported
0Fh	iProduct	1	00h	No	This optional string is not supported
10h	iSerialNumber	1	00h	No	This optional string is not supported
11h	bNumConfigurations	1	01h	No	Supports one configuration

Note 3.4 Default value is dependent on device revision. MSB matches the device revision and LSB hardcoded to 00h.

3.5.4.2.2 DEVICE QUALIFIER DESCRIPTOR

Table 3.7 provides the Device Qualifier Descriptor values for the High-Speed operation.

Table 3.7 Device Qualifier Descriptor (High-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	0Ah	No	Size of Descriptor in bytes (10 bytes)
01h	bDescriptorType	1	06h	No	Device Qualifier Descriptor (0x06)
02h	bcdUSB	2	0200h	No	USB Specification Number which device complies to.
04h	bDeviceClass	1	09h	No	Class Code assigned by USB-IF for Hubs
05h	bDeviceSubClass	1	00h	No	Subclass Code assigned by USB-IF for Hubs
06h	bDeviceProtocol	1	xxh	Yes	Protocol Code xx = 01 if OEM selects Single-TT xx = 02 if OEM selects Multi-TT
07h	bMaxPacketSize0	1	40h	No	Maximum Packet Size for the other speed
08h	bNumConfigurations	1	01h	No	Supports one other speed configuration
09h	Reserved	1	00h	No	Must be zero

3.5.4.2.3 CONFIGURATION DESCRIPTORS

Table 3.8 provides the Configuration Descriptor values for Full-Speed operation.

Table 3.8 Configuration Descriptor (Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of the Configuration Descriptor in bytes (9 bytes)
01h	bDescriptorType	1	02h	No	Configuration Descriptor (0x02)
02h	wTotalLength	2	yyyyyh	No	Total combined length of all descriptors for this configuration (configuration, interface, endpoint, and class- or vendor-specific). yyyyh = 0019h
04h	bNumInterfaces	1	01h	No	Number of Interfaces supported by this configuration
05h	bConfigurationValue	1	01h	No	Value to use as an argument to the SetConfiguration() request to select this configuration
06h	iConfiguration	1	00h	No	Index of String Descriptor describing this configuration (string not supported)
07h	bmAttributes	1	E0h	Yes	Configuration characteristics Note 3.5
08h	bMaxPower	1	01h	Yes	Maximum Power Consumption of the Hub From VBUS when fully operational Note 3.6

Note 3.5 Communicates the capabilities of the Hub regarding Remote Wakeup capability, and also reports the self-power status. The value reported to the Host always indicates that the Hub supports Remote Wakeup.

The value reported to the Host is dependent upon the Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register:

= A0h for Bus-Powered (Self or Bus Power (SELF_BUS_PWR) = 0)
= E0h for Self-Powered (Self or Bus Power (SELF_BUS_PWR) = 1)

All other values are reserved.

Note 3.6 This value includes all support circuitry associated with the Hub (including an attached “embedded” peripheral if the Hub is part of a compound device), and is in 2mA increments. The Hub supports Self-Powered and Bus-Powered operation. The Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register is used to determine which of the values below are reported. The value reported to the Host must coincide with the current operating mode, and will be determined by the following rules:

Max Power Bus-Powered (MAX_PWR_BP) if Self or Bus Power (SELF_BUS_PWR) = 0
Max Power Self-Powered (MAX_PWR_SP) if Self or Bus Power (SELF_BUS_PWR) = 1

In all cases, the reported value is sourced from the MAX POWER data field (for Self or Bus power) that was loaded from EEPROM.

3.5.4.2.4 INTERFACE DESCRIPTOR (FULL-SPEED)

Table 3.9 provides the Interface Descriptor values for Full-Speed operation.

Table 3.9 Interface Descriptor (Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in Bytes (9 Bytes
01h	bDescriptorType	1	04h	No	Interface Descriptor (0x04)
02h	bInterfaceNumber	1	00h	No	Number identifying this Interface
03h	bAlternateSetting	1	00h	No	Value used to select this alternative setting for the interface
04h	bNumEndpoints	1	01h	No	Number of Endpoints used for this interface (Less endpoint 0)
05h	bInterfaceClass	1	09h	No	Hub Class Code
06h	bInterfaceSubClass	1	00h	No	Subclass Code
07h	bInterfaceProtocol	1	00h	No	Protocol Code
08h	iInterface	1	00h	No	Index of String Descriptor Describing this interface (strings not supported)

3.5.4.2.5 ENDPOINT DESCRIPTOR (FULL-SPEED)

Table 3.10 provides the Endpoint Descriptor values for Full-Speed operation.

Table 3.10 Endpoint Descriptor (For Status Change Endpoint, Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	07h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	05h	No	Endpoint Descriptor
02h	bEndpointAddress	1	81h	No	The address of the endpoint on the USB device
03h	bmAttributes	1	03h	No	Describes the endpoint's attributes (interrupt only, no synchronization, data endpoint)
04h	wMaxPacketSize	2	0001h	No	Maximum Packet Size this endpoint is capable of sending
06h	bInterval	1	FFh	No	Interval for polling endpoint for data transfers (Maximum possible)

3.5.4.2.6 OTHER-SPEED CONFIGURATION DESCRIPTOR

Table 3.44 provides the Other-Speed Configuration Descriptor values for High-Speed operation.

Table 3.11 Other Speed Configuration Descriptor (High-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in bytes (9 bytes)
01h	bDescriptorType	1	07h	No	Other Speed Configuration Descriptor (0x07)
02h	wTotalLength	2	zzzzh	No	Total combined length of all descriptors for this configuration. zzzzh = 0019h if Multiple TT Enable (MTT_ENABLE) = 0 zzzzh = 0029h if Multiple TT Enable (MTT_ENABLE) = 1
04h	bNumInterfaces	1	01h	No	Number of Interfaces supported by this configuration
05h	bConfigurationValue	1	01h	No	Value to use as an argument to select this configuration
06h	iConfiguration	1	00h	No	Index of String Descriptor describing this configuration (string not supported)
07h	bmAttributes	1	A0h	Yes	Configuration characteristics Note 3.7
08h	bMaxPower	1	01h	Yes	Maximum Power Consumption of the Hub From VBUS when fully operational Note 3.8

Note 3.7 Communicates the capabilities of the Hub regarding Remote Wakeup capability, and also reports the self-power status. The value reported to the Host always indicates that the Hub supports Remote Wakeup.

The value reported to the Host is dependent upon the Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register:

= A0h for Bus-Powered (Self or Bus Power (SELF_BUS_PWR) = 0)
= E0h for Self-Powered (Self or Bus Power (SELF_BUS_PWR) = 1)

All other values are reserved.

Note 3.8 This value includes all support circuitry associated with the Hub (including an attached "embedded" peripheral if the Hub is part of a compound device), and is in 2mA increments. The Hub supports Self-Powered and Bus-Powered operation. The Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register is used to determine which of the values below are reported. The value reported to the Host must coincide with the current operating mode, and will be determined by the following rules:

Max Power Bus-Powered (MAX_PWR_BP) if Self or Bus Power (SELF_BUS_PWR) = 0
Max Power Self-Powered (MAX_PWR_SP) if Self or Bus Power (SELF_BUS_PWR) = 1

In all cases, the reported value is sourced from the MAX POWER data field (for Self or Bus power) that was loaded from EEPROM.

3.5.4.2.7 INTERFACE DESCRIPTOR (SINGLE-TT)

Table 3.12 provides the Interface Descriptor values for Single-TT, High-Speed operation.

Table 3.12 Interface Descriptor (Single-TT, High-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in Bytes (9 Bytes
01h	bDescriptorType	1	04h	No	Interface Descriptor (0x04)
02h	bInterfaceNumber	1	00h	No	Number identifying this Interface
03h	bAlternateSetting	1	00h	No	Value used to select this alternative setting for the interface
04h	bNumEndpoints	1	01h	No	Number of Endpoints used for this interface (Less endpoint 0)
05h	bInterfaceClass	1	09h	No	Hub Class Code
06h	bInterfaceSubClass	1	00h	No	Subclass Code
07h	bInterfaceProtocol	1	xxh	No	Protocol Code xx = 00h if bNumInterfaces = 01h (Single-TT) xx = 01h if bNumInterfaces = 02 (Multi-TT)
08h	iInterface	1	00h	No	Index of String Descriptor Describing this interface (strings not supported)

3.5.4.2.8 ENDPOINT DESCRIPTOR (SINGLE-TT)

Table 3.13 provides the Endpoint Descriptor values for Single-TT operation.

Table 3.13 Endpoint Descriptor (For Status Change Endpoint, Single-TT)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	07h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	05h	No	Endpoint Descriptor
02h	bEndpointAddress	1	81h	No	The address of the endpoint on the USB device
03h	bmAttributes	1	03h	No	Describes the endpoint's attributes (interrupt only, no synchronization, data endpoint)
04h	wMaxPacketSize	2	0001h	No	Maximum Packet Size this endpoint is capable of sending
06h	bInterval	1	0Ch	No	Interval for polling endpoint for data transfers (Maximum possible)

3.5.4.2.9 INTERFACE DESCRIPTOR (MULTI-TT)

Table 3.14 provides the Interface Descriptor values for Multi-TT, High-Speed operation.

Note: This is only available if Multi-TT is reported in the Other_Speed Configuration Descriptor.

Table 3.14 Interface Descriptor (Multi-TT, High-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in Bytes (9 Bytes
01h	bDescriptorType	1	04h	No	Interface Descriptor (0x04)
02h	bInterfaceNumber	1	00h	No	Number identifying this Interface
03h	bAlternateSetting	1	01h	No	Value used to select this alternative setting for the interface
04h	bNumEndpoints	1	01h	No	Number of Endpoints used for this interface (Less endpoint 0)
05h	bInterfaceClass	1	09h	No	Hub Class Code
06h	bInterfaceSubClass	1	00h	No	Subclass Code
07h	bInterfaceProtocol	1	02h	No	Protocol Code
08h	iInterface	1	00h	No	Index of String Descriptor Describing this interface (strings not supported)

3.5.4.2.10 ENDPOINT DESCRIPTOR (MULTI-TT)

Table 3.13 provides the Endpoint Descriptor values for Multi-TT operation.

Note: This is only available if Multi-TT is reported in the Other-Speed Configuration Descriptor.

Table 3.15 Endpoint Descriptor (For Status Change Endpoint, Multi-TT)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	07h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	05h	No	Endpoint Descriptor
02h	bEndpointAddress	1	81h	No	The address of the endpoint on the USB device
03h	bmAttributes	1	03h	No	Describes the endpoint's attributes (interrupt only, no synchronization, data endpoint)
04h	wMaxPacketSize	2	0001h	No	Maximum Packet Size this endpoint is capable of sending
06h	bInterval	1	0Ch	No	Interval for polling endpoint for data transfers (Maximum possible)

3.5.4.3 Hub Attached as a High-Speed Device (OEM-Configured for Single-TT Support Only)

The following tables provide descriptor information for OEM-Configured Single-TT-Only Hubs attached for use with High-Speed devices.

3.5.4.3.1 STANDARD DEVICE DESCRIPTOR

Table 3.16 provides the descriptor values for High-Speed operation.

Table 3.16 Device Descriptor (High-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	12h	No	Size of the Descriptor in Bytes (18 bytes)
01h	bDescriptorType	1	01h	No	Device Descriptor (0x01)
02h	bcdUSB	2	0200h	No	USB Specification Number which device complies to.
04h	bDeviceClass	1	09h	No	Class Code assigned by USB-IF for Hubs
05h	bDeviceSubClass	1	00h	No	Subclass Code assigned by USB-IF for Hubs
06h	bDeviceProtocol	1	01h	No	Protocol Code
07h	bMaxPacketSize	1	40h	No	Maximum Packet Size for Endpoint 0
08h	IdVendor	2	0424h	Yes	Vendor ID - OEM value defined in EEPROM
0Ah	IdProduct	2	9512h	Yes	Product ID - OEM value defined in EEPROM
0Ch	bcdDevice	2	Note 3.9	Yes	Device Release Number - OEM value defined in EEPROM
0Eh	iManufacturer	1	00h	No	This optional string is not supported
0Fh	iProduct	1	00h	No	This optional string is not supported
10h	iSerialNumber	1	00h	No	This optional string is not supported
11h	bNumConfigurations	1	01h	No	Supports one configuration

Note 3.9 Default value is dependent on device revision. MSB matches the device revision and LSB hardcoded to 00h.

3.5.4.3.2 DEVICE QUALIFIER DESCRIPTOR

Table 3.17 provides the Device Qualifier Descriptor values for Full-Speed operation.

Table 3.17 Device Qualifier Descriptor (Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	0Ah	No	Size of Descriptor in bytes (10 bytes)
01h	bDescriptorType	1	06h	No	Device Qualifier Descriptor (0x06)
02h	bcdUSB	2	0200h	No	USB Specification Number which device complies to.
04h	bDeviceClass	1	09h	No	Class Code assigned by USB-IF for Hubs
05h	bDeviceSubClass	1	00h	No	Subclass Code assigned by USB-IF for Hubs
06h	bDeviceProtocol	1	00h	No	Protocol Code
07h	bMaxPacketSize0	1	40h	No	Maximum Packet Size for the other speed
08h	bNumConfigurations	1	01h	No	Supports one other speed configuration
09h	Reserved	1	00h	No	Must be zero

3.5.4.3.3 CONFIGURATION DESCRIPTOR

Table 3.18 provides the Configuration Descriptor values for High-Speed, Single-TT Only operation.

Table 3.18 Configuration Descriptor (High-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of the Configuration Descriptor in bytes (9 bytes)
01h	bDescriptorType	1	02h	No	Configuration Descriptor (0x02)
02h	wTotalLength	2	yyyyyh	No	Total combined length of all descriptors for this configuration (configuration, interface, endpoint, and class- or vendor-specific). yyyyh = 0019h
04h	bNumInterfaces	1	01h	No	Number of Interfaces supported by this configuration
05h	bConfigurationValue	1	01h	No	Value to use as an argument to the SetConfiguration() request to select this configuration
06h	iConfiguration	1	00h	No	Index of String Descriptor describing this configuration (string not supported)
07h	bmAttributes	1	E0h	Yes	Configuration characteristics Note 3.10
08h	bMaxPower	1	01h	Yes	Maximum Power Consumption of the Hub From VBUS when fully operational Note 3.11

Note 3.10 Communicates the capabilities of the Hub regarding Remote Wakeup capability, and also reports the self-power status. The value reported to the Host always indicates that the Hub supports Remote Wakeup.

The value reported to the Host is dependent upon the Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register:

= A0h for Bus-Powered (Self or Bus Power (SELF_BUS_PWR) = 0)
= E0h for Self-Powered (Self or Bus Power (SELF_BUS_PWR) = 1)

All other values are reserved.

Note 3.11 This value includes all support circuitry associated with the Hub (including an attached "embedded" peripheral if the Hub is part of a compound device), and is in 2mA increments. The Hub supports Self-Powered and Bus-Powered operation. The Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register is used to determine which of the values below are reported. The value reported to the Host must coincide with the current operating mode, and will be determined by the following rules:

Max Power Bus-Powered (MAX_PWR_BP) if Self or Bus Power (SELF_BUS_PWR) = 0
Max Power Self-Powered (MAX_PWR_SP) if Self or Bus Power (SELF_BUS_PWR) = 1

In all cases, the reported value is sourced from the MAX POWER data field (for Self or Bus power) that was loaded from EEPROM.

3.5.4.3.4 INTERFACE DESCRIPTOR (SINGLE-TT)

Table 3.19 provides the Interface Descriptor values for High-Speed, Single-TT operation.

Table 3.19 Interface Descriptor (High-Speed, Single-TT)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in Bytes (9 Bytes
01h	bDescriptorType	1	04h	No	Interface Descriptor (0x04)
02h	bInterfaceNumber	1	00h	No	Number identifying this Interface
03h	bAlternateSetting	1	00h	No	Value used to select this alternative setting for the interface
04h	bNumEndpoints	1	01h	No	Number of Endpoints used for this interface (Less endpoint 0)
05h	bInterfaceClass	1	09h	No	Hub Class Code
06h	bInterfaceSubClass	1	00h	No	Subclass Code
07h	bInterfaceProtocol	1	00h	No	Protocol Code (Single-TT)
08h	iInterface	1	00h	No	Index of String Descriptor Describing this interface (strings not supported)

3.5.4.3.5 ENDPOINT DESCRIPTOR (SINGLE-TT)

Table 3.20 provides the Endpoint Descriptor values for Single-TT operation.

Table 3.20 Endpoint Descriptor (For Status Change Endpoint, Single-TT)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	07h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	05h	No	Endpoint Descriptor
02h	bEndpointAddress	1	81h	No	The address of the endpoint on the USB device
03h	bmAttributes	1	03h	No	Describes the endpoint’s attributes (interrupt only, no synchronization, data endpoint)
04h	wMaxPacketSize	2	0001h	No	Maximum Packet Size this endpoint is capable of sending
06h	bInterval	1	0Ch	No	Interval for polling endpoint for data transfers (Maximum possible)

3.5.4.3.6 OTHER-SPEED CONFIGURATION DESCRIPTOR (FULL-SPEED)

Table 3.21 provides the Other-Speed Configuration Descriptor values for Full-Speed operation.

Table 3.21 Other Speed Configuration Descriptor (Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in bytes (9 bytes)
01h	bDescriptorType	1	07h	No	Other Speed Configuration Descriptor (0x07)
02h	wTotalLength	2	yyyyh	No	Total combined length of all descriptors for this configuration. yyyyh = 0019h
04h	bNumInterfaces	1	01h	No	Number of Interfaces supported by this configuration
05h	bConfigurationValue	1	01h	No	Value to use as an argument to select this configuration
06h	iConfiguration	1	00h	No	Index of String Descriptor describing this configuration (string not supported)
07h	bmAttributes	1	E0h	Yes	Configuration characteristics Note 3.12
08h	bMaxPower	1	01h	Yes	Maximum Power Consumption of the Hub From VBUS when fully operational Note 3.13

Note 3.12 Communicates the capabilities of the Hub regarding Remote Wakeup capability, and also reports the self-power status. The value reported to the Host always indicates that the Hub supports Remote Wakeup.

The value reported to the Host is dependent upon the Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register:

= A0h for Bus-Powered (Self or Bus Power (SELF_BUS_PWR) = 0)
= E0h for Self-Powered (Self or Bus Power (SELF_BUS_PWR) = 1)

All other values are reserved.

Note 3.13 This value includes all support circuitry associated with the Hub (including an attached “embedded” peripheral if the Hub is part of a compound device), and is in 2mA increments. The Hub supports Self-Powered and Bus-Powered operation. The Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register is used to determine which of the values below are reported. The value reported to the Host must coincide with the current operating mode, and will be determined by the following rules:

Max Power Bus-Powered (MAX_PWR_BP) if Self or Bus Power (SELF_BUS_PWR) = 0
Max Power Self-Powered (MAX_PWR_SP) if Self or Bus Power (SELF_BUS_PWR) = 1

In all cases, the reported value is sourced from the MAX POWER data field (for Self or Bus power) that was loaded from EEPROM.

3.5.4.3.7 INTERFACE DESCRIPTOR (FULL-SPEED)

Table 3.22 provides the Interface Descriptor values for Full-Speed operation.

Table 3.22 Interface Descriptor (Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in Bytes (9 Bytes
01h	bDescriptorType	1	04h	No	Interface Descriptor (0x04)
02h	bInterfaceNumber	1	00h	No	Number identifying this Interface
03h	bAlternateSetting	1	00h	No	Value used to select this alternative setting for the interface
04h	bNumEndpoints	1	01h	No	Number of Endpoints used for this interface (Less endpoint 0)
05h	bInterfaceClass	1	09h	No	Hub Class Code
06h	bInterfaceSubClass	1	00h	No	Subclass Code
07h	bInterfaceProtocol	1	00h	No	Protocol Code
08h	iInterface	1	00h	No	Index of String Descriptor Describing this interface (strings not supported)

3.5.4.3.8 ENDPOINT DESCRIPTOR (FULL-SPEED)

Table 3.23 provides the Endpoint Descriptor values for Full-Speed operation.

Table 3.23 Endpoint Descriptor (For Status Change Endpoint, Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	07h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	05h	No	Endpoint Descriptor
02h	bEndpointAddress	1	81h	No	The address of the endpoint on the USB device
03h	bmAttributes	1	03h	No	Describes the endpoint’s attributes (interrupt only, no synchronization, data endpoint)
04h	wMaxPacketSize	2	0001h	No	Maximum Packet Size this endpoint is capable of sending
06h	bInterval	1	FFh	No	Interval for polling endpoint for data transfers (Maximum possible)

3.5.4.4 Hub Attached as a High-Speed Device (OEM-Configured as Multi-TT Capable)

The following tables provide descriptor information for OEM-Configured Multi-TT Hubs attached for use with High-Speed devices.

3.5.4.4.1 STANDARD DEVICE DESCRIPTOR

Table 3.24 provides the descriptor values for High-Speed operation.

Table 3.24 Device Descriptor (High-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	12h	No	Size of the Descriptor in Bytes (18 bytes)
01h	bDescriptorType	1	01h	No	Device Descriptor (0x01)
02h	bcdUSB	2	0200h	No	USB Specification Number which device complies to.
04h	bDeviceClass	1	09h	No	Class Code assigned by USB-IF for Hubs
05h	bDeviceSubClass	1	00h	No	Subclass Code assigned by USB-IF for Hubs
06h	bDeviceProtocol	1	02h	No	Protocol Code (Multi-TTs)
07h	bMaxPacketSize	1	40h	No	Maximum Packet Size for Endpoint 0
08h	IdVendor	2	0424h	Yes	Vendor ID - OEM value defined in EEPROM
0Ah	IdProduct	2	9512h	Yes	Product ID - OEM value defined in EEPROM
0Ch	bcdDevice	2	Note 3.14	Yes	Device Release Number - OEM value defined in EEPROM
0Eh	iManufacturer	1	00h	No	This optional string is not supported
0Fh	iProduct	1	00h	No	This optional string is not supported
10h	iSerialNumber	1	00h	No	This optional string is not supported
11h	bNumConfigurations	1	01h	No	Supports one configuration

Note 3.14 Default value is dependent on device revision. MSB matches the device revision and LSB hardcoded to 00h.

3.5.4.4.2 DEVICE QUALIFIER DESCRIPTOR

Table 3.25 provides the Device Qualifier Descriptor values for Full-Speed operation.

Table 3.25 Device Qualifier Descriptor (Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	0Ah	No	Size of Descriptor in bytes (10 bytes)
01h	bDescriptorType	1	06h	No	Device Qualifier Descriptor (0x06)
02h	bcdUSB	2	0200h	No	USB Specification Number which device complies to.
04h	bDeviceClass	1	09h	No	Class Code assigned by USB-IF for Hubs
05h	bDeviceSubClass	1	00h	No	Subclass Code assigned by USB-IF for Hubs
06h	bDeviceProtocol	1	00h	No	Protocol Code
07h	bMaxPacketSize0	1	40h	No	Maximum Packet Size for the other speed
08h	bNumConfigurations	1	01h	No	Supports one other speed configuration
09h	Reserved	1	00h	No	Must be zero

3.5.4.4.3 CONFIGURATION DESCRIPTOR

Table 3.26 provides the Configuration Descriptor values for High-Speed operation.

Table 3.26 Configuration Descriptor (High-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of the Configuration Descriptor in bytes (9 bytes)
01h	bDescriptorType	1	02h	No	Configuration Descriptor (0x02)
02h	wTotalLength	2	yyyyyh	No	Total combined length of all descriptors for this configuration (configuration, interface, endpoint, and class- or vendor-specific). yyyyh = 0029h
04h	bNumInterfaces	1	01h	No	Number of Interfaces supported by this configuration
05h	bConfigurationValue	1	01h	No	Value to use as an argument to the SetConfiguration() request to select this configuration
06h	iConfiguration	1	00h	No	Index of String Descriptor describing this configuration (string not supported)
07h	bmAttributes	1	E0h	Yes	Configuration characteristics Note 3.15
08h	bMaxPower	1	01h	Yes	Maximum Power Consumption of the Hub From VBUS when fully operational Note 3.16

Note 3.15 Communicates the capabilities of the Hub regarding Remote Wakeup capability, and also reports the self-power status. The value reported to the Host always indicates that the Hub supports Remote Wakeup.

The value reported to the Host is dependent upon the Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register:

= A0h for Bus-Powered (Self or Bus Power (SELF_BUS_PWR) = 0)
= E0h for Self-Powered (Self or Bus Power (SELF_BUS_PWR) = 1)

All other values are reserved.

Note 3.16 This value includes all support circuitry associated with the Hub (including an attached "embedded" peripheral if the Hub is part of a compound device), and is in 2mA increments. The Hub supports Self-Powered and Bus-Powered operation. The Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register is used to determine which of the values below are reported. The value reported to the Host must coincide with the current operating mode, and will be determined by the following rules:

Max Power Bus-Powered (MAX_PWR_BP) if Self or Bus Power (SELF_BUS_PWR) = 0
Max Power Self-Powered (MAX_PWR_SP) if Self or Bus Power (SELF_BUS_PWR) = 1

In all cases, the reported value is sourced from the MAX POWER data field (for Self or Bus power) that was loaded from EEPROM.

3.5.4.4.4 INTERFACE DESCRIPTOR (SINGLE-TT)

Table 3.27 provides the Interface Descriptor values for High-Speed, Single-TT operation.

Table 3.27 Interface Descriptor (High-Speed, Single-TT)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in Bytes (9 Bytes
01h	bDescriptorType	1	04h	No	Interface Descriptor (0x04)
02h	bInterfaceNumber	1	00h	No	Number identifying this Interface
03h	bAlternateSetting	1	00h	No	Value used to select this alternative setting for the interface
04h	bNumEndpoints	1	01h	No	Number of Endpoints used for this interface (Less endpoint 0)
05h	bInterfaceClass	1	09h	No	Hub Class Code
06h	bInterfaceSubClass	1	00h	No	Subclass Code
07h	bInterfaceProtocol	1	01h	No	Protocol Code (Single-TT)
08h	iInterface	1	00h	No	Index of String Descriptor Describing this interface (strings not supported)

3.5.4.4.5 ENDPOINT DESCRIPTOR (SINGLE-TT)

Table 3.28 provides the Endpoint Descriptor values for Single-TT operation.

Table 3.28 Endpoint Descriptor (For Status Change Endpoint, Single-TT)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	07h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	05h	No	Endpoint Descriptor
02h	bEndpointAddress	1	81h	No	The address of the endpoint on the USB device
03h	bmAttributes	1	03h	No	Describes the endpoint's attributes (interrupt only, no synchronization, data endpoint)
04h	wMaxPacketSize	2	0001h	No	Maximum Packet Size this endpoint is capable of sending
06h	bInterval	1	0Ch	No	Interval for polling endpoint for data transfers (Maximum possible)

3.5.4.4.6 INTERFACE DESCRIPTOR (MULTI-TT)

Table 3.29 provides the Interface Descriptor values for High-Speed, Multi-TT operation.

Table 3.29 Interface Descriptor (High-Speed, Multi-TT)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in Bytes (9 Bytes
01h	bDescriptorType	1	04h	No	Interface Descriptor (0x04)
02h	bInterfaceNumber	1	00h	No	Number identifying this Interface
03h	bAlternateSetting	1	01h	No	Value used to select this alternative setting for the interface
04h	bNumEndpoints	1	01h	No	Number of Endpoints used for this interface (Less endpoint 0)
05h	bInterfaceClass	1	09h	No	Hub Class Code
06h	bInterfaceSubClass	1	00h	No	Subclass Code
07h	bInterfaceProtocol	1	02h	No	Protocol Code (Multi-TT)
08h	iInterface	1	00h	No	Index of String Descriptor Describing this interface (strings not supported)

3.5.4.4.7 ENDPOINT DESCRIPTOR (MULTI-TT)

Table 3.30 provides the Endpoint Descriptor values for Multi-TT operation.

Table 3.30 Endpoint Descriptor (For Status Change Endpoint, Multi-TT)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	07h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	05h	No	Endpoint Descriptor
02h	bEndpointAddress	1	81h	No	The address of the endpoint on the USB device
03h	bmAttributes	1	03h	No	Describes the endpoint's attributes (interrupt only, no synchronization, data endpoint)
04h	wMaxPacketSize	2	0001h	No	Maximum Packet Size this endpoint is capable of sending
06h	bInterval	1	0Ch	No	Interval for polling endpoint for data transfers (Maximum possible)

3.5.4.4.8 OTHER-SPEED CONFIGURATION DESCRIPTOR

Table 3.31 provides the Other-Speed Configuration Descriptor values for Full-Speed operation.

Table 3.31 Other Speed Configuration Descriptor (Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in bytes (9 bytes)
01h	bDescriptorType	1	07h	No	Other Speed Configuration Descriptor (0x07)
02h	wTotalLength	2	yyyyh	No	Total combined length of all descriptors for this configuration. yyyyh = 0019h
04h	bNumInterfaces	1	01h	No	Number of Interfaces supported by this configuration
05h	bConfigurationValue	1	01h	No	Value to use as an argument to select this configuration
06h	iConfiguration	1	00h	No	Index of String Descriptor describing this configuration (string not supported)
07h	bmAttributes	1	E0h	Yes	Configuration characteristics Note 3.17
08h	bMaxPower	1	01h	Yes	Maximum Power Consumption of the Hub From VBUS when fully operational Note 3.18

Note 3.17 Communicates the capabilities of the Hub regarding Remote Wakeup capability, and also reports the self-power status. The value reported to the Host always indicates that the Hub supports Remote Wakeup.

The value reported to the Host is dependent upon the Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register:

= A0h for Bus-Powered (Self or Bus Power (SELF_BUS_PWR) = 0)
= E0h for Self-Powered (Self or Bus Power (SELF_BUS_PWR) = 1)

All other values are reserved.

Note 3.18 This value includes all support circuitry associated with the Hub (including an attached "embedded" peripheral if the Hub is part of a compound device), and is in 2mA increments. The Hub supports Self-Powered and Bus-Powered operation. The Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register is used to determine which of the values below are reported. The value reported to the Host must coincide with the current operating mode, and will be determined by the following rules:

Max Power Bus-Powered (MAX_PWR_BP) if Self or Bus Power (SELF_BUS_PWR) = 0
Max Power Self-Powered (MAX_PWR_SP) if Self or Bus Power (SELF_BUS_PWR) = 1

In all cases, the reported value is sourced from the MAX POWER data field (for Self or Bus power) that was loaded from EEPROM.

3.5.4.4.9 INTERFACE DESCRIPTOR (FULL-SPEED)

Table 3.32 provides the Interface Descriptor values for Full-Speed operation.

Table 3.32 Interface Descriptor (Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in Bytes (9 Bytes
01h	bDescriptorType	1	04h	No	Interface Descriptor (0x04)
02h	bInterfaceNumber	1	00h	No	Number identifying this Interface
03h	bAlternateSetting	1	00h	No	Value used to select this alternative setting for the interface
04h	bNumEndpoints	1	01h	No	Number of Endpoints used for this interface (Less endpoint 0)
05h	bInterfaceClass	1	09h	No	Hub Class Code
06h	bInterfaceSubClass	1	00h	No	Subclass Code
07h	bInterfaceProtocol	1	00h	No	Protocol Code
08h	iInterface	1	00h	No	Index of String Descriptor Describing this interface (strings not supported)

3.5.4.4.10 ENDPOINT DESCRIPTOR (FULL-SPEED)

Table 3.33 provides the Endpoint Descriptor values for Full-Speed operation.

Table 3.33 Endpoint Descriptor (For Status Change Endpoint, Full-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	07h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	05h	No	Endpoint Descriptor
02h	bEndpointAddress	1	81h	No	The address of the endpoint on the USB device
04h	bmAttributes	1	03h	No	Describes the endpoint’s attributes (interrupt only, no synchronization, data endpoint)
05h	wMaxPacketSize	2	0001h	No	Maximum Packet Size this endpoint is capable of sending
07h	bInterval	1	FFh	No	Interval for polling endpoint for data transfers (Maximum possible)

3.5.4.5 Class-Specific Hub Descriptor

Table 3.34 provides class-specific Hub descriptor values for Full-Speed and High-Speed operation.

Note: The Hub must respond to Hub Class Descriptor type 29h (the USB 1.1 and USB 2.0 value) and 00h (the USB 1.0 value).

Table 3.34 Class-Specific Hub Descriptor (Full-Speed & High-Speed)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	29h	No	Hub Descriptor Type
02h	bNbrPortsL	1	03h	Yes	Number of downstream facing ports this Hub supports Note 3.19 Note 3.20
03h	wHubCharacteristics	2	000Dh	Yes	Defines support for Logical power switching mode, Compound Device support, Over-current protection, TT Think Time and Port Indicator support. Note 3.19 Note 3.21
05h	bPwrOn2PwrGood	1	32h	Yes	Time (in 2 ms intervals) from the time the power-on sequence begins on a port until power is good on that port. Note 3.19 The value contained in the Power On Time (POWER_ON_TIME) field of the Power-On Time (PWRT) Register is directly reported to the Host, and is determined by the EEPROM load.
06h	bHubContrCurrent	1	01h	Yes	Maximum current requirements of the Hub Controller electronics in mA Note 3.19 Note 3.22
07h	bDeviceRemovable	1	02h	Yes	Indicates if the port has a removable device attached. Note 3.19 The value contained in the Non-Removable Device (NR_DEVICE) field of the Non-Removable Devices (NRD) Register is directly reported to the Host, and is determined by the EEPROM load.
08h	bPortPwrCtlMask	1	FFh	No	Field for backwards USB 1.0 compatibility

Note 3.19 Please refer to Section 11.23.2.1 of the USB Specification for additional details regarding the use of this field.

Note 3.20 The value reported is implementation dependent, and is derived from the value defined during EEPROM load. The Port Disable Self-Powered (PORT_DIS_SP) field of the Port Disable For Self-Powered Operation (PDS) Register defines the ports that are permanently disabled when in Self-Powered operation. The Port Disable Bus-Powered (PORT_DIS_BP) field of the Port Disable For Bus-Powered Operation (PDB) Register defines the ports that are permanently disabled when in Bus-Powered operation. The Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register is used to determine which field is used to create the value reported.

Internal logic will subtract the number of ports which are disabled, from the total number available (which is 3 or 5 depending upon the part number), and will report the remainder as the number of ports supported. The value reported to the Host must coincide with the current operating mode, and will be determined by the following rules:

Port Disable Bus-Powered (PORT_DIS_BP) if Self or Bus Power (SELF_BUS_PWR) = 0
Port Disable Self-Powered (PORT_DIS_SP) if Self or Bus Power (SELF_BUS_PWR) = 1

Note 3.21 The values delivered to a Host are all derived from values defined during EEPROM load, and are assigned as follows:

D1:0 = 00b if Port Power Switching (PORT_PWR) = 0
D1:0 = 01b if Port Power Switching (PORT_PWR) = 1

D2 = Compound Device (COMPOUND)

D4:3 = Over Current Sense (CURRENT_SNS)

D6:5 = 00b for 8FS (max) bit times of TT think time

D7 = hardcoded to 0b (no Port Indicator Support)

D15:8 = 00000000b

Note 3.22 This field reports the maximum current that only the Hub consumes from upstream VBUS when fully operational. This value includes all support circuitry associated with the Hub (but does not include the current consumption of any permanently attached peripherals if the Hub is part of a compound device). The field reports the values in 2mA increments.

The Hub supports Self-Powered and Bus-Powered operation. The Self or Bus Power (SELF_BUS_PWR) bit of the Config Data Byte 1 (CFG1) Register is used to determine which of the stored values are reported. The value reported to the Host must coincide with the current operating mode, and will be determined by the following rules:

‘HC_MAX_C_BP’ if Self or Bus Power (SELF_BUS_PWR) = 0
‘HC_MAX_C_SP’ if Self or Bus Power (SELF_BUS_PWR) = 1

In all cases, the reported value is sourced from the Hub Controller Max Current data field (for Self or Bus power) that was determined by EEPROM load.

3.6 USB 2.0 Device Controller (UDC)

USB functionality within the Ethernet Controller is supported by the UDC (USB Device Controller), URX (USB Bulk Out Receiver), UTX (USB Bulk In Receiver), and CTL (USB Control Block). They are illustrated in Figure 1.2 Ethernet Controller Block Diagram on page 15.

The URX and UTX implement the Bulk Out and Bulk In endpoints respectively. The CTL manages control and interrupt endpoints.

The UDC is connected to Port 1 of the Hub and is a USB low-level protocol interpreter. The UDC controls the USB bus protocol, packet generation/extraction, PID/Device ID parsing, and CRC coding/decoding with autonomous error handling. It is capable of operating either in USB 1.1 or 2.0 compliant modes. It has autonomous protocol handling functions like stall condition clearing on setup packets, suspend/resume/reset conditions, and remote wakeup. It also autonomously handles error conditions such as retry for CRC errors, Data toggle errors, and generation of NYET, STALL, ACK and NACK, depending on the endpoint buffer status.

The UDC is configured to support one configuration, one interface, one alternate setting, and four endpoints.

3.6.1 Supported Endpoints

Table 3.35 lists the supported endpoints. The following subsections discuss these endpoints in detail.

Table 3.35 Supported Endpoints

ENDPOINT NUMBER	DESCRIPTION
0	Control Endpoint
1	Bulk In Endpoint
2	Bulk Out Endpoint
3	Interrupt Endpoint

The URX and UTX implement the Bulk Out and Bulk In endpoints, respectively. The CTL manages the Control and Interrupt endpoints.

3.6.1.1 Endpoint 1 (Bulk In)

The Bulk In Endpoint is controlled by the UTX (USB Bulk In Transmitter). The UTX is responsible for encapsulating Ethernet data into a USB Bulk In packet. Ethernet frames are retrieved from the FCT's RX FIFO. The UTX supports the following two modes of operation: MEF and SEF, selected via the Multiple Ethernet Frames per USB Packet (MEF) bit of the Hardware Configuration Register (HW_CFG).

- MEF: Multiple Ethernet frames per Bulk In packet. This mode will maximize USB bus utilization by allowing multiple Ethernet frames to be packed into a USB packet. Frames greater than 512 bytes are split across multiple Bulk In packets.
- SEF: Single Ethernet frame per Bulk In packet. This mode will not maximize USB bus utilization, but can potentially ease the burden on a low end Host processor. Frames greater than 512 bytes are split across multiple Bulk In packets.

Each Ethernet frame is prepended with an RX Status Word by the FCT. The status word contains the frame length that is used by the UTX to perform the encapsulation functions. The RX Status word is generated by the RX Transaction Layer Interface (RX TLI). The TLI resides between the MAC and the FCT.

Padding may be inserted between the RX Status Word and the Ethernet frame by the FCT. This condition exists when the RXDOFF register has a nonzero value (refer to Section 4.3.5, "Hardware Configuration Register (HW_CFG)" for details). The padding is implemented by the FCT barrel shifting the Ethernet frame by the specified byte offset.

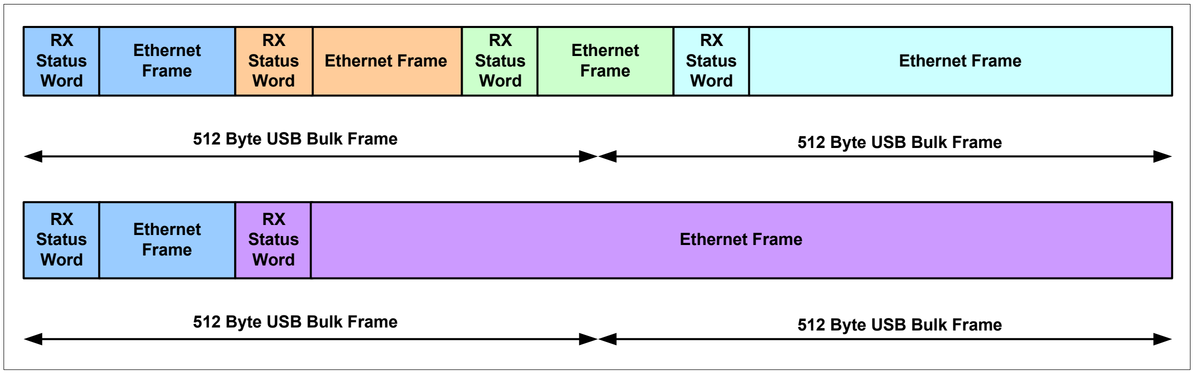


Figure 3.1 MEF USB Encapsulation

In accordance with the USB protocol, the UTX terminates a burst with either a ZLP or a Bulk In packet with a size of less than the Bulk In maximum packet size (512 for HS, 64 for FS). The ZLP is needed when the total amount of data transmitted is a multiple of a Bulk In packet size. The UTX monitors the RX FIFO size signal from the FCT to determine when a burst has ended.

Note: In SEF mode, a ZLP is transmitted if the Ethernet frame is the same size as a Bulk In packet, or a multiple of the Bulk In packet size.

An Ethernet frame always begins on a DWORD boundary. In MEF mode, the UTX will not concatenate the end of the current frame and the beginning of the next frame into the same DWORD. Therefore, the last DWORD of an Ethernet frame may have unused bytes added to ensure DWORD alignment of the next frame. The addition of pad bytes depends on whether another frame is available for transmission after the current one. If the current frame is the last frame to be transmitted, no pad bytes will be added, as the USB protocol allows for termination of the packet on a byte boundary. If, however, another frame is available for transmission, the current frame will be padded out so that it ends on the DWORD boundary. This ensures the next frame to be transmitted will start on a DWORD boundary.

If the UTX receives a Bulk In token when the RX FIFO is empty, it will transmit a ZLP.

Note: Any unused bytes that were added to the last DWORD of a frame are not counted in the length field of the RX status word.

Note: The Host ignores unused bytes that exist in the first and last words of an Ethernet frame.

Note: When using SEF mode, there will never be any unused bytes added for end alignment padding. The USB transfer always ends on the last byte of the Ethernet frame.

Note: When RX COE is enabled, the last byte would pertain to the RX COE word.

Once a decision is made to end a transfer and a short packet or ZLP has been sent, it is possible that an Ethernet frame will arrive prior to the UTX seeing an ACK from the Host for the previous Bulk In packet. In this case, the UTX must continue to repeat the short packet or ZLP until the ACK is received for the end of the previous transfer. The UTX must not start a new transfer, or re-use the previous data toggle, to begin sending the next Ethernet frame until the ACK has been received for the end of the previous transfer.

In order to more efficiently utilize USB bandwidth in MEF mode, the UTX has a mechanism for delaying the transmission of a short packet, or ZLP. This mode entails having the UTX wait a time defined by the Bulk In Delay Register (BULK_IN_DLY) before terminating the burst. A value of zero in this register disables this feature. By default, a delay of 34 us is used.

After the UTX transmits the last USB wPacketSize packet in a burst, the UTX will enable an internal timer. When the internal timer is equal to the Bulk In Delay, any Bulk In data will be transmitted upon reception of the next Bulk In Token. If enough data arrives before the timer elapses to build at least one maximum sized packet, then the UTX will transmit this packet when it receives the next Bulk In Token. After packet transmission, the UTX will then reset its internal timer and delay the short packet, or ZLP, transmission until the Bulk In Delay time elapses.

In the case where the FIFO is empty and a single Ethernet packet less than the USB wPacketSize has been received, the UTX will enable its internal timer. If enough data arrives before the timer elapses to build at least one maximum sized packet, then the UTX will transmit this packet when it receives the next Bulk In Token and will reset the internal timer. Otherwise, the short packet, or ZLP, is sent in response to the first Bulk In Token received after the timer expires.

The UTX will NAK any Bulk In tokens while waiting for the Bulk In Delay to elapse. This NAK response is not affected by the Bulk In Empty Response (BIR). The Bulk In Empty Response (BIR) setting only applies after the Bulk In Delay time expires.

The UTX, via the Burst Cap Register (BURST_CAP), is capable or prematurely terminating a burst. When the amount transmitted exceeds the value specified in this register, the UTX transmits a ZLP after the current Bulk In packet completes. The Burst Cap Register (BURST_CAP) uses units of USB packet size (512 bytes). To enable use of the Burst Cap register, the Burst Cap Enable (BCE) bit in the Hardware Configuration Register (HW_CFG) must be set. For proper operation, the BURST_CAP field should be set to value greater than 4 for HS mode and greater than 32 for FS mode. Burst Cap enforcement is disabled if BURST_CAP is set to a value less than or equal to 4 for HS mode and less than or equal to 32 for FS mode.

Whenever Burst Cap enforcement is disabled, the UTX will respond with a ZLP (when Bulk In Empty Response (BIR) =0) or with NAK (when Bulk In Empty Response (BIR) = 1).

Whenever Burst Cap enforcement is enabled (BURST_CAP value is legal), the following holds:

- For HS Operation:

  Let BURST = BURST_CAP * 512

  The burst may terminate at BURST-4, BURST-3, BURST-2, BURST-1, or BURST bytes, or, when the RX FIFO runs out of data. The burst is terminated with either a short USB packet or with a ZLP.

- For FS operation:

  The burst will terminate after BURST_CAP * 64 bytes.

Note: Ethernet frames are not fragmented across bursts when using Burst Cap Enforcement.

