LAN9254

2/3-Port EtherCAT® Slave Controller with Integrated Ethernet PHYs and Demultiplexed HBI / 32 DIGIOs

Highlights

- 2/3-port EtherCAT slave controller with 8 Fieldbus Memory Management Units (FMMUs) and 8 SyncManagers
- Interfaces to most 8/16-bit embedded controllers and 32-bit embedded controllers with an 8/16-bit bus
- Integrated Ethernet PHYs with HP Auto-MDIX
- Wake on LAN (WoL) support
- Low power mode allows systems to enter sleep mode until addressed by the Master
- Cable diagnostic support
- 1.8V to 3.3V variable voltage I/O
- Integrated 1.2V regulator for single 3.3V operation

Target Applications

- Motor Motion Control
- Process/Factory Automation
- Communication Modules, Interface Cards
- Sensors
- Hydraulic & Pneumatic Valve Systems
- Operator Interfaces

Key Benefits

- Integrated high-performance 100Mbps Ethernet transceivers
  - Compliant with IEEE 802.3/802.3u (Fast Ethernet)
  - Signal Quality Index diagnostics
  - Loop-back modes
  - Automatic polarity detection and correction
  - HP Auto-MDIX
  - Compatible with EtherCAT P
- EtherCAT slave controller
  - Supports 8 FMMUs
  - Supports 8 SyncManagers
  - Distributed clock support allows synchronization with other EtherCAT devices
  - 8K bytes of DPRAM
- 8/16-Bit Host Bus Interface
  - Indexed register, multiplexed or demultiplexed bus
  - Allows local host to enter sleep mode until addressed by EtherCAT Master
  - SPI / SQI (Quad SPI) support
- Digital I/O Mode for optimized system cost
  - 32 available Digital I/Os
- 3rd port for flexible network configurations
- Comprehensive power management features
  - 3 power-down levels
  - Wake on link status change (energy detect)
  - Magic packet wakeup, Wake on LAN (WoL), wake on broadcast, wake on perfect DA
  - Wakeup indicator event signal
- Power and I/O
  - Integrated power-on reset circuit
  - Latch-up performance exceeds 150mA per EIA/JESD78, Class II
  - JEDEC Class 3A ESD performance
  - Single 3.3V power supply (integrated 1.2V regulator)
- Additional Features
  - EEPROM emulation
  - Transformer-less link support
  - Multifunction GPIOs
  - Ability to use low cost 25MHz crystal for reduced BOM
  - 25MHz clock output for reference clock daisy chaining
- Packaging
  - Pb-free RoHS compliant 80-pin TQFP-EP
- Available in commercial, industrial, and extended industrial temp. ranges

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Documentation

To obtain the most up-to-date version of this documentation, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include -literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

Table of Contents

1.0 PREFACE

1.1 General Terms

TABLE 1-1: GENERAL TERMS

Term	Description
10BASE-T	10 Mbps Ethernet, IEEE 802.3 compliant
100BASE-TX	100 Mbps Fast Ethernet, IEEE802.3u compliant
ADC	Analog-to-Digital Converter
ALR	Address Logic Resolution
AN	Auto-Negotiation
BLW	Baseline Wander
BM	Buffer Manager - Part of the switch fabric
BPDU	Bridge Protocol Data Unit - Messages which carry the Spanning Tree Protocol information
Byte	8 bits
CSMA/CD	Carrier Sense Multiple Access/Collision Detect
CSR	Control and Status Registers
CTR	Counter
DA	Destination Address
DWORD	32 bits
EPC	EEPROM Controller
FCS	Frame Check Sequence - The extra checksum characters added to the end of an Ethernet frame, used for error detection and correction.
FIFO	First In First Out buffer
FSM	Finite State Machine
GPIO	General Purpose I/O
Host	External system (Includes processor, application software, etc.)
IGMP	Internet Group Management Protocol
Inbound	Refers to data input to the device from the host
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.
lsb	Least Significant Bit
LSB	Least Significant Byte
LVDS	Low Voltage Differential Signaling
MDI	Medium Dependent Interface
MDIX	Media Independent Interface with Crossover
MII	Media Independent Interface
MIIM	Media Independent Interface Management
MIL	MAC Interface Layer
MLD	Multicast Listening Discovery
MLT-3	Multi-Level Transmission Encoding (3-Levels). A tri-level encoding method 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”.
msb	Most Significant Bit
MSB	Most Significant Byte

TABLE 1-1: GENERAL TERMS (CONTINUED)

Term	Description
NRZI	Non Return to Zero Inverted. This encoding method inverts the signal for a “1” and leaves the signal unchanged for a “0”
N/A	Not Applicable
NC	No Connect
OUI	Organizationally Unique Identifier
Outbound	Refers to data output from the device to the host
PISO	Parallel In Serial Out
PLL	Phase Locked Loop
PTP	Precision Time Protocol
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.
RTC	Real-Time Clock
SA	Source Address
SFD	Start of Frame Delimiter - The 8-bit value indicating the end of the preamble of an Ethernet frame.
SIPO	Serial In Parallel Out
SMI	Serial Management Interface
SQE	Signal Quality Error (also known as “heartbeat”)
SSD	Start of Stream Delimiter
UDP	User Datagram Protocol - A connectionless protocol run on top of IP networks
UUID	Universally Unique IDentifier
WORD	16 bits

1.2 Buffer Types

TABLE 1-2: BUFFER TYPES

BUFFER TYPE	DESCRIPTION
IS	Schmitt-triggered input
O8	3.3V output with 8 mA sink and 8 mA source
OD8	3.3V open-drain output with 8 mA sink
OD12	3.3V open-drain output with 12 mA sink
OS12	3.3V open-source output with 12 mA source
VIS	Variable voltage Schmitt-triggered input
VO8	Variable voltage output with 8 mA sink and 8 mA source
VOD8	Variable voltage open-drain output with 8 mA sink
VOS8	Variable voltage open-source output with 8 mA source
VO12	Variable voltage output with 12 mA sink and 12 mA source
VOD12	Variable voltage open-drain output with 12 mA sink
VOS12	Variable voltage open-source output with 12 mA source
PU	50 µA (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 the device. When connected to a load that must be pulled high, an external resistor must be added.
PD	50 µA (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 the device. When connected to a load that must be pulled low, an external resistor must be added.
AI	Analog input
AIO	Analog bidirectional
ICLK	Crystal oscillator input pin
OCLK	Crystal oscillator output pin
P	Power pin

1.3 Register Nomenclature

TABLE 1-3: REGISTER NOMENCLATURE

Register Bit Type Notation	Register Bit Description
R	**Read:** A register or bit with this attribute can be read.
W	**Read:** A register or bit with this attribute can be written.
RO	**Read only:** Read only. Writes have no effect.
WO	**Write only:** If a register or bit is write-only, reads will return unspecified data.
W1S	**Write One to Set:** Writing a one sets the value. Writing a zero has no effect
W1C	**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 are self-cleared after the being set. Writes of zero have no effect. Contents can be read.
RO/LH	**Read Only, Latch High:** Bits with this attribute will stay high until the bit is read. After it is read, the bit will either remain high if the high condition remains, or will go low if the high condition has been removed. If the bit has not been read, the bit will remain high regardless of a change to the high condition. This mode is used in some Ethernet PHY registers.
NASR	**Not Affected by Software Reset.** The state of NASR bits do not change on assertion of a software reset.
RESERVED	**Reserved Field:** Reserved fields must be written with zeros to ensure future compatibility. The value of reserved bits is not guaranteed on a read.

2.0 GENERAL DESCRIPTION

The LAN9254 is a 2/3-port EtherCAT slave controller with dual integrated Ethernet PHYs, each containing a full-duplex 100BASE-TX transceiver that supports 100Mbps (100BASE-TX) operation. The LAN9254 supports HP Auto-MDIX, allowing the use of direct connect or cross-over LAN cables. EtherCAT P and Signal Quality Index PHY diagnostics are supported.

The LAN9254 includes an EtherCAT slave controller with 8K bytes of Dual Port memory (DPRAM) and 8 Fieldbus Memory Management Units (FMMUs). Each FMMU performs the task of mapping logical addresses to physical addresses. The EtherCAT slave controller also includes 8 SyncManagers to allow the exchange of data between the EtherCAT master and the local application. Each SyncManager's direction and mode of operation is configured by the EtherCAT master. Two modes of operation are available: buffered mode or mailbox mode. In the buffered mode, both the local microcontroller and EtherCAT master can write to the device concurrently. The buffer within the LAN9254 will always contain the latest data. If newer data arrives before the old data can be read out, the old data will be dropped. In mailbox mode, access to the buffer by the local microcontroller and the EtherCAT master is performed using handshakes, guaranteeing that no data will be dropped.

The LAN9254 provides an EtherCAT direct mapping mode, which inserts the EtherCAT registers into the base address space. The EtherCAT direct mapping mode eliminates the need for a command/data access structure, which can increase the speed of smaller data blocks.

The LAN9254 supports numerous power management and wakeup features. The LAN9254 can be placed in a reduced power mode and can be programmed to issue an external wake signal (IRQ or PME) via several methods, including “Magic Packet”, “Wake on LAN”, wake on broadcast, wake on perfect DA, and “Link Status Change”. This signal is ideal for triggering system power-up using remote Ethernet wakeup events. The device can be removed from the low power state via a host processor command or one of the wake events.

The LAN9254 contains an I^2C master EEPROM controller for connection to an EEPROM, allowing for the storage and retrieval of static data. The internal EEPROM Loader automatically loads stored configuration settings from the EEPROM into the device at reset. An EEPROM emulation feature, which requires additional software support, is available for EEPROM-less operation.

For simple digital modules without microcontrollers, the LAN9254 can also operate in Digital I/O Mode where 32 digital signals can be controlled or monitored by the EtherCAT master.

To enable star or tree network topologies, the device can be configured as a 3-port slave, providing an additional MII port. This port can be connected to an external PHY, forming a tap along the current daisy chain, or to another LAN9254 creating a 4-port solution. The MII port can point upstream (as Port 0) or downstream (as Port 2).

LED support consists of a standard RUN indicator and a LINK / Activity indicator per port. Configuration options enable ERR and STATE_RUN indicators.

A 64-bit distributed clock is included to enable high-precision synchronization and to provide accurate information about the local timing of data acquisition.

The LAN9254 can be configured to operate via a single 3.3V supply utilizing an integrated 3.3V to 1.2V linear regulator. The linear regulator may be optionally disabled, allowing usage of a high efficiency external regulator for lower system power dissipation.

The LAN9254 is available in commercial, industrial, and extended industrial temperature ranges. Figure 2-1 details a typical system application, while Figure 2-2 provides an internal block diagram of the LAN9254.

Note: Extended temp. (105ºC) is supported only with an external voltage regulator (internal regulator must be disabled) and 2.5V (typ) Ethernet magnetics.

FIGURE 2-1: SYSTEM BLOCK DIAGRAM

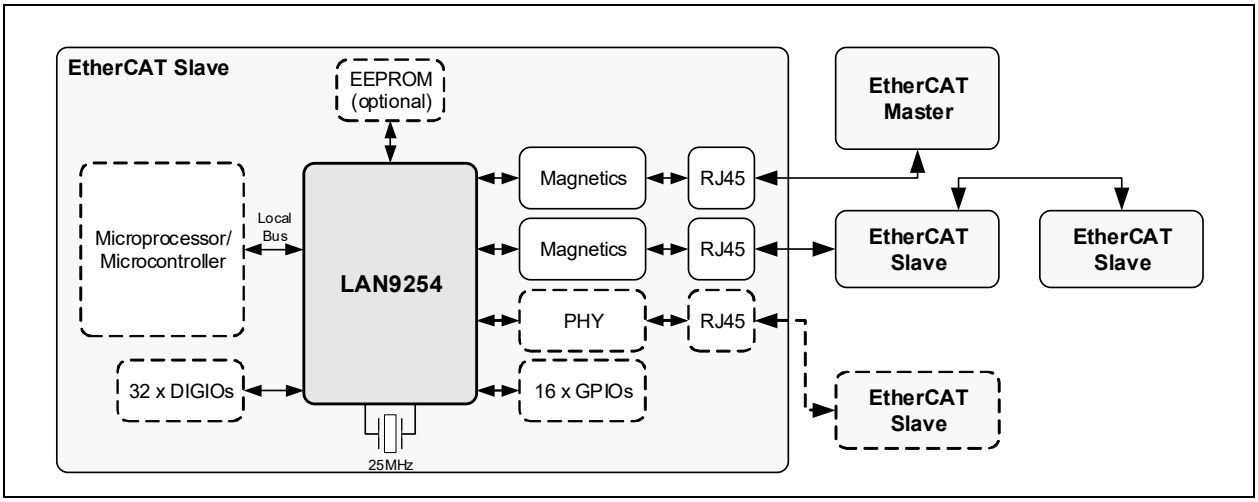
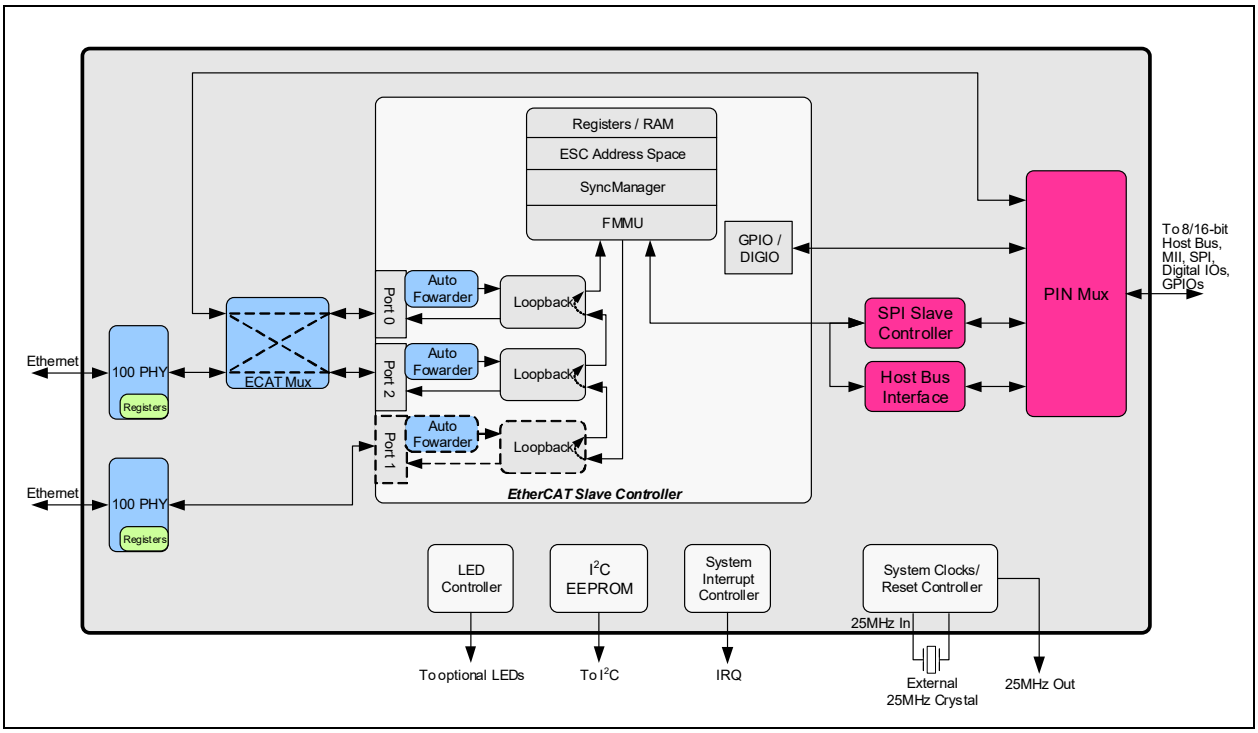


FIGURE 2-2: INTERNAL BLOCK DIAGRAM



The LAN9254 can operate in Microcontroller, Expansion, or Digital I/O mode:

Microcontroller Mode: The LAN9254 communicates with the microcontroller through an SRAM-like slave interface. The simple, yet highly functional host bus interface provides a glue-less connection to most common 8 or 16-bit microprocessors and microcontrollers as well as 32-bit microprocessors with an 8 or 16-bit external bus.

The integrated Host Bus Interface (HBI) supports 8/16-bit operation with big, little, and mixed endian operations. Two process data RAM FIFOs interface the HBI to the EtherCAT slave controller and facilitate the transferring of process data information between the host CPU and the EtherCAT slave. A configurable host interrupt pin allows the device to inform the host CPU of any internal interrupts.

Three user selectable Microcontroller Mode Host Bus Interface (HBI) options are available:

- ***Indexed register access***
  This implementation provides three index/data register banks, each with independent Byte/WORD to DWORD conversion. Internal registers are accessed by first writing one of the three index registers, followed by reading or writing the corresponding data register. Three index/data register banks support up to 3 independent driver threads without access conflicts. Each thread can write its assigned index register without the issue of another thread overwriting it. Two 16-bit cycles or four 8-bit cycles are required within the same 32-bit index/data register - however, these access can be interleaved. Non-indexed read and write accesses are supported to the process data FIFOs. The non-indexed FIFO access provides independent Byte/WORD to DWORD conversion, supporting interleaved accesses with the index/data registers.
- ***Multiplexed address/data bus***
  This implementation provides a multiplexed address and data bus with both single phase and dual phase address support. The address is loaded with an address strobe followed by data access using a read or write strobe. Two back to back 16-bit data cycles or 4 back to back 8-bit data cycles are required within the same 32-bit DWORD. These accesses must be sequential without any interleaved accesses to other registers.
- ***Demultiplexed address/data bus***
  This implementation provides a demultiplexed address and data bus with the address and endianness select inputs directly provided by the host. Two back to back 16-bit data cycles or 4 back to back 8-bit data cycles are required within the same 32-bit DWORD. These accesses must be sequential without any interleaved accesses to other registers.

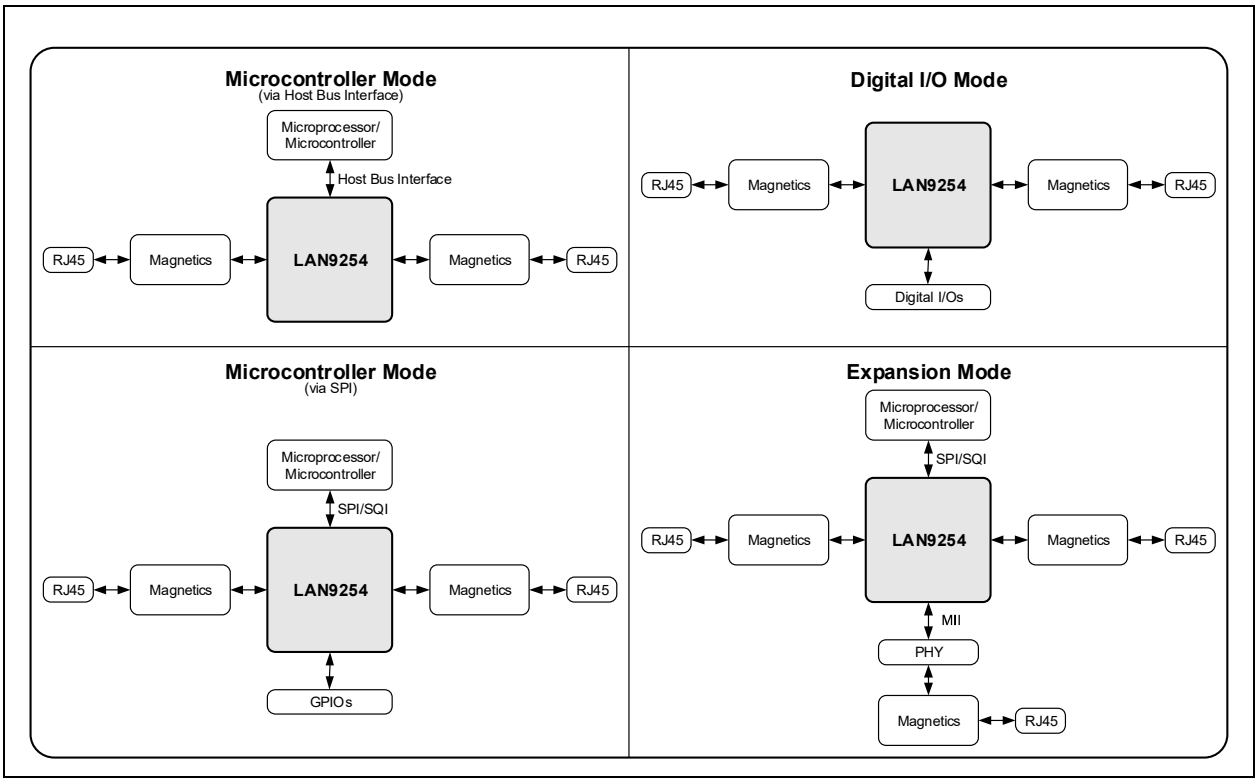
Alternatively, the device can be accessed via SPI/SQI, while also providing up to 16 inputs or outputs for general purpose usage. An SPI / SQI (Quad SPI) slave controller provides a low pin count synchronous slave interface that facilitates communication between the device and a host system. The SPI / SQI slave allows access to the System CSRs, internal FIFOs and memories. It supports single and multiple register read and write commands with incrementing, decrementing and static addressing. Single, Dual and Quad bit lanes are supported with a clock rate of up to 80 MHz.

Expansion Mode: While the device is in SPI/SQI mode, a third networking port can be enabled to provide an additional MII port. This port can be connected to an external PHY, to enable star or tree network topologies, or to another LAN9254 to create a four port solution. This port can be configured for the upstream or downstream direction.

Digital I/O Mode: For simple digital modules without microcontrollers, the LAN9254 can operate in Digital I/O Mode where 32 digital signals can be controlled or monitored by the EtherCAT master. Up to 7 control signals are also provided.

Figure 2-3 provides a system level overview of each mode of operation.

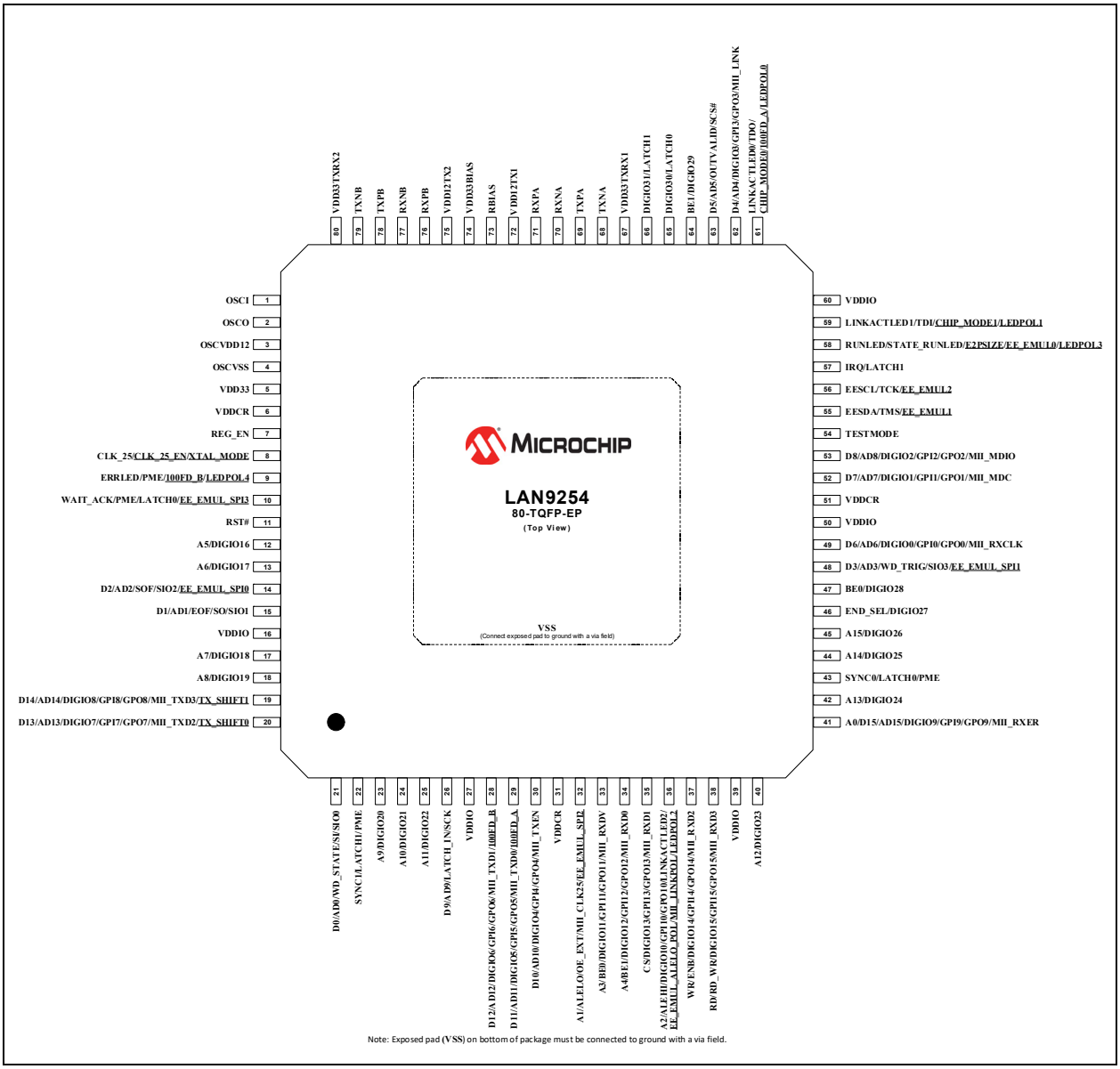
FIGURE 2-3: MODES OF OPERATION



3.0 PIN DESCRIPTIONS AND CONFIGURATION

3.1 Pin Assignments

FIGURE 3-1: PIN ASSIGNMENTS (TOP VIEW)



Note: When a “#” is used at the end of the signal name, it indicates that the signal is active low. For example, **RST#** indicates that the reset signal is active low.

Configuration straps are identified by an underlined symbol name. Refer to Section 3.3, "Configuration Straps," on page 36 for additional information.

The buffer type for each signal is indicated in the “Buffer Type” column of the pin description tables in Section 3.2, "Pin Descriptions". A description of the buffer types is provided in Section 1.2, "Buffer Types".

Table 3-1 details the pin assignments in table format. As shown, select pin functions may change based on the device's mode of operation. For modes where a specific pin has no function, the table cell will be marked with "-".

TABLE 3-1: PIN ASSIGNMENTS

Pin Number	HBI Indexed Mode Pin Name	HBI Multiplexed Mode Pin Name	HBI Demultiplexed Mode Pin Name	SPI/SQI with GPIO Mode Pin Name	SPI/SQI with MII Mode Pin Name	Digital I/O Mode Pin Name
1	OSCI					
2	OSCO					
3	OSCVDD12					
4	OSCVSS					
5	VDD33					
6	VDDCR					
7	REG_EN					
8	CLK_25/CLK_25_EN/XTAL_MODE					
9	ERRLED/PME/100FD_B/LEDPOL4				ERRLED/ PME/ LEDPOL4	ERRLED/ 100FD_B/ LEDPOL4
10	WAIT_ACK/PME			WAIT_ACK/PME/EE_EMUL_S-PI3		LATCH0
11	RST#					
12	-	-	A5	-	-	DIGIO16
13	-	-	A6	-	-	DIGIO17
14	D2	AD2	D2	SIO2/EE_EMUL_SPI0		SOF
15	D1	AD1	D1	SO/SIO1		EOF
16	VDDIO					
17	-	-	A7	-	-	DIGIO18
18	-	-	A8	-	-	DIGIO19
19	D14	AD14	D14	GPI8/GPO8	MII_TXD3/ TX_SHIFT1	DIGIO8
20	D13	AD13	D13	GPI7/GPO7	MII_TXD2/ TX_SHIFT0	DIGIO7
21	D0	AD0	D0	SI/SIO0		WD_STATE
22	SYNC1/LATCH1/PME					SYNC1/ LATCH1
23	-	-	A9	-	-	DIGIO20
24	-	-	A10	-	-	DIGIO21
25	-	-	A11	-	-	DIGIO22
26	D9	AD9	D9	SCK		LATCH_IN
27	VDDIO					

TABLE 3-1: PIN ASSIGNMENTS (CONTINUED)

Pin Number	HBI Indexed Mode Pin Name	HBI Multiplexed Mode Pin Name	HBI Demultiplexed Mode Pin Name	SPI/SQI with GPIO Mode Pin Name	SPI/SQI with MII Mode Pin Name	Digital I/O Mode Pin Name
28	D12	AD12	D12	GPI6/GPO6	MII_TXD1/ 100FD_B	DIGIO6
29	D11	AD11	D11	GPI5/GPO5	MII_TXD0/ 100FD_A	DIGIO5
30	D10	AD10	D10	GPI4/GPO4	MII_TXEN	DIGIO4
31	VDDCR					
32	A1	ALELO	A1	-	MII_CLK25/ EE_EMUL_SPI2	OE_EXT
33	A3	BE0	A3	GPI11/GPO11	MII_RXDV	DIGIO11
34	A4	BE1	A4	GPI12/GPO12	MII_RXD0	DIGIO12
35	CS			GPI13/GPO13	MII_RXD1	DIGIO13
36	A2	ALEHI/ EE_EMUL_ALELO_POL	A2	GPI10/GPO10	LINKACTLED2/ MII_LINKPOL/ LEDPOL2	DIGIO10
37	WR/ENB			GPI14/GPO14	MII_RXD2	DIGIO14
38	RD/RD_WR			GPI15/GPO15	MII_RXD3	DIGIO15
39	VDDIO					
40	-	-	A12	-	-	DIGIO23
41	A0/D15	AD15	A0/D15	GPI9/GPO9	MII_RXER	DIGIO9
42	-	-	A13	-	-	DIGIO24
43	SYNC0/LATCH0/PME					SYNC0/ LATCH0
44	-	-	A14	-	-	DIGIO25
45	-	-	A15	-	-	DIGIO26
46	-	-	END_SEL	-	-	DIGIO27
47	-	-	BE0	-	-	DIGIO28
48	D3	AD3	D3	SIO3/EE_EMUL_SPI1		WD_TRIG
49	D6	AD6	D6	GPI0/GPO0	MII_RXCLK	DIGIO0
50	VDDIO					
51	VDDCR					
52	D7	AD7	D7	GPI1/GPO1	MII_MDC	DIGIO1
53	D8	AD8	D8	GPI2/GPO2	MII_MDIO	DIGIO2
54	TESTMODE					
55	EESDA/TMS/EE_EMUL1					EESDA/TMS
56	EESCL/TCK/EE_EMUL2					EESCL/TCK
57	IRQ					LATCH1

TABLE 3-1: PIN ASSIGNMENTS (CONTINUED)

Pin Number	HBI Indexed Mode Pin Name	HBI Multiplexed Mode Pin Name	HBI Demultiplexed Mode Pin Name	SPI/SQI with GPIO Mode Pin Name	SPI/SQI with MII Mode Pin Name	Digital I/O Mode Pin Name
58	RUNLED/STATE_RUNLED/E2PSIZE/EE_EMUL0/LEDPOL3					RUNLED/ STATE_RUNLED/ E2PSIZE/ LEDPOL3
59	LINKACTLED1/TDI/CHIP_MODE1/LEDPOL1					
60	VDDIO					
61	LINKACTLED0/TDO/CHIP_MODE0/100FD_A/LEDPOL0				LINKACTLED0/ TDO/ CHIP_MODE0/ LEDPOL0	LINKACTLED0/ TDO/ CHIP_MODE0/ 100FD_A/ LEDPOL0
62	D4	AD4	D4	GPI3/GPO3	MII_LINK	DIGIO3
63	D5	AD5	D5	SCS#		OUTVALID
64	-	-	BE1	-	-	DIGIO29
65	LATCH0					DIGIO30
66	LATCH1					DIGIO31
67	VDD33TXRX1					
68	TXNA					
69	TXPA					
70	RXNA					
71	RXPA					
72	VDD12TX1					
73	RBIAS					
74	VDD33BIAS					
75	VDD12TX2					
76	RXPB					
77	RXNB					
78	TXPB					
78	TXNB					
80	VDD33TXRX2					
Exposed Pad	VSS					

3.2 Pin Descriptions

This section contains descriptions of the various LAN9254 pins. The pin descriptions have been broken into functional groups as follows:

- LAN Port A Pins
- LAN Port B Pins
- LAN Port A & B Power and Common Pins
- EtherCAT MII Port & Configuration Strap Pins
- Host Bus Pins
- SPI/SQI Pins
- EtherCAT Distributed Clock Pins
- EtherCAT Digital I/O and GPIO Pins
- EEPROM Pins
- LED & Configuration Strap Pins
- Miscellaneous Pins
- JTAG Pins
- Core and I/O Power and Ground Pins

Note: Table 3-1 details how the functions described in this section are mapped on the physical device pins. Not all functions are used for all modes of operation.

TABLE 3-2: LAN PORT A PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Port A TP TX/ RX Positive Channel 1	**TXPA**	AIO	Port A Twisted Pair Transmit/Receive Positive Channel 1. See Note 3-1.
1	Port A TP TX/ RX Negative Channel 1	**TXNA**	AIO	Port A Twisted Pair Transmit/Receive Negative Channel 1. See Note 3-1.
1	Port A TP TX/ RX Positive Channel 2	**RXPA**	AIO	Port A Twisted Pair Transmit/Receive Positive Channel 2. See Note 3-1.
1	Port A TP TX/ RX Negative Channel 2	**RXNA**	AIO	Port A Twisted Pair Transmit/Receive Negative Channel 2. See Note 3-1.

Note 3-1 Either channel 1 or 2 may function as the transmit pair while the other channel functions as the receive pair. The pin name symbols for the twisted pair pins apply to a normal connection. If Auto-MDIX is enabled and a reverse connection is detected or manually selected, the RX and TX pins will be swapped internally.

Note: Port A is connected EtherCAT port 0 or 2.

TABLE 3-3: LAN PORT B PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Port B TP TX/ RX Positive Channel 1	**TXPB**	AIO	Port B Twisted Pair Transmit/Receive Positive Channel 1. See Note 3-2.
1	Port B TP TX/ RX Negative Channel 1	**TXNB**	AIO	Port B Twisted Pair Transmit/Receive Negative Channel 1. See Note 3-2.
1	Port B TP TX/ RX Positive Channel 2	**RXPB**	AIO	Port B Twisted Pair Transmit/Receive Positive Channel 2. See Note 3-2.
1	Port B TP TX/ RX Negative Channel 2	**RXNB**	AIO	Port B Twisted Pair Transmit/Receive Negative Channel 2. See Note 3-2.

Note 3-2 Either channel 1 or 2 may function as the transmit pair while the other channel functions as the receive pair. The pin name symbols for the twisted pair pins apply to a normal connection. If Auto-MDIX is enabled and a reverse connection is detected or manually selected, the RX and TX pins will be swapped internally.

Note: Port B is connected EtherCAT port 1.

TABLE 3-4: LAN PORT A & B POWER AND COMMON PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Bias Reference	**RBIAS**	AI	Used for internal bias circuits. Connect to an external 12.1 kΩ, 1% resistor to ground. Refer to the device reference schematic for connection information. **Note:** The nominal voltage is 1.2 V and the resistor will dissipate approximately 1 mW of power.
1	+3.3 V Port A Analog Power Supply	**VDD33TXRX1**	P	See Note 3-3.
1	+3.3 V Port B Analog Power Supply	**VDD33TXRX2**	P	See Note 3-3.
1	+3.3 V Master Bias Power Supply	**VDD33BIAS**	P	See Note 3-3.

TABLE 3-4: LAN PORT A & B POWER AND COMMON PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Port A Transmitter +1.2 V Power Supply	**VDD12TX1**	P	This pin is supplied from the **VDDCR** supply either externally or from the internal voltage regulator. This pin must be tied to the **VDD12TX2** pin for proper operation. See Note 3-3.
1	Port B Transmitter +1.2 V Power Supply	**VDD12TX2**	P	This pin is supplied from the **VDDCR** supply either externally or from the internal voltage regulator. This pin must be tied to the **VDD12TX1** pin for proper operation. See Note 3-3.

Note 3-3 Refer to the Power Connections section of the datasheet, the device reference schematic, and the device LANCheck schematic checklist for additional connection information.

TABLE 3-5: ETHERCAT MII PORT & CONFIGURATION STRAP PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	25MHz Clock	**MII_CLK25**	VO12 Note 3-4	This pin is a free running 25MHz clock that can be used as the clock input to the PHY.
	EEPROM Emulation SPI Configuration Strap 2	**EE_EMUL_SPI2** Note 3-5	VIS (PD)	This configuration strap, along with **EE_EMUL_SPI0**, **EE_EMUL_SPI1**, and **EE_EMUL_SPI3**, configures the operation of the Beckhoff SPI interface during EEPROM Emulation mode. **EE_EMUL_SPI2** configures SCS# polarity. **0:** SCS# active low **1:** SCS# active high
4	Receive Data MII Port	**MII_RXD[3:0]**	VIS (PD)	These pins are the receive data from the external PHY.
1	Receive Data Valid MII Port	**MII_RXDV**	VIS (PD)	This pin is the receive data valid signal from the external PHY.
1	Receive Error MII Port	**MII_RXER**	VIS (PD)	This pin is the receive error signal from the external PHY.
1	Receive Clock MII Port	**MII_RXCLK**	VIS (PD)	This pin is the receive clock from the external PHY.

TABLE 3-5: ETHERCAT MII PORT & CONFIGURATION STRAP PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
4	Transmit Data MII Port	**MII_TXD[3:0]**	VO8	These pins are the transmit data to the external PHY.
	MII Transmit Timing Shift Configuration Straps	**TX_SHIFT[1:0]** Note 3-5	VIS (PU) Note 3-6	These straps configure the value of the external MII Bus TX timing. **TX_SHIFT1** is on the **MII_TXD3** pin and **TX_SHIFT0** is on the **MII_TXD2** pin. **TX_SHIFT[1:0]** **00:** 20ns **01:** 30ns **10:** 0ns **11:** 10ns
	100Mbps Full Duplex Configuration Strap B	**100FD_B** Note 3-5	VIS (PD) Note 3-6	For 3 port mode (as selected by **CHIP_MODE1**), this strap configures the default of the ANEG Disable PHY B and AMDIX Disable PHY B fields in the Hardware Configuration Register (HW_CFG) and sets the PHY to fixed 100Mbps full-duplex operation by default. **0:** Auto-negotiation and AMDIX enabled by default **1:** Auto-negotiation and AMDIX disabled (fixed 100Mbps full-duplex) by default **100FD_B** is located on the **MII_TXD1** pin. **Note:** In 2 port mode, this strap is on the **ERRLED** pin.
	100Mbps Full Duplex Configuration Strap A	**100FD_A** Note 3-5	VIS (PD) Note 3-6	For 3 port mode (as selected by **CHIP_MODE1**), this strap configures the default of the ANEG Disable PHY A and AMDIX Disable PHY A fields in the Hardware Configuration Register (HW_CFG) and sets the PHY to fixed 100Mbps full-duplex operation by default. **0:** Auto-negotiation and AMDIX enabled by default **1:** Auto-negotiation and AMDIX disabled (fixed 100Mbps full-duplex) by default **100FD_A** is located on the **MII_TXD0** pin. **Note:** In 2 port mode, this strap is on the **LINKACTLED0** pin.
1	Transmit Data Enable MII Port	**MII_TXEN**	VO8	This pin is the transmit data enable signal to the external PHY.
1	Link Status MII Port	**MII_LINK**	VIS	This pin is the provided by the PHY to indicate that a 100 Mbit/s Full Duplex link is established. The polarity is configurable via the **MII_LINKPOL** configuration strap.

TABLE 3-5: ETHERCAT MII PORT & CONFIGURATION STRAP PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	SMI Clock	MII_MDC	VO8	This pin is the serial management clock to the external PHY.
1	SMI Data	MII_MDIO	VIS/ VO8	This pin is the serial management Interface data input/output to the external PHY. **Note:** An external pull-up is required to ensure that the non-driven state of the MDIO signal is a logic one.

Note 3-4 A series terminating resistor is recommended for the best PCB signal integrity.

Note 3-5 Configuration strap pins are identified by an underlined symbol name. Configuration strap values are latched on power-on reset, EtherCAT reset, or RST# de-assertion. Refer to Section 3.3, "Configuration Straps," on page 36 for further information.

Note 3-6 An external supplemental pull-up or pull-down may be needed, depending upon the input current loading of the external MAC/PHY device.

TABLE 3-6: HOST BUS PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Read	RD	VIS	This pin is the host bus read strobe. Normally active low, the polarity can be changed via configuration register settings.
	Read or Write	RD_WR	VIS	This pin is the host bus direction control. Used in conjunction with the ENB pin it indicates a read or write operation. The normal polarity is read when 1, write when 0 (R/nW) but can be changed via configuration register settings.
1	Write	WR	VIS	This pin is the host bus write strobe. Normally active low, the polarity can be changed via configuration register settings.
	Enable	ENB	VIS	This pin is the host bus data enable strobe. Used in conjunction with the RD_WR pin it indicates the data phase of the operation. Normally active low, the polarity can be changed via configuration register settings.
1	Chip Select	CS	VIS	This pin is the host bus chip select and indicates that the device is selected for the current transfer. Normally active low, the polarity can be changed via configuration register settings.

TABLE 3-6: HOST BUS PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
2	Byte Enable	**BE1** **BE0**	VIS (PD)	In 16-bit data mode, these pins indicate which byte(s) is(are) to be written or read. In 8-bit data mode, these pins are not used. These pins are only available in multiplexed and demultiplexed modes. Normally active low, the polarity can be changed via configuration register settings. **Note:** The location of these signals varies dependent on the device mode. **Note:** The pull-down holds the undriven state of these pins to active, such that previous PCB designs which do not drive these signals may work with 16-bit only bus cycles.
16	Address	**A[15:0]**	VIS	These pins provide the address for indexed and demultiplexed address modes. In 16-bit data mode, bit 0 is not used. Address bit 0 is shared with data bit 15. In Indexed Address Mode, A[15:5] are not used.
16	Data	**D[15:0]**	VIS/ VO8	These pins are the host bus data bus for non-multiplexed address modes. In 8-bit data mode, bits 15-8 are not used. Their input and output drivers are disabled. Address bit 0 is shared with data bit 15.
	Address and Data	**AD[15:0]**	VIS/ VO8	These pins are the host bus address / data bus for multiplexed address mode. Bits 15-8 provide the upper byte of address for single phase multiplexed address mode. Bits 7-0 provide the lower byte of address for single phase multiplexed address mode and both bytes of address for dual phase multiplexed address mode. In 8-bit data dual phase multiplexed address mode, bits 15-8 are not used. Their input and output drivers are disabled.

TABLE 3-6: HOST BUS PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Address Latch Enable High	**ALEHI**	VIS	This pin indicates the address phase for multiplexed address modes. It is used to load the higher address byte in dual phase multiplexed address mode. Normally active low (address saved on rising edge), the polarity can be changed via configuration register settings.
	EEPROM Emulation ALELO Polarity Strap 0	**EE_EMUL_ALELO_POL** Note 3-7	VIS (PU)	During EEPROM Emulation mode, if the default PDI selection is set to HBI Multiplexed 1 Phase, this strap is used to set the HBI ALE polarity until the EEPROM configuration data has been loaded.
1	Address Latch Enable Low	**ALELO**	VIS	This pin indicates the address phase for multiplexed address modes. It is used to load both address bytes in single phase multiplexed address mode and the lower address byte in dual phase multiplexed address mode. Normally active low (address saved on rising edge), the polarity can be changed via configuration register settings.
1	Wait / Acknowledge	**WAIT_ACK**	VO8/ VOD8	This pin indicates when the host bus cycle may be finished. This pin is tri-state when the device is not selected. Normally push-pull, the buffer type can be changed to open-drain via configuration register settings. Normally active low indicating wait, for push-pull operation, the polarity can be changed via configuration register settings. This pin is disabled when both bits are low.
1	Endianess Select	**END_SEL**	VIS	This pin provides the endianess control for demultiplexed address mode. A high chooses big endian mode and a low chooses little endian mode. This can be dynamically changed or held static.

Note 3-7 Configuration strap pins are identified by an underlined symbol name. Configuration strap values are latched on power-on reset, EtherCAT reset, or **RST#** de-assertion. Refer to Section 3.3, "Configuration Straps," on page 36 for further information.

TABLE 3-7: SPI/SQI PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	SPI/SQI Slave Chip Select	**SCS#**	VIS (PU)	This pin is the SPI/SQI slave chip select input. When low, the SPI/SQI slave is selected for SPI/SQI transfers. When high, the SPI/SQI serial data output(s) is(are) 3-stated. When the Beckhoff SPI interface is used, the polarity of this signal is controlled via an internal register.
1	SPI/SQI Slave Serial Clock	**SCK**	VIS (PU)	This pin is the SPI/SQI slave serial clock input.
4	SPI/SQI Slave Serial Data Input/Output	**SIO[3:0]**	VIS/ VO8 (PU)	These pins are the SPI/SQI slave data input and output for multiple bit I/O.
	SPI Slave Serial Data Input	**SI**	VIS (PU)	This pin is the SPI slave serial data input. SI is shared with SIO0.
	SPI Slave Serial Data Output	**SO**	VO8 (PU) Note 3-8	This pin is the SPI slave serial data output. SO is shared with SIO1.
	EEPROM Emulation SPI Configuration Strap 1	**EE_EMUL_SPI1** Note 3-9	VIS (PU)	This configuration strap, along with **EE_EMUL_SPI0**, **EE_EMUL_SPI2**, and **EE_EMUL_SPI3**, configures the operation of the Beckhoff SPI interface during EEPROM Emulation mode. **EE_EMUL_SPI[1:0]** configure the SPI mode. **EE_EMUL_SPI1** is located on the **SIO3** pin. **EE_EMUL_SPI[1:0]** **00:** SPI mode 0 **01:** SPI mode 1 **10:** SPI mode 2 **11:** SPI mode 3
	EEPROM Emulation SPI Configuration Strap 0	**EE_EMUL_SPI0** Note 3-9	VIS (PU)	This configuration strap, along with **EE_EMUL_SPI1**, **EE_EMUL_SPI2**, and **EE_EMUL_SPI3**, configures the operation of the Beckhoff SPI interface during EEPROM Emulation mode. **EE_EMUL_SPI[1:0]** configure the SPI mode. Refer to the **EE_EMUL_SPI1** pin description for details. **EE_EMUL_SPI0** is located on the **SIO2** pin.