In the case of an error condition, the UTX will issue a rewind to the FCT. This occurs when the UTX completes transmitting a Bulk In packet and does not receive an ACK from the Host. In this case, the next frame received by the UTX will be another In token and the Bulk In packet is retransmitted. When the ACK is finally received, the UTX notifies the FCT. The FCT will then advance the read head pointer to the next packet.

Note: The UTX will never stall the endpoint. The endpoint can only be stalled by the Host.

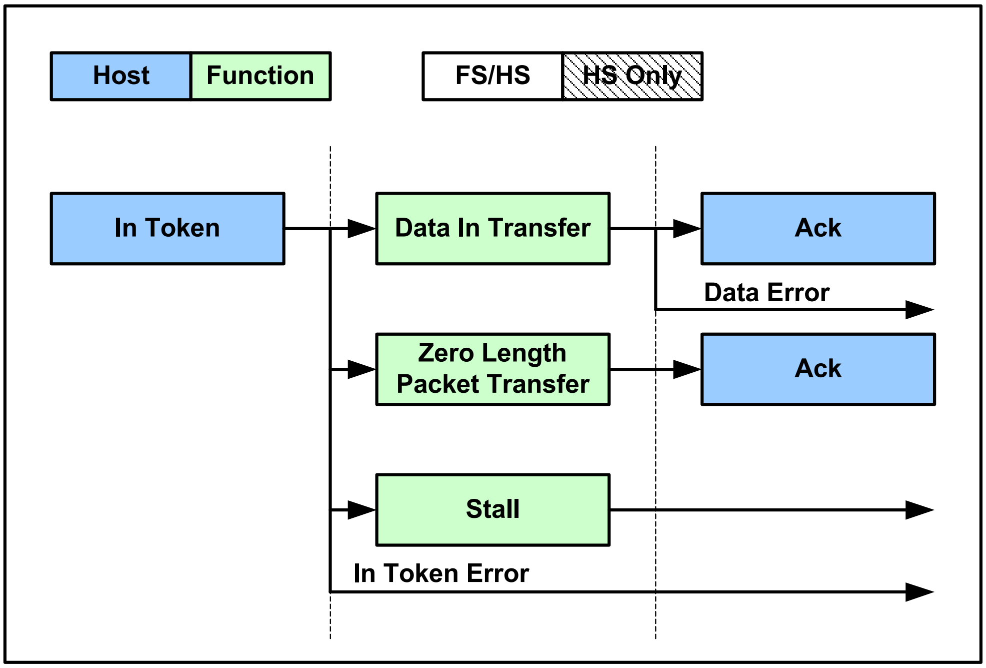


Figure 3.2 USB Bulk In Transaction Summary

3.6.1.2 Endpoint 2 (Bulk Out)

The Bulk Out Endpoint is controlled by the URX (USB Bulk Out Receiver). The URX is responsible for receiving Ethernet data encapsulated over a USB Bulk Out packet. Unlike the UTX, the URX does not explicitly track Ethernet frames. It views all received packets as purely USB data. The extraction of Ethernet frames is handled by the FCT and the Transaction Layer Interface (TLI).

The URX always simultaneously supports multiple Ethernet frames per USB packet, as well as a single Ethernet frame per USB packet. Unlike the UTX, no mechanism exists to select between modes.

The URX monitors the amount of free space in the TX FIFO. If at least 512 bytes of space exists, the URX can accept an additional Bulk In frame and responds to a Bulk Out Token with an ACK or NYET. The NYET response is used when less than 1024 bytes of free space exists. This means that the current Bulk Out packet was accepted, but room does not exist for a second packet. If less than 512 bytes exists, the URX responds with a NAK. The URX supports the PING protocol.

.

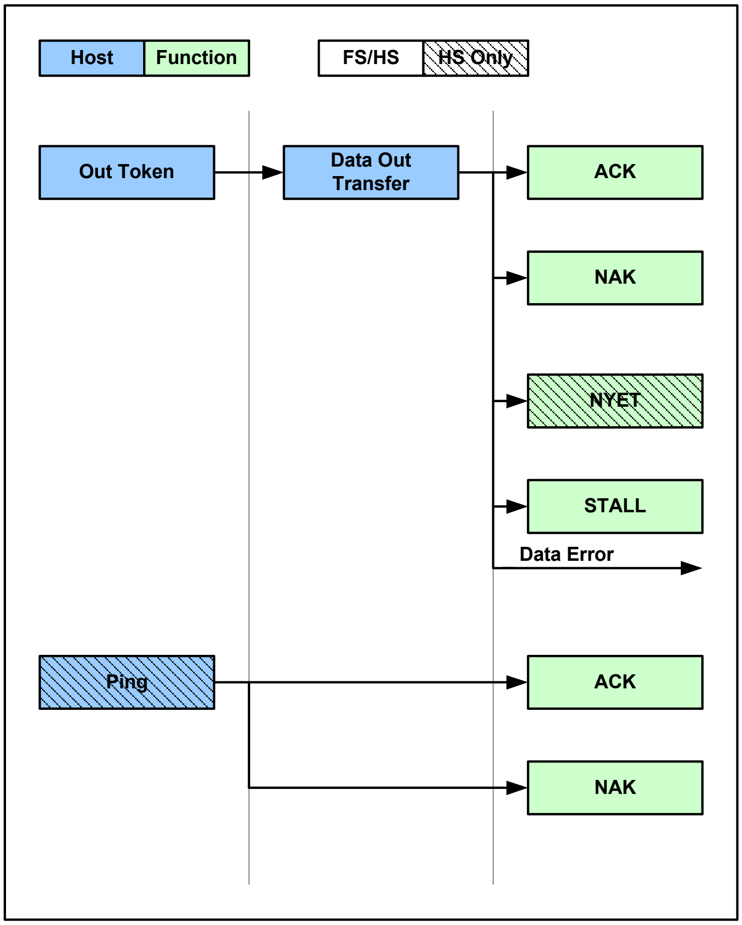


Figure 3.3 USB Bulk Out Transaction Summary

In the case where the Bulk Out packet is errored, the URX does not respond to the Host. The URX will request that the FCT rewinds the packet. It is the Hosts responsibility to retransmit the packet at a later time.

The FCT notifies the URX when it detects loss of sync. When this occurs, the URX stalls the Bulk Out pipe. This is an appropriate response, as loss of sync is a catastrophic error. This behavior is configurable via the Hardware Configuration Register (HW_CFG) on page 154.

3.6.1.3 Endpoint 3 (Interrupt)

The Interrupt endpoint is responsible for indicating device status at each polling interval. The Interrupt endpoint is implemented via the CTL module. When the endpoint is accessed, the Interrupt packet specified in Table 3.36 is presented to the Host.

Table 3.36 Interrupt Packet Format

BITS	DESCRIPTION
31:20	**RESERVED**
19	**MACRTO_INT**
18	**RX FIFO Has Frame.** The RX FIFO has at least one complete Ethernet frame.
17	**TXSTOP_INT**
16	**RXSTOP_INT**
15	**PHY_INT**
14	**TXE**
13	**TDFU**
12	**TDFO**
11	**RXDF_INT**
10:0	**GPIO_INT**

If there is no interrupt status to report, the device responds with a NAK.

Note: The polling interval is static and set through the EEPROM. The Host can change the polling interval by updating the contents of the EEPROM and resetting the part.

The interrupt status can be cleared by writing to Interrupt Status Register (INT_STS) on page 151.

3.6.1.4 Endpoint 0 (Control)

The Control endpoint is handled by the CTL (USB Control) module. The CTL module is responsible for handling USB standard commands, as well as USB vendor commands. In order to support these commands, the CTL must compile a variety of statistics and store the programmable portions of the USB descriptors. The supported USB commands can be found in Section 3.6.5, "USB Standard Commands," on page 75.

3.6.2 USB Command Processing

The UDC is programmed to decode USB commands. After a standard command is decoded by the UDC, it may be passed to the CTL for completion. The CTL is responsible for implementing the Get Descriptor and vendor commands.

In order to implement the Get Descriptor command for string descriptors, the CTL manages a 128x32 register file which stores the string values for Language ID, Manufacturer ID, Product ID, and Serial Number. The RAM's contents is initialized via the EEPROM, after a system reset occurs.

Table 3.37 String Descriptor Index Mappings

INDEX	STRING NAME
0	Language ID
1	Manufacturer ID
2	Product ID
3	Serial Number
4	Configuration String
5	Interface String

When the UDC decodes a Get Descriptor command, it will pass a pointer to the CTL. The CTL uses this pointer to determine what the command is and how to fill it.

3.6.3 Ethernet Controller USB Descriptors

The following subsections describe the Ethernet Controller USB descriptors. The images of these descriptors, if specified in EEPROM, are copied into internal memory for use.

3.6.3.1 Device Descriptor

The Device Descriptors are initialized based on values stored in EEPROM. Table 3.38 shows the default Device Descriptor values. These values are used for both Full-Speed and Hi-Speed operation.

Table 3.38 Device Descriptor

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	12h	Note 3.23	Size of the Descriptor in Bytes (18 bytes)
01h	bDescriptorType	1	01h	Note 3.23	Device Descriptor (0x01)
02h	bcdUSB	2	0200h	Note 3.24	USB Specification Number which device complies to.
04h	bDeviceClass	1	FFh	Yes	Class Code
05h	bDeviceSubClass	1	00h	Yes	Subclass Code
06h	bDeviceProtocol	1	01h	Yes	Protocol Code
07h	bMaxPacketSize	1	40h	Note 3.24	Maximum Packet Size for Endpoint 0
08h	IdVendor	2	0424h	Yes	Vendor ID
0Ah	IdProduct	2	EC00h	Yes	Product ID
0Ch	bcdDevice	2	Note 3.25	Yes	Device Release Number
0Eh	iManufacturer	1	00h	Yes	Index of Manufacturer String Descriptor
0Fh	iProduct	1	00h	Yes	Index of Product String Descriptor
10h	iSerialNumber	1	00h	Yes	Index of Serial Number String Descriptor
11h	bNumConfigurations	1	01h	Note 3.24	Number of Possible Configurations

Note 3.23 The descriptor length and descriptor type for Device Descriptors specified in EEPROM are “don’t cares” and are always overwritten by hardware as 0x12 and 0x01, respectively.

Note 3.24 Value is loaded from EEPROM, but must be equal to the Default Value in order to comply with the USB 2.0 Specification and provide for normal device operation. Specification of any other value will result in unwanted behavior and untoward operation.

Note 3.25 Default value is dependent on device revision. MSB matches the device revision and LSB is hardcoded to 00h.

3.6.3.2 Configuration Descriptor

The Configuration Descriptor is initialized based on values stored in EEPROM. Table 3.39 shows the default Configuration Descriptor values. These values are used for both Full-Speed and Hi-Speed operation.

Table 3.39 Configuration Descriptor

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	Note 3.26	Size of the Configuration Descriptor in bytes (9 bytes)
01h	bDescriptorType	1	02h	Note 3.27	Configuration Descriptor (0x02)
02h	wTotalLength	2	0027h	Note 3.26	Total length in bytes of data returned (39 bytes)
04h	bNumInterfaces	1	01h	Note 3.26	Number of Interfaces
05h	bConfigurationValue	1	01h	Note 3.26	Value to use as an argument to select this configuration
06h	iConfiguration	1	00h	Yes	Index of String Descriptor describing this configuration
07h	bmAttributes	1	E0h	Yes	Self powered and remote wakeup enabled.
08h	bMaxPower	1	01h	Yes	Maximum Power Consumption is 2 mA.

Note 3.26 Value is loaded from EEPROM, but must be equal to the Default Value in order to comply with the USB 2.0 Specification and provide for normal device operation. Specification of any other value will result in unwanted behavior and untoward operation.

Note 3.27 The descriptor type for Configuration Descriptors specified in EEPROM is a “don’t care” and is always overwritten by hardware as 0x02.

3.6.3.3 Interface Descriptor 0 Default

Table 3.40 shows the default value for Interface Descriptor 0. This descriptor is initialized based on values stored in EEPROM.

Table 3.40 Interface Descriptor 0

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	Note 3.28	Size of Descriptor in Bytes (9 Bytes
01h	bDescriptorType	1	04h	Note 3.28	Interface Descriptor (0x04)
02h	bInterfaceNumber	1	00h	Note 3.28	Number identifying this Interface
03h	bAlternateSetting	1	00h	Note 3.28	Value used to select alternative setting
04h	bNumEndpoints	1	03h	Note 3.28	Number of Endpoints used for this interface (Less endpoint 0)
05h	bInterfaceClass	1	FFh	Yes	Class Code
06h	bInterfaceSubClass	1	00h	Yes	Subclass Code
07h	bInterfaceProtocol	1	FFh	Yes	Protocol Code
08h	iInterface	1	00h	Yes	Index of String Descriptor Describing this interface

Note 3.28 Value is loaded from EEPROM, but must be equal to the Default Value in order to comply with the USB 2.0 Specification and provide for normal device operation. Specification of any other value will result in unwanted behavior and untoward operation.

3.6.3.4 Endpoint 1 (Bulk In) Descriptor

Table 3.41 shows the default value for Endpoint Descriptor 1. This descriptor is not initialized from values stored in EEPROM.

Table 3.41 Endpoint 1 Descriptor

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	07h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	05h	No	Endpoint Descriptor
02h	bEndpointAddress	1	81h	No	Endpoint Address
03h	bmAttributes	1	02h	No	Bulk Transfer Type
04h	wMaxPacketSize	2	Note 3.29	No	Maximum Packet Size this endpoint is capable of sending.
06h	bInterval	1	00h	No	Interval for polling endpoint data transfers. Ignored for bulk endpoints.

Note 3.29 64 bytes for Full-Speed mode. 512 bytes for Hi-Speed mode.

3.6.3.5 Endpoint 2 (Bulk Out) Descriptor

Table 3.42 shows the default value for Endpoint Descriptor 2. This descriptor is not initialized from values stored in EEPROM.

Table 3.42 Endpoint 2 Descriptor

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	07h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	05h	No	Endpoint Descriptor
02h	bEndpointAddress	1	02h	No	Endpoint Address
04h	bmAttributes	1	02h	No	Bulk Transfer Type
05h	wMaxPacketSize	2	Note 3.30	No	Maximum Packet Size this endpoint is capable of sending.
07h	bInterval	1	00h	No	Interval for polling endpoint data transfers. Ignored for bulk endpoints.

Note 3.30 64 bytes for Full-Speed mode. 512 bytes for Hi-Speed mode.

3.6.3.6 Endpoint 3 (Interrupt) Descriptor

Table 3.43 shows the default value for Endpoint Descriptor 3. Only the bInterval field of this descriptor is initialized from EEPROM.

Table 3.43 Endpoint 3 Descriptor

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	07h	No	Size of Descriptor in bytes
01h	bDescriptorType	1	05h	No	Endpoint Descriptor
02h	bEndpointAddress	1	83h	No	Endpoint Address
04h	bmAttributes	1	03h	No	Interrupt Transfer Type
05h	wMaxPacketSize	2	10h	No	Maximum Packet Size this endpoint is capable of sending.
07h	bInterval	1	Note 3.31	Yes	Interval for polling endpoint data transfers.

Note 3.31 This value is loaded from the EEPROM. A Full-Speed and Hi-Speed polling interval exists. If no EEPROM exists, then this value defaults to 04h for HS and 01h for FS.

3.6.3.7 Other Speed Configuration Descriptor

The fields in this descriptor are derived from Configuration Descriptor information that is stored in the EEPROM.

Table 3.44 Other Speed Configuration Descriptor

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	09h	Note 3.32	Size of Descriptor in bytes (9 bytes)
01h	bDescriptorType	1	07h	No	Other Speed Configuration Descriptor (0x07)
02h	wTotalLength	2	0027h	Note 3.32	Total length in bytes of data returned (39 bytes)
04h	bNumInterfaces	1	01h	Note 3.32	Number of Interfaces
05h	bConfigurationValue	1	01h	Note 3.32	Value to use as an argument to select this configuration
06h	iConfiguration	1	00h	Yes	Index of String Descriptor describing this configuration
07h	bmAttributes	1	E0h	Yes	Self powered and remote wakeup enabled.
08h	bMaxPower	1	01h	Yes	Maximum Power Consumption is 2 mA.

Note: EEPROM values are obtained for the Configuration Descriptor at the other USB speed. I.e., if the current operating speed is FS, then the HS Configuration Descriptor values are used, and vice-versa.

Note 3.32 Value is loaded from EEPROM, but must be equal to the Default Value in order to comply with the USB 2.0 Specification and provide for normal device operation. Specification of any other value will result in unwanted behavior and untoward operation.

3.6.3.8 Device Qualifier Descriptor

The fields in this descriptor are derived from Device Descriptor information that is stored in the EEPROM.

Table 3.45 Device Qualifier Descriptor

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	0Ah	No	Size of Descriptor in bytes (10 bytes)
01h	bDescriptorType	1	06h	No	Device Qualifier Descriptor (0x06)
02h	bcdUSB	2	0200h	Note 3.33	USB Specification Number which device complies to.
04h	bDeviceClass	1	FFh	Yes	Class Code
05h	bDeviceSubClass	1	00h	Yes	Subclass Code
06h	bDeviceProtocol	1	01h	Yes	Protocol Code
07h	bMaxPacketSize0	1	40h	Note 3.33	Maximum Packet Size
08h	bNumConfigurations	1	01h	Note 3.33	Number of Other-Speed Configurations
09h	Reserved	1	00h	No	Must be zero

Note: EEPROM values are from the Device Descriptor (including any EEPROM override) at the opposite HS/FS operating speed. I.e., if the current operating speed is HS, then Device Qualifier data is based on the FS Device Descriptor, and vice-versa.

Note 3.33 Value is loaded from EEPROM, but must be equal to the Default Value in order to comply with the USB 2.0 Specification and provide for normal device operation. Specification of any other value will result in unwanted behavior and untoward operation.

3.6.3.9 String Descriptors

3.6.3.9.1 STRING INDEX = 0 (LANGID)

Table 3.46 LANGID String Descriptor

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	04h	No	Size of LANGID Descriptor in bytes (4 bytes)
01h	bDescriptorType	1	03h	No	String Descriptor (0x03)
02h	LANGID	2	None	Yes	Must be set to 0x0409 (US English).

Note: If there is no valid/enabled EEPROM, or if all string lengths in the EEPROM are 0, then there are no strings, so any Host attempt to read the LANGID string will return stall in the Data Stage of the Control Transfer.

If there is a valid/enabled EEPROM, and if at least one of the string lengths in the EEPROM is not 0, then the value contained at EEPROM addresses 0x0A-0x0B will be returned. These must be 0x0409 to allow for proper device operation.

Note: The device ignores the LANGID field in Control Read's of Strings, and will not return the String (if it exists), regardless of whether the requested LANGID is 0x0409 or not.

3.6.3.9.2 STRING INDICES 1-5

Table 3.47 String Descriptor (Indices 1-5)

OFFSET	FIELD	SIZE (BYTES)	DEFAULT VALUE	LOADED FROM EEPROM	DESCRIPTION
00h	bLength	1	none	Yes	Size of the String Descriptor in bytes (4 bytes)
01h	bDescriptorType	1	none	Yes	String Descriptor (0x03)
02h	Unicode String	2*N	none	Yes	2 bytes per unicode character, no trailing NULL.

Note: If there is no valid/enabled EEPROM, or if the corresponding String Length and offset in the EEPROM for a given string index are zero, then that string does not exist, so any Host attempt to read that string will return stall in the Data Stage of the Control Transfer.

Note: The device returns whatever bytes are in the designated EEPROM area for each of these strings it is the responsibility of the EEPROM programmer to correctly set the bLength and bDescriptorType fields in the descriptor consistent with the byte length specified in the corresponding EEPROM locations.

3.6.4 Statistics

The CTL tracks the statistics listed in Table 3.48. The statistics are read via the Get Statistics Vendor Command. The counters do not rollover and they are cleared on read.

Error conditions are indicated via the RX Status Word, Table 3.74 on page 86, or the TX Status Word, Table 3.78 on page 92.

Table 3.48 Statistics Counters

NAME	DESCRIPTION	SIZE (BITS)
RX Good Frames	Number of good RX frames received. Includes frames dropped by the FCT.	32
RX CRC Errors	Number of RX frames received with CRC-32 errors. **Note:** A CRC error is indicated when the CRC error flag is set and the dribbling bit flag is not set.	20
RX Runt Frame Errors	Number of RX frames received with a length of less than 64 bytes and a CRC error.	20
RX Alignment Errors	Number of RX frames received with alignment errors. **Note:** An alignment error is indicated by the presence of the CRC error flag is set and the dribbling bit flag is set.	20
RX Frame Too Long Error	Number of RX frames received with a length greater than the programmed maximum Ethernet frame size.	20
RX Later Collision Error	Number of RX frames received where a late collision has occurred.	20
RX Bad Frames	Total number of errored Ethernet frames received. This counter does not include RX FIFO Dropped Frames.	20
RX FIFO Dropped Frames	Number of RX frames dropped by the FCT due to insufficient room in the RX FIFO. **Note:** If an RX FIFO dropped frame has an Ethernet error, i.e CRC error, it must only be counted by the RX FIFO Dropped Frames counters.	20
TX Good Frames	Number of successfully transmitted TX frames. **Note:** Does not count pause frames.	32
TX Pause Frames	Number of successfully transmitted pause frames.	20
TX Single Collisions	Number of successfully transmitted frames with one collision.	20
TX Multiple Collisions	Number of successfully transmitted frames with more than one collision.	20
TX Excessive Collision Errors	Number of transmitted frames aborted due to excessive collisions.	20
TX Late Collision Errors	Number of transmitted frames aborted due to late collisions.	20
TX Buffer Underrun Errors	Number of transmitted frames aborted due to Tx buffer under run.	20
TX Excessive Deferral Errors	Number of transmitted frames aborted due to excessive deferrals.	20
TX Carrier Errors	Number of frames transmitted in which the carrier signal was lost or in which the carrier signal was not present.	20
TX Bad Frames	Total number of errored Ethernet frames transmitted.	20

3.6.5 USB Standard Commands

This section lists the formats of the supported USB Standard Commands. The Set Descriptor, Set Interface, and Synch Frame commands are not supported.

3.6.5.1 Clear Feature

This command clears the Stall status of the targeted endpoint or the device remote wakeup.

Table 3.49 Format of Clear Feature Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	Note 3.34
1h	bRequest	01h
2h	wValue	Selects feature to clear.
4h	wIndex	Note 3.35
6h	wLength	00h

Note 3.34 Set to 00h to clear device remote wakeup event. Set to 02h to clear the endpoint stall status.

Note 3.35 When the bmRequestType field specifies an endpoint, the wIndex field selects the endpoint (0, 1, 2, or 3) targeted by the command.

3.6.5.2 Get Configuration

Table 3.50 Format of Clear Feature Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	80h
1h	bRequest	08h
2h	wValue	00h
4h	wIndex	00h
6h	wLength	01h

Table 3.51 Format of Get Configuration Data Stage

OFFSET	FIELD
0h	Returns bConfigurationValue

3.6.5.3 Get Descriptor

Table 3.52 Format for Get Descriptor Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	80h
1h	bRequest	06h
2h	wValue	Note 3.36
4h	wIndex	Note 3.37
6h	wLength	Length of descriptor

Note 3.36 Selects descriptor type. The support descriptors for this command are Device, Configuration, String, Device Qualifier, and Other Speed Configuration.

Note 3.37 Set to zero or Language ID.

Note: The Interface and Endpoint descriptors are not supported by this command. The UDC will stall these requests.

3.6.5.4 Get Interface

Table 3.53 Format of Get Interface Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	81h
1h	bRequest	0Ah
2h	wValue	00h
4h	wIndex	00h
6h	wLength	01h

Table 3.54 Get Interface Data Stage

OFFSET	FIELD
0h	Alternate Setting

Note: The device only supports a single interface.

3.6.5.5 Get Status

3.6.5.5.1 DEVICE STATUS

Table 3.55 Format of Get Status (Device) Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	80h
1h	bRequest	00h
2h	wValue	00h
4h	wIndex	00h
6h	wLength	02h

Table 3.56 Format of Get Status (Device) Data Stage

OFFSET	FIELD
0h	{00h, 0h, 00b, Remote Wakeup, Self Powered}

3.6.5.5.2 ENDPOINT 1 STATUS (BULK IN)

Table 3.57 Format of Get Status (Endpoint 1) Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	82h
1h	bRequest	00h
2h	wValue	00h
4h	wIndex	81h
6h	wLength	02h

Table 3.58 Format of Get Status (Endpoint 1) Data Stage

OFFSET	FIELD
0h	{00h, 0h, 000b, Stall status}

3.6.5.5.3 ENDPOINT 2 STATUS (BULK OUT)

Table 3.59 Format of Get Status (Endpoint 2) Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	82h
1h	bRequest	00h
2h	wValue	00h
4h	wIndex	02h
6h	wLength	02h

Table 3.60 Format of Get Status (Endpoint 2) Data Stage

OFFSET	FIELD
0h	{00h, 0h, 000b, Stall status}

3.6.5.5.4 ENDPOINT 3 STATUS (INTERRUPT)

Table 3.61 Format of Get Status (Endpoint 3) Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	82h
1h	bRequest	00h
2h	wValue	00h
4h	wIndex	83h
6h	wLength	02h

Table 3.62 Format of Get Status (Endpoint 3) Data Stage

OFFSET	FIELD
0h	{00h, 0h, 000b, Stall status}

3.6.5.5.5 SET ADDRESS

Table 3.63 Format of Set Address Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	00h
1h	bRequest	05h
2h	wValue	Device address
4h	wIndex	00h
6h	wLength	00h

3.6.5.5.6 SET FEATURE

This command sets the Stall feature for all supported endpoints. It also supports the Device Remote Wakeup Feature and Test mode.

Table 3.64 Format of Set Feature Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	■ 00h for device ■ 02h for endpoint
1h	bRequest	03h
2h	wValue	■ 01h for DEVICE_REMOTE_WAKEUP ■ 00h for ENDPOINT_HALT ■ 02h for EST_MODE
4h	wIndex	■ 00h for device remote wakeup ■ 00h for TEST_MODE ■ Interface endpoint number for halt
6h	wLength	00h

3.6.5.5.7 SET CONFIGURATION

The device supports only one configuration. An occurrence of this command places the device into the Configured state.

Table 3.65 Format of Set Configuration Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	00h
1h	bRequest	09h
2h	wValue	Configuration Value
4h	wIndex	00h
6h	wLength	00h

Since only one configuration is supported, 01h is the only supported configuration value.

3.6.5.5.8 SET INTERFACE

Only one interface is supported by the device. Therefore, this command is of marginal use. If the command is issued with an alternative setting of 00h and interface setting of 00h, as shown in Table 3.66, the device responds with an ACK. Otherwise it responds with a STALL handshake.

Table 3.66 Format of Set Interface Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	01h
1h	bRequest	0Bh
2h	wValue	00h
4h	wIndex	00h
6h	wLength	00h

3.6.6 USB Vendor Commands

The device implements several vendor specific commands in order to access CSRs and efficiently gather statistics. The vendor commands allow direct access to Systems CSRs and MAC CSRs.

Note: When in the Normal state, accesses to the MAC CSRs are stalled.

3.6.6.1 Register Write Command

The commands allows the Host to write a single register. Burst writes are not supported. All writes are 32-bits.

Table 3.67 Format of Register Write Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	40h
1h	bRequest	A0h
2h	wValue	00h
4h	wIndex	{0h, CSR Address[11:0]}
6h	wLength	04h

Table 3.68 Format of Register Write Data Stage

OFFSET	FIELD
0h	Register Write Data [31:0]

3.6.6.2 Register Read Command

The commands allows the Host to read a single register. Burst reads are not supported. All reads return 32-bits.

Table 3.69 Format of Register Read Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	C0h
1h	bRequest	A1h
2h	wValue	00h
4h	wIndex	{0h, CSR Address[11:0]}
6h	wLength	04h

Table 3.70 Format of Register Read Data Stage

OFFSET	FIELD
0h	Register Read Data [31:0]

3.6.6.3 Get Statistics Command

The Get Statistics Command returns the entire contents of the statistics RAMs. After the command is issued, the contents of the statistics RAM is cleared. The wIndex field is used to select the RX or TX statistics.

Table 3.71 Format of Get Statistics Setup Stage

OFFSET	FIELD	VALUE
0h	bmRequestType	C0h
1h	bRequest	A2h
2h	wValue	00h
4h	wIndex	Note 3.38
6h	wLength	Note 3.39

Note 3.38 0b - Retrieves RX statistics. 1b - Retrieves TX statistics.

Note 3.39 20h for RX statistics. 28h for TX statistics.

Table 3.72 Format of Get Statistics Data Stage (RX)

OFFSET	FIELD
00h	RX Good Frames
04h	RX CRC Errors
08h	RX Runt Frame Errors
0Ch	RX Alignment Errors
10h	RX Frame Too Long Error
14h	RX Later Collision Error
18h	RX Bad Frames
1Ch	RX FIFO Dropped Frames

Table 3.73 Format of Get Statistics Data Stage (TX)

OFFSET	FIELD
00h	TX Good Frames
04h	TX Pause Frames
08h	TX Single Collisions
0Ch	TX Multiple Collisions
10h	TX Excessive Collision Errors
14h	TX Late Collision Errors
18h	TX Buffer Underrun Errors
1Ch	TX Excessive Deferral Errors
20h	TX Carrier Errors
24h	TX Bad Frames

3.7 FIFO Controller (FCT)

The FIFO controller uses a 28 KB internal SRAM to buffer RX and TX traffic. 20 KB is allocated for received Ethernet-USB traffic (RX buffer), while 8 KB is allocated for USB-Ethernet traffic (TX buffer). Bulk-Out packets from the USB controller are directly stored into the TX buffer. The FCT is responsible for extracting Ethernet frames from the USB packet data and passing the frames to the MAC. Ethernet Frames are directly stored into the RX buffer and become the basis for bulk-in packets. The FCT passes the stored data to the UTX in blocks typically 512 or 64 bytes in size, depending on the current HS/FS USB operating speed.

3.7.1 RX Path (Ethernet -> USB)

The 20 KB RX FIFO buffers Ethernet frames received from the TLI. The UTX extracts these frames from the FCT to form USB Bulk In packets. The Host drivers will ultimately reassemble the Ethernet frames from the USB packets.

The FCT manages the writing of data into the RX FIFO through the use of two pointers - the rx_wr_ptr and the rx_wr_hd_ptr. The rx_wr_ptr is used to write Ethernet frame data into the FIFO. The rx_wr_hd_ptr points to the location prior to the first DWORD of the frame. It is used to write the RX Status Word received from the TLI, upon completion of a frame transaction. This status word contains status information associated with the frame and the frame transaction. Figure 3.4 illustrates how a frame is stored in the FIFO, along with pointer usage.

When the RX TLI signals that it has Data ready, the RX TLI controller starts passing the RX packet data to the FCT. The FCT updates the RX FIFO pointers as the data is written into the FIFO. The last transfer from the TLI is the RX Status Word.

The FCT may insert 0 - 3 bytes at the start of the Ethernet frame. The value of the RX Data Offset (RXDOFF) field of the Hardware Configuration Register (HW_CFG) on page 154 determines the number of bytes inserted.

A received Ethernet frame is not visible to the UTX until the complete frame, including the RX Status Word, has been written into the RX FIFO. This is due to the fact that the frame may have to be removed via a rewind (pointer adjustment), in case of an error. Such is the case when a FIFO overflow condition is detected as the frame is being received. The FCT may also be configured to rewind errored frames. Please refer to Section 3.7.1.1, "RX Error Detection," on page 85 for further details.

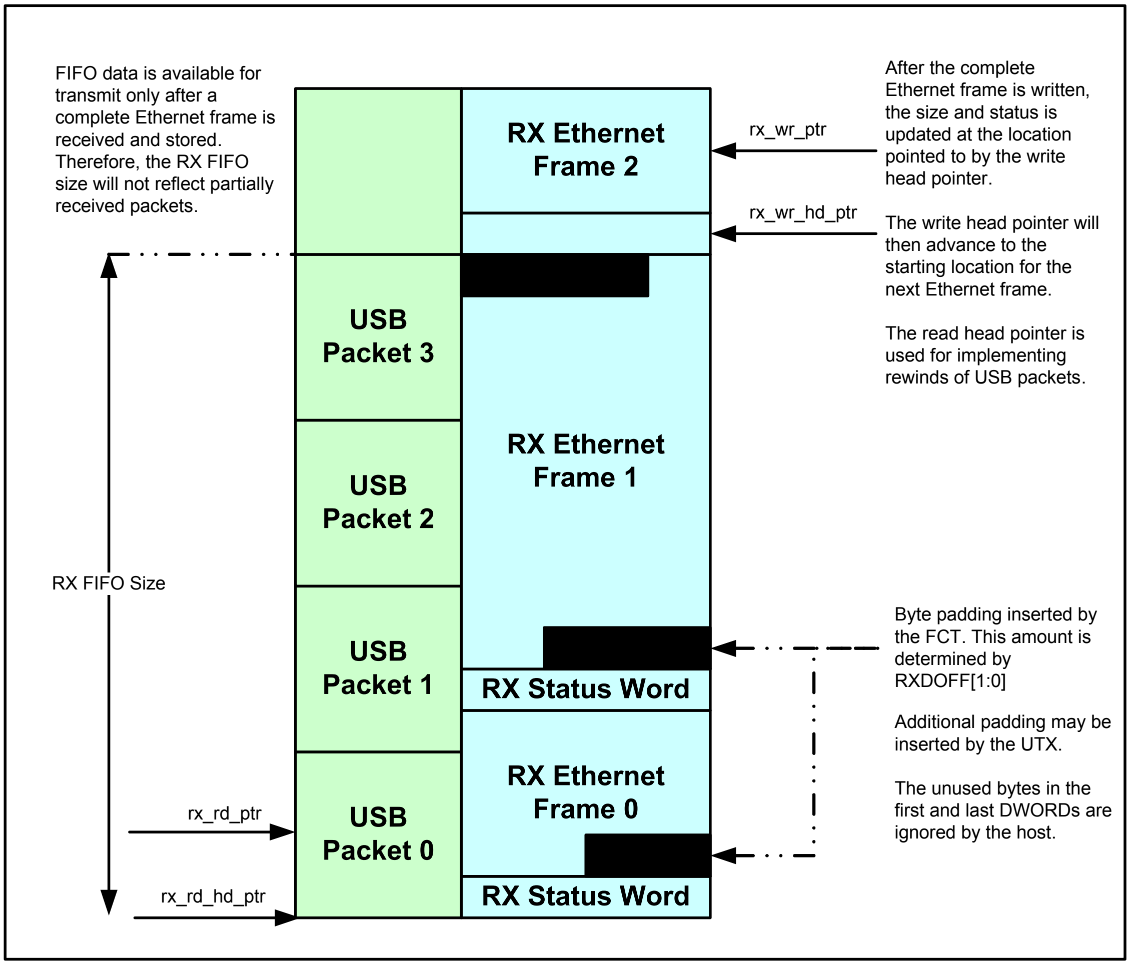

FIFO data is available for transmit only after a complete Ethernet frame is received and stored. Therefore, the RX FIFO size will not reflect partially received packets.
RX Ethernet Frame 2
rx_wr_ptr
After the complete Ethernet frame is written, the size and status is updated at the location pointed to by the write head pointer.
rx_wr_hd_ptr
The write head pointer will then advance to the starting location for the next Ethernet frame.
The read head pointer is used for implementing rewinds of USB packets.
USB Packet 3
RX Ethernet Frame 1
USB Packet 2
RX FIFO Size
USB Packet 1
Byte padding inserted by the FCT. This amount is determined by RXDOFF[1:0]
Additional padding may be inserted by the UTX.
RX Status Word
RX Ethernet Frame 0
The unused bytes in the first and last DWORDs are ignored by the host.
rx_rd_ptr
USB Packet 0
rx_rd_hd_ptr
RX Status Word

Figure 3.4 RX FIFO Storage

3.7.1.1 RX Error Detection

The FCT can be configured to drop Ethernet frames when certain error conditions occur. The setting of the Discard Errored Received Ethernet Frame (DRP) bit of the Hardware Configuration Register (HW_CFG) on page 154 determines if the frame will be retained or dropped. Error conditions are indicated in the Rx Status Word. The following error conditions are tracked by the TLI:

- CRC Error
- Collision Seen
- Frame Too Long
- Runt Frame

Please refer to Section 3.6.4, "Statistics," on page 74 for more details on the error conditions tracked by the device.

The FCT also drops frames when it detects a FIFO overflow condition. This occurs when the FIFO full condition occurs while a frame is being received. The FCT also maintains a count of the number of times a FIFO overflow condition has occurred.

Dropping an Ethernet frame is implemented by rewinding the received frame. A write side rewind is implemented by setting the rx_wr_ptr to be equal to the rx_wr_hd_ptr. Similarly, a read side rewind is implemented by setting the rx_rd_ptr to be equal to the rx_rd_hd_ptr.

For the case where the frame is dropped due to overflow, the FCT ignores the remainder of the frame. It will not begin writing into the RX FIFO again until the next frame is received.

In the read direction, the FCT must also support rewinds for the UTX. This is needed for the case where the USB Bulk Out packet is not successfully received by the Host and needs to be retransmitted.

3.7.1.2 RX Status Format

Table 3.74 illustrates the format of the RX Status Word.

.

Table 3.74 RX Status Word Format

BITS	DESCRIPTION
31	**RESERVED**
30	**Filtering Fail** When set, this bit indicates that the associated frame failed the address recognizing filtering.
29:16	**Frame Length** The size, in bytes, of the corresponding received frame.
15	**Error Status (ES)** When set, this bit indicates that the TLI has reported an error. This bit is the logical OR of bits 11, 7, 6, 1 in this status word.
14	**RESERVED**
13	**Broadcast Frame** When set, this bit indicates that the received frame has a Broadcast address.
12	**Length Error (LE)** When set, this bit indicates that the actual length does not match with the length/type field of the received frame.
11	**Runt Frame** When set, this bit indicates that frame was prematurely terminated before the collision window (64 bytes). Runt frames are passed on to the Host only if the Pass Bad Frames bit MAC_CR Bit [16] is set.
10	**Multicast Frame** When set, this bit indicates that the received frame has a Multicast address.
9:8	**RESERVED**
7	**Frame Too Long** When set, this bit indicates that the frame length exceeds the maximum Ethernet specification of 1518 bytes. This is only a frame too long indication and will not cause the frame reception to be truncated.
6	**Collision Seen** When set, this bit indicates that the frame has seen a collision after the collision window. This indicates that a late collision has occurred.
5	**Frame Type** When set, this bit indicates that the frame is an Ethernet-type frame (Length/Type field in the frame is greater than 1500). When reset, it indicates the incoming frame was an 802.3 type frame. This bit is not set for Runt frames less than 14 bytes.
4	**Receive Watchdog time-out** When set, this bit indicates that the incoming frame is greater than 2048 bytes through 2560 bytes, therefore expiring the Receive Watchdog Timer.

Table 3.74 RX Status Word Format (continued)

BITS	DESCRIPTION
3	**MII Error** When set, this bit indicates that a receive error (RX_ER asserted) was detected during frame reception.
2	**Dribbling Bit** When set, this bit indicates that the frame contained a no-integer multiple of 8 bits. This error is reported only if the number of dribbling bits in the last byte is 4 in the MII operating mode, or at least 3 in the 10 Mbps operating mode. This bit will not be set when the Collision Seen bit[6] is set. If set and the CRC error[1] bit is reset, then the frame is considered to be valid.
1	**CRC Error** When set, this bit indicates that a CRC error was detected. This bit is also set when the RX_ER pin is asserted during the reception of a frame even though the CRC may be correct. This bit is not valid if the received frame is a Runt frame, or a late collision was detected or when the Watchdog Time-out occurs.
0	**RESERVED**

3.7.1.3 Flushing the RX FIFO

The device allows for the Host to the flush the entire contents of the FCT RX FIFO. When a flush is activated, the read and write pointers of the RX FIFO are returned to their reset state.

Before flushing the RX FIFO, the device’s receiver must be stopped, as specified in Section 3.7.1.4. Once the receiver stop completion is confirmed, the Receive FIFO Flush bit can be set in the Receive Configuration Register (RX_CFG) on page 152 to initiate the flush operation. This bit is cleared after the flush is complete.

3.7.1.4 Stopping and Starting the Receiver

To stop the receiver, the Host must clear the Receiver Enable (RXEN) bit in the MAC Control Register (MAC_CR) on page 181. When the receiver is halted, the RXSTOP_INT will be pulsed. Once stopped, the Host can optionally clear the RX Status and RX FIFOs. The Host must re-enable the receiver by setting the RXEN bit.

3.7.2 TX Path (USB -> Ethernet)

The 8 KB TX FIFO buffers USB Bulk Out packets received by the URX. The FCT is responsible for extracting the Ethernet frames embedded in the USB Bulk Out Packets and passing them to the TLI. The Ethernet frames were segmented across the USB packets by the Host drivers.

The FCT manages the writing of data into the TX FIFO through the use of two pointers - the tx_wr_ptr and the tx_wr_hd_ptr. These pointers are used to manage the storing of USB Bulk Out packets. They support rewinding the stored USB packet, in the event that the Bulk Out Packet is errored and needs to be retransmitted by the Host. The write side of the FCT does not perform any processing on the USB packet data. The read side of the TX FIFO is responsible for extracting the Ethernet frames. The Ethernet frames may be split across multiple buffers, as shown in Figure 3.5.

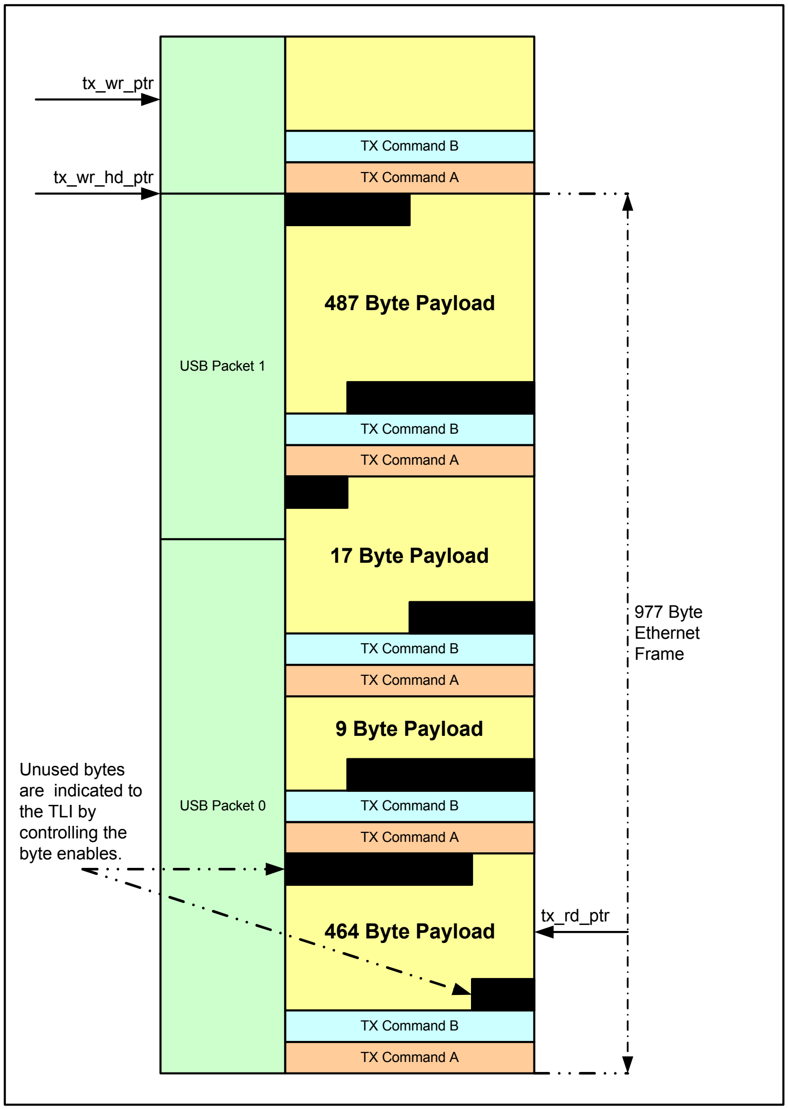


Figure 3.5 TX FIFO Storage

3.7.2.1 TX Command Format

As shown in Figure 3.5, each buffer starts with a two DWORD TX Command. The TX Command instructs the FCT on the handling of the associated buffer. The command precedes the data to be transmitted. The TX command is divided into two, 32-bit words; TX Command A and TX command B.

Both TX command A and TX command B are required for each buffer in a given packet. TX command B must be identical for every buffer in a given packet, with the exception of the TX Checksum Enable (CK) bit. If the TX command B DWORDs do not match, the FCT will assert the Transmitter Error (TXE) flag.

Frame boundaries are delineated using control bits within the TX command. The Frame Length field in TX Command B specifies the number of bytes in the associated frame. All Frame Length fields must have the same value for all buffers in a given Frame. Hardware compares the Frame Length field and the actual amount of data received. If the actual frame length count does not match the Frame Length field, an error has occurred.

The formats of TX Command A and TX Command B are shown in Table 3.75 and Table 3.76, respectively.

Table 3.75 TX Command A Format

BITS	DESCRIPTION
31:18	**RESERVED**
17:16	**Data Start Offset (bytes)** This field specifies the offset of the first byte of TX Data. The offset value ranges between 0 bytes and 3 bytes.
15:14	**RESERVED**
13	**First Segment** When set, this bit indicates that the associated buffer is the first segment of the frame.
12	**Last Segment** When set, this bit indicates that the associated buffer is the last segment of the frame.
11	**RESERVED**
10:0	**Buffer Size (bytes)** This field indicates the number of bytes contained in the buffer following the two command DWORDS (TX Command A and TX Command B). This value, along with the Data Start Offset field, is used by the FCT to determine how many extra bytes were added to the end of the Buffer. A running count is also maintained in the FCT of the cumulative buffer sizes for a given frame. This cumulative value is compared against the Frame Length field in the TX Command B word and if they do not correlate, the TXE flag is set. The buffer size specified does not include bytes added due to the end of buffer alignment padding or the Data Start Offset field.

Table 3.76 TX Command B Format

BITS	DESCRIPTION
31:15	**RESERVED**
14	**TX Checksum Enable (CK)** If this bit is set in conjunction with the first segment bit (FS) in TX Command 'A' and the TX checksum offload engine enable bit (TXCOE_EN) in the checksum offload engine control register (COE_CR), the TX checksum offload engine (TXCOE) will calculate an L3 checksum for the associated frame. **Note:** This bit only needs to be set for the first buffer of a frame.
13	**Add CRC Disable** When set, the automatic addition of the CRC is disabled.
12	**Disable Ethernet Frame Padding** When set, this bit prevents the automatic addition of padding to an Ethernet frame of less than 64 bytes. The CRC field is also added despite the state of the Add CRC Disable field.
11	**RESERVED**
10:0	**Frame Length (bytes)** This field indicates the total number of bytes in the current frame. This length does not include the offset or padding. If the Frame Length field does not match the actual number of bytes in the frame, the Transmitter Error (TXE) flag will be set (in the Interrupt Status Register (INT_STS) and the interrupt endpoint). This value is read by the TX FIFO controller, and is used to determine the amount of data that must be moved from the TX data FIFO into the TLI block. If the byte count is not aligned to a DWORD boundary, the TX FIFO Controller will issue the correct byte enables to the TLI layer during the last write. Invalid bytes in the last DWORD will not be passed to the TLI for transmission.

3.7.2.2 TX Data Format

The TX data section begins at the third DWORD in the TX buffer (after TX Command A and TX Command B). The location of the first byte of valid buffer data to be transmitted is specified in the Data Start Offset field of TX Command A. Table 3.77, "TX Data Start Offset" shows the correlation between the setting of the LSB's in the Data Start Offset field and the byte location of the first valid data byte.

Table 3.77 TX Data Start Offset

DATA START OFFSET[1:0]	11	10	01	00
First TX Data Byte	D[31:24]	D[23:16]	D[15:8]	D[7:0]

TX data is contiguous until the end of the buffer. The buffer may end on a byte boundary. Unused bytes at the end of the packet will not be sent to the TLI for transmission.

3.7.2.3 TX Buffer Fragmentation Rules

Transmit buffers must adhere to the following rules:

- Each buffer may start and end on any arbitrary byte alignment.
- The first buffer of any transmit packet can be any length.
- Middle buffers (i.e., those with First Segment = Last Segment = 0) must be greater than, or equal to 4 bytes in length.
- The final buffer of any transmit packet can be any length.

3.7.2.4 FCT Actions

The FCT performs basic sanity checks on the correctness of the buffer configuration, as described in Section 3.7.2.5, "TX Error Detection," on page 91. Errors in this regard indicate the TX path is out of sync, which is catastrophic and requires a reinitialization of the TX path.

The FCT performs the following steps when extracting an Ethernet frame:

- Strip out TX Command A
- Strip out TX Command B
- Account for the byte offset at the beginning of the frame. Based upon the buffer size and DataStartOffset[1:0] field of TX Command A, the FCT can numerically determine any unused bytes in the first and last word of the buffer. When transferring these respective DWORDs to the TLI, the FCT adjusts the byte enables accordingly.

Note: When a packet is split into multiple buffers, each successive buffer's data payload may begin on any arbitrary byte.

Unlike the write side, the read side of the TX FIFO does not need to support rewinds. Errors are reported via the Transmitter Error (TXE) flag, which is visible to the Host via the Interrupt Endpoint and is also set in theInterrupt Status Register (INT_STS).

3.7.2.5 TX Error Detection

As previously stated, both TX Command A and TX Command B are required for each buffer in a given frame. TX Command B must be identical for every buffer in a given frame, with the exception of the TX Checksum Enable (CK) bit. If the TX Command B words do not match, then the TX path is out of sync and the FCT asserts the Transmitter Error (TXE) flag.

Similarly, the FCT numerically adds up the size of the frame's buffers. If there is a numerical mismatch, the TX path is out of sync and the FCT asserts the Transmitter Error (TXE) flag. The following error conditions are tracked by the FCT:

- Missing FS - The expected first buffer of a frame does not have the FS bit set.
- Unexpected FS - The FS bit is set when the total size of buffers so far opened is less than the frame size.
- Missing LS - The total size of the buffers opened is equal to or exceeds the size of the frame. The FCT expects this buffer to have the LS bit set and it is not set.
- Unexpected LS - The LS bit is set when the aggregate total size of descriptor buffers so far opened is less than the frame size.
- Buffer Size is Zero Error - The buffer length field is zero.
- Buffer Size Error - The total sum of the buffers received is not equal to the frame length.

Note: The FCT can be configured to stall the Bulk Out pipe when a Transmit Error is detected. This is accomplished via the Stall Bulk Out Pipe Disable (SBP) bit of the Hardware Configuration Register (HW_CFG). Please refer to Section 4.3.5, "Hardware Configuration Register (HW_CFG)," on page 154 for further details.

Note: A TX Error is a catastrophic condition. The device should be reset in order to recover from it.

3.7.2.6 TX Status Format

After an Ethernet frame is transmitted, the TLI returns the TX Status Word to the FCT, as illustrated in Table 3.78. The contents of the TX Status Word is used for statistics generation and interrupt status creation. Please refer to Section 3.6.4, "Statistics," on page 74 and Section 4.3.2, "Interrupt Status Register (INT_STS)" for further details.

Table 3.78 TX Status Word Format

BITS	DESCRIPTION
31:16	**RESERVED**
15	**Error Status (ES)** When set, this bit indicates that the TLI has reported an error. This bit is the logical OR of bits 11, 10, 9, 8, 2, 1 in this status word.
14:12	**RESERVED**
11	**Loss of Carrier** When set, this bit indicates the loss of carrier during transmission.
10	**No Carrier** When set, this bit indicates that the carrier signal from the transceiver was not present during transmission.
9	**Late Collision** When set, indicates that the packet transmission was aborted after the collision window of 64 bytes.
8	**Excessive Collisions** When set, this bit indicates that the transmission was aborted after 16 collisions while attempting to transmit the current packet.
7	**RESERVED**
6:3	**Collision Count** This counter indicates the number of collisions that occurred before the packet was transmitted. It is not valid when excessive collisions (bit 8) is also set.
2	**Excessive Deferral** If the deferred bit is set in the control register, the setting of the excessive deferral bit indicates that the transmission has ended because of a deferral of over 24288 bit times during transmission.
1	**Underrun Error** When set, this bit indicates that the transmitter aborted the associated frame because of an underrun condition on the TX Data FIFO. TX Underrun will cause the assertion of the TDFU flag in the Interrupt Status Register (INT_STS) and the interrupt endpoint.
0	**Deferred** When set, this bit indicates that the current packet transmission was deferred.

3.7.2.7 Transmit Examples

3.7.2.7.1 TX EXAMPLE 1

In this example a single, 1064-Byte Ethernet frame will be transmitted. This packet is divided into three buffers. The three buffers are as follows:

Buffer 0:

- 3-Byte “Data Start Offset”
- 499-Bytes of payload data

Buffer 1:

- 0-Byte “Data Start Offset”
- 503-Bytes of payload data

Buffer 2:

- 2-Byte “Data Start Offset”

- 62-Bytes of payload data

Figure 3.6, "TX Example 1" illustrates the TX command structure for this example, and also shows how data is passed to the TX data FIFO.

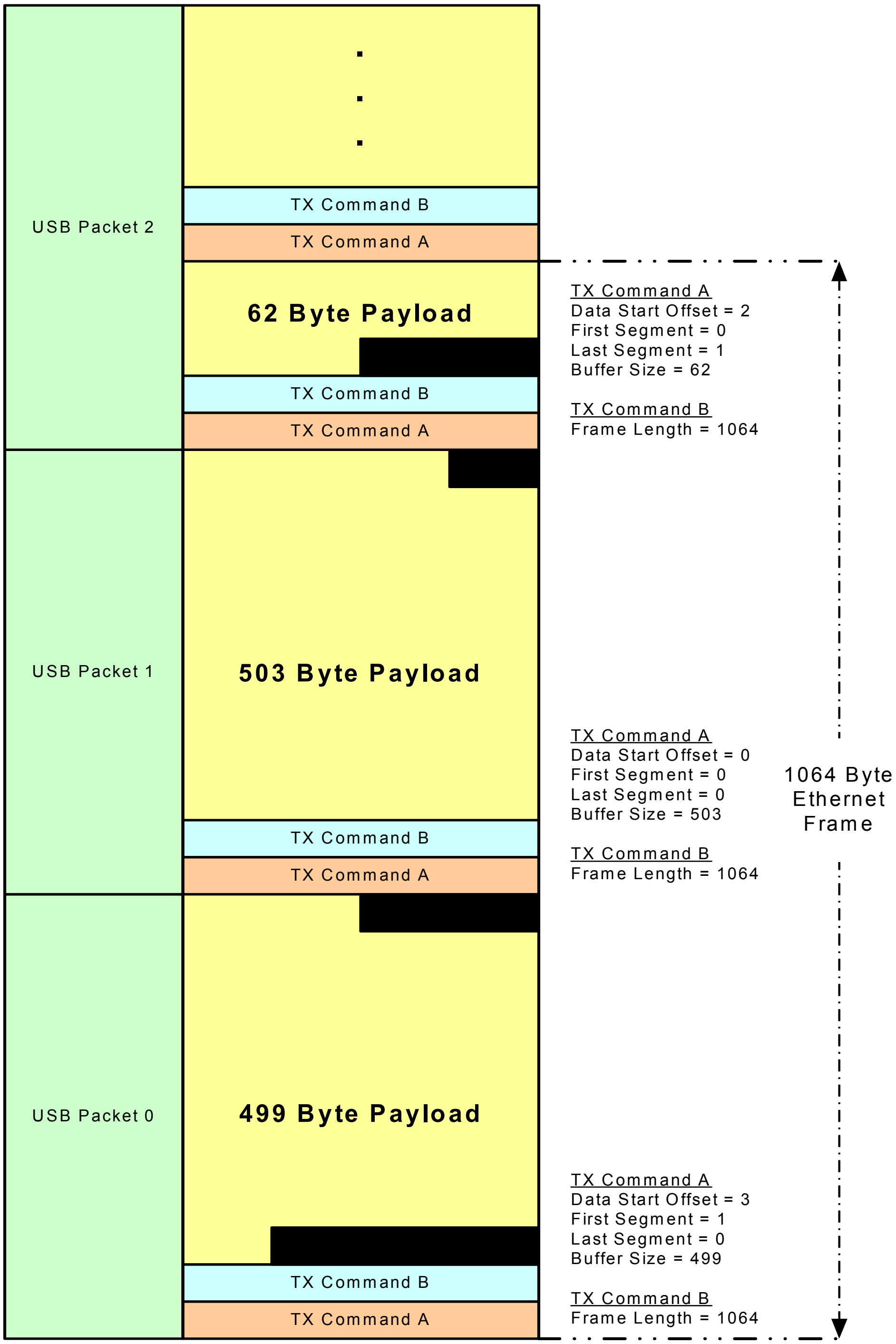


Figure 3.6 TX Example 1

3.7.2.8 TX Example 2

In this example, a single 183-Byte Ethernet frame will be transmitted. This packet is in a single buffer as follows:

- 2-Byte "Data Start Offset"
- 183-Bytes of payload data

Figure 3.7, "TX Example 2" illustrates the TX command structure for this example, and also shows how data is passed to the TX data FIFO. Note that the packet resides in a single TX Buffer, therefore both the FS and LS bits are set in TX Command A.

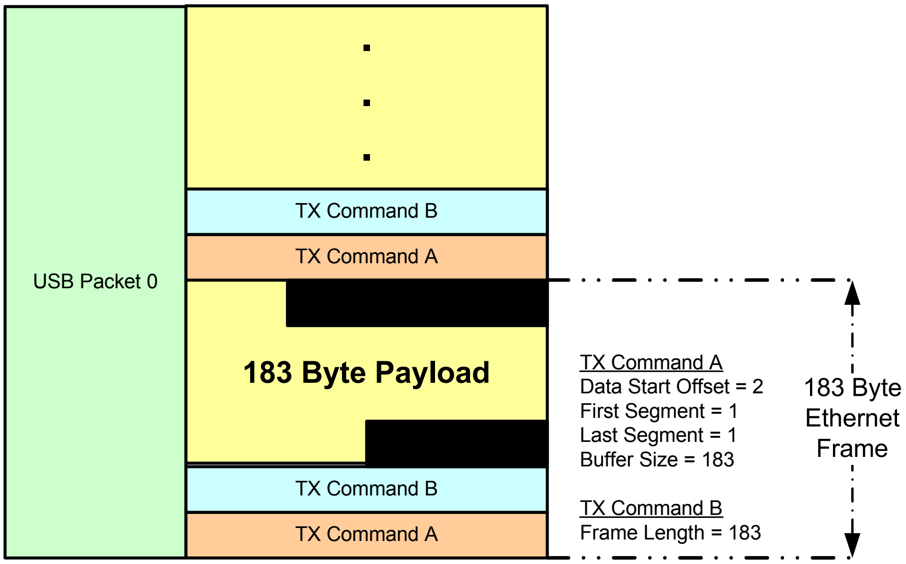


Figure 3.7 TX Example 2

3.7.2.9 TX Example 3

In this example a single, 111-Byte Ethernet frame will be transmitted with a TX checksum. This packet is divided into four buffers. The four buffers are as follows:

Buffer 0:

- 0-Byte “Data Start Offset”
- 4-Byte Checksum Preamble

Buffer 1:

- 3-Byte “Data Start Offset”
- 79-Bytes of payload data

Buffer 2:

- 0-Byte “Data Start Offset”
- 15-Bytes of payload data

Buffer 3:

- 2-Byte “Data Start Offset”
- 17-Bytes of payload data

Figure 3.8, "TX Example 3" illustrates the TX command structure for this example, and also shows how data is passed to the TX data FIFO.

Note: When enabled, the TX Checksum Preamble is pre-pended to the data to be transmitted. The FS bit in TX Command A, the TX Checksum Enable bit (CK) of TX Command B, and the TXCOE_EN bit of the COE_CR register must all be set for the TX checksum to be generated. FS must not be set for subsequent fragments of the same packet. Please refer to Section 3.8.7, "Transmit Checksum Offload Engine (TXCOE)" for further information.

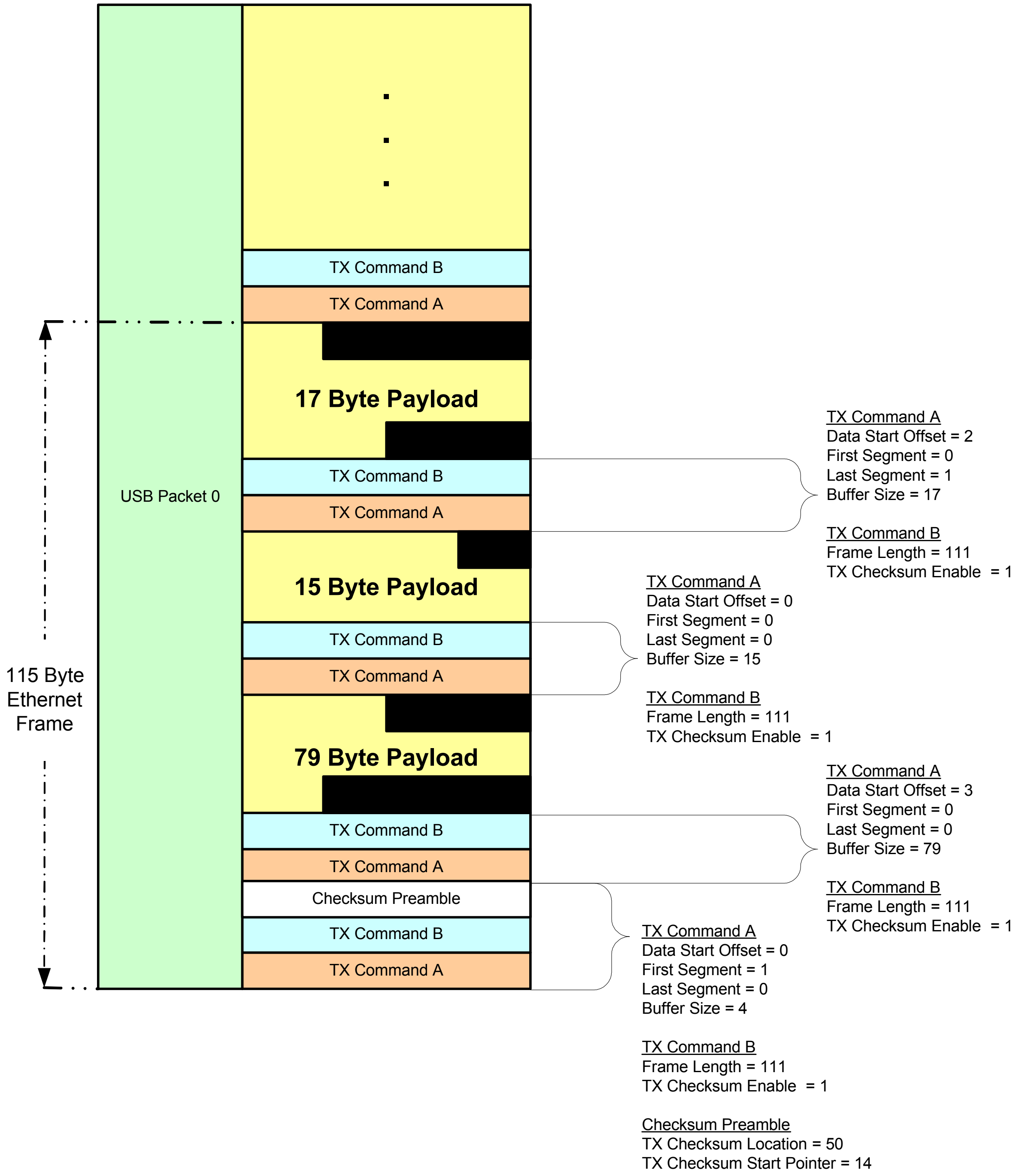


Figure 3.8 TX Example 3

3.7.2.10 Flushing the TX FIFO

The device allows for the Host to the flush the entire contents of the FCT TX FIFO. When a flush is activated, the read and write pointers for the TX FIFO are returned to their reset state.

Before flushing the TX FIFO, the device's transmitter must be stopped, as specified in Section 3.7.2.11. Once the transmitter stop completion is confirmed, the Transmit FIFO Flush bit can be set in the Transmit Configuration Register (TX_CFG) on page 153. This bit is cleared after the flush is complete.

3.7.2.11 Stopping and Starting the Transmitter

To halt the transmitter, the Host must set the Stop Transmitter (STOP_TX) bit in the TX_CFG register. The transmitter will finish sending the current frame (if there is a frame transmission in progress). When the transmitter has received the TX Status for the current frame, it will clear the STOP_TX and TX_ON bits in the TX_CFG register, and will pulse TXSTOP_INT.

Once stopped, the Host can optionally flush the TX FIFO, and can optionally disable the MAC by clearing TXEN. The Host must re-enable the transmitter by setting the TX_ON and TXEN bits. If the there are frames pending in the TX FIFO (i.e., the TX FIFO was not purged), the transmission will resume with this data.

Note: The TX Stop mechanism described here assumes that the MAC will return a status for every TX frame.

3.7.3 Arbitration

The FCT must arbitrate access to the RX and TX FIFOs to the URX, UTX, TLI RX, and TLI TX. Highest priority is always given to the USB. The TLI RX/TX can be wait stated as frame buffering exists in the TLI (2 KB TX, 128 Byte RX).

FCT strict priority order:

1. URX Request (Bulk Out Packet)
2. UTX Request (Bulk In Packet)
3. TLI RX (Received Ethernet Frame)
4. TLI TX (Transmitted Ethernet Frame)

Note: By nature of the USB bus and UDC operation, the URX and UTX should not request bandwidth simultaneously.

3.8 10/100 Ethernet MAC

The Ethernet Media Access controller (MAC) incorporates the essential protocol requirements for operating an Ethernet/IEEE 802.3-compliant node and provides an interface between the Host subsystem and the Ethernet PHY. The MAC can operate in either 100-Mbps or 10-Mbps mode.

The MAC operates in both half-duplex and full-duplex modes. When operating in half-duplex mode, the MAC complies fully with Section 4 of ISO/IEC 8802-3 (ANSI/IEEE standard) and ANSI/IEEE 802.3 standards. When operating in full-duplex mode, the MAC complies with IEEE 802.3x full-duplex operation standard.

The MAC provides programmable enhanced features designed to minimize Host supervision, bus utilization, and pre- or post-message processing. These features include the ability to disable retries after a collision, dynamic FCS (Frame Check Sequence) generation on a frame-by-frame basis, automatic pad field insertion and deletion to enforce minimum frame size attributes, layer 3 checksum calculation for transmit and receive operations, and automatic retransmission and detection of collision frames.

The MAC can sustain transmission or reception of minimally-sized back-to-back packets at full line speed with an interpacket gap (IPG) of 9.6 microseconds for 10 Mbps and 0.96 microseconds for 100 Mbps.

The primary attributes of the MAC Function are:

- Transmit and receive message data encapsulation
- Framing (frame boundary delimitation, frame synchronization)
- Error detection (physical medium transmission errors)
- Media access management
- Medium allocation (collision detection, except in full-duplex operation)
- Contention resolution (collision handling, except in full-duplex operation)
- Flow control during full-duplex mode
- Decoding of control frames (PAUSE command) and disabling the transmitter
- Generation of control frames
- Interface to the Ethernet PHY
- Checksum offload engine for calculation of layer 3 transmit and receive checksum

The transmit and receive data paths are separate within the device from the MAC to Host interface, allowing the highest performance, especially in full duplex mode. Payload data, as well as transmit and receive status, are passed on these busses.

A third internal bus is used to access the MAC’s “Control and Status Registers” (CSR’s). This bus is also accessible from the Host.

On the backend, the MAC interfaces with the 10/100 PHY through an MII (Media Independent Interface) port which is internal to the device. The MAC CSR's also provide a mechanism for accessing the PHY’s internal registers through the internal SMI (Serial Management Interface) bus.

The receive and transmit FIFOs allow increased packet buffer storage to the MAC. The FIFOs are a conduit between the Host interface and the MAC through which all transmitted and received data and status information is passed. Deep FIFOs allow a high degree of latency tolerance relative to the various transport and OS software stacks reducing and minimizing overrun conditions. Like the MAC, the FIFOs have separate receive and transmit data paths.

3.8.1 Flow Control

The device’s Ethernet MAC supports full-duplex flow control using the pause operation and control frame. It also supports half-duplex flow control using back pressure. In order for flow control to be invoked, the Flow Control Enable (FCEN) bit of the Flow Control Register (FLOW) must be set.

3.8.1.1 Full-Duplex Flow Control

The pause operation inhibits data transmission of data frames for a specified period of time. A Pause operation consists of a frame containing the globally assigned multicast address (01-80-C2-00-00-01), the PAUSE opcode, and a parameter indicating the quantum of slot time (512 bit times) to inhibit data transmissions. The PAUSE parameter may range from 0 to 65,535 slot times. The Ethernet MAC logic, on receiving a frame with the reserved multicast address and PAUSE opcode, inhibits data frame transmissions for the length of time indicated. If a Pause request is received while a transmission is in progress, then the pause will take effect after the transmission is complete. Control frames are received and processed by the MAC and are passed on.

The device will automatically transmit pause frames based on the settings of Automatic Flow Control Configuration Register (AFC_CFG) and the Flow Control Register (FLOW). When the RX FIFO reaches the level set in the Automatic Flow Control High Level (AFC_HI) field of AFC_CFG, the device will transmit a pause frame. The pause time field that is transmitted is set in the Pause Time (FCPT) field of the FLOW register. When the RX FIFO drops below the level set in the Automatic Flow Control

Low Level (AFC_LO) field of AFC_CFG, the device will automatically transmit a pause frame with a pause time of zero. The device will only send another pause frame when the RX FIFO level falls below AFC_LO and then exceeds AFC_HI again.

3.8.1.2 Half-Duplex Flow Control (Backpressure)

In half-duplex mode, back pressure is used for flow control. Whenever the RX FIFO crosses a certain threshold level, the MAC starts sending a Jam signal. The MAC transmit logic enters a state at the end of current transmission (if any), where it waits for the beginning of a received frame. Once a new frame starts, the MAC starts sending the jam signal, which will result in a collision. After sensing the collision, the remote station will back off its transmission. The MAC continues sending the jam signal to make other stations defer transmission. The MAC only generates this collision-based back pressure when it receives a new frame, in order to avoid any late collisions.

The device will automatically assert back pressure based on the setting of the Automatic Flow Control Configuration Register (AFC_CFG). When the RX FIFO reaches the level set by Automatic Flow Control High Level (AFC_HI) field of AFC_CFG, the Back pressure Duration Timer will start. The device will assert back pressure for any received frames, as defined by the values of the FCANY, FCADD, FCMULT and FCBRD control bits of AFC_CFG. This continues until the Back pressure Duration Timer reaches the time specified by the BACK_DUR field of AFC_CFG. After the BACK_DUR time period has elapsed, the receiver will accept one frame. If, after receiving one RX frame, the RX FIFO is still above the threshold set in the Automatic Flow Control Low Level (AFC_LO) field of AFC_CFG, the device will again start the Back pressure duration timer and will assert back pressure for subsequent frames, repeating the process described here until the RX Data FIFO level drops below the AFC_LO setting. If the RX FIFO drops below AFC_LO before the Back pressure Duration Timer has expired, the timer will immediately reset and back pressure will not be asserted until the RX FIFO level exceeds AFC_HI.

If the AFC_LO value is set to all ones (0xFF) and the AFC_HI value is set to all zeros (0x00), the flow controller will assert back pressure for received frames as if the AFC_HI threshold is always exceeded. This mechanism can be used to generate software-controlled flow control by enabling and disabling the FCANY, FCADD, FCMULT and FCBRD bits.

3.8.2 Virtual Local Area Network (VLAN) Support

Virtual Local Area Networks or VLANs, as defined within the IEEE 802.3 standard, provide network administrators one means of grouping nodes within a larger network into broadcast domains. To implement a VLAN, four extra bytes are added to the basic Ethernet packet. As shown in Figure 3.9, "VLAN Frame", the four bytes are inserted after the Source Address Field and before the Type/Length field. The first two bytes of the VLAN tag identify the tag, and by convention are set to the value 0x8100. The last two bytes identify the specific VLAN associated with the packet; they also provide a priority field.

The device supports VLAN-tagged packets and provides two registers which are used to identify them. One register should normally be set to the conventional VLAN ID of 0x8100. The other register provides a way of identifying VLAN frames tagged with a proprietary (not 0x8100) identifier. If a packet arrives bearing either of these tags in the two bytes succeeding the Source Address field, the controller will recognize the packet as a VLAN-tagged packet. In this case, the controller increases the maximum allowed packet size from 1518 to 1522 bytes (normally the controller filters packets larger than 1518 bytes). This allows the packet to be received, and then processed by Host software, or to be transmitted on the network.

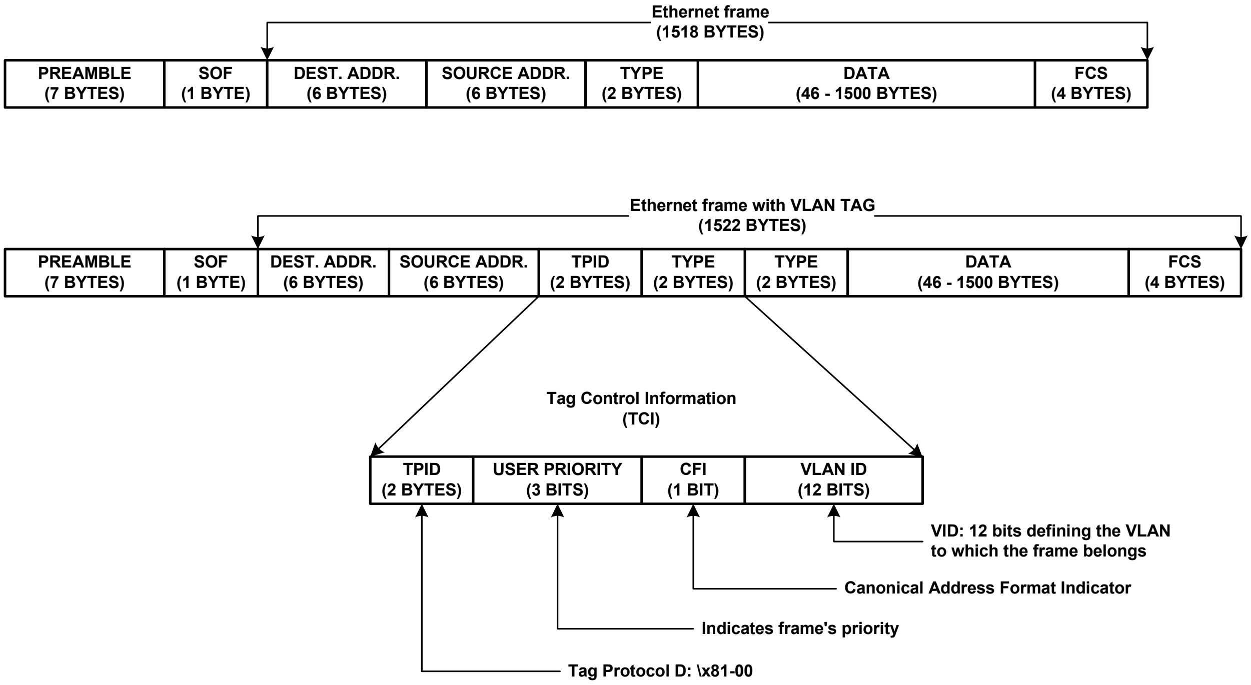


Figure 3.9 VLAN Frame

3.8.3 Address Filtering Functional Description

The Ethernet address fields of an Ethernet Packet, consists of two 6-byte fields: one for the destination address and one for the source address. The first bit of the destination address signifies whether it is a physical address or a multicast address.

The device's address check logic filters the frame based on the Ethernet receive filter mode that has been enabled. Filter modes are specified based on the state of the control bits in Table 3.79, "Address Filtering Modes", which shows the various filtering modes used by the Ethernet MAC Function. These bits are defined in more detail in the "MAC Control Register". Please refer to Section 4.4.1, "MAC Control Register (MAC_CR)," on page 181 for more information on this register.

If the frame fails the filter, the Ethernet MAC function does not receive the packet. The Host has the option of accepting or ignoring the packet.

Table 3.79 Address Filtering Modes

MCPAS	PRMS	INVFILT	HO	HPFILT	DESCRIPTION
0	0	0	0	0	MAC address perfect filtering only for all addresses.
0	0	0	0	1	MAC address perfect filtering for physical address and hash filtering for multicast addresses
0	0	0	1	1	Hash Filtering for physical and multicast addresses

Table 3.79 Address Filtering Modes (continued)

MCPAS	PRMS	INVFILT	HO	HPFILT	DESCRIPTION
0	0	1	0	0	Inverse Filtering
X	1	0	X	X	Promiscuous
1	0	0	0	X	Pass all multicast frames. Frames with physical addresses are perfect-filtered
1	0	0	1	1	Pass all multicast frames. Frames with physical addresses are hash-filtered

3.8.4 Filtering Modes

3.8.4.1 Perfect Filtering

This filtering mode passes only incoming frames whose destination address field exactly matches the value programmed into the MAC Address High register and the MAC address low register. The MAC address is formed by the concatenation of the above two registers in the MAC CSR Function.

3.8.4.2 Hash Only Filtering Mode

This type of filtering checks for incoming Receive packets with either multicast or physical destination addresses, and executes an imperfect address filtering against the hash table.

During imperfect hash filtering, the destination address in the incoming frame is passed through the CRC logic and the upper six bits of the CRC register are used to index the contents of the hash table. The hash table is formed by merging the register's multicast hash table high and multicast hash table low in the MAC CSR Function to form a 64-bit hash table. The most significant bit determines the register to be used (High/Low), while the other five bits determine the bit within the register. A value of 00000 selects Bit 0 of the multicast hash table low register and a value of 11111 selects Bit 31 of the multicast hash table high register.

3.8.4.3 Hash Perfect Filtering

In hash perfect filtering, if the received frame is a physical address, the device's Packet Filter block perfect-filters the incoming frame's destination field with the value programmed into the MAC Address High register and the MAC Address Low register. If the incoming frame is a multicast frame, however, the device's packet filter function performs an imperfect address filtering against the hash table.

The imperfect filtering against the hash table is the same imperfect filtering process described in Section 3.8.4.2, "Hash Only Filtering Mode".

3.8.4.4 Inverse Filtering

In inverse filtering, the Packet Filter Block accepts incoming frames with a destination address not matching the perfect address (i.e., the value programmed into the MAC Address High register and the MAC Address Low register in the CRC block and rejects frames with destination addresses matching the perfect address.

For all filtering modes, when MCPAS is set, all multicast frames are accepted. When the PRMS bit is set, all frames are accepted regardless of their destination address. This includes all broadcast frames as well.

3.8.5 Wakeup Frame Detection

Setting the Wakeup Frame Enable (WUEN) bit in the Wakeup Control and Status Register (WUCSR), places the device's MAC in the wake-up frame detection mode. In this mode, normal data reception is disabled, and detection logic within the MAC examines receive data for the pre-programmed wake-up frame patterns. When a wakeup pattern is received, the Remote Wakeup Frame Received (WUFR) bit in the WUCSR is set, the device places itself in a fully operational state, and remote wakeup is issued. The Host will then resume the device and read the WUSCR register to determine the condition that caused the remote wakeup. Upon determining that the WUFR bit is set, the Host will know a wakeup frame detection event was the cause. The Host will then clear the WUFR bit, and clear the WUEN bit to resume normal receive operation. Please refer to Section 4.4.12, "Wakeup Control and Status Register (WUCSR)," on page 194 for additional information on this register.

Before putting the MAC into the wake-up frame detection state, the Host must provide the detection logic with a list of sample frames and their corresponding byte masks. This information is written into the Wake-up Frame Filter register (WUFF). Please refer to Section 4.4.11, "Wakeup Frame Filter (WUFF)," on page 193 for additional information on this register.

The MAC supports eight programmable filters that support many different receive packet patterns. If remote wake-up mode is enabled, the remote wake-up function receives all frames addressed to the MAC. It then checks each frame against the enabled filter and recognizes the frame as a remote wake-up frame if it passes the WUFF's address filtering and CRC value match.

In order to determine which bytes of the frames should be checked by the CRC module, the MAC uses a programmable byte mask and a programmable pattern offset for each of the eight supported filters.

The pattern's offset defines the location of the first byte that should be checked in the frame. The byte mask is a 128-bit field that specifies whether or not each of the 128 contiguous bytes within the frame, beginning in the pattern offset, should be checked. If bit j in the byte mask is set, the detection logic checks byte offset +j in the frame.

In order to load the Wake-up Frame Filter register, the Host LAN driver software must perform forty writes to the Wake-up Frame Filter register (WUFF). The Diagram shown in Table 3.80, "Wakeup Frame Filter Register Structure" below, shows the wake-up frame filter register's structure.

The contents of the Wakeup Frame Filter register may be obtained by reading it. A total of forty reads is required to extract the entire contents.

Table 3.80 Wakeup Frame Filter Register Structure

Filter 0 Byte Mask 0
Filter 0 Byte Mask 1
Filter 0 Byte Mask 2
Filter 0 Byte Mask 3
Filter 1 Byte Mask 0
Filter 1 Byte Mask 1
Filter 1 Byte Mask 2
Filter 1 Byte Mask 3
Filter 2 Byte Mask 0
Filter 2 Byte Mask 1
Filter 2 Byte Mask 2
Filter 2 Byte Mask 3

Table 3.80 Wakeup Frame Filter Register Structure (continued)

Filter 3 Byte Mask 0							
Filter 3 Byte Mask 1							
Filter 3 Byte Mask 2							
Filter 3 Byte Mask 3							
Filter 4 Byte Mask 0							
Filter 4 Byte Mask 1							
Filter 4 Byte Mask 2							
Filter 4 Byte Mask 3							
Filter 5 Byte Mask 0							
Filter 5 Byte Mask 1							
Filter 5 Byte Mask 2							
Filter 5 Byte Mask 3							
Filter 6 Byte Mask 0							
Filter 6 Byte Mask 1							
Filter 6 Byte Mask 2							
Filter 6 Byte Mask 3							
Filter 7 Byte Mask 0							
Filter 7 Byte Mask 1							
Filter 7 Byte Mask 2							
Filter 7 Byte Mask 3							
Reserved	Filter 3 Command	Reserved	Filter 2 Command	Reserved	Filter 1 Command	Reserved	Filter 0 Command
Reserved	Filter 7 Command	Reserved	Filter 6 Command	Reserved	Filter 5 Command	Reserved	Filter 4 Command
Filter 3 Offset		Filter 2 Offset		Filter 1Offset		Filter 0 Offset	
Filter 7 Offset		Filter 6 Offset		Filter 5 Offset		Filter 4 Offset	
Filter 1 CRC-16				Filter 0 CRC-16			
Filter 3 CRC-16				Filter 2 CRC-16			
Filter 5 CRC-16				Filter 4 CRC-16			
Filter 7 CRC-16				Filter 6 CRC-16			

The Filter i Byte Mask defines which incoming frame bytes Filter i will examine to determine whether or not this is a Wakeup Frame. Table 3.81, describes the byte mask's bit fields.

Filter x Mask 0 corresponds to bits [31:0]. Where the lsb corresponds to the first byte on the wire.

Filter x Mask 1 corresponds to bits [63:32]. Where the lsb corresponds to the first byte on the wire.

Filter x Mask 2 corresponds to bits [95:64]. Where the lsb corresponds to the first byte on the wire.

Filter x Mask 3 corresponds to bits [127:96]. Where the lsb corresponds to the first byte on the wire.

Table 3.81 Filter i Byte Mask Bit Definitions

FILTER I BYTE MASK DESCRIPTION	
BITS	**DESCRIPTION**
127:0	**Byte Mask:** If bit j of the byte mask is set, the CRC machine processes byte *pattern-offset* + *j* of the incoming frame. Otherwise, byte *pattern-offset* + *j* is ignored.

The Filter i command register controls Filter i operation. Table 3.82 shows the Filter I command register.

Table 3.82 Filter i Command Bit Definitions

FILTER i COMMANDS	
BITS	**DESCRIPTION**
3:2	**Address Type:** Defines the destination address type of the pattern. 00 = Pattern applies only to unicast frames. 10 = Pattern applies only to multicast frames. X1 = Pattern applies to all frames that have passed the regular receive filter.
1	**RESERVED**
0	**Enable Filter:** When bit is set, Filter i is enabled, otherwise, Filter i is disabled.

The Filter i Offset register defines the offset in the frame's destination address field from which the frames are examined by Filter i. Table 3.83 describes the Filter i Offset bit fields.

Table 3.83 Filter i Offset Bit Definitions

FILTER I OFFSET DESCRIPTION	
BITS	**DESCRIPTION**
7:0	**Pattern Offset:** The offset of the first byte in the frame on which CRC is checked for Wakeup Frame recognition. The MAC checks the first offset byte of the frame for CRC and checks to determine whether the frame is a Wakeup Frame. Offset 0 is the first byte of the incoming frame's destination address.

The Filter i CRC-16 register contains the CRC-16 result of the frame that should pass Filter i.

Table 3.84 describes the Filter i CRC-16 bit fields.

Table 3.84 Filter i CRC-16 Bit Definitions

FILTER I CRC-16 DESCRIPTION	
BITS	**DESCRIPTION**
15:0	**Pattern CRC-16:** This field contains the 16-bit CRC value from the pattern and the byte mask programmed to the Wakeup Filter register function. This value is compared against the CRC calculated on the incoming frame, and a match indicates the reception of a Wakeup Frame.

Table 3.85 indicates the cases that produce a wake when the Wakeup Frame Enable (WUEN) bit of the Wakeup Control and Status Register (WUCSR) is set. All other cases do not generate a wake.

Table 3.85 Wakeup Generation Cases

FILTER ENABLED (Note 3.40)	CRC MATCH (Note 3.41)	GLOBAL UNICAST ENABLED (Note 3.42)	PASS REGULAR RECEIVE FILTER	ADDRESS TYPE (Note 3.43)	BROAD-CAST FRAME (Note 3.44)	MULTI-CAST FRAME	UNICAST FRAME
Yes	Yes	x	x	x	Yes	No	No
Yes	Yes	Yes	x	x	No	No	Yes
Yes	Yes	x	Yes	Multicast (=10)	No	Yes	No
Yes	Yes	x	Yes	Unicast (=00)	No	No	Yes
Yes	Yes	x	Yes	Passed Receive Filter (=x1b)	x	x	x

Note 3.40 As determined by bit 0 of Filter i Command.

Note 3.41 CRC matches Filter i CRC-16 field.

Note 3.42 As determined by bit 9 of WUCSR.

Note 3.43 As determined by bits 3:2 of Filter i Command.

Note 3.44 When wakeup frame detection is enabled via the Wakeup Frame Enable (WUEN) bit of the Wakeup Control and Status Register (WUCSR), a broadcast wakeup frame will wake up the device despite the state of the Disable Broadcast Frames (BCAST) bit in the MAC Control Register (MAC_CR).

Note: x indicates "don't care".

3.8.5.1 Magic Packet Detection

Setting the Magic Packet Enable (MPEN) bit in the Wakeup Control and Status Register (WUCSR), places the device's MAC in the "Magic Packet" detection mode. In this mode, normal data reception is disabled, and detection logic within the MAC examines receive data for a Magic Packet. When a Magic Packet is received, the Magic Packet Received (MPR) bit in the WUCSR is set, the device places itself in a fully operational state, and remote wakeup is issued. The Host will then resume the device and read the WUSCR register to determine the condition that caused the remote wakeup. Upon determining that the MPR bit is set, the Host will know reception of a Magic Packet was the cause. The Host will then clear the MPR bit, and clear the WUEN bit to resume normal receive operation. Please refer to Section 4.4.12, "Wakeup Control and Status Register (WUCSR)," on page 194 for additional information on this register.

In Magic Packet mode, the Power Management Logic constantly monitors each frame addressed to the node for a specific Magic Packet pattern. It checks only packets with the MAC's address or a broadcast address to meet the Magic Packet requirement. The Power Management Logic checks each received frame for the pattern 48h FF_FF_FF_FF_FF_FF after the destination and source address field.

Then the Function looks in the frame for 16 repetitions of the MAC address without any breaks or interruptions. In case of a break in the 16 address repetitions, the PMT Function scans for the 48'hFF_FF_FF_FF_FF_FF pattern again in the incoming frame.

The 16 repetitions may be anywhere in the frame but must be preceded by the synchronization stream. The device will also accept a multicast frame, as long as it detects the 16 duplications of the MAC address. If the MAC address of a node is 00h 11h 22h 33h 44h 55h, then the MAC scans for the following data sequence in an Ethernet: Frame.

Destination Address Source Address ....FF FF FF FF FF FF

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

...CRC

3.8.6 Receive Checksum Offload Engine (RXCOE)

The receive checksum offload engine provides assistance to the Host by calculating a 16-bit checksum for a received Ethernet frame. The RXCOE readily supports the following IEEE802.3 frame formats:

- Type II Ethernet frames
- SNAP encapsulated frames
- Support for up to 2, 802.1q VLAN tags

The resulting checksum value can also be modified by software to support other frame formats.

The RXCOE has two modes of operation. In mode 0, the RXCOE calculates the checksum between the first 14 bytes of the Ethernet frame and the FCS. This is illustrated in Figure 3.10.