TABLE 3-7: SPI/SQI PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Wait / Acknowledge	WAIT_ACK	VO8/ VOD8	This pin indicates when the host bus cycle may be finished. This pin is tri-state when the device is not selected. Normally push-pull, the buffer type can be changed to open-drain via configuration register settings. Normally active low indicating wait, for push-pull operation, the polarity can be changed via configuration register settings. This pin is disabled when both bits are low. When the Beckhoff SPI interface is used, this pin is not used and is disabled.
	EEPROM Emulation SPI Configuration Strap 3	EE_EMUL_SPI3 Note 3-9	VIS (PD)	This configuration strap, along with EE_EMUL_SPI0, EE_EMUL_SPI1, and EE_EMUL_SPI2, configures the operation of the Beckhoff SPI interface during EEPROM Emulation mode. EE_EMUL_SPI3 configures the Data Out sample mode. **0:** Normal Data Out sample (SPI_DO and SPI_DI are sampled at the same SPI_CLK edge) **1:** Late Data Out sample (SPI_DO and SPI_DI are sampled at different SPI_CLK edges)

Note 3-8 Although this pin is an output for SPI instructions, it includes a pull-up, since it is actually SIO bit 1.

Note 3-9 Configuration strap pins are identified by an underlined symbol name. Configuration strap values are latched on power-on reset, EtherCAT reset, or RST# de-assertion. Refer to Section 3.3, "Configuration Straps," on page 36 for further information.

TABLE 3-8: ETHERCAT DISTRIBUTED CLOCK PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
2	Sync	**SYNC1** **SYNC0**	VO8/ VOD8/ VOS8	These pins are the Distributed Clock Sync (OUT) signals. **Note:** These signals are not driven (high impedance) until the EEPROM is loaded.
2 (See Note)	Latch	**LATCH1** **LATCH0**	VIS	These pins are the Distributed Clock Latch (IN) signals. **Note:** Normally shared with the **SYNC0/SYNC1** functions, these signals are mapped onto alternate pins in some device configurations when the SYNC function is enabled. Refer to Section 11.2.1, SYNC/LATCH Pin Multiplexing for additional information.

TABLE 3-9: ETHERCAT DIGITAL I/O AND GPIO PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
16	General Purpose Input	**GPI[15:0]**	VIS	These pins are the general purpose inputs and are directly mapped into the General Purpose Input Register. Consistency of the general purpose inputs is not provided.
	General Purpose Output	**GPO[15:0]**	VO8/ VOD8	These pins are the general purpose outputs and reflect the values of the General Purpose Output Register without watchdog protection. **Note:** These signals are not driven (high impedance) until the EEPROM is loaded.
32	Digital I/O	**DIGIO[31:0]**	VIS/ VO8	These pins are the Input/Output or Bidirectional data. **Note:** These signals are not driven (high impedance) until the EEPROM is loaded.
1	Output Valid	**OUTVALID**	VO8	This pin indicates that the outputs are valid and can be captured into external registers. **Note:** This signal is not driven (high impedance) until the EEPROM is loaded.
1	Latch In	**LATCH_IN**	VIS	This pin is the external data latch signal. The input data is sampled each time a rising edge of **LATCH_IN** is recognized.
1	Watchdog Trigger	**WD_TRIG**	VO8	This pin is the SyncManager Watchdog Trigger output. **Note:** This signal is not driven (high impedance) until the EEPROM is loaded.

TABLE 3-9: ETHERCAT DIGITAL I/O AND GPIO PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Watchdog State	WD_STATE	VO8	This pin is the SyncManager Watchdog State output. A 0 indicates the watchdog has expired. **Note:** This signal is not driven (high impedance) until the EEPROM is loaded.
1	Start of Frame	SOF	VO8	This pin is the Start of Frame output and indicates the start of an Ethernet/EtherCAT frame. **Note:** This signal is not driven (high impedance) until the EEPROM is loaded.
1	End of Frame	EOF	VO8	This pin is the End of Frame output and indicates the end of an Ethernet/EtherCAT frame. **Note:** This signal is not driven (high impedance) until the EEPROM is loaded.
1	Output Enable	OE_EXT	VIS	This pin is the Output Enable input. When low, it clears the output data.

TABLE 3-10: EEPROM PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	EEPROM I^2C Serial Data Input/Output	EESDA	VIS/ VOD8	When the device is accessing an external EEPROM, this pin is the I^2C serial data input/open-drain output. **Note:** This pin must be pulled-up by an external resistor at all times.
	EEPROM Emulation Configuration Strap 1	EE_EMUL1 Note 3-10	VIS	This strap, along with EE_EMUL2, configures the value of the EEPROM emulation hard-strap. Either low enables emulation. This strap, along with EE_EMUL0 and EE_EMUL2 configures the default PDI selection during EEPROM Emulation mode. Refer to the EE_EMUL2 pin description for details.

TABLE 3-10: EEPROM PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
	EEPROM I^2C Serial Clock	**EESCL**	VIS/ VOD8	When the device is accessing an external EEPROM this pin is the I^2C clock input/open-drain output. **Note:** If an EEPROM is used, this pin must be pulled-up by an external resistor at all times.
1	EEPROM Emulation Configuration Strap 2	**EE_EMUL2** Note 3-10	VIS	This strap, along with **EE_EMUL1**, configures the value of the EEPROM emulation hard-strap. Either low enables emulation. This strap, along with **EE_EMUL0** and **EE_EMUL1** configures the default PDI selection during EEPROM Emulation mode. **EE_EMUL[2:0]** **000:** SPI **001:** HBI Demultiplexed 16-bit EtherCAT Direct Mapped **010:** HBI Multiplexed 1 Phase 16-bit EtherCAT Direct Mapped **011:** HBI Multiplexed 2 Phase 16-bit EtherCAT Direct Mapped **100:** SPI EtherCAT Direct Mapped **101:** Beckhoff SPI Mode **110:** N/A (EEPROM is enabled) **111:** N/A (EEPROM is enabled)

Note 3-10 Configuration strap pins are identified by an underlined symbol name. Configuration strap values are latched on power-on reset, EtherCAT reset, or **RST#** de-assertion. Refer to Section 3.3, "Configuration Straps," on page 36 for further information.

TABLE 3-11: LED & CONFIGURATION STRAP PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	ERR LED	**ERRLED**	OD12/ OS12	This pin is the ERR LED output and is controlled either by the ESC or the local MCU. This pin is configured to be an open-drain/open-source output. The default choice of open-drain vs. open-source as well as the default polarity of this pin depends upon the strap value sampled at reset. This pin is enabled as an output via bit 14 - ERR LED Enable of the ASIC Configuration Register (0x0142:0x0143). If active low, this pin is forced enabled in the event of an EEPROM loading error.
	100Mbps Full Duplex Configuration Strap B	**100FD_B** / **LEDPOL4** Note 3-12	IS (PD)	For 2 port mode (as selected by **CHIP_MODE1**), this strap configures the default of the ANEG Disable PHY B and AMDIX Disable PHY B fields in the Hardware Configuration Register (HW_CFG) and sets the PHY to fixed 100Mbps full-duplex operation by default. **0:** Auto-negotiation and AMDIX enabled by default **1:** Auto-negotiation and AMDIX disabled (fixed 100Mbps full-duplex) by default In 3 port mode (as selected by **CHIP_MODE1**), this strap is moved to the **MII_TXD1** pin. **Note:** If an external pull-up is used, it should be connected to **VDD33**. **Note:** If **ERRLED** is used as the PME output (with potential to be a wired-OR'd shared signal), careful consideration must be paid to the strap value. Host software might need to correct the strap results via the AMDIX Disable PHY B and ANEG Disable PHY B bits in the Hardware Configuration Register (HW_CFG) as well as reconfiguring PHY B.
	LED 4 Polarity Configuration Strap			Since this pin is shared with a configuration strap, the default polarity of the **ERRLED** pin is determined during strap loading. If the strap value is 0, the LED is set as active high by default, since it is assumed that if an LED is present it is used as the pull-down. See Note 3-11. If the strap value is 1, the LED is set as active low by default, since it is assumed that an LED to **VDD33** is used as the pull-up.

TABLE 3-11: LED & CONFIGURATION STRAP PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	RUN LED	**RUNLED**	VOD12/ VOS12	This pin is the RUN LED output and is controlled by the AL Status Register. This pin is configured to be open-drain/open-source output. The default choice of open-drain vs. open-source as well as the default polarity of this pin depends upon the strap value sampled at reset.
	STATE_RUN LED	**STATE_RUNLED**	VOD12/ VOS12	This pin is the STATE_RUN LED output and is equal to the RUN LED combined with the negation of the ERR LED. It can be used to control the RUN side of a bi-color RUN/ERR LED. This pin is configured to be open-drain/open-source output. The default choice of open-drain vs. open-source as well as the default polarity of this pin depends upon the strap value sampled at reset. The selection between RUN and STATE_RUN is via bit 6 - STATE_RUN LED enable, of the ASIC Configuration Register (0x0142:0x0143).
	EEPROM Size Configuration Strap	**E2PSIZE** / **EE_EMUL0** / **LEDPOL3** Note 3-12	VIS (PU)	This strap configures the size of the EEPROM. **0:** Selects 1K bits (128 x 8) through 16K bits (2K x 8). **1:** 32K bits (4K x 8) through 4Mbits (512K x 8).
	EEPROM Emulation Configuration Strap 0			This strap, along with **EE_EMUL1** and **EE_EMUL2** configures the default PDI selection during EEPROM Emulation mode. Refer to the **EE_EMUL2** pin description for details.
	LED 3 Polarity Configuration Strap			Since this pin is shared with a configuration strap, the default polarity of the RUNLED / STATE_RUN-LED pin is determined during strap loading. If the strap value is 0, the LED is set as active high be default, since it is assumed that an LED to ground is used as the pull-down. See Note 3-11. If the strap value is 1, the LED is set as active low by default, since it is assumed that an LED to **VDDIO** is used as the pull-up.

TABLE 3-11: LED & CONFIGURATION STRAP PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Link / Activity LED Port 2	**LINKACTLED2**	VOD12/ VOS12	This pin is the Link/Activity LED output (off=no link, on=link without activity, blinking=link and activity) for port 2. This pin is configured to be open-drain/open-source output. The default choice of open-drain vs. open-source as well as the default polarity of this pin depends upon the strap value sampled at reset.
	MII Port Link Polarity	**MII_LINKPOL** / **LEDPOL2** Note 3-12	VIS (PU)	This strap configures the polarity of the **MII_LINK** pin. **0: MII_LINK** low indicates a 100 Mbit/s Full-Duplex link is established. **1: MII_LINK** high indicates a 100 Mbit/s Full-Duplex link is established.
	LED 2 Polarity Configuration Strap			Since this pin is shared with a configuration strap, the default polarity of the **LINKACTLED2** pin is determined during strap loading. If the strap value is 0, the LED is set as active high by default, since it is assumed that an LED to ground is used as the pull-down. See Note 3-11. If the strap value is 1, the LED is set as active low by default, since it is assumed that an LED to **VDDIO** is used as the pull-up.

TABLE 3-11: LED & CONFIGURATION STRAP PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Link / Activity LED Port 1	**LINKACTLED1**	VOD12/ VOS12	This pin is the Link/Activity LED output (off=no link, on=link without activity, blinking=link and activity) for port 1. This pin is configured to be open-drain/open-source output. The default choice of open-drain vs. open-source as well as the default polarity of this pin depends upon the strap value sampled at reset.
	Chip Mode Strap 1	**CHIP_MODE1** / **LEDPOL1** Note 3-12	VIS (PU)	This strap, along with **CHIP_MODE0**, configures the value of the EtherCAT Chip Mode: **CHIP_MODE[1:0]** **0X:** 2-Port Mode. Ports 0 and 1 are connected to the internal PHYs A and B. **10:** 3-Port Downstream Mode. Ports 0 and 1 are connected to internal PHYs A and B. Port 2 is connected to the external MII pins. **11:** 3-Port Upstream Mode. Ports 2 and 1 are connected to internal PHYs A and B. Port 0 is connected to the external MII pins.
	LED 1 Polarity Configuration Strap			Since this pin is shared with a configuration strap, the default polarity of the **LINKACTLED1** pin is determined during strap loading. If the strap value is 0, the LED is set as active high by default, since it is assumed that an LED to ground is used as the pull-down. See Note 3-11 If the strap value is 1, the LED is set as active low by default, since it is assumed that an LED to **VDDIO** is used as the pull-up.

TABLE 3-11: LED & CONFIGURATION STRAP PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Link / Activity LED Port 0	**LINKACTLED0**	VOD12/ VOS12	This pin is the Link/Activity LED output (off=no link, on=link without activity, blinking=link and activity) for port 0. This pin is configured to be open-drain/open-source output. The default choice of open-drain vs. open-source as well as the default polarity of this pin depends upon the strap value sampled at reset.
	Chip Mode Strap 0	**CHIP_MODE0** / **100FD_A** / **LEDPOL0** Note 3-12	VIS (PU)	This strap, along with **CHIP_MODE1**, configures the value of the Chip Mode hard-strap. Refer to the **CHIP_MODE1** description for details on the chip mode strap settings. **Note:** **CHIP_MODE0** is unused in 2 port mode.
	100Mbps Full Duplex Configuration Strap A			For 2 port mode (as selected by **CHIP_MODE1**), this strap configures the default of the ANEG Disable PHY A and AMDIX Disable PHY A fields in the Hardware Configuration Register (HW_CFG) and sets the PHY to fixed 100Mbps full-duplex operation by default. **0:** Auto-negotiation and AMDIX enabled by default **1:** Auto-negotiation and AMDIX disabled (fixed 100Mbps full-duplex) by default In 3 port mode (as selected by **CHIP_MODE1**), this strap is moved to the **MII_TXD0** pin.
	LED 0 Polarity Configuration Strap			Since this pin is shared with a configuration strap, the default polarity of the **LINKACTLED0** pin is determined during strap loading. If the strap value is 0, the LED is set as active high by default, since it is assumed that an LED to ground is used as the pull-down. See Note 3-11. If the strap value is 1, the LED is set as active low by default, since it is assumed that an LED to **VDDIO** is used as the pull-up.

Note 3-11 When using a LED as a pull-down strap, a external supplemental pull-down is needed to ensure a valid low level.

Note 3-12 Configuration strap pins are identified by an underlined symbol name. Configuration strap values are latched on power-on reset, EtherCAT reset, or **RST#** de-assertion. Refer to Section 3.3, "Configuration Straps," on page 36 for further information.

TABLE 3-12: MISCELLANEOUS PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Interrupt Output	**IRQ**	VO8/ VOD8	Interrupt request output. The polarity, source and buffer type of this signal is programmable via configuration register settings.
0 (See Note)	Power Management Event Output	**PME**	VO8/ VOD8 O8/ OD8 (see note)	When programmed accordingly, this signal is asserted upon detection of a wakeup event. The polarity and buffer type of this signal is programmable via an internal register. Refer to the Power Management section for additional information on the power management features. **Note:** Optionally enabled, this pin may be mapped onto several other pins via an internal register. **Note:** When mapped into the **ERRLED** pin, this signal is in the **VDD33** voltage domain and has an O8/OD8 buffer type. **Note:** When mapped onto the **ERRLED** pin, in the event of an EEPROM loading error, the **ERRLED** function is forced enabled and false **PME** events might occur.
1	System Reset Input	**RST#**	VIS/ VOD8 (PU)	As an input, this active low signal allows external hardware to reset the device. The device also contains an internal power-on reset circuit. Thus this signal may be left unconnected if an external hardware reset is not needed. When used, this signal must adhere to the reset timing requirements as detailed in the Operational Characteritics section. As an output, this signal is driven low during POR or in response to an EtherCAT reset command sequence from the Master Controller or Host interface.
1	Regulator Enable	**REG_EN**	AI	When tied to 3.3 V, the internal 1.2 V regulators are enabled.
1	Test Mode Select	**TESTMODE**	VIS (PD)	This pin must be tied to VSS for proper operation.
1	Crystal Input	**OSCI**	ICLK	External 25 MHz crystal input. This signal can also be over-driven by a single-ended clock oscillator. When this method is used, **OSCO** should be left unconnected. **Note:** In clock daisy chaining configuration, using the **CLK_25** of the previous devices as the input clock source, this pin should be set to Schmitt trigger input mode via the **XTAL_MODE** strap input.

TABLE 3-12: MISCELLANEOUS PINS (CONTINUED)

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Crystal Output	**OSCO**	OCLK	External 25 MHz crystal output.
1	Crystal Clock Output	**CLK_25**	O8	25 MHz crystal output.
1	Crystal Clock Output Enable Configuration Strap	**CLK_25_EN** Note 3-14	AI (see note)	This multiple DC level strap enables the **CLK_25** output and selects between the operation of a crystal oscillator amplifier or a Schmitt trigger input on the **OSCI** pin. A level below 0.8V disables the **CLK_25** output and selects the crystal oscillator amplifier. A level of 1.5V enables the **CLK_25** output and selects the crystal oscillator amplifier. A level above 2.2V enables the **CLK_25** output and selects the Schmitt trigger input. **Note:** If left floated during strap load, the pin will be biased to **VDD33**/2. An external voltage divider is not required. If an external pull-up is used to set this pin above 2.2V, it should be connected to **VDD33**. An external pull-down or connecting to **VSS** may be used to set this pin to below 0.8V.
	Crystal Clock Input Mode Configuration Strap	**XTAL_MODE** Note 3-14		
1	Crystal +1.2 V Power Supply	**OSCVDD12**	P	Supplied by the on-chip regulator unless configured for regulator off mode via the **REG_EN** pin. See Note 3-13.
1	Crystal Ground	**OSCVSS**	P	Crystal ground.

Note 3-13 Refer to the Power Connections section, the device reference schematic, and the device LANCheck schematic checklist for additional connection information.

Note 3-14 Configuration strap pins are identified by an underlined symbol name. Configuration strap values are latched on power-on reset, EtherCAT reset, or **RST#** de-assertion. Refer to Section 3.3, "Configuration Straps," on page 36 for further information.

TABLE 3-13: JTAG PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	JTAG Test Mux Select	**TMS**	VIS	JTAG test mode select.
1	JTAG Test Clock	**TCK**	VIS	JTAG test clock.
1	JTAG Test Data Input	**TDI**	VIS	JTAG data input.
1	JTAG Test Data Output	**TDO**	VO12	JTAG data output.

TABLE 3-14: CORE AND I/O POWER AND GROUND PINS

NUM PINS	NAME	SYMBOL	BUFFER TYPE	DESCRIPTION
1	Regulator +3.3 V Power Supply	**VDD33**	P	+3.3 V power supply for internal regulators. **Note:** +3.3 V must be supplied to this pin even if the internal regulators are disabled. See Note 3-15.
5	+1.8 V to +3.3 V Variable I/O Power	**VDDIO**	P	+1.8 V to +3.3 V variable I/O power. See Note 3-15.
3	+1.2 V Digital Core Power Supply	**VDDCR**	P	Supplied by the on-chip regulator unless configured for regulator off mode via the **REG_EN** pin. 1 µF and 470 pF decoupling capacitors in parallel to ground should be used on pin 6 (the regulator output pin). See Note 3-15.
1	Ground	**VSS**	P	Common ground. This exposed pad must be connected to the ground plane with a via array. **Note:** The crystal oscillator has its own ground pin **OSCVSS**.

Note 3-15 Refer to the Power Connections section, the device reference schematic, and the device LANCheck schematic checklist for additional connection information.

3.3 Configuration Straps

Configuration straps allow various features of the device to be automatically configured to user defined values. Configuration straps are latched upon Power-On Reset (POR), EtherCAT reset, or pin reset (**RST#**). Configuration straps are identified by an underlined symbol name and are defined throughout Section 3.2, "Pin Descriptions," on page 16.

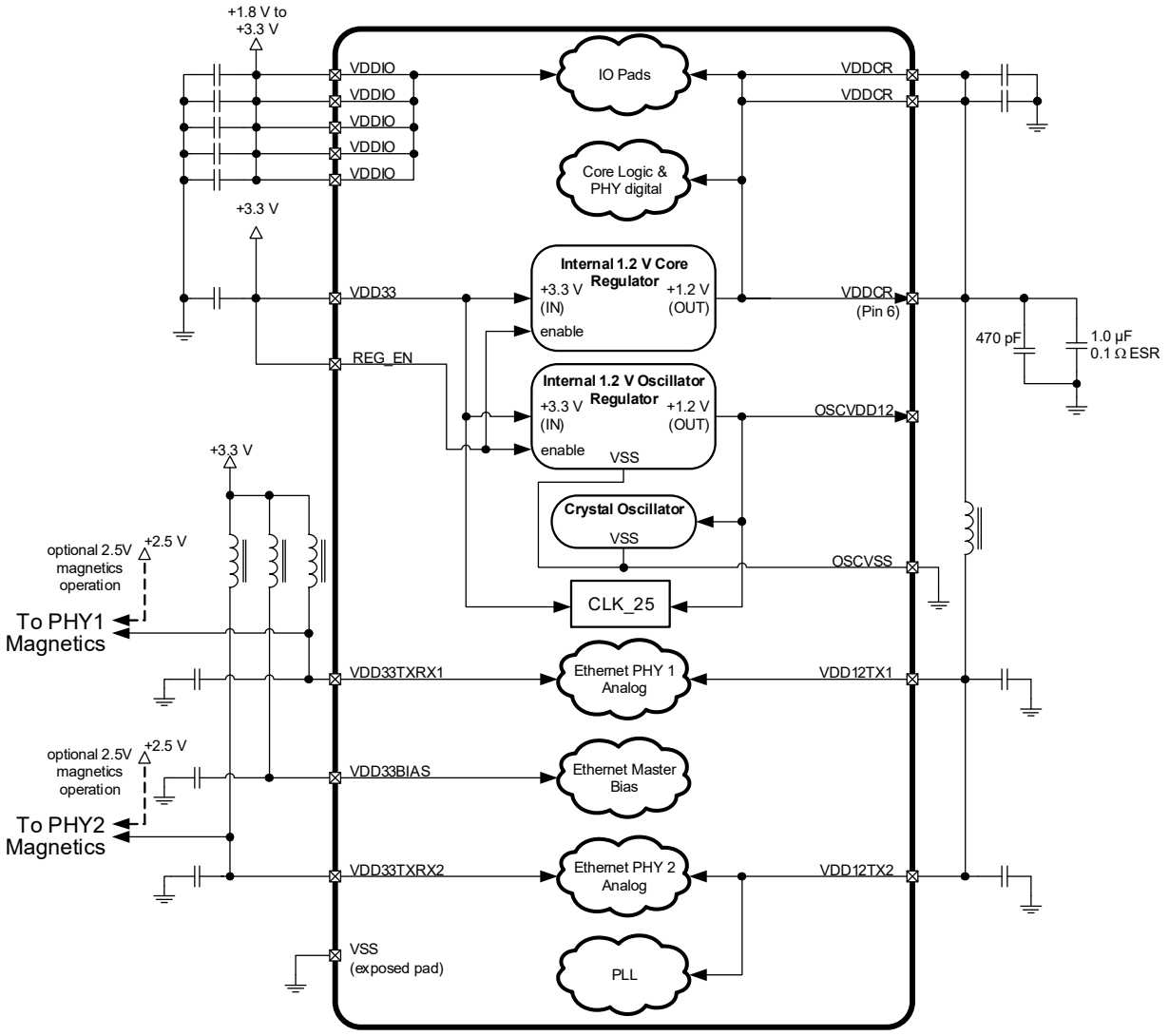
Configuration straps include internal resistors in order to prevent the signal from floating when unconnected. If a particular configuration strap is connected to a load, an external pull-up or pull-down resistor should be used to augment the internal resistor to ensure that it reaches the required voltage level prior to latching. The internal resistor can also be overridden by the addition of an external resistor.

Note:	The system designer must guarantee that configuration strap pins meet the timing requirements specified in Section 17.6.3, "Reset and Configuration Strap Timing". If configuration strap pins are not at the correct voltage level prior to being latched, the device may capture incorrect strap values.

4.0 POWER CONNECTIONS

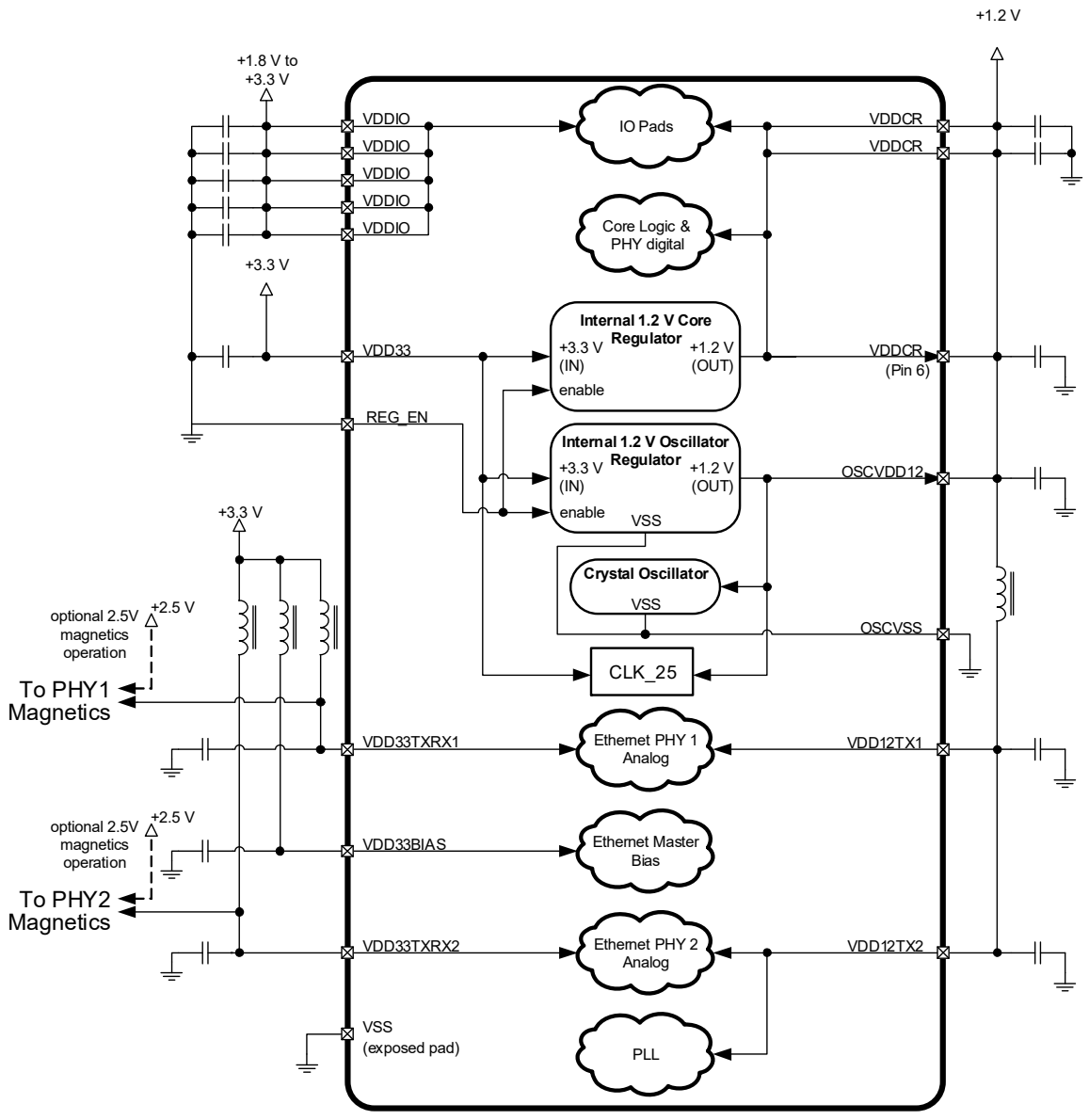
Figure 4-1 and Figure 4-2 illustrate the device power connections for regulator enabled and disabled cases, respectively. Refer to the device reference schematic and the device LANCheck schematic checklist for additional information. Section 4.1 provides additional information on the devices internal voltage regulators.

FIGURE 4-1: POWER CONNECTIONS - REGULATORS ENABLED



Note: Bypass and bulk caps as needed for PCB

FIGURE 4-2: POWER CONNECTIONS - REGULATORS DISABLED



Note: Bypass and bulk caps as needed for PCB

4.1 Internal Voltage Regulators

The device contains two internal 1.2 V regulators:

- 1.2 V Core Regulator
- 1.2 V Crystal Oscillator Regulator

4.1.1 1.2 V CORE REGULATOR

The core regulator supplies 1.2 V volts to the main core digital logic, the I/O pads, and the PHY's digital logic and can be used to supply the 1.2 V power to the PHY analog sections (via an external connection).

When the **REG_EN** input pin is connected to 3.3 V, the core regulator is enabled and receives 3.3 V on the **VDD33** pin. A 1.0 uF 0.1 Ω ESR capacitor must be connected to the **VDDCR** pin associated with the regulator.

When the **REG_EN** input pin is connected to **VSS**, the core regulator is disabled. However, 3.3 V must still be supplied to the **VDD33** pin. The 1.2 V core voltage must then be externally input into the **VDDCR** pins.

4.1.2 1.2 V CRYSTAL OSCILLATOR REGULATOR

The crystal oscillator regulator supplies 1.2 V volts to the crystal oscillator and the **CLK_25** pin. When the **REG_EN** input pin is connected to 3.3 V, the crystal oscillator regulator is enabled and receives 3.3 V on the **VDD33** pin. An external capacitor is not required.

When the **REG_EN** input pin is connected to **VSS**, the crystal oscillator regulator is disabled. However, 3.3 V must still be supplied to the **VDD33** pin. The 1.2 V crystal oscillator voltage must then be externally input into the **OSCVDD12** pin.

5.0 REGISTER MAP

This chapter details the device register map and summarizes the various directly addressable System Control and Status Registers (CSRs). Detailed descriptions of the System CSRs are provided in the chapters corresponding to their function. Additional indirectly addressable registers are available in the various sub-blocks of the device. These registers are also detailed in their corresponding chapters.

Directly Addressable Registers

- Section 11.15, "EtherCAT CSR and Process Data RAM Access Registers (Directly Addressable)," on page 305
- EtherCAT Core Control and Status Registers and Process RAM while EtherCAT Direct Mapped mode
- Section 5.1, "System Control and Status Registers," on page 42

Indirectly Addressable Registers

- Section 10.2.18, "PHY Registers," on page 225
- Section 11.16, "EtherCAT Core CSR Registers (Indirectly Addressable)," on page 314

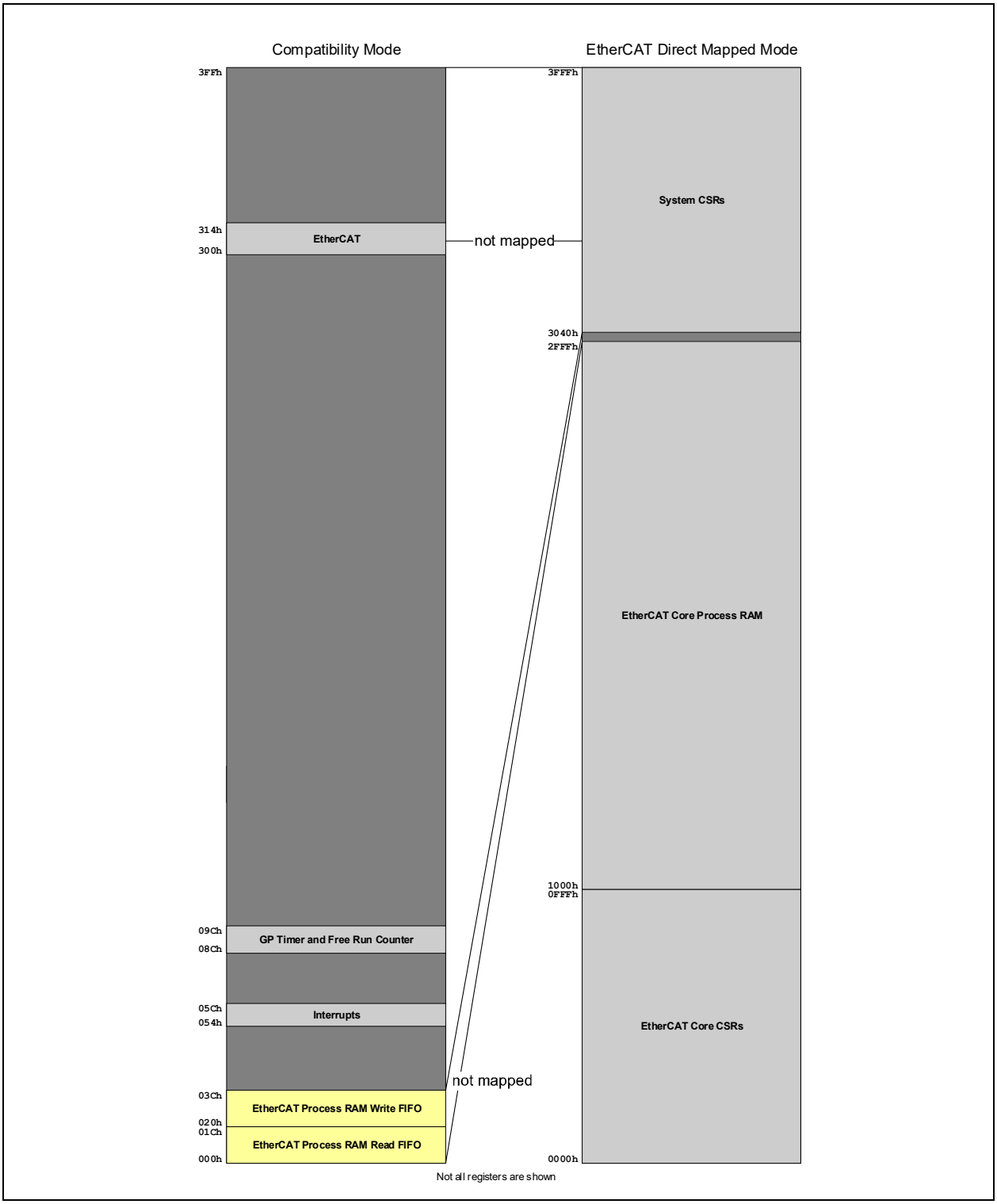
Figure 5-1 contains an overall base register memory map of the device. This memory map is not drawn to scale, and should be used for general reference only. Table 5-1 provides a summary of all directly addressable CSRs and their corresponding addresses.

There are two address modes. By default, the register mapping is compatible to the LAN9252 with the EtherCAT Core Control and Status Registers and Process RAM accessed via a command and data register structure. Once enabled, EtherCAT Direct Mapped mode maps the EtherCAT Core Control and Status Registers and EtherCAT Core Process RAM to there native addresses (byte address 0h to FFFh and 1000h to 2FFFh respectively) while remapping the System Control and Status Registers starting at a base offset of 3000h.

Note:	The Section 11.15, "EtherCAT CSR and Process Data RAM Access Registers (Directly Addressable)," on page 305 (including the EtherCAT Process RAM Read and Write Data FIFO) are not used and are not accessible during EtherCAT Direct Mapped mode.

Note:	Register bit type definitions are provided in Section 1.3, "Register Nomenclature," on page 7. Not all device registers are memory mapped or directly addressable. For details on the accessibility of the various device registers, refer the register sub-sections listed above.

FIGURE 5-1: REGISTER ADDRESS MAP



5.1 System Control and Status Registers

The System CSRs are directly addressable memory mapped registers with a base address offset range of 050h to 314h or from 3050h to 31FCh while in EtherCAT Direct Mapped mode. These registers are addressable by the Host via the Host Bus Interface (HBI) or SPI/SQI. For more information on the various device modes and their corresponding address configurations, see Section 2.0, "General Description," on page 8.

Table 5-1 lists the System CSRs and their corresponding addresses in order. All system CSRs are reset to their default value on the assertion of a chip-level reset.

The System CSRs can be divided into the following sub-categories. Each of these sub-categories is located in the corresponding chapter and contains the System CSR descriptions of the associated registers. The register descriptions are categorized as follows:

- Section 6.2.3, "Reset Registers," on page 50
- Section 6.3.5, "Power Management Registers," on page 56
- Section 7.3, "Interrupt Registers," on page 63
- Section 11.15, "EtherCAT CSR and Process Data RAM Access Registers (Directly Addressable)," on page 305
- Section 15.1, "Miscellaneous System Configuration & Status Registers," on page 397
- Section 14.3, "General Purpose Timer and Free-Running Clock Registers," on page 394

Note: Unlisted registers are reserved for future use.

TABLE 5-1: SYSTEM CONTROL AND STATUS REGISTERS

Address	EtherCAT Direct Mapped Mode	Register Name (Symbol)
000h-01Ch	N/A	EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA)
020h-03Ch	N/A	EtherCAT Process RAM Write Data FIFO (ECAT_PRAM_WR_DATA)
050h	3050h	Chip ID and Revision (ID_REV)
054h	3054h	Interrupt Configuration Register (IRQ_CFG)
058h	3058h	Interrupt Status Register (INT_STS)
05Ch	305Ch	Interrupt Enable Register (INT_EN)
064h	3064h	Byte Order Test Register (BYTE_TEST)
074h	3074h	Hardware Configuration Register (HW_CFG)
084h	3084h	Power Management Control Register (PMT_CTRL)
08Ch	308Ch	General Purpose Timer Configuration Register (GPT_CFG)
090h	3090h	General Purpose Timer Count Register (GPT_CNT)
09Ch	309Ch	Free Running 25MHz Counter Register (FREE_RUN)
1F8h	31F8h	Reset Control Register (RESET_CTL)
EtherCAT Registers		
300h	N/A	EtherCAT CSR Interface Data Register (ECAT_CSR_DATA)
304h	N/A	EtherCAT CSR Interface Command Register (ECAT_CSR_CMD)
308h	N/A	EtherCAT Process RAM Read Address and Length Register (ECAT_PRAM_RD_ADDR_LEN)
30Ch	N/A	EtherCAT Process RAM Read Command Register (ECAT_PRAM_RD_CMD)
310h	N/A	EtherCAT Process RAM Write Address and Length Register (ECAT_PRAM_WR_ADDR_LEN)
314h	N/A	EtherCAT Process RAM Write Command Register (ECAT_PRAM_WR_CMD)

5.2 Special Restrictions on Back-to-Back Cycles

5.2.1 BACK-TO-BACK WRITE-READ CYCLES

It is important to note that there are specific restrictions on the timing of back-to-back host write-read operations. These restrictions concern reading registers after any write cycle that may affect the register. In all cases there is a delay between writing to a register and the new value becoming available to be read. In other cases, there is a delay between writing to a register and the subsequent side effect on other registers.

In order to prevent the host from reading stale data after a write operation, minimum wait periods have been established. These periods are specified in Table 5-2. The host processor is required to wait the specified period of time after writing to the indicated register before reading the resource specified in the table. Note that the required wait period is dependent upon the register being read after the write.

Performing "dummy" reads of the Byte Order Test Register (BYTE_TEST) register is a convenient way to guarantee that the minimum write-to-read timing restriction is met. Table 5-2 shows the number of dummy reads that are required before reading the register indicated. The number of BYTE_TEST reads in this table is based on the minimum cycle timing of 45ns. For microprocessors with slower busses the number of reads may be reduced as long as the total time is equal to, or greater than the time specified in the table. Note that dummy reads of the BYTE_TEST register are not required as long as the minimum time period is met.

Note that depending on the host interface mode in use, the basic host interface cycle may naturally provide sufficient time between writes and read. It is required of the system design and register access mechanisms to ensure the proper timing. For example, a write and read to the same register may occur faster than a write and read to different registers.

For 8 and 16-bit write cycles, the wait time for the back-to-back write-read operation applies only to the writing of the last BYTE or WORD of the register, which completes a single DWORD transfer.

For Indexed Address mode HBI operation, the wait time for the back-to-back write-read operation applies only to access to the internal registers and FIFOs. It does not apply to the Host Bus Interface Index Registers or the Host Bus Interface Configuration Register.

TABLE 5-2: READ AFTER WRITE TIMING RULES

After Writing...	wait for this many nanoseconds...	or Perform this many Reads of BYTE_TEST… (assuming T_{cyc} of 45ns)	before reading...
any register	45	1	the same register or any other register affected by the write
Interrupt Configuration Register (IRQ_CFG)	60	2	Interrupt Configuration Register (IRQ_CFG)
Interrupt Enable Register (INT_EN)	90	2	Interrupt Configuration Register (IRQ_CFG)
	60	2	Interrupt Status Register (INT_STS)
Interrupt Status Register (INT_STS)	180	4	Interrupt Configuration Register (IRQ_CFG)
	170	4	Interrupt Status Register (INT_STS)
Power Management Control Register (PMT_CTRL)	165	4	Power Management Control Register (PMT_CTRL)
	170	4	Interrupt Configuration Register (IRQ_CFG)
	160	4	Interrupt Status Register (INT_STS)

TABLE 5-2: READ AFTER WRITE TIMING RULES (CONTINUED)

After Writing...	wait for this many nanoseconds...	or Perform this many Reads of BYTE_TEST… (assuming T_{cyc} of 45ns)	before reading...
General Purpose Timer Configuration Register (GPT_CFG)	55	2	General Purpose Timer Configuration Register (GPT_CFG)
	170	4	General Purpose Timer Count Register (GPT_CNT)
EtherCAT Process RAM Write Data FIFO (ECAT_PRAM_WR_DATA)	50	2	EtherCAT Process RAM Write Command Register (ECAT_PRAM_WR_CMD)

5.2.2 BACK-TO-BACK READ CYCLES

There are also restrictions on specific back-to-back host read operations. These restrictions concern reading specific registers after reading a resource that has side effects. In many cases there is a delay between reading the device, and the subsequent indication of the expected change in the control and status register values.

In order to prevent the host from reading stale data on back-to-back reads, minimum wait periods have been established. These periods are specified in Table 5-3. The host processor is required to wait the specified period of time between read operations of specific combinations of resources. The wait period is dependent upon the combination of registers being read.

Performing “dummy” reads of the Byte Order Test Register (BYTE_TEST) register is a convenient way to guarantee that the minimum wait time restriction is met. Table 5-3 below also shows the number of dummy reads that are required for back-to-back read operations. The number of BYTE_TEST reads in this table is based on the minimum timing for T_{cyc} (45ns). For microprocessors with slower busses the number of reads may be reduced as long as the total time is equal to, or greater than the time specified in the table. Dummy reads of the BYTE_TEST register are not required as long as the minimum time period is met.

Note that depending on the host interface mode in use, the basic host interface cycle may naturally provide sufficient time between reads. It is required of the system design and register access mechanisms to ensure the proper timing. For example, multiple reads to the same register may occur faster than reads to different registers.

For 8 and 16-bit read cycles, the wait time for the back-to-back read operation is required only after the reading of the last BYTE or WORD of the register, which completes a single DWORD transfer. There is no wait requirement between the BYTE or WORD accesses within the DWORD transfer.

TABLE 5-3: READ AFTER READ TIMING RULES

After reading...	wait for this many nanoseconds...	or Perform this many Reads of BYTE_TEST… (assuming T_{cyc} of 45ns)	before reading...
EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA)	50	2	EtherCAT Process RAM Read Command Register (ECAT_PRAM_RD_CMD)

6.0 CLOCKS, RESETS, AND POWER MANAGEMENT

6.1 Clocks

The device provides generation of all system clocks as required by the various sub-modules of the device. The clocking sub-system is comprised of the following:

- Crystal Oscillator
- PHY PLL

6.1.1 CRYSTAL OSCILLATOR

The device requires a fixed-frequency 25 MHz clock source for use by the internal clock oscillator and PLL. This is typically provided by attaching a 25 MHz crystal to the **OSCI** and **OSCO** pins as specified in Section 17.7, "Clock Circuit," on page 416. Optionally, this clock can be provided by driving the **OSCI** input pin with a single-ended 25 MHz clock source. If a single-ended source is selected, the clock input must run continuously for normal device operation.

Application Note: In clock daisy chaining configuration, using the CLK_25 of the previous devices as the input clock source, this pin should be set to Schmitt trigger input mode via the **XTAL_MODE** strap input.

Power savings modes allow for the oscillator or external clock input to be halted. The crystal oscillator can be disabled as describe in Section 6.3.4, "Chip Level Power Management," on page 54.

For system level verification, the crystal oscillator output can be enabled onto the IRQ pin. See Section 7.2.7, "Clock Output Test Mode," on page 63.