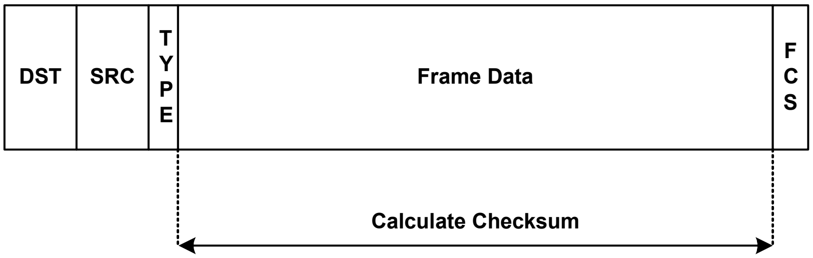


Figure 3.10 RXCOE Checksum Calculation

In mode 1, the RXCOE supports VLAN tags and a SNAP header. In this mode, the RXCOE calculates the checksum at the start of L3 packet. The VLAN1 tag register is used by the RXCOE to indicate what protocol type is to be used to indicate the existence of a VLAN tag. This value is typically 8100h.

Example frame configurations:

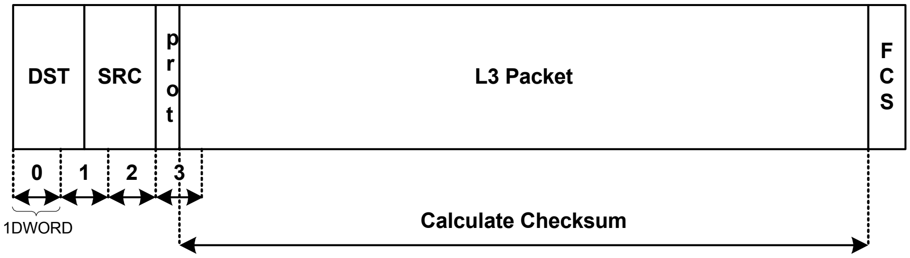


Figure 3.11 Type II Ethernet Frame

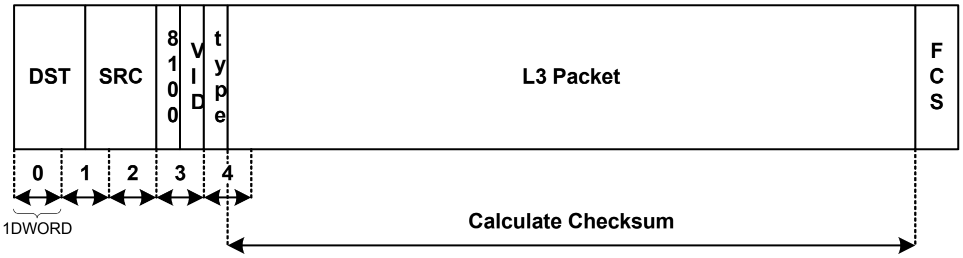


Figure 3.12 Ethernet Frame with VLAN Tag

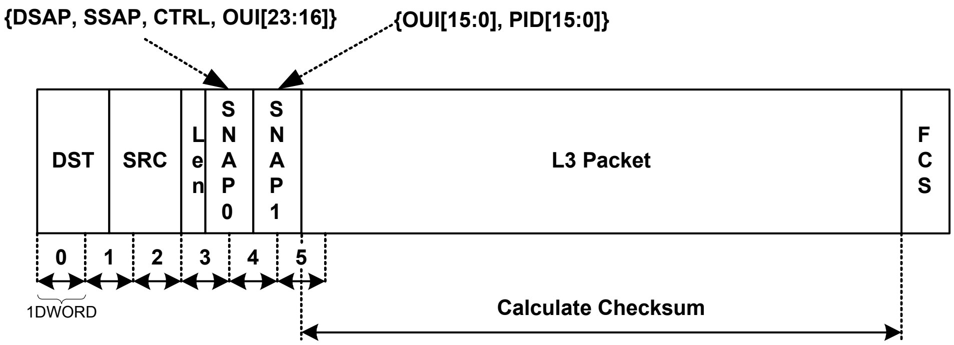


Figure 3.13 Ethernet Frame with Length Field and SNAP Header

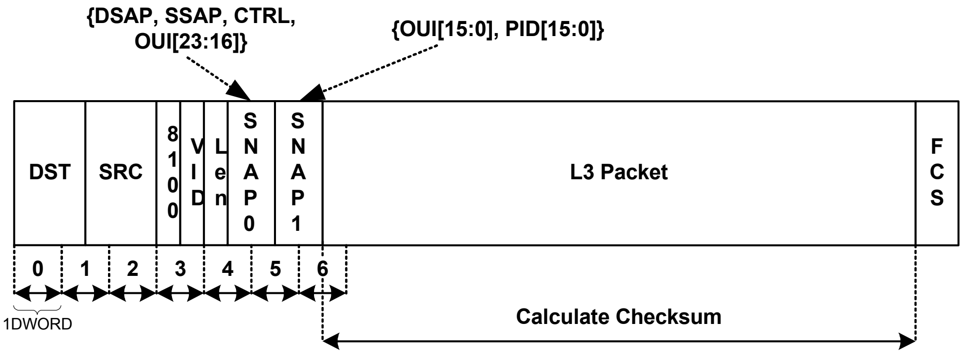


Figure 3.14 Ethernet Frame with VLAN Tag and SNAP Header

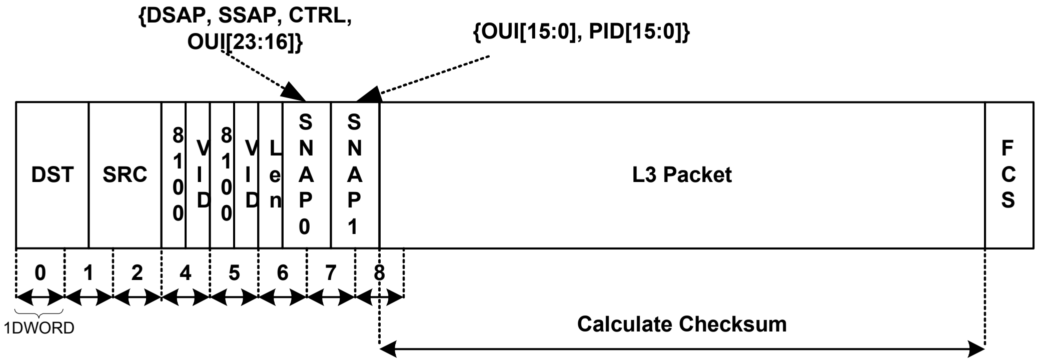


Figure 3.15 Ethernet Frame with multiple VLAN Tags and SNAP Header

The RXCOE supports a maximum of two VLAN tags. If there are more than two VLAN tags, the VLAN protocol identifier for the third tag is treated as an Ethernet type field. The checksum calculation will begin immediately after the type field.

The RXCOE resides in the RX path within the MAC. As the RXCOE receives an Ethernet frame, it calculates the 16-bit checksum. The RXCOE passes the Ethernet frame to the RX FIFO with the checksum appended to the end of the frame. The RXCOE inserts the checksum immediately after the last byte of the Ethernet frame and before it transmits the status word. The packet length field in the RX Status Word (refer to Section 3.7.1.2) will indicate that the frame size has increased by two bytes to accommodate the checksum.

Note: When enabled, the RXCOE calculates a checksum for every received frame.

Setting the RXCOE_EN bit in the Checksum Offload Engine Control Register (COE_CR) enables the RXCOE, while the RXCOE_MODE bit selects the operating mode. When the RXCOE is disabled, the the received data is simply passed through the RXCOE unmodified.

Note: Software applications must stop the receiver and flush the RX data path before changing the state of the RXCOE_EN or RXCOE_MODE bits.

Note: When the RXCOE is enabled, automatic pad stripping must be disabled (bit 8 (PADSTR) of the MAC Control Register (MAC_CR)) and vice versa. These functions cannot be enabled simultaneously.

3.8.6.1 RX Checksum Calculation

The checksum is calculated 16 bits at a time. In the case of an odd sized frame, an extra byte of zero is used to pad up to 16 bits.

Consider the following packet: DA, SA, Type, B0, B1, B2 … BN, FCS

Let [A, B] = A*256 + B;

If the packet has an even number of octets then

checksum = [B1, B0] + C0 + [B3, B2] + C1 + … + [BN, BN-1] + CN-1

Where C0, C1, ... CN-1 are the carry out results of the intermediate sums.

If the packet has an odd number of octets then

checksum = [B1, B0] + C0 + [B3, B2] + C1 + … + [0, BN] + CN-1

3.8.7 Transmit Checksum Offload Engine (TXCOE)

The transmit checksum offload engine provides assistance to the CPU by calculating a 16-bit checksum, typically for TCP, for a transmit Ethernet frame. The TXCOE calculates the checksum and inserts the results back into the data stream as it is transferred to the MAC.

To activate the TXCOE and perform a checksum calculation, the Host must first set the TX Checksum Offload Engine Enable (TX_COE_EN) bit in the Checksum Offload Engine Control Register (COE_CR). The Host then pre-pends a 3 DWORD buffer to the data that will be transmitted. The pre-pended buffer includes a TX Command A, TX Command B, and a 32-bit TX checksum preamble (refer to Table 3.86). When the CK bit of the TX Command ‘B’ is set in conjunction with the FS bit of TX Command ‘A’ and the TX_COE_EN bit of the COE_CR register, the TXCOE will perform a checksum calculation on the associated packet. The TX checksum preamble instructs the TXCOE on the handling of the associated packet. The TXCSSP - TX Checksum Start Pointer field of the TX checksum preamble defines the byte offset at which the data checksum calculation will begin. The checksum calculation will begin at this offset and will continue until the end of the packet. The data checksum calculation must not begin in the MAC header (first 14 bytes) or in the last 4 bytes of the TX packet. When the calculation is complete, the checksum will be inserted into the packet at the byte offset defined by the TXCSLOC - TX Checksum Location field of the TX checksum preamble. The TX checksum cannot be inserted in the MAC header (first 14 bytes) or in the last 4 bytes of the TX packet. If the CK bit is not set in the first TX Command ‘B’ of a packet, the packet is passed directly through the TXCOE without modification, regardless if the TXCOE_EN is set. An example of a TX packet with a pre-pended TX checksum preamble can be found in Section 3.7.2.9, "TX Example 3". In this example, the Host provides the Ethernet frame to the ethernet controller (via a USB packet) in four fragments, the first containing the TX Checksum Preamble. Figure 3.8 shows how these fragments are loaded into the TX Data FIFO. For more information on the TX Command ‘A’ and TX Command ‘B’, refer to Section 3.7.2.1, "TX Command Format," on page 89.

If the TX packet already includes a partial checksum calculation (perhaps inserted by an upper layer protocol), this checksum can be included in the hardware checksum calculation by setting the TXCSSP field in the TX checksum preamble to include the partial checksum. The partial checksum can be replaced by the completed checksum calculation by setting the TXCSLOC pointer to point to the location of the partial checksum.

Table 3.86 TX Checksum Preamble

FIELD	DESCRIPTION
31:28	**RESERVED**
27:16	**TXCSLOC - TX Checksum Location** This field specifies the byte offset where the TX checksum will be inserted in the TX packet. The checksum will replace two bytes of data starting at this offset. **Note:** The TX checksum cannot be inserted in the MAC header (first 14 bytes) or in the last 4 bytes of the TX packet.
15:12	**RESERVED**
11:0	**TXCSSP - TX Checksum Start Pointer** This field indicates start offset, in bytes, where the checksum calculation will begin in the associated TX packet. **Note:** The data checksum calculation must not begin in the MAC header (first 14 bytes) or in the last 4 bytes of the TX packet.

Note: When the TXCOE is enabled, the third DWORD of the pre-pended packet is not transmitted. However, 4 bytes must be added to the packet length field in TX Command B.

Note: Software applications must stop the transmitter and flush the TX data path before changing the state of the TXCOE_EN bit. However, the CK bit of TX Command B can be set or cleared on a per-packet basis.

Note: The TXCOE_MODE may only be changed if the TX path is disabled. If it is desired to change this value during run time, it is safe to do so only after the TX Ethernet path is disabled and the TLI is empty.

Note: The TX checksum preamble must be DWORD-aligned.

Note: TX preamble size is accounted for in both the buffer length and packet length.

Note: The first buffer, which contains the TX preamble, may not contain any Ethernet frame data

Figure 3.16 on page 111 illustrates the use of a pre-pended checksum preamble when transmitting an Ethernet frame consisting of 3 payload buffers.

NOTE: The TX Checksum Preamble is pre-pended to data to be transmitted. FS is set in TX Command 'A' and CK is set in TX Command 'B'. No start offset may be added. FS must not be set for subsequent fragments of the same packet.

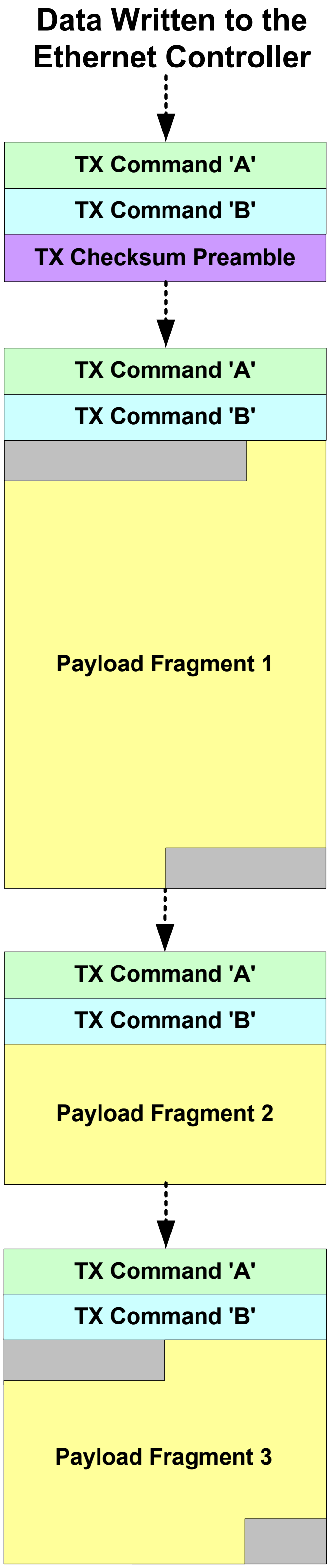


Figure 3.16 TX Example Illustrating a Pre-pended TX Checksum Preamble

3.8.7.1 TX Checksum Calculation

The TX checksum calculation is performed using the same operation as the RX checksum shown in Section 3.8.6.1, with the exception that the calculation starts as indicated by the preamble, and the transmitted checksum is the one's-compliment of the final calculation.

Note: When the TX checksum offload feature is invoked, if the calculated checksum is 0000h, it is left unaltered. UDP checksums are optional under IPv4, and a zero checksum calculated by the TX checksum offload feature will erroneously indicate to the receiver that no checksum was calculated, however, the packet will typically not be rejected by the receiver. Under IPv6, however, according to RFC 2460, the UDP checksum is not optional. A calculated checksum that yields a result of zero must be changed to FFFFh for insertion into the UDP header. IPv6 receivers discard UDP packets containing a zero checksum. **Thus, this feature must not be used for UDP checksum calculation under IPv6.**

3.8.8 MAC Control and Status Registers (MCSR)

Please refer to Section 4.4, "MAC Control and Status Registers," on page 180 for a complete description of the MCSR.

3.9 10/100 Ethernet PHY

The device integrates an IEEE 802.3 Physical Layer for Twisted Pair Ethernet applications. The PHY can be configured for either 100 Mbps (100Base-TX) or 10 Mbps (10Base-T) Ethernet operation in either Full or Half Duplex configurations. The PHY block includes auto-negotiation. Minimal external components are required for the utilization of the PHY.

Functionally, the PHY can be divided into the following sections:

- 100Base-TX transmit and receive
- 10Base-T transmit and receive
- Internal MII interface to the Ethernet Media Access Controller
- Auto-negotiation to automatically determine the best speed and duplex possible
- Management Control to read status registers and write control registers

3.9.1 100BASE-TX Transmit

The data path of the 100Base-TX is shown in Figure 3.17. Each major block is explained in the following sections.

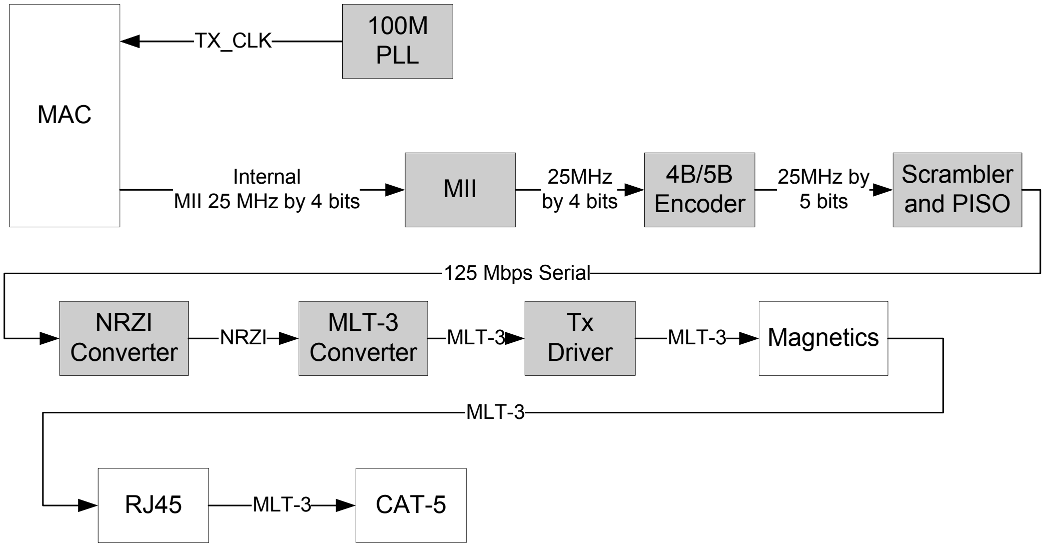


Figure 3.17 100Base-TX Data Path

3.9.1.1 4B/5B Encoding

The transmit data passes from the MII block to the 4B/5B encoder. This block encodes the data from 4-bit nibbles to 5-bit symbols (known as “code-groups”) according to Table 3.87. Each 4-bit data-nibble is mapped to 16 of the 32 possible code-groups. The remaining 16 code-groups are either used for control information or are not valid.

The first 16 code-groups are referred to by the hexadecimal values of their corresponding data nibbles, 0 through F. The remaining code-groups are given letter designations with slashes on either side. For example, an IDLE code-group is /I/, a transmit error code-group is /H/, etc.

The encoding process may be bypassed by clearing bit 6 of register 31. When the encoding is bypassed the 5th transmit data bit is equivalent to TX_ER.

Table 3.87 4B/5B Code Table

CODE GROUP	SYM	RECEIVER INTERPRETATION			TRANSMITTER INTERPRETATION		
11110	0	0	0000	DATA	0	0000	DATA
01001	1	1	0001		1	0001	
10100	2	2	0010		2	0010	
10101	3	3	0011		3	0011	
01010	4	4	0100		4	0100	
01011	5	5	0101		5	0101	
01110	6	6	0110		6	0110	
01111	7	7	0111		7	0111	
10010	8	8	1000		8	1000	
10011	9	9	1001		9	1001	
10110	A	A	1010		A	1010	

Table 3.87 4B/5B Code Table (continued)

CODE GROUP	SYM	RECEIVER INTERPRETATION			TRANSMITTER INTERPRETATION		
10111	B	B	1011		B	1011	
11010	C	C	1100		C	1100	
11011	D	D	1101		D	1101	
11100	E	E	1110		E	1110	
11101	F	F	1111		F	1111	
11111	I	IDLE			Sent after /T/R until TX_EN		
11000	J	First nibble of SSD, translated to “0101” following IDLE, else RX_ER			Sent for rising TX_EN		
10001	K	Second nibble of SSD, translated to “0101” following J, else RX_ER			Sent for rising TX_EN		
01101	T	First nibble of ESD, causes de-assertion of CRS if followed by /R/, else assertion of RX_ER			Sent for falling TX_EN		
00111	R	Second nibble of ESD, causes deassertion of CRS if following /T/, else assertion of RX_ER			Sent for falling TX_EN		
00100	H	Transmit Error Symbol			Sent for rising TX_ER		
00110	V	INVALID, RX_ER if during RX_DV			INVALID		
11001	V	INVALID, RX_ER if during RX_DV			INVALID		
00000	V	INVALID, RX_ER if during RX_DV			INVALID		
00001	V	INVALID, RX_ER if during RX_DV			INVALID		
00010	V	INVALID, RX_ER if during RX_DV			INVALID		
00011	V	INVALID, RX_ER if during RX_DV			INVALID		
00101	V	INVALID, RX_ER if during RX_DV			INVALID		
01000	V	INVALID, RX_ER if during RX_DV			INVALID		
01100	V	INVALID, RX_ER if during RX_DV			INVALID		
10000	V	INVALID, RX_ER if during RX_DV			INVALID		

3.9.1.2 Scrambling

Repeated data patterns (especially the IDLE code-group) can have power spectral densities with large narrow-band peaks. Scrambling the data helps eliminate these peaks and spread the signal power more uniformly over the entire channel bandwidth. This uniform spectral density is required by FCC regulations to prevent excessive EMI from being radiated by the physical wiring.

The scrambler also performs the Parallel In Serial Out conversion (PISO) of the data.

3.9.1.3 NRZI and MLT3 Encoding

The scrambler block passes the 5-bit wide parallel data to the NRZI converter where it becomes a serial 125MHz NRZI data stream. The NRZI is encoded to MLT-3. MLT3 is a tri-level code where a change in the logic level represents a code bit “1” and the logic output remaining at the same level represents a code bit “0”.

3.9.1.4 100M Transmit Driver

The MLT3 data is then passed to the analog transmitter, which launches the differential MLT-3 signal, on outputs TXP and TXN, to the twisted pair media via a 1:1 ratio isolation transformer. The 10Base-T and 100Base-TX signals pass through the same transformer so that common “magnetics” can be used for both. The transmitter drives into the 100Ω impedance of the CAT-5 cable. Cable termination and impedance matching require external components.

3.9.1.5 100M Phase Lock Loop (PLL)

The 100M PLL locks onto reference clock and generates the 125MHz clock used to drive the 125 MHz logic and the 100Base-Tx Transmitter.

3.9.2 100BASE-TX Receive

The receive data path is shown in Figure 3.18. Detailed descriptions are given in the following subsections.

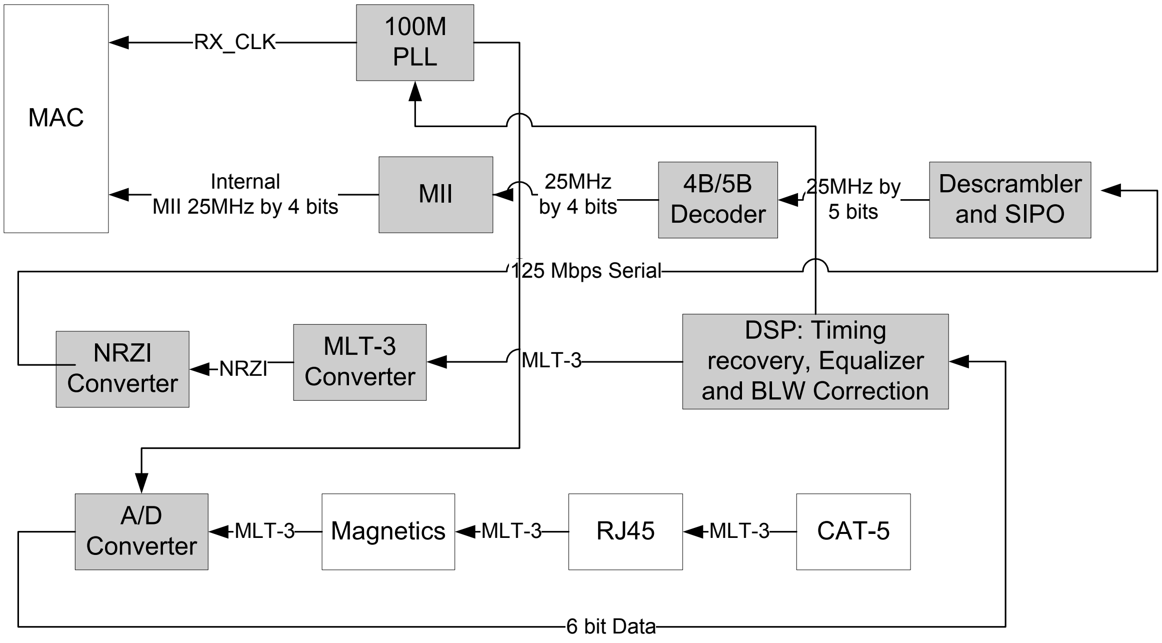


Figure 3.18 Receive Data Path

3.9.2.1 100M Receive Input

The MLT-3 from the cable is fed into the PHY (on inputs RXP and RXN) via a 1:1 ratio transformer. The ADC samples the incoming differential signal at a rate of 125M samples per second. Using a 64-level quanitizer, it generates 6 digital bits to represent each sample. The DSP adjusts the gain of the ADC according to the observed signal levels such that the full dynamic range of the ADC can be used

3.9.2.2 Equalizer, Baseline Wander Correction and Clock and Data Recovery

The 6 bits from the ADC are fed into the DSP block. The equalizer in the DSP section compensates for phase and amplitude distortion caused by the physical channel consisting of magnetics, connectors, and CAT- 5 cable. The equalizer can restore the signal for any good-quality CAT-5 cable between 1m and 150m.

If the DC content of the signal is such that the low-frequency components fall below the low frequency pole of the isolation transformer, then the droop characteristics of the transformer will become significant and Baseline Wander (BLW) on the received signal will result. To prevent corruption of the received data, the PHY corrects for BLW and can receive the ANSI X3.263-1995 FDDI TP-PMD defined “killer packet” with no bit errors.

The 100M PLL generates multiple phases of the 125MHz clock. A multiplexer, controlled by the timing unit of the DSP, selects the optimum phase for sampling the data. This is used as the received recovered clock. This clock is used to extract the serial data from the received signal.

3.9.2.3 NRZI and MLT-3 Decoding

The DSP generates the MLT-3 recovered levels that are fed to the MLT-3 converter. The MLT-3 is then converted to an NRZI data stream.

3.9.2.4 Descrambling

The descrambler performs an inverse function to the scrambler in the transmitter and also performs the Serial In Parallel Out (SIPO) conversion of the data.

During reception of IDLE (/I/) symbols. the descrambler synchronizes its descrambler key to the incoming stream. Once synchronization is achieved, the descrambler locks on this key and is able to descramble incoming data.

Special logic in the descrambler ensures synchronization with the remote PHY by searching for IDLE symbols within a window of 4000 bytes (40us). This window ensures that a maximum packet size of 1514 bytes, allowed by the IEEE 802.3 standard, can be received with no interference. If no IDLE-symbols are detected within this time-period, receive operation is aborted and the descrambler re-starts the synchronization process.

The descrambler can be bypassed by setting bit 0 of register 31.

3.9.2.5 Alignment

The de-scrambled signal is then aligned into 5-bit code-groups by recognizing the /J/K/ Start-of-Stream Delimiter (SSD) pair at the start of a packet. Once the code-word alignment is determined, it is stored and utilized until the next start of frame.

3.9.2.6 5B/4B Decoding

The 5-bit code-groups are translated into 4-bit data nibbles according to the 4B/5B table. The SSD, /J/K/, is translated to “0101 0101” as the first 2 nibbles of the MAC preamble. Reception of the SSD causes the PHY to assert the internal RX_DV signal, indicating that valid data is available on the Internal RXD bus. Successive valid code-groups are translated to data nibbles. Reception of either the End of Stream Delimiter (ESD) consisting of the /T/R/ symbols, or at least two /I/ symbols causes the PHY to de-assert the internal carrier sense and RX_DV.

These symbols are not translated into data.

3.9.2.7 Receiver Errors

During a frame, unexpected code-groups are considered receive errors. Expected code groups are the DATA set (0 through F), and the /T/R/ (ESD) symbol pair. When a receive error occurs, the internal MII’s RX_ER signal is asserted and arbitrary data is driven onto the internal receive data bus (RXD) to the MAC. Should an error be detected during the time that the /J/K/ delimiter is being decoded (bad SSD error), RX_ER is asserted and the value 1110b is driven onto the internal receive data bus (RXD) to the MAC. Note that the internal MII's data valid signal (RX_DV) is not yet asserted when the bad SSD occurs.

3.9.3 10BASE-T Transmit

Data to be transmitted comes from the MAC layer controller. The 10Base-T transmitter receives 4-bit nibbles from the MII at a rate of 2.5MHz and converts them to a 10Mbps serial data stream. The data stream is then Manchester encoded and sent to the analog transmitter, which drives a signal onto the twisted pair via the external magnetics.

The 10M transmitter uses the following blocks:

- MII (digital)
- TX 10M (digital)
- 10M Transmitter (analog)
- 10M PLL (analog)

3.9.3.1 10M Transmit Data Across the Internal MII Bus

The MAC controller drives the transmit data onto the internal TXD BUS. When the controller has driven TX_EN high to indicate valid data, the data is latched by the MII block on the rising edge of TX_CLK. The data is in the form of 4-bit wide 2.5MHz data.

3.9.3.2 Manchester Encoding

The 4-bit wide data is sent to the TX10M block. The nibbles are converted to a 10Mbps serial NRZI data stream. The 10M PLL locks onto the external clock or internal oscillator and produces a 20MHz clock. This is used to Manchester encode the NRZ data stream. When no data is being transmitted (TX_EN is low), the TX10M block outputs Normal Link Pulses (NLPs) to maintain communications with the remote link partner.

3.9.3.3 10M Transmit Drivers

The Manchester encoded data is sent to the analog transmitter where it is shaped and filtered before being driven out as a differential signal across the TXP and TXN outputs.

3.9.4 10BASE-T Receive

The 10Base-T receiver gets the Manchester encoded analog signal from the cable via the magnetics. It recovers the receive clock from the signal and uses this clock to recover the NRZI data stream. This 10M serial data is converted to 4-bit data nibbles which are passed to the controller across the MII at a rate of 2.5MHz.

This 10M receiver uses the following blocks:

- Filter and SQUELCH (analog)
- 10M PLL (analog)
- RX 10M (digital)
- MII (digital)

3.9.4.1 10M Receive Input and Squelch

The Manchester signal from the cable is fed into the PHY (on inputs RXP and RXN) via 1:1 ratio magnetics. It is first filtered to reduce any out-of-band noise. It then passes through a SQUELCH circuit. The SQUELCH is a set of amplitude and timing comparators that normally reject differential voltage levels below 300mV and detect and recognize differential voltages above 585mV.

3.9.4.2 Manchester Decoding

The output of the SQUELCH goes to the RX10M block where it is validated as Manchester encoded data. The polarity of the signal is also checked. If the polarity is reversed (local RXP is connected to RXN of the remote partner and vice versa), then this is identified and corrected. The reversed condition is indicated by the flag “XPOL“, bit 4 in register 27. The 10M PLL is locked onto the received Manchester signal and from this, generates the received 20MHz clock. Using this clock, the Manchester encoded data is extracted and converted to a 10MHz NRZI data stream. It is then converted from serial to 4-bit wide parallel data.

The RX10M block also detects valid 10Base-T IDLE signals - Normal Link Pulses (NLPs) - to maintain the link.

3.9.4.3 Jabber Detection

Jabber is a condition in which a station transmits for a period of time longer than the maximum permissible packet length, usually due to a fault condition, that results in holding the TX_EN input for a long period. Special logic is used to detect the jabber state and abort the transmission to the line, within 45 mS. Once TX_EN is deasserted, the logic resets the jabber condition.

3.9.5 Auto-negotiation

The purpose of the Auto-negotiation function is to automatically configure the PHY to the optimum link parameters based on the capabilities of its link partner. Auto-negotiation is a mechanism for exchanging configuration information between two link-partners and automatically selecting the highest performance mode of operation supported by both sides. Auto-negotiation is fully defined in clause 28 of the IEEE 802.3 specification.

Once auto-negotiation has completed, information about the resolved link can be passed back to the controller via the internal Serial Management Interface (SMI). The results of the negotiation process are reflected in the Speed Indication bits in register 31, as well as the Link Partner Ability Register (Register 5).

The auto-negotiation protocol is a purely physical layer activity and proceeds independently of the MAC controller.

The advertised capabilities of the PHY are stored in register 4 of the SMI registers. The default advertised by the PHY is determined by user-defined on-chip signal options.

The following blocks are activated during an Auto-negotiation session:

- Auto-negotiation (digital)
- 100M ADC (analog)
- 100M PLL (analog)
- 100M equalizer/BLW/clock recovery (DSP)
- 10M SQUELCH (analog)
- 10M PLL (analog)
- 10M Transmitter (analog)

When enabled, auto-negotiation is started by the occurrence of one of the following events:

- Hardware reset
- Software reset
- Power-down reset
- Link status down
- Setting register 0, bit 9 high (auto-negotiation restart)

On detection of one of these events, the PHY begins auto-negotiation by transmitting bursts of Fast Link Pulses (FLP). These are bursts of link pulses from the 10M transmitter. They are shaped as Normal Link Pulses and can pass uncorrupted down CAT-3 or CAT-5 cable. A Fast Link Pulse Burst consists of up to 33 pulses. The 17 odd-numbered pulses, which are always present, frame the FLP burst. The 16 even-numbered pulses, which may be present or absent, contain the data word being transmitted. Presence of a data pulse represents a “1”, while absence represents a “0”.

The data transmitted by an FLP burst is known as a “Link Code Word.” These are defined fully in IEEE 802.3 clause 28. In summary, the PHY advertises 802.3 compliance in its selector field (the first 5 bits of the Link Code Word). It advertises its technology ability according to the bits set in register 4 of the SMI registers.

There are 4 possible matches of the technology abilities. In the order of priority these are:

- 100M full-duplex (Highest priority)
- 100M half-duplex
- 10M full-duplex
- 10M half-duplex

If the full capabilities of the PHY are advertised (100M, full-duplex), and if the link partner is capable of 10M and 100M, then auto-negotiation selects 100M as the highest performance mode. If the link partner is capable of half and full-duplex modes, then auto-negotiation selects full-duplex as the highest performance operation.

Once a capability match has been determined, the link code words are repeated with the acknowledge bit set. Any difference in the main content of the link code words at this time will cause auto-negotiation to re-start. Auto-negotiation will also re-start if not all of the required FLP bursts are received.

Writing register 4 bits [8:5] allows software control of the capabilities advertised by the PHY. Writing register 4 does not automatically re-start auto-negotiation. Register 0, bit 9 must be set before the new abilities will be advertised. Auto-negotiation can also be disabled via software by clearing register 0, bit 12.

The device does not support “Next Page" capability.

3.9.6 Parallel Detection

If the device is connected to a another device lacking the ability to auto-negotiate (i.e. no FLPs are detected), it is able to determine the speed of the link based on either 100M MLT-3 symbols or 10M Normal Link Pulses. In this case the link is presumed to be half-duplex per the IEEE standard. This ability is known as “Parallel Detection”. This feature ensures inter operability with legacy link partners. If a link is formed via parallel detection, then bit 0 in register 6 is cleared to indicate that the Link Partner is not capable of auto-negotiation. The Ethernet MAC has access to this information via the management interface. If a fault occurs during parallel detection, bit 4 of register 6 is set.

Register 5 is used to store the Link Partner Ability information, which is coded in the received FLPs. If the Link Partner is not auto-negotiation capable, then register 5 is updated after completion of parallel detection to reflect the speed capability of the Link Partner.

3.9.6.1 Re-starting Auto-negotiation

Auto-negotiation can be re-started at any time by setting register 0, bit 9. Auto-negotiation will also re-start if the link is broken at any time. A broken link is caused by signal loss. This may occur because of a cable break, or because of an interruption in the signal transmitted by the Link Partner. Auto-negotiation resumes in an attempt to determine the new link configuration.

If the management entity re-starts Auto-negotiation by writing to bit 9 of the control register, the device will respond by stopping all transmission/receiving operations. Once the break_link_timer is done, in the Auto-negotiation state-machine (approximately 1200 mS) the auto-negotiation will re-start. The Link

Partner will have also dropped the link due to lack of a received signal, so it too will resume auto-negotiation.

3.9.6.2 Disabling Auto-negotiation

Auto-negotiation can be disabled by setting register 0, bit 12 to zero. The device will then force its speed of operation to reflect the information in register 0, bit 13 (speed) and register 0, bit 8 (duplex). The speed and duplex bits in register 0 should be ignored when auto-negotiation is enabled.

3.9.6.3 Half vs. Full-Duplex

Half-duplex operation relies on the CSMA/CD (Carrier Sense Multiple Access / Collision Detect) protocol to handle network traffic and collisions. In this mode, the internal carrier sense signal, CRS, responds to both transmit and receive activity. In this mode, If data is received while the PHY is transmitting, a collision results.

In full-duplex mode, the PHY is able to transmit and receive data simultaneously. In this mode, the internal CRS responds only to receive activity. The CSMA/CD protocol does not apply and collision detection is disabled.

Table 3.88 describes the behavior of the internal CRS bit under all receive/transmit conditions.

The internal CRS signal is used to trigger bit 10 (No Carrier) of the TX Status Word (See Section 3.7.2.6, "TX Status Format," on page 91). The CRS value, and subsequently the No Carrier value, are invalid during any full-duplex transmission. Therefore, these signals cannot be used as a verification method of transmitted packets when transmitting in 10/100 Mbps full-duplex modes.

Table 3.88 CRS Behavior

MODE	SPEED	DUPLEX	ACTIVITY	CRS BEHAVIOR (Note 3.45)
Manual	10 Mbps	Half-Duplex	Transmitting	Active
Manual	10 Mbps	Half-Duplex	Receiving	Active
Manual	10 Mbps	Full-Duplex	Transmitting	Low
Manual	10 Mbps	Full-Duplex	Receiving	Active
Manual	100 Mbps	Half-Duplex	Transmitting	Active
Manual	100 Mbps	Half-Duplex	Receiving	Active
Manual	100 Mbps	Full-Duplex	Transmitting	Low
Manual	100 Mbps	Full-Duplex	Receiving	Active
Auto-Negotiation	10 Mbps	Half-Duplex	Transmitting	Active
Auto-Negotiation	10 Mbps	Half-Duplex	Receiving	Active
Auto-Negotiation	10 Mbps	Full-Duplex	Transmitting	Low
Auto-Negotiation	10 Mbps	Full-Duplex	Receiving	Active
Auto-Negotiation	100 Mbps	Half-Duplex	Transmitting	Active
Auto-Negotiation	100 Mbps	Half-Duplex	Receiving	Active
Auto-Negotiation	100 Mbps	Full-Duplex	Transmitting	Low
Auto-Negotiation	100 Mbps	Full-Duplex	Receiving	Active

Note 3.45 The 10/100 PHY internal CRS signal operates in two modes: Active and Low. When in Active mode, the internal CRS will transition high and low upon line activity, where a high value indicates a carrier has been detected. In Low mode, the internal CRS stays low and does not indicate carrier detection. The internal CRS signal and No Carrier (bit 10 of the TX Status Word) cannot be used as a verification method of transmitted packets when transmitting in 10/100 Mbps full-duplex mode.

3.9.7 HP Auto-MDIX

HP Auto-MDIX facilitates the use of CAT-3 (10 Base-T) or CAT-5 (100 Base-T) media UTP interconnect cable without consideration of interface wiring scheme. If a user plugs in either a direct connect LAN cable, or a cross-over patch cable, as shown in Figure 3.19, the Auto-MDIX PHY is capable of configuring the TPO and TPI twisted pair pins for correct transceiver operation.

The internal logic of the device detects the TX and RX pins of the connecting device. Since the RX and TX line pairs are interchangeable, special PCB design considerations are needed to accommodate the symmetrical magnetics and termination of an Auto-MDIX design.

The Auto-MDIX function can be disabled through an internal register 27.15, or the external control pins AMDIX_EN. When disabled the TX and RX pins can be configured with the Channel Select (CH_SELECT) pin as desired.

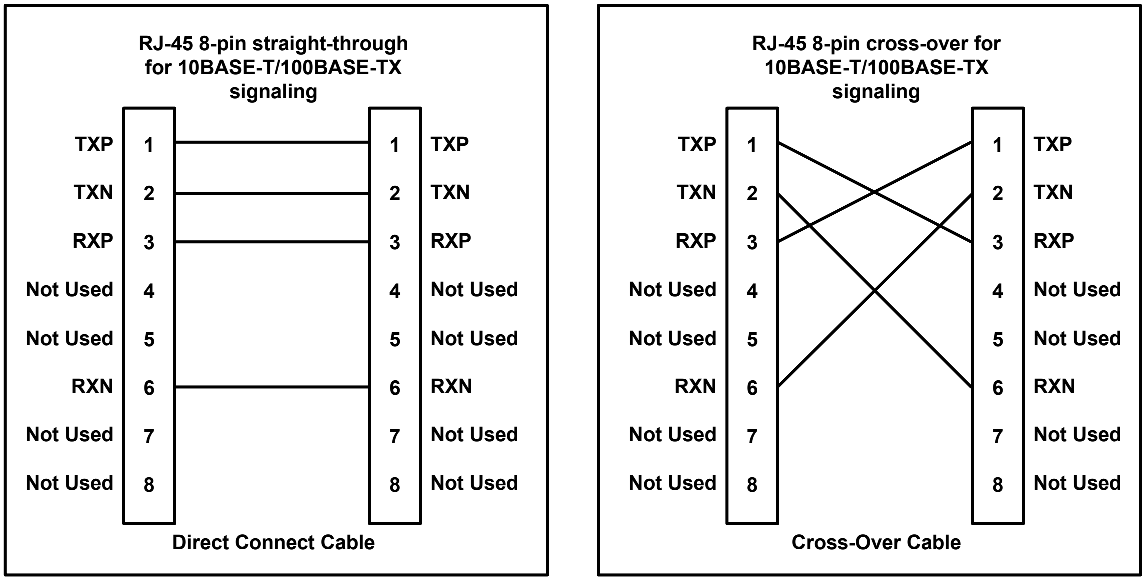


Figure 3.19 Direct cable connection vs. Cross-over cable connection.

3.9.8 PHY Power-Down Modes

There are 2 power-down modes for the PHY as discussed in the following sections.

3.9.8.1 General Power-Down

This power-down is controlled by register 0, bit 11. In this mode the PHY, except the management interface, is powered-down and stays in that condition as long as PHY register bit 0.11 is HIGH. When bit 0.11 is cleared, the PHY powers up and is automatically reset. Please refer to Section 4.5.1, "Basic Control Register," on page 197 for additional information on this register.

Note: For maximum power savings, auto-negotiation should be disabled before enabling the General Power-Down mode.

3.9.8.2 Energy Detect Power-Down

This power-down mode is activated by setting the PHY register bit 17.13 to 1. Please refer to Section 4.5.8, "Mode Control/Status Register," on page 204 for additional information on this register. In this mode when no energy is present on the line, the PHY is powered down, with the exception of the management interface, the SQUELCH circuit and the ENERGYON logic. The ENERGYON logic is used to detect the presence of valid energy from 100BASE-TX, 10BASE-T, or Auto-negotiation signals

In this mode, when the ENERGYON signal is low, the PHY is powered-down, and nothing is transmitted. When energy is received - link pulses or packets - the internal ENERGYON signal is asserted, and the PHY powers-up. It automatically resets itself into the state it had prior to power-down, and asserts the INT7 bit of the PHY Interrupt Source Flag Register register. If the ENERGYON interrupt is enabled, this event will cause a PHY interrupt to the Interrupt Controller and the power management event detection logic.

The first and possibly the second packet to activate ENERGYON may be lost.

When 17.13 is low, energy detect power-down is disabled.

3.9.9 PHY Resets

In addition to a chip-level reset, the PHY supports two software-initiated resets. These are discussed in the following sections.

3.9.9.1 PHY Soft Reset via PMT_CTL Register PHY Reset (PHY_RST) Bit

The PHY soft reset is initiated by writing a ‘1’ to the PHY Reset (PHY_RST) bit of the Power Management Control Register (PMT_CTL). This self-clearing bit will return to ‘0’ after approximately 100µs, at which time the PHY reset is complete.

3.9.9.2 PHY Soft Reset via PHY Basic Control Register Bit 15 (PHY Reg. 0.15)

The PHY Reg. 0.15 Soft Reset is initiated by writing a ‘1’ to bit 15 of the PHY’s Basic Control Register. This self-clearing bit will return to ‘0’ after approximately 256µs, at which time the PHY reset is complete. The BCR reset initializes the logic within the PHY, with the exception of register bits marked as NASR (Not Affected by Software Reset).

3.9.10 Required Ethernet Magnetics

The magnetics selected for use with the device should be an Auto-MDIX style magnetic available from several vendors. The user is urged to review SMSC Application Note 8.13 "Suggested Magnetics" for the latest qualified and suggested magnetics. Vendors and part numbers are provided in this application note.

3.9.11 PHY Registers

Please refer to Section 4.5, "PHY Registers," on page 196 for a complete description of the PHY registers.

3.10 EEPROM Controller (EPC)

The device may use an external EEPROM to store the default values for the MAC address, the Hub Configuration registers, and the USB descriptors. The EEPROM controller supports most “93C46” type EEPROMs. A total of nine address bits are used to support 256/512 byte EEPROMs.

Note: A 3-wire style 2K/4K EEPROM that is organized for 256/512 x 8-bit operation must be used.

The MAC address is used as the default Ethernet MAC address and is loaded into the MAC’s ADDRH and ADDRL registers. If a properly configured EEPROM is not detected, it is the responsibility of the Host LAN Driver to set the IEEE addresses.

After a system-level reset occurs, the device will load the default values from a properly configured EEPROM. The device will not accept USB transactions from the Host until this process is completed.

The EEPROM controller also allows the Host system to read, write and erase the contents of the Serial EEPROM.

3.10.1 EEPROM Format

Table 3.89 illustrates the format in which data is stored inside of the EEPROM.

Note the EEPROM offsets are given in units of 16-bit word offsets. A length field with a value of zero indicates that the field does not exist in the EEPROM. The device will use the field’s HW default value in this case.

Note: Ethernet Controller related data is stored in addresses 00h through 1Fh. Ethernet Controller descriptor images for loading may appear in EEPROM after address 39h. Only the Ethernet Controller descriptor images may be specified in their entirety in EEPROM. Hub related parameters reside in addresses 20h through 39h. These Hub parameters are used to initialize the Hub registers. Hub descriptors are generated and stored internally, as dictated by the Hub registers. If a valid EEPROM is not present, the Hub descriptors are generated according to the Internal Defaults of the Hub registers.

Note: For Device Descriptors, the only valid values for the length are 0 and 18.

Note: For Configuration and Interface Descriptors, the only valid values for the length are 0 and 18.

Note: If no Configuration Descriptor is present in the EEPROM, then the Configuration Flags affect the values of bmAttributes and bMaxPower in the Ethernet Controller Configuration Descriptor.

Note: The EEPROM programmer must ensure that if a String Descriptor does not exist in the EEPROM, the referencing descriptor must contain 00h for the respective string index field.

Note: If all String Descriptor lengths are zero, then a Language ID will not be supported.

Table 3.89 EEPROM Format

EEPROM ADDRESS	EEPROM CONTENTS
00h	0xA5
01h	MAC Address [7:0]
02h	MAC Address [15:8]
03h	MAC Address [23:16]

Table 3.89 EEPROM Format (continued)

EEPROM ADDRESS	EEPROM CONTENTS
04h	MAC Address [31:24]
05h	MAC Address [39:32]
06h	MAC Address [47:40]
07h	Full-Speed Polling Interval for Interrupt Endpoint
08h	Hi-Speed Polling Interval for Interrupt Endpoint
09h	Configuration Flags
0Ah	Language ID Descriptor [7:0]
0Bh	Language ID Descriptor [15:8]
0Ch	Manufacturer ID String Descriptor Length (bytes)
0Dh	Manufacturer ID String Descriptor EEPROM Word Offset
0Eh	Product Name String Descriptor Length (bytes)
0Fh	Product Name String Descriptor EEPROM Word Offset
10h	Serial Number String Descriptor Length (bytes)
11h	Serial Number String Descriptor EEPROM Word Offset
12h	Configuration String Descriptor Length (bytes)
13h	Configuration String Descriptor Word Offset
14h	Interface String Descriptor Length (bytes)
15h	Interface String Descriptor Word Offset
16h	Hi-Speed Device Descriptor Length (bytes)
17h	Hi-Speed Device Descriptor Word Offset
18h	Hi-Speed Configuration and Interface Descriptor Length (bytes)
19h	Hi-Speed Configuration and Interface Descriptor Word Offset
1Ah	Full-Speed Device Descriptor Length (bytes)
1Bh	Full-Speed Device Descriptor Word Offset
1Ch	Full-Speed Configuration and Interface Descriptor Length (bytes)
1Dh	Full-Speed Configuration and Interface Descriptor Word Offset
1Eh-1Fh	RESERVED
20h	Vendor ID LSB (VIDL) Register
21h	Vendor ID MSB (VIDM) Register
22h	Product ID LSB (PIDL) Register
23h	Product ID MSB (PIDM) Register
24h	Device ID LSB (DIDL) Register

Table 3.89 EEPROM Format (continued)

EEPROM ADDRESS	EEPROM CONTENTS
25h	Device ID MSB (DIDM) Register
26h	Config Data Byte 1 (CFG1) Register
27h	Config Data Byte 2 (CFG2) Register
28h	Config Data Byte 3 (CFG3) Register
29h	Non-Removable Devices (NRD) Register
2Ah	Port Disable For Self-Powered Operation (PDS) Register
2Bh	Port Disable For Bus-Powered Operation (PDB) Register
2Ch	Max Power For Self-Powered Operation (MAXPS) Register
2Dh	Max Power For Bus-Powered Operation (MAXPB) Register
2Eh	Hub Controller Max Current For Self-Powered Operation (HCMCS) Register
2Fh	Hub Controller Max Current For Bus-Powered Operation (HCMCB) Register
30h	Power-On Time (PWRT) Register
31h	Boost_Up (BOOSTUP) Register
32h	RESERVED
33h	Boost_3:2 (BOOST32) Register
34h	RESERVED
35h	Port Swap (PRTSP) Register
36h	Port Remap 12 (PRTR12) Register
37h	Port Remap 3 (PRTR3) Register
38h	RESERVED
39h	Status/Command (STCD) Register

Note: The descriptor type for the device descriptors specified in the EEPROM is a don't care and always overwritten by HW to 0x1.
The descriptor size for the device descriptors specified in the EEPROM is a don't care and always overwritten by HW to 0x12.
The descriptor type for the configuration descriptors specified in the EEPROM is a don't care and always overwritten by HW to 0x2.

Note: Descriptors specified in EEPROM having bcdUSB, bMaxPacketSize0, and bNumConfigurations fields defined with values other than 0200h, 40h, and 1, respectively, will result in unwanted behavior and untoward results.

Note: EEPROM byte addresses past 39h can be used to store data for any purpose.

Table 3.90 describes the Configuration Flags.

Table 3.90 Configuration Flags Description

BIT	NAME	DESCRIPTION
7:3	**RESERVED**	-
2	Remote Wakeup Support	0 = The device does not support remote wakeup. 1 = The device supports remote wakeup.
1	**RESERVED**	-
0	Power Method	0 = The device is bus powered. 1 = The device is self powered.

3.10.2 EEPROM Defaults

The signature value of 0xA5 is stored at address 0. A different signature value indicates to the EEPROM controller that no EEPROM or an un-programmed EEPROM is attached to the device. In this case, default values are used to initialize the Hub configuration registers and the Ethernet Controller whenever system level resets occur that cause either the Hub or the Ethernet Controller to be reset.

Default values assumed by the Hub registers can be found in Section 4.6, "Hub Configuration Registers (HCFG)," on page 210. The Hub registers defaults are also used to internally generate the Hub USB descriptors. Please refer to Section 3.5.4, "Hub USB Descriptors," on page 33 for further information.

Default values assumed by the Ethernet Controller for its initialization are shown in Table 3.91. Please refer to Section 3.6.3, "Ethernet Controller USB Descriptors," on page 67 and Section 3.6.4, "Statistics," on page 74 for further information about the default USB values assumed by the Ethernet Controller.

Table 3.91 Ethernet Controller EEPROM Defaults

FIELD	DEFAULT VALUE
Ethernet Controller MAC Address	FFFFFFFFFFFFh
Ethernet Controller Full-Speed Polling Interval (mS)	01h
Ethernet Controller Hi-Speed Polling Interval (mS)	04h
Ethernet Controller Configuration Flags	05h
Ethernet Controller Maximum Power (mA)	01h
Ethernet Controller Vendor ID	0424h
Ethernet Controller Product ID	EC00h

3.10.3 EEPROM Auto-Load

Certain resets (USB reset, POR, nRESET, and SRST) cause the EEPROM contents to be loaded into the device. After a reset, the EEPROM controller attempts to read the first byte of data from the EEPROM. If the value 0xA5 is read from the first address, then the EEPROM controller will assume that the external serial EEPROM is configured for auto-loading. If a value other 0xA5 is read from the first address, the EEPROM auto-load will not commence.

The EEPROM Controller will then load the entire contents of the EEPROM into an internal 512 byte SRAM. Depending on type of reset, the contents of the SRAM are accessed by the CTL (USB Control Block) as needed (I.E. to fill Get Descriptor commands, initialize Hub configuration registers, etc.). A detailed explanation of the EEPROM byte ordering with respect to the MAC address is given in Section 4.4.3, "MAC Address Low Register (ADDRL)," on page 185.

If an 0xA5h is not read from the first address, the EEPROM controller will end initialization. Depending on the type of reset, assumption of the default values, as specified in Section 3.10.2, will occur. In the case where the type of reset has caused the Ethernet Controller to assume the default value for the MAC address, the Host LAN driver software must set the IEEE address by writing to the MAC's ADDRH and ADDRL registers.

The device may not respond to the USB Host until the EEPROM loading sequence has completed for the Hub.

Note: POR or nRESET result in the Hub configuration registers being initialized with their images, as read from the EEPROM (0xA5 signature present), or default values (0xA5 signature absent). In addition, the Hub USB descriptors will be internally generated. No Ethernet Controller initialization from EEPROM information occurs as a result of these resets.

The Ethernet Controller may subsequently be initialized upon the Host issuing set_feature (power) for port 1, Soft Reset (SRST), or a EEPROM RELOAD command. The information read from the EEPROM (0xA5 signature present) or the defaults (Table 3.91 if 0xA5 signature absent) will be used.

Note: USB reset or the issuance of a set_feature (reset) for port 1 by the Host causes the MAC address to be reloaded with information read from the EEPROM (0xA5 signature present), or the default value specified in Table 3.91 (0xA5 signature absent).

3.10.4 EEPROM Host Operations

After the EEPROM controller has finished reading (or attempting to read) the EEPROM after a system-level reset, the Host is free to perform other EEPROM operations. EEPROM operations are performed using the EEPROM Command (E2P_CMD) and EEPROM Data (E2P_DATA) registers. Section 4.3.12, "EEPROM Command Register (E2P_CMD)," on page 167 provides an explanation of the supported EEPROM operations.

If the EEPROM operation is the “write location” (WRITE) or “write all” (WRAL) commands, the Host must first write the desired data into the E2P_DATA register. The Host must then issue the WRITE or WRAL command using the E2P_CMD register by setting the EPC_CMD field appropriately. If the operation is a WRITE, the EPC_ADDR field in E2P_CMD must also be set to the desired location. The command is executed when the Host sets the EPC_BSY bit high. The completion of the operation is indicated when the EPC_BSY bit is cleared.

If the EEPROM operation is the “read location” (READ) operation, the Host must issue the READ command using the E2P_CMD register with the EPC_ADDR set to the desired location. The command is executed when the Host sets the EPC_BSY bit high. The completion of the operation is indicated when the EPC_BSY bit is cleared, at which time the data from the EEPROM may be read from the E2P_DATA register.

Other EEPROM operations are performed by writing the appropriate command to the E2P_CMD register. The command is executed when the Host sets the EPC_BSY bit high. The completion of the operation is indicated when the EPC_BSY bit is cleared. In all cases, the Host must wait for EPC_BSY to clear before modifying the E2P_CMD register.

Note: The EEPROM device powers-up in the erase/write disabled state. To modify the contents of the EEPROM, the Host must first issue the EWEN command.

If an operation is attempted, and an EEPROM device does not respond within 30mS, the device will timeout, and the EPC Time-out bit (EPC_TO) in the E2P_CMD register will be set.

Figure 3.20 illustrates the Host accesses required to perform an EEPROM Read or Write operation.

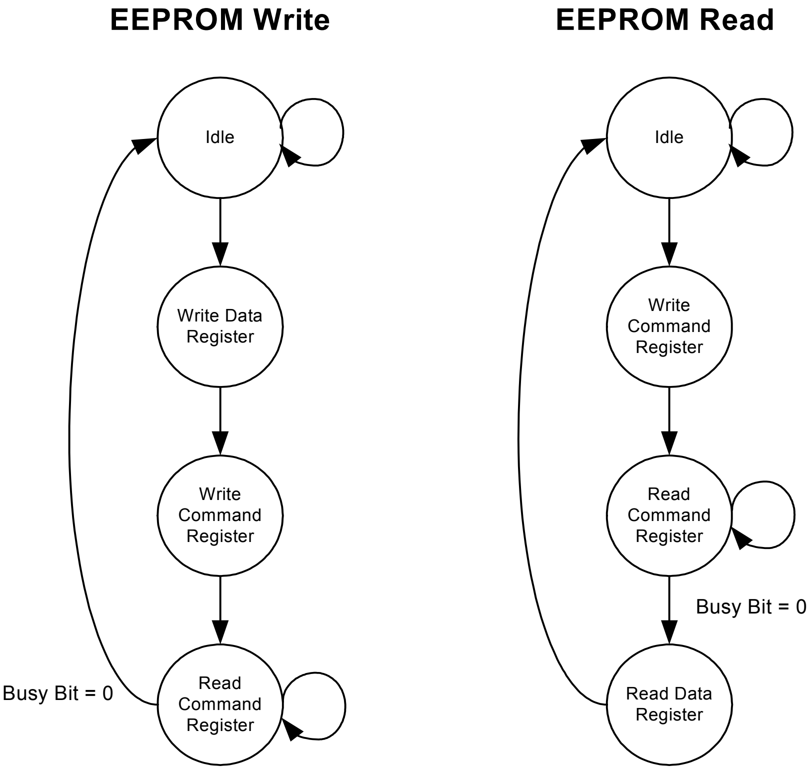


Figure 3.20 EEPROM Access Flow Diagram

3.10.4.1 Supported EEPROM Operations

The EEPROM controller supports the following EEPROM operations under Host control via the E2P_CMD register. The operations are commonly supported by “93C46” EEPROM devices. A description and functional timing diagram is provided below for each operation. Please refer to the E2P_CMD register description in Section 4.3.12, "EEPROM Command Register (E2P_CMD)," on page 167 for E2P_CMD field settings for each command.

ERASE (Erase Location): If erase/write operations are enabled in the EEPROM, this command will erase the location selected by the EPC Address field (EPC_ADDR). The EPC_TO bit is set if the EEPROM does not respond within 30mS.

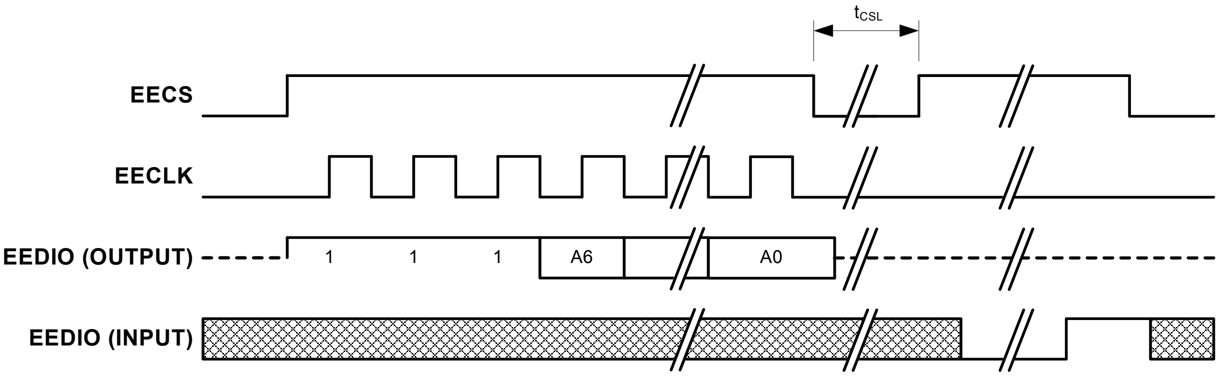


Figure 3.21 EEPROM ERASE Cycle

ERAL (Erase All): If erase/write operations are enabled in the EEPROM, this command will initiate a bulk erase of the entire EEPROM.The EPC_TO bit is set if the EEPROM does not respond within 30mS.

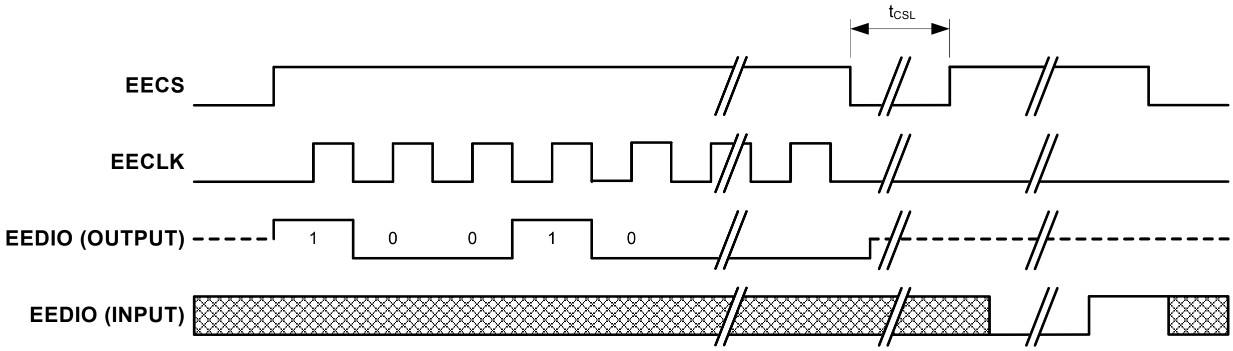


Figure 3.22 EEPROM ERAL Cycle

EWDS (Erase/Write Disable): After issued, the EEPROM will ignore erase and write commands. To re-enable erase/write operations issue the EWEN command.

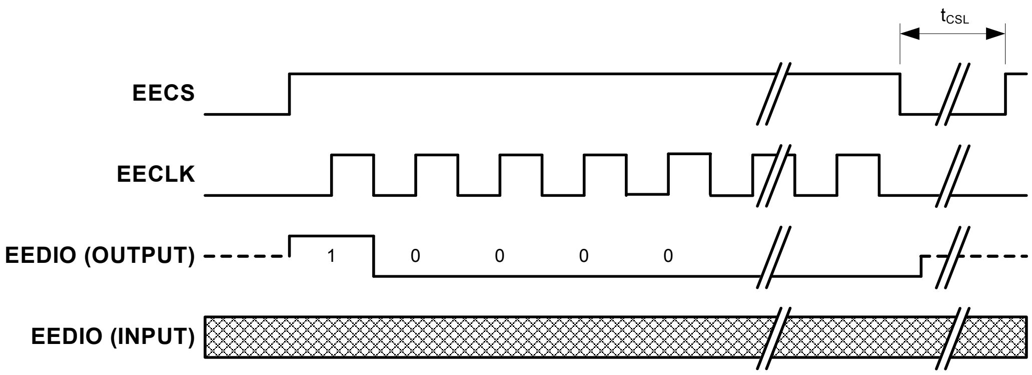


Figure 3.23 EEPROM EWDS Cycle

EWEN (Erase/Write Enable): Enables the EEPROM for erase and write operations. The EEPROM will allow erase and write operations until the "Erase/Write Disable" command is sent, or until power is cycled.

Note: The EEPROM device will power-up in the erase/write-disabled state. Any erase or write operations will fail until an Erase/Write Enable command is issued.

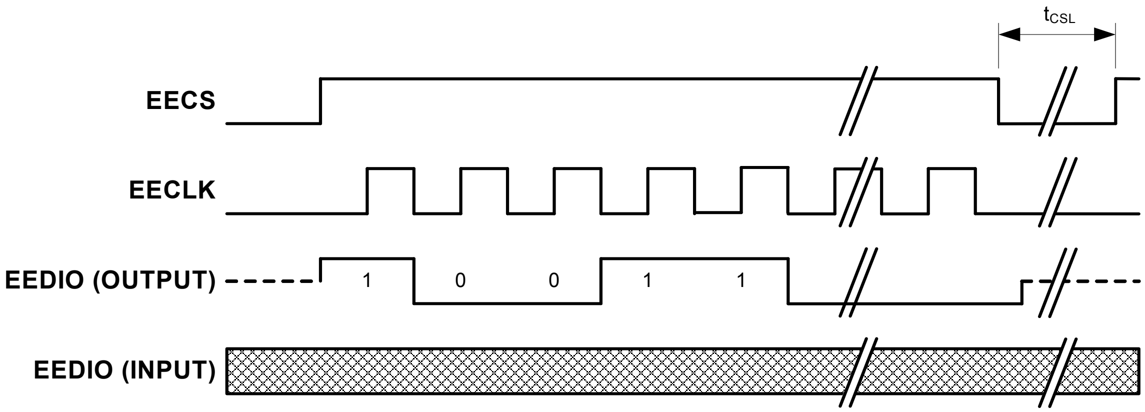


Figure 3.24 EEPROM EWEN Cycle

READ (Read Location): This command will cause a read of the EEPROM location pointed to by EPC Address (EPC_ADDR). The result of the read is available in the E2P_DATA register.

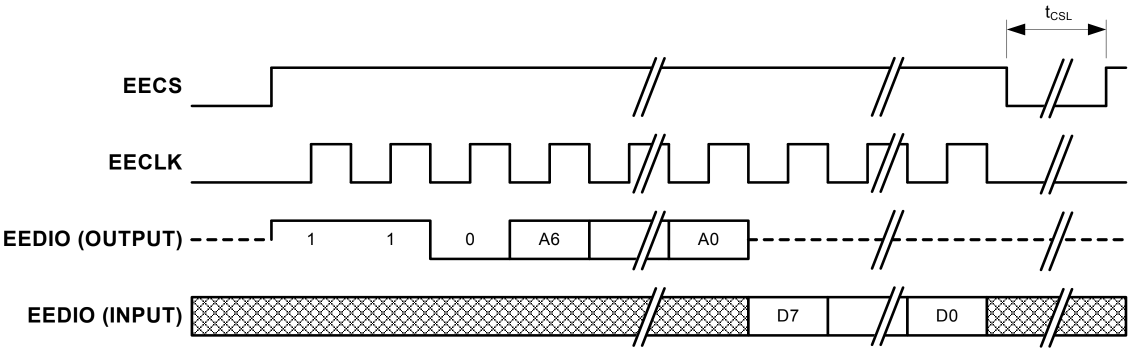


Figure 3.25 EEPROM READ Cycle

WRITE (Write Location): If erase/write operations are enabled in the EEPROM, this command will cause the contents of the E2P_DATA register to be written to the EEPROM location selected by the EPC Address field (EPC_ADDR). The EPC_TO bit is set if the EEPROM does not respond within 30mS.

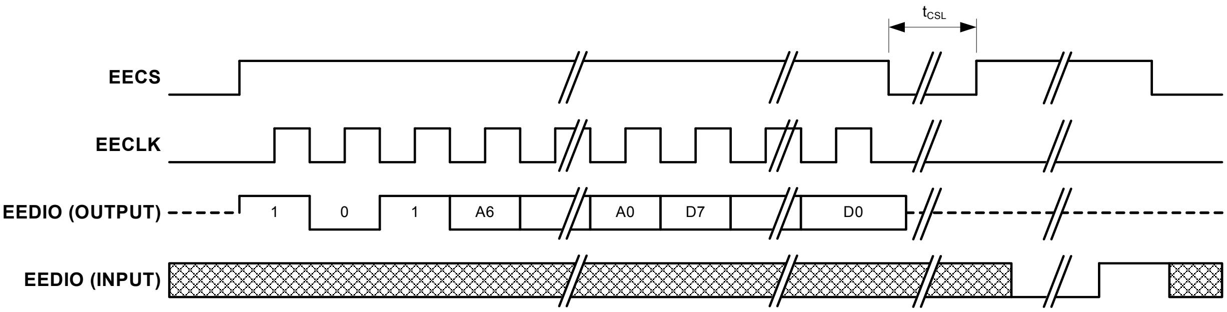


Figure 3.26 EEPROM WRITE Cycle

WRAL (Write All): If erase/write operations are enabled in the EEPROM, this command will cause the contents of the E2P_DATA register to be written to every EEPROM memory location. The EPC_TO bit is set if the EEPROM does not respond within 30mS.

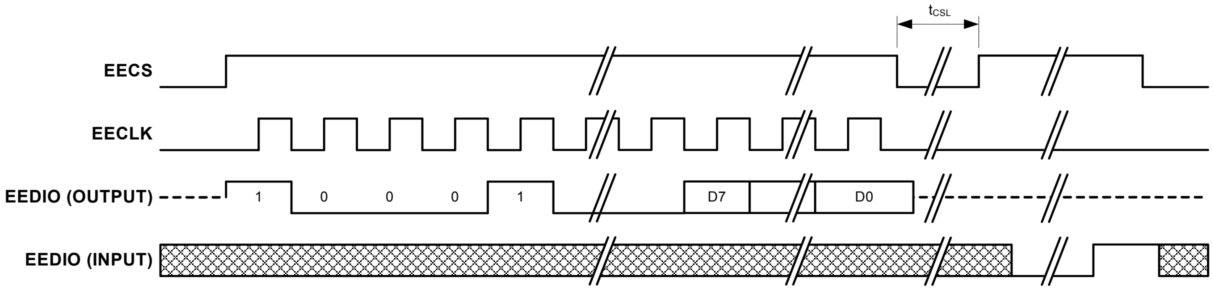


Figure 3.27 EEPROM WRAL Cycle

Table 3.92, "Required EECLK Cycles", shown below, shows the number of EECLK cycles required for each EEPROM operation.

Table 3.92 Required EECLK Cycles

OPERATION	REQUIRED EECLK CYCLES
ERASE	10
ERAL	10
EWDS	10
EWEN	10
READ	18
WRITE	18
WRAL	18

3.10.4.2 Host Initiated EEPROM Reload

The Host can initiate a reload of the EEPROM by issuing the RELOAD command via the E2P Command (E2P_CMD) register. If the first byte read from the EEPROM is not 0xA5, it is assumed that the EEPROM is not present, or not programmed, and the reload will fail. The Data Loaded bit of the E2P_CMD register indicates a successful reload of the EEPROM.

Note: It is not recommended that the RELOAD command be used as part of normal operation, as race conditions can occur with USB Commands that access descriptor data. It is best for the Host to issue a SRST to reload the EEPROM data.

3.10.4.3 EEPROM Command and Data Registers

Refer to Section 4.3.12, "EEPROM Command Register (E2P_CMD)," on page 167 and Section 4.3.13, "EEPROM Data Register (E2P_DATA)," on page 170 for a detailed description of these registers. Supported EEPROM operations are described in these sections.

3.10.4.4 EEPROM Timing

Refer to Section 5.5.3, "EEPROM Timing," on page 241 for detailed EEPROM timing specifications.

3.10.5 An Example of EEPROM Format Interpretation

Table 3.93 and Table 3.94 provide an example of how the contents of a EEPROM are formatted. Table 3.93 is a dump of the EEPROM memory (256-byte EEPROM), while Table 3.94 illustrates, byte by byte, how the EEPROM is formatted.

Table 3.93 Dump of EEPROM Memory

OFFSET BYTE	VALUE
0000h	A5 12 34 56 78 9A BC 01
0008h	04 05 09 04 0A 1D 00 00
0010h	00 00 00 00 00 00 12 22
0018h	12 2B 12 34 12 3D 00 00
0020h	24 04 12 95 00 01 9B 18
0028h	00 02 30 30 01 00 01 00
0030h	32 00 00 00 00 00 21 03
0038h	00 01 0A 03 53 00 4D 00
0040h	53 00 43 00 12 01 00 02
0048h	FF 00 01 40 24 04 00 EC
0050h	00 01 01 00 00 01 09 02
0058h	27 00 01 01 00 E0 01 09
0060h	04 00 00 03 FF 00 FF 00
0068h	12 01 00 02 FF 00 FF 40
0070h	24 04 00 EC 00 01 01 00
0078h	00 01 09 02 27 00 01 01
0080h	00 E0 01 09 04 00 00 03
0088h	FF 00 FF 00 ....
0090h - 00FFh	....

Table 3.94 EEPROM Example - 256 Byte EEPROM

EEPROM ADDRESS	EEPROM CONTENTS (HEX)	DESCRIPTION
00h	A5	EEPROM Programmed Indicator
01h-06h	12 34 56 78 9A BC	MAC Address 12 34 56 78 9A BC
07h	01	Full-Speed Polling Interval for Interrupt Endpoint (1ms)
08h	04	Hi-Speed Polling Interval for Interrupt Endpoint (4ms)
09h	05	Configuration Flags - The device is self powered and supports remote wakeup.
0Ah-0Bh	09 04	Language ID Descriptor 0409h, English
0Ch	0A	Manufacturer ID String Descriptor Length (10 bytes)
0Dh	1D	Manufacturer ID String Descriptor EEPROM Word Offset (1Dh) Corresponds to EEPROM Byte Offset 3Ah
0Eh	00	Product Name String Descriptor Length (0 bytes - NA)
0Fh	00	Product Name String Descriptor EEPROM Word Offset (Don’t Care)
10h	00	Serial Number String Descriptor Length (0 bytes - NA)
11h	00	Serial Number String Descriptor EEPROM Word Offset (Don’t Care)
12h	00	Configuration String Descriptor Length (0 bytes - NA)
13h	00	Configuration String Descriptor Word Offset (Don’t Care)
14h	00	Interface String Descriptor Length (0 bytes - NA)
15h	00	Interface String Descriptor Word Offset (Don’t Care)
16h	12	Hi-Speed Device Descriptor Length (18 bytes)
17h	22h	Hi-Speed Device Descriptor Word Offset (22h) Corresponds to EEPROM Byte Offset 44h
18h	12	Hi-Speed Configuration and Interface Descriptor Length (18 bytes)
19h	2B	Hi-Speed Configuration and Interface Descriptor Word Offset (2Bh) Corresponds to EEPROM Byte Offset 56h
1Ah	12	Full-Speed Device Descriptor Length (18 bytes)
1Bh	34	Full-Speed Device Descriptor Word Offset (34h) Corresponds to EEPROM Byte Offset 68h
1Ch	12	Full-Speed Configuration and Interface Descriptor Length (18bytes)
1Dh	3D	Full-Speed Configuration and Interface Descriptor Word Offset (3Dh) Corresponds to EEPROM Byte Offset 7Ah
1Eh	00	RESERVED
1Fh	00	RESERVED
20h	24	Vendor ID LSB (VIDL) Register

Table 3.94 EEPROM Example - 256 Byte EEPROM (continued)

EEPROM ADDRESS	EEPROM CONTENTS (HEX)	DESCRIPTION
21h	04	Vendor ID MSB (VIDM) Register
22h	12	Product ID LSB (PIDL) Register
23h	95	Product ID MSB (PIDM) Register
24h	00	Device ID LSB (DIDL) Register
25h	01	Device ID MSB (DIDM) Register
26h	9B	Config Data Byte 1 (CFG1) Register
27h	18	Config Data Byte 2 (CFG2) Register
28h	00	Config Data Byte 3 (CFG3) Register
29h	02	Non-Removable Devices (NRD) Register
2Ah	30	Port Disable For Self-Powered Operation (PDS) Register
2Bh	30	Port Disable For Bus-Powered Operation (PDB) Register
2Ch	01	Max Power For Self-Powered Operation (MAXPS) Register
2Dh	00	Max Power For Bus-Powered Operation (MAXPB) Register
2Eh	01	Hub Controller Max Current For Self-Powered Operation (HCMCS) Register
2Fh	00	Hub Controller Max Current For Bus-Powered Operation (HCMCB) Register
30h	32	Power-On Time (PWRT) Register
31h	00	Boost_Up (BOOSTUP) Register
32h	00	RESERVED
33h	00	Boost_3:2 (BOOST32) Register
34h	00	RESERVED
35h	00	Port Swap (PRTSP) Register
36h	21	Port Remap 12 (PRTR12) Register
37h	03	Port Remap 3 (PRTR3) Register
38h	00	RESERVED
39h	01	Status/Command (STCD) Register
3A	0A	Size of Manufacturer ID String Descriptor (10 bytes)
3Bh	03	Descriptor Type (String Descriptor - 03h)
3Ch-43h	53 00 4D 00 53 00 43 00	Manufacturer ID String (“SMSC” in UNICODE)
44h	12	Size of Hi-Speed Device Descriptor in Bytes (18 bytes)
45h	01	Descriptor Type (Device Descriptor - 01h)

Table 3.94 EEPROM Example - 256 Byte EEPROM (continued)

EEPROM ADDRESS	EEPROM CONTENTS (HEX)	DESCRIPTION
46h-47h	00 02	USB Specification Number that the device complies with (0200h)
48h	FF	Class Code
49h	00	Subclass Code
4Ah	FF	Protocol Code
4Bh	40	Maximum Packet Size for Endpoint 0
4Ch-4Dh	24 04	Vendor ID (0424h)
4Eh-4Fh	00 EC	Product ID (EC00h)
50h-51h	00 01	Device Release Number (0100h)
52h	01	Index of Manufacturer String Descriptor
53h	00	Index of Product String Descriptor
54h	00	Index of Serial Number String Descriptor
55h	01	Number of Possible Configurations
56h	09	Size of Hi-Speed Configuration Descriptor in bytes (9 bytes)
57h	02	Descriptor Type (Configuration Descriptor - 02h)
58h-59h	27 00	Total length in bytes of data returned (0027h = 39 bytes)
5Ah	01	Number of Interfaces
5Bh	01	Value to use as an argument to select this configuration
5Ch	00	Index of String Descriptor describing this configuration
5Dh	E0	Self powered and remote wakeup enabled
5Eh	01	Maximum Power Consumption is 2 mA
5Fh	09	Size of Descriptor in Bytes (9 Bytes)
60h	04	Descriptor Type (Interface Descriptor - 04h)
61h	00	Number identifying this Interface
62h	00	Value used to select alternative setting
63h	03	Number of Endpoints used for this interface (Less endpoint 0)
64h	FF	Class Code
65h	00	Subclass Code
66h	FF	Protocol Code
67h	00	Index of String Descriptor Describing this interface
68h	12	Size of Full-Speed Device Descriptor in Bytes (18 Bytes)
69h	01	Descriptor Type (Device Descriptor - 01h)

Table 3.94 EEPROM Example - 256 Byte EEPROM (continued)

EEPROM ADDRESS	EEPROM CONTENTS (HEX)	DESCRIPTION
6Ah-6Bh	00 02	USB Specification Number that the device complies with (0200h)
6Ch	FF	Class Code
6Dh	00	Subclass Code
6Eh	FF	Protocol Code
6Fh	40	Maximum Packet Size for Endpoint 0
70h-71h	24 04	Vendor ID (0424h)
72h-73h	00 EC	Product ID (EC00h)
74h-75h	00 01	Device Release Number (0100h)
76	01	Index of Manufacturer String Descriptor
77h	00	Index of Product String Descriptor
78h	00	Index of Serial Number String Descriptor
79h	01	Number of Possible Configurations
7Ah	09	Size of Full-Speed Configuration Descriptor in bytes (9 bytes)
7Bh	02	Descriptor Type (Configuration Descriptor - 02h)
7Ch-7Dh	27 00	Total length in bytes of data returned (0027h = 39 bytes)
7Eh	01	Number of Interfaces
7Fh	01	Value to use as an argument to select this configuration
80h	00	Index of String Descriptor describing this configuration
81h	E0	Bus powered and remote wakeup enabled
82h	01	Maximum Power Consumption is 2 mA
83h	09	Size of Full-Speed Interface Descriptor in Bytes (9 Bytes)
84h	04	Descriptor Type (Interface Descriptor - 04h)
85h	00	Number identifying this Interface
86h	00	Value used to select alternative setting
87h	03	Number of Endpoints used for this interface (Less endpoint 0)
88h	FF	Class Code
89h	00	Subclass Code
8Ah	FF	Protocol Code
8Bh	00	Index of String Descriptor describing this interface
8Ch-FFh	-	Data storage for use by Host as desired

3.11 Device Clocking

The device requires a fixed-frequency 25MHz clock source. This is typically provided by attaching a 25MHz crystal to the XI and XO pins. The clock can optionally be provided by driving the XI input pin with a single-ended 25MHz clock source. If a single-ended source is selected, the clock input must run continuously for normal device operation.

Internally, the device generates its required clocks with a phase-locked loop (PLL). The device reduces its power consumption in several of its operating states by disabling its internal PLL and derivative clocks. The 25MHz clock remains operational in all states where power is applied.

3.12 Device Power Sources

The device may be soft powered by the USB bus or self powered via external power supplies. The following external 3.3V power supplies are required when power is not being furnished by the USB bus:

- VDD33IO, VDD33A

Note: The device also requires 1.8V, but this is supplied by an internal regulator and connection does not vary. Since the 1.8V supply is derived from VDD33IO, there is no need to discuss it separately.

3.13 Ethernet Controller Power States

The following power states are supported for the Ethernet Controller:

- UNPOWERED
- NORMAL (Unconfigured and Configured)
- Suspend (SUSPEND0, SUSPEND1, and SUSPEND2)

Figure 3.28 illustrates the power states and allowed state transitions.

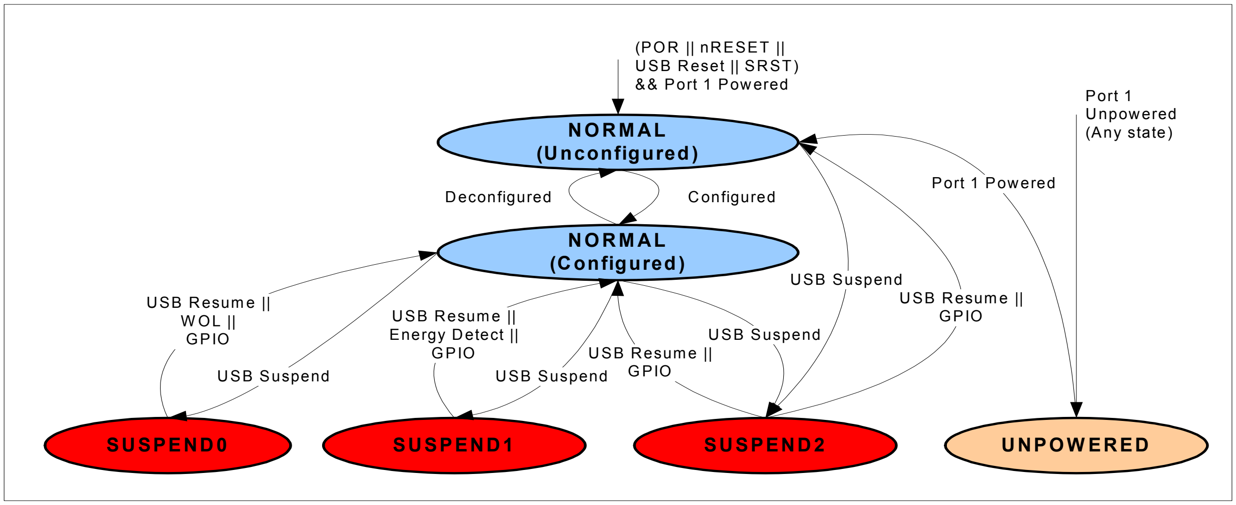


Figure 3.28 Power States

Note: It is not possible to transition from SUSPEND2 to NORMAL Configured if SUSPEND2 was entered via a transition from NORMAL Unconfigured.

3.13.1 UNPOWERED State

The UNPOWERED state provides a mechanism to conserve power when Port 1 is not powered by the Hub.

The Ethernet Controller may initially enter the UNPOWERED state when a POR occurs and USB power is not detected. This state persists until the VBUS_DET is asserted and the Hub then applies power to Port 1. The UNPOWERED state is alternatively entered whenever VBUS_DET deasserts and the Hub removes power from Port 1.

In order to make the Ethernet Controller fully operational, the Host must configure the Ethernet Controller, which places it in the NORMAL Configured state.

3.13.2 NORMAL State

The NORMAL state is the fully functional state of the Ethernet Controller. The are two flavors of the NORMAL state, NORMAL Configured and NORMAL Unconfigured. In the Configured variation, all Ethernet Controller subsystem modules are enabled. The Unconfigured variation has only a subset of the modules enabled. The reduced functionality allows for power savings.

This NORMAL state is entered by any of the following methods.

- A system reset (POR or nRESET), VBUS_DET is asserted, an the Hub powers Port 1.
- The Ethernet Controller is in the UNPOWERED state and the Hub powers Port 1.
- The Ethernet Controller is suspended and the Host issues resume signaling.
- The Ethernet Controller is suspended and a wake event is detected.

3.13.2.1 Unconfigured

Upon initially entering the NORMAL state, the Ethernet Controller is unconfigured. The Ethernet Controller transitions to the NORMAL Configured state upon the Host completion of the USB configuration.

It is possible for the Ethernet Controller to be deconfigured by the Host after being placed in the NORMAL configured state, via a set_configuration command. In this case the CPM must place the Ethernet Controller back into the NORMAL Unconfigured state.

3.13.2.2 Reset Operation

After a reset, the Ethernet Controller is placed into the NORMAL Unconfigured state. When in the NORMAL state, the READY bit in the Power Management Control Register (PMT_CTL) is set. This READY bit is useful to the Host after a USB reset occurs. In this case, it indicates that the values in the EEPROM have been completely loaded.

3.13.2.3 Suspend Operation

When returning to the NORMAL state from the SUSPEND state, the USB context is maintained. After entering the NORMAL state the READY bit in the PMT_CTL register is asserted.

Note: If the originating suspend state is SUSPEND2, the Host is required to reinitialize the Ethernet PHY registers.

3.13.3 SUSPEND States

The SUSPEND state is entered after the USB Host suspends the Ethernet Controller. Three variations of the SUSPEND state are available. Each state offers different options in terms of power consumption and wakeup support.

A SUSPEND state is entered via a transition from the NORMAL state. The SUSPEND_MODE field in the Power Management Control Register (PMT_CTL), indicates which SUSPEND state is to be used. The Host sets the value of this field to select the desired suspend state, then sends suspend signaling. A transfer back to the NORMAL state occurs when the Host sends resume signaling or a wakeup event is detected.

The Ethernet Controller can be suspended from the NORMAL Unconfigured state. In this scenario, it is only possible to transition to the SUSPEND2 state. Subsequent resume signaling or a wakeup event will cause the Ethernet Controller to transition back to the NORMAL Unconfigured state.

Note: If the Ethernet Controller is deconfigured, the SUSPEND_MODE field in the Power Management Control Register (PMT_CTL) resets to 10b.

3.13.3.1 Reset from Suspend

All suspend states must respond to a USB Reset and pin reset, nRESET. The application of these resets result in the Ethernet Controller's hardware being re-initialized and placed into the NORMAL Unconfigured state.

3.13.3.2 SUSPEND0

This state is entered from the NORMAL state when the Ethernet Controller is suspended and the SUSPEND_MODE field in the Power Management Control Register (PMT_CTL) is set to 00b.

In this state, the MAC can optionally be programmed to detect a Wake-On-Lan event or Magic Packet event. The Host may take the Ethernet Controller out of the SUSPEND0 state at any time.

3.13.3.3 SUSPEND1

This state is entered from the NORMAL state when the Ethernet Controller is suspended and the SUSPEND_MODE field in the Power Management Control Register (PMT_CTL) is set to 01b.

In this state, the Ethernet PHY can be optionally programmed for energy detect.

3.13.3.4 SUSPEND2

This state is entered from the NORMAL state when the Ethernet Controller is suspended and the SUSPEND_MODE field in the Power Management Control Register (PMT_CTL) is set to 10b or 11b. SUSPEND2 is the default suspend mode.

This state consumes the least amount of power. In this state, the Ethernet Controller may only be awakened by the Host or GPIO assertion.

The state of the Ethernet PHY is lost when entering SUSPEND2. Therefore, the Host must reinitialize the PHY after the Ethernet Controller returns to the NORMAL state.

3.14 Wake Events

The following events can wake up/enable the Ethernet Controller, depending on the power state.

- USB Host Resume
- VBUS_DET assertion and the Hub enables Port 1 Power
- Wake On LAN (Wakeup Frame and Magic Packet)
- PHY Energy Detect
- GPIO[7:0]

Table 3.95 illustrates the wake events permitted in each of the power states.

Table 3.95 Power State/Wake Event Mapping

POWER STATE	USB HOST RESUME SIGNALING	PORT 1 POWERED	WOL	PHY ENERGY DETECT	GPIO[7:0]
SUSPEND0	YES	NO	YES	NO	YES
SUSPEND1	YES	NO	NO	YES	YES
SUSPEND2	YES	NO	NO	NO	YES
UNPOWERED	NO	YES	NO	NO	NO

The occurrence of a GPIO wake event causes the corresponding bit in the INT_STS, Section 4.3.2, "Interrupt Status Register (INT_STS)", to be set. Before suspending the Ethernet Controller, the Host must ensure that any pending wake events are cleared. Otherwise, the Ethernet Controller will immediately be awakened after being suspended.

3.14.1 Detecting Wakeup Events

The Ethernet Controller supports the ability to generate remote wake events on detection of a GPIO event or Magic Packet, Wakeup Frame or Ethernet link status change (energy detect). A simplified diagram of the wake event detection logic is shown in Figure 3.29. GPIO wake detection logic for GPIOs 0-2 and GPIOs 3-7 is illustrated in Figure 3.30 and Figure 3.31, respectively.