Power for the crystal oscillator is provided by a dedicated regulator or separate input pin. See Section 4.1.2, "1.2 V Crystal Oscillator Regulator," on page 39.

Note: Crystal specifications are provided in Table 17-12, “Crystal Specifications,” on page 416.

6.1.1.1 Crystal Clock Output Pin

The crystal clock may be output onto the dedicated **CLK_25** pin for use as the reference clock to another device. This pin is enabled when the **CLK_25_EN** is high.

6.1.2 PHY PLL

The PHY module receives the 25 MHz reference clock and, in addition to its internal clock usage, outputs a main system clock that is used to derive device sub-system clocks.

The PHY PLL can be disabled as describe in Section 6.3.4, "Chip Level Power Management," on page 54. The PHY PLL will be disabled only when requested *and* if the PHY ports are in a power down mode.

Power for PHY PLL is provided by an external input pin, usually sourced by the device’s 1.2V core regulator. See Section 4.0, "Power Connections," on page 37.

6.2 Resets

The device provides multiple hardware and software reset sources, which allow varying levels of the device to be reset. All resets can be categorized into three reset types as described in the following sections:

- Chip-Level Resets
  - Power-On Reset (POR)
  - RST# Pin Reset
  - EtherCAT System Reset
- Multi-Module Resets
  - DIGITAL RESET (DIGITAL_RST)
- Single-Module Resets
  - Port A PHY Reset
  - Port B PHY Reset
  - EtherCAT Controller Reset

The device supports the use of configuration straps to allow automatic custom configurations of various device parameters. These configuration strap values are set upon de-assertion of all chip-level resets and can be used to easily set the default parameters of the chip at power-on or pin (RST#) reset. Refer to Section 6.3, "Power Management," on page 51 for detailed information on the usage of these straps.

Table 6-1 summarizes the effect of the various reset sources on the device. Refer to the following sections for detailed information on each of these reset types.

TABLE 6-1: RESET SOURCES AND AFFECTED DEVICE FUNCTIONALITY

Module/ Functionality	POR	RST# Pin	EtherCAT System Reset	Digital Reset
25 MHz Oscillator	(1)			
Voltage Regulators	(2)			
EtherCAT Core	X	X	X	X
PHY A	X	X	X	
PHY B	X	X	X	
PHY Common	(3)			
Voltage Supervision	(3)			
PLL	(3)			
SPI/SQI Slave	X	X	X	X
Host Bus Interface	X	X	X	X
Power Management	X	X	X	X
General Purpose Timer	X	X	X	X
Free Running Counter	X	X	X	X
System CSR	X	X	X	X
Config. Straps Latched	YES	YES	YES	NO(4)
EEPROM Loader Run	YES	YES	YES	YES
Tristate Output Pins(5)	YES	YES	YES	
RST# Pin Driven Low	YES		YES	

Note 1: POR is performed by the XTAL voltage regulator, not at the system level
2: POR is performed internal to the voltage regulators
3: POR is performed internal to the PHY
4: Strap inputs are not re-latched
5: Only those output pins that are used for straps

6.2.1 CHIP-LEVEL RESETS

A chip-level reset event activates all internal resets, effectively resetting the entire device. A chip-level reset is initiated by assertion of any of the following input events:

- Power-On Reset (POR)
- RST# Pin Reset
- EtherCAT System Reset

Chip-level reset/configuration completion can be determined by first polling the Byte Order Test Register (BYTE_TEST). The returned data will be invalid until the Host interface resets are complete. Once the returned data is the correct byte ordering value, the Host interface resets have completed.

The completion of the entire chip-level reset must be determined by polling the READY bit of the Hardware Configuration Register (HW_CFG) or the Power Management Control Register (PMT_CTRL) until it is set. When set, the READY bit indicates that the reset has completed and the device is ready to be accessed.

With the exception of the Hardware Configuration Register (HW_CFG), Power Management Control Register (PMT_CTRL), Byte Order Test Register (BYTE_TEST), and Reset Control Register (RESET_CTL), read access to any internal resources should not be done by S/W while the READY bit is cleared. Writes to any address are invalid until the READY bit is set.

A chip-level reset involves tuning of the variable output level pads, latching of configuration straps and generation of the master reset.

CONFIGURATION STRAPS LATCHING

During POR, EtherCAT reset or **RST#** pin reset, the latches for the straps are open. Following the release of POR, EtherCAT reset or **RST#** pin reset, the latches for the straps are closed.

The **CLK_25_EN** and the **XTAL_MODE** configuration straps and the output enable, pull-up, and pull-down for the **CLK_25** pin are not controlled by the EtherCAT reset or the **RST#** pin reset; the latches for the straps are closed.

VARIABLE LEVEL I/O PAD TUNING

Following the release of the EtherCAT, POR or **RST#** pin resets, a 1 uS pulse (active low), is sent into the VO tuning circuit. 2 uS later, the output pins are enabled. The 2 uS delay allows time for the variable output level pins to tune before enabling the outputs and also provides input hold time for strap pins that are shared with output pins.

MASTER RESET AND CLOCK GENERATION RESET

Following the enabling of the output pins, the reset is synchronized to the main system clock to become the master reset. Master reset is used to generate the local resets and to reset the clocks generation.

6.2.1.1 Power-On Reset (POR)

A power-on reset occurs whenever power is initially applied to the device or if the power is removed and reapplied to the device. This event resets all circuitry within the device. Configuration straps are latched and EEPROM loading is performed as a result of this reset. The POR is used to trigger the tuning of the Variable Level I/O Pads as well as a chip-level reset.

The POR can also used as a system level reset. **RST#** becomes an open-drain output and is asserted for the POR time. Its purpose is to perform a complete reset of the EtherCAT slave and/or to hold an external PHY in reset while the EtherCAT core is in reset. As an open-drain output, RST is intended to be wired OR'd into the system reset.

Note:	The Ethernet PHY should be connected to the **RST#** pin so that the PHY is held in reset until the EtherCAT Slave is ready. Otherwise, the far end Link Partner would detect valid link signals from the PHY and would "open" its port assuming that the local EtherCAT Slave was ready. The **RST#** pin is not driven until all voltages are operational. External, system level solutions are necessary if the system needs to be held in reset during power ramp-up.

Following valid voltage levels, a POR reset typically takes approximately 21 ms.

6.2.1.2 RST# Pin Reset

Driving the RST# input pin low initiates a chip-level reset. This event resets all circuitry within the device. Use of this reset input is optional, but when used, it must be driven for the period of time specified in Section 17.6.3, "Reset and Configuration Strap Timing," on page 413. Configuration straps are latched, and EEPROM loading is performed as a result of this reset.

A RST# pin reset typically takes approximately 1.3 ms.

Note:	The RST# pin is pulled-high internally. If unused, this signal can be left unconnected. Do not rely on internal pull-up resistors to drive signals external to the device.

Please refer to Table 3-12, “Miscellaneous Pins,” on page 33 for a description of the RST# pin.

6.2.1.3 EtherCAT System Reset

An EtherCAT system reset, initiated by a special sequence of three independent and consecutive frames/commands, is functionally identical to a RST# pin reset, except that during an EtherCAT system reset, the RST# pin becomes an open-drain output and is asserted for the minimum required time of 80 ms.

The RST# is an open-drain output intended to be wired OR’d into the system reset.

Note:	The purpose of connecting the RST# pin into the system reset is to perform a complete reset of the EtherCAT slave. The EtherCAT master issues this reset in rare and extreme cases when the local microcontroller is seriously halted and can not be otherwise informed to reinitialize.

6.2.2 BLOCK-LEVEL RESETS

The block level resets contain an assortment of reset register bit inputs and generate resets for the various blocks. Block level resets can affect one or multiple modules.

6.2.2.1 Multi-Module Resets

Multi-module resets activate multiple internal resets, but do not reset the entire chip. Configuration straps are *not* latched upon multi-module resets. A multi-module reset is initiated by assertion of the following:

- DIGITAL RESET (DIGITAL_RST)

Multi-module reset/configuration completion can be determined by first polling the Byte Order Test Register (BYTE_TEST). The returned data will be invalid until the Host interface resets are complete. Once the returned data is the correct byte ordering value, the Host interface resets have completed.

The completion of the entire chip-level reset must be determined by polling the READY bit of the Hardware Configuration Register (HW_CFG) or Power Management Control Register (PMT_CTRL) until it is set. When set, the READY bit indicates that the reset has completed and the device is ready to be accessed.

With the exception of the Hardware Configuration Register (HW_CFG), Power Management Control Register (PMT_CTRL), Byte Order Test Register (BYTE_TEST), and Reset Control Register (RESET_CTL), read access to any internal resources should not be done by S/W while the READY bit is cleared. Writes to any address are invalid until the READY bit is set.

Note:	The digital reset does not reset register bits designated as NASR.

DIGITAL RESET (DIGITAL_RST)

A digital reset is performed by setting the DIGITAL_RST bit of the Reset Control Register (RESET_CTL). A digital reset will reset all device sub-modules except the Ethernet PHYs. EEPROM loading is performed following this reset. Configuration straps are *not* latched as a result of a digital reset.

A digital reset typically takes approximately 1.3 ms.

6.2.2.2 Single-Module Resets

A single-module reset will reset only the specified module. Single-module resets do *not* latch the configuration straps. A single-module reset is initiated by assertion of the following:

- Port A PHY Reset
- Port B PHY Reset
- EtherCAT Controller Reset

Port A PHY Reset

A Port A PHY reset is performed by setting the PHY_A_RST bit of the Reset Control Register (RESET_CTL) or the Soft Reset bit in the PHY x Basic Control Register (PHY_BASIC_CONTROL_x). Upon completion of the Port A PHY reset, the PHY_A_RST and Soft Reset bits are automatically cleared. No other modules of the device are affected by this reset.

Port A PHY reset completion can be determined by polling the PHY_A_RST bit in the Reset Control Register (RESET_CTL) or the Soft Reset bit in the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) until it clears. Under normal conditions, the PHY_A_RST and Soft Reset bit will clear approximately 102 uS after the Port A PHY reset occurrence.

Note:	When using the Soft Reset bit to reset the Port A PHY, register bits designated as NASR are not reset.

In addition to the methods above, the Port A PHY is automatically reset after returning from a PHY power-down mode. This reset differs in that the PHY power-down mode reset does not reload or reset any of the PHY registers. Refer to Section 10.2.9, "PHY Power-Down Modes," on page 212 for additional information.

Refer to Section 10.2.11, "Resets," on page 216 for additional information on Port A PHY resets.

Port B PHY Reset

A Port B PHY reset is performed by setting the PHY_B_RST bit of the Reset Control Register (RESET_CTL) or the Soft Reset bit in the PHY x Basic Control Register (PHY_BASIC_CONTROL_x). Upon completion of the Port B PHY reset, the PHY_B_RST and Soft Reset bits are automatically cleared. No other modules of the device are affected by this reset.

Port B PHY reset completion can be determined by polling the PHY_B_RST bit in the Reset Control Register (RESET_CTL) or the Soft Reset bit in the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) until it clears. Under normal conditions, the PHY_B_RST and Soft Reset bit will clear approximately 102 us after the Port B PHY reset occurrence.

Note:	When using the Soft Reset bit to reset the Port B PHY, register bits designated as NASR are not reset.

In addition to the methods above, the Port B PHY is automatically reset after returning from a PHY power-down mode. This reset differs in that the PHY power-down mode reset does not reload or reset any of the PHY registers. Refer to Section 10.2.9, "PHY Power-Down Modes," on page 212 for additional information.

Refer to Section 10.2.11, "Resets," on page 216 for additional information on Port B PHY resets.

EtherCAT Controller Reset

A compete device and system reset can be initiated by either the EtherCAT master or by the local host by writing the value sequence of 0x52 (‘R’), 0x45 (‘E’) and 0x53 (‘S’) into the ESC Reset ECAT Register (for the master) or the ESC Reset PDI Register (for the local host). This will trigger the reset described in Section 6.2.1.3, "EtherCAT System Reset".

A reset of just the EtherCAT Controller may be performed by setting the ETHERCAT_RST bit in the Reset Control Register (RESET_CTL).

This will reset the EtherCAT Core and its registers. It will also reset the EtherCAT CSR and Process Data RAM Access logic described in Section 11.13, on page 297 and will reset the registers described in Section 11.15, "EtherCAT CSR and Process Data RAM Access Registers (Directly Addressable)," on page 305.

Since the EtherCAT module will reconfigure the device from the EEPROM, the Host interfaces will be disabled until reset is complete. Completion of the reset must be determined by using the methods described in Section 8.4.2.2, on page 78 for HBI and Section 9.2.1.1, on page 167 for SPI/SQI.

An EtherCAT Controller reset typically takes approximately 1.3 ms.

6.2.3 RESET REGISTERS

6.2.3.1 Reset Control Register (RESET_CTL)

Offset: 1F8h / 31F8h Size: 32 bits

This register contains software controlled resets.

Note: This register can be read while the device is in the reset or not ready / power savings states without leaving the host interface in an intermediate state. If the host interface is in a reset state, returned data may be invalid.

It is not necessary to read all four bytes of this register. DWORD access rules do not apply to this register.

Bits	Description	Type	Default
31:7	**RESERVED**	RO	-
6	**EtherCAT Reset (ETHERCAT_RST)** Setting this bit resets the EtherCAT core. When the EtherCAT core is released from reset, this bit is automatically cleared. All writes to this bit are ignored while this bit is set.	R/W SC	0b
5:3	**RESERVED**	RO	-
2	**Port B PHY Reset (PHY_B_RST)** Setting this bit resets the Port B PHY. The internal logic automatically holds the PHY reset for a minimum of 102uS. When the Port B PHY is released from reset, this bit is automatically cleared. All writes to this bit are ignored while this bit is set.	R/W SC	0b
1	**Port A PHY Reset (PHY_A_RST)** Setting this bit resets the Port A PHY. The internal logic automatically holds the PHY reset for a minimum of 102uS. When the Port A PHY is released from reset, this bit is automatically cleared. All writes to this bit are ignored while this bit is set.	R/W SC	0b
0	**Digital Reset (DIGITAL_RST)** Setting this bit resets the complete chip except the PLL, Port B PHY and Port A PHY. All system CSRs are reset except for any NASR type bits. Any in progress EEPROM commands are terminated. When the chip is released from reset, this bit is automatically cleared. All writes to this bit are ignored while this bit is set.	R/W SC	0b

6.3 Power Management

The device supports several block and chip level power management features as well as wake-up event detection and notification.

6.3.1 WAKE-UP EVENT DETECTION

6.3.1.1 PHY A & B Energy Detect

Energy Detect Power Down mode reduces PHY power consumption. In energy-detect power-down mode, the PHY will resume from power-down when energy is seen on the cable (typically from link pulses) and set the ENERGYON interrupt bit in the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x).

Refer to Section 10.2.9.2, "Energy Detect Power-Down," on page 212 for details on the operation and configuration of the PHY energy-detect power-down mode.

Note: If a carrier is present when Energy Detect Power Down is enabled, then detection will occur immediately.

If enabled, via the PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x), the PHY will generate an interrupt. This interrupt is reflected in the Interrupt Status Register (INT_STS), bit 26 (PHY_INT_A) for PHY A and bit 27 (PHY_INT_B) for PHY B. The INT_STS register bits will trigger the IRQ interrupt output pin if enabled, as described in Section 7.2.1, "Ethernet PHY Interrupts," on page 61.

The energy-detect PHY interrupts will also set the appropriate Energy-Detect / WoL Status Port A (ED_WOL_STS_A) or Energy-Detect / WoL Status Port B (ED_WOL_STS_B) bit of the Power Management Control Register (PMT_CTRL). The Energy-Detect / WoL Enable Port A (ED_WOL_EN_A) and Energy-Detect / WoL Enable Port B (ED_WOL_EN_B) bits will enable the corresponding status bits as a PME event.

Note: Any PHY interrupt will set the above status bits. The Host should only enable the appropriate PHY interrupt source in the PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x).

6.3.1.2 PHY A & B Wake on LAN (WoL)

PHY A and B provide WoL event detection of Perfect DA, Broadcast, Magic Packet, and Wakeup frames.

When enabled, the PHY will detect WoL events and set the WoL interrupt bit in the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x). If enabled via the PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x), the PHY will generate an interrupt. This interrupt is reflected in the Interrupt Status Register (INT_STS), bit 26 (PHY_INT_A) for PHY A and bit 27 (PHY_INT_B) for PHY B. The INT_STS register bits will trigger the IRQ interrupt output pin if enabled, as described in Section 7.2.1, "Ethernet PHY Interrupts," on page 61.

Refer to Section 10.2.10, "Wake on LAN (WoL)," on page 213 for details on the operation and configuration of the PHY WoL.

The WoL PHY interrupts will also set the appropriate Energy-Detect / WoL Status Port A (ED_WOL_STS_A) or Energy-Detect / WoL Status Port B (ED_WOL_STS_B) bit of the Power Management Control Register (PMT_CTRL). The Energy-Detect / WoL Enable Port A (ED_WOL_EN_A) and Energy-Detect / WoL Enable Port B (ED_WOL_EN_B) bits enable the corresponding status bits as a PME event.

Note: Any PHY interrupt will set the above status bits. The Host should only enable the appropriate PHY interrupt source in the PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x).

6.3.2 WAKE-UP (PME) NOTIFICATION

A simplified diagram of the logic that controls the PME interrupt can be seen in Figure 6-1.

The PME module handles the latching of the PHY B Energy-Detect / WoL Status Port B (ED_WOL_STS_B) bit and the PHY A Energy-Detect / WoL Status Port A (ED_WOL_STS_A) bit in the Power Management Control Register (PMT_CTRL).

This module also masks the status bits with the corresponding enable bits (Energy-Detect / WoL Enable Port B (ED_WOL_EN_B) and Energy-Detect / WoL Enable Port A (ED_WOL_EN_A)) and combines the results together to generate the Power Management Interrupt Event (PME_INT) status bit in the Interrupt Status Register (INT_STS). The PME_INT status bit is then masked with the Power Management Event Interrupt Enable (PME_INT_EN) bit and combined with the other interrupt sources to drive the **IRQ** output pin.

Note:	The PME interrupt status bit (PME_INT) in the INT_STS register is set regardless of the setting of PME_INT_EN.

In addition to generating interrupt events, the PME event can also drive the **PME** output pin to indicate wake-up events exclusively. The PME event is enabled with the PME Enable (PME_EN) bit in the Power Management Control Register (PMT_CTRL), The **PME** output pin characteristics can be configured via the PME Buffer Type (PME_TYPE), PME Indication (PME_IND) and PME Polarity (PME_POL) bits of the Power Management Control Register (PMT_CTRL). These bits allow the **PME** output pin to be open-drain, active high push-pull or active-low push-pull and configure the output to be continuous, or pulse for 50 ms.

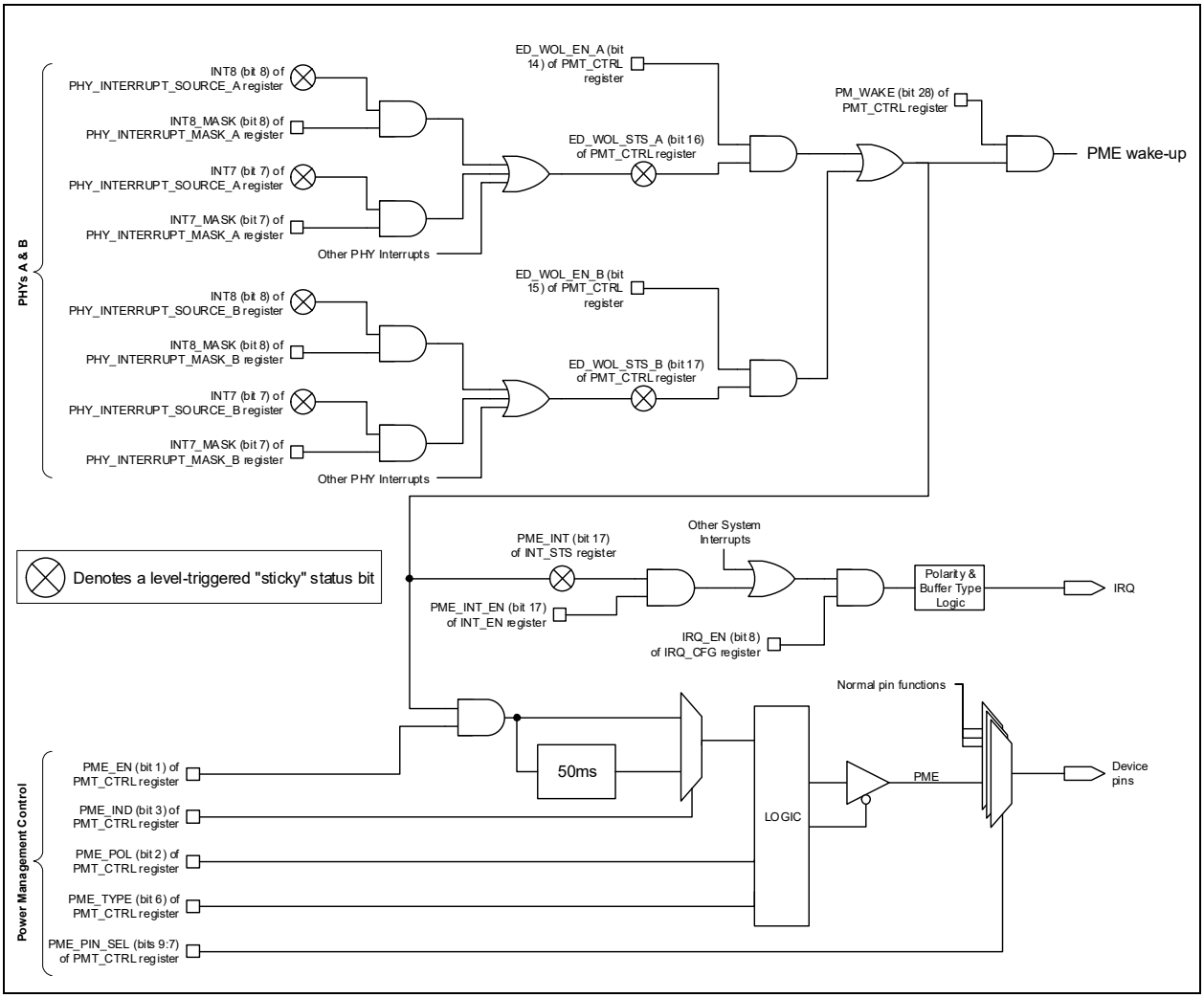
The **PME** output does not have a dedicated pin and is mapped onto several other pins as controlled by the PME Pin Map (PME_PIN_SEL) field in the Power Management Control Register (PMT_CTRL), replacing the normal pin function. Even when mapped, the PME output must be enabled, otherwise the mapped pin remains undriven. By default, PME is not enabled or mapped.

Note:	Before PME is mapped onto another pin, the default state of that pin is that of the pin's normal function. The pin may or may not be driven and an internal or external pull-up or pull-down may be present. It is the responsibility of the system designer to account for this, keeping in mind: ***ERRLED:*** This is enabled via the EEPROM contents and should be set to disabled. In the event of an EEPROM loading error, the **ERRLED** function is forced enabled and false PME events might occur. **ERRLED** is also used as the **100FD_B** configuration strap pin. If **ERRLED** is used as the PME output (with potential to be wire-ORed shared signal), careful consideration must be paid to strap value. Host software might need to correct the strap results via the AMDIX Disable PHY B and ANEG Disable PHY B bits in the Hardware Configuration Register (HW_CFG) as well as reconfiguring PHY B. Also note that **ERRLED** is in the **VDD33** I/O voltage domain. ***WAIT_ACK***: This is enabled via the EEPROM contents and should be set to disabled. ***SYNC0/LATCH0 / SYNC1/LATCH1***: This is configured via the EEPROM contents. If set as ***SYNC0/SYNC1***, the driven value could be incorrect until the PME Pin Map (PME_PIN_SEL) value is set.

In system configurations where the **PME** output pin is shared among multiple devices (wired ORed), the ED_WOL_STS_B and ED_WOL_STS_A bits within the PMT_CTRL register can be read to determine which device is driving the PME signal.

When the PM_WAKE bit of the Power Management Control Register (PMT_CTRL) is set, the PME event will automatically wake up the system in certain chip level power modes, as described in Section 6.3.4.2, "Exiting Low Power Modes," on page 55. This is done independent from the values of the PME_EN, PME_POL, PME_IND and PME_TYPE register bits.

FIGURE 6-1: PME PIN AND PME INTERRUPT SIGNAL GENERATION



6.3.3 BLOCK LEVEL POWER MANAGEMENT

The device supports software controlled clock disabling of various modules in order to reduce power consumption.

Note:	Disabling individual blocks does not automatically reset the block, it only places it into a static non-operational state in order to reduce the power consumption of the device. If a block reset is not performed before re-enabling the block, then care must be taken to ensure that the block is in a state where it can be disabled and then re-enabled.

6.3.3.1 Disabling The EtherCAT Core

The entire EtherCAT Core may be disabled by setting the ECAT_DIS bit in the Power Management Control Register (PMT_CTRL). As a safety precaution, in order for this bit to be set, it must be written as a 1 two consecutive times. A write of a 0 will reset the count.

6.3.3.2 PHY Power Down

A PHY may be placed into power-down as described in Section 10.2.9, "PHY Power-Down Modes," on page 212.

6.3.3.3 LED Pins Power Down

All LED outputs may be disabled by setting the LED_DIS bit in the Power Management Control Register (PMT_CTRL). Open-drain / open-source LEDs are un-driven. Push-pull LEDs are still driven but are set to their inactive state.

6.3.4 CHIP LEVEL POWER MANAGEMENT

The device supports power-down modes to allow applications to minimize power consumption.

Power is reduced by disabling the clocks as outlined in Table 6-2, "Power Management States". All configuration data is saved when in any power state. Register contents are not affected unless specifically indicated in the register description.

There is one normal operating power state, D0, and three power saving states: D1, D2 and D3. Although appropriate for various wake-up detection functions, the power states do not directly enable and are not enforced by these functions.

D0: Normal Mode - This is the normal mode of operation of this device. In this mode, all functionality is available. This mode is entered automatically on any chip-level reset (POR, **RST#** pin reset, EtherCAT system reset).

D1: System Clocks Disabled, XTAL, PLL and network clocks enabled - In this low power mode, all clocks derived from the PLL clock are disabled. The network clocks remain enabled if supplied by the PHYs or externally. The crystal oscillator and the PLL remain enabled. Exit from this mode may be done manually or automatically.

This mode could be used for PHY General Power Down mode, PHY WoL mode and PHY Energy Detect Power Down mode.

D2: System Clocks Disabled, PLL disable requested, XTAL enabled - In this low power mode, all clocks derived from the PLL clock are disabled. The PLL is allowed to be disabled (and will disable if both of the PHYs are in either Energy Detect or General Power Down). The network clocks remain enabled if supplied by the PHYs or externally. The crystal oscillator remains enabled. Exit from this mode may be done manually or automatically.

This mode is useful for PHY Energy Detect Power Down mode and PHY WoL mode. This mode could be used for PHY General Power Down mode.

D3: System Clocks Disabled, PLL disabled, XTAL disabled - In this low power mode, all clocks derived from the 100 MHz PLL clock are disabled. The PLL will be disabled. External network clocks are gated off. The crystal oscillator is disabled. Exit from this mode may be only be done manually.

This mode is useful for PHY General Power Down mode.

The Host must place the PHY into General Power Down mode by setting the Power Down (PHY_PWR_DWN) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) before setting this power state.

Note:	Disabling the crystal oscillator will disable the **CLK_25** output signal.

TABLE 6-2: POWER MANAGEMENT STATES

Clock Source	D0	D1	D2	D3
25 MHz Crystal Oscillator	ON	ON	ON	OFF
PLL	ON	ON	OFF(2)	OFF
system clocks (100 MHz, 50 MHz, 25 MHz and others)	ON	OFF	OFF	OFF
network clocks	available(1)	available(1)	available(1)	OFF(3)

Note 1: If supplied by the PHYs or externally
2: PLL is requested to be turned off and will disable if both of the PHYs are in either Energy Detect or General Power Down
3: PHY clocks are off, external clocks are gated off

6.3.4.1 Entering Low Power Modes

To enter any of the low power modes (D1 - D3) from normal mode (D0), follow these steps:

1. Write the PM_MODE and PM_WAKE fields in the Power Management Control Register (PMT_CTRL) to their desired values
2. Set the wake-up detection desired per Section 6.3.1, "Wake-Up Event Detection".
3. Set the appropriate wake-up notification per Section 6.3.2, "Wake-Up (PME) Notification".
4. Ensure that the device is in a state where it can safely be placed into a low power mode (all packets transmitted, receivers disabled, packets processed / flushed, etc.)
5. Set the PM_SLEEP_EN bit in the Power Management Control Register (PMT_CTRL).

Note: The PM_MODE field cannot be changed at the same time as the PM_SLEEP_EN bit is set and the PM_SLEEP_EN bit cannot be set at the same time that the PM_MODE field is changed.

Upon entering any low power mode, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) and the Power Management Control Register (PMT_CTRL) is forced low.

Note: Upon entry into any of the power saving states the host interfaces are not functional.

6.3.4.2 Exiting Low Power Modes

Exiting from a low power mode can be done manually or automatically.

An automatic wake-up will occur based on the events described in Section 6.3.2, "Wake-Up (PME) Notification". Automatic wake-up is enabled with the Power Management Wakeup (PM_WAKE) bit in the Power Management Control Register (PMT_CTRL).

A manual wake-up is initiated by the host when:

- an HBI write (**CS** and **WR** or **CS**, **RD_WR** and **ENB**) is performed to the device. Although all writes are ignored until the device has been woken and a read performed, the host should direct the write to the Byte Order Test Register (BYTE_TEST). Writes to any other addresses should not be attempted until the device is awake.
- an SPI/SQI cycle (**SCS#** low and **SCK** high) is performed to the device. Although all reads and writes are ignored until the device has been woken, the host should direct the use a read of the Byte Order Test Register (BYTE_TEST) to wake the device. Reads and writes to any other addresses should not be attempted until the device is awake.

To determine when the host interface is functional, the Byte Order Test Register (BYTE_TEST) should be polled. Once the correct pattern is read, the interface can be considered functional. At this point, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) or the Power Management Control Register (PMT_CTRL) can be polled to determine when the device is fully awake.

For both automatic and manual wake-up, the Device Ready (READY) bit will go high once the device is returned to power savings state D0 and the PLL has re-stabilized. The PM_MODE and PM_SLEEP_EN fields in the Power Management Control Register (PMT_CTRL) will also clear at this point.

Under normal conditions, the device will wake-up within 2 ms.

6.3.5 POWER MANAGEMENT REGISTERS

6.3.5.1 Power Management Control Register (PMT_CTRL)

Offset: 084h / 3084h Size: 32 bits

This read-write register controls the power management features and the PME pin of the device. The ready state of the device be determined via the Device Ready (READY) bit of this register.

Note: This register can be read while the device is in the reset or not ready / power savings states without leaving the host interface in an intermediate state. If the host interface is in a reset state, returned data may be invalid.

It is not necessary to read all four bytes of this register. DWORD access rules do not apply to this register.

Bits	Description	Type	Default
31:29	**Power Management Mode (PM_MODE)** This register field determines the chip level power management mode that will be entered when the Power Management Sleep Enable (PM_SLEEP_EN) bit is set. **000: D0** **001: D1** **010: D2** **011: D3** **100: Reserved** **101: Reserved** **110: Reserved** **111: Reserved** Writes to this field are ignored if Power Management Sleep Enable (PM_SLEEP_EN) is also being written with a 1. This field is cleared when the device wakes up.	R/W/SC	000b
28	**Power Management Sleep Enable (PM_SLEEP_EN)** Setting this bit enters the chip level power management mode specified with the Power Management Mode (PM_MODE) field. **0: Device is not in a low power sleep state** **1: Device is in a low power sleep state** This bit can not be written at the same time as the PM_MODE register field. The PM_MODE field must be set, and then this bit must be set for proper device operation. Writes to this bit with a value of 1 are ignored if Power Management Mode (PM_MODE) is being written with a new value. **Note:** Although not prevented by H/W, this bit should not be written with a value of 1 while Power Management Mode (PM_MODE) has a value of “D0”. This field is cleared when the device wakes up.	R/W/SC	0b

Bits	Description	Type	Default
27	**Power Management Wakeup (PM_WAKE)** When set, this bit enables automatic wake-up based on PME events. **0: Manual Wakeup only** **1: Auto Wakeup enabled**	R/W	0b
26	**LED Disable (LED_DIS)** This bit disables LED outputs. Open-drain / open-source LEDs are un-driven. Push-pull LEDs are still driven but are set to their inactive state. **0: LEDs are enabled** **1: LEDs are disabled**	R/W	0b
25:22	**RESERVED**	RO	-
21	**EtherCAT Core Clock Disable (ECAT_DIS)** This bit disables the clocks for the EtherCAT core. **0: Clocks are enabled** **1: Clocks are disabled** In order for this bit to be set, it must be written as a 1 two consecutive times. A write of a 0 will reset the count.	R/W	0b
20:18	**RESERVED**	RO	-
17	**Energy-Detect / WoL Status Port B (ED_WOL_STS_B)** This bit indicates an energy detect or WoL event occurred on the Port B PHY. In order to clear this bit, it is required that the event in the PHY be cleared as well. The event sources are described in Section 6.3, "Power Management," on page 51.	R/W1C	0b
16	**Energy-Detect / WoL Status Port A (ED_WOL_STS_A)** This bit indicates an energy detect or WoL event occurred on the Port A PHY. In order to clear this bit, it is required that the event in the PHY be cleared as well. The event sources are described in Section 6.3, "Power Management," on page 51.	R/W1C	0b
15	**Energy-Detect / WoL Enable Port B (ED_WOL_EN_B)** When set, the PME_INT bit in the Interrupt Status Register (INT_STS) will be asserted upon an energy-detect or WoL event from Port B.	R/W	0b
14	**Energy-Detect / WoL Enable Port A (ED_WOL_EN_A)** When set, the PME_INT bit in the Interrupt Status Register (INT_STS) will be asserted upon an energy-detect or WoL event from Port A.	R/W	0b
13:10	**RESERVED**	RO	-

Bits	Description	Type	Default
9:7	**PME Pin Map (PME_PIN_SEL)** This field is used to map the PME signal to one of the various device pins. **000: None** **001: ERRLED** **010: WAIT_ACK** **011: SYNC0/LATCH0** **100: SYNC1/LATCH1** **101: Reserved** **110: Reserved** **111: Reserved** **Note:** Even when mapped, the PME output must be enabled, otherwise the mapped pin remains undriven.	R/W NASR Note	000b
6	**PME Buffer Type (PME_TYPE)** When this bit is cleared, the PME output pin functions as an open-drain buffer for use in a wired-or configuration. When set, the PME output pin is a push-pull driver. **0: PME pin open-drain output** **1: PME pin push-pull driver** **Note:** When the PME output pin is configured as an open-drain output, the PME_POL field of this register is ignored and the output is always active low.	R/W NASR Note	0b
5:4	**RESERVED**	RO	-
3	**PME Indication (PME_IND)** The PME signal can be configured as a pulsed output or a static signal, which is asserted upon detection of a wake-up event. When set, the PME signal will pulse active for 50mS upon detection of a wake-up event. When cleared, the PME signal is driven continuously upon detection of a wake-up event. **0: PME driven continuously on detection of event** **1: PME 50mS pulse on detection of event** The PME signal can be deactivated by clearing the above status bit(s) or by clearing the appropriate enable(s).	R/W	0b
2	**PME Polarity (PME_POL)** This bit controls the polarity of the PME signal. When set, the PME output is an active high signal. When cleared, it is active low. **Note:** When PME is configured as an open-drain output, this field is ignored and the output is always active low. **0: PME active low** **1: PME active high**	R/W NASR Note	0b
1	**PME Enable (PME_EN)** When set, this bit enables the external PME signal pin. When cleared, the external PME signal is disabled. **Note:** This bit does not affect the PME_INT interrupt bit of the Interrupt Status Register (INT_STS). **0: PME pin disabled** **1: PME pin enabled**	R/W	0b

Bits	Description	Type	Default
0	**Device Ready (READY)** When set, this bit indicates that the device is ready to be accessed. Upon power-up, RST# reset, return from power savings states, EtherCAT chip level or module level reset, or digital reset, the host processor may interrogate this field as an indication that the device has stabilized and is fully active. This rising edge of this bit will assert the Device Ready (READY) bit in INT_STS and can cause an interrupt if enabled. **Note:** With the exception of the HW_CFG, PMT_CTRL, BYTE_TEST, and RESET_CTL registers, read access to any internal resources is forbidden while the READY bit is cleared. Writes to any address are invalid until this bit is set. **Note:** This bit is identical to bit 27 of the Hardware Configuration Register (HW_CFG).	RO	0b

Note: Register bits designated as NASR are not reset when the DIGITAL_RST bit in the Reset Control Register (RESET_CTL) is set.

6.4 Device Ready Operation

The device supports a Ready status register bit that indicates to the Host software when the device is fully ready for operation. This bit may be read via the Power Management Control Register (PMT_CTRL) or the Hardware Configuration Register (HW_CFG).

Following power-up reset, RST# reset, EtherCAT chip level reset or digital reset (see Section 6.2, "Resets"), the Device Ready (READY) bit indicates that the device has read, and is configured from, the contents of the EEPROM.

An EtherCAT reset via the Reset Control Register (RESET_CTL) will cause the EtherCAT core to reload from the EEPROM, temporarily causing the Device Ready (READY) to be low.

Entry into any power savings state (see Section 6.3.4, "Chip Level Power Management") other than D0 will cause Device Ready (READY) to be low. Upon wake-up, the Device Ready (READY) bit will go high once the device is returned to power savings state D0 and the PLL has re-stabilized.

7.0 SYSTEM INTERRUPTS

7.1 Functional Overview

This chapter describes the system interrupt structure of the device. The device provides a multi-tier programmable interrupt structure which is controlled by the System Interrupt Controller. The programmable system interrupts are generated internally by the various device sub-modules and can be configured to generate a single external host interrupt via the **IRQ** interrupt output pin. The programmable nature of the host interrupt provides the user with the ability to optimize performance dependent upon the application requirements. The **IRQ** interrupt buffer type, polarity and de-assertion interval are modifiable. The **IRQ** interrupt can be configured as an open-drain output to facilitate the sharing of interrupts with other devices. All internal interrupts are maskable and capable of triggering the **IRQ** interrupt.

7.2 Interrupt Sources

The device is capable of generating the following interrupt types:

- Ethernet PHY Interrupts
- Power Management Interrupts
- General Purpose Timer Interrupt (GPT)
- EtherCAT Interrupt
- Software Interrupt (General Purpose)
- Device Ready Interrupt
- Clock Output Test Mode

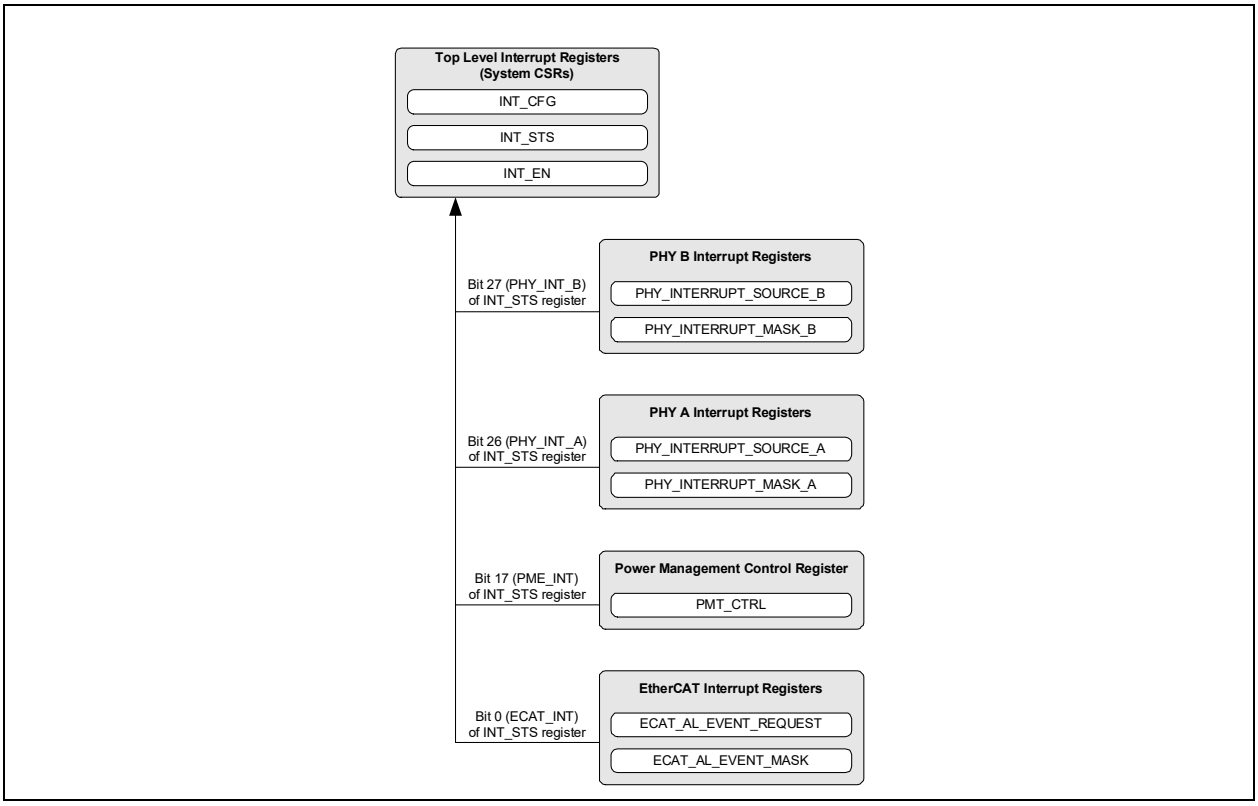
All interrupts are accessed and configured via registers arranged into a multi-tier, branch-like structure, as shown in Figure 7-1. At the top level of the device interrupt structure are the Interrupt Status Register (INT_STS), Interrupt Enable Register (INT_EN) and Interrupt Configuration Register (IRQ_CFG).

The Interrupt Status Register (INT_STS) and Interrupt Enable Register (INT_EN) aggregate and enable/disable all interrupts from the various device sub-modules, combining them together to create the **IRQ** interrupt. These registers provide direct interrupt access/configuration to the General Purpose Timer, software and device ready interrupts. These interrupts can be monitored, enabled/disabled and cleared, directly within these two registers. In addition, event indications are provided for the EtherCAT Slave, Power Management, and Ethernet PHY interrupts. These interrupts differ in that the interrupt sources are generated and cleared in other sub-block registers. The INT_STS register does not provide details on what specific event within the sub-module caused the interrupt and requires the software to poll an additional sub-module interrupt register (as shown in Figure 7-1) to determine the exact interrupt source and clear it. For interrupts which involve multiple registers, only after the interrupt has been serviced and cleared at its source will it be cleared in the INT_STS register.

The Interrupt Configuration Register (IRQ_CFG) is responsible for enabling/disabling the **IRQ** interrupt output pin as well as configuring its properties. The IRQ_CFG register allows the modification of the **IRQ** pin buffer type, polarity and de-assertion interval. The de-assertion timer guarantees a minimum interrupt de-assertion period for the **IRQ** output and is programmable via the Interrupt De-assertion Interval (INT_DEAS) field of the Interrupt Configuration Register (IRQ_CFG). A setting of all zeros disables the de-assertion timer. The de-assertion interval starts when the **IRQ** pin de-asserts, regardless of the reason.

Note:	The de-assertion timer does not apply to the PME interrupt. The PME interrupt is ORed into the IRQ logic following the deassertion timer gating. Assertion of the PME interrupt does not affect the de-assertion timer.

FIGURE 7-1: FUNCTIONAL INTERRUPT HIERARCHY



The following sections detail each category of interrupts and their related registers. Refer to the corresponding function's chapter for bit-level definitions of all interrupt registers.

7.2.1 ETHERNET PHY INTERRUPTS

The Ethernet PHYs each provide a set of identical interrupt sources. The top-level PHY A Interrupt Event (PHY_INT_A) and PHY B Interrupt Event (PHY_INT_B) bits of the Interrupt Status Register (INT_STS) provides indication that a PHY interrupt event occurred in the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x).

PHY interrupts are enabled/disabled via their respective PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x). The source of a PHY interrupt can be determined and cleared via the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x). Unique interrupts are generated based on the following events:

- ENERGYON Activated
- Auto-Negotiation Complete
- Remote Fault Detected
- Link Down (Link Status Negated)
- Link Up (Link Status Asserted)
- Auto-Negotiation LP Acknowledge
- Parallel Detection Fault
- Auto-Negotiation Page Received
- Wake-on-LAN Event Detected

In order for an interrupt event to trigger the external **IRQ** interrupt pin, the desired PHY interrupt event must be enabled in the corresponding PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x), the PHY A Interrupt Event Enable (PHY_INT_A_EN) and/or PHY B Interrupt Event Enable (PHY_INT_B_EN) bits of the Interrupt Enable Register (INT_EN) must be set and the IRQ output must be enabled via the IRQ Enable (IRQ_EN) bit of the Interrupt Configuration Register (IRQ_CFG).

For additional details on the Ethernet PHY interrupts, refer to Section 10.2.8, "PHY Interrupts," on page 209.