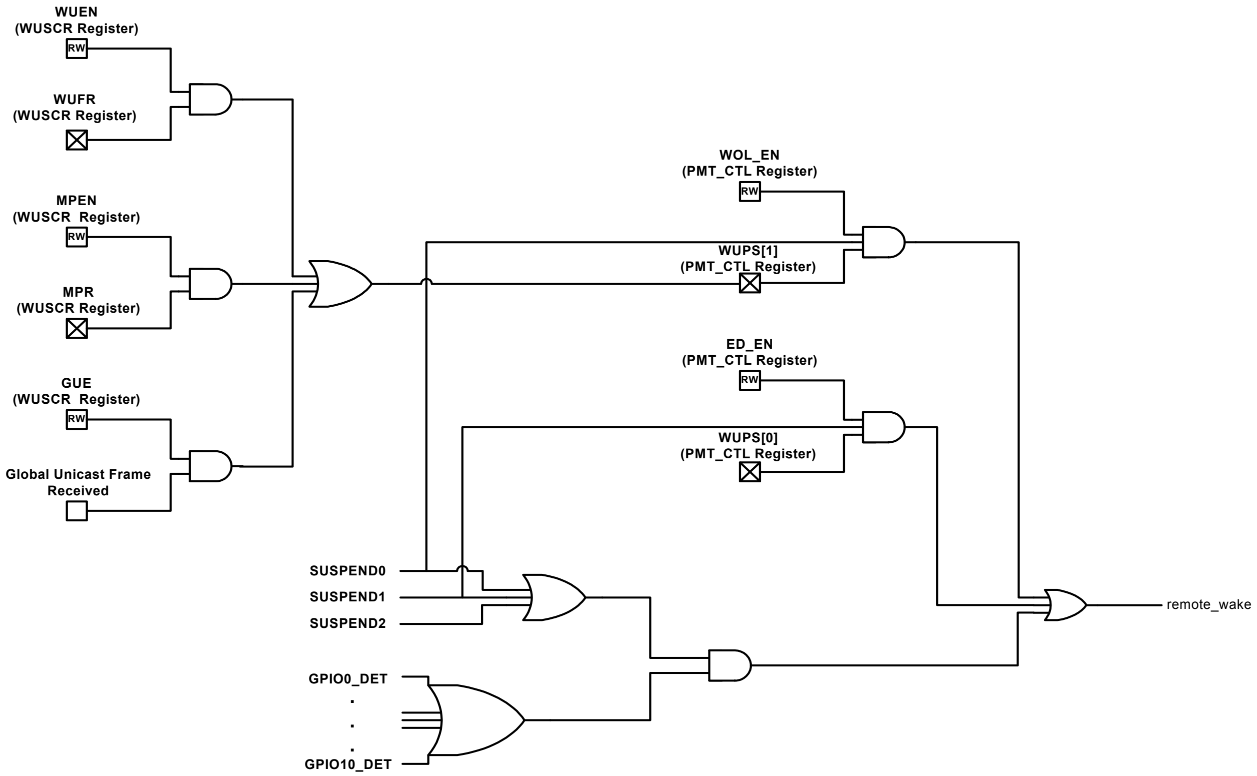


Figure 3.29 Wake Event Detection Block Diagram

Note: Diagram does not represent actual hardware implementation.

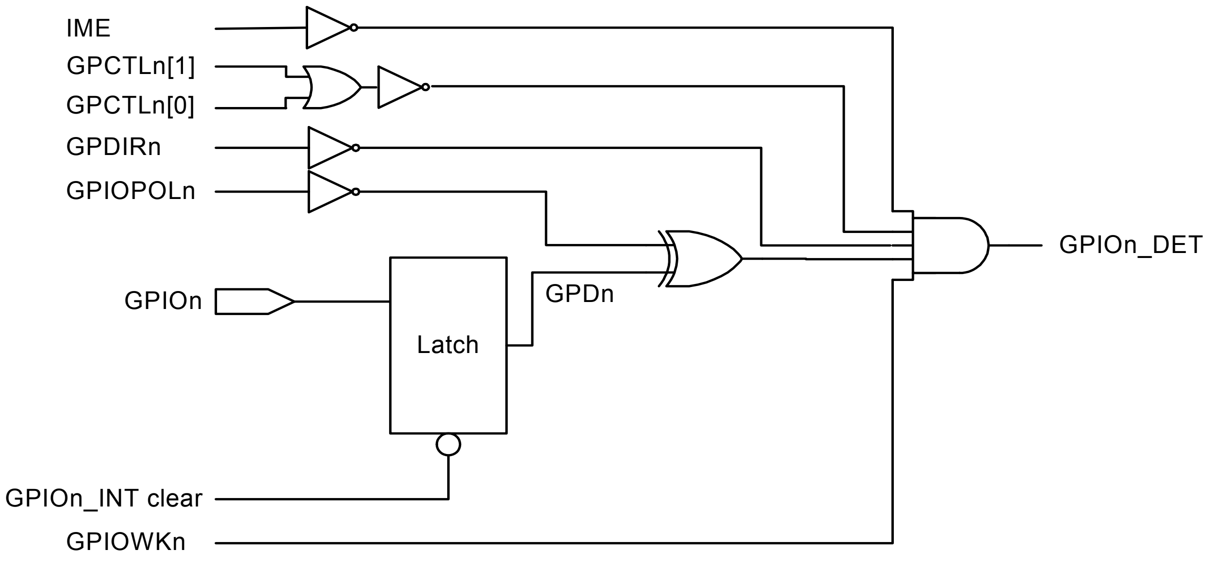


Figure 3.30 GPIOs 0-2 Wake Detection Logic

Note: The IME bit is in the Hardware Configuration Register (HW_CFG). LED General Purpose IO Configuration Register (LED_GPIO_CFG) and General Purpose IO Wake Enable and Polarity Register (GPIO_WAKE) must be set accordingly. Diagram does not represent actual hardware implementation.

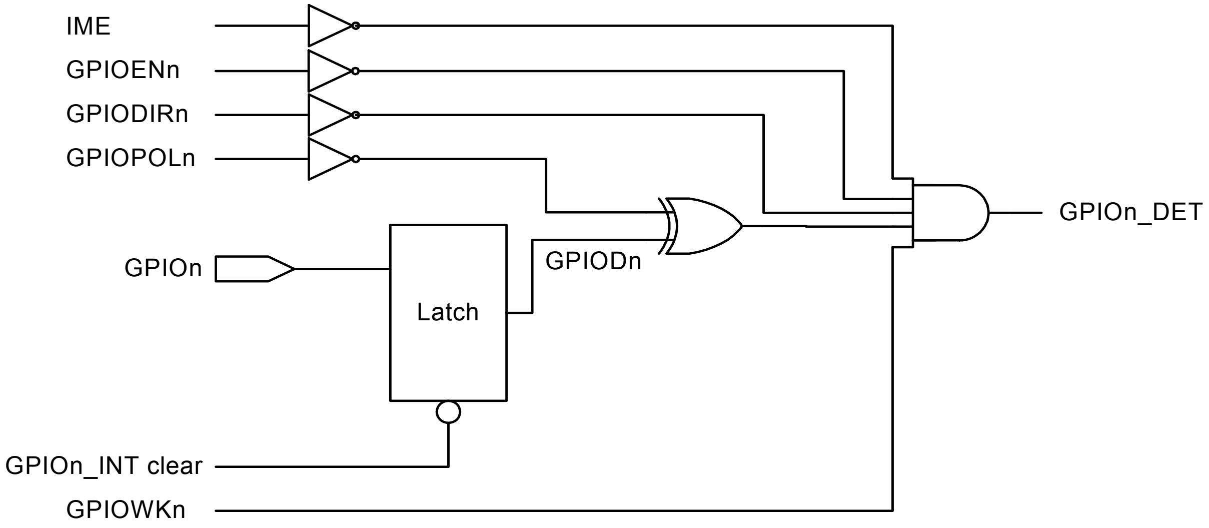


Figure 3.31 GPIOs 3-7 Wake Detection Logic

Note: The IME bit is in the Hardware Configuration Register (HW_CFG). General Purpose IO Configuration Register (GPIO_CFG) and General Purpose IO Wake Enable and Polarity Register (GPIO_WAKE) must be set accordingly. Diagram does not represent actual hardware implementation.

Three control bits are implemented in the MAC's Wakeup Control and Status Register (WUCSR) register to control Global Unicast Frame Wakeup, Magic Packet Wakeup, and Wake Up Frame Detection Wakeup: GUE, MPEN, and WUEN, respectively. A composite signal, depending on the state of these control bits and the associated event, is generated and propagated for further processing, as discussed in the following text.

Two control bits are implemented in the PMT_CTRL SCSR: Wake-on-LAN enable (WOL_EN) and Energy Detect enable (ED_EN). Depending on the state of these control bits, the logic will generate an internal wake event interrupt when the MAC detects a wakeup event (Global Unicast Frame, Wakeup Frame, or Magic Packet - depending on the state of the aforementioned composite signal), or a PHY interrupt is asserted (energy detect). Two WAKE-UP Status (WUPS) bits are implemented in the SCSR space. These bits are set depending on the corresponding wake event. (See Section 4.3.8, "Power Management Control Register (PMT_CTL)," on page 158 for further information). If a Wake-on-LAN event is detected, then further resolution on the source of the event can be obtained by examining the Remote Wakeup Frame Received (WUFR) and Magic Packet Received (MPR) status bits in the MAC's Wakeup Control and Status Register (WUCSR).

Note: Wake-on-LAN events resulting in the generation of a remote-wake event may only occur when in SUSPEND0 state.

Note: Energy Detect events resulting in the generation of a remote-wake event may only occur when in SUSPEND1 state.

Wakeup Frame detection must be enabled in the MAC before detection can occur. Likewise, the energy detect interrupt must be enabled in the PHY before this interrupt can be used as a wake event. If the Ethernet Controller is properly configured, the internal wake event interrupt will cause the assertion of the remote_wake signal on detection of a wake event.

GPIO pins 0 through 7 may cause the generation of a remote-wake event when properly configured and in any of the SUSPEND states. The pin must first be programmed to operate as a GPIO input, the polarity of the wakeup signal must be specified, and the pin must be enabled to trigger the wakeup event.

GPIO Pins 0 through 2 may be programmed to operate as GPIO inputs by clearing the GPIO x Control (GPIOCTLx) field (0<= x <=2) and GPIO Direction (GPDIR[2:0]) bit corresponding to the pin. GPIO Pins 3 through 7 may be programmed to operate as GPIO inputs by clearing the corresponding bits in the GPIO Enable 3-7 (GPIOENn) and GPIO Direction 3-7 (GPIODIRn) fields of the General Purpose IO Configuration Register (GPIO_CFG). The polarity of the wakeup signal is selected and the pin is enabled by configuring the bits associated with the pin in the GPIO Polarity 0-7 (GPIOPOLn) and GPIO Wake 0-7 (GPIOWKn) fields of the General Purpose IO Wake Enable and Polarity Register (GPIO_WAKE).

The GPIO [7:0] (GPIOx_INT) field of the Interrupt Status Register (INT_STS) contains the status bits that may be examined to determine the source of the event.

3.14.1.1 Enabling GPIO Wake Events

The Host system must perform the following steps to enable the Ethernet Controller to assert a remote_wake event on detection of a GPIO wake event.

1. All transmit and receive operations must be halted:

a. All pending Ethernet TX and RX operations must be completed.

b. The MAC must be halted.

2. The GPIO pin is programmed and enabled, as specified in Section 3.14.1 to facilitate generation of the wake event.

3. The Host places the Ethernet Controller in the any one of the SUSPEND states by setting the Suspend Mode (SUSPEND_MODE) field in the Power Management Control Register (PMT_CTL) to indicate the desired suspend state, then sends suspend signaling.

On detection of an enabled GPIO wake event, the Ethernet Controller will transition back to the NORMAL state and signal a remote_wake event. The Host may then examine the GPIO [7:0] (GPIOx_INT) status bits of the Interrupt Status Register (INT_STS) to determine the source of the wakeup.

3.14.1.2 Enabling Wakeup Frame Wake Events

The Host system must perform the following steps to enable the Ethernet Controller to assert a remote_wake event on detection of a Wakeup frame.

1. All transmit and receive operations must be halted:

a. All pending Ethernet TX and RX operations must be completed.

b. The MAC must be halted.

2. The MAC must be configured to detect the desired wake event. This process is explained in Section 3.8.5, "Wakeup Frame Detection," on page 102.

3. Bit 1 of the Wakeup Status (WUPS[1]) in the Power Management Control Register (PMT_CTL) must be cleared since a set bit will cause the immediate assertion of wake event when the Wake-On-Lan Enable (WOL_EN) bit is set. The WUPS[1] bit will not clear if the internal MAC wakeup event is asserted.

4. Set the Wake-On-Lan Enable (WOL_EN) bit in the Power Management Control Register (PMT_CTL).

5. The Host places the Ethernet Controller in the SUSPEND0 state by setting the Suspend Mode (SUSPEND_MODE) field in the Power Management Control Register (PMT_CTL) to 00b, to indicate the desired suspend state, then sends suspend signaling.

On detection of an enabled wakeup frame, the Ethernet Controller will transition back to the NORMAL state and signal a remote_wake event.

3.14.2 Enabling Link Status Change (Energy Detect) Wake Events

The Host system must perform the following steps to enable the Ethernet Controller to assert a remote_wake event on detection of an Ethernet link status change.

1. All transmit and receive operations must be halted:

a. All pending Ethernet TX and RX operations must be completed.

b. The MAC must be halted.

2. The ENERGYON event must be enabled as a PHY interrupt source. This is done by setting the INT7 bit in the PHY’s Interrupt Source Flag Register.

3. The PHY must be enabled for the energy detect power down mode This is done by clearing the EDPWRDOWN bit in the PHY’s Mode Control/Status Register. Enabling the energy detect power-down mode places the PHY in a reduced power state. In this mode of operation the PHY is not capable of receiving or transmitting Ethernet data. In this state, the PHY will assert its internal interrupt if it detects Ethernet activity. Refer to Section 3.9.8.2, "Energy Detect Power-Down," on page 122 for more information.

4. Bit 0 of the Wakeup Status (WUPS[0]) in the Power Management Control Register (PMT_CTL) must be cleared, since a set bit will cause the immediate assertion of wake event when Energy-Detect Enable (ED_EN) is set. The WUPS[0] bit will not clear if the internal PHY interrupt is asserted.

5. Set the Energy-Detect Enable (ED_EN) bit in the Power Management Control Register (PMT_CTL).

6. The Host places the Ethernet Controller in the SUSPEND1 state by setting the Suspend Mode (SUSPEND_MODE) field in the Power Management Control Register (PMT_CTL) to 01b, to indicate the desired suspend state, then sends suspend signaling.

On detection of Ethernet activity (energy), the Ethernet Controller will transition back to the NORMAL state and signal a remote_wake event.

3.15 Resets

The device has the following chip level reset sources:

- Power-On Reset (POR)
- External Chip Reset (nRESET)
- Lite Reset (LRST)
- Soft Reset (SRST)
- USB Reset
- PHY Software Reset
- nTRST
- VBUS_DET

3.15.1 Power-On Reset (POR)

A Power-On reset occurs whenever power is initially applied to the device, or if power is removed and reapplied to the device. A timer within the device will assert the internal reset for approximately 22mS.

Note: POR results in the Hub configuration registers being initialized with their images, as read from the EEPROM (0xA5 signature present), or default values (0xA5 signature absent). In addition, the Hub USB descriptors will be internally generated. No Ethernet Controller initialization from EEPROM information occurs as a result of these resets.

The Ethernet Controller may subsequently be initialized upon the Host issuing set_feature (power) for port 1. The configuration read from the EEPROM (0xA5 signature present) or the hardware defaults (Table 3.91 if 0xA5 signature absent) will be used.

Note: After the assertion of the POR, the Ethernet PHY is put into general power down mode.

3.15.2 External Chip Reset (nRESET)

A hardware reset will occur when the nRESET pin is driven low. After the nRESET is deasserted, the Ethernet Controller may be taken out of reset via a set_feature (power) for port 1 and configured via its control registers. If unused, nRESET must be tied to VDD. If used, nRESET must be driven low for a minimum period as defined in Section 5.5.2, "Reset Timing," on page 240.

Note: After the assertion of nRESET, the Ethernet PHY is put into general power down mode.

3.15.3 Lite Reset (LRST)

This reset is initiated via the LRST bit in the Section 4.3.5, "Hardware Configuration Register (HW_CFG)". It will reset the Ethernet Controller

Note: This reset does not cause the USB contents from the EEPROM to be reloaded.

Note: There is no effect on the HUB.

Note: This reset does not place the Ethernet Controller into the Unconfigured state.

Note: After the LRST, the Ethernet Controller's USB pipes corresponding to the Bulk In, Bulk Out, and Interrupt endpoints must be reset. This process entails clearing the ENDPOINT_HALT feature and resetting the data toggle on the Host side.

3.15.4 Soft Reset (SRST)

A Soft Reset is initiated by writing a ‘1’ to bit 0 of the HW_CFG register (SRST). This self-clearing bit will return to ‘0’ after approximately 2 μs, at which time the Soft Reset is complete. Soft reset does not clear control register bits marked as NASR.

Note: The Ethernet Controller will be initialized by this reset. The information read from the EEPROM (0xA5 signature present) or the defaults (Table 3.91 if 0xA5 signature absent) will be used.

Note: After the assertion of the SRST, the Ethernet PHY is put into general power down mode.

Writing SRST=1 will cause the Ethernet Controller to disconnect from the USB Hub shortly after the first good OUT Data pkt during the Data Phase. In HS mode, a brief delay will allow enough time for the Ethernet Controller to send the ACK for the Data Stage, but the Ethernet Controller will be disconnected (causing a 3-strikes timeout failure) for any next transaction (e.g., the Status Stage, or a repeated Data Stage, if there were any bus errors). In FS mode, the brief delay will be short enough that the Ethernet Controller will disconnect during the ACK pkt, causing CRC, bit-stuff, etc. errors on USB. To the USB Host, the aforementioned behaviors are the same as what happens during any Surprise Removal of a USB Device. This behavior is completely normal, and a compliant Host must be tolerant of it.

3.15.5 USB Reset

A USB reset causes a reset of the device. In response to the upstream port signaling a reset to the device, the Hub module does the following:

1. Sets default address to 0.
2. Sets configuration to: Unconfigured.
3. Negates port power to all downstream ports.
4. Clears the TT buffer RAM.
5. Moves the state of downstream devices from suspended to active (if suspended).
6. Complies with Section 11.10 of the USB 2.0 Specification for behavior after completion of the reset sequence.

The Host then configures the Hub and the Hub's downstream port devices in accordance with the USB Specification.

The Hub does not propagate the upstream USB reset to downstream devices

After a USB reset of the Ethernet Controller, the READY bit in its PMT_CTRL register can be read by the Host and will read back a ‘0’ until the EEPROM contents are loaded (provided one is present). Upon completion of the EEPROM contents load, the READY bit in PMT_CTRL is set high, and the Ethernet Controller can be configured via its control registers.

Note: This reset does not cause the USB contents from the EEPROM to be reloaded. Only the MAC address is reloaded.

Note: After the assertion of the USB Reset, the Ethernet PHY is put into general power down mode.

3.15.6 PHY Software Reset

The Ethernet PHY can be reset via two software-initiated resets. Please refer to Section 3.9.9, "PHY Resets," on page 122 for details.

3.15.7 nTRST

This active-low reset is used by the TAP controller.

3.15.8 VBUS_DET

The removal of USB power causes the Ethernet Controller to transition to the UNPOWERED state.

Note: After transitioning out of the UNPOWERED state, the Ethernet PHY is in general power down mode.

Note: Transition from the UNPOWERED state causes a reset.

Chapter 4 Register Descriptions

4.1 Register Nomenclature

Table 4.1 describes the register bit attributes used throughout this document.

Table 4.1 Register Bit Types

REGISTER BIT TYPE NOTATION	REGISTER BIT DESCRIPTION
R	**Read:** A register or bit with this attribute can be read.
W	**Write:** A register or bit with this attribute can be written.
RO	**Read only:** Read only. Writes have no effect.
RS	**Read to Set:** This bit is set on read.
WO	**Write only:** If a register or bit is write-only, reads will return unspecified data.
WC	**Write One to Clear:** writing a one clears the value. Writing a zero has no effect.
WAC	**Write Anything to Clear:** writing anything clears the value.
RC	**Read to Clear:** Contents is cleared after the read. Writes have no effect.
LL	**Latch Low:** Clear on read of register.
LH	**Latch High:** Clear on read of register.
SC	**Self-Clearing:** Contents is self-cleared after the being set. Writes of zero have no effect. Contents can be read.
RO/LH	**Read Only, Latch High:** This mode is used by the Ethernet PHY registers. Bits with this attribute will stay high until the bit is read. After it a read, the bit will remain high, but will change to low if the condition that caused the bit to go high is removed. If the bit has not been read the bit will remain high regardless of if its cause has been removed.
NASR	**Not Affected by Software Reset.** The state of NASR bits does not change on assertion of a software reset.
RESERVED	**Reserved Field:** Reserved fields must be written with zeros, unless otherwise indicated, to ensure future compatibility. The value of reserved bits is not guaranteed on a read.

4.2 Register Memory Map

Table 4.2 LAN9512/LAN9512i Register Memory Map

ADDRESS	SYMBOL	REGISTER NAME
000h - 0FFh	SCSR	System Control and Status Registers
100h - 1FCh	MCSR	MAC Control and Status Registers
3000h - 30FFh	HCFG	Hub Configuration Registers

4.3 System Control and Status Registers

Table 4.3 LAN9512/LAN9512i Device Control and Status Register Map

ADDRESS	SYMBOL	REGISTER NAME
000h	ID_REV	Device ID and Revision Register
004h	RESERVED	Reserved for future expansion
008h	INT_STS	Interrupt Status Register
00Ch	RX_CFG	Receive Configuration Register
010h	TX_CFG	Transmit Configuration Register
014h	HW_CFG	Hardware Configuration Register
018h	RX_FIFO_INF	Receive FIFO Information Register
01Ch	TX_FIFO_INF	Transmit FIFO Information Register
020h	PMT_CTL	Power Management Control Register
024h	LED_GPIO_CFG	LED General Purpose IO Configuration Register
028h	GPIO_CFG	General Purpose IO Configuration Register
02Ch	AFC_CFG	Automatic Flow Control Configuration Register
030h	E2P_CMD	EEPROM Command Register
034h	E2P_DATA	EEPROM Data Register
038h	BURST_CAP	Burst Cap Register
03Ch	RESERVED	Reserved for future expansion
040h	DP_SEL	Data Port Select Register
044h	DP_CMD	Data Port Command Register
048h	DP_ADDR	Data Port Address Register
04Ch	DP_DATA0	Data Port Data 0 Register
050h	DP_DATA1	Data Port Data 1 Register
054h – 060h	RESERVED	Reserved for future expansion
064h	GPIO_WAKE	General Purpose IO Wake Enable and Polarity Register
068h	INT_EP_CTL	Interrupt Endpoint Control Register
06Ch	BULK_IN_DLY	Bulk In Delay Register
070h – 0FFh	RESERVED	Reserved for future expansion

4.3.1 Device ID and Revision Register (ID_REV)

Address: 000h Size: 32 bits

BITS	DESCRIPTION	TYPE	DEFAULT
31:16	**Chip ID** This read-only field identifies the device model.	RO	EC00h
15:0	**Chip Revision** This is the revision of the device.	RO	Note 4.1

Note 4.1 Default value is dependent on device revision.

4.3.2 Interrupt Status Register (INT_STS)

Address:	008h	Size:	32 bits

BITS	DESCRIPTION	TYPE	DEFAULT
31:19	**RESERVED**	RO	-
18	**MAC Reset Time Out (MACRTO_INT)** This interrupt signifies that the 8 ms reset watchdog timer has timed out. This means that the Ethernet PHY is not supplying the rx_clk or tx_clk. After the timer times out, the MAC reset is deasserted asynchronously.	R/WC	0b
17	**TX Stopped (TXSTOP_INT)** This interrupt is issued when the Stop Transmitter (STOP_TX) bit in Transmit Configuration Register (TX_CFG) is set and the transmitter is halted. **Note:** The source of this interrupt is a pulse.	R/WC	0b
16	**RX Stopped (RXSTOP_INT)** This interrupt is issued when the receiver is halted. **Note:** The source of this interrupt is a pulse.	R/WC	0b
15	**PHY Interrupt (PHY_INT)** Indicates a PHY Interrupt event. **Note:** The source of this interrupt is a level. The interrupt persists until it is cleared in the PHY.	RO	-
14	**Transmitter Error (TXE)** When generated, indicates that the transmitter has encountered an error. Refer to Section 3.7.2.5, "TX Error Detection" for a description of the conditions that will cause a TXE. **Note:** The source of this interrupt is a pulse.	R/WC	0b
13	**TX Data FIFO Underrun Interrupt (TDFU)** Generated when the TX Data FIFO underruns. **Note:** The source of this interrupt is a pulse.	R/WC	0b
12	**TX Data FIFO Overrun Interrupt (TDFO)** Generated when the TX Data FIFO is full, and another write is attempted. **Note:** This interrupt should never occur and indicates a catastrophic hardware error. **Note:** The source of this interrupt is a pulse.	R/WC	0b
11	**RX Dropped Frame Interrupt (RXDF_INT)** This interrupt is issued whenever a receive frame is dropped. **Note:** The source of this interrupt is a pulse.	R/WC	0b
10:8	**RESERVED**	RO	-
7:0	**GPIO [7:0] (GPIOx_INT)** Interrupts are generated from the GPIOs. These interrupts are configured through the GPIO_CFG and LED_GPIO_CFG registers. **Note:** The sources for these interrupts are a level.	R/WC Note 4.3	Note 4.2

Note 4.2 The default depends on the state of the GPIO pin.

Note 4.3 The clearing of a GPIOx_INT bit also clears the corresponding GPIO wake event.

4.3.3 Receive Configuration Register (RX_CFG)

Address:	00Ch	Size:	32 bits

BITS	DESCRIPTION	TYPE	DEFAULT
31:1	**RESERVED**	RO	-
0	**Receive FIFO Flush** Setting this bit will reset the RX FIFO pointers.	SC	0b

4.3.4 Transmit Configuration Register (TX_CFG)

Address:	010h	Size:	32 bits

BITS	DESCRIPTION	TYPE	DEFAULT
31:3	**RESERVED**	RO	-
2	**Transmitter Enable (TX_ON)** When this bit is set, the transmitter is enabled. Any data in the TX FIFO will be sent. This bit is cleared automatically when STOP_TX is set and the transmitter is halted.	R/W	0b
1	**Stop Transmitter (STOP_TX)** When this bit is set, the transmitter will finish the current frame being read from the TX FIFO, and will then stop transmitting. When the transmitter has stopped, this bit will clear. All writes to this bit are ignored while this bit is high. **Note:** After this bit clears, there will be no TX Ethernet frame data in the TIC, TLI, or MAC.	SC	0b
0	**Transmit FIFO Flush** Setting this bit will reset the TX FIFO pointers.	SC	0b

4.3.5 Hardware Configuration Register (HW_CFG)

Address:	014h	Size:	32 bits

BITS	DESCRIPTION	TYPE	DEFAULT
31:13	**RESERVED**	RO	-
12	**Bulk In Empty Response (BIR)** This bit controls the response to Bulk IN tokens when the RX FIFO is empty. 0 = Respond to the IN token with a ZLP 1 = Respond to the IN token with a NAK	R/W	0b
11	**Activity LED 80 ms Bypass (LEDB)** When set, the Activity LED on/off time is reduced to approximately 15us/15us.	R/W	0b
10:9	**RX Data Offset (RXDOFF)** This field controls the amount of offset, in bytes, that is added to the beginning of an RX Data packet. The start of the valid data will be shifted by the amount of bytes specified in this field. An offset of 0-3 bytes is a valid number of offset bytes. **Note:** This register may not be modified after the RX datapath has been enabled.	R/W	00b
8	**Stall Bulk Out Pipe Disable (SBP)** This bit controls the operation of the Bulk Out pipe when the FCT detects the loss of sync condition. Please refer to Section 3.7.2.5, "TX Error Detection" for details. 0 = Stall the Bulk Out pipe when loss of sync detected. 1 = Do not stall the Bulk Out pipe when loss of sync detected.	R/W	0b
7	**Internal MII Visibility Enable (IME)** This register enables a subset of the MII interface to be visible on unused pins. See Table 4.4, “IME Mapping,” on page 155 for the signals affected by this bit. 0 = The MII signals are not visible. The MII pins function as inputs. 1 = The MII signals are visible. The MII pins function as outputs. **Note:** The IME has priority over the GPIO_CFG register.	RW	0b
6	**Discard Errored Received Ethernet Frame (DRP)** This bit will cause errored Ethernet frames to be discarded when enabled. 0 = Do not discard errored Ethernet frames 1 = Discard errored Ethernet frames.	R/W	0b
5	**Multiple Ethernet Frames per USB Packet (MEF)** This bit enables the UTX to pack multiple Ethernet frames per USB packet whenever possible. 0 = Support no more than one Ethernet frame per USB packet 1 = Support packing multiple Ethernet frames per USB packet **Note:** The URX supports this mode by default.	R/W	0b

BITS	DESCRIPTION	TYPE	DEFAULT
4	**EEPROM Time-out Control (ETC)** This bit controls the length of time used by the EEPOM controller to detect a time-out. 0 = Time-out occurs if no response received from EEPROM after 30 ms. 1 = Time-out occurs if no response received from EEPROM after 1.28 us.	R/W	0b
3	**Soft Lite Reset (LRST)** Writing 1 generates the lite software reset of the device. A lite reset will not affect the UDC or the datapath functionality of the CTL block. Additionally, the contents of the EEPROM will not be reloaded. This bit clears after the reset sequence has completed.	SC	0b
2	**RESERVED**	RO	-
1	**Burst Cap Enable (BCE)** This register enables use of the burst cap register, Section 4.3.14, "Burst Cap Register (BURST_CAP)". 0 = Burst Cap register is not used to limit the TX burst size. 1 = Burst Cap register is used to limit the TX burst size.	R/W	0b
0	**Soft Reset (SRST)** Writing 1 generates a software initiated reset of the Ethernet Controller. A software reset will result in the contents of the EEPROM being reloaded for the Ethernet Controller (EC). While the reset sequence is in progress, the EC will be disconnected from the Hub. After the EC has been reinitialized, it will take it's USB controller out of the disconnect state and be visible to the Hub.	SC	0b

.

Table 4.4 IME Mapping

PACKAGE PIN	INTERNAL MII SIGNAL
GPIO3	TXEN
GPIO4	TX_CLK
GPIO5	RX_CLK
GPIO6	RXDO
GPIO7	RXD1
TEST1	RXDV
TEST4	nPHY_RST

4.3.6 Receive FIFO Information Register (RX_FIFO_INF)

Address:	018h	Size:	32 bits

BITS	DESCRIPTION	TYPE	DEFAULT
31:16	**RESERVED**	RO	-
15:0	**RX Data FIFO Used Space (RXDUSED)** Reads the amount of space in bytes used in the RX Data FIFO. For each receive frame, this field is incremented by the length of the receive data rounded up to the nearest DWORD (if the payload does not end on a DWORD boundary).	RO	0000h

4.3.7 Transmit FIFO Information Register (TX_FIFO_INF)

Address: 01Ch Size: 32 bits

BITS	DESCRIPTION	TYPE	DEFAULT
31:16	**RESERVED**	RO	-
15:0	**TX Data FIFO Free Space (TDFREE)** Reads the amount of space, in bytes, available in the TX Data FIFO.	RO	2000h

4.3.8 Power Management Control Register (PMT_CTL)

Address:	020h	Size:	32 bits

This register controls the power management features.

.

BITS	DESCRIPTION	TYPE	DEFAULT
31:10	**RESERVED**	RO	-
9	**Resume Clears Remote Wakeup Status (RES_CLR_WKP_STS)** When set, the Remote Wakeup Frame Received (WUFR) and Magic Packet Received (MPFR) status signals in the MAC WUCSR will clear upon the completion of a resume sequence. When set, this bit also affects the WUPS field. WUPS[1] will clear upon completion of a resume event. Only resume sequences initiated by a wakeup frame or magic packet are affected by RES_CLR_WKP_STS. When cleared, the wakeup status signals are not cleared after a resume.	R/W	0b
8	**Resume Clears Remote Wakeup Enables (RES_CLR_WKP_EN)** When asserted, all wakeup enable bits are cleared after a resume sequence, initiated from a remote wakeup, completes. Resumes initiated by the Host do not clear the wakeup enables.	R/W	1b
7	**Device Ready (READY)** When set, this bit indicates that the Ethernet Controller is in the NORMAL Configured state and the initial hardware configuration of the Ethernet Controller has completed. **Note:** This bit is useful for events (USB Reset) that do not trigger a soft disconnect. **Note:** In the case where no PHY clocks are present to complete a system reset, this bit will not be set until the watchdog timer expires. This is applicable for a Lite Reset and when transitioning to the Normal Configured state.	RO	0b
6:5	**Suspend Mode (SUSPEND_MODE)** Indicates which suspend power state to use after the Host suspends the Ethernet Controller. If the Ethernet Controller is deconfigured, it transitions to the NORMAL Unconfigured state and this register will reset to the value 10b. SUSPEND_MODE encoding: 00 = SUSPEND0 01 = SUSPEND1 10 = SUSPEND2 11 = SUSPEND2 **Note:** It is not valid to select any suspend variant besides SUSPEND2 when in the NORMAL Unconfigured state.	R/W	10b

BITS	DESCRIPTION	TYPE	DEFAULT
4	**PHY Reset (PHY_RST)** Writing a '1' to this bit resets the PHY. The internal logic automatically holds the PHY reset for a minimum of 2 ms. When the PHY is released from reset, this bit is automatically cleared. All writes to this bit are ignored while this bit is high. **Note:** The Ethernet Controller will NAK all USB transfers until the PHY reset completes.	SC	0b
3	**Wake-On-Lan Enable (WOL_EN)** Enables WOL as a wakeup event.	R/W	0b
2	**Energy-Detect Enable (ED_EN)** Enables Energy-Detect as a wakeup event.	R/W	0b
1:0	**WAKE-UP Status (WUPS)** This field indicates the cause of the current wake-up event. The WUPS field (both bits) are cleared by writing a 1 to either, or both bits. The encoding of these bits is as follows: 00 = No wake-up event detected 01 = Energy-Detect 10 = Wake-On-Lan 11 = Indicates multiple events occurred The WUPS field will not be set unless the corresponding event is enabled prior to entering the reduced power state. If the RES_CLR_WKP_STS bit is set, WUPS[1] will clear upon completion of a resume. See the RES_CLR_WKP_STS bit for further details.	R/WC	00b

4.3.9 LED General Purpose IO Configuration Register (LED_GPIO_CFG)

Address:	024h	Size:	32 bits

This register configures the external GPIO[2:0] pins.

In order for a GPIO to function as a wake event or interrupt source, it must be configured as an input. GPIO pins used to generate wake events must also be enabled by the GPIO_WAKE register, see Section 4.3.20, "General Purpose IO Wake Enable and Polarity Register (GPIO_WAKE)".

BITS	DESCRIPTION	TYPE	DEFAULT
31:26	**RESERVED**	RO	-
25:24	**GPIO 2 Control (GPCTL2)** The value of this field determines the function of the external GPIO2 pin as follows: 00 = GPIO2 01 = nSPD_LED (Ethernet speed indicator LED) 10 = CRS 11 = TXD2 **Note:** When enabled as CRS or TXD2, the external device pin will reflect the state of the corresponding internal MII signal. This feature is useful as a diagnostic tool.	R/W	00b
23:22	**RESERVED**	RO	-
21:20	**GPIO 1 Control (GPCTL1)** The value of this field determines the function of the external GPIO1 pin as follows: 00 = GPIO1 01 = nLNKA_LED (Ethernet link activity LED) 10 = COL 11 = TXD1 **Note:** When enabled as COL or TXD1, the external device pin will reflect the state of the corresponding internal MII signal. This feature is useful as a diagnostic tool.	R/W	00b
19:18	**RESERVED**	RO	-
17:16	**GPIO 0 Control (GPCTL0)** The value of this field determines the function of the external GPIO0 pin as follows: 00 = GPIO0 01 = nFDX_LED (Ethernet full-duplex LED) 10 = TXD3 11 = TXD0 **Note:** When enabled as TXD3 or TXD0, the external device pin will reflect the state of the corresponding internal MII signal. This feature is useful as a diagnostic tool.	R/W	00b
15:11	**RESERVED**	RO	-

BITS	DESCRIPTION	TYPE	DEFAULT
10:8	**GPIO Buffer Type (GPBUF[2:0])** When set, the output buffer for the corresponding GPIO signal is configured as a push/pull driver. When cleared, the corresponding GPIO signal is configured as an open-drain driver. Bits are assigned as follows: GPBUF2 – bit 2 GPBUF1 – bit 1 GPBUF0 – bit 0	R/W	000b
7	**RESERVED**	RO	-
6:4	**GPIO Direction (GPDIR[2:0])** When set, enables the corresponding GPIO as an output. When cleared the GPIO is enabled as an input. Bits are assigned as follows: GPDIR2 – bit 2 GPDIR1 – bit 1 GPDIR0 – bit 0	R/W	000b
3	**RESERVED**	RO	-
2:0	**GPIO Data (GPD[2:0])** When enabled as an output, the value written is reflected on GPIOn. When read, GPIOn reflects the current state of the corresponding GPIO pin. Bits are assigned as follows: GPD2 – bit 2 GPD1 – bit 1 GPD0 – bit 0	R/W	Note 4.4

Note 4.4 The default value depends on the state of the GPIO pin.

4.3.10 General Purpose IO Configuration Register (GPIO_CFG)

Address:	028h	Size:	32 bits

This register configures GPIOs 3-7.

In order for a GPIO to function as a wake event or interrupt source, it must be configured as an input. GPIOs used as wake events must also be enabled by the GPIO_WAKE register, see Section 4.3.20, "General Purpose IO Wake Enable and Polarity Register (GPIO_WAKE)".

BITS	DESCRIPTION	TYPE	DEFAULT
31:29	**RESERVED**	RO	-
28:24	**GPIO Enable 3-7 (GPIOENn)** A '1' sets the associated pin to use the default function. When cleared low, the pin functions as a GPIO signal. GPIO3 - GPIO7 can be used to mirror internal MII signals when not enabled. See the IME bit in Section 4.3.5, "Hardware Configuration Register (HW_CFG)" GPIOEN3 - bit 24 GPIOEN4 - bit 25 GPIOEN5 - bit 26 GPIOEN6 - bit 27 GPIOEN7 - bit 28 **Note:** These GPIOs are disabled after a reset.	R/W	11111b
23:21	**RESERVED**	RO	-
20:16	**GPIO Buffer Type 3-7 (GPIOBUFn)** When set, the output buffer for the corresponding GPIO signal is configured as a push/pull driver. When cleared, the corresponding GPIO signal is configured as an open-drain driver. GPIOBUF3 - bit 16 GPIOBUF4 - bit 17 GPIOBUF5 - bit 18 GPIOBUF6 - bit 19 GPIOBUF7 - bit 20	R/W	00000b
15:13	**RESERVED**	RO	-
12:8	**GPIO Direction 3-7 (GPIODIRn)** When set, enables the corresponding GPIO as output. When cleared, the GPIO is enabled as an input. GPIODIR3 - bit 8 GPIODIR4 - bit 9 GPIODIR5 - bit 10 GPIODIR6 - bit 11 GPIODIR7 - bit 12	R/W	00000b
7:5	**RESERVED**	RO	-

BITS	DESCRIPTION	TYPE	DEFAULT
4:0	**GPIO Data 0-7 (GPIODn)** When enabled as an output, the value written is reflected on GPIOn. When read, GPIOn reflects the current state of the corresponding GPIO pin. GPIOD3 - bit 0 GPIOD4 - bit 1 GPIOD5 - bit 2 GPIOD6 - bit 3 GPIOD7 - bit 4	R/W	Note 4.5

Note 4.5 The default value depends on the state of the GPIO pin.

4.3.11 Automatic Flow Control Configuration Register (AFC_CFG)

Address:	02Ch	Size:	32 bits

This register configures the mechanism that controls both the automatic, and software-initiated transmission of pause frames and back pressure. Refer to Section 3.8.1, "Flow Control," on page 98 for more information on flow control operation.

Note: The device will not transmit pause frames or assert back pressure if the transmitter is disabled.

BITS	DESCRIPTION	TYPE	DEFAULT
31:24	**RESERVED**	RO	-
23:16	**Automatic Flow Control High Level (AFC_HI)** Specifies, in multiples of 64 bytes, the level at which flow control will trigger. When this limit is reached, the chip will apply back pressure or will transmit a pause frame, as programmed in bits [3:0] of this register. During full-duplex operation, only a single pause frame is transmitted when this level is reached. The pause time transmitted in this frame is programmed in the Pause Time (FCPT) field of the Flow Control Register (FLOW), contained in the MAC CSR space. During half-duplex operation, each incoming frame that matches the criteria in bits [3:0] of this register will be jammed for the period set in the BACK_DUR field.	R/W	00h
15:8	**Automatic Flow Control Low Level (AFC_LO)** Specifies, in multiples of 64 bytes, the level at which a pause frame is transmitted with a pause time setting of zero. When the amount of data in the RX Data FIFO falls below this level, the pause frame is transmitted. A pause time value of zero instructs the other transmitting device to immediately resume transmission. The zero time pause frame will only be transmitted if the RX Data FIFO had reached the AFC_HI level and a pause frame was sent. A zero pause time frame is sent whenever automatic flow control in enabled in bits [3:0] of this register.	R/W	00h

BITS	DESCRIPTION	TYPE	DEFAULT
7:4	**Back pressure Duration (BACK_DUR)** This field is used to select the time period for the Back pressure Duration Timer. This field has no function in full-duplex mode. **Note:** Back pressure Duration is slightly greater in 10Mbs mode. Back pressure Duration 100 Mbps Mode: 0h = 5 us 1h = 10 us 2h = 15 us 3h = 25 us 4h = 50 us 5h = 100 us 6h = 150 us 7h = 200 us 8h = 250 us 9h = 300 us Ah = 350 us Bh = 400 us Ch = 450 us Dh = 500 us Eh = 550 us Fh = 600 us 10 Mbps Mode: 0h = 7.2 us 1h = 12.2 us 2h = 17.2 us 3h = 27.2 us 4h = 52.2 us 5h = 102.2 us 6h = 152.2 us 7h = 202.2 us 8h = 252.2 us 9h = 302.2 us Ah = 352.2 us Bh = 402.2 us Ch = 452.2 us Dh = 502.2 us Eh = 552.2 us Fh = 602.2 us	R/W	0h
3	**Flow Control on Multicast Frame (FCMULT)** When this bit is set, back pressure will be asserted when the AFC level is reached and a multicast frame is received. This field has no function in full-duplex mode.	R/W	0b
2	**Flow Control on Broadcast Frame (FCBRD)** When this bit is set, back pressure will be asserted when the AFC level is reached and a broadcast frame is received. This field has no function in full-duplex mode.	R/W	0b
1	**Flow Control on Address Decode (FCADD)** When this bit is set, back pressure will be asserted when the AFC level is reached and a frame addressed to the device is received. This field has no function in full-duplex mode.	R/W	0b

BITS	DESCRIPTION	TYPE	DEFAULT
0	**Flow Control on Any Frame (FCANY)** When this bit is set, the device will assert back pressure, or transmit a pause frame when the AFC level is reached and any frame is received. Setting this bit enables full-duplex flow control when the device is operating in full-duplex mode. When this mode is enabled during half-duplex operation, the Flow Controller does not decode the MAC address and will send a pause frame upon receipt of a valid preamble (i.e., immediately at the beginning of the next frame after the RX Data FIFO level is reached). When this mode is enabled during full-duplex operation, the Flow Controller will immediately instruct the MAC to send a pause frame when the RX Data FIFO level is reached. The MAC will queue the pause frame transmission for the next available window. Setting this bit overrides bits [3:1] of this register.	R/W	0b

4.3.12 EEPROM Command Register (E2P_CMD)

Address:	030h	Size:	32 bits

This register is used to control the read and write operations on the Serial EEPROM.

BITS	DESCRIPTION	TYPE	DEFAULT
31	**EPC Busy** When a "1" is written into this bit, the operation specified in the EPC Command field is performed at the specified EEPROM address. This bit will remain set until the operation is complete. In the case of a read, this means that the Host can read valid data from the E2P Data register. The E2P_CMD and E2P_DATA registers should not be modified until this bit is cleared. In the case where a write is attempted and an EEPROM is not present, the EPC Busy remains busy until the EPC Time-out occurs. At that time, the busy bit is cleared. **Note:** EPC busy will be high immediately following power-up, chip-level, or USB reset. After the EEPROM controller has finished reading (or attempting to read) the USB Descriptors and Ethernet default register values, the EPC Busy bit is cleared.	SC	0b

BITS	DESCRIPTION	TYPE	DEFAULT
30:28	**EPC Command** This field is used to issue commands to the EEPROM controller. The EPC will execute commands when the EPC Busy bit is set. A new command must not be issued until the previous command completes. This field is encoded as follows: 000 = READ 001 = EWDS 010 = EWEN 011 = WRITE 100 = WRAL 101 = ERASE 110 = ERAL 111 = RELOAD **READ (Read Location):** This command will cause a read of the EEPROM location pointed to by EPC Address. The result of the read is available in the E2P_DATA register. **EWDS (Erase/Write Disable):** After issued, the EEPROM will ignore erase and write commands. To re-enable erase/write operations, issue the EWEN command. **EWEN (Erase/Write Enable):** Enables the EEPROM for erase and write operations. The EEPROM will allow erase and write operations until the Erase/Write Disable command is sent, or until power is cycled. **Note:** The EEPROM device will power-up in the erase/write-disabled state. Any erase or write operations will fail until an Erase/Write Enable command is issued. **WRITE (Write Location):** If erase/write operations are enabled in the EEPROM, this command will cause the contents of the E2P_DATA register to be written to the EEPROM location selected by the EPC Address field. **WRAL (Write All):** If erase/write operations are enabled in the EEPROM, this command will cause the contents of the E2P_DATA register to be written to every EEPROM memory location. **ERASE (Erase Location):** If erase/write operations are enabled in the EEPROM, this command will erase the location selected by the EPC Address field. **ERAL (Erase All):** If erase/write operations are enabled in the EEPROM, this command will initiate a bulk erase of the entire EEPROM. **RELOAD (Data Reload):** Instructs the EEPROM controller to reload the data from the EEPROM. If a value of A5h is not found in the first address of the EEPROM, the EEPROM is assumed to be un-programmed and the Reload operation will fail. The “Data Loaded” bit indicates a successful load of the data.	R/W	000b
27:11	**RESERVED**	RO	-
10	**EPC Time-out** If an EEPROM operation is performed, and there is no response from the EEPROM within 30mS, the EEPROM controller will time-out and return to its idle state. This bit is set when a time-out occurs, indicating that the last operation was unsuccessful. **Note:** If the EEDI pin is pulled-high (default if left unconnected), EPC commands will not time out if the EEPROM device is missing. In this case, the EPC Busy bit will be cleared as soon as the command sequence is complete. It should also be noted that the ERASE, ERAL, WRITE and WRAL commands are the only EPC commands that will time-out if an EEPROM device is not present and the EEDI signal is pulled low.	R/WC	0

BITS	DESCRIPTION	TYPE	DEFAULT
9	**Data Loaded** When set, this bit indicates that a valid EEPROM was found, and that the USB and Ethernet Data programming has completed normally. This bit is set after a successful load of the data after power-up, or after a RELOAD command has completed.	R/WC	0
8:0	**EPC Address** The 9-bit value in this field is used by the EEPROM Controller to address a specific memory location in the Serial EEPROM. This is a BYTE aligned address.	R/W	00h

4.3.13 EEPROM Data Register (E2P_DATA)

Address: 034h Size: 32 bits

This register is used in conjunction with the E2P_CMD register to perform read and write operations to the Serial EEPROM.

BITS	DESCRIPTION	TYPE	DEFAULT
31:8	**RESERVED**	RO	-
7:0	**EEPROM Data** Value read from or written to the EEPROM.	R/W	00h

4.3.14 Burst Cap Register (BURST_CAP)

Address: 038h Size: 32 bits

This register is used to limit the size of the data burst transmitted by the UTX. When more than the amount specified in the BURST_CAP register is transmitted, the UTX will send a ZLP.

Note: This register must be enabled through the Section 4.3.5, "Hardware Configuration Register (HW_CFG)".

BITS	DESCRIPTION	TYPE	DEFAULT
31:8	**RESERVED**	RO	-
7:0	**BURST_CAP** The maximum amount of contiguous data that may be transmitted by the UTX before a ZLP is sent. This field has units of 512 bytes for HS mode and 64 bytes for FS mode. **Note:** A value less than or equal to 4 in HS mode or less than or equal to 32 in FS mode indicates that burst cap enforcement is disabled. In this case, the UTX always responds to In Tokens with a ZLP when the Bulk In Empty Response (BIR) bit in the Hardware Configuration Register (HW_CFG) is deasserted. It will respond with NAKs if the Bulk In Empty Response (BIR) bit is set.	R/W	00h

4.3.15 Data Port Select Register (DP_SEL)

Address: 040h Size: 32 bits

Before accessing the internal RAMs, the TESTEN bit must be set. It is not valid to use the RAM data port during run time.

The RAM Test Mode Select chooses which internal RAM to access.

The Data Port Ready bit indicates when the data port RAM access has completed. In the case of a read operation, this indicates when the read data has been stored in the DP_DATA register.

BITS	DESCRIPTION	TYPE	DEFAULT
31	**Data Port Ready (DPRDY)** 0 = data port is busy processing a transaction 1 = data port is ready	RO	1b
30:4	**RESERVED**	RO	-
3:1	**RAM Test Select (RSEL)** Selects which RAM to access. 00b: FCT Data RAM 01b: EEPROM storage RAM 10b: TX TLI RAM 11b: RX TLI RAM 100b: TT RAM 101b: TT RAM 110b: TT RAM 111b: TT RAM	R/W	0b
0	**RAM Test Mode Enable (TESTEN**) Put all test accessible RAMs in test mode.	R/W	0b

4.3.16 Data Port Command Register (DP_CMD)

Address:	044h	Size:	32 bits

This register commences the data port access. Writing a one to this register will enable a write access, while writing a zero will do a read access.

The address and data registers need to be configured appropriately for the desired read or write operation before accessing this register.

BITS	DESCRIPTION	TYPE	DEFAULT
31:1	**RESERVED**	RO	-
0	**Data Port Write**. Selects operation. Writing to this bit initiates the data port access. 0 = read operation 1 = write operation	R/W	0b

4.3.17 Data Port Address Register (DP_ADDR)

Address:	048h	Size:	32 bits

Indicates the address to be used for the data port access.

BITS	DESCRIPTION	TYPE	DEFAULT
31:15	**RESERVED**	RO	-
14:0	**Data Port Address[14:0]**	R/W	0000h

4.3.18 Data Port Data 0 Register (DP_DATA0)

Address:	04Ch	Size:	32 bits

The Data Port Data register holds the write data for a write access and the resultant read data for a read access.

Before reading this register for the result of a read operation, the Data Port Ready bit should be checked. The Data Port Ready bit must indicate the data port is ready. Otherwise the read operation is still in progress.

BITS	DESCRIPTION	TYPE	DEFAULT
31:0	**Data Port Data [31:0]**	R/W	0000_0000h

4.3.19 Data Port Data 1 Register (DP_DATA1)

Address:	050h	Size:	32 bits

The Data Port Data register holds the write data for a write access and the resultant read data for a read access.

Before reading the this register for the result of a read operation, the Data Port Ready bit should be checked. The Data Port Ready bit must indicate the data port is ready. Otherwise the read operation is still in progress.

This register required when accessing the RX TLI and TX TLI RAMs. These RAMs have a width of 37 bits.

BITS	DESCRIPTION	TYPE	DEFAULT
31:5	**RESERVED**	RO	-
4:0	**Data Port Data [36:32]**	R/W	00h

4.3.20 General Purpose IO Wake Enable and Polarity Register (GPIO_WAKE)

Address:	064h	Size:	32 bits

This register enables the GPIOs to function as wake events for the device when asserted. It also allows the polarity used for a wake event/interrupt to be configured.

Note: GPIOs must not cause a wake event to the device when not configured as a GPIO.

BITS	DESCRIPTION	TYPE	DEFAULT
31:24	**RESERVED**	RO	-
23:16	**GPIO Polarity 0-7 (GPIOPOLn)** 0 = Wakeup/interrupt is triggered when GPIO is driven low 1 = Wakeup/interrupt is triggered when GPIO is driven high GPIOPOL0 - bit 16 GPIOPOL1 - bit 17 GPIOPOL2 - bit 18 GPIOPOL3 - bit 19 GPIOPOL4 - bit 20 GPIOPOL5 - bit 21 GPIOPOL6 - bit 22 GPIOPOL7 - bit 23	R/W	00h
15:8	**RESERVED**	RO	-
7:0	**GPIO Wake 0-7 (GPIOWKn)** 0 = The GPIO can not wake up the Ethernet Controller. 1 = The GPIO can trigger a wake up event. GPIOWK0 - bit 0 GPIOWK1 - bit 1 GPIOWK2 - bit 2 GPIOWK3 - bit 3 GPIOWK4 - bit 4 GPIOWK5 - bit 5 GPIOWK6 - bit 6 GPIOWK7 - bit 7	R/W	00h

4.3.21 Interrupt Endpoint Control Register (INT_EP_CTL)

Address:	068h	Size:	32 bits

This register determines which events cause status to be reported by the interrupt endpoint. See Section 3.6.1.3, "Endpoint 3 (Interrupt)" for more details.

BITS	DESCRIPTION	TYPE	DEFAULT
31	**Interrupt Endpoint Always On (INTEP_ON)** When this bit is set, an interrupt packet will always be sent at the interrupt endpoint interval. 0 = Only allow the transmission of an interrupt packet when an interrupt source is enabled and occurs. 1 = Always transmit an interrupt packet at the interrupt interval.	R/W	0b
30:20	**RESERVED**	RO	-
19	**MAC Reset Time Out (MACRTO_EN)** 0 = This event can not cause an interrupt packet to be issued. 1 = This event can cause an interrupt packet to be issued.	R/W	0b
18	**RX FIFO Has Frame Enable (RX_FIFO_EN)** 0 = This event can not cause an interrupt packet to be issued. 1 = This event can cause an interrupt packet to be issued.	R/W	0b
17	**TX Stopped Enable (TXSTOP_EN)** 0 = This event can not cause an interrupt packet to be issued. 1 = This event can cause an interrupt packet to be issued.	R/W	0b
16	**RX Stopped Enable (RXSTOP_EN)** 0 = This event can not cause an interrupt packet to be issued. 1 = This event can cause an interrupt packet to be issued.	R/W	0b
15	**PHY Interrupt Enable (PHY_EN)** 0 = This event can not cause an interrupt packet to be issued. 1 = This event can cause an interrupt packet to be issued.	R/W	0b
14	**Transmitter Error Enable (TXE_EN)** 0 = This event can not cause an interrupt packet to be issued. 1 = This event can cause an interrupt packet to be issued.	R/W	0b
13	**TX Data FIFO Underrun Interrupt Enable (TDFU_EN)** 0 = This event can not cause an interrupt packet to be issued. 1 = This event can cause an interrupt packet to be issued.	R/W	0b
12	**TX Data FIFO Overrun Interrupt Enable (TDFO_EN)** 0 = This event can not cause an interrupt packet to be issued. 1 = This event can cause an interrupt packet to be issued.	R/W	0b
11	**RX Dropped Frame Interrupt Enable (RXDF_EN)** 0 = This event can not cause an interrupt packet to be issued. 1 = This event can cause an interrupt packet to be issued.	R/W	0b
10:8	**RESERVED**	RO	-
7:0	**GPIOx Interrupt Enable (GPIOx_EN)** 0 = This event can not cause an interrupt packet to be issued. 1 = This event can cause an interrupt packet to be issued.	R/W	0b

4.3.22 Bulk In Delay Register (BULK_IN_DLY)

Address: 06Ch Size: 32 bits

BITS	DESCRIPTION	TYPE	DEFAULT
31:16	**RESERVED**	RO	-
15:0	**Bulk In Delay** Before sending a short packet the UTX waits the delay specified by this register. This register has units of 16.667 ns and a default interval of 34.133 us.	R/W	800h

4.4 MAC Control and Status Registers

Table 4.5 lists the registers contained in this section.

Table 4.5 MAC Control and Status Register (MCSR) Map

ADDRESS	SYMBOL	REGISTER NAME
100h	MAC_CR	MAC Control Register
104h	ADDRH	MAC Address High Register
108h	ADDRL	MAC Address Low Register
10Ch	HASHH	Multicast Hash Table High Register
110h	HASHL	Multicast Hash Table Low Register
114h	MII_ACCESS	MII Access Register
118h	MII_DATA	MII Data Register
11Ch	FLOW	Flow Control Register
120h	VLAN1	VLAN1 Tag Register
124h	VLAN2	VLAN2 Tag Register
128h	WUFF	Wakeup Frame Filter Register
12Ch	WUCSR	Wakeup Control and Status Register
130h	COE_CR	Checksum Offload Engine Control Register
134h - 1FCh	RESERVED	Reserved for future use

4.4.1 MAC Control Register (MAC_CR)

Address:	100h	Size:	32 bits

This register establishes the RX and TX operating modes and includes controls for address filtering and packet filtering.

.

BITS	DESCRIPTION	TYPE	DEFAULT
31	**Receive All Mode (RXALL)** When set, all incoming packets will be received and passed on to the address filtering function for processing of the selected filtering mode on the received frame. Address filtering then occurs and is reported in Receive Status. When reset, only frames that pass Destination Address filtering will be sent to the Application.	R/W	0b
30-24	**RESERVED**	RO	-
23	**Disable Receive Own (RCVOWN)** When set, the MAC disables the reception of frames when TXEN is asserted. The MAC blocks the transmitted frame on the receive path. When reset, the MAC receives all packets the PHY gives, including those transmitted by the MAC.This bit should be reset when the Full Duplex Mode bit is set.	R/W	0b
22	**RESERVED**	RO	-
21	**Loopback Operation Mode (LOOPBK)** Selects the loop back operation modes for the MAC. This is only for full duplex mode 0 = Normal. No feedback 1 = Internal through MII In internal loopback mode, the TX frame is received by the Internal MII interface, and sent back to the MAC without being sent to the PHY. **Note:** When enabling or disabling the loopback mode, it can take up to 10µs for the mode change to occur. The transmitter and receiver must be stopped and disabled when modifying the LOOPBK bit. The transmitter or receiver should not be enabled within10µs of modifying the LOOPBK bit.	R/W	0b
20	**Full Duplex Mode (FDPX)** When set, the MAC operates in Full-Duplex mode, in which it can transmit and receive simultaneously.	R/W	0b
19	**Pass All Multicast (MCPAS)** When set, indicates that all incoming frames with a Multicast destination address (first bit in the destination address field is 1) are received. Incoming frames with physical address (Individual Address/Unicast) destinations are filtered and received only if the address matches the MAC Address.	R/W	0b
18	**Promiscuous Mode (PRMS)** When set, indicates that any incoming frame is received regardless of its destination address.	R/W	1b
17	**Inverse filtering (INVFILT)** When set, the address check Function operates in Inverse filtering mode. This is valid only during Perfect filtering mode.	R/W	0b

BITS	DESCRIPTION	TYPE	DEFAULT
16	**Pass Bad Frames (PASSBAD)** When set, all incoming frames that passed address filtering are received, including runt frames, collided frames or truncated frames caused by buffer underrun.	R/W	0b
15	**Hash Only Filtering mode (HO)** When set, the address check Function operates in the imperfect address filtering mode both for physical and multicast addresses.	R/W	0b
14	**RESERVED**	RO	-
13	**Hash/Perfect Filtering Mode (HPFILT)** When reset (0), the device will implement a perfect address filter on incoming frames, according the address specified in the MAC address register. When set (1), the address check function does imperfect address filtering of multicast incoming frames according to the hash table specified in the multicast hash table register. If the Hash Only Filtering mode (HO) bit is set (1), then the physical (IA) are imperfect filtered too. If the Hash Only Filtering mode (HO) bit is reset (0), then the IA addresses are perfect address filtered according to the MAC Address register	R/W	0b
12	**Late Collision Control (LCOLL)** When set, enables retransmission of the collided frame even after the collision period (late collision). When reset, the MAC disables frame transmission on a late collision. In any case, the Late Collision status is appropriately updated in the Transmit Packet status.	R/W	0b
11	**Disable Broadcast Frames (BCAST)** When set, disables the reception of broadcast frames. When reset, forwards all broadcast frames to the application. **Note:** When wakeup frame detection is enabled via the Wakeup Frame Enable (WUEN) bit of the Wakeup Control and Status Register (WUCSR), a broadcast wakeup frame will wake up the device despite the state of this bit.	R/W	0b
10	**Disable Retry (DISRTY)** When set, the MAC attempts only one transmission. When a collision is seen on the bus, the MAC ignores the current frame and goes to the next frame and a retry error is reported in the Transmit status. When reset, the MAC attempts 16 transmissions before signaling a retry error.	R/W	0b
9	**RESERVED**	RO	-
8	**Automatic Pad Stripping (PADSTR)** When set, the MAC strips the pad field on all incoming frames, if the length field is less than 46 bytes. The FCS field is also stripped, since it is computed at the transmitting station based on the data and pad field characters, and is invalid for a received frame that has had the pad characters stripped. Receive frames with a 46-byte or greater length field are passed to the Application unmodified (FCS is not stripped). When reset, the MAC passes all incoming frames to system memory unmodified.	R/W	0b

<table>
<tr><th>BITS</th><th>DESCRIPTION</th><th>TYPE</th><th>DEFAULT</th></tr>
<tr><td>7:6</td><td><b>BackOff Limit (BOLMT)</b>
The BOLMT bits allow the user to set its back-off limit in a relaxed or aggressive mode. According to IEEE 802.3, the MAC has to wait for a random number [r] of slot-times (Note 4.6) after it detects a collision, where:
(eq.1)$0 < r < {}_{2}K$
The exponent K is dependent on how many times the current frame to be transmitted has been retried, as follows:
(eq.2)K = min <i>(n,</i> 10) where <i>n</i> is the current number of retries.
If a frame has been retried three times, then K = 3 and r= 8 slot-times maximum. If it has been retried 12 times, then K = 10, and r = 1024 slot-times maximum.
An LFSR (linear feedback shift register) 20-bit counter emulates a 20bit random number generator, from which r is obtained. Once a collision is detected, the number of the current retry of the current frame is used to obtain K (eq.2). This value of K translates into the number of bits to use from the LFSR counter. If the value of K is 3, the MAC takes the value in the first three bits of the LFSR counter and uses it to count down to zero on every slot-time. This effectively causes the MAC to wait eight slot-times. To give the user more flexibility, the BOLMT value forces the number of bits to be used from the LFSR counter to a predetermined value as in the table below.
<table>
<tr><th>BOLMT Value</th><th># Bits Used from LFSR Counter</th></tr>
<tr><td>2’b00</td><td>10</td></tr>
<tr><td>2’b01</td><td>8</td></tr>
<tr><td>2’b10</td><td>4</td></tr>
<tr><td>2’b11</td><td>1</td></tr>
</table>
Thus, if the value of K = 10, the MAC will look at the BOLMT if it is 00, then use the lower ten bits of the LFSR counter for the wait countdown. If the BOLMT is 10, then it will only use the value in the first four bits for the wait countdown, etc.
<b>Note 4.6</b> Slot-time = 512 bit times. (See IEEE 802.3 Spec., Secs. 4.2.3.25 and 4.4.2.1)</td><td>R/W</td><td>00b</td></tr>
<tr><td>5</td><td><b>Deferral Check (DFCHK)</b>
When set, enables the deferral check in the MAC. The MAC will abort the transmission attempt if it has deferred for more than 24,288 bit times. Deferral starts when the transmitter is ready to transmit, but is prevented from doing so because the CRS is active. Defer time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of back-off, the deferral timer resets to 0 and restarts. When reset, the deferral check is disabled in the MAC and the MAC defers indefinitely.</td><td>R/W</td><td>0b</td></tr>
<tr><td>4</td><td><b>RESERVED</b></td><td>RO</td><td>-</td></tr>
<tr><td>3</td><td><b>Transmitter enable (TXEN)</b>
When set, the MAC’s transmitter is enabled and it will transmit frames from the buffer onto the cable. When reset, the MAC’s transmitter is disabled and will not transmit any frames.</td><td>R/W</td><td>0b</td></tr>
<tr><td>2</td><td><b>Receiver Enable (RXEN)</b>
When set (1), the MAC’s receiver is enabled and will receive frames from the internal PHY. When reset, the MAC’s receiver is disabled and will not receive any frames from the internal PHY.</td><td>R/W</td><td>0b</td></tr>
<tr><td>1:0</td><td><b>RESERVED</b></td><td>RO</td><td>-</td></tr>
</table>

4.4.2 MAC Address High Register (ADDRH)

Address: 104h Size: 32 bits

This register contains the upper 16 bits of the physical address of the MAC, where ADDRH[15:8] is the 6th octet of the RX frame.

BITS	DESCRIPTION	TYPE	DEFAULT
31:16	**RESERVED**	RO	-
15:0	**Physical Address [47:32]** This field contains the upper 16 bits (47:32) of the physical address of the device.	R/W	FFFFh

4.4.3 MAC Address Low Register (ADDRL)

Address:	108h	Size:	32 bits

This register contains the lower 32 bits of the physical address of the MAC, where ADDRL[7:0] is the first octet of the Ethernet frame.

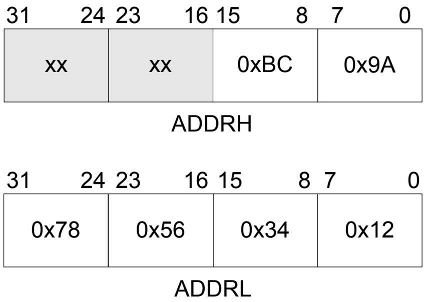
BITS	DESCRIPTION	TYPE	DEFAULT
31:0	**Physical Address [31:0]** This field contains the lower 32 bits (32:0) of the Physical Address of this MAC device.	R/W	32‘hF

Table 4.6 illustrates the byte ordering of the ADDRL and ADDRH registers with respect to the reception of the Ethernet physical address.

Table 4.6 ADDRL, ADDRH Byte Ordering

ADDRL, ADDRH	ORDER OF RECEPTION ON ETHERNET
ADDRL[7:0]	1st
ADDRL[15:8]	2nd
ADDRL[23:16]	3rd
ADDRL[31:24]	4th
ADDRH[7:0]	5th
ADDRH[15:8]	6th

As an example, if the desired Ethernet physical address is 12-34-56-78-9A-BC, the ADDRL and ADDRH registers would be programmed as shown in Figure 4.1.



31 24	23 16	15 8	7 0
xx	xx	0xBC	0x9A

ADDRH

31 24	23 16	15 8	7 0
0x78	0x56	0x34	0x12

ADDRL

Figure 4.1 Example ADDRL, ADDRH Address Ordering

4.4.4 Multicast Hash Table High Register (HASHH)

Address:	10Ch	Size:	32 bits

The 64-bit Multicast table is used for group address filtering. For hash filtering, the contents of the destination address in the incoming frame is used to index the contents of the Hash table. The most significant bit determines the register to be used (Hi/Low), while the other five bits determine the bit within the register. A value of 00000 selects Bit 0 of the Multicast Hash Table Lo register and a value of 11111 selects the Bit 31 of the Multicast Hash Table Hi register.

If the corresponding bit is 1, then the multicast frame is accepted. Otherwise, it is rejected. If the “Pass All Multicast” (MCPAS) bit is set (1), then all multicast frames are accepted regardless of the multicast hash values.

The Multicast Hash Table Hi register contains the higher 32 bits of the hash table and the Multicast Hash Table Low register contains the lower 32 bits of the hash table.

BITS	DESCRIPTION	TYPE	DEFAULT
31:0	**Upper 32 bits of the 64-bit Hash Table**	R/W	32‘h0

4.4.5 Multicast Hash Table Low Register (HASHL)

Address: 110h Size: 32 bits

This register defines the lower 32-bits of the Multicast Hash Table. Please refer to Section 4.4.4, "Multicast Hash Table High Register (HASHH)," on page 186 for further details.

BITS	DESCRIPTION	TYPE	DEFAULT
31:0	**Lower 32 bits of the 64-bit Hash Table**	R/W	32‘h0

4.4.6 MII Access Register (MII_ACCESS)

Address:	114h	Size:	32 bits

This register is used to control the management cycles to the internal PHY.

BITS	DESCRIPTION	TYPE	DEFAULT
31:16	**RESERVED**	RO	-
15:11	**PHY Address** For every access to this register, this field must be set to 00001b.	R/W	00000b
10:6	**MII Register Index (MIIRINDA)** These bits select the desired MII register in the PHY.	R/W	00000b
5:2	**RESERVED**	RO	-
1	**MII Write (MIIWnR)** Setting this bit tells the PHY that this will be a write operation using the MII data register. If this bit is not set, this will be a read operation, packing the data in the MII data register.	R/W	0b
0	**MII Busy (MIIBZY)** This bit must be polled to determine when the MII register access is complete. This bit must read a logical 0 before writing to this register or to the MII data register. The LAN driver software must set (1) this bit in order for the Host to read or write any of the MII PHY registers. During a MII register access, this bit will be set, signifying a read or write access is in progress. The MII data register must be kept valid until the MAC clears this bit during a PHY write operation. The MII data register is invalid until the MAC has cleared this bit during a PHY read operation.	R/W/SC	0b

4.4.7 MII Data Register (MII_DATA)

Address:	118h	Size:	32 bits

This register contains either the data to be written to the PHY register specified in the MII Access Register, or the read data from the PHY register whose index is specified in the MII Access Register. Refer to Section 4.4.6, "MII Access Register (MII_ACCESS)," on page 188 for further details.

Note: The MIIBZY bit in the MII_ACCESS register must be cleared when writing to this register.

BITS	DESCRIPTION		
31:16	**RESERVED**	RO	-
15:0	**MII Data** This contains the 16-bit value read from the PHY read operation or the 16-bit data value to be written to the PHY before an MII write operation.	R/W	0000h

4.4.8 Flow Control Register (FLOW)

Address: 11Ch Size: 32 bits

This register is used to control the generation and reception of the Control frames by the MAC's flow control block. A write to this register with busy bit set to 1 will trigger the Flow control block to generate a Control frame. Before writing to this register, the application has to make sure that the busy bit is not set.

BITS	DESCRIPTION	TYPE	DEFAULT
31:16	**Pause Time (FCPT)** This field indicates the value to be used in the PAUSE TIME field in the control frame.	R/W	0000h
15:3	**RESERVED**	RO	-
2	**Pass Control Frames (FCPASS)** When set, the MAC sets the packet filter bit in the receive packet status to indicate to the application that a valid pause frame has been received. The application must accept or discard a received frame based on the packet filter control bit. The MAC receives, decodes and performs the pause function when a valid pause frame is received in full-duplex mode and when flow control is enabled (FCEN bit set). When reset, the MAC resets the packet filter bit in the receive packet status. The MAC always passes the data of all frames it receives (including flow control frames) to the application. Frames that do not pass address filtering, as well as frames with errors, are passed to the application. The application must discard or retain the received frame's data based on the received frame's STATUS field. Filtering modes (promiscuous mode, for example) take precedence over the FCPASS bit.	R/W	0b
1	**Flow Control Enable (FCEN)** When set, enables the MAC flow control function. The MAC decodes all incoming frames for control frames; if it receives a valid control frame (PAUSE command), it disables the transmitter for a specified time (Decoded pause time x slot time). When reset, the MAC flow control function is disabled; the MAC does not decode frames for control frames. **Note:** Flow Control is applicable when the MAC is set in full duplex mode. In half-duplex mode, this bit enables the back pressure function to control the flow of received frames to the MAC.	R/W	0b
0	**Flow Control Busy (FCBSY)** This bit is set high whenever a pause frame or back pressure is being transmitted. This bit should read logical 0 before writing to the Flow Control (FLOW) register. During a transfer of Control Frame, this bit continues to be set, signifying that a frame transmission is in progress. After the PAUSE control frame's transmission is complete, the MAC resets to 0. **Note:** When writing this register the FCBSY bit must always be zero. **Note:** Applications must always write a zero to this bit.	R/W	0b

4.4.9 VLAN1 Tag Register (VLAN1)

Address:	120h	Size:	32 bits

This register contains the VLAN tag field to identify VLAN1 frames. For VLAN frames, the legal frame length is increased from 1518 bytes to 1522 bytes.

The RXCOE also uses this register to determine the protocol value to use to indicate the existence of a VLAN tag. When using the RXCOE, this value may only be changed if the Ethernet Controller RX path is disabled. If it is desired to change this value during run time, it is safe to do so only after the MAC is disabled and the TLI is empty.

BITS	DESCRIPTION	TYPE	DEFAULT
31:16	**RESERVED**	RO	-
15:0	**VLAN1 Tag Identifier (VTI1)** This contains the VLAN Tag field to identify the VLAN1 frames. This field is compared with the 13th and 14th bytes of the incoming frames for VLAN1 frame detection.	R/W	FFFFh

4.4.10 VLAN2 Tag Register (VLAN2)

Address: 124h Size: 32 bits

This register contains the VLAN tag field to identify VLAN2 frames. For VLAN frames the legal frame length is increased from 1518 bytes to 1522 bytes.

BITS	DESCRIPTION	TYPE	DEFAULT
31:16	**RESERVED**	RO	-
15:0	**VLAN2 Tag Identifier (VTI2)** This contains the VLAN Tag field to identify the VLAN2 frames. This field is compared with the 13th and 14th bytes of the incoming frames for VLAN2 frame detection.	R/W	FFFFh

4.4.11 Wakeup Frame Filter (WUFF)

Address:	128h	Size:	32 bits

This register is used to configure the Wakeup Frame Filter.

BITS	DESCRIPTION	TYPE	DEFAULT
31:0	**Wakeup Frame Filter (WFF)** The Wakeup Frame Filter is configured through this register using an indexing mechanism. Following a reset, the MAC loads the first value written to this location to the first DWORD in the Wakeup Frame Filter (Filter 0 Byte Mask 0). The second value written to this location is loaded to the second DWORD in the Wakeup Frame Filter (Filter 0 Byte Mask 1) and so on. Once all forty DWORDs have been written, the internal pointer will once again point to the first entry and the filter entries can be modified in the same manner. Similarly, forty DWORDS can be read sequentially to obtain the values stored in the WFF. **Note:** This register should be read and written using forty consecutive DWORD operations. Failure to read or write the entire contents of the WFF may cause the internal read/write pointers to be left in a position other than pointing to the first entry. A mechanism for resetting the internal pointers to the beginning of the WFF is available via the WFF Pointer Reset (WFF_PTR_RST) bit of the Wakeup Control and Status Register (WUCSR). This mechanism enables the application program to re-synchronize with the internal WFF pointers if it has not previously read/written the complete contents of the WFF.	R/W	00000000h

4.4.12 Wakeup Control and Status Register (WUCSR)

Address: 12Ch Size: 32 bits

This register contains data pertaining to the MAC’s remote wakeup status and capabilities.

BITS	DESCRIPTION	TYPE	DEFAULT
31	**WFF Pointer Reset (WFF_PTR_RST)** This self-clearing bit resets the Wakeup Frame Filter (WFF) internal read and write pointers to the beginning of the WFF.	SC	0b
30:10	**RESERVED**	RO	-
9	**Global Unicast Enable (GUE)** When set, the MAC wakes up from power-saving mode on receipt of a global unicast frame. A global unicast frame has the MAC Address [0] bit set to 0.	R/W	0b
8:7	**RESERVED**	RO	-
6	**Remote Wakeup Frame Received (WUFR)** The MAC sets this bit upon receiving a valid remote Wakeup Frame.	R/WC	0b
5	**Magic Packet Received (MPR)** The MAC sets this bit upon receiving a valid Magic Packet.	R/WC	0b
4-3	**RESERVED**	RO	-
2	**Wakeup Frame Enable (WUEN)** When set, remote wakeup mode is enabled and the MAC is capable of detecting Wakeup Frames as programmed in the Wakeup Frame Filter.	R/W	0b
1	**Magic Packet Enable (MPEN)** When set, Magic Packet wakeup mode is enabled.	R/W	0b
0	**RESERVED**	RO	-

4.4.13 Checksum Offload Engine Control Register (COE_CR)

Address:	130h	Size:	32 bits

This register controls the RX and TX checksum offload engines.

BITS	DESCRIPTION	TYPE	DEFAULT
31:17	**RESERVED**	RO	-
16	**TX Checksum Offload Engine Enable (TX_COE_EN)** TX_COE_EN may only be changed if the TX path is disabled. If it is desired to change this value during run time, it is safe to do so only after the MAC is disabled and the TLI is empty. 0 = The TXCOE is bypassed 1 = The TXCOE is enabled	R/W	0b
15:2	**RESERVED**	RO	-
1	**RX Checksum Offload Engine Mode (RX_COE_MODE)** This register indicates whether the COE will check for VLAN tags or a SNAP header prior to beginning its checksum calculation. In its default mode, the calculation will always begin 14 bytes into the frame. RX_COE_MODE may only be changed if the RX path is disabled. If it is desired to change this value during run time, it is safe to do so only after the MAC is disabled and the TLI is empty. 0 = Begin checksum calculation after first 14 bytes of Ethernet Frame 1 = Begin checksum calculation at start of L3 packet by adjusting for VLAN tags and/or SNAP header.	R/W	0b
0	**RX Checksum Offload Engine Enable (RX_COE_EN)** RX_COE_EN may only be changed if the Ethernet Controller RX path is disabled. If it is desired to change this value during run time, it is safe to do so only after the MAC is disabled and the TLI is empty. 0 = The RXCOE is bypassed 1 = The RXCOE is enabled	R/W	0b

4.5 PHY Registers

The PHY registers are not memory mapped. These registers are accessed indirectly through the MAC via the MII_ACCESS and MII_DATA registers. An index is used to access individual PHY registers. PHY Register Indexes are shown in Table 4.7, "PHY Control and Status Register" below.

Note: The NASR (**Not Affected by Software Reset)** designation is only applicable when bit 15 of the PHY Basic Control Register (Reset) is set.

Table 4.7 PHY Control and Status Register

INDEX (IN DECIMAL)	REGISTER NAME
0	Basic Control Register
1	Basic Status Register
2	PHY Identifier 1
3	PHY Identifier 2
4	Auto-Negotiation Advertisement Register
5	Auto-Negotiation Link Partner Ability Register
6	Auto-Negotiation Expansion Register
17	Mode Control/Status Register
18	Special Modes
27	Control / Status Indication Register
29	Interrupt Source Register
30	Interrupt Mask Register
31	PHY Special Control/Status Register

4.5.1 Basic Control Register

Index (In Decimal): 0 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15	**PHY Soft Reset** 1 = PHY software reset. Bit is self-clearing. When setting this bit do not set other bits in this register. **Note:** The PHY will be in the normal mode after a PHY software reset.	R/W/SC	0b
14	**Loopback** 0 = normal operation 1 = loopback mode	R/W	0b
13	**Speed Select** 0 = 10Mbps 1 = 100Mbps **Note:** Ignored if Auto Negotiation is enabled (0.12 = 1).	R/W	1b
12	**Auto-Negotiation Enable** 0 = disable auto-negotiate process 1 = enable auto-negotiate process (overrides 0.13 and 0.8)	R/W	1b
11	**Power Down** 0 = normal operation 1 = General power down mode **Note:** The Auto-Negotiation Enable must be cleared before setting the Power Down.	R/W	0b
10	**RESERVED**	RO	-
9	**Restart Auto-Negotiate** 0 = normal operation 1 = restart auto-negotiate process **Note:** Bit is self-clearing.	R/W/SC	0b
8	**Duplex Mode** 0 = half duplex 1 = full duplex **Note:** Ignored if Auto Negotiation is enabled (0.12 = 1).	R/W	0b
7	**Collision Test** 0 = disable COL test 1 = enable COL test	R/W	0b
6:0	**RESERVED**	RO	-

4.5.2 Basic Status Register

Index (In Decimal): 1 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15	**100BASE-T4** 0 = no T4 ability 1 = T4 able	RO	0b
14	**100BASE-TX Full Duplex** 0 = no TX full duplex ability 1 = TX with full duplex	RO	1b
13	**100BASE-TX Half Duplex** 0 = no TX half duplex ability 1 = TX with half duplex	RO	1b
12	**10BASE-T Full Duplex** 0 = no 10Mbps with full duplex ability 1 = 10Mbps with full duplex	RO	1b
11	**10BASE-T Half Duplex** 0 = no 10Mbps with half duplex ability 1 = 10Mbps with half duplex	RO	1b
10:6	**RESERVED**	RO	-
5	**Auto-Negotiate Complete** 0 = auto-negotiate process not completed 1 = auto-negotiate process completed	RO	0b
4	**Remote Fault** 1 = remote fault condition detected 0 = no remote fault	RO/LH	0b
3	**Auto-Negotiate Ability** 0 = unable to perform auto-negotiation function 1 = able to perform auto-negotiation function	RO	1b
2	**Link Status** 0 = link is down 1 = link is up	RO/LL	0b
1	**Jabber Detect** 0 = no jabber condition detected 1 = jabber condition detected	RO/LH	0b
0	**Extended Capabilities** 0 = does not support extended capabilities registers 1 = supports extended capabilities registers	RO	1b

4.5.3 PHY Identifier 1 Register

Index (In Decimal):	2	Size:	16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:0	**PHY ID Number** Assigned to the 3rd through 18th bits of the Organizationally Unique Identifier (OUI), respectively.	R/W	0007h

4.5.4 PHY Identifier 2 Register

Index (In Decimal):	3	Size:	16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:10	**PHY ID Number b** Assigned to the 19th through 24th bits of the OUI.	R/W	C0C3
9:4	**Model Number** Six-bit manufacturer’s model number.	R/W	
3:0	**Revision Number** Four-bit manufacturer’s revision number.	R/W	

4.5.5 Auto Negotiation Advertisement Register

Index (In Decimal): 4 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:14	**RESERVED**	RO	-
13	**Remote Fault** 0 = no remote fault 1 = remote fault detected	R/W	0b
12	**RESERVED**	RO	-
11:10	**Pause Operation** 00 = No PAUSE 01 = Symmetric PAUSE 10 = Asymmetric PAUSE toward link partner 11 = Advertise support for both Symmetric PAUSE and Asymmetric PAUSE toward local device **Note:** When both Symmetric PAUSE and Asymmetric PAUSE are set, the device will only be configured to, at most, one of the two settings upon autonegotiation completion.	R/W	00b
9	**RESERVED**	RO	-
8	**100BASE-TX Full Duplex** 0 = no TX full duplex ability 1 = TX with full duplex	R/W	1b
7	**100BASE-TX** 0 = no TX ability 1 = TX able	R/W	1b
6	**10BASE-T Full Duplex** 0 = no 10Mbps with full duplex ability 1 = 10Mbps with full duplex	R/W	1b
5	**10BASE-T** 0 = no 10Mbps ability 1 = 10Mbps able	R/W	1b
4:0	**Selector Field** 00001 = IEEE 802.3	R/W	00001b

4.5.6 Auto Negotiation Link Partner Ability Register

Index (In Decimal): 5 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15	**Next Page** 0 = no next page ability 1 = next page capable **Note:** This device does not support next page ability.	RO	0b
14	**Acknowledge** 0 = link code word not yet received 1 = link code word received from partner	RO	0b
13	**Remote Fault** 0 = no remote fault 1 = remote fault detected	RO	0b
12	**RESERVED**	RO	-
11:10	**Pause Operation** 00 = No PAUSE supported by partner station 01 = Symmetric PAUSE supported by partner station 10 = Asymmetric PAUSE supported by partner station 11 = Both Symmetric PAUSE and Asymmetric PAUSE supported by partner station	RO	00b
9	**100BASE-T4** 0 = no T4 ability 1 = T4 able	RO	0b
8	**100BASE-TX Full Duplex** 0 = no TX full duplex ability 1 = TX with full duplex	RO	0b
7	**100BASE-TX** 0 = no TX ability 1 = TX able	RO	0b
6	**10BASE-T Full Duplex** 0 = no 10Mbps with full duplex ability 1 = 10Mbps with full duplex	RO	0b
5	**10BASE-T** 0 = no 10Mbps ability 1 = 10Mbps able	RO	0b
4:0	**Selector Field** 00001 = IEEE 802.3	RO	00001b

4.5.7 Auto Negotiation Expansion Register

Index (In Decimal): 6 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:5	**RESERVED**	RO	-
4	**Parallel Detection Fault** 0 = no fault detected by parallel detection logic 1 = fault detected by parallel detection logic	RO/LH	0b
3	**Link Partner Next Page Able** 0 = link partner does not have next page ability 1 = link partner has next page ability	RO	0b
2	**Next Page Able** 0 = local device does not have next page ability 1 = local device has next page ability	RO	0b
1	**Page Received** 0 = new page not yet received 1 = new page received	RO/LH	0b
0	**Link Partner Auto-Negotiation Able** 0 = link partner does not have auto-negotiation ability 1 = link partner has auto-negotiation ability	RO	0b

4.5.8 Mode Control/Status Register

Index (In Decimal): 17 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:14	**RESERVED**	RO	-
13	**EDPWRDOWN** Enable the Energy Detect Power-Down mode: 0 = Energy Detect Power-Down is disabled 1 = Energy Detect Power-Down is enabled	R/W	0b
12:2	**RESERVED**	RO	-
1	**ENERGYON** Indicates whether energy is detected. This bit goes to a “0” if no valid energy is detected within 256ms. Reset to “1” by hardware reset, unaffected by SW reset.	RO	1b
0	**RESERVED**	R/W	0b

4.5.9 Special Modes Register

Index (In Decimal): 18 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:8	**RESERVED**	RO	-
7:5	**MODE** PHY Mode of operation. Refer to Table 4.8 for more details.	R/W NASR	111b
4:0	**PHYADD** PHY Address. The PHY Address is used for the SMI address.	R/W NASR	00001b

Table 4.8 MODE Control

MODE	MODE DEFINITIONS	DEFAULT REGISTER BIT VALUES	
		REGISTER 0	REGISTER 4
		[13,12,8]	[8,7,6,5]
000b	10BASE-T Half Duplex. Auto-negotiation disabled.	000	N/A
001b	10BASE-T Full Duplex. Auto-negotiation disabled.	001	N/A
010b	100BASE-TX Half Duplex. Auto-negotiation disabled. CRS is active during Transmit & Receive.	100	N/A
011b	100BASE-TX Full Duplex. Auto-negotiation disabled. CRS is active during Receive.	101	N/A
100b	100ase-TX Half Duplex is advertised. Auto-negotiation enabled. CRS is active during Transmit & Receive.	110	0100
101b	Repeater mode. Auto-negotiation enabled. 100BASE-TX Half Duplex is advertised. CRS is active during Receive.	110	0100
110b	RESERVED - Do not set the device in this mode.	N/A	N/A
111b	All capable. Auto-negotiation enabled.	X1X	1111

4.5.10 Special Control/Status Indications Register

Index (In Decimal): 27 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15	**Override AUTOMDIX_EN Strap** 0 = AUTOMDIX_EN configuration strap enables or disables HP Auto MDIX 1 = Override AUTOMDIX_EN configuration strap. PHY Register 27.14 and 27.13 determine MDIX function	R/W	0b
14	**Auto-MDIX Enable** Only effective when 27.15=1, otherwise ignored. 0 = Disable Auto-MDIX. 27.13 determines normal or reversed connection. 1 = Enable Auto-MDIX. 27.13 must be set to 0.	R/W	0b
13	**Auto-MDIX State** Only effective when 27.15=1, otherwise ignored. When 27.14 = 0 (manually set MDIX state): 0 = no crossover (TPO = output, TPI = input) 1 = crossover (TPO = input, TPI = output) When 27.14 = 1 (automatic MDIX) this bit must be set to 0. Do not use the combination 27.15=1, 27.14=1, 27.13=1.	R/W	0b
12:11	**RESERVED**	RO	-
10	**VCOOFF_LP** Forces the Receive PLL 10M to lock on the reference clock at all times: 0 = Receive PLL 10M can lock on reference or line as needed (normal operation). 1 = Receive PLL 10M is locked on the reference clock. In this mode 10M data packets cannot be received.	R/W NASR	0b
9:5	**RESERVED**	RO	-
4	**XPOL** Polarity state of the 10BASE-T: 0 = Normal polarity 1 = Reversed polarity	RO	0b
3:0	**RESERVED**	RO	-

4.5.11 Interrupt Source Flag Register

Index (In Decimal): 29 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:8	**RESERVED**	RO	-
7	**INT7** 0 = not source of interrupt 1 = ENERGYON generated	RO/LH	0b
6	**INT6** 0 = not source of interrupt 1 = Auto-Negotiation complete	RO/LH	0b
5	**INT5** 0 = not source of interrupt 1 = Remote Fault Detected	RO/LH	0b
4	**INT4** 0 = not source of interrupt 1 = Link Down (link status negated	RO/LH	0b
3	**INT3** 0 = not source of interrupt 1 = Auto-Negotiation LP Acknowledge	RO/LH	0b
2	**INT2** 0 = not source of interrupt 1 = Parallel Detection Fault	RO/LH	0b
1	**INT1** 0 = not source of interrupt 1 = Auto-Negotiation Page Received	RO/LH	0b
0	**RESERVED**	RO	0b

4.5.12 Interrupt Mask Register

Index (In Decimal): 30 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:8	**RESERVED**	RO	-
7:0	**Mask Bits** 0 = interrupt source is masked 1 = interrupt source is enabled	R/W	00h

4.5.13 PHY Special Control/Status Register

Index (In Decimal):	31	Size:	16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:13	**RESERVED**	RO	-
12	**Autodone** Auto-negotiation done indication: 0 = Auto-negotiation is not done or disabled (or not active) 1 = Auto-negotiation is done	RO	0
11:5	**RESERVED - Write as 0000010b, ignore on read.**	R/W	0000010b
4:2	**Speed Indication** HCDSPEED value: 001 = 10Mbps half-duplex 101 = 10Mbps full-duplex 010 = 100BASE-TX half-duplex 110 = 100BASE-TX full-duplex	RO	000b
1:0	**RESERVED**	RO	-

4.6 Hub Configuration Registers (HCFG)

Table 4.9 Hub Configuration Registers Memory Map

EEPROM ADDRESS	SYMBOL	REGISTER NAME
20h	VIDL	Vendor ID LSB Register
21h	VIDM	Vendor ID MSB Register
22h	PIDL	Product ID LSB Register
23h	PIDM	Product ID MSB Register
24h	DIDL	Device ID LSB Register
25h	DIDM	Device ID MSB Register
26h	CFG1	Config Data Byte 1 Register
27h	CFG2	Config Data Byte 2 Register
28h	CFG3	Config Data Byte 3 Register
29h	NRD	Non-Removable Devices Register
2Ah	PDS	Port Disable (Self) Register
2Bh	PDB	Port Disable (Bus) Register
2Ch	MAXPS	Max Power (Self) Register
2Dh	MAXPB	Max Power (Bus) Register
2Eh	HCMCS	Hub Controller Max Current (Self) Register
2Fh	HCMCB	Hub Controller Max Current (Bus) Register
30h	PWRT	Power-on Time Register
31h	BOOSTUP	Boost_Up Register
32h	RESERVED	Reserved for future expansion
33h	BOOST32	Boost_3:2 Register
34h	RESERVED	Reserved for future expansion
35h	PRTSP	Port Swap Register
36h	PRTR12	Port Remap 12 Register
37h	PRTR3	Port Remap 3 Register
38h	RESERVED	Reserved for future expansion
39h	STCD	Status/Command Register

The following sections define the Hub Configuration registers. The “TYPE” listing for all these registers is “LFE” (Load From EEPROM).