7.2.2 POWER MANAGEMENT INTERRUPTS

Multiple Power Management Event interrupt sources are provided by the device. The top-level Power Management Interrupt Event (PME_INT) bit of the Interrupt Status Register (INT_STS) provides indication that a Power Management interrupt event occurred in the Power Management Control Register (PMT_CTRL).

The Power Management Control Register (PMT_CTRL) provides enabling/disabling and status of all Power Management conditions. These include energy-detect on the PHYs and Wake-On-LAN (Perfect DA, Broadcast, Wake-up frame or Magic Packet) detection by PHYs A&B.

In order for a Power Management interrupt event to trigger the external **IRQ** interrupt pin, the desired Power Management interrupt event must be enabled in the Power Management Control Register (PMT_CTRL), the Power Management Event Interrupt Enable (PME_INT_EN) bit of the Interrupt Enable Register (INT_EN) must be set and the **IRQ** output must be enabled via the IRQ Enable (IRQ_EN) bit 8 of the Interrupt Configuration Register (IRQ_CFG).

The power management interrupts are only a portion of the power management features of the device. For additional details on power management, refer to Section 6.3, "Power Management," on page 51.

7.2.3 GENERAL PURPOSE TIMER INTERRUPT

A GP Timer (GPT_INT) interrupt is provided in the top-level Interrupt Status Register (INT_STS) and Interrupt Enable Register (INT_EN). This interrupt is issued when the General Purpose Timer Count Register (GPT_CNT) wraps past zero to FFFFh and is cleared when the GP Timer (GPT_INT) bit of the Interrupt Status Register (INT_STS) is written with 1.

In order for a General Purpose Timer interrupt event to trigger the external **IRQ** interrupt pin, the GPT must be enabled via the General Purpose Timer Enable (TIMER_EN) bit in the General Purpose Timer Configuration Register (GPT_CFG), the GP Timer Interrupt Enable (GPT_INT_EN) bit of the Interrupt Enable Register (INT_EN) must be set and the **IRQ** output must be enabled via the IRQ Enable (IRQ_EN) bit of the Interrupt Configuration Register (IRQ_CFG).

For additional details on the General Purpose Timer, refer to Section 14.1, "General Purpose Timer," on page 394.

7.2.4 ETHERCAT INTERRUPT

The top-level EtherCAT Interrupt Event (ECAT_INT) of the Interrupt Status Register (INT_STS) provides indication that an EtherCAT interrupt event occurred in the AL Event Request Register. The AL Event Mask Register provides enabling/disabling of all EtherCAT interrupt conditions. The AL Event Request Register provides the status of all EtherCAT interrupts.

In order for an EtherCAT interrupt event to trigger the external **IRQ** interrupt pin, the desired EtherCAT interrupt must be enabled in the AL Event Mask Register, the EtherCAT Interrupt Event Enable (ECAT_INT_EN) bit of the Interrupt Enable Register (INT_EN) must be set and the **IRQ** output must be enabled via the IRQ Enable (IRQ_EN) bit of the Interrupt Configuration Register (IRQ_CFG).

For additional details on the EtherCAT interrupts, refer to Section 11.0, "EtherCAT," on page 282.

7.2.5 SOFTWARE INTERRUPT

A general purpose software interrupt is provided in the top level Interrupt Status Register (INT_STS) and Interrupt Enable Register (INT_EN). The Software Interrupt (SW_INT) bit of the Interrupt Status Register (INT_STS) is generated when the Software Interrupt Enable (SW_INT_EN) bit of the Interrupt Enable Register (INT_EN) changes from cleared to set (i.e. on the rising edge of the enable). This interrupt provides an easy way for software to generate an interrupt and is designed for general software usage.

In order for a Software interrupt event to trigger the external **IRQ** interrupt pin, the **IRQ** output must be enabled via the IRQ Enable (IRQ_EN) bit of the Interrupt Configuration Register (IRQ_CFG).

7.2.6 DEVICE READY INTERRUPT

A device ready interrupt is provided in the top-level Interrupt Status Register (INT_STS) and Interrupt Enable Register (INT_EN). The Device Ready (READY) bit of the Interrupt Status Register (INT_STS) indicates that the device is ready to be accessed after a power-up or reset condition. Writing a 1 to this bit in the Interrupt Status Register (INT_STS) will clear it.

In order for a device ready interrupt event to trigger the external **IRQ** interrupt pin, the Device Ready Enable (READY_EN) bit of the Interrupt Enable Register (INT_EN) must be set and the **IRQ** output must be enabled via the IRQ Enable (IRQ_EN) bit of the Interrupt Configuration Register (IRQ_CFG).

7.2.7 CLOCK OUTPUT TEST MODE

In order to facilitate system level debug, the crystal clock can be enabled onto the **IRQ** pin by setting the IRQ Clock Select (IRQ_CLK_SELECT) bit of the Interrupt Configuration Register (IRQ_CFG).

The **IRQ** pin should be set to a push-pull driver by using the IRQ Buffer Type (IRQ_TYPE) bit for the best result.

7.3 Interrupt Registers

This section details the directly addressable interrupt related System CSRs. These registers control, configure and monitor the **IRQ** interrupt output pin and the various device interrupt sources. For an overview of the entire directly addressable register map, refer to Section 5.0, "Register Map," on page 40.

TABLE 7-1: INTERRUPT REGISTERS

ADDRESS	ETHERCAT DIRECT MAPPED MODE	REGISTER NAME (SYMBOL)
054h	3054h	Interrupt Configuration Register (IRQ_CFG)
058h	3058h	Interrupt Status Register (INT_STS)
05Ch	305Ch	Interrupt Enable Register (INT_EN)

7.3.1 INTERRUPT CONFIGURATION REGISTER (IRQ_CFG)

Offset: 054h / 3054h Size: 32 bits

This read/write register configures and indicates the state of the IRQ signal.

Bits	Description	Type	Default
31:24	**Interrupt De-assertion Interval (INT_DEAS)** This field determines the Interrupt Request De-assertion Interval in multiples of 10 microseconds. Setting this field to zero causes the device to disable the INT_DEAS Interval, reset the interval counter and issue any pending interrupts. If a new, non-zero value is written to this field, any subsequent interrupts will obey the new setting. This field does not apply to the PME_INT interrupt.	R/W	00h
23:15	**RESERVED**	RO	-
14	**Interrupt De-assertion Interval Clear (INT_DEAS_CLR)** Writing a 1 to this register clears the de-assertion counter in the Interrupt Controller, thus causing a new de-assertion interval to begin (regardless of whether or not the Interrupt Controller is currently in an active de-assertion interval). **0: Normal operation** **1: Clear de-assertion counter**	R/W SC	0h
13	**Interrupt De-assertion Status (INT_DEAS_STS)** When set, this bit indicates that the interrupt controller is currently in a de-assertion interval and potential interrupts will not be sent to the IRQ pin. When this bit is clear, the interrupt controller is not currently in a de-assertion interval and interrupts will be sent to the IRQ pin. **0: Interrupt controller not in de-assertion interval** **1: Interrupt controller in de-assertion interval**	RO	0b
12	**Master Interrupt (IRQ_INT)** This read-only bit indicates the state of the internal IRQ line, regardless of the setting of the IRQ_EN bit, or the state of the interrupt de-assertion function. When this bit is set, one of the enabled interrupts is currently active. **0: No enabled interrupts active** **1: One or more enabled interrupts active**	RO	0b
11:9	**RESERVED**	RO	-
8	**IRQ Enable (IRQ_EN)** This bit controls the final interrupt output to the IRQ pin. When clear, the IRQ output is disabled and permanently de-asserted. This bit has no effect on any internal interrupt status bits. **0: Disable output on IRQ pin** **1: Enable output on IRQ pin**	R/W	0b
7:5	**RESERVED**	RO	-

Bits	Description	Type	Default
4	**IRQ Polarity (IRQ_POL)** When cleared, this bit enables the IRQ line to function as an active low output. When set, the IRQ output is active high. When the IRQ is configured as an open-drain output (via the IRQ_TYPE bit), this bit is ignored and the interrupt is always active low. **0: IRQ active low output** **1: IRQ active high output**	R/W NASR Note 1	0b
3:2	**RESERVED**	RO	-
1	**IRQ Clock Select (IRQ_CLK_SELECT)** When this bit is set, the crystal clock may be output on the IRQ pin. This is intended to be used for system debug purposes in order to observe the clock and not for any functional purpose. **Note:** When using this bit, the IRQ pin should be set to a push-pull driver.	R/W	0b
0	**IRQ Buffer Type (IRQ_TYPE)** When this bit is cleared, the IRQ pin functions as an open-drain output for use in a wired-or interrupt configuration. When set, the IRQ is a push-pull driver. **Note:** When configured as an open-drain output, the IRQ_POL bit is ignored and the interrupt output is always active low. **0: IRQ pin open-drain output** **1: IRQ pin push-pull driver**	R/W NASR Note 1	0b

Note 1: Register bits designated as NASR are not reset when the DIGITAL_RST bit in the Reset Control Register (RESET_CTL) is set.

7.3.2 INTERRUPT STATUS REGISTER (INT_STS)

Offset: 058h / 3058h Size: 32 bits

This register contains the current status of the generated interrupts. A value of 1 indicates the corresponding interrupt conditions have been met, while a value of 0 indicates the interrupt conditions have not been met. The bits of this register reflect the status of the interrupt source regardless of whether the source has been enabled as an interrupt in the Interrupt Enable Register (INT_EN). Where indicated as R/W1C, writing a 1 to the corresponding bits acknowledges and clears the interrupt.

Bits	Description	Type	Default
31	**Software Interrupt (SW_INT)** This interrupt is generated when the Software Interrupt Enable (SW_INT_EN) bit of the Interrupt Enable Register (INT_EN) is set high. Writing a one clears this interrupt.	R/W1C	0b
30	**Device Ready (READY)** This interrupt indicates that the device is ready to be accessed after a power-up or reset condition.	R/W1C	0b
29:28	**RESERVED**	RO	-
27	**PHY B Interrupt Event (PHY_INT_B)** This bit indicates an interrupt event from the PHY B. The source of the interrupt can be determined by polling the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x).	RO	0b
26	**PHY A Interrupt Event (PHY_INT_A)** This bit indicates an interrupt event from the PHY A. The source of the interrupt can be determined by polling the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x).	RO	0b
25:20	**RESERVED**	RO	-
19	**GP Timer (GPT_INT)** This interrupt is issued when the General Purpose Timer Count Register (GPT_CNT) wraps past zero to FFFFh.	R/W1C	0b
18	**RESERVED**	RO	-
17	**Power Management Interrupt Event (PME_INT)** This interrupt is issued when a Power Management Event is detected as configured in the Power Management Control Register (PMT_CTRL). This interrupt functions independent of the PME signal and will still function if the PME signal is disabled. Writing a '1' clears this bit regardless of the state of the PME hardware signal. In order to clear this bit, all unmasked bits in the Power Management Control Register (PMT_CTRL) must first be cleared. **Note:** The Interrupt De-assertion interval does not apply to the PME interrupt.	R/W1C	0b
16:1	**RESERVED**	RO	-
0	**EtherCAT Interrupt Event (ECAT_INT)** This bit indicates an EtherCAT interrupt event. The source of the interrupt can be determined by polling the AL Event Request Register.	RO	0b

7.3.3 INTERRUPT ENABLE REGISTER (INT_EN)

Offset: 05Ch / 305Ch Size: 32 bits

This register contains the interrupt enables for the **IRQ** output pin. Writing 1 to any of the bits enables the corresponding interrupt as a source for **IRQ**. Bits in the Interrupt Status Register (INT_STS) register will still reflect the status of the interrupt source regardless of whether the source is enabled as an interrupt in this register (with the exception of Software Interrupt Enable (SW_INT_EN). For descriptions of each interrupt, refer to the Interrupt Status Register (INT_STS) bits, which mimic the layout of this register.

Bits	Description	Type	Default
31	**Software Interrupt Enable (SW_INT_EN)**	R/W	0b
30	**Device Ready Enable (READY_EN)**	R/W	0b
29:28	**RESERVED**	RO	-
27	**PHY B Interrupt Event Enable (PHY_INT_B_EN)**	R/W	0b
26	**PHY A Interrupt Event Enable (PHY_INT_A_EN)**	R/W	0b
25:20	**RESERVED**	RO	-
19	**GP Timer Interrupt Enable (GPT_INT_EN)**	R/W	0b
18	**RESERVED**	RO	-
17	**Power Management Event Interrupt Enable (PME_INT_EN)**	R/W	0b
16:1	**RESERVED**	RO	-
0	**EtherCAT Interrupt Event Enable (ECAT_INT_EN)**	R/W	0b

8.0 HOST BUS INTERFACE

8.1 Functional Overview

The Host Bus Interface (HBI) module provides a high-speed asynchronous slave interface that facilitates communication between the device and a host system. The HBI allows access to the System CSRs and internal FIFOs and memories and handles byte swapping based on the endianness select.

The following is an overview of the functions provided by the HBI:

- **Address bus input:** Three addressing modes are supported. These are a Multiplexed Address / Data Mode, a Demultiplexed Address Mode, and a Indexed Address Mode. The mode selection is done through a configuration input.
- **Selectable data bus width:** Selected through a configuration input, the host data bus width is selectable.

  LAN9252 compatibility Mode: 16 and 8-bit data modes are supported.
  The HBI performs BYTE and WORD to DWORD assembly on write data and keeps track of the BYTE / WORD count for reads. Individual BYTE access is not supported.

  EtherCAT Direct Mapped Mode: 16 and 8-bit data modes are supported. **BE1** and **BE0** are used for byte selection. For the non-EtherCAT Core registers, The HBI performs BYTE and WORD to DWORD assembly / disassembly. Individual BYTE access (in 16-bit mode) is not supported. For the EtherCAT Core registers and Process RAM, individual BYTE access is supported. BYTE to DWORD assembly / disassembly is not necessary.
- **Selectable read / write control modes:** Two control modes are available. Separate read and write pins or an enable and direction pin. The mode selection is done through a configuration input.
- **Selectable control line polarity / buffer type:** The polarity of the wait / acknowledge chip select, read / write, byte enables, and address latch signals, and the buffer type of the wait / acknowledge signal are selectable through configuration inputs.
- **Dynamic Endianness control:** The HBI supports the selection of big and little endian host byte ordering based on the endianness signal. This highly flexible interface provides mixed endian access for registers and memory. Depending on the addressing mode of the device, this signal is either configuration register controlled, pin controlled, or as part of the strobed address input.
- **Bus Wait State support:** Used with EtherCAT Direct Mapped mode, the HBI provides a Wait / Acknowledge signal to indicate when read data is available or write cycles may be completed.
- **Direct FIFO access:** A FIFO direct select signal directs all host write operations to the EtherCAT Process RAM Write Data FIFO (Multiplexed Address Mode only) and all host read operations from the EtherCAT Process RAM Read Data FIFO (Multiplexed Address Mode only). This signal is strobed as part of the address input.

8.1.1 ETHERCAT DIRECT MAPPED MODE PCB BACKWARDS COMPATIBILITY

While in EtherCAT Direct Mapped Mode, additional pins are required for full functionality. However with some caveats, EtherCAT Direct Mapped Mode can be made to work with PCBs designed for the LAN9252.

WAIT_ACK: The **WAIT_ACK** pin on the LAN9252 functioned as the fiber mode signal detect as well as the Port B FX-SD Enable. For copper twisted pair operation, this pin would either be tied to or pulled down to ground. **WAIT_ACK** is disabled by default to avoid a short. Without the **WAIT_ACK** connected to the host processor, the host bus cycles must assume worst case cycle timing.

BE1/BE0: When in 16-bit data width mode, the byte enables are required to select which byte or bytes are written or read from the EtherCAT core. In multiplexed mode, these pins on the LAN9252 are unused and most likely not driven. The **BE1/BE0** pins have internal pull-downs such that if left undriven, they would assume an active state, This allows 16-bit only access to the device. Caution is required to not over-write the adjacent byte when only BYTE access is desired.

8.2 Control Logic

8.2.1 READ / WRITE CONTROL SIGNALS

The device supports two distinct read / write signal methods:

- read (**RD**) and write (**WR**) strobes are input on separate pins.
- read and write signals are decoded from an enable input (**ENB**) and a direction input (**RD_WR**).

8.2.2 CONTROL LINE POLARITY AND BUFFER TYPE

The device supports polarity control on the following:

- chip select input (**CS**)
- read strobe (**RD**) / direction input (**RD_WR**)
- write strobe (**WR**) / enable input (**ENB**)
- byte enable (**BE0** and **BE1**)
- address latch control (**ALELO** and **ALEHI**)
- wait / acknowledge (**WAIT_ACK**)

The device supports buffer type control on the following:

- wait / acknowledge (**WAIT_ACK**)

8.2.3 CS QUALIFICATION OF ALE (MULTIPLEXED ADDRESS / DATA MODE ONLY)

Qualification of the **ALELO** and **ALEHI** signals with the **CS** signal is selectable. When qualification is enabled, **CS** must be active during **ALELO** and **ALEHI** in order to strobe the address inputs. When qualification is not enabled, **CS** is a don't care during the address phase.

8.2.4 WAIT / ACKNOWLEDGE OPERATION

When operating in EtherCAT Direct Mapped mode (refer to Section 5.0, Register Map), the host system must meet the device's timing access requirements. For reads from the EtherCAT Core CSR or Process Data RAM, the read cycles must wait until read data has been internally retrieved. All write cycles must wait for any prior write access to the EtherCAT Core CSR or Process Data RAM to internally complete. The host system may either wait the specified worst case access time, or may use the **WAIT_ACK** signal.

WAIT_ACK is only used when operating in EtherCAT Direct Mapped mode. When not in EtherCAT Direct Mapped mode, if **WAIT_ACK** is enabled it is always inactive (showing ACK).

WAIT_ACK may be set as an active low open drain output for wire-AND systems or as a three-stated push-pull output. **WAIT_ACK** operation is described in Section 8.4.3.3.

8.3 Device Addressing, Endianess Control and Data FIFO Selection

8.3.1 MULTIPLEXED ADDRESS / DATA MODE

In Multiplexed Address / Data mode, the address, FIFO Direct Select and endianness select inputs are shared with the data bus. Two methods are supported, Single Phase Address Latching, utilizing up to 16 address / data pins and Dual Phase Address Latching, utilizing only the lower 8 data bits.

8.3.1.1 Single Phase Address Latching

In Single Phase mode, all address bits, the FIFO Direct Select signal and the endianness select are strobed into the device using the trailing edge of the **ALELO** signal. The address latch is implemented on all 16 address / data pins. In 8-bit data mode, where pins **AD[15:8]** are used exclusively for addressing, it is not necessary to drive these upper address lines with a valid address continually through read and write operations. However, this operation, referred to as Partial Address Multiplexing, is acceptable since the device will never drive these pins.

Qualification of the **ALELO** signal with the **CS** signal is selectable. When qualification is enabled, **CS** must be active during **ALELO** in order to strobe the address inputs. When qualification is not enabled, **CS** is a don't care during the address phase.

The address is retained for all future read and write operations. It is retained until either a reset event occurs or a new address is loaded. This allows multiple read and write requests to take place to the same address, without requiring multiple address latching operations.

8.3.1.2 Dual Phase Address Latching

In Dual Phase mode, the lower 8 address bits are strobed into the device using the inactive going edge of the **ALELO** signal and the remaining upper address bits, the FIFO Direct Select signals and the endianness select are strobed into the device using the trailing edge of the **ALEHI** signal. The strobes can be in either order. In 8-bit data mode, pins **AD[15:8]** are not used. In 16-bit data mode, pins **D[15:8]** are used only for data.

Qualification of the **ALELO** and **ALEHI** signals with the CS signal is selectable. When qualification is enabled, **CS** must be active during **ALELO** and **ALEHI** in order to strobe the address inputs. When qualification is not enabled, **CS** is a don't care during the address phase.

The address is retained for all future read and write operations. It is retained until either a reset event occurs or a new address is loaded. This allows multiple read and write requests to take place to the same address, without requiring multiple address latching operations.

8.3.1.3 Address Bit to Address / Data Pin Mapping

In 8-bit data mode, address bit 0 is multiplexed onto pin **AD[0]**, address bit 1 onto pin **AD[1]**, etc.
In LAN9252 compatible address mode, the highest address bit is bit 9 and is multiplexed onto pin **AD[9]** (single phase) or **AD[1]** (dual phase). The address latched into the device is considered a BYTE address and covers 1K bytes (0 to 3FFh).
In EtherCAT Direct Mapped mode, the highest address bit is bit 13 and is multiplexed onto pin **AD[13]** (single phase) or **AD[5]** (dual phase). The address latched into the device is considered a BYTE address and covers 16K bytes (0 to 3FFFh).

In 16-bit data mode, address bit 1 is multiplexed onto pin **AD[0]**, address bit 2 onto pin **AD[1]**, etc.
In LAN9252 compatible address mode, the highest address bit is bit 9 and is multiplexed onto pin **AD[8]** (single phase) or **AD[0]** (dual phase). The address latched into the device is considered a WORD address and covers 512 words (0 to 1FFh).
In EtherCAT Direct Mapped mode, the highest address bit is bit 13 and is multiplexed onto pin **AD[12]** (single phase) or **AD[4]** (dual phase). The address latched into the device is considered a WORD address and covers 8K words (0 to 1FFFh).

EtherCAT Direct Mapped mode increases the address bit count by 4.

When the address is sent to the rest of the device, it is converted to a BYTE address.

8.3.1.4 Endianness Select to Address / Data Pin Mapping

The endianness select is included into the multiplexed address to allow the host system to dynamically select the endianness based on the memory address used. This allows for mixed endian access for registers and memory.

The endianness selection is multiplexed to the data pin one bit above the last address bit as shown in Table 8-1.

TABLE 8-1: ENDIANNESS SELECT TO ADDRESS / DATA PIN MAPPING

	Address Mode			
	LAN9252 compatible address mode		**EtherCAT Direct Mapped mode**	
Data Mode	**Single Phase**	**Dual Phase**	**Single Phase**	**Dual Phase**
8-bit	AD10	AD2	AD14	AD6
16-bit	AD9	AD1	AD13	AD5

8.3.1.5 FIFO Direct Select to Address / Data Pin Mapping

The FIFO Direct Select signal is included into the multiplexed address to allow the host system to address the EtherCAT Process RAM Data FIFOs as if they were a large flat address space.

FIFO Direct Select is not used in EtherCAT Direct Mapped mode since the Process RAM FIFOs are not used.

The FIFO Direct Select signal is multiplexed to the data pin two bits above the last address bit as shown in Table 8-2.

TABLE 8-2: FIFO DIRECT SELECT TO ADDRESS / DATA PIN MAPPING

	Address Mode			
	LAN9252 compatible address mode		EtherCAT Direct Mapped mode	
Data Mode	Single Phase	Dual Phase	Single Phase	Dual Phase
8-bit	AD11	AD3	N/A	N/A
16-bit	AD10	AD2	N/A	N/A

8.3.2 DEMULTIPLEXED ADDRESS MODE

In Demultiplexed Address mode, the address and endianness select inputs are directly provided by the host. A FIFO Direct Select signal is not provided by the host and is internally held inactive. The address mode is controlled by PDI Control Register.

8.3.2.1 Address Bit Usage

The address input to the device is always considered a BYTE address. In 8-bit data mode, all address bits are used.

In 16-bit data mode, address bit 0 is unused.

8.3.3 INDEXED ADDRESS MODE

In Indexed Address mode, access to the internal registers and memory of the device are indirectly mapped using Index and Data registers. The desired internal address is written into the device at a particular offset. The value written is then used as the internal address when the associate Data register address is accessed.

Three Index / Data register sets are provided allowing for multi-threaded operation without the concern of one thread corrupting the Index set by another thread.

The selection of the appropriate Index register which then selects the internal register is done using the host address inputs directly (asynchronously).

A FIFO Direct Select signal is not provided by the host however, it is emulated when the host accesses the Data register located at BYTE address 18h-1Bh. As discussed below in Section 8.3.4.2, "Index Register Bypass FIFO Access (Indexed Address Mode Only)", the EtherCAT Process RAM Data FIFOs are accessed when reading or writing the Data register located at BYTE address 18h-1Bh. Index Register Bypass FIFO Access is not used in EtherCAT Direct Mapped mode since the Process RAM FIFOs are not used.

An endianness signal is not provided by the host, however, endianness can be configured per Index / Data pair and for the Index Register Bypass FIFO Access method.

The host address register map is given below. In 8-bit data mode, the host address input (ADDR[4:0]) is a BYTE address. In 16-bit data mode, ADDR0 is not provided and the host address input (ADDR[4:1]) is a WORD address.

TABLE 8-3: HOST BUS INTERFACE INDEXED ADDRESS MODE REGISTER MAP

BYTE ADDRESS	SYMBOL	REGISTER NAME
00h-03h	HBI_IDX_0	Host Bus Interface Index Register 0
04h-07h	HBI_DATA_0	Host Bus Interface Data Register 0
08h-0Bh	HBI_IDX_1	Host Bus Interface Index Register 1
0Ch-0Fh	HBI_DATA_1	Host Bus Interface Data Register 1

TABLE 8-3: HOST BUS INTERFACE INDEXED ADDRESS MODE REGISTER MAP

BYTE ADDRESS	SYMBOL	REGISTER NAME
10h-13h	HBI_IDX_2	Host Bus Interface Index Register 2
14h-17h	HBI_DATA_2	Host Bus Interface Data Register 2
18h-1Bh	PROCESS_RAM_FIFO (not used for when in EtherCAT Direct Mapped mode)	Process RAM Write Data FIFO Process RAM Read Data FIFO
1Ch-1Fh	HBI_CFG	Host Bus Interface Configuration Register

8.3.3.1 Host Bus Interface Index Register

The Index registers are writable as WORDs or as BYTEs, depending upon the data mode. There is no concern about DWORD assembly rules when writing these registers. The Index registers are formatted as follows:

Bits	Description	Type	Default
31:17	**RESERVED**	RO	-
16	**Byte High Enable** When in 16-bit data mode with EtherCAT Direct Mapped mode enabled, this active low bit enables the upper byte of the WORD for accesses to the EtherCAT Core registers or Process Data RAM. 0 = upper byte enabled 1 = upper byte disabled This bit is used for reads as well as writes. This bit is unused for accesses to the non-EtherCAT Core registers since they are always DWORD wide (via the DWORD assembly / disassemble process).	R/W	0b
15:0	**Internal Address** The address used when the corresponding Data register is accessed. **Note:** The internal address provided by each Index register is always considered to be a BYTE address. For accesses to the non-EtherCAT Core registers, address 1:0 are unused since these registers are always DWORD aligned. When in 16-bit data mode with EtherCAT Direct Mapped mode enabled, address 0 acts as an active low Byte Low Enable, enabling the lower byte of the WORD for accesses to the EtherCAT Core registers or Process Data RAM. 0 = lower byte enabled 1 = lower byte disabled This bit is used for reads as well as writes.	R/W	1234h Note 2

Note 2: The default may be used to help determine the endianness of the register.

8.3.3.2 Host Bus Interface Data Registers Usage

The Host Bus Interface Data Registers provide a 4 byte window onto the internal registers.

Accesses to the non-EtherCAT Core registers are always DWORD aligned and sized.The lower two bits of the address field in the index registers are not used.

The HBI performs the DWORD assemble / disassemble with the internal access being a DWORD. Bits 1 and 0 (8-bit data mode only) of the host bus address are used to select one of the four BYTEs (8-bit data mode) or one of the two WORDs (16-bit data mode) of the data register. Software may choose to address these BYTEs (8-bit data mode) or WORDs (16-bit data mode) as BYTEs/WORDs at the various addresses (e.g. @HBI_DATA_0, @HBI_DATA_0+1, etc.) within the data register or it may choose to address these as a DWORD and let the host's Bus Interface Unit (BIU) split the larger DWORD data type into BYTEs or WORDs. In either case, the data register would be accessed multiple times at increasing addresses (e.g. @HBI_DATA_0, @HBI_DATA_0+1, etc.).

When using EtherCAT Direct Mapped mode, accesses to the EtherCAT Core registers or Process Data RAM do not require DWORD alignment. Bits 1 and 0 (8-bit data mode only) of the address field in the index registers are used as part of the internal pointer.

Internal data access is done at the native host bus width, however the selected EtherCAT Core register or Process Data RAM may be a DWORD, WORD or BYTE. Two methods exist to access the multiple WORDs/BYTEs when the register is wider than the host bus.

- Access the lower WORD/BYTE by addressing the lower WORD/BYTE of the data register (e.g. @HBI_DATA_0).
  Increment the address field in the index register.
  Access the next WORD/BYTE by once again addressing the **lower** WORD/BYTE of the data register (e.g. @HBI_DATA_0).
  (repeat twice more for 8-bit data mode)
- Access the lower WORD/BYTE by addressing the lower WORD/BYTE of the data register (e.g. @HBI_DATA_0).
  Access the next WORD/BYTE by addressing the **next** WORD/BYTE of the data register (e.g. @HBI_DATA_0+1 or +2).
  (repeat twice more for 8-bit data mode)

With the first method, software would need to reassemble or split the data. With the second method, although software does not need to increment the address field between data accesses, it would still need to reassemble or split the data. The second method, however, allows S/W to address the data register as a DWORD or WORD and let the host's Bus Interface Unit (BIU) split larger data types (DWORDs or WORDs) into WORDs and BYTEs.

In order to access sequential locations of the EtherCAT Core registers or Process Data RAM when sequential locations of the data register are addressed, hardware adds the lower 2 bits of the host address to the address field of the index register, emulating the incrementing of the address field.

8.3.3.3 Host Bus Interface Configuration Register

The HBI Configuration register is used to specify the endianness of the interface. Endianess for each Index / Data pair and for the Index Register Bypass FIFO Access method can be individually specified.

The endianness of this register is irrelevant since each byte is shadowed into 4 positions.

The HBI Configuration register is writable as a DWORD, as WORDs or as BYTEs, depending upon the data mode. There is no concern about DWORD assembly rules when writing this register.The Configuration register is formatted as follows:

Bits	Description	Type	Default
31:28	**RESERVED**	RO	-
27	**FIFO Endianness Shadow 3** This bit is a shadow of bit 3.	R/W	0b
26	**Host Bus Interface Index / Data Register 2 Endianness Shadow 3** This bit is a shadow of bit 2.	R/W	0b
25	**Host Bus Interface Index / Data Register 1 Endianness Shadow 3** This bit is a shadow of bit 1.	R/W	0b
24	**Host Bus Interface Index / Data Register 0 Endianness Shadow 3** This bit is a shadow of bit 0.	R/W	0b
23:20	**RESERVED**	RO	-
19	**FIFO Endianness Shadow 2** This bit is a shadow of bit 3.	R/W	0b

Bits	Description	Type	Default
18	**Host Bus Interface Index / Data Register 2 Endianness Shadow 2** This bit is a shadow of bit 2.	R/W	0b
17	**Host Bus Interface Index / Data Register 1 Endianness Shadow 2** This bit is a shadow of bit 1.	R/W	0b
16	**Host Bus Interface Index / Data Register 0 Endianness Shadow 2** This bit is a shadow of bit 0.	R/W	0b
15:12	**RESERVED**	RO	-
11	**FIFO Endianness Shadow 1** This bit is a shadow of bit 3.	R/W	0b
10	**Host Bus Interface Index / Data Register 2 Endianness Shadow 1** This bit is a shadow of bit 2.	R/W	0b
9	**Host Bus Interface Index / Data Register 1 Endianness Shadow 1** This bit is a shadow of bit 1.	R/W	0b
8	**Host Bus Interface Index / Data Register 0 Endianness Shadow 1** This bit is a shadow of bit 0.	R/W	0b
7:4	**RESERVED**	RO	-
3	**FIFO Endianness** This bit specifies the endianness of FIFO accesses when they are accessed by means other than the Index / Data Register method. **0 = Little Endian** **1 = Big Endian** **Note:** In order to avoid any ambiguity with the endianness of this register, bits 3, 11, 19 and 27 are shadowed. If any of these bits are set during a write, all of the bits will be set.	R/W	0b
2	**Host Bus Interface Index / Data Register 2 Endianness** This bit specifies the endianness of the Index and Data register set 2. **0 = Little Endian** **1 = Big Endian** **Note:** In order to avoid any ambiguity with the endianness of this register, bits 2, 10, 18 and 26 are shadowed. If any of these bits are set during a write, all of the bits will be set.	R/W	0b
1	**Host Bus Interface Index / Data Register 1 Endianness** This bit specifies the endianness of the Index and Data register set 1. **0 = Little Endian** **1 = Big Endian** **Note:** In order to avoid any ambiguity with the endianness of this register, bits 1, 9, 17 and 25 are shadowed. If any of these bits are set during a write, all of the bits will be set.	R/W	0b

Bits	Description	Type	Default
0	**Host Bus Interface Index / Data Register 0 Endianness** This bit specifies the endianness of the Index and Data register set 0. **0 = Little Endian** **1 = Big Endian** **Note:** In order to avoid any ambiguity with the endianness of this register, bits 0, 8, 16 and 24 are shadowed. If any of these bits are set during a write, all of the bits will be set.	R/W	0b

8.3.4 ETHERCAT PROCESS RAM DATA FIFO ACCESS

8.3.4.1 FIFO Direct Select Access (Multiplexed Address / Data Mode Only)

As mentioned in Section 8.3.1.5, a FIFO Direct Select signal is provided allows the host system to address the EtherCAT Process RAM Data FIFOs as if they were a large flat address space. When the FIFO Direct Select signal, which was latched during the address latch cycle, is active all host write operations are to the EtherCAT Process RAM Write Data FIFO and all host read operations are from the EtherCAT Process RAM Read Data FIFO. Only the lower latched address signals are decoded in order to select the proper BYTE or WORD. All other address inputs are ignored in this mode. All other operations are the same (DWORD assembly, FIFO popping, etc.).

8.3.4.2 Index Register Bypass FIFO Access (Indexed Address Mode Only)

In addition to the indexed access, the Index Registers can be bypassed and the FIFOs accessed at address 18h-1Bh. At this address, host write operations are to the EtherCAT Process RAM Write Data FIFO and host read operations are from the EtherCAT Process RAM Read Data FIFO. There is no associated Index Register.

8.4 Data Cycles

Data cycles consist of read and write cycles to the internal device registers, and for Indexed Address mode, read and write cycles to the Index and Configuration registers.

Data cycles are similar for all address modes. Differences are noted where appropriate.

8.4.1 INTERNAL REGISTER DATA ACCESS

Note:	Internal registers do not include the EtherCAT Core registers and Process Data RAM which are directly accessible from the host while in EtherCAT Direct Mapped mode. Access to the EtherCAT Core is described in Section 8.4.3.

The host data bus can be 16 or 8-bits wide while all internal registers are 32 bits wide. The Host Bus Interface performs the conversion from WORDs or BYTEs to DWORD, while in 8 or 16-bit data mode.Two or four contiguous accesses within the same DWORD are required in order to perform a write or read.

For Indexed Address mode, each Data register (four total including one for the Index Register Bypass FIFO Access), has a separate WORD or BYTE to DWORD conversion. Accesses may be mixed among these (and the HBI Index and Configuration registers) without concern of data corruption.

8.4.1.1 Write Cycles

A write cycle occurs when **CS** and **WR** are active (or when **CS** and **ENB** are active with **RD_WR** indicating write).

On the trailing edge of the write cycle (either **WR** or **CS** or **ENB** going inactive), the host data is captured into registers (one of four sets for Indexed Address mode) in the HBI. Depending on the bus width, either a WORD or a BYTE is captured. For 8 or 16-bit data modes, this functions as the DWORD assembly with the affected WORD or BYTE determined by the lower address inputs (the live host address inputs in the case of Indexed Address mode). BYTE swapping is also done at this point based on the endianness. Endianness determination is described in Section 8.4.4.

WRITES FOLLOWING INITIALIZATION

Following device initialization, writes from the Host Bus are ignored until after a read cycle is performed.

WRITES DURING AND FOLLOWING POWER MANAGEMENT

During and following any power management mode other than D0, writes from the Host Bus are ignored until after a read cycle is performed.

8 AND 16-BIT ACCESS

While in 8 or 16-bit data mode, the host is required to perform two or four, 16 or 8-bit writes to complete a single DWORD transfer. No ordering requirements exist. The host can access either the low or high WORD or BYTE first, as long as the other write(s) is(are) performed to the remaining WORD or BYTEs.

Note:	Writing the same WORD or BYTEs into the same DWORD may cause undefined or undesirable operation. The HBI hardware does not protect against this operation. For Indexed Address mode, accessing the same internal register using two Index / Data register pairs may cause undefined or undesirable operation. The HBI hardware does not protect against this operation. For Indexed Address mode, mixing reads and writes into the same Data register may cause undefined or undesirable operation. The HBI hardware does not protect against this operation.

A write BYTE / WORD counter (a separate counter for each of the four Data Registers in Indexed Address mode) keeps track of the number of writes. At the trailing edge of the write cycle, the counter (one of four for Indexed Address mode based on the captured host address) is incremented. Once all writes occur, a 32-bit write is performed to the internal register using the address captured on the leading edge of the write cycle and the data captured on the trailing edge of the write cycle. (For Indexed Address mode one of the four Data Registers sets is selected based on the captured host address.)

The write BYTE / WORD counter(s) is(are) reset if the power management mode is set to anything other than D0.

8.4.1.2 Read Cycles

A read cycle occurs when **CS** and **RD** are active (or when **CS** and **ENB** are active with **RD_WR** indicating read).

At the beginning of the read cycle, the appropriate device register is selected and its data is driven onto the data pins.

Depending on the bus width, either a WORD or a BYTE is read. For 8 or 16-bit data modes, the returned BYTE or WORD is determined by the endianness and the lower address inputs (the live host address inputs in the case of Indexed Address mode). Endianness determination is described in Section 8.4.4.

POLLING FOR INITIALIZATION COMPLETE

Before device initialization, the HBI will not return valid data. To determine when the HBI is functional the following procedure should be followed:

For Multiplexed Address / Data mode, the Byte Order Test Register (BYTE_TEST) should be polled. Each poll should consist of an address latch cycle(s) and a data cycle. Once the correct pattern is read, the interface can be considered functional. At this point, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) can be polled to determine when the device is fully configured.

For Demultiplexed Address mode, the Byte Order Test Register (BYTE_TEST) should be polled. Once the correct pattern is read, the interface can be considered functional. At this point, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) can be polled to determine when the device is fully configured.

For Indexed Address mode, first the Host Bus Interface Index Register 0 should be polled, then the Byte Order Test Register (BYTE_TEST) should be polled. Once the correct pattern is read, the interface can be considered functional. At this point, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) can be polled to determine when the device is fully configured.

READS DURING AND FOLLOWING POWER MANAGEMENT

During any power management mode other than D0, reads from the Host Bus are ignored. If the power management mode changes back to D0 during an active read cycle, the tail end of the read cycle is ignored. Internal registers are not affected and the state of the HBI does not change.

8 AND 16-BIT ACCESS

For certain register accesses, the host is required to perform two or four consecutive 16 or 8-bit reads to complete a single DWORD transfer. No ordering requirements exist. The host can access either the low or high WORD or BYTE first, as long as the other read(s) is(are) performed from the remaining WORD or BYTEs.

Note:	Reading the same WORD or BYTEs from the same DWORD may cause undefined or undesirable operation. The HBI hardware does not protect against this operation. The HBI simply counts that four BYTEs have been read. For Indexed Address mode, accessing the same internal register using two Index / Data register pairs may cause undefined or undesirable operation. The HBI hardware does not protect against this operation. For Indexed Address mode, mixing reads and writes into the same Data register may cause undefined or undesirable operation. The HBI hardware does not protect against this operation.

A read BYTE / WORD counter (a separate counter for each of the four Data Registers in Indexed Address mode) keeps track of the number of reads. This(these) counter(s) is(are) separate from the write counter(s) above. At the trailing edge of the read cycle, the counter (one of four for Indexed Address mode based on the captured host address) is incremented. On the last read for the DWORD, an internal read is performed to update any Change on Read CSRs or FIFOs.

The read BYTE / WORD counter(s) is(are) reset if the power management mode is set to anything other than D0.

SPECIAL CSR HANDLING

Live Bits

Any register bit that is updated by a hardware event is held at the beginning of the read cycle to prevent it from changing during the read cycle.

Multiple BYTE / WORD Live Registers in 16 or 8-Bit Modes

Some registers have “live” fields or related fields that span across multiple BYTEs or WORDs. For 16 and 8-bit data reads, it is possible for the value of these fields to change between host read cycles. In order to prevent reading intermediate values, these registers are locked when the first byte or word is read and unlocked when the last byte or word is read.

Registers that have this function, have a note in their register description. The registers are unlocked if the power management mode is set to anything other than D0.

Register Polling During Reset or Initialization

Some registers support polling during reset or device initialization to determine when the device is accessible. For these registers, only one read may be performed without the need to read the other WORD or BYTEs. The same BYTE or WORD of the register may be re-read repeatedly.

A register that is 16 or 8-bit readable or readable during reset or device initialization, is noted in its register description.

8.4.2 INDEXED ADDRESS MODE INDEX AND CONFIGURATION REGISTER DATA ACCESS

As described in Section 8.3.3, Indexed Address mode, contains Index and Configuration registers.

The host data bus can be 16 or 8-bits wide. The HBI Index registers and the HBI Configuration register are 32-bits wide and are writable as WORDs or as BYTEs, depending upon the data mode. They do not have nor do they require WORDs or BYTEs to DWORD conversion.

8.4.2.1 Write Cycles

A write cycle occurs when **CS** and **WR** are active (or when **CS** and **ENB** are active with **RD_WR** indicating write).

On the trailing edge of the write cycle (either **WR** or **CS** or **ENB** going inactive), the host data is captured into the Configuration register or one for the Index registers.

Depending on the bus width, either a WORD or a BYTE is written. The affected WORD or BYTE is determined by the endianness of the register (specified in the Host Bus Interface Configuration Register) and the lower address inputs. Individual BYTE (in 16-bit data mode) access is not supported.

WRITES FOLLOWING INITIALIZATION

Following device initialization, writes from the Host Bus are ignored until after a read cycle is performed.

WRITES DURING AND FOLLOWING POWER MANAGEMENT

During and following any power management mode other than D0, writes from the Host Bus are ignored until after a read cycle is performed.

8.4.2.2 Read Cycles

A read cycle occurs when **CS** and **RD** are active (or when **CS** and **ENB** are active with **RD_WR** indicating read). The host address is used directly from the Host Bus.

At the beginning of the read cycle, the appropriate register is selected and its data is driven onto the data pins. Depending on the bus width, either a WORD or a BYTE is read. For 8 or 16-bit data modes, the returned BYTE or WORD is determined by the endianness of the register (specified in the Host Bus Interface Configuration Register) and the lower host address inputs.

8.4.3 ETHERCAT DIRECT MAPPED MODE

Note:	While in EtherCAT Direct Mapped mode, non-EtherCAT Core registers continue to be accessed using the methods described in Section 8.4.1.1 and Section 8.4.2.2. While in EtherCAT Direct Mapped mode, Indexed Address Mode Index and Configuration registers continue to be accessed using the methods described in Section 8.4.2.1 and Section 8.4.2.2.

While in EtherCAT Direct Mapped mode, the EtherCAT Core registers and Process Data RAM are mapped into the host address space. The host data bus can be 16 or 8-bits wide and the EtherCAT Core interface is set to match. The Host Bus Interface does not perform any BYTE to WORD conversion. While in 16-bit mode, BYTE or WORD access may be performed.

8.4.3.1 EtherCAT Core Register and Process Data RAM Write Access

A write cycle starts when **CS** and **WR** are active (or when **CS** and **ENB** are active with **RD_WR** indicating write). The host address and byte high enable must be valid at the start of the write cycle.

Depending on the configuration, the internal write access is either performed during (non-posted) or following (posted) the host write cycle.

If non-posted write is selected, the internal write operation begins at the start of the write cycle with the **WAIT_ACK** pin indicating wait. When the internal write operation is finished, the **WAIT_ACK** pin indicates acknowledge allowing the host write cycle to complete. Note that non-posted writes require that the host data be valid at the start of the host write cycle.

If posted write is selected, the internal operation begins at the end of the host write cycle. In this case, the external write access is very fast but any access that follows immediately will be delayed by the pending write (as controlled by the **WAIT_ACK** pin). The maximum device access time is higher in this case.

Write cycles may be WORD or BYTE wide. WORD assembly is not performed by the HBI for accesses to the EtherCAT Core. All host accesses are sent to the EtherCAT Core interface.

Note:	In Indexed 16-bit mode, although the host bus may have byte enables available, they are not used by the device. The byte enables are provided internally as bits in the index register. The host may actually perform a WORD access however the byte enable bits specify the actually bytes affected.

WRITES FOLLOWING INITIALIZATION

Following device initialization, writes from the Host Bus are ignored until after a read cycle is performed.

WRITES DURING AND FOLLOWING POWER MANAGMENT

During and following any power management mode other than D0, writes from the Host Bus are ignored until after a read cycle is performed.

8.4.3.2 EtherCAT Core Register and Process Data RAM Read Access

A read cycle occurs when **CS** and **RD** are active (or when **CS** and **ENB** are active with **RD_WR** indicating read). The host address and byte high enable must be valid at the start of the read cycle.

The internal read operation begins at the start of the read cycle with the **WAIT_ACK** pin indicating wait. Following any potential posted internal write, the read operation is performed and valid data is available from the EtherCAT Core interface. Read data is saved in the EtherCAT Core interface and returned to the HBI where it is multiplexed based on the asynchronous host address.