The registers are initialized via EEPROM load. All registers are set to zero after nReset or POR. However, if no EEPROM is present, they will be automatically be updated with the default values listed. No other provisions exist for writing these registers.

Reserved fields must be written with zeros, unless otherwise indicated, to ensure future compatibility. The value of reserved bits is not guaranteed on a read.

4.6.1 Vendor ID LSB (VIDL) Register

EEPROM Address: 20h   Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Vendor ID LSB (VID_LSB)** Least Significant Byte of the Vendor ID. This is a 16-bit value that uniquely identifies the Vendor of the user device (assigned by USB-Interface Forum).	LFE	24h

4.6.2 Vendor ID MSB (VIDM) Register

EEPROM Address: 21h Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Vendor ID MSB (VID_MSB)** Most Significant Byte of the Vendor ID. This is a 16-bit value that uniquely identifies the Vendor of the user device (assigned by USB-Interface Forum).	LFE	04h

4.6.3 Product ID LSB (PIDL) Register

EEPROM Address: 22h Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Product ID LSB (PID_LSB)** Least Significant Byte of the Product ID. This is a 16-bit value that the Vendor can assign that uniquely identifies this particular product (assigned by the OEM).	LFE	12h

4.6.4 Product ID MSB (PIDM) Register

EEPROM Address: 23h Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Product ID MSB (PID_MSB)** Most Significant Byte of the Product ID. This is a 16-bit value that the Vendor can assign that uniquely identifies this particular product (assigned by the OEM).	LFE	95h

4.6.5 Device ID LSB (DIDL) Register

EEPROM Address: 24h Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Device ID LSB (DID_LSB)** Least Significant Byte of the Device ID. This is a 16-bit device release number in BCD format (assigned by the OEM).	LFE	00h

4.6.6 Device ID MSB (DIDM) Register

EEPROM Address:	25h	Size:	8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Device ID MSB (DID_MSB)** Most Significant Byte of the Device ID. This is a 16-bit device release number in BCD format (assigned by the OEM).	LFE	Note 4.7

Note 4.7 Default value is dependent on device revision.

4.6.7 Config Data Byte 1 (CFG1) Register

EEPROM Address:	26h	Size:	8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7	**Self or Bus Power (SELF_BUS_PWR)** Selects between Self or Bus-Powered operation. 0 = Bus-Powered 1 = Self-Powered The Hub is either Self-Powered (draws less than 2mA of upstream bus power) or Bus-Powered (limited to a 100mA maximum of upstream power prior to being configured by the Host controller). When configured as a Bus-Powered device, the SMSC Hub consumes less than 100mA of current prior to being configured. After configuration, the Bus-Powered SMSC Hub (along with all associated Hub circuitry, any embedded devices if part of a compound device, and 100mA per externally available downstream port) must consume no more than 500mA of upstream VBUS current. The current consumption is system dependent, and the OEM must ensure that the USB2.0 specifications are not violated. When configured as a Self-Powered device, <1mA of upstream VBUS current is consumed and all ports are available, with each port being capable of sourcing 500mA of current.	LFE	1b
6	**RESERVED**	-	-
5	**High Speed Disable (HS_DISABLE)** Disables the capability to attach as either a High/Full-Speed device, and forces attachment as Full-Speed only (no High-Speed support). 0 = High-/Full-Speed 1 = Full-Speed-Only (High-Speed disabled)	LFE	0b
4	**Multiple TT Enable (MTT_ENABLE)** Enables one transaction translator per port operation. Selects between a mode where only one transaction translator is available for all ports (Single-TT), or each port gets a dedicated transaction translator (Multi-TT) {Note: The Host may force Single-TT mode only}. 0 = Single TT for all ports. 1 = One TT per port (multiple TT's supported)	LFE	1b

BITS	DESCRIPTION	TYPE	DEFAULT
3	**EOP Disable (EOP_DISABLE)** Disables EOP generation of EOF1 when in Full-Speed mode. During FS operation only, this permits the Hub to send EOP if no downstream traffic is detected at EOF1. See Section 11.3.1 of the USB 2.0 Specification for additional details. **Note:** Generation of an EOP at the EOF1 point may prevent a Host controller (operating in FS mode) from placing the USB bus in suspend. 0 = An EOP is generated at the EOF1 point if no traffic is detected. 1 = EOP generation at EOF1 is disabled (note: this is normal USB operation). **Note:** This is a rarely used feature in the PC environment, existing drivers may not have been thoroughly debugged with this feature enabled. It is included because it is a permitted feature in Chapter 11 of the USB specification.	LFE	1b
2:1	**Over Current Sense (CURRENT_SNS)** Selects current sensing on a port-by-port basis, all ports ganged, or none (only for bus-powered hubs). The ability to support current sensing on a port or ganged basis is hardware implementation dependent. 00 = Ganged sensing (all ports together) 01 = Individual port-by-port 1x = Over current sensing not supported (must only be used with Bus-Powered configurations!)	LFE	1b
0	**Port Power Switching (PORT_PWR)** Enables power switching on all ports simultaneously (ganged), or port power is individually switched on and off on a port by port basis (individual). The ability to support power enabling on a port or ganged basis is hardware implementation dependent. 0 = Ganged switching (all ports together) 1 = Individual port by port switching	LFE	1b

4.6.8 Config Data Byte 2 (CFG2) Register

EEPROM Address: 27h Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:6	**RESERVED**	-	-
5:4	**Over Current Timer (OC_TIMER)** Over Current Timer delay 00 = 50ns 01 = 100ns (This is the recommended value) 10 = 200ns 11 = 400ns	LFE	01b
3	**Compound Device (COMPOUND)** Allows the OEM to indicate that the Hub is part of a compound (see the USB Specification for definition) device. The applicable port(s) must also be defined as having a “Non-Removable Device”. 0 = No 1 = Yes, Hub is part of a compound device	LFE	1b
2:0	**RESERVED**	-	-

4.6.9 Config Data Byte 3 (CFG3) Register

EEPROM Address: 28h   Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:4	**RESERVED**	-	-
3	**Port Re-Mapping Enable (PRTMAP_EN)** Selects the method used by the Hub to assign port numbers and disable ports. 0 = Standard Mode. The following registers are used to define which ports are enabled. The ports mapped as Port'n' on the Hub are reported as Port'n' to the Host, unless one of the ports is disabled, then the higher numbered ports are remapped in order to report contiguous port numbers to the Host. Register 300Ah: Port Disable for Self-Powered operation Register 300Bh: Port Disable for Bus-Powered operation 1 = Port Re-Map mode. The mode enables remapping via the following registers: Register 30FBh: Port Remap 12 Register 30FCh: Port Remap 3	LFE	0b
2:0	**RESERVED**	-	-

4.6.10 Non-Removable Devices (NRD) Register

EEPROM Address: 29h Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Non-Removable Device (NR_DEVICE)** Indicates which port(s) include non-removable devices. 0 = Port is removable 1 = Port is non-removable Informs the Host if one of the active ports has a permanent device that is not detachable from the Hub. **Note:** The device must provide its own descriptor data. Bit 7 = RESERVED Bit 6 = RESERVED Bit 5 = RESERVED Bit 4 = RESERVED Bit 3 = 1; Port 3 non-removable Bit 2 = 1; Port 2 non-removable Bit 1 = 1; Port 1 non-removable Bit 0 is RESERVED, always = 0b **Note:** Bit 1 must be set to 1 by firmware for proper identification of the Ethernet Controller as a non-removable device.	LFE	02h

4.6.11 Port Disable For Self-Powered Operation (PDS) Register

EEPROM Address:	2Ah	Size:	8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Port Disable Self-Powered (PORT_DIS_SP)** Disables 1 or more ports. 0 = Port is available 1 = Port is disabled During Self-Powered operation, this selects the ports which will be permanently disabled, and are not available to be enabled or enumerated by a Host controller. The ports can be disabled in any order, the internal logic will automatically report the correct number of enabled ports to the USB Host, and will reorder the active ports in order to ensure proper function. Bit 7 = RESERVED Bit 6 = RESERVED Bit 5 = RESERVED Bit 4 = RESERVED Bit 3 = 1; Port 3 disabled Bit 2 = 1; Port 2 disabled Bit 1 = 1; Port 1 disabled Bit 0 is RESERVED, always = 0b	LFE	30h

4.6.12 Port Disable For Bus-Powered Operation (PDB) Register

EEPROM Address:	2Bh	Size:	8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Port Disable Bus-Powered (PORT_DIS_BP)** Disables 1 or more ports. 0 = Port is available 1 = Port is disabled During Bus-Powered operation, this selects the ports which will be permanently disabled, and are not available to be enabled or enumerated by a Host controller. The ports can be disabled in any order, the internal logic will automatically report the correct number of enabled ports to the USB Host, and will reorder the active ports in order to ensure proper function. Bit 7 = RESERVED Bit 6 = RESERVED Bit 5 = RESERVED Bit 4 = RESERVED Bit 3 = 1; Port 3 disabled Bit 2 = 1; Port 2 disabled Bit 1 = 1; Port 1 disabled Bit 0 is RESERVED, always = 0b	LFE	30h

4.6.13 Max Power For Self-Powered Operation (MAXPS) Register

EEPROM Address:	2Ch	Size:	8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Max Power Self-Powered (MAX_PWR_SP)** Value in 2mA increments that the Hub consumes from an upstream port (VBUS) when operating as a self-powered hub. This value includes the Hub silicon along with the combined power consumption (from VBUS) of all associated circuitry on the board. This value also includes the power consumption of a permanently attached peripheral if the Hub is configured as a compound device, and the embedded peripheral reports 0mA in its descriptors. **Note:** The USB2.0 Specification does not permit this value to exceed 100mA.	LFE	01h

4.6.14 Max Power For Bus-Powered Operation (MAXPB) Register

EEPROM Address:	2Dh	Size:	8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Max Power Bus-Powered (MAX_PWR_BP)** Value in 2mA increments that the Hub consumes from an upstream port (VBUS) when operating as a bus-powered hub. This value includes the Hub silicon along with the combined power consumption (from VBUS) of all associated circuitry on the board. This value also includes the power consumption of a permanently attached peripheral if the Hub is configured as a compound device, and the embedded peripheral reports 0mA in its descriptors.	LFE	00h

4.6.15 Hub Controller Max Current For Self-Powered Operation (HCMCS) Register

EEPROM Address: 2Eh Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Hub Controller Max Current Self-Powered (HC_MAX_C_SP)** Value in 2mA increments that the Hub consumes from an upstream port (VBUS) when operating as a self-powered hub. This value includes the Hub silicon along with the combined power consumption (from VBUS) of all associated circuitry on the board. This value does NOT include the power consumption of a permanently attached peripheral if the Hub is configured as a compound device. **Note:** The USB2.0 Specification does not permit this value to exceed 100mA.	LFE	01h

4.6.16 Hub Controller Max Current For Bus-Powered Operation (HCMCB) Register

EEPROM Address:	2Fh	Size:	8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Hub Controller Max Current Bus-Powered (HC_MAX_C_BP)** Value in 2mA increments that the Hub consumes from an upstream port (VBUS) when operating as a bus-powered hub. This value includes the Hub silicon along with the combined power consumption (from VBUS) of all associated circuitry on the board. This value does NOT include the power consumption of a permanently attached peripheral if the Hub is configured as a compound device.	LFE	00h

4.6.17 Power-On Time (PWRT) Register

EEPROM Address: 30h Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Power On Time (POWER_ON_TIME)** The length of time that it takes (in 2mS intervals) from the time the Host initiated power-on sequence begins on a port until power is good on that port. System software uses this value to determine how long to wait before accessing a powered-on port.	LFE	32h

4.6.18 Boost_Up (BOOSTUP) Register

EEPROM Address: 31h   Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:2	**RESERVED**	-	-
1:0	**Upstream USB Electrical Signaling Drive Strength Boost Bit for Upstream Port A (BOOST_IOUT_A)** 00 = Normal electrical drive strength 01 = Elevated electrical drive strength (+4% boost) 10 = Elevated electrical drive strength (+8% boost) 11 = Elevated electrical drive strength (+12% boost)	LFE	00b

4.6.19 Boost_3:2 (BOOST32) Register

EEPROM Address: 33h Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:6	**RESERVED**	-	-
5:4	**Upstream USB Electrical Signaling Drive Strength Boost Bit for Downstream Port 3 (BOOST_IOUT_3)** 00 = Normal electrical drive strength 01 = Elevated electrical drive strength (+4% boost) 10 = Elevated electrical drive strength (+8% boost) 11 = Elevated electrical drive strength (+12% boost)	LFE	00b
3:2	**Upstream USB Electrical Signaling Drive Strength Boost Bit for Downstream Port 2 (BOOST_IOUT_2)** 00 = Normal electrical drive strength 01 = Elevated electrical drive strength (+4% boost) 10 = Elevated electrical drive strength (+8% boost) 11 = Elevated electrical drive strength (+12% boost)	LFE	00b
1:0	**RESERVED**	-	-

4.6.20 Port Swap (PRTSP) Register

EEPROM Address: 35h Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:0	**Port Swap (PRTSP)** Swaps the Upstream and Downstream USBDPn and USBDMn pins for ease of board routing to devices and connectors. 0 = USB D+ functionality is associated with the USBDPn pin and D- functionality is associated with the USBDMn pin. 1 = USB D+ functionality is associated with the USBDMn pin and D- functionality is associated with the USBDPn pin. Bit 7 = RESERVED Bit 6 = RESERVED Bit 5 = RESERVED Bit 4 = RESERVED Bit 3 = 1; Port 3 USBDP3/USBDM3 are swapped Bit 2 = 1; Port 2 USBDP2/USBDM2 are swapped Bit 1 = RESERVED Bit 0 = 1; Upstream Port USBDP0/USBDM0 are swapped	LFE	00h

4.6.21 Port Remap 12 (PRTR12) Register

EEPROM Address: 36h Size: 8 bits

<table>
<tr><th>BITS</th><th>DESCRIPTION</th><th>TYPE</th><th>DEFAULT</th></tr>
<tr><td>7:0</td><td><b>Port Remap Register for Ports 1 & 2 (PRTR12)</b>
When a hub is enumerated by a USB Host Controller, the hub is only permitted to report how many ports it has. The hub is not permitted to select a numerical range or assignment. The Host Controller will number the downstream ports of the hub starting with the number 1, up to the number of ports that the hub reported having.
The Host’s port number is referred to as “Logical Port Number” and the physical port on the hub is the “Physical Port Number”. When remapping mode is enabled (see Port Re-Mapping Enable (PRTMAP_EN) bit in Config Data Byte 3 (CFG3) Register) the hub’s downstream port numbers can be remapped to different logical port numbers (assigned by the Host).
<b>Note:</b> The OEM must ensure that Contiguous Logical Port Numbers are used, starting from #1 up to the maximum number of enabled ports. This ensures that the hub’s ports are numbered in accordance with the way a Host will communicate with the ports.
<table>
<tr><td>Bit [7:4] =</td><td>0000</td><td>Physical Port 2 is Disabled</td></tr>
<tr><td></td><td>0001</td><td>Physical Port 2 is mapped to Logical Port 1</td></tr>
<tr><td></td><td>0010</td><td>Physical Port 2 is mapped to Logical Port 2</td></tr>
<tr><td></td><td>0011</td><td>Physical Port 2 is mapped to Logical Port 3</td></tr>
<tr><td></td><td></td><td>All others RESERVED</td></tr>
<tr><td>Bit [3:0] =</td><td>0000</td><td>Physical Port 1 is Disabled</td></tr>
<tr><td></td><td>0001</td><td>Physical Port 1 is mapped to Logical Port 1</td></tr>
<tr><td></td><td>0010</td><td>Physical Port 1 is mapped to Logical Port 2</td></tr>
<tr><td></td><td>0011</td><td>Physical Port 1 is mapped to Logical Port 3</td></tr>
<tr><td></td><td></td><td>All others RESERVED</td></tr>
</table></td><td>LFE</td><td>21h</td></tr>
</table>

4.6.22 Port Remap 3 (PRTR3) Register

EEPROM Address:	37h	Size:	8 bits

<table>
<tr><th>BITS</th><th>DESCRIPTION</th><th>TYPE</th><th>DEFAULT</th></tr>
<tr><td>7:0</td><td><b>Port Remap Register for Port 3 (PRTR3)</b>
When a hub is enumerated by a USB Host Controller, the hub is only permitted to report how many ports it has. The hub is not permitted to select a numerical range or assignment. The Host Controller will number the downstream ports of the hub starting with the number 1, up to the number of ports that the hub reported having.
The Host's port number is referred to as "Logical Port Number" and the physical port on the hub is the "Physical Port Number". When remapping mode is enabled (see Port Re-Mapping Enable (PRTMAP_EN) bit in Config Data Byte 3 (CFG3) Register) the hub's downstream port numbers can be remapped to different logical port numbers (assigned by the Host).
<b>Note:</b> The OEM must ensure that Contiguous Logical Port Numbers are used, starting from #1 up to the maximum number of enabled ports, this ensures that the hub's ports are numbered in accordance with the way a Host will communicate with the ports.
<table>
<tr><td>Bit [7:4] =</td><td>-</td><td>RESERVED</td></tr>
<tr><td>Bit [3:0] =</td><td>0000</td><td>Physical Port 3 is Disabled</td></tr>
<tr><td></td><td>0001</td><td>Physical Port 3 is mapped to Logical Port 1</td></tr>
<tr><td></td><td>0010</td><td>Physical Port 3 is mapped to Logical Port 2</td></tr>
<tr><td></td><td>0011</td><td>Physical Port 3 is mapped to Logical Port 3</td></tr>
<tr><td></td><td></td><td>All others RESERVED</td></tr>
</table>
</td><td>LFE</td><td>03h</td></tr>
</table>

4.6.23 Status/Command (STCD) Register

EEPROM Address: 39h Size: 8 bits

BITS	DESCRIPTION	TYPE	DEFAULT
7:2	**RESERVED**	-	-
1	**Reset (RESET)** Resets the internal memory back to nRESET assertion default settings. 0 = Normal Run/Idle State 1 = Force a reset of the registers to their default state **Note:** During this reset, this bit is automatically cleared to its default value of 0.	LFE	0b
0	**USB Attach and Write Protect (USB_ATTACH)** 0 = Device is in configuration state 1 = Hub will signal a USB attach event to an upstream device, and the internal memory (address range 00h - FEh) is “write-protected” to prevent unintentional data corruption. **Note:** This bit is write once and is only cleared by assertion of the external nRESET or POR.	LFE	1b

Chapter 5 Operational Characteristics

5.1 Absolute Maximum Ratings*

Supply Voltage (VDD33IO, VDD33A) (Note 5.1)	0V to +3.6V
Positive voltage on signal pins, with respect to ground (Note 5.2)	+6V
Negative voltage on signal pins, with respect to ground (Note 5.3)	-0.5V
Positive voltage on XI, with respect to ground	+4.6V
Positive voltage on XO, with respect to ground	+2.5V
Ambient Operating Temperature in Still Air (T_A)	Note 5.4
Storage Temperature	-55°C to +150°C
Lead Temperature Range	Refer to JEDEC Spec. J-STD-020
HBM ESD Performance per JESD 22-A114-E	+/- 8kV
Contact Discharge ESD Performance per IEC61000-4-2 (Note 5.5)	+/- 8kV
Air-Gap Discharge ESD Performance per IEC61000-4-2 (Note 5.5)	+/- 15kV
Latch-up Performance per EIA/JESD 78	+/- 200mA

Note 5.1 When powering this device from laboratory or system power supplies, it is important that the absolute maximum ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes on their outputs when AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists, it is suggested that a clamp circuit be used.

Note 5.2 This rating does not apply to the following pins: XI, XO, EXRES, USBRBIAS.

Note 5.3 This rating does not apply to the following pins: EXRES, USBRBIAS.

Note 5.4 0°C to +70°C for commercial version, -40°C to +85°C for industrial version.

Note 5.5 Performed by independent 3rd party test facility.

*Stresses exceeding those listed in this section could cause permanent damage to the device. This is a stress rating only. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at any condition exceeding those indicated in Section 5.2, "Operating Conditions**", Section 5.4, "DC Specifications", or any other applicable section of this specification is not implied. Note, device signals are *NOT* 5 volt tolerant unless specified otherwise.

5.2 Operating Conditions**

Supply Voltage (VDD33A, VDD33BIAS, VDD33IO)	+3.3V +/- 300mV
Ambient Operating Temperature in Still Air (T_A)	Note 5.4

**Proper operation of LAN9512/LAN9512i is guaranteed only within the ranges specified in this section.

5.3 Power Consumption

This section details the power consumption of the device as measured during various modes of operation. Power dissipation is determined by temperature, supply voltage, and external source/sink requirements.

5.3.1 SUSPEND0

Table 5.1 SUSPEND0 Current Consumption and Power Dissipation (VDD33IO = VDD33A = 3.3V)

PARAMETER	MIN	TYPICAL	MAX	UNIT
Supply current (VDD33IO, VDD33A)		74		mA
Power Dissipation (Device Only)		245		mW
Power Dissipation (Device and Ethernet components)		379		mW

5.3.2 SUSPEND1

Table 5.2 SUSPEND1 Current Consumption and Power Dissipation (VDD33IO = VDD33A = 3.3V)

PARAMETER	MIN	TYPICAL	MAX	UNIT
Supply current (VDD33IO, VDD33A)		68		mA
Power Dissipation (Device Only)		224		mW
Power Dissipation (Device and Ethernet components)		229		mW

5.3.3 SUSPEND2

Table 5.3 SUSPEND2 Current Consumption and Power Dissipation (VDD33IO = VDD33A = 3.3V)

PARAMETER	MIN	TYPICAL	MAX	UNIT
Supply current (VDD33IO, VDD33A)		4.2		mA
Power Dissipation (Device Only)		14.0		mW
Power Dissipation (Device and Ethernet components)		14.1		mW

5.3.4 Operational Current Consumption & Power Dissipation

Table 5.4 Operational Current Consumption and Power Dissipation (VDD33IO = VDD33A = 3.3V)

PARAMETER	MIN	TYPICAL	MAX	UNIT
100BASE-TX Full Duplex (USB High-Speed)				
Supply current (VDD33IO, VDD33A)		231		mA
Power Dissipation (Device Only)		763		mW
10BASE-T Full Duplex (USB High-Speed)				
Supply current (VDD33IO, VDD33A)		188		mA
Power Dissipation (Device Only)		621		mW
10BASE-T Full Duplex (USB Full-Speed)				
Supply current (VDD33IO, VDD33A)		152		mA
Power Dissipation (Device Only)		502		mW

Note: All values measured with maximum simultaneous traffic on the Ethernet port and all USB ports.

Note: Magnetic power consumption:

- 100BASE-TX: ~42mA
- 10BASE-T: ~104mA

5.4 DC Specifications

Table 5.5 I/O Buffer Characteristics

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
IS Type Input Buffer						
Low Input Level	V_{ILI}	-0.3			V	
High Input Level	V_{IHI}			3.6	V	
Negative-Going Threshold	V_{ILT}	1.01	1.18	1.35	V	Schmitt trigger
Positive-Going Threshold	V_{IHT}	1.39	1.6	1.8	V	Schmitt trigger
SchmittTrigger Hysteresis (V_{IHT} - V_{ILT})	V_{HYS}	345	420	485	mV	
Input Leakage (V_{IN} = VSS or VDD33IO)	I_{IH}	-10		10	uA	Note 5.6
Input Capacitance	C_{IN}			2.5	pF	
IS_5V Type Input Buffer						
Low Input Level	V_{ILI}	-0.3			V	
High Input Level	V_{IHI}			5.5	V	
Negative-Going Threshold	V_{ILT}	1.01	1.18	1.35	V	Schmitt trigger
Positive-Going Threshold	V_{IHT}	1.39	1.6	1.8	V	Schmitt trigger
SchmittTrigger Hysteresis (V_{IHT} - V_{ILT})	V_{HYS}	345	420	485	mV	
Input Leakage (V_{IN} = VSS or VDD33IO)	I_{IH}	-10		10	uA	Note 5.6
Input Leakage (V_{IN} = 5.5V)	I_{IH}			120	uA	Note 5.6, Note 5.7
Input Capacitance	C_{IN}			3.5	pF	
O8 Type Buffers						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8mA
High Output Level	V_{OH}	VDD33IO - 0.4			V	I_{OH} = -8mA
OD8 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8mA
O12 Type Buffers						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 12mA
High Output Level	V_{OH}	VDD33IO - 0.4			V	I_{OH} = -12mA
OD12 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 12mA
ICLK Type Buffer (XI Input)						Note 5.8
Low Input Level	V_{ILI}	-0.3		0.5	V	
High Input Level	V_{IHI}	1.4		3.6	V	

Note 5.6 This specification applies to all inputs and tri-stated bi-directional pins. Internal pull-down and pull-up resistors add +/- 50uA per-pin (typical).

Note 5.7 This is the total 5.5V input leakage for the entire device. This value should be divided by the number of pins driven to 5.5V to calculate per-pin leakage. For example, if both 5V tolerant inputs are driven to 5.5V, the per-pin leakage is TBD/2.

Note 5.8 XI can optionally be driven from a 25MHz single-ended clock oscillator.

Table 5.6 100BASE-TX Transceiver Characteristics

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Peak Differential Output Voltage High	V_{PPH}	950	-	1050	mVpk	Note 5.9
Peak Differential Output Voltage Low	V_{PPL}	-950	-	-1050	mVpk	Note 5.9
Signal Amplitude Symmetry	V_{SS}	98	-	102	%	Note 5.9
Signal Rise and Fall Time	T_{RF}	3.0	-	5.0	nS	Note 5.9
Rise and Fall Symmetry	T_{RFS}	-	-	0.5	nS	Note 5.9
Duty Cycle Distortion	D_{CD}	35	50	65	%	Note 5.10
Overshoot and Undershoot	V_{OS}	-	-	5	%	
Jitter				1.4	nS	Note 5.11

Note 5.9 Measured at line side of transformer, line replaced by 100Ω (+/- 1%) resistor.

Note 5.10 Offset from 16nS pulse width at 50% of pulse peak.

Note 5.11 Measured differentially.

Table 5.7 10BASE-T Transceiver Characteristics

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Transmitter Peak Differential Output Voltage	V_{OUT}	2.2	2.5	2.8	V	Note 5.12
Receiver Differential Squelch Threshold	V_{DS}	300	420	585	mV	

Note 5.12 Min/max voltages guaranteed as measured with 100Ω resistive load.

5.5 AC Specifications

This section details the various AC timing specifications of the LAN9512/LAN9512i.

Note: The USBDP and USBDM pin timing adheres to the USB 2.0 specification. Refer to the Universal Serial Bus Revision 2.0 specification for detailed USB timing information.

5.5.1 Equivalent Test Load

Output timing specifications assume the 25pF equivalent test load illustrated in Figure 5.1 below.

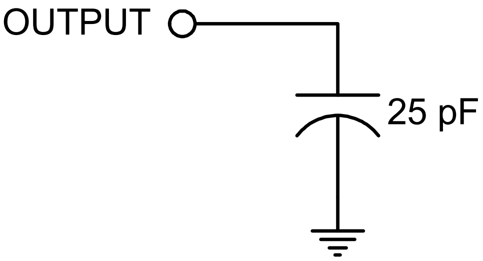


Figure 5.1 Output Equivalent Test Load

5.5.2 Reset Timing

The nRESET pin input assertion time must be a minimum of 1 μS. Assertion of nRESET is not a requirement. However, if used, it must be asserted for the minimum period specified.

5.5.3 EEPROM Timing

The following specifies the EEPROM timing requirements for LAN9512/LAN9512i:

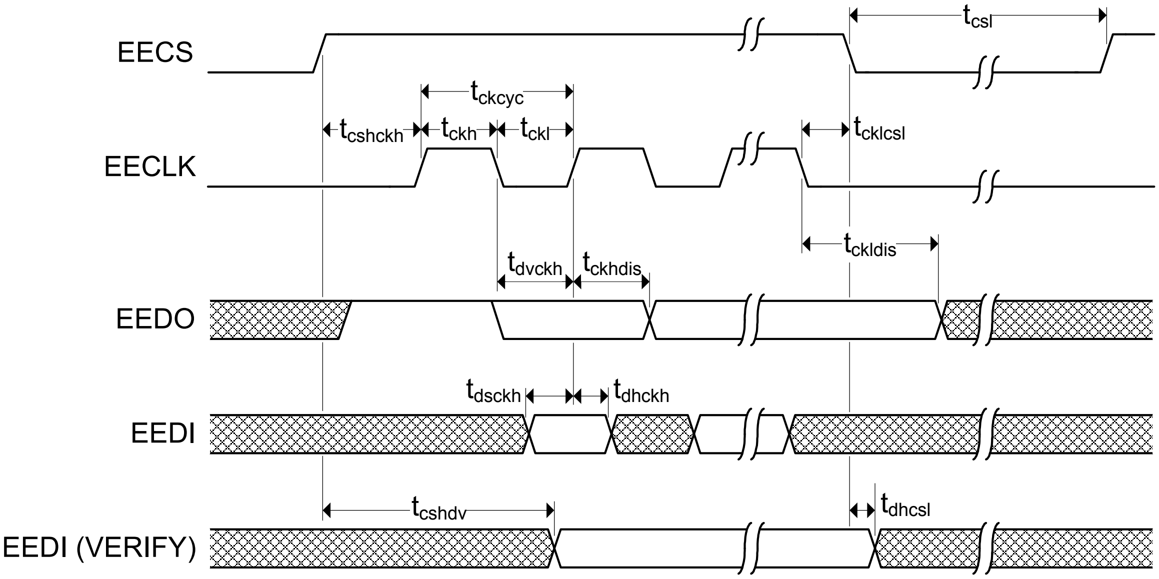


Figure 5.1 EEPROM Timing

Table 5.8 EEPROM Timing Values

SYMBOL	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{ckcyc}	EECLK Cycle time	1110		1130	ns
t_{ckh}	EECLK High time	550		570	ns
t_{ckl}	EECLK Low time	550		570	ns
t_{cshckh}	EECS high before rising edge of EECLK	1070			ns
t_{cklcsl}	EECLK falling edge to EECS low	30			ns
t_{dvckh}	EEDO valid before rising edge of EECLK	550			ns
t_{ckhdis}	EEDO disable after rising edge EECLK	550			ns
t_{dsckh}	EEDI setup to rising edge of EECLK	90			ns
t_{dhckh}	EEDI hold after rising edge of EECLK	0			ns
t_{ckldis}	EECLK low to data disable (OUTPUT)	580			ns
t_{cshdv}	EEDIO valid after EECS high (VERIFY)			600	ns
t_{dhcsl}	EEDIO hold after EECS low (VERIFY)	0			ns
t_{csl}	EECS low	1070			ns

5.5.4 JTAG Timing

This section specifies the JTAG timing of the device.

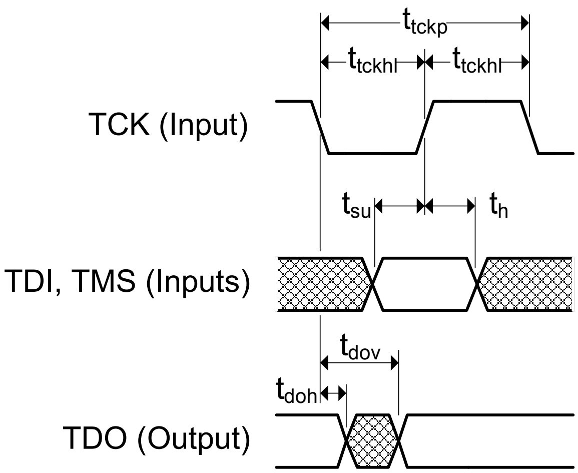


Figure 5.2 JTAG Timing

Table 5.9 JTAG Timing Values

SYMBOL	DESCRIPTION	MIN	MAX	UNITS	NOTES
t_{tckp}	TCK clock period	66.67		ns	
t_{tckhl}	TCK clock high/low time	t_{tckp}*0.4	t_{tckp}*0.6	ns	
t_{su}	TDI, TMS setup to TCK rising edge	10		ns	
t_{h}	TDI, TMS hold from TCK rising edge	10		ns	
t_{dov}	TDO output valid from TCK falling edge		16	ns	
t_{doh}	TDO output hold from TCK falling edge	0		ns	

5.6 Clock Circuit

LAN9512/LAN9512i can accept either a 25MHz crystal (preferred) or a 25MHz single-ended clock oscillator (+/- 50ppm) input. If the single-ended clock oscillator method is implemented, XO should be left unconnected and XI should be driven with a nominal 0-3.3V clock signal. The input clock duty cycle is 40% minimum, 50% typical and 60% maximum.

It is recommended that a crystal utilizing matching parallel load capacitors be used for the crystal input/output signals (XI/XO). See Table 5.10 for the recommended crystal specifications.

Table 5.10 LAN9512/LAN9512i Crystal Specifications

PARAMETER	SYMBOL	MIN	NOM	MAX	UNITS	NOTES
Crystal Cut	AT, typ					
Crystal Oscillation Mode	Fundamental Mode					
Crystal Calibration Mode	Parallel Resonant Mode					
Frequency	F_{fund}	-	25.000	-	MHz	
Frequency Tolerance @ 25°C	F_{tol}	-	-	+/-50	PPM	Note 5.13
Frequency Stability Over Temp	F_{temp}	-	-	+/-50	PPM	Note 5.13
Frequency Deviation Over Time	F_{age}	-	+/-3 to 5	-	PPM	Note 5.14
Total Allowable PPM Budget		-	-	+/-50	PPM	Note 5.15
Shunt Capacitance	C_O	-	7 typ	-	pF	
Load Capacitance	C_L	-	20 typ	-	pF	
Drive Level	P_W	300	-	-	uW	
Equivalent Series Resistance	R_1	-	-	50	Ohm	
Operating Temperature Range		Note 5.16	-	Note 5.17	°C	
LAN9512/LAN9512i XI Pin Capacitance		-	3 typ	-	pF	Note 5.18
LAN9512/LAN9512i XO Pin Capacitance		-	3 typ	-	pF	Note 5.18

Note 5.13 The maximum allowable values for Frequency Tolerance and Frequency Stability are application dependant. Since any particular application must meet the IEEE +/-50 PPM Total PPM Budget, the combination of these two values must be approximately +/-45 PPM (allowing for aging).

Note 5.14 Frequency Deviation Over Time is also referred to as Aging.

Note 5.15 The total deviation for the Transmitter Clock Frequency is specified by IEEE 802.3u as +/- 50 PPM.

Note 5.16 0°C for commercial version, -40°C for industrial version.

Note 5.17 +70°C for commercial version, +85°C for industrial version.

Note 5.18 This number includes the pad, the bond wire and the lead frame. PCB capacitance is not included in this value. The XO/XI pin and PCB capacitance values are required to accurately calculate the value of the two external load capacitors. These two external load capacitors determine the accuracy of the 25.000 MHz frequency.

Chapter 6 Package Outline

6.1 64-QFN Package

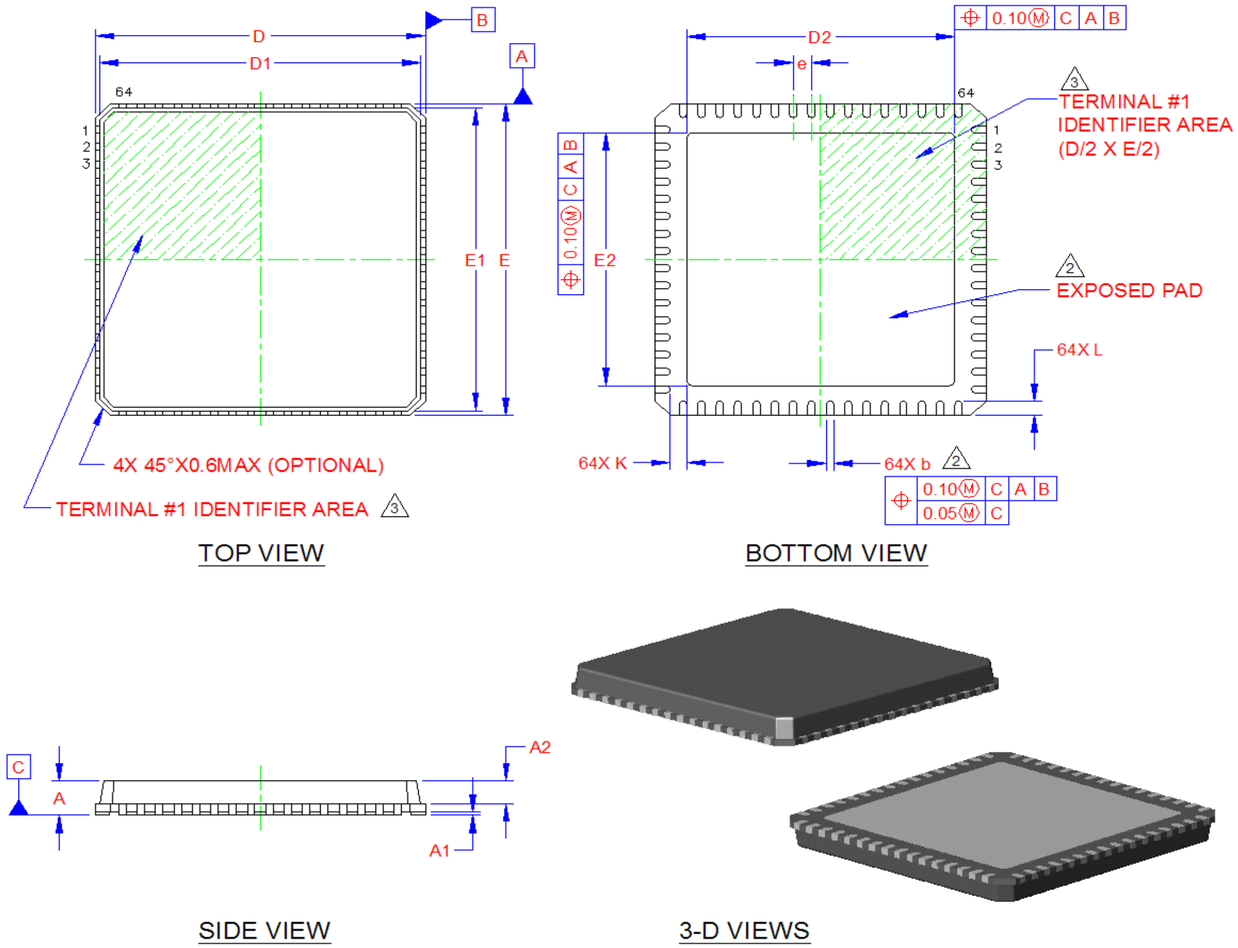


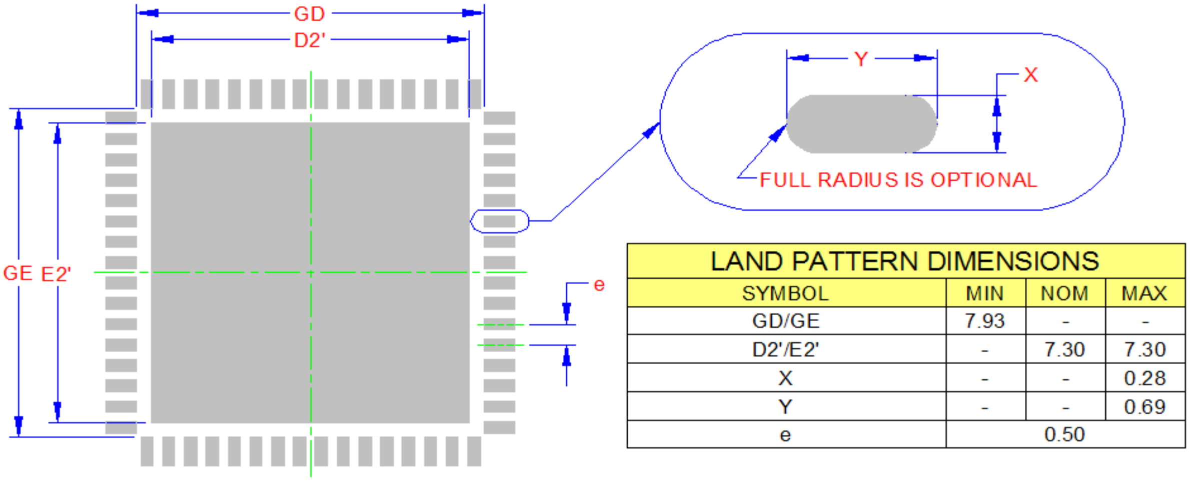
Figure 6.1 LAN9512/LAN9512i 64-QFN Package Definition

Table 6.1 LAN9512/LAN9512i 64-QFN Dimensions

	MIN	NOMINAL	MAX	REMARKS
A	0.80	0.85	1.00	Overall Package Height
A1	0.00	0.02	0.05	Standoff
A2	-	0.65	0.80	Mold Cap Thickness
D/E	8.90	9.00	9.10	X/Y Body Size
D1/E1	8.65	8.75	8.85	X/Y Mold Cap Size
D2/E2	7.20	7.30	7.40	X/Y Exposed Pad Size
L	0.30	0.40	0.50	Terminal Length
b	0.18	0.25	0.30	Terminal Width
e	0.50 BSC			Terminal Pitch
K	0.35	-	-	Pin to Center Pad Clearance

Notes:

1. All dimensions are in millimeters unless otherwise noted.
2. Dimension “b” applies to plated terminals and is measured between 0.15 and 0.30 mm from the terminal tip.
3. Details of terminal #1 identifier are optional, but must be located within the area indicated. The terminal #1 identifier may be either a mold or marked feature.



LAND PATTERN DIMENSIONS

SYMBOL	MIN	NOM	MAX
GD/GE	7.93	-	-
D2'/E2'	-	7.30	7.30
X	-	-	0.28
Y	-	-	0.69
e		0.50	

THE USER MAY MODIFY THE PCB LAND PATTERN DIMENSIONS
BASED ON THEIR EXPERIENCE AND/OR PROCESS CAPABILITY

RECOMMENDED PCB LAND PATTERN

Figure 6.2 LAN9512/LAN9512i Recommended PCB Land Pattern

Chapter 7 Databook Revision History

Table 7.1 Customer Revision History

REVISION LEVEL AND DATE	SECTION/FIGURE/ENTRY	CORRECTION
Rev. 1.2 (03-01-12)	Section 5.3, "Power Consumption," on page 236	Added suspend 0, suspend 1, and suspend 2 power consumption data.
Rev. 1.1 (09-19-11)	All	Fixed typos.
	Section 3.8.7.1, "TX Checksum Calculation," on page 112	Added note stating TX Checksum calculation should not be used for UDP packets under IPv6.
	Section 2.1, "Power Connections," on page 27	Added power connections section with diagram.
Rev. 1.1 (11-24-09)	All: Cover, Ordering Code, Operational Characteristics	Added industrial temperature range option: (-40°C to +85°C)
	Section 5.5.4, "JTAG Timing," on page 242	Added JTAG timing information
Rev. 1.0 (04-22-09)	Section 5.1, "Absolute Maximum Ratings*," on page 235	Added ESD information.
	Section 3.10, "EEPROM Controller (EPC)," on page 123	Updated supported EEPROM information.
	Section 5.3, "Power Consumption," on page 236	Added power consumption values.
	Section 5.4, "DC Specifications," on page 238	Added input capacitance and leakage values.
Rev. 1.0 (03-03-09)	All	Initial revision