The **WAIT_ACK** pin changes to indicate acknowledge allowing the host read cycle to complete. Configured within the EtherCAT Core interface, **WAIT_ACK** deassertion for read accesses can be additionally delayed for 15 ns to provided additional external DATA setup requirements with respect to **WAIT_ACK**.

Read cycles may be WORD or BYTE wide. WORD disassembly is not performed by the HBI for accesses to the EtherCAT Core. All host accesses are sent to the EtherCAT Core interface.

8.4.3.3 Wait / Acknowledge Operation

When operating in EtherCAT Direct Mapped mode, the host system must meet the device's timing access requirements. All read and write cycles must wait for any posted write access to internally complete. All reads and non-posted writes must wait until the internal data cycle completes.

The host system may either wait the specified worst case access time, or may use the **WAIT_ACK** signal.

For consistency, reads and writes to the non-EtherCAT Core registers as well as to the Indexed Address Mode Index and Configuration registers follow the same **WAIT_ACK** operation.

Read and write data cycles start with the leading edge of **CS**, which enables the **WAIT_ACK** output. The host may start the read or write cycle at any point following or along with **CS**.

If a posted EtherCAT Core write operation is internally pending, the **WAIT_ACK** will initially indicate wait. In the absence of a pending posted write, **WAIT_ACK** will initially indicate acknowledge (not busy).

For posted writes, if a prior posted write operation was not internally pending or once a prior posted write operation has completed, **WAIT_ACK** will indicate acknowledge. Once acknowledge is indicated, the host may complete the write cycle on its bus. The write cycle is internally captured and executed by the EtherCAT Core interface.

For non-posted writes and for reads, **WAIT_ACK** indicate wait when the host starts the data portion of the cycle. Once the write or read cycle is completed internally, **WAIT_ACK** will indicate acknowledge and the host may complete its cycle.

WAIT_ACK becomes undriven with the negation of **CS**.

8.4.4 HOST ENDIANNESS

The device supports big and little endian host byte ordering. the endianness selection is provided differently for the different addressing modes. For Multiplexed Address / Data mode, the endianness selection is latched during the address latch cycle. For Demultiplexed Address mode, the endianness selection is an input pin along with the address. For Indexed Address mode, the endianness selection is from the four endianness bits (one for each Index / Data pair and one for Index Register Bypass FIFO Access method) in the Host Bus Interface Configuration Register.

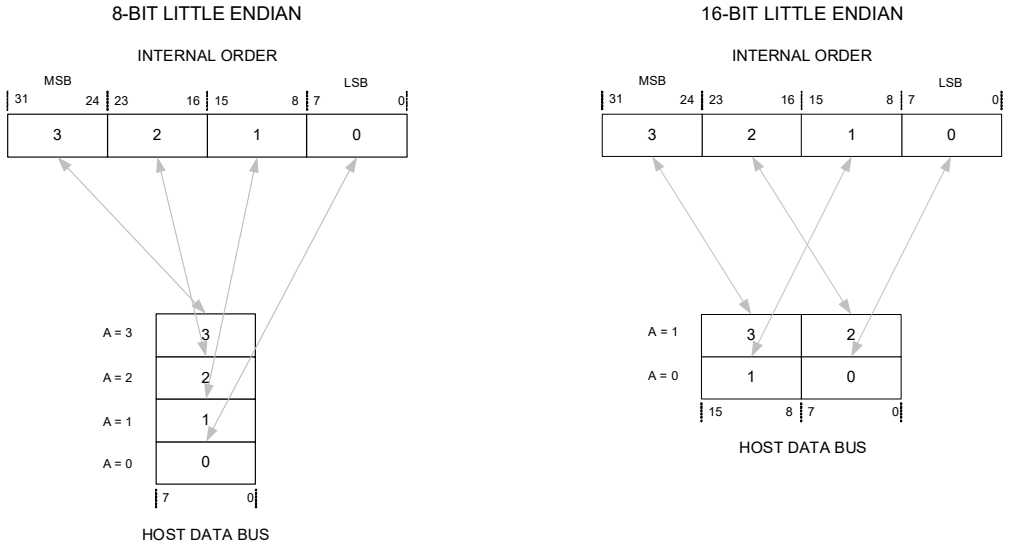
When the endianness select is low, host access is little endian and when high, host access is big endian. For multiplexed and demultiplexed modes, the endianness select may be connected to a high-order address line, making endian selection address-based. This highly flexible interface provides mixed endian access for registers and memory for both PIO and host DMA access.

Most internal buses are 32-bit with little endian byte ordering used internally. Logic within the Host Bus Interface reorders bytes based on the appropriate endianness bit, and the state of the least significant host address bits.

Data path operations for the supported endian configurations and data bus sizes are illustrated in Figure 8-1 and Figure 8-2.

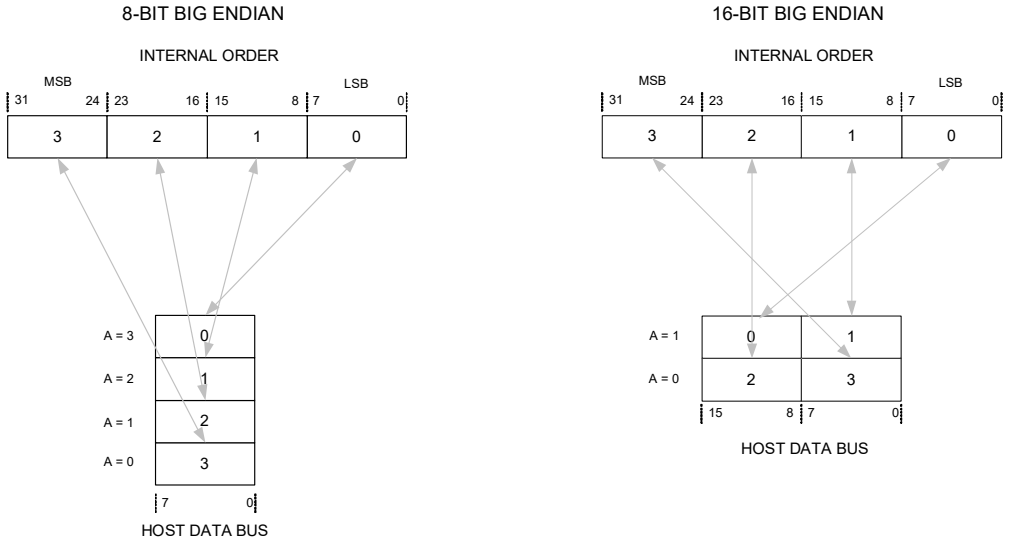
.

FIGURE 8-1: LITTLE ENDIAN BYTE ORDERING



.

FIGURE 8-2: BIG ENDIAN BYTE ORDERING



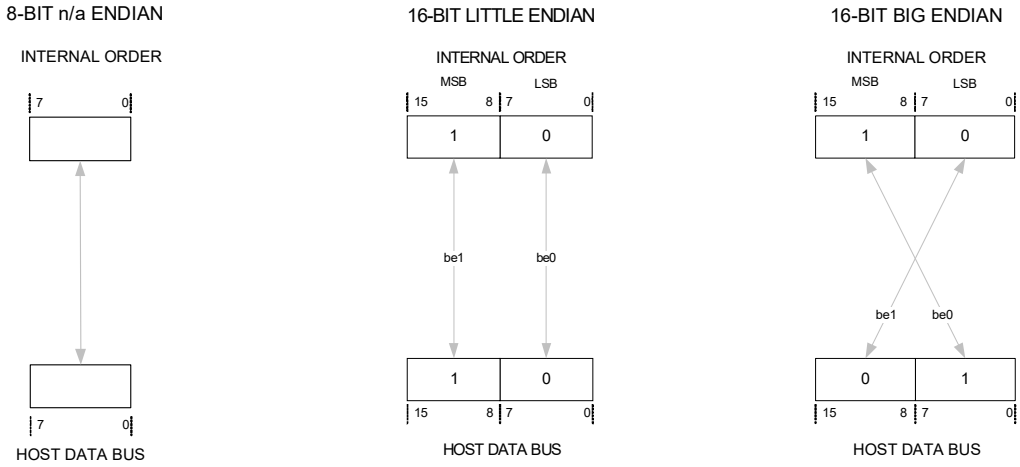
8.4.4.1 EtherCAT Direct Mapped Mode

While in EtherCAT Direct Mapped mode, the EtherCAT Core interface is 8-bit or 16-bit wide matching the host bus width selection.

Endianess in 8-bit mode is not applicable (the upper 8 bits of the internal data bus are unused). When 16-bits in width, little endian byte ordering is used internally. The Host Bus Interface swaps the low and high order bytes if the host access is big endian.

Note:	The byte enables **BE1/BE0** always follow the host bus pins, therefore in big endian mode, the byte enables are also swapped.

FIGURE 8-3: BYTE ORDERING - ETHERCAT DIRECT MAPPED MODE, ETHERCAT CORE



8.5 Functional Timing Diagrams - Legacy Mode

The following sections present functional timing diagrams while in LAN9252 compatible Legacy mode (i.e. not in EtherCAT Direct Mapped mode). Byte Enables **BE1/BE0** are not used in Legacy mode and are not shown. If enabled for output, **WAIT_ACK** is held inactive (ACK) and is shown as such.

8.5.1 MULTIPLEXED ADDRESSING MODE FUNCTIONAL TIMING DIAGRAMS

The following timing diagrams illustrate example multiplexed addressing mode read and write cycles for various address/data configurations and bus sizes. These diagrams do not cover every supported host bus permutation, but are selected to detail the main configuration differences (bus size, dual/single phase address latching) within the multiplexed addressing mode of operation.

The following should be noted for the timing diagrams in this section:

- The diagrams in this section depict active-low push-pull **WAIT_ACK** and active-high **ALEHI/ALELO**, **CS**, **RD**, and **WR** signals. The polarities of these signals are selectable via the HBI WAIT_ACK Polarity, HBI ALE Polarity, HBI Chip Select Polarity, HBI Read, Read/Write Polarity, and HBI Write, Enable Polarity bits of the PDI Configuration Register (HBI Modes), respectively. Refer to Section 8.2.2, "Control Line Polarity and Buffer Type," on page 69 for additional details.
- The diagrams in this section depict little endian byte ordering. However, dynamic big and little endianess are supported via the endianess signal. Endianess changes only the order of the bytes involved, and not the overall timing requirements. Refer to Section 8.3.1.4, "Endianness Select to Address / Data Pin Mapping," on page 70 for additional information.
- The diagrams in Section 8.5.1.1, "Dual Phase Address Latching" and Section 8.5.1.2, "Single Phase Address Latching" utilize **RD** and **WR** signals. Alternative **RD_WR** and **ENB** signaling is also supported, as shown in Section 8.5.1.3, "RD_WR / ENB Control Mode Examples". The HBI read/write mode is selectable via the HBI Read/Write Mode bit of the PDI Configuration Register (HBI Modes). The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI Read, Read/Write Polarity and HBI Write, Enable Polarity bits of the PDI Configuration Register (HBI Modes).
- Qualification of the **ALELO** and/or **ALEHI** with the **CS** signal is selectable via the HBI ALE Qualification bit of the PDI Configuration Register (HBI Modes). Refer to Section 8.3.1.1, "Single Phase Address Latching," on page 69 and Section 8.3.1.2, "Dual Phase Address Latching," on page 69 for additional information.
- In dual phase address latching mode, the **ALEHI** and **ALELO** cycles can be in any order. Either or both **ALELO** and **ALEHI** cycles may be skipped and the device retains the last latched address.
- In single phase address latching mode, the **ALELO** cycle may be skipped and the device retains the last latched address.

> **Note:** In 8 and 16-bit modes, the **ALELO** cycle is normally not skipped since sequential BYTEs or WORDs are accessed in order to satisfy a complete DWORD cycle. However, there are registers for which a single BYTE or WORD access is allowed, in which case multiple accesses to these registers may be performed without the need to re-latch the repeated address.

- For 16 and 8-bit modes, consecutive address cycles must be within the same DWORD until the DWORD is completely accessed (with the register exceptions noted above). Although BYTEs and WORDs can be accessed in any order, the diagrams in this section depict accessing the lower address BYTE or WORD first.

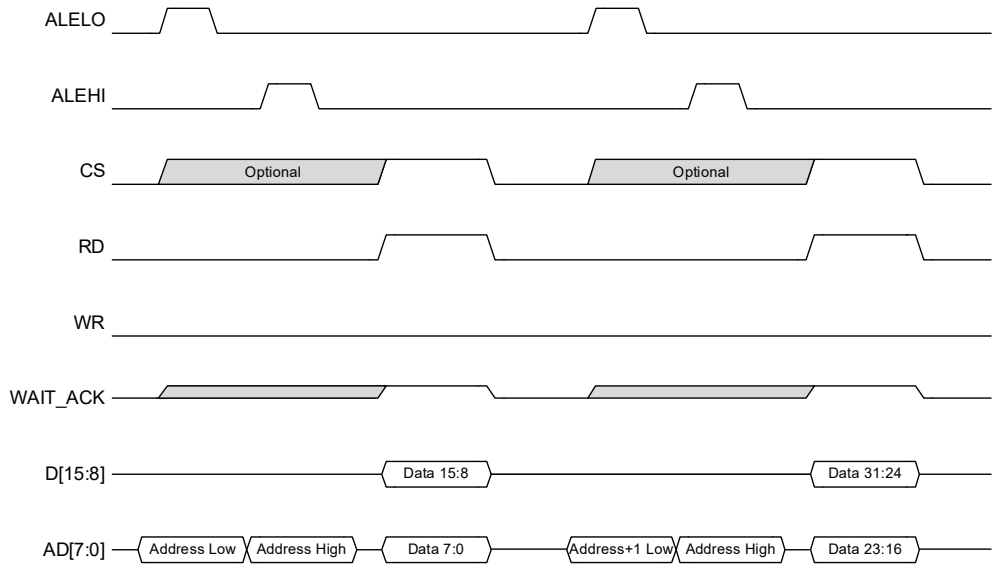
8.5.1.1 Dual Phase Address Latching

The figures in this section detail read and write operations in multiplexed addressing mode with dual phase address latching for 16 and 8-bit modes.

16-BIT READ

The WORD address is latched sequentially from **AD[7:0]**. A read on **D[15:8]** and **AD[15:0]** follows. The cycle is repeated for the other 16-bits of the DWORD.

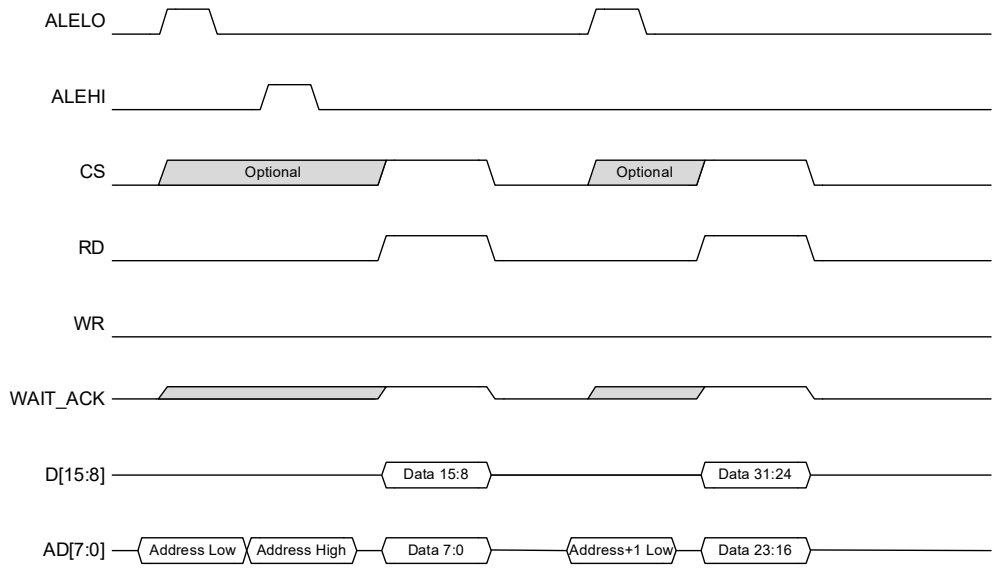
FIGURE 8-4: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT READ



16-BIT READ WITH SUPPRESSED ALEHI

The WORD address is latched sequentially from **AD[7:0]**. A read on **D[15:8]** and **AD[7:0]** follows. The lower address is then updated to access the opposite WORD.

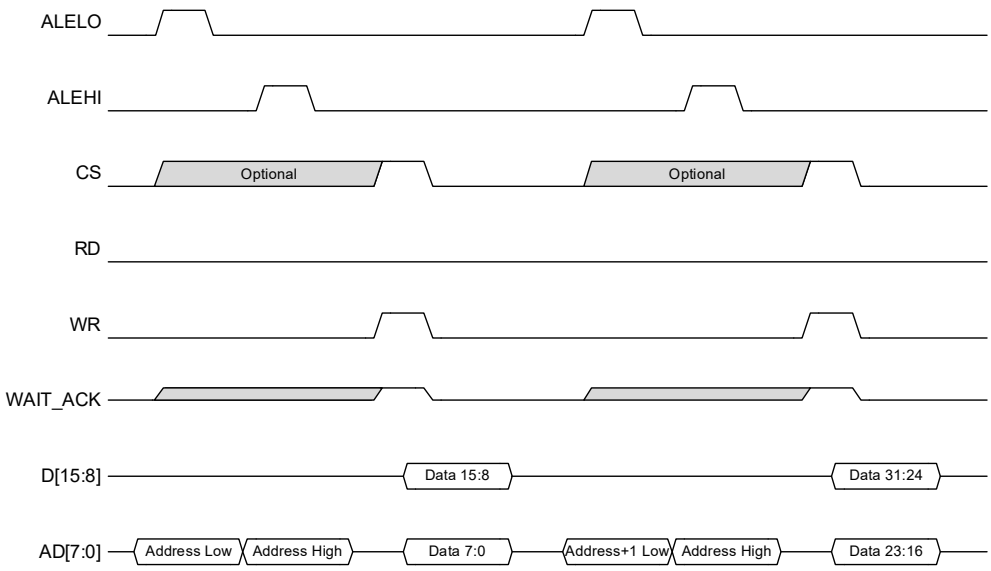
FIGURE 8-5: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT READ W/ O ALEHI



16-BIT WRITE

The WORD address is latched sequentially from **AD[7:0]**. A write on **D[15:8]** and **AD[7:0]** follows. The cycle is repeated for the other 16-bits of the DWORD.

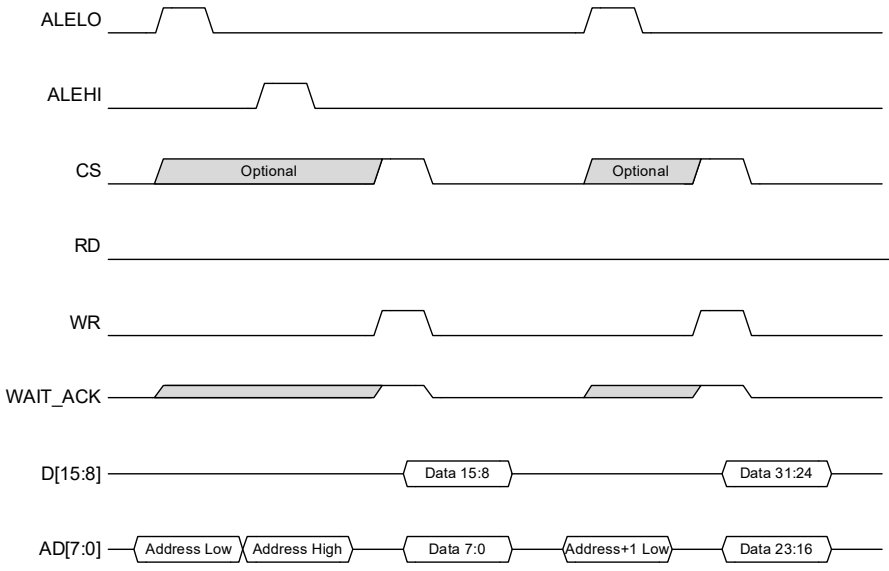
FIGURE 8-6: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT WRITE



16-BIT WRITE WITH SUPRESSED ALEHI

The WORD address is latched sequentially from **AD[7:0]**. A write on **D[15:8]** and **AD[7:0]** follows. The lower address is then updated to access the opposite WORD.

FIGURE 8-7: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT WRITE W/ O ALEHI

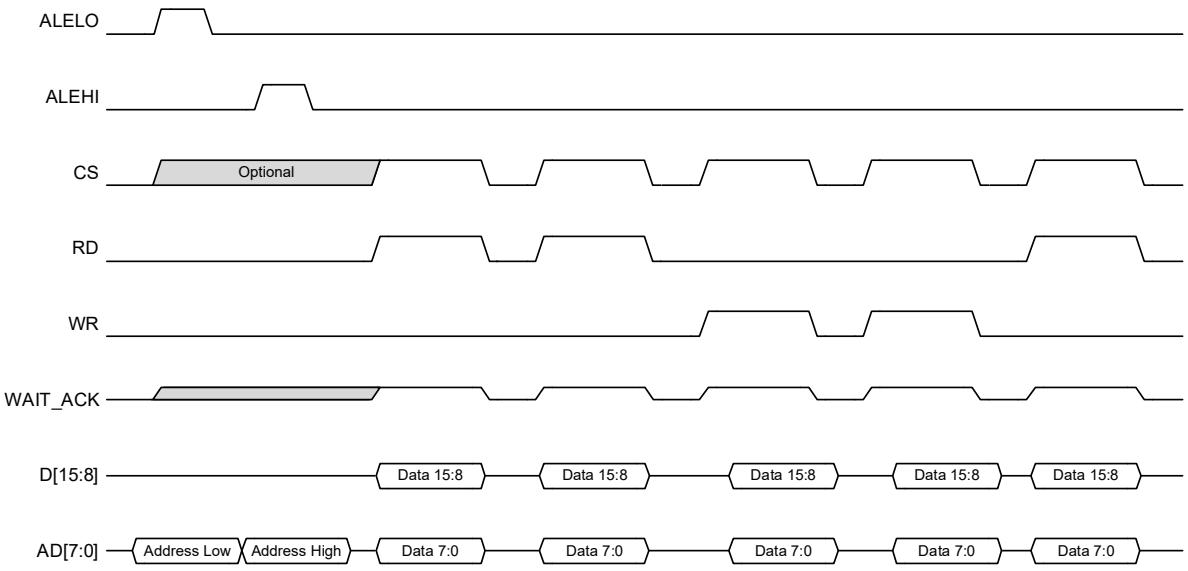


16-BIT READ AND WRITES TO CONSTANT ADDRESS

The WORD address is latched sequentially from **AD[7:0]**. A mix of reads and writes on **D[15:8]** and **AD[7:0]** follows.

Note:	Generally, two 16-bit reads to opposite WORDs of the same DWORD are required, with at least the lower address changing using ALELO. 16-bit reads and writes to the same WORD is a special case.

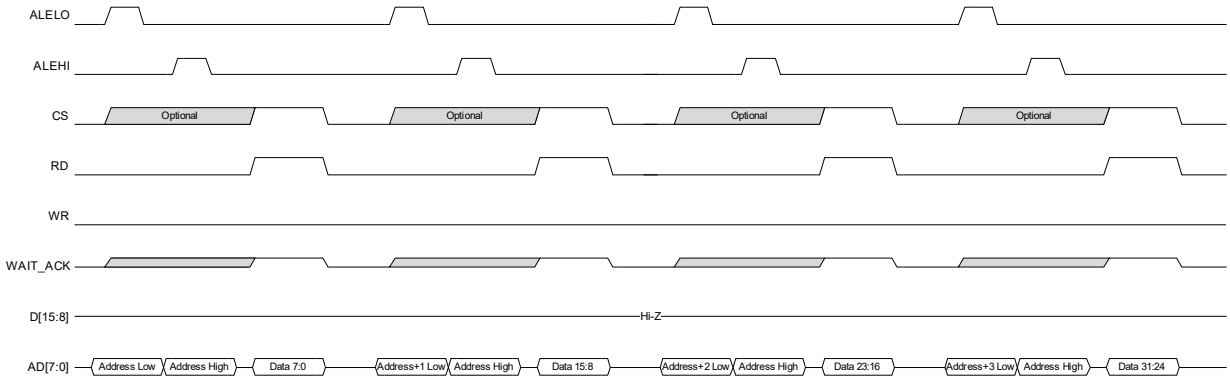
FIGURE 8-8: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT READS AND WRITES CONSTANT ADDRESS



8-BIT READ

The BYTE address is latched sequentially from **AD[7:0]**. A read on **AD[7:0]** follows. **D[15:8]** pins are not used or driven. The cycle is repeated for the other BYTEs of the DWORD.

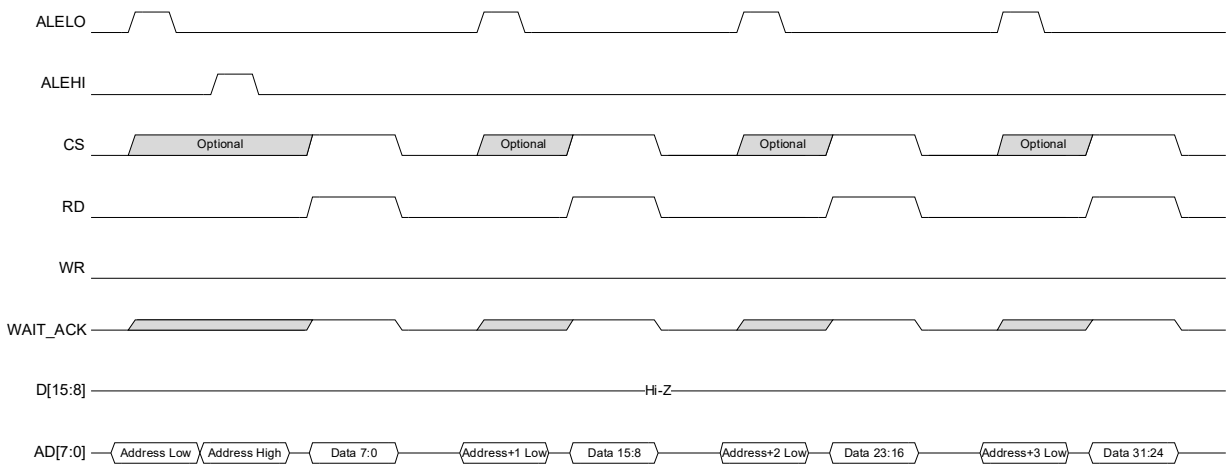
FIGURE 8-9: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT READ



8-BIT READ WITH SUPRESSED ALEHI

The BYTE address is latched sequentially from **AD[7:0]**. A read on **AD[7:0]** follows. **D[15:8]** pins are not used or driven. The lower address is then updated to access the other BYTEs.

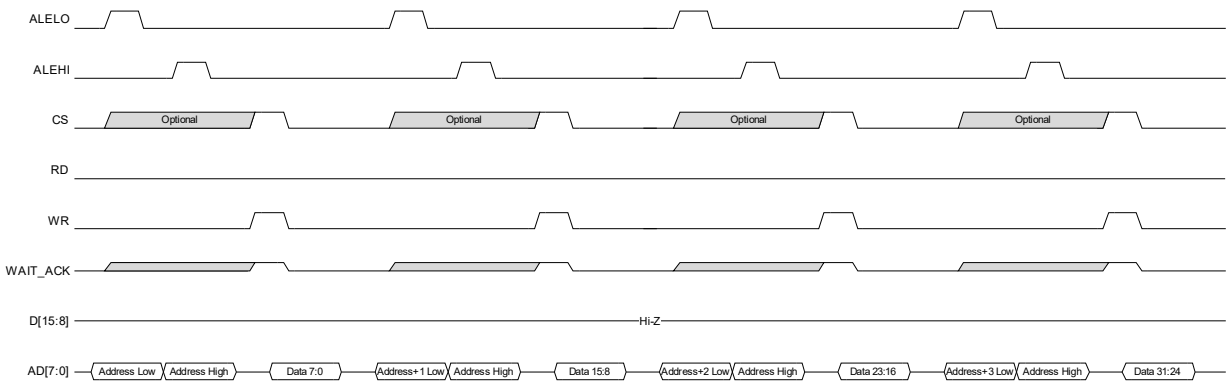
FIGURE 8-10: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT READ W/O ALEHI



8-BIT WRITE

The BYTE address is latched sequentially from **AD[7:0]**. A write on **AD[7:0]** follows. **D[15:8]** pins are not used or driven. The cycle is repeated for the other BYTEs of the DWORD.

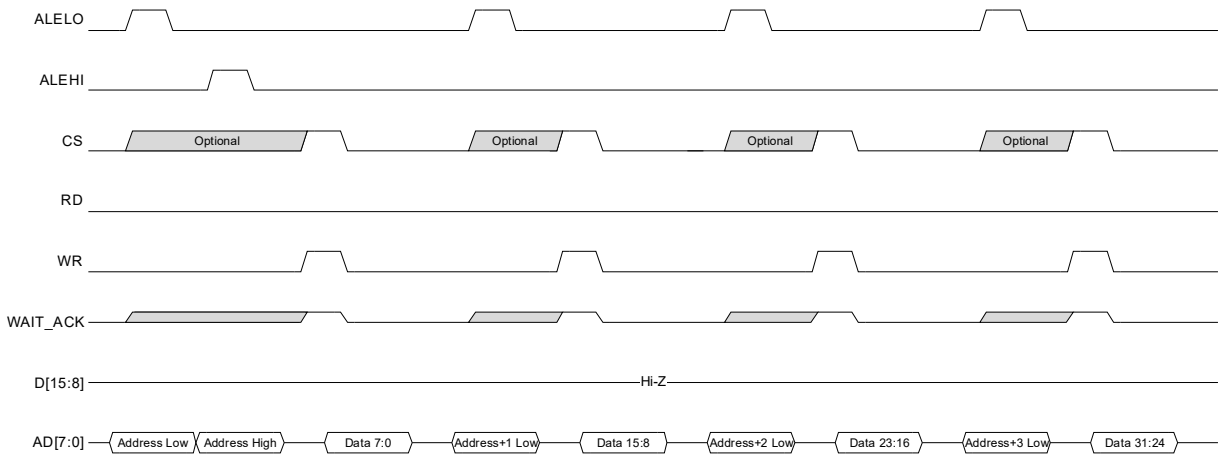
FIGURE 8-11: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT WRITE



8-BIT WRITE WITH SUPRESSED ALEHI

The BYTE address is latched sequentially from **AD[7:0]**. A write on **AD[7:0]** follows. **D[15:8]** pins are not used or driven. The lower address is then updated to access the other BYTEs.

FIGURE 8-12: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT WRITE W/ O ALEHI

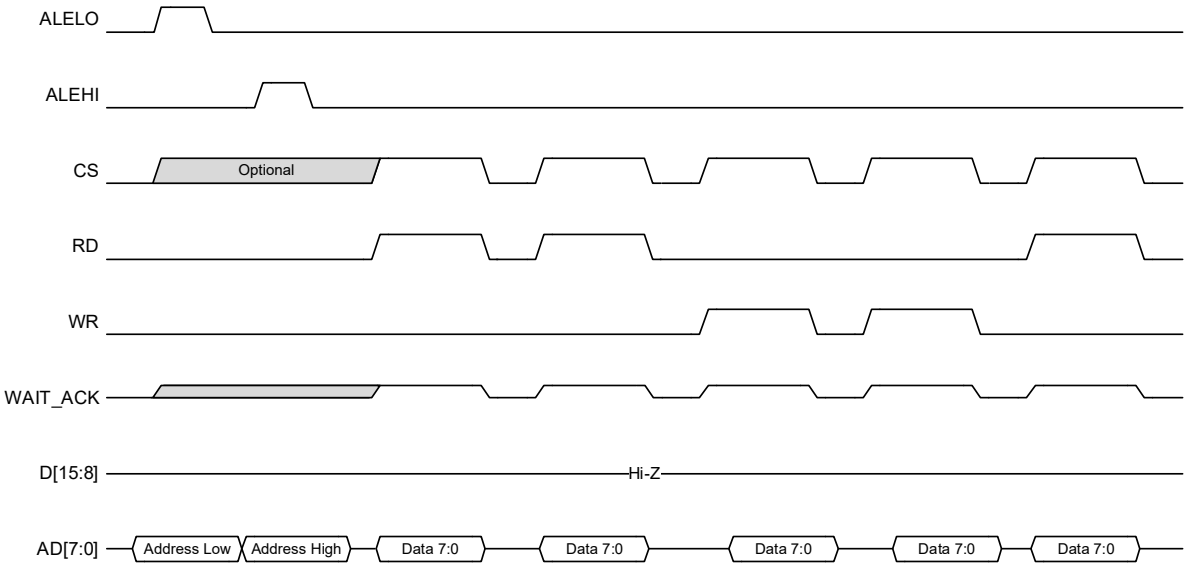


8-BIT READS AND WRITES TO CONSTANT ADDRESS

The BYTE address is latched sequentially from **AD[7:0]**. A mix of reads and writes on **AD[7:0]** follows. **D[15:8]** pins are not used or driven.

Note:	Generally, four 8-bit reads to opposite BYTEs of the same DWORD are required, with at least the lower address changing using ALELO. 8-bit reads and writes to the same BYTE is a special case.

FIGURE 8-13: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT READS AND WRITES CONSTANT ADDRESS



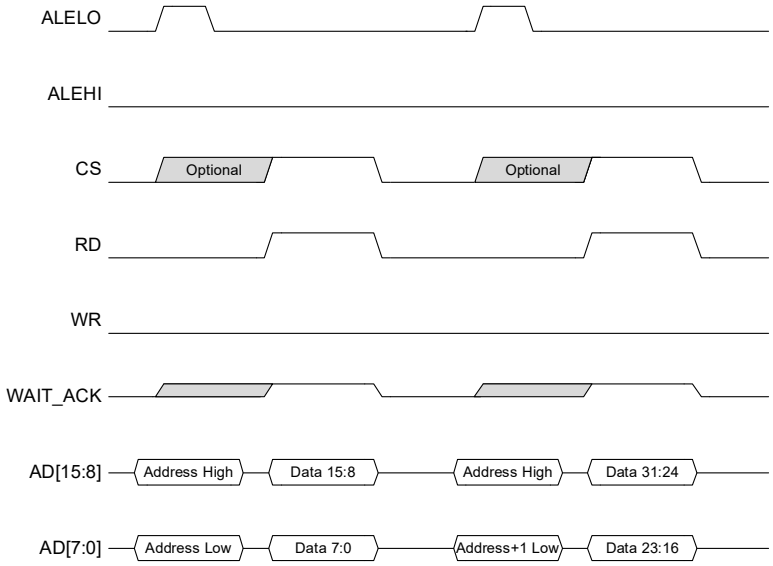
8.5.1.2 Single Phase Address Latching

The figures in this section detail multiplexed addressing mode with single phase addressing for 16 and 8-bit modes of operation.

16-BIT READ

The WORD address is latched simultaneously from **AD[7:0]** and **AD[15:8]**. A read on **AD[15:0]** follows. The cycle is repeated for the other 16-bits of the DWORD.

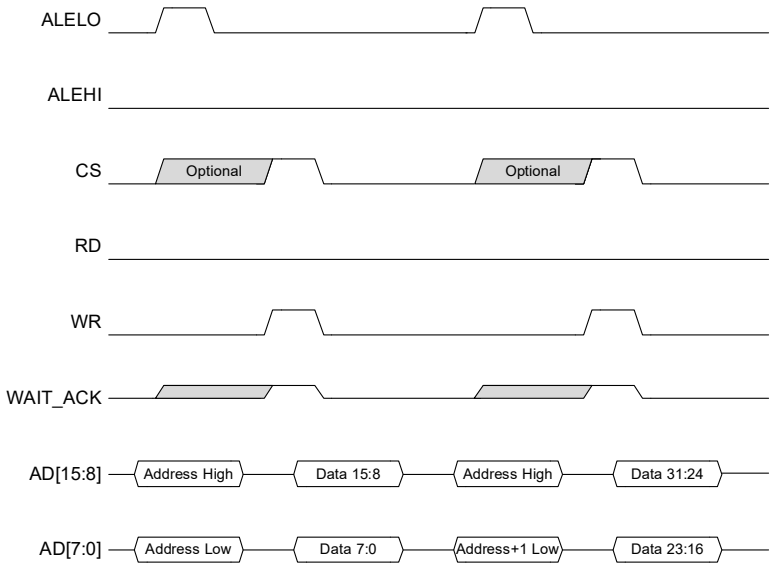
FIGURE 8-14: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 16-BIT READ



16-BIT WRITE

The WORD address is latched simultaneously from **AD[7:0]** and **AD[15:8]**. A write on **AD[15:0]** follows. The cycle is repeated for the other 16-bits of the DWORD.

FIGURE 8-15: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 16-BIT WRITE

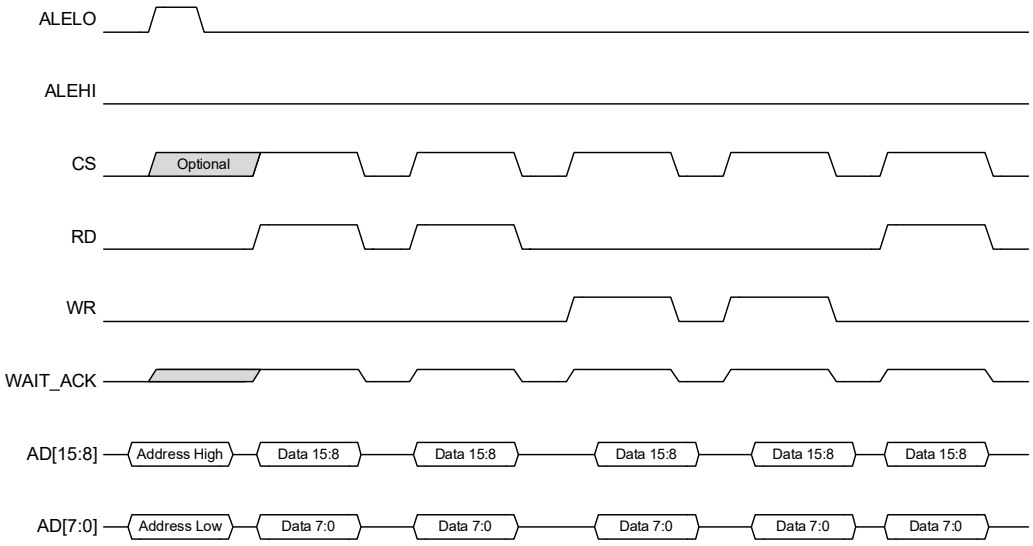


16-BIT READS AND WRITES TO CONSTANT ADDRESS

The WORD address is latched simultaneously from **AD[7:0]** and **AD[15:8]**. A mix of reads and writes on **AD[15:0]** follows.

Note:	Generally, two 16-bit reads to opposite WORDs of the same DWORD are required. 16-bit reads and writes to the same WORD is a special case.

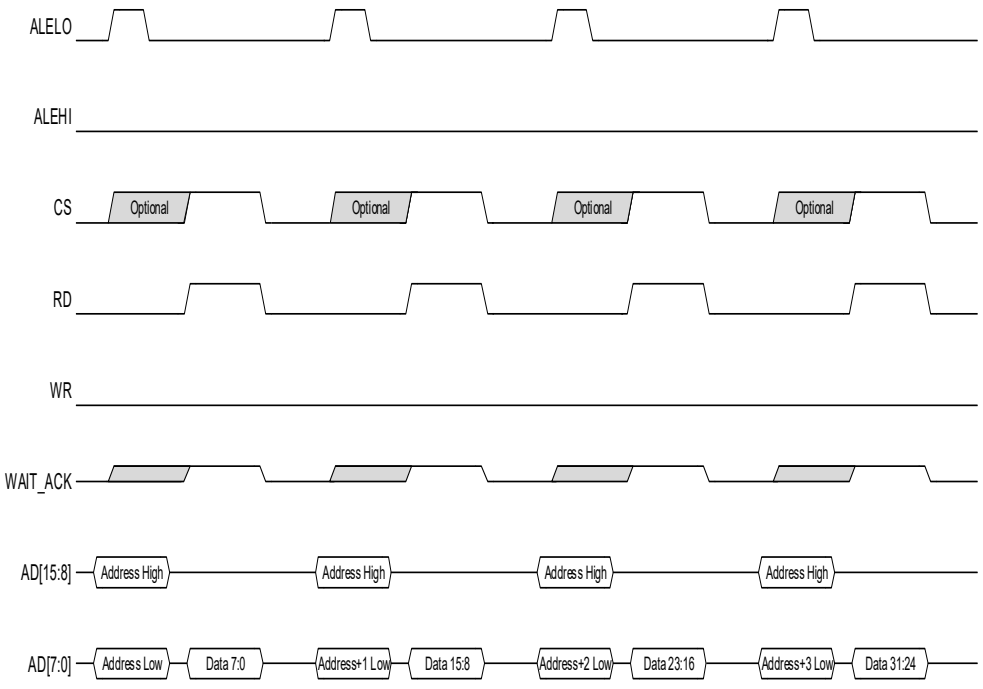
FIGURE 8-16: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 16-BIT READS AND WRITES CONSTANT ADDRESS



8-BIT READ

The BYTE address is latched simultaneously from **AD[7:0]** and **AD[15:8]**. A read on **AD[7:0]** follows. **AD[15:8]** pins are not used or driven for the data phase as the host could potentially continue to drive the upper address on these signals. The cycle is repeated for the other BYTEs of the DWORD.

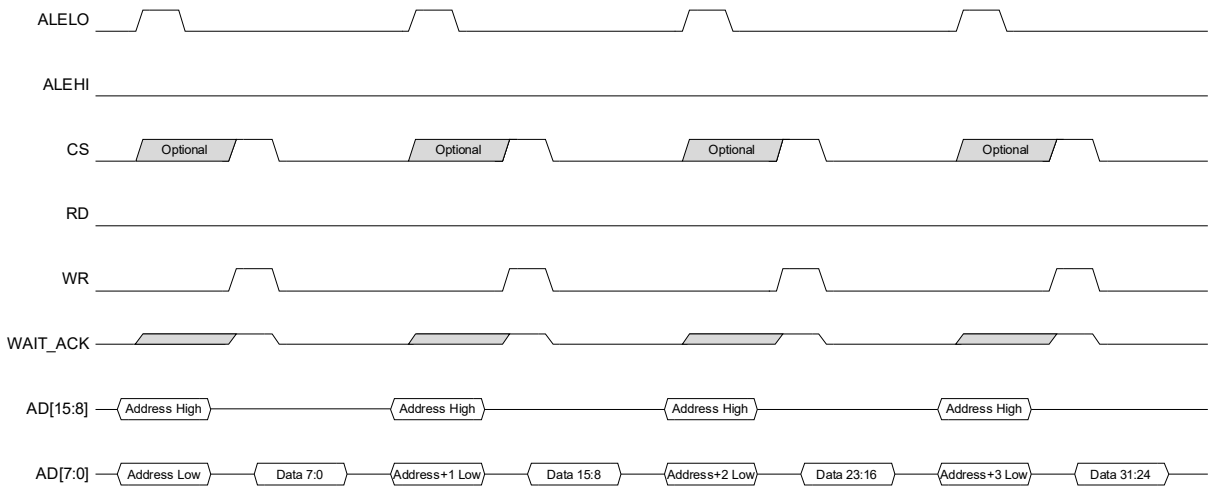
FIGURE 8-17: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 8-BIT READ



8-BIT WRITE

The BYTE address is latched simultaneously from **AD[7:0]** and **AD[15:8]**. A write on **AD[7:0]** follows. **AD[15:8]** pins are not used or driven for the data phase as the host could potentially continue to drive the upper address on these signals. The cycle is repeated for the other BYTEs of the DWORD.

FIGURE 8-18: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 8-BIT WRITE

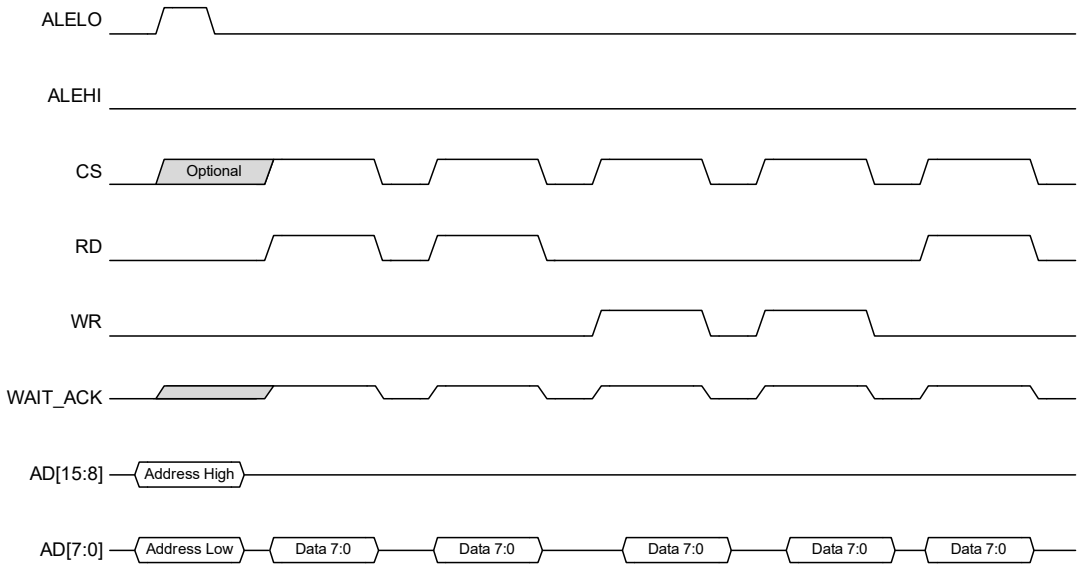


8-BIT READS AND WRITES TO CONSTANT ADDRESS

The BYTE address is latched simultaneously from **AD[7:0]** and **AD[15:8]**. A mix of reads and writes on **AD[7:0]** follows. **AD[15:8]** pins are not used or driven for the data phase as the host could potentially continue to drive the upper address on these signals.

Note:	Generally, four 8-bit reads to opposite BYTEs of the same DWORD are required. 8-bit reads and writes to the same BYTE is a special case.

FIGURE 8-19: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 8-BIT READS AND WRITES CONSTANT ADDRESS



8.5.1.3 RD_WR / ENB Control Mode Examples

The figures in this section detail read and write operations utilizing the alternative **RD_WR** and **ENB** signaling. The HBI read/write mode is selectable via the HBI Read/Write Mode bit of the PDI Configuration Register (HBI Modes).

Note: The examples in this section detail 16-bit mode with dual phase latching. However, the **RD_WR** and **ENB** signaling can be used identically in all other multiplexed addressing modes of operation as well as in 8-bit modes.

The examples in this section show the **ENB** signal active-high and the **RD_WR** signal low for read and high for write. The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI Read, Read/Write Polarity and HBI Write, Enable Polarity bits of the PDI Configuration Register (HBI Modes).

16-BIT

FIGURE 8-20: MULTIPLEXED ADDRESSING RD_WR / ENB CONTROL MODE EXAMPLE - 16-BIT READ

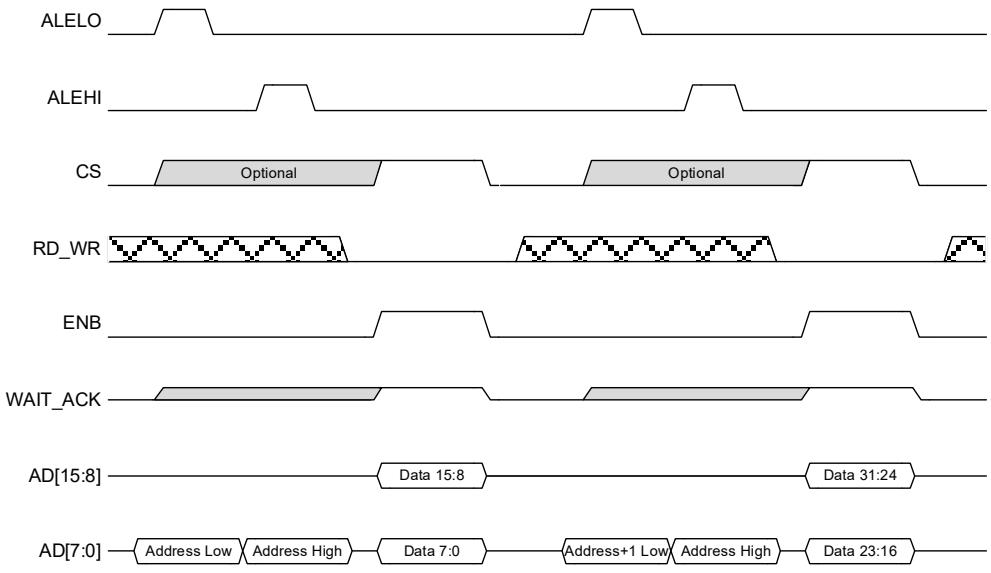
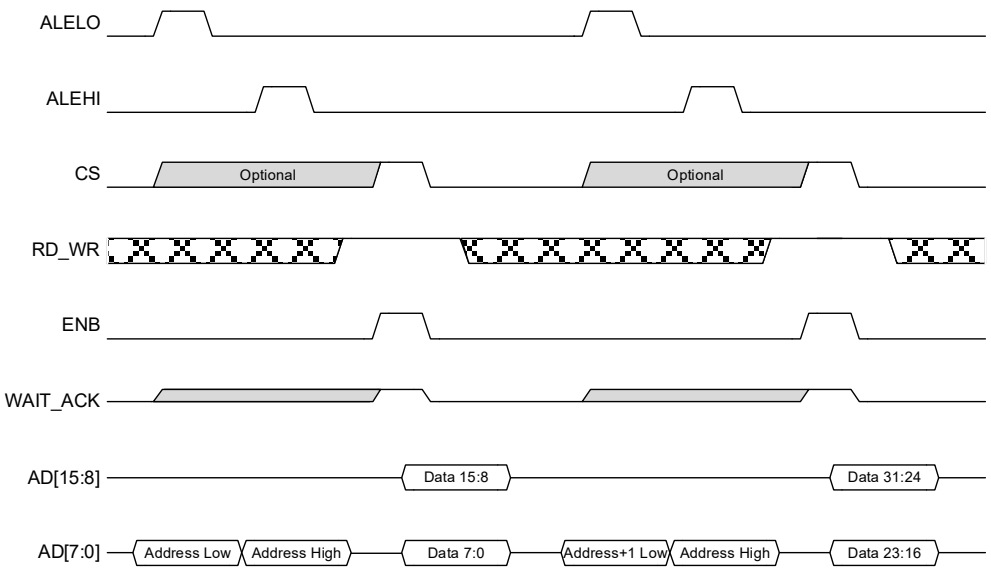


FIGURE 8-21: MULTIPLEXED ADDRESSING RD_WR / ENB CONTROL MODE EXAMPLE - 16-BIT WRITE



8.5.2 DEMULTIPLEXED ADDRESSING MODE FUNCTIONAL TIMING DIAGRAMS

The following timing diagrams illustrate example demultiplexed addressing mode read and write cycles for various configurations and bus sizes. These diagrams do not cover every supported host bus permutation, but are selected to detail the main configuration differences (bus size) within the demultiplexed addressing mode of operation.

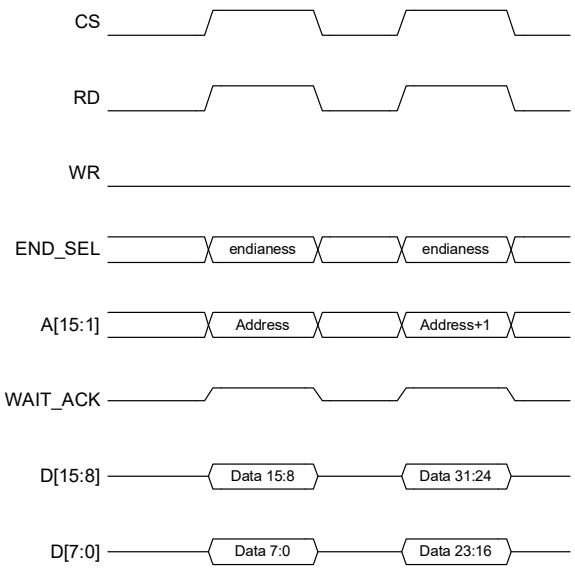
The following should be noted for the timing diagrams in this section:

- The diagrams in this section depict active-low push-pull **WAIT_ACK** and active-high **CS**, **RD**, and **WR** signals. The polarities of these signals are selectable via the HBI WAIT_ACK Polarity, HBI Chip Select Polarity, HBI Read, Read/Write Polarity, and HBI Write, Enable Polarity bits of the PDI Configuration Register (HBI Modes), respectively. Refer to Section 8.2.2, "Control Line Polarity and Buffer Type," on page 69 for additional details.
- The diagrams in this section depict little endian byte ordering. However, dynamic big and little endianness are supported via the endianness signal. Endianness changes only the order of the bytes involved, and not the overall timing requirements. Refer to Section 8.4.4, "Host Endianness," on page 79 for additional information.
- The diagrams in this section utilize **RD** and **WR** signals. Alternative **RD_WR** and **ENB** signaling is also supported, as shown in Section 8.5.2.1, "RD_WR / ENB Control Mode Examples". The HBI read/write mode is selectable via the HBI Read/Write Mode bit of the PDI Configuration Register. The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI Read, Read/Write Polarity and the HBI Write, Enable Polarity configuration inputs.
- For 16 and 8-bit modes, consecutive address cycles must be within the same DWORD until the DWORD is completely accessed (with the register exceptions noted above). Although BYTEs and WORDs can be accessed in any order, the diagrams in this section depict accessing the lower address BYTE or WORD first.

16-BIT READ

The WORD address and endianess is input concurrently with the control. Read data is driven on **D[15:0]** during **RD** active. The cycle is repeated for the other 16-bits of the DWORD.

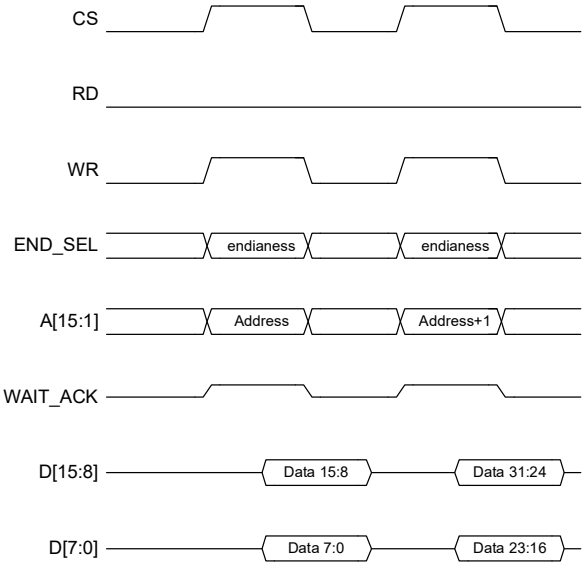
FIGURE 8-22: DEMULTIPLEXED ADDRESSING - 16-BIT READ



16-BIT WRITE

The WORD address and endianess is input concurrently with the control. Data on **D[15:0]** is written on the trailing edge of **WR**. The cycle is repeated for the other 16-bits of the DWORD.

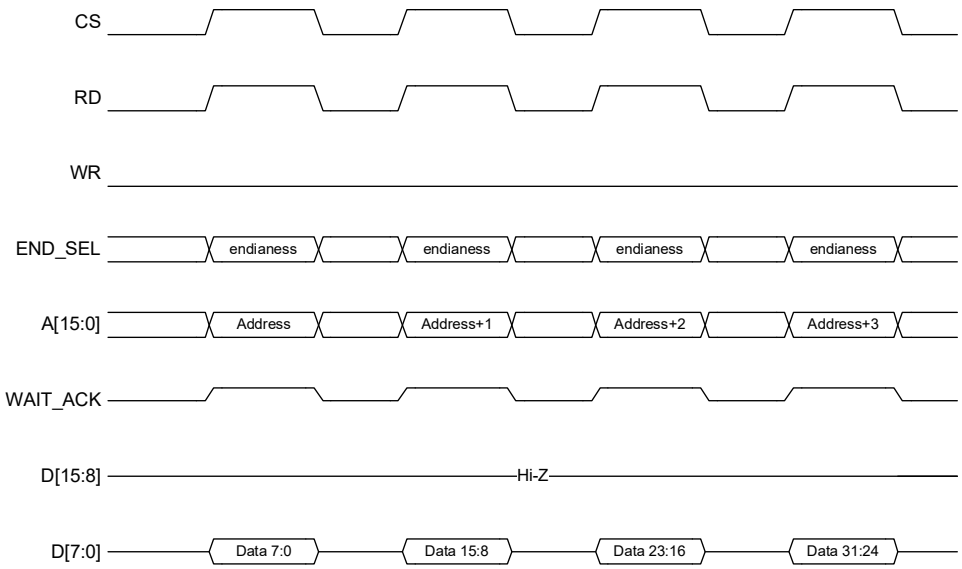
FIGURE 8-23: DEMULTIPLEXED ADDRESSING - 16-BIT WRITE



8-BIT READ

The BYTE address and endianess is input concurrently with the control. Read data is driven on **D[7:0]** during **RD** active. **D[15:8]** pins are not used or driven. The cycle is repeated for the other BYTEs of the DWORD.

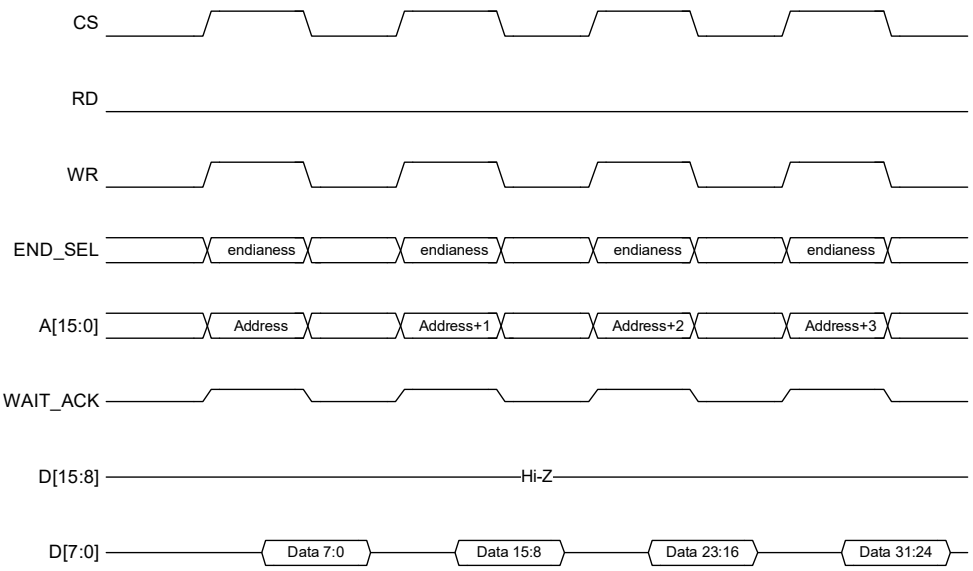
FIGURE 8-24: DEMULTIPLEXED ADDRESSING - 8-BIT READ



8-BIT WRITE

The BYTE address and endianess is input concurrently with the control. Data on **D[7:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven. The cycle is repeated for the other BYTEs of the DWORD.

FIGURE 8-25: DEMULTIPLEXED ADDRESSING - 8-BIT WRITE



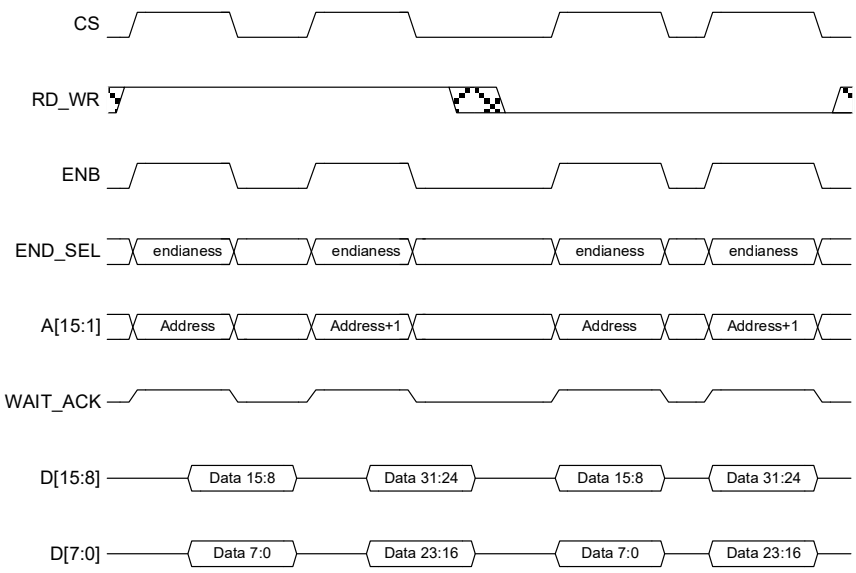
8.5.2.1 RD_WR / ENB Control Mode Examples

The figures in this section detail read and write operations utilizing the alternative **RD_WR** and **ENB** signaling. The HBI read/write mode is selectable via the HBI Read/Write Mode bit of the PDI Configuration Register (HBI Modes).

Note:	The examples in this section detail 16-bit mode. However, the **RD_WR** and **ENB** signaling can be used identically in 8-bit modes. The examples in this section show the **ENB** signal active-high and the **RD_WR** signal low for read and high for write. The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI Read, Read/Write Polarity and HBI Write, Enable Polarity bits of the PDI Configuration Register (HBI Modes).

16-BIT

FIGURE 8-26: DEMULTIPLEXED ADDRESSING RD_WR / ENB CONTROL MODE EXAMPLE - 16-BIT WRITE/READ



8.5.3 INDEXED ADDRESS MODE FUNCTIONAL TIMING DIAGRAMS

The following timing diagrams illustrate example indexed (non-multiplexed) addressing mode read and write cycles for various configurations and bus sizes. These diagrams do not cover every supported host bus permutation, but are selected to detail the main configuration differences (bus size, Configuration / Index / Data cycles) within the indexed addressing mode of operation.

The following should be noted for the timing diagrams in this section:

- The diagrams in this section depict active-low push-pull **WAIT_ACK** and active-high **CS**, **RD**, and **WR** signals. The polarities of these signals are selectable via the HBI WAIT_ACK Polarity, HBI Chip Select Polarity, HBI Read, Read/Write Polarity, and HBI Write, Enable Polarity bits of the PDI Configuration Register (HBI Modes), respectively. Refer to Section 8.2.2, "Control Line Polarity and Buffer Type," on page 69 for additional details.
- The diagrams in this section depict little endian byte ordering. However, configurable big and little endianness are supported via the endianness bits in the Host Bus Interface Configuration Register. Endianness changes only the order of the bytes involved, and not the overall timing requirements. Refer to Section 8.4.4, "Host Endianness," on page 79 for additional information.
- The diagrams in this section utilize **RD** and **WR** signals. Alternative **RD_WR** and **ENB** signaling is also supported, as shown in Section 8.5.3.4, "RD_WR / ENB Control Mode Examples". The HBI read/write mode is selectable via the HBI Read/Write Mode bit of the PDI Configuration Register. The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI Read, Read/Write Polarity and HBI Write, Enable Polarity configuration inputs.
- For 16 and 8-bit modes, consecutive address cycles must be within the same DWORD until the DWORD is completely accessed (some internal registers are excluded from this requirement). Although BYTEs and WORDs can be accessed in any order, the diagrams in this section depict accessing the lower address BYTE or WORD first.

8.5.3.1 Configuration Register Data Access

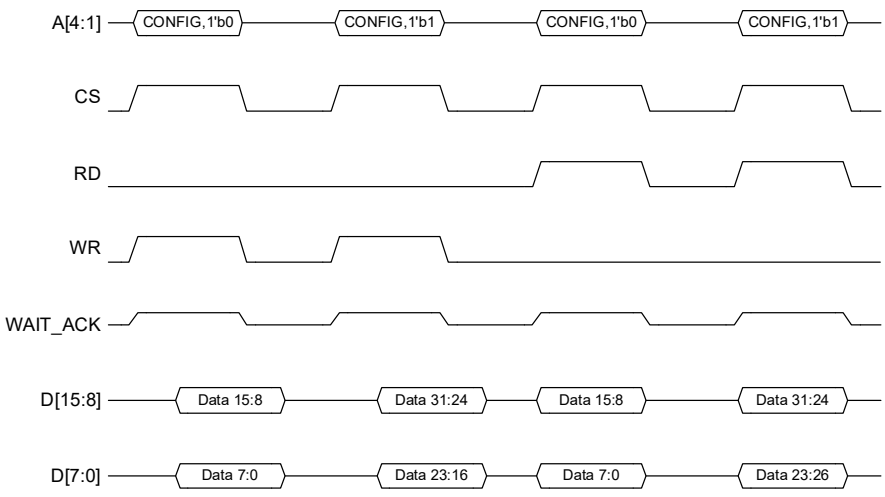
The figures in this section detail configuration register read and write operations in indexed address mode for 16 and 8-bit modes.

16-BIT READ AND WRITE

For writes, the address is set to access the lower WORD of the Configuration Register. Data on **D[15:0]** is written on the trailing edge of **WR**. The cycle repeats for the upper WORD of the Configuration Register, if desired by the host.

For reads, the address is set to access the lower WORD of the Configuration Register. Read data is driven on **D[15:0]** during **RD** active. The cycle repeats for the upper WORD of the Configuration Register, if desired by the host.

FIGURE 8-27: INDEXED ADDRESSING CONFIGURATION REGISTER ACCESS - 16-BIT WRITE/READ

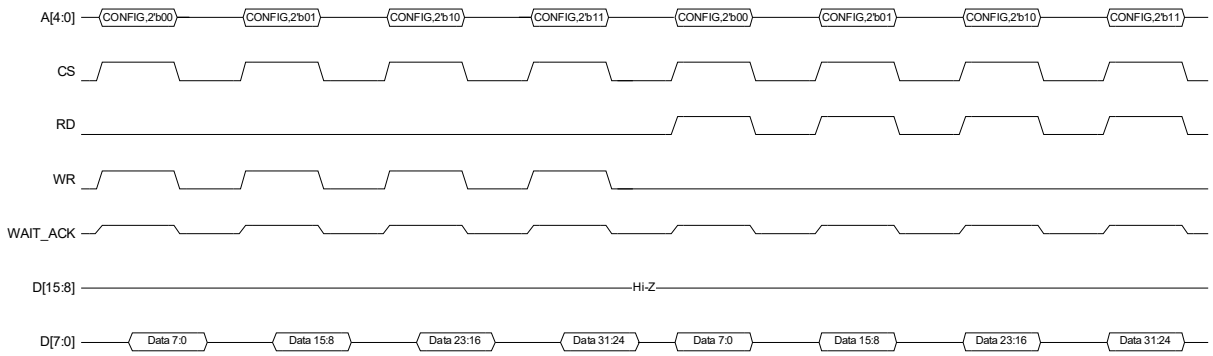


8-Bit READ AND WRITE

For writes, the address is set to access the lower BYTE of the Configuration Register. Data on **D[7:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining BYTEs of the Configuration Register, if desired by the host.

For reads, the address is set to access the lower BYTE of the Configuration Register. Read data is driven on **D[7:0]** during **RD** active. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining BYTEs of the Configuration Register, if desired by the host.

FIGURE 8-28: INDEXED ADDRESSING CONFIGURATION REGISTER ACCESS - 8-BIT WRITE/ READ



8.5.3.2 Index Register Data Access

The figures in this section detail index register read and write operations in indexed address mode for 16 and 8-bit modes.

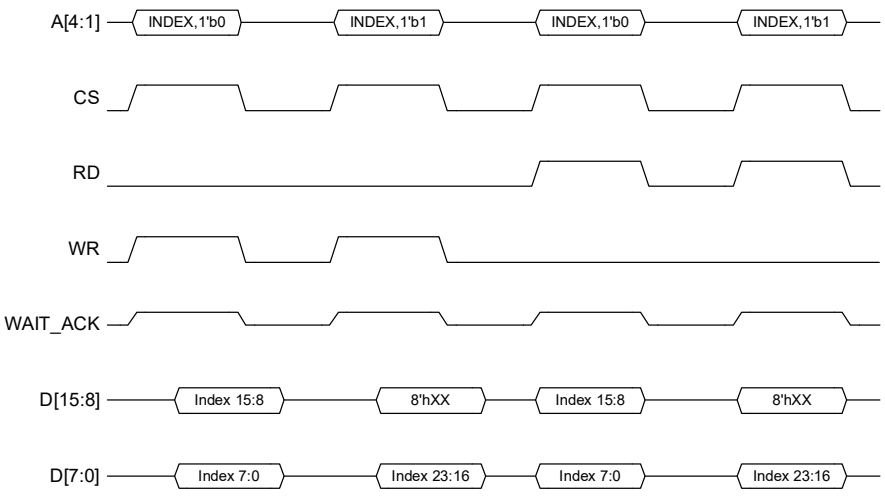
16-BIT READ AND WRITE

For writes, the address is set to access the lower WORD of one of the Index Registers. Data on **D[15:0]** is written on the trailing edge of **WR**. The cycle repeats for the upper WORD of the Index Register, if desired by the host.

For reads, the address is set to access the lower WORD of one of the Index Registers. Read data is driven on **D[15:0]** during **RD** active. The cycle repeats for the upper WORD of the Index Register, if desired by the host.

Note:	The upper BYTE of Index Register is reserved and don't care.

FIGURE 8-29: INDEXED ADDRESSING INDEX REGISTER ACCESS - 16-BIT WRITE/READ



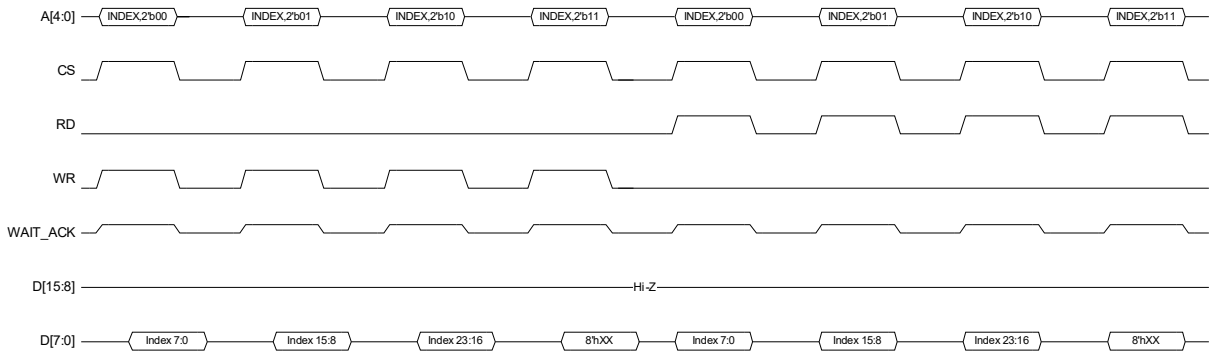
8-BIT READ AND WRITE

For writes, the address is set to access the lower BYTE of one of the Index Registers. Data on **D[7:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining BYTEs of the Index Register, if desired by the host.

For reads, the address is set to access the lower BYTE of one of the Index Registers. Read data is driven on **D[7:0]** during **RD** active. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining BYTEs of the Index Register, if desired by the host.

Note:	The upper BYTE of Index Register is reserved and don't care. Therefore reads and writes to that BYTE is not useful.

FIGURE 8-30: INDEXED ADDRESSING INDEX REGISTER ACCESS - 8-BIT WRITE/READ



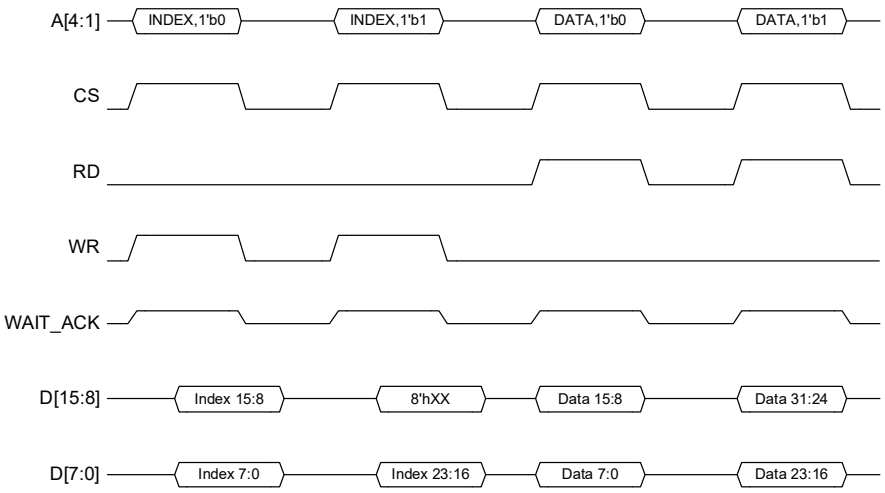
8.5.3.3 Internal Register Data Access

The figures in this section detail typical internal register data read and write cycles in indexed address mode for 16 and 8-bit modes. This includes an index register write followed by either a data read or write.

16-BIT READ

One of the Index Registers is set as described above. The address is then set to access the lower WORD of the corresponding Data Register. Read data is driven on **D[15:0]** during **RD** active. The cycle repeats for the upper WORD of the Data Register.

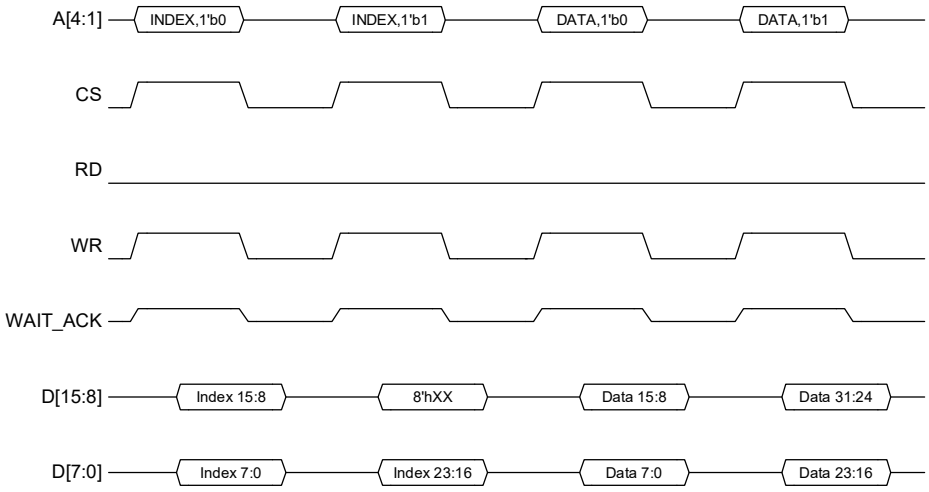
FIGURE 8-31: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 16-BIT READ



16-BIT WRITE

One of the Index Registers is set as described above. The address is then set to access the corresponding Data Register. Data on **D[15:0]** is written on the trailing edge of **WR**. The cycle repeats for the upper WORD of the Data Register.

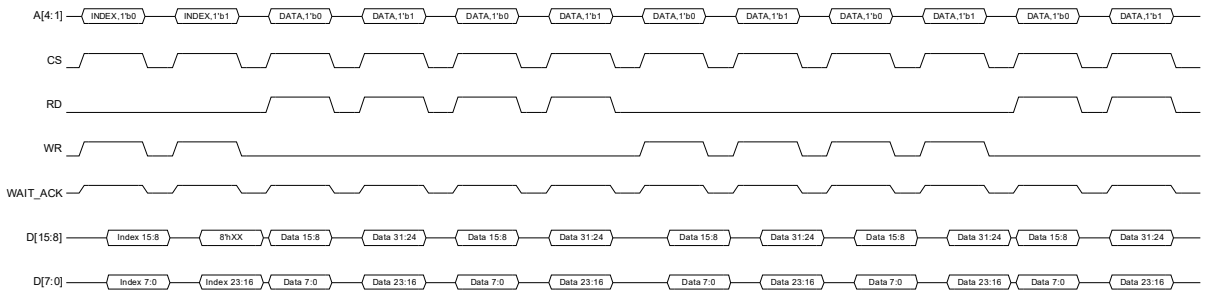
FIGURE 8-32: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 16-BIT WRITE



16-BIT READ AND WRITES TO CONSTANT INTERNAL ADDRESS

One of the Index Registers is set as described above. A mix of reads and writes on **D[15:0]** follows, with each read or write consisting of an access to both the lower and upper WORDs of the corresponding Data Register.

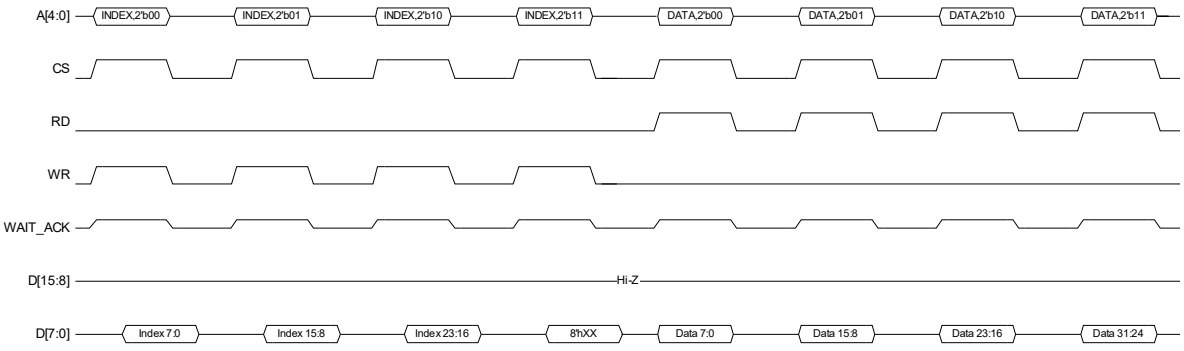
FIGURE 8-33: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 16-BIT READS AND WRITES CONSTANT ADDRESS



8-BIT READ

One of the Index Registers is set as described above. The address is then set to access the lower BYTE of the corresponding Data Register. Read data is driven on **D[7:0]** during RD active. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining BYTEs of the Data Register.

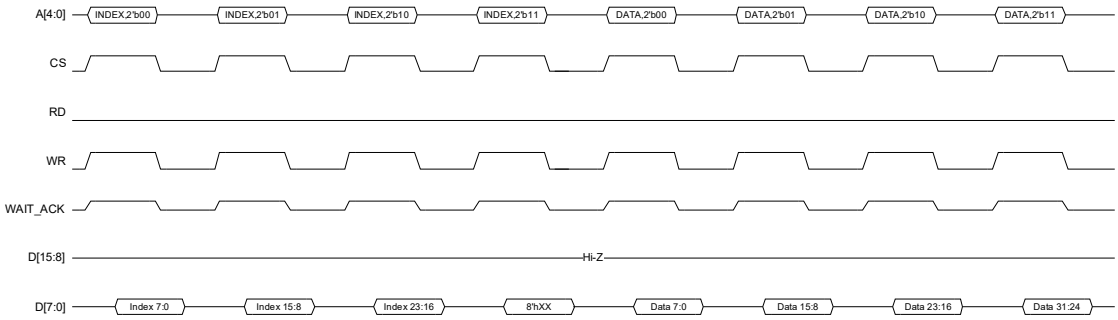
FIGURE 8-34: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 8-BIT READ



8-BIT WRITE

One of the Index Registers is set as described above. The address is then set to access the corresponding Data Register. Data on **D[7:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining BYTEs of the Data Register.

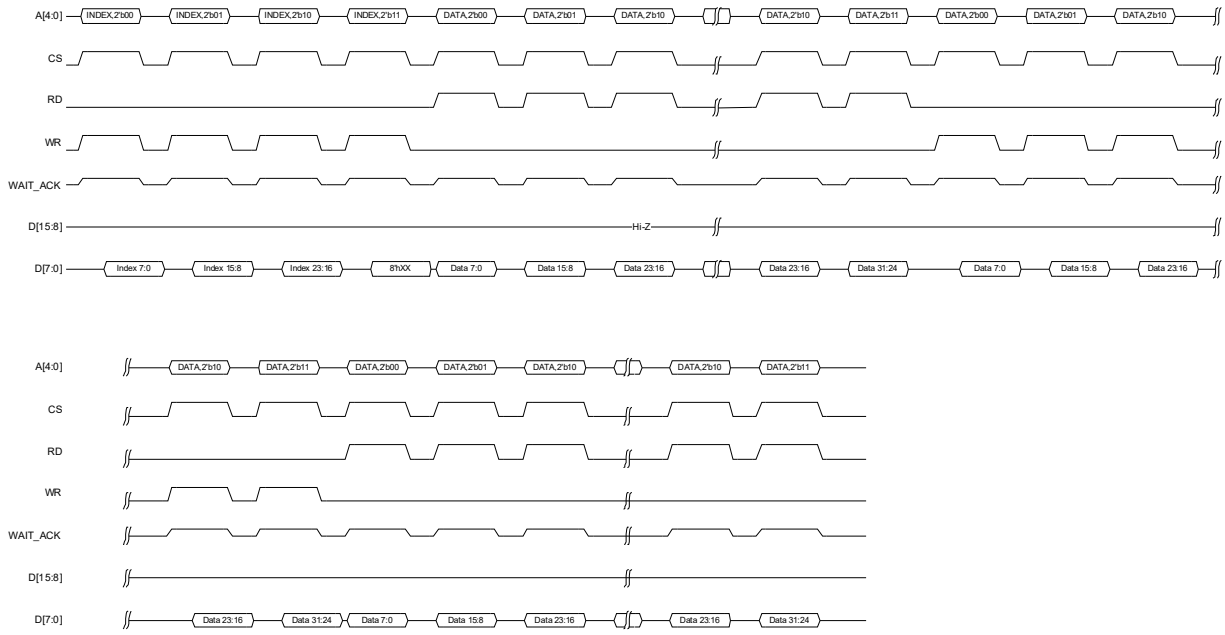
FIGURE 8-35: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 8-BIT WRITE



8-BIT READS AND WRITES TO CONSTANT INTERNAL ADDRESS

One of the Index Registers is set as described above. A mix of reads and writes on **D[7:0]** follows, with each read or write consisting of an access to all four BYTES of the corresponding Data Register.

FIGURE 8-36: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 8-BIT READS AND WRITES CONSTANT ADDRESS



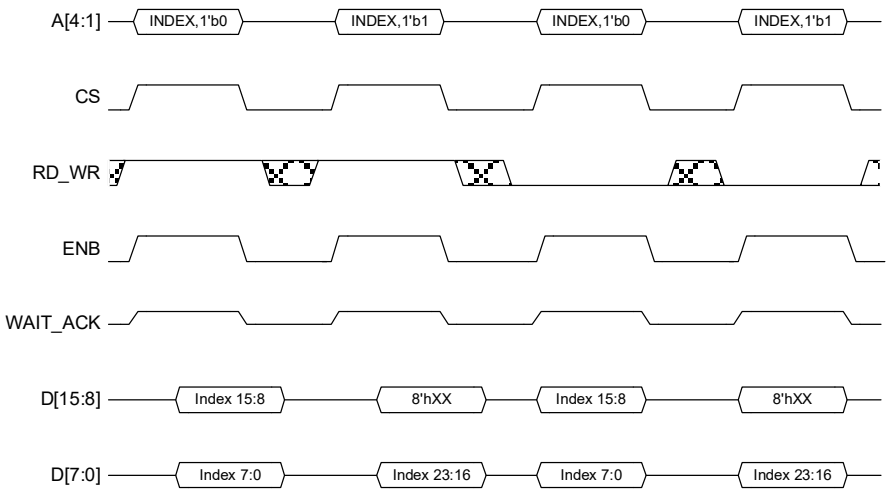
8.5.3.4 RD_WR / ENB Control Mode Examples

The figures in this section detail read and write operations utilizing the alternative **RD_WR** and **ENB** signaling. The HBI read/write mode is selectable via the HBI Read/Write Mode bit of the PDI Configuration Register (HBI Modes).

Note:	The examples in this section detail 16-bit mode access to an Index Register. However, the **RD_WR** and **ENB** signaling can be used identically for all other accesses including Index Register Bypass FIFO Access as well as in 8-bit modes. The examples in this section show the **ENB** signal active-high and the **RD_WR** signal low for read and high for write. The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI Read, Read/Write Polarity and HBI Write, Enable Polarity bits of the PDI Configuration Register (HBI Modes).

16-BIT

FIGURE 8-37: INDEXED ADDRESSING RD_WR / ENB CONTROL MODE EXAMPLE - 16-BIT WRITE/READ



8.6 Functional Timing Diagrams - EtherCAT Direct Mapped Mode

The following sections present functional timing diagrams while in EtherCAT Direct Mapped mode.

8.6.1 MULTIPLEXED ADDRESSING MODE FUNCTIONAL TIMING DIAGRAMS

The following timing diagrams illustrate example multiplexed addressing mode read and write cycles for various address/data configurations and bus sizes. These diagrams do not cover every supported host bus permutation, but are selected to detail the main configuration differences (bus size, dual/single phase address latching) within the multiplexed addressing mode of operation.

The following should be noted for the timing diagrams in this section:

- The diagrams in this section depict active-low push-pull **WAIT_ACK** and active-high **BE1/0**, **ALEHI/ALELO**, **CS**, **RD**, and **WR** signals. The polarities of these signals are selectable via the HBI WAIT_ACK Polarity, HBI BE1/BE0 Polarity, HBI ALE Polarity, HBI Chip Select Polarity, HBI Read, Read/Write Polarity, and HBI Write, Enable Polarity bits of the PDI Configuration Register and Extended PDI Configuration Register (HBI Modes), respectively. Refer to Section 8.2.2, "Control Line Polarity and Buffer Type," on page 69 for additional details.
- The diagrams in this section depict little endian byte ordering. However, dynamic big and little endianness are supported for 16-bit mode via the endianess signal. Endianness changes only the order of the bytes involved, and not the overall timing requirements. Refer to Section 8.4.4, "Host Endianness," on page 79 for additional information.
- The diagrams in this section utilize **RD** and **WR** signals. Alternative **RD_WR** and **ENB** signaling is also supported, similar to the multiplexed examples in Section 8.5.1.3, "RD_WR / ENB Control Mode Examples". The HBI read/write mode is selectable via the HBI Read/Write Mode bit of the PDI Configuration Register. The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI Read, Read/Write Polarity and HBI Write, Enable Polarity configuration inputs.
- Qualification of the **ALELO** and/or **ALEHI** with the **CS** signal is selectable via the HBI ALE Qualification bit of the PDI Configuration Register. Refer to Section 8.3.1.1, "Single Phase Address Latching," on page 69 and Section 8.3.1.2, "Dual Phase Address Latching," on page 69 for additional information.
- In dual phase address latching mode, the **ALEHI** and **ALELO** cycles can be in any order. Either or both **ALELO** and **ALEHI** cycles may be skipped and the device retains the last latched address.
- In single phase address latching mode, the **ALELO** cycle may be skipped and the device retains the last latched address.

> **Note:** While accessing non-EtherCAT Core registers the **ALELO** cycle is normally not skipped since sequential BYTEs or WORDs are accessed in order to satisfy a complete DWORD cycle. However, there are registers for which a single BYTE or WORD access is allowed, in which case multiple accesses to these registers may be performed without the need to re-latch the repeated address.

- While accessing non-EtherCAT Core registers, for proper DWORD assembly, consecutive address cycles must be within the same DWORD until the DWORD is completely accessed (with the register exceptions noted above). Although BYTEs and WORDs can be accessed in any order, the diagrams in this section depict accessing the lower address BYTE or WORD first.
- While accessing non-EtherCAT Core registers in 16-bit mode, all cycles are 16-bit wide with **BE1/BE0** unused. While accessing EtherCAT Core registers and the Process RAM in 16-bit mode, 8 or 16-bit transfers can be selected via the **BE1/BE0** pins.

8.6.1.1 Dual Phase Address Latching

The figures in this section detail read and write operations in multiplexed addressing mode with dual phase address latching for 16 and 8-bit modes.

16-BIT MODE READ

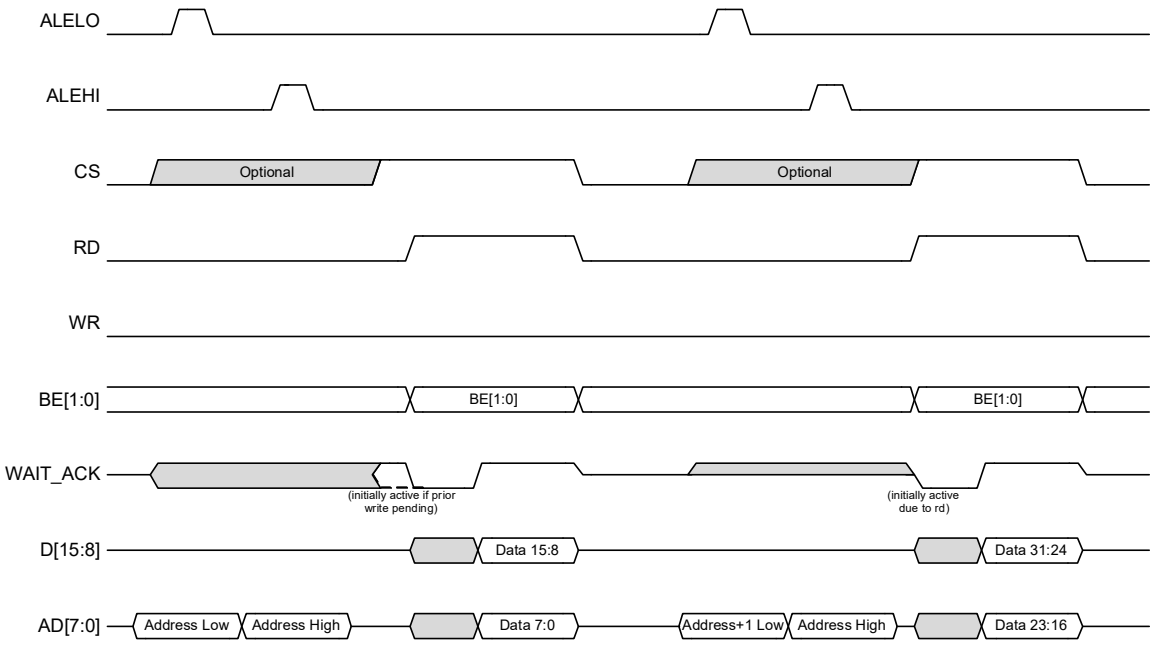
The WORD address is latched sequentially from **AD[7:0]**.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **RD** active. **CS** and **RD** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

D[15:8] and **AD[7:0]** are driven active on during **RD** active and valid once **WAIT_ACK** becomes inactive.

The cycle is repeated for the other 16-bits of the DWORD if required.

FIGURE 8-38: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT READ



16-BIT MODE POSTED WRITE

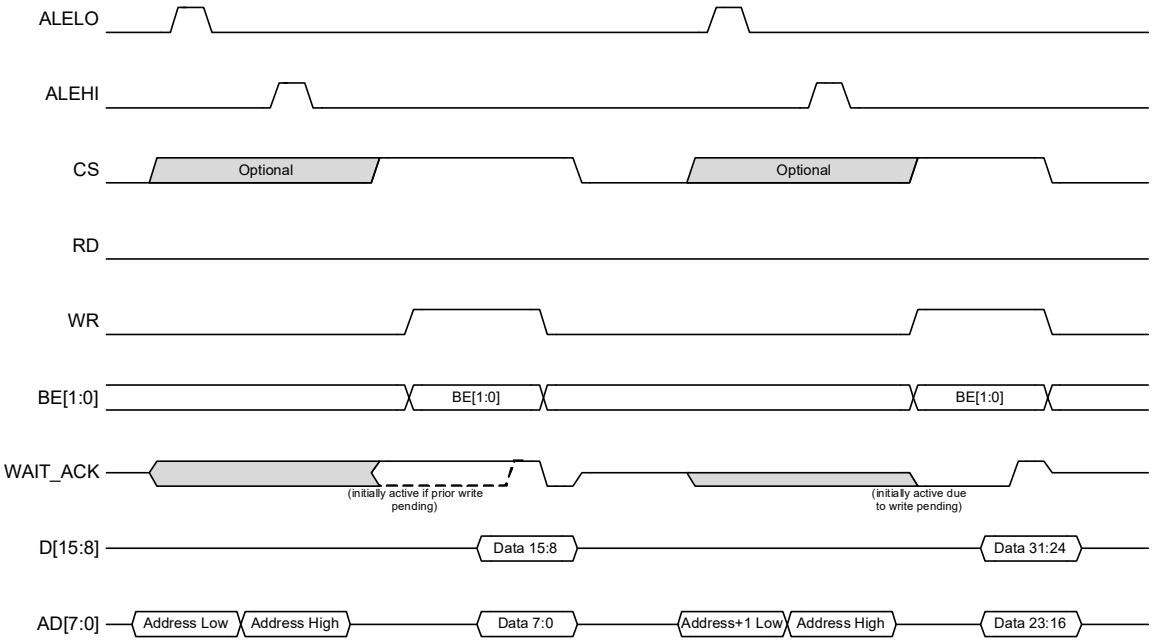
The WORD address is latched sequentially from **AD[7:0]**.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **WR** inactive. **CS** and **WR** may become inactive simultaneously in which case **WAIT_ACK** active is not seen (second example).

Data on **D[15:8]** and **AD[7:0]** is written on the trailing edge of **WR**.

The cycle is repeated for the other 16-bits of the DWORD if required.

FIGURE 8-39: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT POSTED WRITE



16-BIT MODE NON-POSTED WRITE

The WORD address is latched sequentially from **AD[7:0]**.

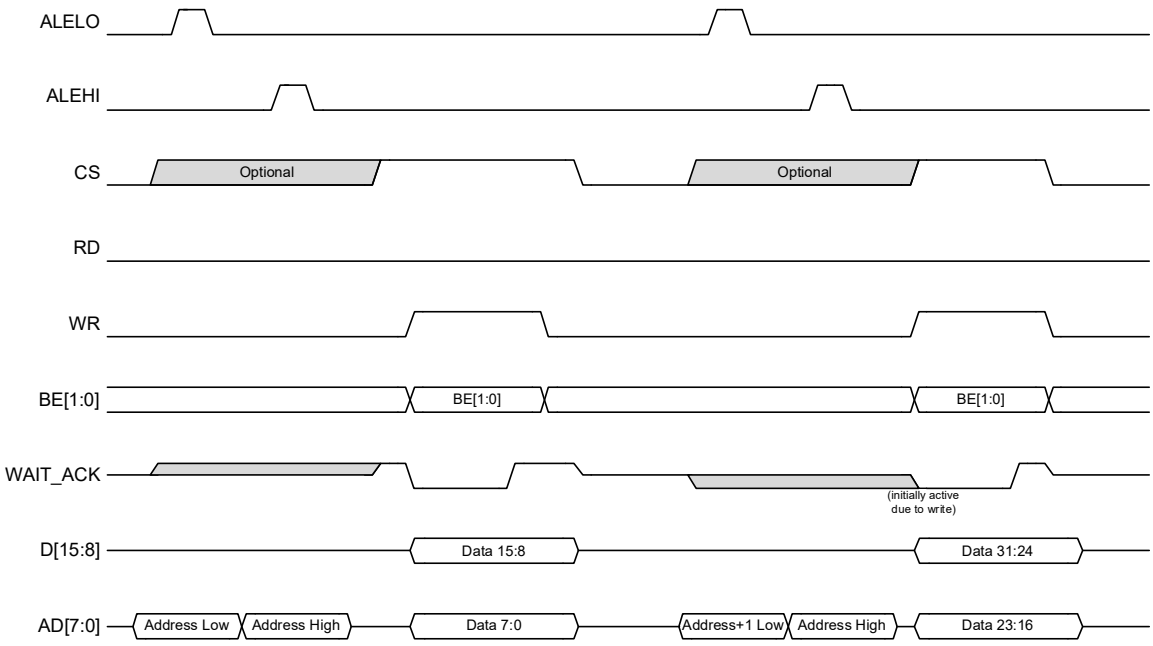
WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven inactive since there will never be a prior pending write. **WAIT_ACK** is driven active upon **WR** active. **CS** and **WR** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

Data on **D[15:8]** and **AD[7:0]** is written starting on the leading edge of **WR**. **WR** may be made inactive once **WAIT_ACK** becomes inactive.

For EtherCAT Core registers and the Process RAM, the write cycle occurs during **WR** active. For-non EtherCAT Core registers, the write cycle occurs upon **WR** inactive. Therefore Data is held until **WR** inactive.

The cycle is repeated for the other 16-bits of the DWORD if required.

FIGURE 8-40: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT NON-POSTED WRITE



8-BIT MODE READ

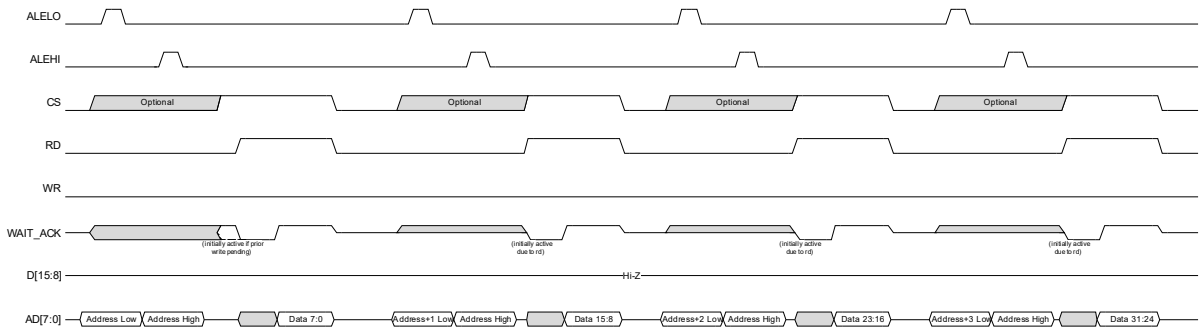
The BYTE address is latched sequentially from AD[7:0].

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **RD** active. **CS** and **RD** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

AD[7:0] are driven active on during **RD** active and valid once **WAIT_ACK** becomes inactive. **D[15:8]** pins are not used or driven.

The cycle is repeated for the other BYTEs of the DWORD if required.

FIGURE 8-41: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT READ



8-BIT MODE POSTED WRITE

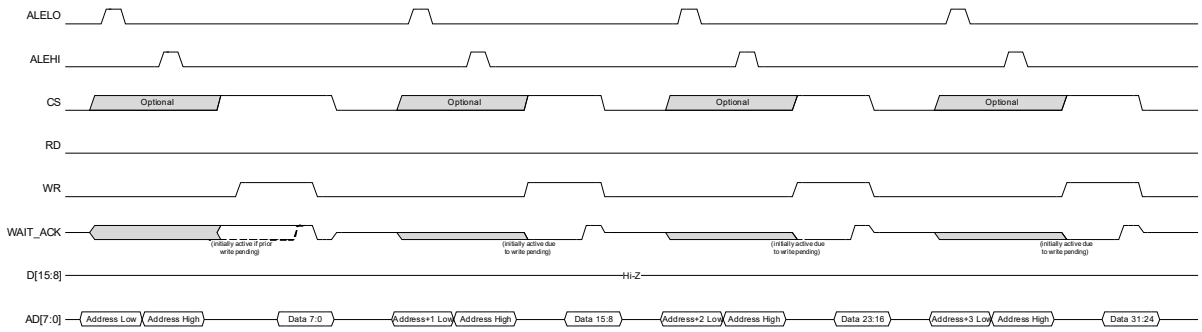
The BYTE address is latched sequentially from **AD[7:0]**.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **WR** inactive. **CS** and **WR** may become inactive simultaneously in which case **WAIT_ACK** active is not seen (second example).

Data on **AD[7:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven.

The cycle is repeated for the other BYTEs of the DWORD if required.

FIGURE 8-42: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT POSTED WRITE



8-BIT MODE NON-POSTED WRITE

The BYTE address is latched sequentially from **AD[7:0]**.

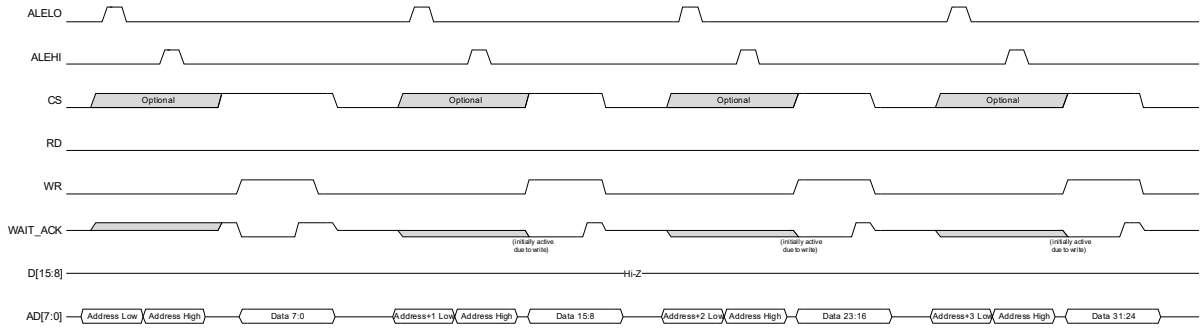
WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven inactive since there will never be a prior pending write. **WAIT_ACK** is driven active upon **WR** active. **CS** and **WR** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

Data on **AD[7:0]** is written starting on the leading edge of **WR**. **WR** may be made inactive once **WAIT_ACK** becomes inactive. **D[15:8]** pins are not used or driven.

For EtherCAT Core registers and the Process RAM, the write cycle occurs during **WR** active. For-non EtherCAT Core registers, the write cycle occurs upon WR inactive. Therefore Data is held until **WR** inactive.

The cycle is repeated for the other BYTEs of the DWORD if required.

FIGURE 8-43: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT NON-POSTED WRITE



8.6.1.2 Single Phase Address Latching

The figures in this section detail read and write operations in multiplexed addressing mode with single phase address latching for 16 and 8-bit modes.

16-BIT MODE READ

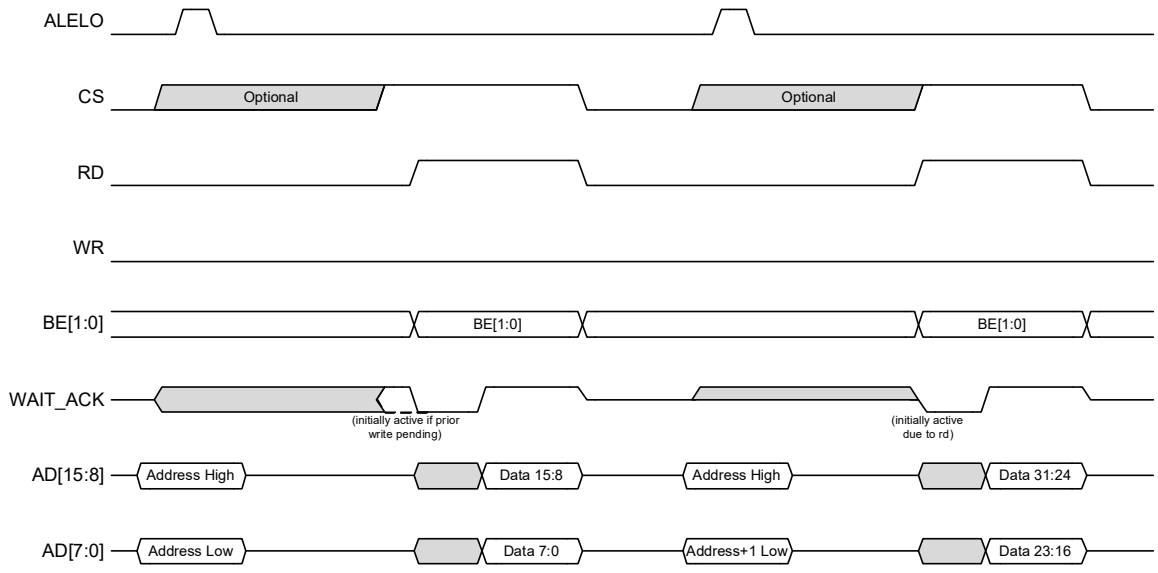
The WORD address is latched simultaneously from **AD[7:0]** and **AD[15:8]**.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **RD** active. **CS** and **RD** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

AD[15:0] are driven active on during **RD** active and valid once **WAIT_ACK** becomes inactive.

The cycle is repeated for the other 16-bits of the DWORD if required.

FIGURE 8-44: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 16-BIT READ



16-BIT MODE POSTED WRITE

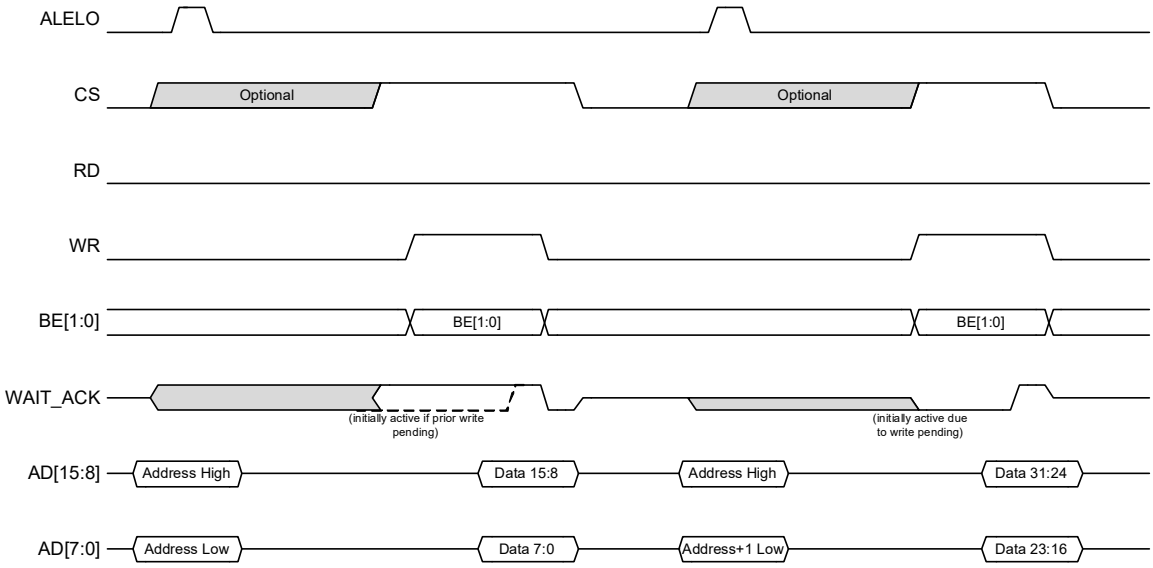
The WORD address is latched simultaneously from **AD[7:0]** and **AD[15:8]**.

WAIT_ACK is driven upon CS active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **WR** inactive. **CS** and **WR** may become inactive simultaneously in which case **WAIT_ACK** active is not seen (second example).

Data on **AD[15:8]** is written on the trailing edge of **WR**.

The cycle is repeated for the other 16-bits of the DWORD if required.

FIGURE 8-45: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 16-BIT POSTED WRITE



16-BIT MODE NON-POSTED WRITE

The WORD address is latched simultaneously from **AD[7:0]** and **AD[15:8]**.

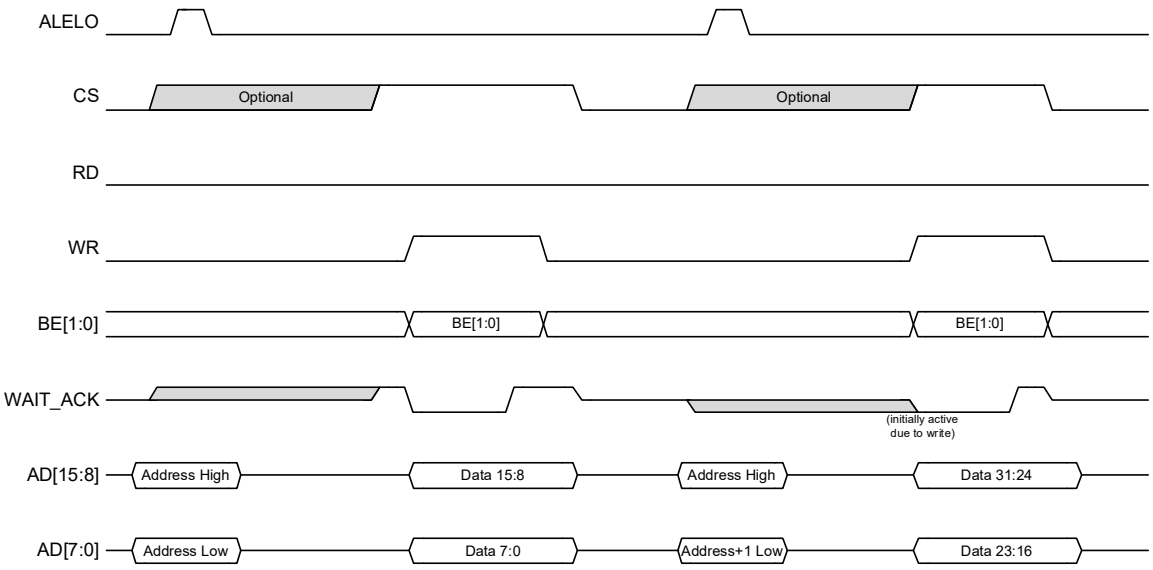
WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven inactive since there will never be a prior pending write. **WAIT_ACK** is driven active upon **WR** active. **CS** and **WR** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

Data on **AD[15:0]** is written starting on the leading edge of **WR**. **WR** may be made inactive once **WAIT_ACK** becomes inactive.

For EtherCAT Core registers and the Process RAM, the write cycle occurs during **WR** active. For-non EtherCAT Core registers, the write cycle occurs upon **WR** inactive. Therefore Data is held until **WR** inactive.

The cycle is repeated for the other 16-bits of the DWORD if required.

FIGURE 8-46: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 16-BIT NON-POSTED WRITE



8-BIT MODE READ

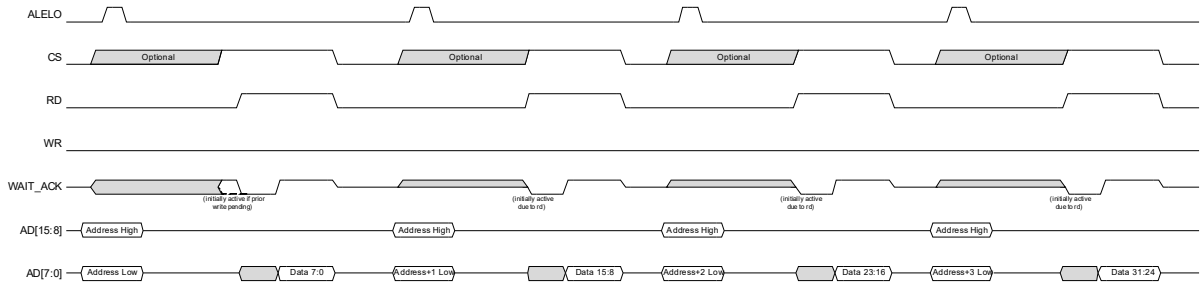
The BYTE address is latched simultaneously from **AD[7:0]** and **AD[15:8]**.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **RD** active. **CS** and **RD** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

AD[7:0] are driven active on during **RD** active and valid once **WAIT_ACK** becomes inactive. **AD[15:8]** pins are not used or driven for the data phase as the host could potentially continue to drive the upper address on these signals.

The cycle is repeated for the other BYTEs of the DWORD if required.

FIGURE 8-47: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 8-BIT READ



8-BIT MODE POSTED WRITE

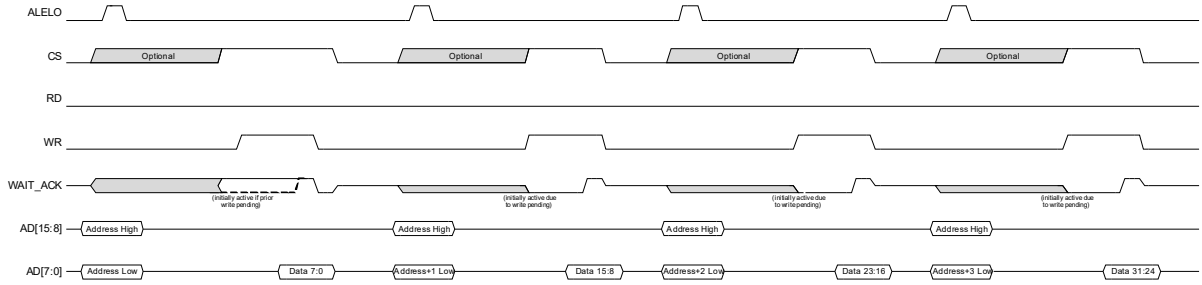
The BYTE address is latched simultaneously from **AD[7:0]** and **AD[15:8]**.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **WR** inactive. **CS** and **WR** may become inactive simultaneously in which case **WAIT_ACK** active is not seen (second example).

Data on **AD[7:0]** is written on the trailing edge of **WR**. **AD[15:8]** pins are not used or driven for the data phase as the host could potentially continue to drive the upper address on these signals.

The cycle is repeated for the other BYTEs of the DWORD if required.

FIGURE 8-48: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 8-BIT POSTED WRITE



8-BIT MODE NON-POSTED WRITE

The BYTE address is latched simultaneously from **AD[7:0]** and **AD[15:8]**.

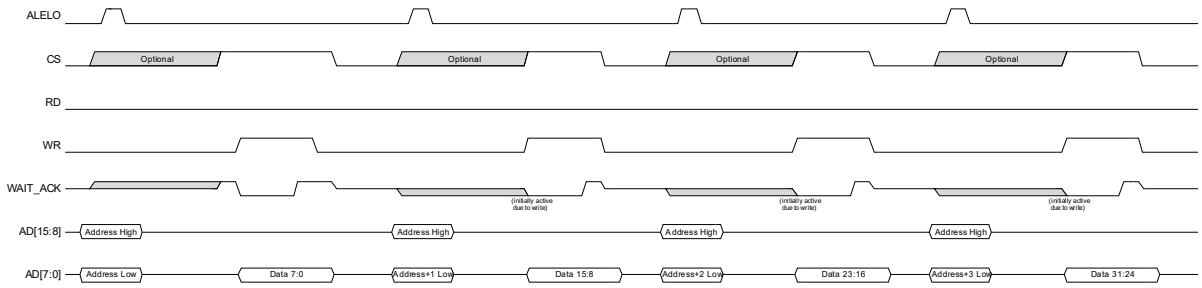
WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven inactive since there will never be a prior pending write. **WAIT_ACK** is driven active upon **WR** active. **CS** and **WR** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

Data on **AD[7:0]** is written starting on the leading edge of **WR**. **WR** may be made inactive once **WAIT_ACK** becomes inactive. **AD[15:8]** pins are not used or driven for the data phase as the host could potentially continue to drive the upper address on these signals.

For EtherCAT Core registers and the Process RAM, the write cycle occurs during **WR** active. For-non EtherCAT Core registers, the write cycle occurs upon **WR** inactive. Therefore Data is held until **WR** inactive.

The cycle is repeated for the other BYTEs of the DWORD if required.

FIGURE 8-49: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 8-BIT NON-POSTED WRITE



8.6.2 DEMULTIPLEXED ADDRESSING MODE FUNCTIONAL TIMING DIAGRAMS

The following timing diagrams illustrate example demultiplexed addressing mode read and write cycles for various configurations and bus sizes. These diagrams do not cover every supported host bus permutation, but are selected to detail the main configuration differences (bus size) within the demultiplexed addressing mode of operation.

The following should be noted for the timing diagrams in this section:

- The diagrams in this section depict active-low push-pull **WAIT_ACK** and active-high **BE1/0**, **CS**, **RD**, and **WR** signals. The polarities of these signals are selectable via the HBI WAIT_ACK Polarity, HBI BE1/BE0 Polarity, HBI Chip Select Polarity, HBI Read, Read/Write Polarity, and HBI Write, Enable Polarity bits of the PDI Configuration Register and Extended PDI Configuration Register (HBI Modes), respectively. Refer to Section 8.2.2, "Control Line Polarity and Buffer Type," on page 69 for additional details.
- The diagrams in this section depict little endian byte ordering. However, dynamic big and little endianness are supported for 16-bit mode via the endianness signal. Endianness changes only the order of the bytes involved, and not the overall timing requirements. Refer to Section 8.4.4, "Host Endianness," on page 79 for additional information.
- The diagrams in this section utilize **RD** and **WR** signals. Alternative **RD_WR** and **ENB** signaling is also supported, similar to the demultiplexed examples in Section 8.5.2.1, "RD_WR / ENB Control Mode Examples". The HBI read/write mode is selectable via the HBI Read/Write Mode bit of the PDI Configuration Register. The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI Read, Read/Write Polarity and HBI Write, Enable Polarity configuration inputs.
- While accessing non-EtherCAT Core registers, for proper DWORD assembly, consecutive address cycles must be within the same DWORD until the DWORD is completely accessed (with the register exceptions noted above). Although BYTEs and WORDs can be accessed in any order, the diagrams in this section depict accessing the lower address BYTE or WORD first.
- While accessing non-EtherCAT Core registers in 16-bit mode, all cycles are 16-bit wide with **BE1/BE0** unused. While accessing EtherCAT Core registers and the Process RAM in 16-bit mode, 8 or 16-bit transfers can be selected via the **BE1/BE0** pins.

16-BIT MODE READ

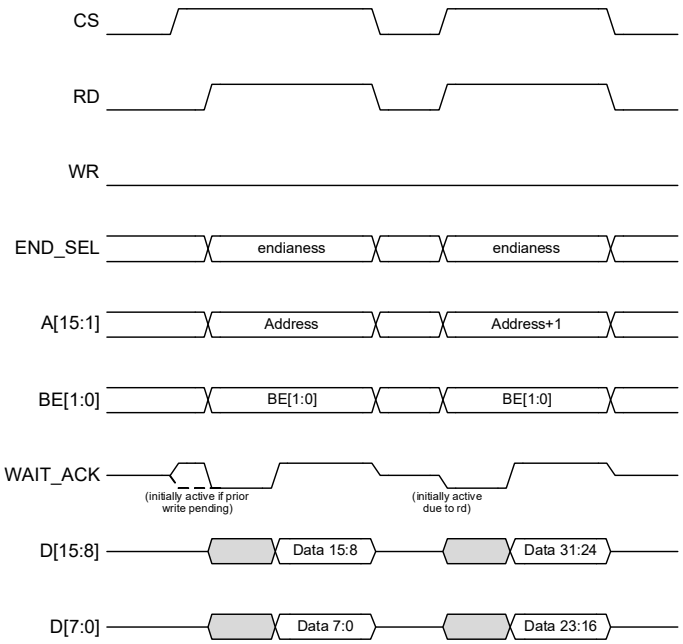
The WORD address, byte enables and endianess is input concurrently with the control.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **RD** active. **CS** and **RD** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

D[15:0] are driven active on during **RD** active and valid once **WAIT_ACK** becomes inactive.

The cycle is repeated for the other 16-bits of the DWORD if required.

FIGURE 8-50: DEMULTIPLEXED ADDRESSING - 16-BIT READ



16-BIT MODE POSTED WRITE

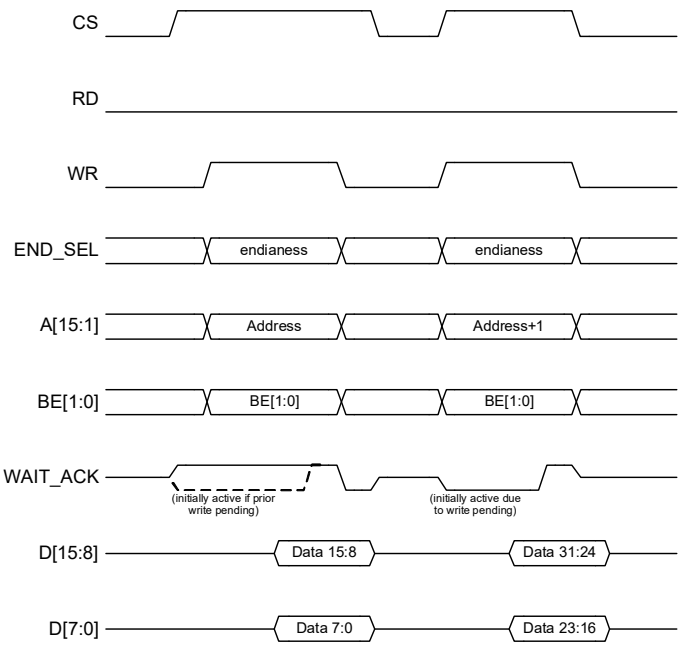
The WORD address, byte enables and endianess is input concurrently with the control.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **WR** inactive. **CS** and **WR** may become inactive simultaneously in which case **WAIT_ACK** active is not seen (second example).

Data on **D[15:0]** is written on the trailing edge of **WR**.

The cycle is repeated for the other 16-bits of the DWORD if required.

FIGURE 8-51: DEMULTIPLEXED ADDRESSING - 16-BIT POSTED WRITE



16-BIT MODE NON-POSTED WRITE

The WORD address, byte enables and endianess is input concurrently with the control.

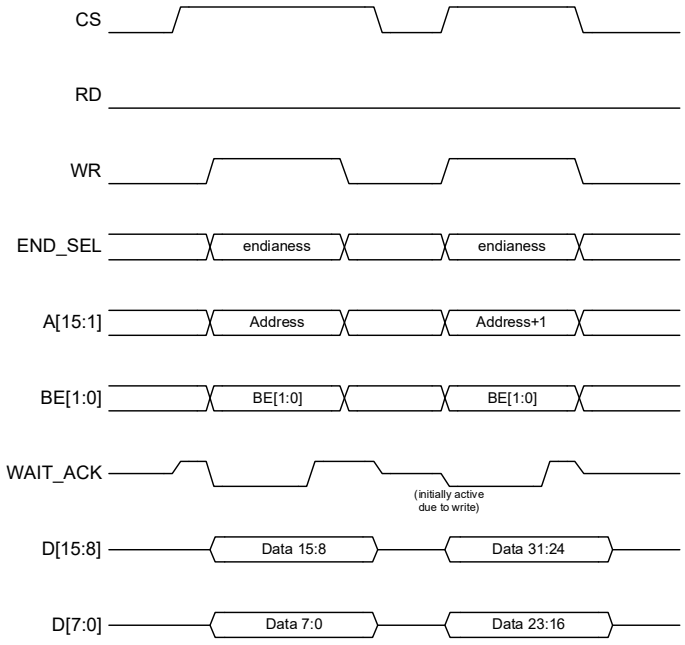
WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven inactive since there will never be a prior pending write. **WAIT_ACK** is driven active upon **WR** active. **CS** and **WR** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

Data on **D[15:0]** is written starting on the leading edge of **WR**. **WR** may be made inactive once **WAIT_ACK** becomes inactive.

For EtherCAT Core registers and the Process RAM, the write cycle occurs during **WR** active. For-non EtherCAT Core registers, the write cycle occurs upon **WR** inactive. Therefore Data is held until WR inactive.

The cycle is repeated for the other 16-bits of the DWORD if required.

FIGURE 8-52: DEMULTIPLEXED ADDRESSING - 16-BIT NON-POSTED WRITE



8-BIT MODE READ

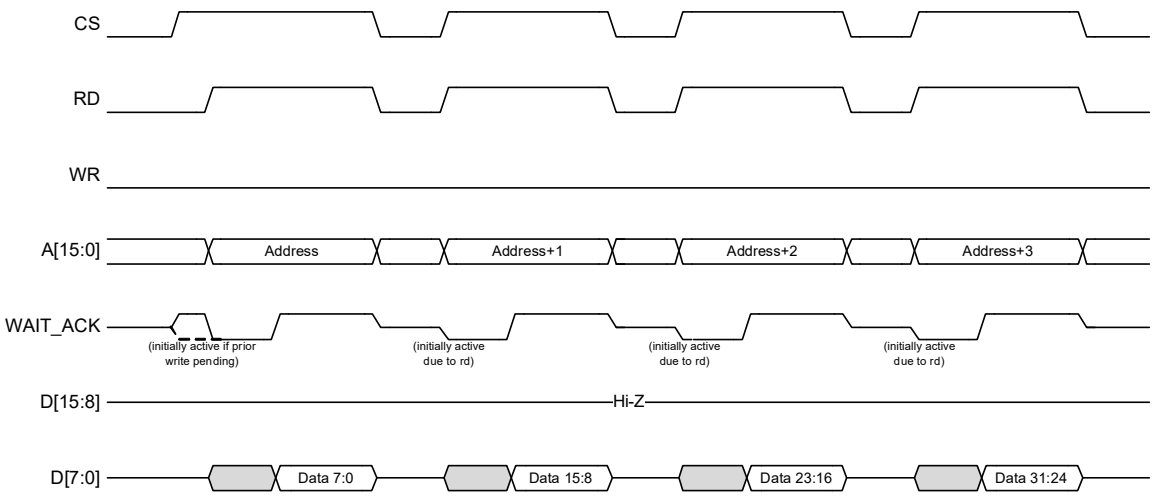
The BYTES address is input concurrently with the control.

WAIT_ACK is driven upon CS active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **RD** active. **CS** and **RD** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

D[7:0] are driven active on during **RD** active and valid once **WAIT_ACK** becomes inactive. **D[15:8]** pins are not used or driven.

The cycle is repeated for the other BYTEs of the DWORD if required.

FIGURE 8-53: DEMULTIPLEXED ADDRESSING - 8-BIT READ



8-BIT MODE POSTED WRITE

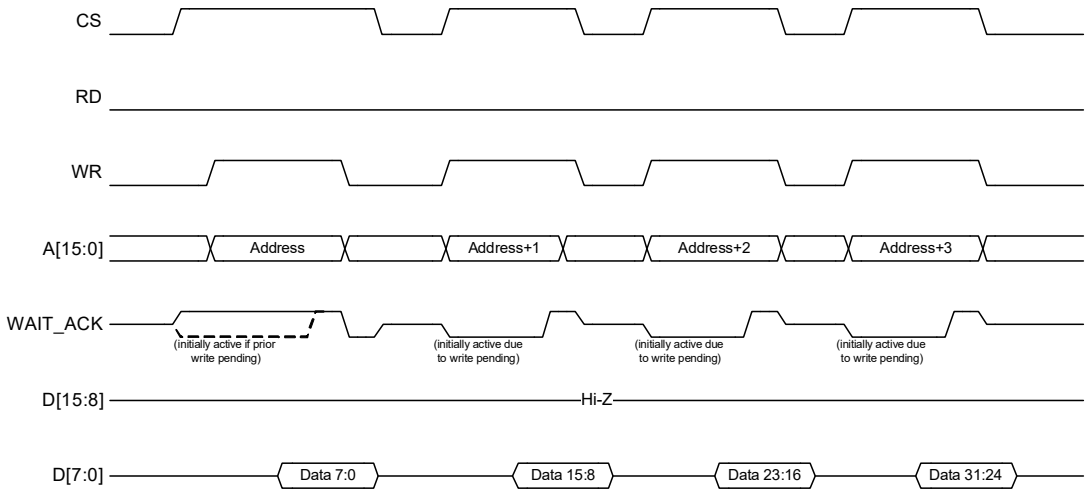
The BYTE address is input concurrently with the control.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **WR** inactive. **CS** and **WR** may become inactive simultaneously in which case **WAIT_ACK** active is not seen (second example).

Data on **D[7:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven.

The cycle is repeated for the other BYTEs of the DWORD if required.

FIGURE 8-54: DEMULTIPLEXED ADDRESSING - 8-BIT POSTED WRITE



8-BIT MODE NON-POSTED WRITE

The BYTE address is input concurrently with the control.

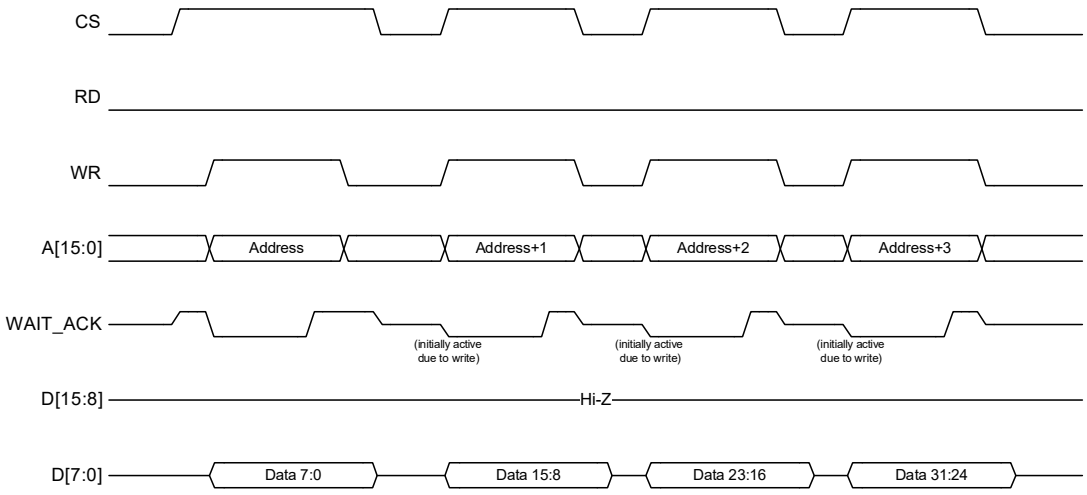
WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven inactive since there will never be a prior pending write. **WAIT_ACK** is driven active upon **WR** active. **CS** and **WR** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

Data on **D[7:0]** is written starting on the leading edge of **WR**. **WR** may be made inactive once **WAIT_ACK** becomes inactive. **D[15:8]** pins are not used or driven.

For EtherCAT Core registers and the Process RAM, the write cycle occurs during WR active. For-non EtherCAT Core registers, the write cycle occurs upon WR inactive. Therefore Data is held until WR inactive.

The cycle is repeated for the other BYTEs of the DWORD if required.

FIGURE 8-55: DEMULTIPLEXED ADDRESSING - 8-BIT NON-POSTED WRITE



8.6.3 INDEXED ADDRESS MODE FUNCTIONAL TIMING DIAGRAMS

The following timing diagrams illustrate example indexed (non-multiplexed) addressing mode read and write cycles for various configurations and bus sizes. These diagrams do not cover every supported host bus permutation, but are selected to detail the main configuration differences (bus size, Configuration / Index / Data cycles) within the indexed addressing mode of operation.

The following should be noted for the timing diagrams in this section:

- The diagrams in this section depict active-low push-pull **WAIT_ACK** and active-high **CS**, **RD**, and **WR** signals. The polarities of these signals are selectable via the HBI WAIT_ACK Polarity, HBI Chip Select Polarity, HBI Read, Read/Write Polarity, and HBI Write, Enable Polarity bits of the PDI Configuration Register (HBI Modes), respectively. Refer to Section 8.2.2, "Control Line Polarity and Buffer Type," on page 69 for additional details.
- The diagrams in this section depict little endian byte ordering. However, configurable big and little endianness are supported for 16-bit mode via the endianness bits in the Host Bus Interface Configuration Register. Endianness changes only the order of the bytes involved, and not the overall timing requirements. Refer to Section 8.4.4, "Host Endianness," on page 79 for additional information.
- The diagrams in this section utilize **RD** and **WR** signals. Alternative **RD_WR** and **ENB** signaling is also supported, similar to the Indexed examples in Section 8.5.3.4, "RD_WR / ENB Control Mode Examples". The HBI read/write mode is selectable via the HBI Read/Write Mode bit of the PDI Configuration Register. The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI Read, Read/Write Polarity and HBI Write, Enable Polarity configuration inputs.
- While accessing non-EtherCAT Core registers, for proper DWORD assembly, consecutive address cycles must be within the same DWORD until the DWORD is completely accessed (some internal registers are excluded from this requirement). Although BYTEs and WORDs can be accessed in any order, the diagrams in this section depict accessing the lower address BYTE or WORD first.
- While accessing index and configuration registers in 16-bit mode, all cycles are 16-bit wide with the internal **BE1**/**BE0** unused. While accessing non-EtherCAT Core registers in 16-bit mode, all cycles are 16-bit wide with the internal **BE1**/**BE0** unused.While accessing EtherCAT Core registers and the Process RAM in 16-bit mode, 8 or 16-bit transfers can be selected via the internal **BE1**/**BE0**.

8.6.3.1 Configuration Register Data Access

The figures in this section detail Configuration register read and write operations in indexed address mode for 16 and 8-bit modes.

16-BIT MODE READ AND WRITE

Posted Writes

For posted writes, the address is set to access the Configuration Register.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **WR** inactive. **CS** and **WR** may become inactive simultaneously in which case **WAIT_ACK** active is not seen (third example).

Data on **D[15:0]** is written on the trailing edge of **WR**.

The cycle repeats for the other WORD of the Configuration Register, if desired by the host.

Non-Posted Writes

For non-posted writes, the address is set to access the Configuration Register.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven inactive since there will never be a prior pending write. **WAIT_ACK** is driven active upon **WR** active. **CS** and **WR** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (fifth example).

Data on **D[15:0]** is written starting on the leading edge of **WR**. **WR** may be made inactive once **WAIT_ACK** becomes inactive. The write cycle occurs upon **WR** inactive, therefore Data is held until **WR** inactive.

The cycle repeats for the other WORD of the Configuration Register, if desired by the host.

Reads

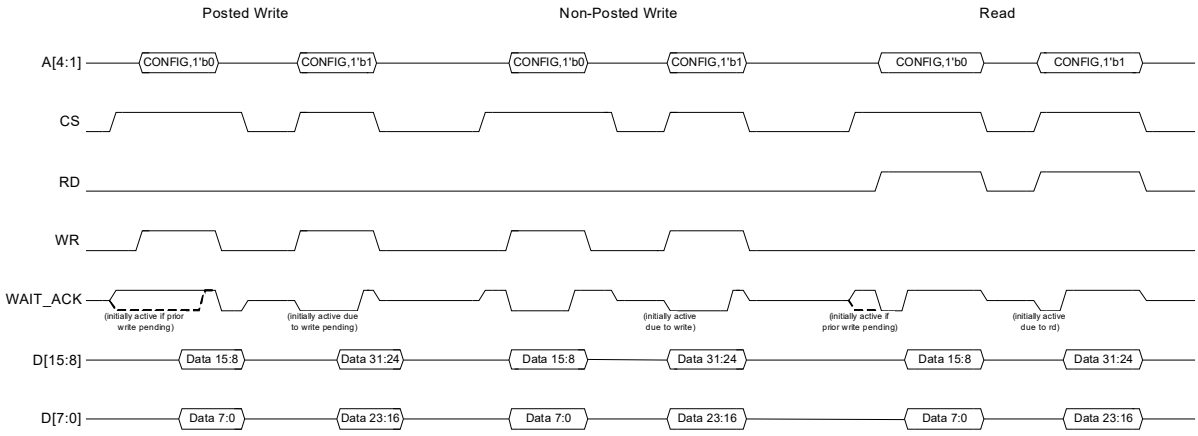
For reads, the address is set to access the Configuration Register.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **RD** active. **CS** and **RD** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (second example).

D[15:0] are driven active on during **RD** active and valid once **WAIT_ACK** becomes inactive.

The cycle repeats for the other WORD of the Configuration Register, if desired by the host.

FIGURE 8-56: INDEXED ADDRESSING CONFIGURATION REGISTER ACCESS - 16-BIT POSTED WRITE / NON-POSTED WRITE / READ



8-BIT MODE READ AND WRITE

Posted Writes

For posted writes, the address is set to access the Configuration Register.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **WR** inactive. **CS** and **WR** may become inactive simultaneously in which case **WAIT_ACK** active is not seen (second example).

Data on **D[7:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven.

The cycle repeats for the remaining BYTEs of the Configuration Register, if desired by the host. The first and last BYTEs are shown here.

Non-Posted Writes

For non-posted writes, the address is set to access the Configuration Register.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven inactive since there will never be a prior pending write. **WAIT_ACK** is driven active upon **WR** active. **CS** and **WR** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (fourth example).

Data on **D[7:0]** is written starting on the leading edge of **WR**. **WR** may be made inactive once **WAIT_ACK** becomes inactive. **D[15:8]** pins are not used or driven. The write cycle occurs upon **WR** inactive, therefore Data is held until **WR** inactive.

The cycle repeats for the remaining BYTEs of the Configuration Register, if desired by the host. The first and last BYTEs are shown here.

Reads

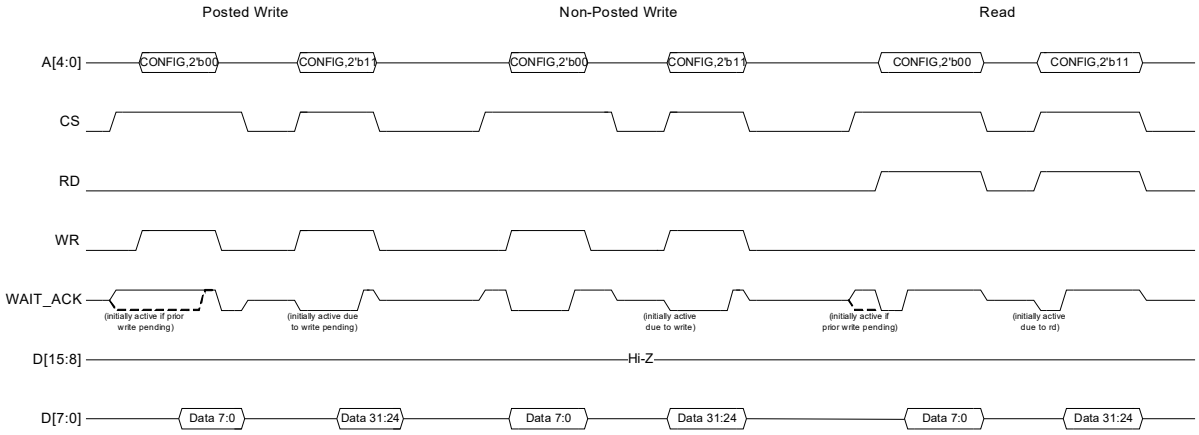
For reads, the address is set to access the Configuration Register.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **RD** active. **CS** and **RD** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (sixth example).

D[7:0] are driven active on during **RD** active and valid once **WAIT_ACK** becomes inactive. **D[15:8]** pins are not used or driven.

The cycle repeats for the remaining BYTEs of the Configuration Register, if desired by the host. The first and last BYTEs are shown here.

FIGURE 8-57: INDEXED ADDRESSING CONFIGURATION REGISTER ACCESS - 8-BIT POSTED WRITE / NON-POSTED WRITE / READ



8.6.3.2 Index Register Data Access

The figures in this section detail index register read and write operations in indexed address mode for 16 and 8-bit modes.

16-BIT MODE READ AND WRITE

Posted Writes

For posted writes, the address is set to access one of the Index Registers.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **WR** inactive. **CS** and **WR** may become inactive simultaneously in which case **WAIT_ACK** active is not seen (second example).

Data on **D[15:0]** is written on the trailing edge of **WR**.

The cycle repeats for the other WORD of the Index Register, if desired by the host.

Non-Posted Writes

For non-posted writes, the address is set to access one of the Index Registers.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven inactive since there will never be a prior pending write. **WAIT_ACK** is driven active upon **WR** active. **CS** and **WR** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (fourth example).

Data on **D[15:0]** is written starting on the leading edge of **WR**. **WR** may be made inactive once **WAIT_ACK** becomes inactive. The write cycle occurs upon WR inactive, therefore Data is held until **WR** inactive.

The cycle repeats for the other WORD of the Index Register, if desired by the host.

Reads

For reads, the address is set to access one of the Index Registers.

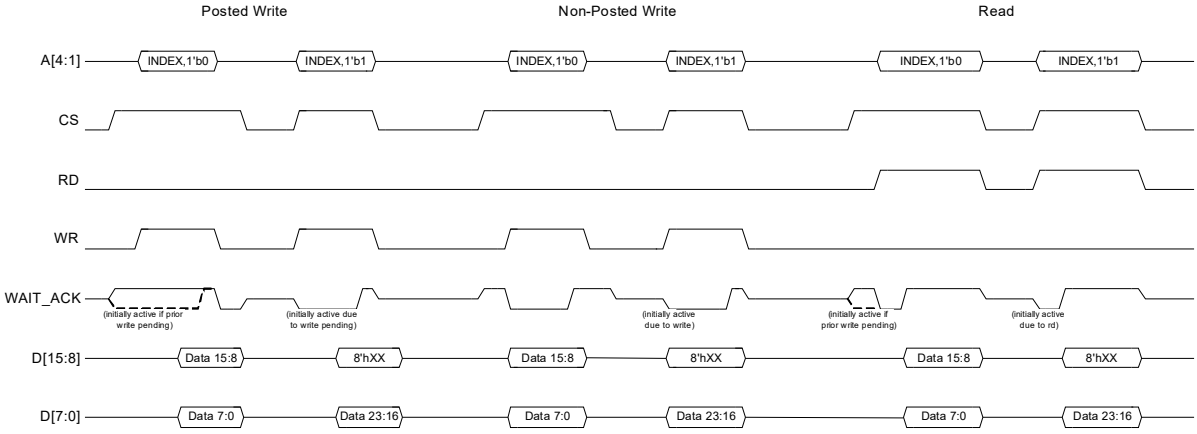
WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **RD** active. **CS** and **RD** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (sixth example).

D[15:0] are driven active on during **RD** active and valid once **WAIT_ACK** becomes inactive.

The cycle repeats for the other WORD of the Index Register, if desired by the host.

Note:	The upper BYTE of Index Register is reserved and don't care.

FIGURE 8-58: INDEXED ADDRESSING INDEX REGISTER ACCESS - 16-BIT POSTED WRITE / NON-POSTED WRITE / READ



8-BIT MODE READ AND WRITE

Posted Writes

For posted writes, the address is set to access one of the Index Registers.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **WR** inactive. **CS** and **WR** may become inactive simultaneously in which case **WAIT_ACK** active is not seen (second example).

Data on **D[7:0]** is written on the trailing edge of WR. **D[15:8]** pins are not used or driven.

The cycle repeats for the remaining BYTEs of the Index Register, if desired by the host. The first and second BYTEs are shown here.

Non-Posted Writes

For non-posted writes, the address is set to access one of the Index Registers.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven inactive since there will never be a prior pending write. **WAIT_ACK** is driven active upon **WR** active. **CS** and **WR** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (fourth example).

Data on **D[7:0]** is written starting on the leading edge of **WR**. **WR** may be made inactive once **WAIT_ACK** becomes inactive. **D[15:8]** pins are not used or driven. The write cycle occurs upon **WR** inactive, therefore Data is held until **WR** inactive.

The cycle repeats for the remaining BYTEs of the Index Register, if desired by the host. The first and second BYTEs are shown here.

Reads

For reads, the address is set to access one of the Index Registers.

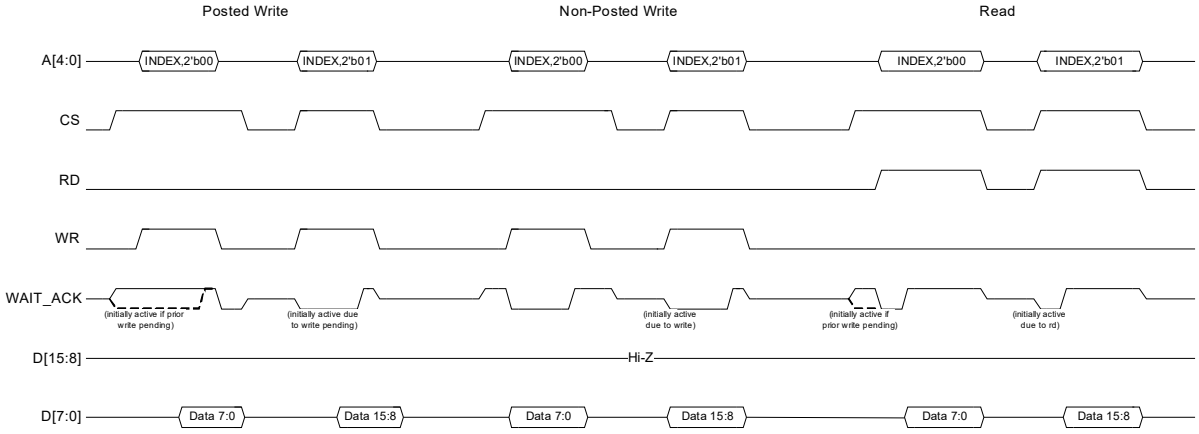
WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **RD** active. **CS** and **RD** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (sixth example).

D[7:0] are driven active on during **RD** active and valid once **WAIT_ACK** becomes inactive. **D[15:8]** pins are not used or driven.

The cycle repeats for the remaining BYTEs of the Index Register, if desired by the host. The first and second BYTEs are shown here.

Note:	The upper BYTE of Index Register is reserved and don't care. Therefore reads and writes to that BYTE is not useful.

FIGURE 8-59: INDEXED ADDRESSING INDEX REGISTER ACCESS - 8-BIT POSTED WRITE / NON-POSTED WRITE / READ



8.6.3.3 Internal Register Data Access

The figures in this section detail typical internal register data read and write cycles in indexed address mode for 16 and 8-bit modes. Although not shown here, a data read or write is usually preceded by an index register write.

16-BIT MODE READ AND WRITE

One of the Index Registers is set as described above.

Posted Writes

For posted writes, the address is then set to access the corresponding Data Register.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **WR** inactive. **CS** and **WR** may become inactive simultaneously in which case **WAIT_ACK** active is not seen (second example).

Data on **D[15:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven.

The cycle repeats for the other WORD of the Data Register, if required.

Non-Posted Writes

For non-posted writes, the address is then set to access the corresponding Data Register.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven inactive since there will never be a prior pending write. **WAIT_ACK** is driven active upon **WR** active. **CS** and **WR** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (fourth example).

Data on **D[15:0]** is written starting on the leading edge of **WR**. **WR** may be made inactive once **WAIT_ACK** becomes inactive. **D[15:8]** pins are not used or driven. The write cycle occurs upon **WR** inactive, therefore Data is held until **WR** inactive.

The cycle repeats for the other WORD of the Data Register, if required.

Reads

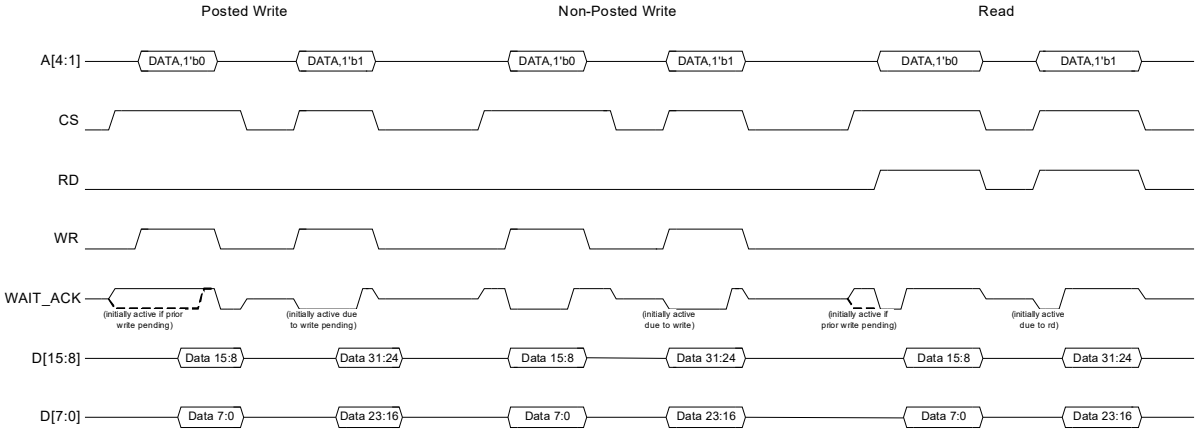
For reads, the address is then set to access the corresponding Data Register.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **RD** active. **CS** and **RD** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (sixth example).

D[15:0] are driven active on during **RD** active and valid once **WAIT_ACK** becomes inactive.

The cycle repeats for the other WORD of the Data Register, if required.

FIGURE 8-60: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 16-BIT POSTED WRITE / NON-POSTED WRITE / READ



8-BIT MODE READ AND WRITE

One of the Index Registers is set as described above.

Posted Writes

For posted writes, the address is then set to access the corresponding Data Register.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **WR** inactive. **CS** and **WR** may become inactive simultaneously in which case **WAIT_ACK** active is not seen (second example).

Data on **D[7:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven.

The cycle repeats for the remaining BYTEs of the Data Register, if required. The first and last BYTEs are shown here.

Non-Posted Writes

For non-posted writes, the address is then set to access the corresponding Data Register.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven inactive since there will never be a prior pending write. **WAIT_ACK** is driven active upon **WR** active. **CS** and **WR** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (fourth example).

Data on **D[7:0]** is written starting on the leading edge of **WR**. **WR** may be made inactive once **WAIT_ACK** becomes inactive. **D[15:8]** pins are not used or driven. The write cycle occurs upon **WR** inactive, therefore Data is held until **WR** inactive.

The cycle repeats for the remaining BYTEs of the Data Register, if required. The first and last BYTEs are shown here.

Reads

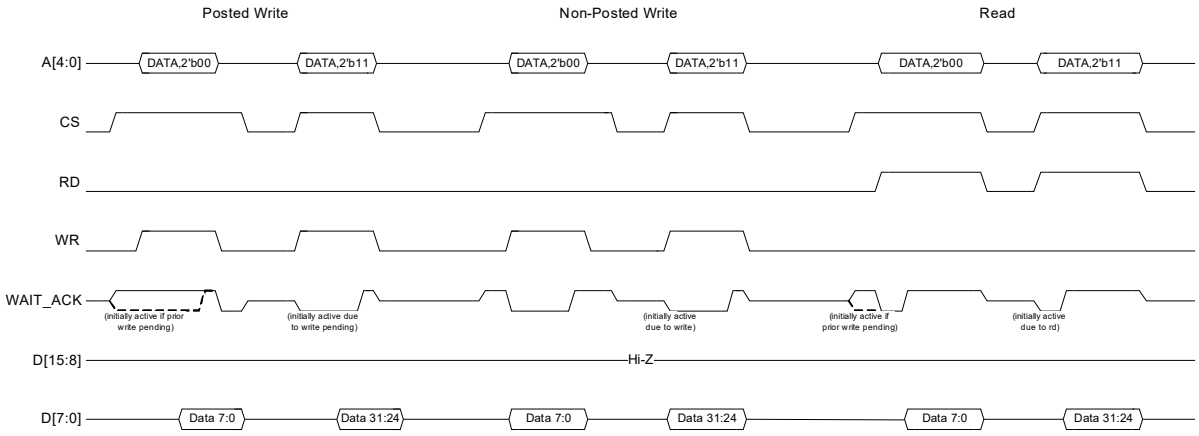
For reads, the address is then set to access the corresponding Data Register.

WAIT_ACK is driven upon **CS** active. **WAIT_ACK** is initially driven active if there was a prior pending write, otherwise it is initially driven inactive. **WAIT_ACK** is driven active upon **RD** active. **CS** and **RD** may become active simultaneously in which case **WAIT_ACK** is always initially driven active (sixth example).

D[7:0] are driven active on during **RD** active and valid once **WAIT_ACK** becomes inactive. **D[15:8]** pins are not used or driven.

The cycle repeats for the remaining BYTEs of the Data Register, if required. The first and last BYTEs are shown here.

FIGURE 8-61: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 8-BIT POSTED WRITE / NON-POSTED WRITE / READ



8.7 Timing Requirements

8.7.1 MULTIPLEXED ADDRESSING MODE TIMING REQUIREMENTS

The following figures and tables specify the timing requirements during Multiplexed Address / Data mode. Since timing requirements are similar across the multitude of operations (e.g. dual vs. single phase, 8 vs. 16-bit), many timing requirements are illustrated onto the same figures and do not necessarily represent any particular functional operation.

The following should be noted for the timing specifications in this section:

- The diagrams in this section depict active-low push-pull **WAIT_ACK** and active-high **ALEHI/ALELO**, **CS**, **RD**, **WR**, **RD_WR** and **ENB** signals. The polarities of these signals are selectable via the HBI WAIT_ACK Polarity, HBI ALE Polarity, HBI Chip Select Polarity, HBI Read, Read/Write Polarity, and HBI Write, Enable Polarity bits of the PDI Configuration Register (HBI Modes), respectively. Refer to Section 8.2.2, "Control Line Polarity and Buffer Type," on page 69 for additional details.
- Qualification of the **ALELO** and/or **ALEHI** with the **CS** signal is selectable via the HBI ALE Qualification bit of the PDI Configuration Register. This is shown as a dashed line. Timing requirements between **ALELO** / **ALEHI** and CS only apply when this mode is active.
- In dual phase address latching mode, the **ALEHI** and **ALELO** cycles can be in any order. **ALEHI** first is depicted in solid line. **ALELO** first is depicted in dashed line.
- A read cycle may be followed by followed by an address cycle, a write cycle or another read cycle. A write cycle may be followed by followed by an address cycle, a read cycle or another write cycle. These are shown in dashed line.

8.7.1.1 Read Timing Requirements - Legacy Mode

The following sections present timing requirements while in LAN9252 compatible Legacy mode (i.e. not in EtherCAT Direct Mapped mode). Byte Enables **BE1/BE0** are not used in Legacy mode and are not shown. If enabled for output, **WAIT_ACK** is held inactive (ACK) and is shown as such.

If **RD** and **WR** signaling is used, a host read cycle begins when **RD** is asserted with **CS** active. The cycle ends when **RD** is deasserted. CS may be asserted and deasserted along with **RD** but not during **RD** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host read cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a read. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.5.1, "Multiplexed Addressing Mode Functional Timing Diagrams," on page 82 for functional descriptions.

FIGURE 8-62: MULTIPLEXED ADDRESSING READ CYCLE TIMING

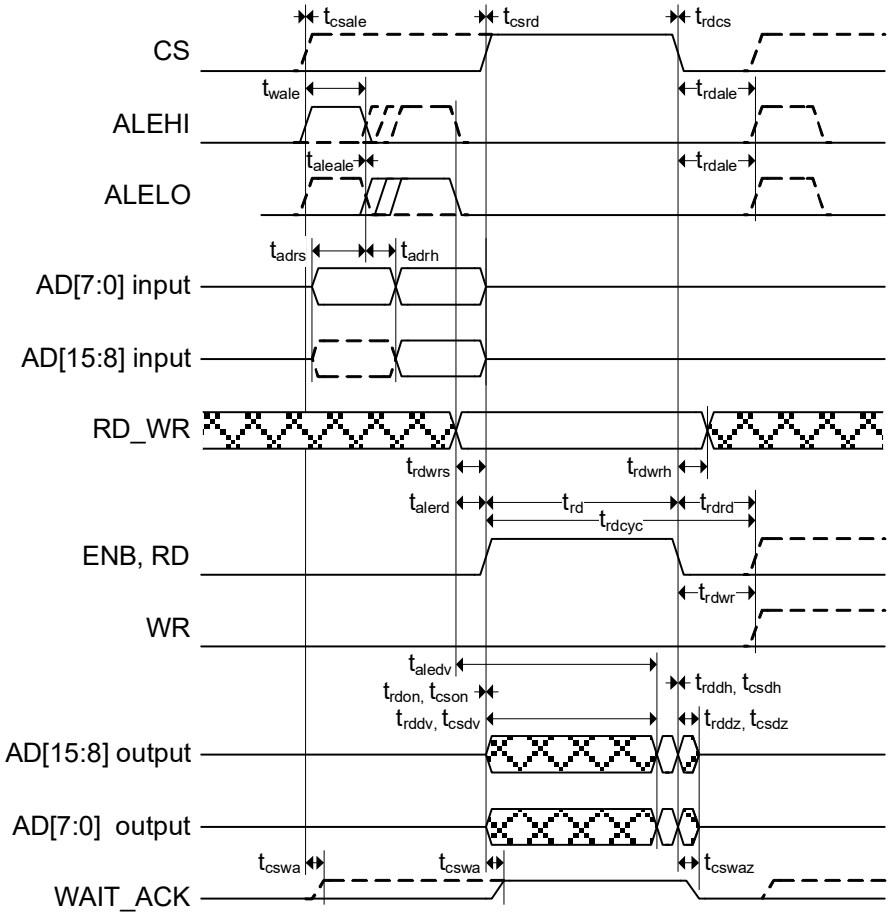


TABLE 8-4: MULTIPLEXED ADDRESSING READ CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{csale}	**CS** Setup to **ALELO**, **ALEHI** Active Note 3, Note 2	0			nS
t_{csrd}	**CS** Setup to **RD** or **ENB** Active	0			nS
t_{rdcs}	**CS** Hold from **RD** or **ENB** Inactive	0			nS
t_{wale}	**ALELO**, **ALEHI** Pulse Width	10			nS
t_{adrs}	Address Setup to **ALELO**, **ALEHI** Inactive	10			nS
t_{adrh}	Address Hold from **ALELO**, **ALEHI** Inactive	5			nS
t_{aleale}	**ALELO** Inactive to **ALEHI** Active **ALEHI** Inactive to **ALELO** Active Note 1, Note 2	0			nS
t_{alerd}	**ALELO**, **ALEHI** Inactive to **RD** or **ENB** Active Note 2	5			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 4	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 4	5			nS
t_{rdon}	**RD** or **ENB** to Data Buffer Turn On	0			nS
t_{rddv}	**RD** or **ENB** Active to Data Valid			30	nS
t_{rddh}	Data Output Hold Time from **RD** or **ENB** Inactive	0			nS
t_{rddz}	Data Buffer Turn Off Time from **RD** or **ENB** Inactive			9	nS
t_{cson}	**CS** to Data Buffer Turn On	0			nS
t_{csdv}	**CS** Active to Data Valid			30	nS
t_{csdh}	Data Output Hold Time from **CS** Inactive	0			nS
t_{csdz}	Data Buffer Turn Off Time from **CS** Inactive			9	nS
t_{aledv}	**ALELO**, **ALEHI** Inactive to Data Valid Note 2			35	nS
t_{rd}	**RD** or **ENB** Active Time	32			nS
t_{rdcyc}	**RD** or **ENB** Cycle Time	45			nS
t_{rdale}	**RD** or **ENB** De-assertion Time before Address Phase	13			nS
t_{rdrd}	**RD** or **ENB** De-assertion Time before Next **RD** or **ENB** Note 5	13			nS
t_{rdwr}	**RD** De-assertion Time before Next **WR** Note 5, Note 6	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 1: Dual Phase Addressing

Note 2: Depends on **ALEHI** / **ALELO** order.

Note 3: **ALELO** and/or **ALEHI** qualified with the **CS**.

Note 4: **RD_WR** and **ENB** signaling.

Note 5: No interposed address phase.

Note 6: **RD** and **WR** signaling.

Note:	Timing values are with respect to an equivalent test load of 25 pF.

8.7.1.2 Read Timing Requirements - EtherCAT Direct Mapped Mode

The following sections present functional timing diagrams while in EtherCAT Direct Mapped mode.

If enabled for output, **WAIT_ACK** is used to indicate when data is valid to be captured, otherwise the host must assume the worst case data access time listed.

WAIT_ACK is output with **CS** active and if not already active due to a pending prior write, becomes active upon **RD** or **ENB**. If **CS** and **RD/ENB** are concurrent, **WAIT_ACK** is driven active.

If **RD** and **WR** signaling is used, a host read cycle begins when **RD** is asserted with **CS** active. The cycle ends when RD is deasserted. **CS** may be asserted and deasserted along with **RD** but not during **RD** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host read cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a read. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.6.1, "Multiplexed Addressing Mode Functional Timing Diagrams," on page 104 for functional descriptions.

FIGURE 8-63: MULTIPLEXED ADDRESSING READ CYCLE TIMING

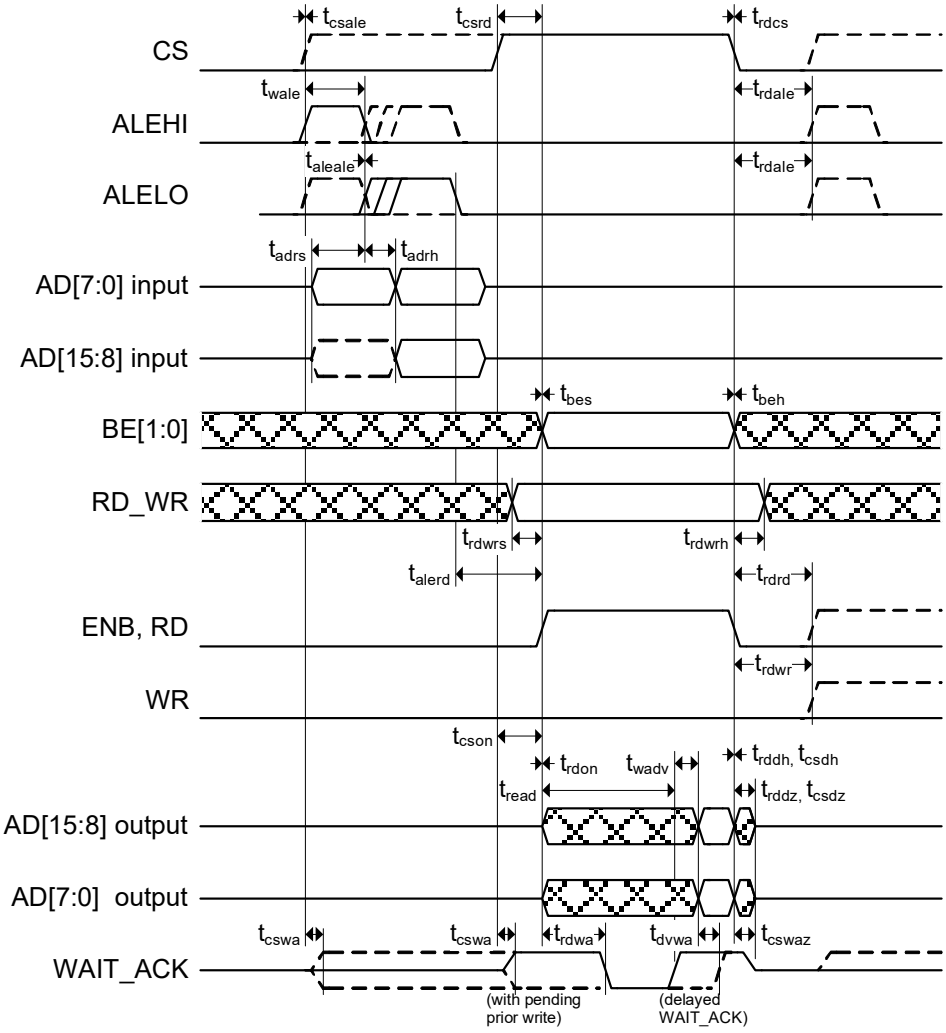


TABLE 8-5: MULTIPLEXED ADDRESSING READ CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{csale}	**CS** Setup to **ALELO**, **ALEHI** Active Note 9, Note 8	0			nS
t_{csrd}	**CS** Setup to **RD** or **ENB** Active	0			nS
t_{rdcs}	**CS** Hold from **RD** or **ENB** Inactive	0			nS
t_{wale}	**ALELO**, **ALEHI** Pulse Width	10			nS
t_{adrs}	Address Setup to **ALELO**, **ALEHI** Inactive	10			nS
t_{adrh}	Address Hold from **ALELO**, **ALEHI** Inactive	5			nS
t_{aleale}	**ALELO** Inactive to **ALEHI** Active **ALEHI** Inactive to **ALELO** Active Note 7, Note 8	0			nS
t_{alerd}	**ALELO**, **ALEHI** Inactive to **RD** or **ENB** Active Note 8	5			nS
t_{bes}	Byte Enable Setup to **RD** or **ENB** Active	0			nS
t_{beh}	Byte Enable Hold from **RD** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to ENB Active Note 10	5			nS
t_{rdwrh}	**RD_WR** Hold from ENB Inactive Note 10	5			nS
t_{rdon}	**RD** or **ENB** to Data Buffer Turn On	0			nS
t_{rddh}	Data Output Hold Time from **RD** or **ENB** Inactive	0			nS
t_{rddz}	Data Buffer Turn Off Time from **RD** or **ENB** Inactive			9	nS
t_{cson}	**CS** to Data Buffer Turn On	0			nS
t_{csdh}	Data Output Hold Time from **CS** Inactive	0			nS
t_{csdz}	Data Buffer Turn Off Time from **CS** Inactive			9	nS
t_{rdale}	**RD** or **ENB** De-assertion Time before Address Phase	13			nS
t_{rdrd}	**RD** or **ENB** De-assertion Time before Next **RD** or **ENB** Note 10	13			nS
t_{rdwr}	**RD** De-assertion Time before Next **WR** Note 11, Note 12	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{rdwa}	**RD** or **ENB** Active to **WAIT_ACK** Active			10	nS

TABLE 8-5: MULTIPLEXED ADDRESSING READ CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{read}	**RD** or **ENB** Active to **WAIT_ACK** Inactive - 8-bit read - no pending prior write Note 13			235	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 8-bit read - 8-bit pending prior write Note 13			435	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 8-bit read - 16-bit pending prior write Note 13			495	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 16-bit read - no pending prior write Note 13			315	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 16-bit read - 8-bit pending prior write Note 13			515	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 16-bit read - 16-bit pending prior write Note 13			575	nS
t_{wadv}	**WAIT_ACK** Inactive to Data Valid - normal **WAIT_ACK**			5	nS
t_{dvwa}	Data Valid Before **WAIT_ACK** Inactive - delayed **WAIT_ACK**	15			nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 7: Dual Phase Addressing

Note 8: Depends on **ALEHI** / **ALELO** order.

Note 9: **ALELO** and/or **ALEHI** qualified with the **CS**.

Note 10: **RD_WR** and **ENB** signaling.

Note 11: No interposed address phase.

Note 12: **RD** and **WR** signaling.

Note 13: Add 20nS if delayed **WAIT_ACK** is enabled.

Note: Timing values are with respect to an equivalent test load of 25 pF.

8.7.1.3 Write Timing Requirements - Legacy Mode

The following sections present timing requirements while in LAN9252 compatible Legacy mode (i.e. not in EtherCAT Direct Mapped mode). Byte Enables **BE1/BE0** are not used in Legacy mode and are not shown. If enabled for output, **WAIT_ACK** is held inactive (ACK) and is shown as such.

If **RD** and **WR** signaling is used, a host write cycle begins when **WR** is asserted with **CS** active. The cycle ends when **WR** is deasserted. **CS** may be asserted and deasserted along with **WR** but not during **WR** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host write cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a write. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.5.1, "Multiplexed Addressing Mode Functional Timing Diagrams," on page 82 for functional descriptions.

FIGURE 8-64: MULTIPLEXED ADDRESSING WRITE CYCLE TIMING

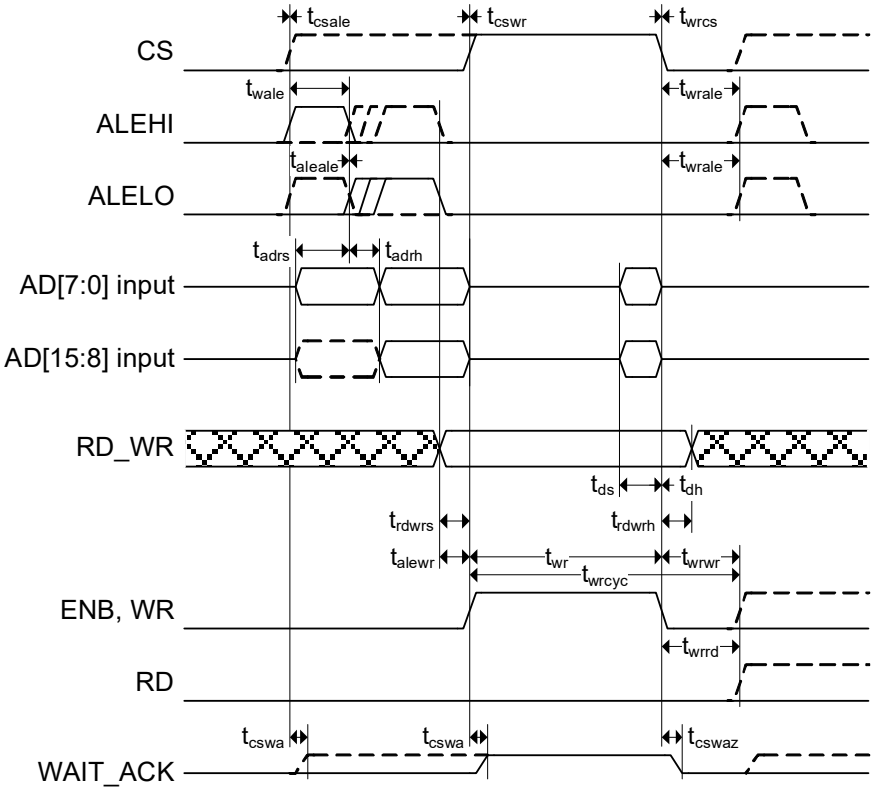


TABLE 8-6: MULTIPLEXED ADDRESSING WRITE CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{csale}	**CS** Setup to **ALELO**, **ALEHI** Active Note 16, Note 15	0			nS
t_{cswr}	**CS** Setup to **WR** or **ENB** Active	0			nS
t_{wrcs}	**CS** Hold from **WR** or **ENB** Inactive	0			nS
t_{wale}	**ALELO**, **ALEHI** Pulse Width	10			nS
t_{adrs}	Address Setup to **ALELO**, **ALEHI** Inactive	10			nS
t_{adrh}	Address Hold from **ALELO**, **ALEHI** Inactive	5			nS
t_{aleale}	**ALELO** Inactive to **ALEHI** Active **ALEHI** Inactive to **ALELO** Active Note 14, Note 15	0			nS
t_{alewr}	**ALELO**, **ALEHI** Inactive to **WR** or **ENB** Active Note 15	5			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 17	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 17	5			nS
t_{ds}	Data Setup to **WR** or **ENB** Inactive	10			nS
t_{dh}	Data Hold from **WR** or **ENB** Inactive	0			nS
t_{wr}	**WR** or **ENB** Active Time	32			nS
t_{wrcyc}	**WR** or **ENB** Cycle Time	45			nS
t_{wrale}	**WR** or **ENB** De-assertion Time before Address Phase	13			nS
t_{wrwr}	**WR** or **ENB** De-assertion Time before Next **WR** or **ENB** Note 18	13			nS
t_{wrrd}	**WR** De-assertion Time before Next **RD** Note 18, Note 19	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 14: Dual Phase Addressing

Note 15: Depends on **ALEHI** / **ALELO** order.

Note 16: **ALELO** and/or **ALEHI** qualified with the **CS**.

Note 17: **RD_WR** and **ENB** signaling.

Note 18: No interposed address phase.

Note 19: **RD** and **WR** signaling.

8.7.1.4 Posted Write Timing Requirements - EtherCAT Direct Mapped Mode

The following sections present functional timing diagrams while in EtherCAT Direct Mapped mode.

If enabled for output, **WAIT_ACK** is used to indicate when the write cycle may be concluded, otherwise the host must assume the worst case write access time listed.

WAIT_ACK is output with **CS** active and will be active if there was a pending prior write. Otherwise it will be inactive. The write strobe (**WR** or **ENB**) may be made active. Once **WAIT_ACK** is inactive the write strobe may be made inactive. **WAIT_ACK** becomes active upon an inactive **WR** or **ENB**. If **CS** and **WD/ENB** become inactive concurrently, **WAIT_ACK** is released and the active state may not be seen.

If **RD** and **WR** signaling is used, a host write cycle begins when **WR** is asserted with **CS** active. The cycle ends when **WR** is deasserted. **CS** may be asserted and deasserted along with **WR** but not during **WR** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host write cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a write. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.6.1, "Multiplexed Addressing Mode Functional Timing Diagrams," on page 104 for functional descriptions.

FIGURE 8-65: MULTIPLEXED ADDRESSING POSTED WRITE CYCLE TIMING

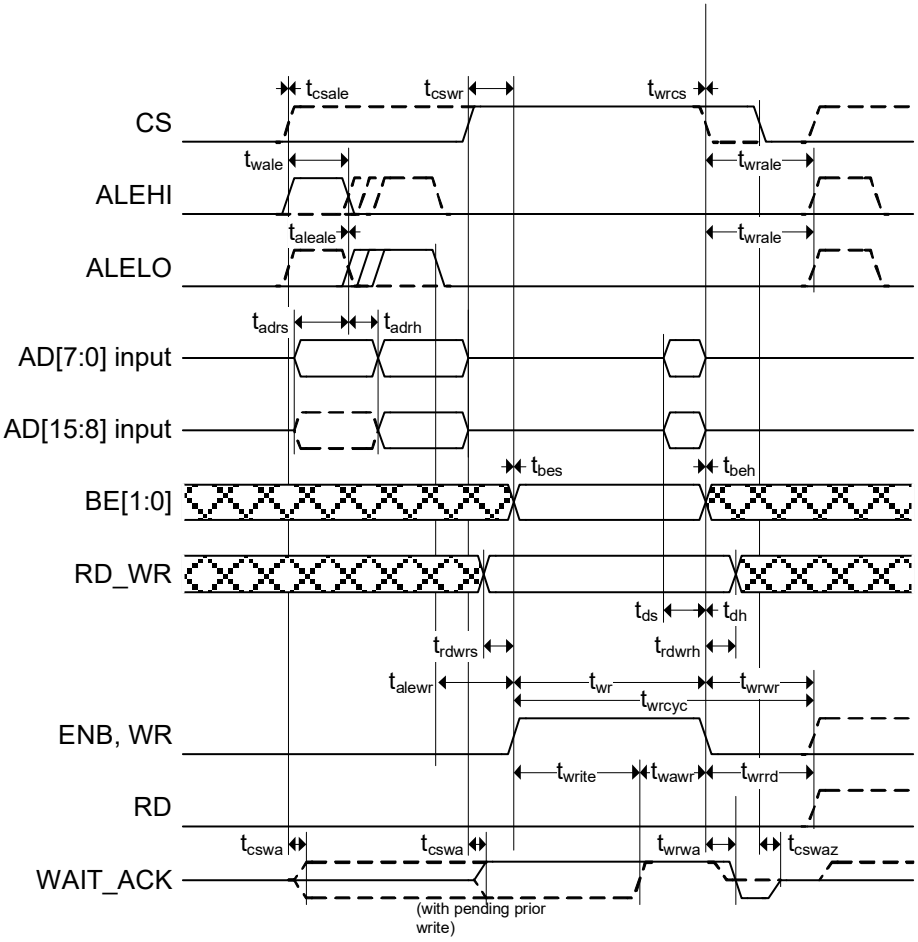


TABLE 8-7: MULTIPLEXED ADDRESSING POSTED WRITE CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{csale}	**CS** Setup to **ALELO**, **ALEHI** Active Note 22, Note 21	0			nS
t_{cswr}	**CS** Setup to **WR** or **ENB** Active	0			nS
t_{wrcs}	**CS** Hold from **WR** or **ENB** Inactive	0			nS
t_{wale}	**ALELO**, **ALEHI** Pulse Width	10			nS
t_{adrs}	Address Setup to **ALELO**, **ALEHI** Inactive	10			nS
t_{adrh}	Address Hold from **ALELO**, **ALEHI** Inactive	5			nS
t_{aleale}	**ALELO** Inactive to **ALEHI** Active **ALEHI** Inactive to **ALELO** Active Note 20, Note 21	0			nS
t_{alewr}	**ALELO**, **ALEHI** Inactive to **WR** or **ENB** Active Note 21	5			nS
t_{bes}	Byte Enable Setup to **WR** or **ENB** Active	0			nS
t_{beh}	Byte Enable Hold from **WR** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 23	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 23	5			nS
t_{ds}	Data Setup to **WR** or **ENB** Inactive	10			nS
t_{dh}	Data Hold from **WR** or **ENB** Inactive	0			nS
t_{wr}	**WR** or **ENB** Active Time - no pending prior write	32			nS
t_{wrcyc}	**WR** or **ENB** Cycle Time - no pending prior write	45			nS
t_{wrale}	**WR** or **ENB** De-assertion Time before Address Phase	13			nS
t_{wrwr}	**WR** or **ENB** De-assertion Time before Next **WR** or **ENB** Note 24	13			nS
t_{wrrd}	**WR** De-assertion Time before Next **RD** Note 24, Note 25	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{write}	**WR** or **ENB** Active to **WAIT_ACK** Inactive - pending prior 8-bit write			200	nS
	WR or **ENB** Active to **WAIT_ACK** Inactive - pending prior 16-bit write			280	nS
t_{wawr}	**WAIT_ACK** Inactive to **WR** or **ENB** Inactive	0			nS
t_{wrwa}	**WR** or **ENB** Inactive to **WAIT_ACK** Active Note 26			10	nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 20: Dual Phase Addressing

Note 21: Depends on ALEHI / ALELO order.

Note 22: ALELO and/or ALEHI qualified with the CS.

Note 23: RD_WR and ENB signaling.

Note 24: No interposed address phase.

Note 25: RD and WR signaling.

Note 26: If not three-stated first.

8.7.1.5 Non-Posted Write Timing Requirements - EtherCAT Direct Mapped Mode

The following sections present functional timing diagrams while in EtherCAT Direct Mapped mode.

If enabled for output, **WAIT_ACK** is used to indicate when the write cycle may be concluded, otherwise the host must assume the worst case write access time listed.

WAIT_ACK is output with **CS** active. It will be initially inactive and becomes active with write strobe (**WR** or **ENB**) active. Once **WAIT_ACK** is inactive the write strobe may be made inactive.

If **RD** and **WR** signaling is used, a host write cycle begins when **WR** is asserted with **CS** active. The cycle ends when **WR** is deasserted. **CS** may be asserted and deasserted along with **WR** but not during **WR** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host write cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a write. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.6.1, "Multiplexed Addressing Mode Functional Timing Diagrams," on page 104 for functional descriptions.

FIGURE 8-66: MULTIPLEXED ADDRESSING NON-POSTED WRITE CYCLE TIMING

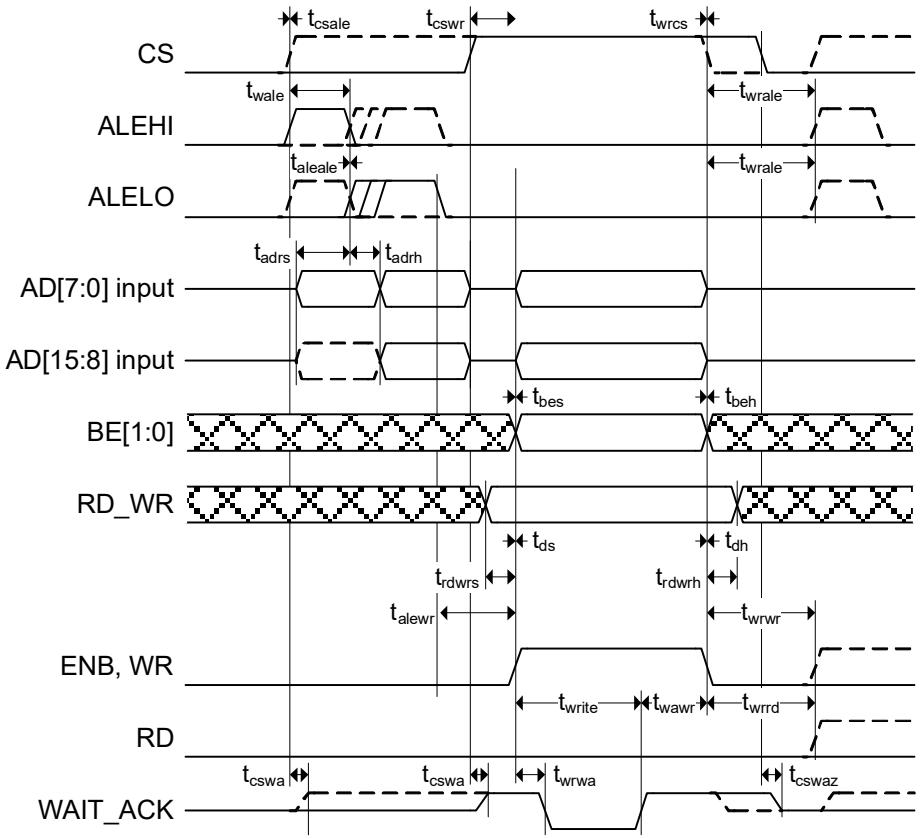


TABLE 8-8: MULTIPLEXED ADDRESSING NON-POSTED WRITE CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{csale}	**CS** Setup to **ALELO**, **ALEHI** Active Note 29, Note 28	0			nS
t_{cswr}	**CS** Setup to **WR** or **ENB** Active	0			nS
t_{wrcs}	**CS** Hold from **WR** or **ENB** Inactive	0			nS
t_{wale}	**ALELO**, **ALEHI** Pulse Width	10			nS
t_{adrs}	Address Setup to **ALELO**, **ALEHI** Inactive	10			nS
t_{adrh}	Address Hold from **ALELO**, **ALEHI** Inactive	5			nS
t_{aleale}	**ALELO** Inactive to **ALEHI** Active **ALEHI** Inactive to **ALELO** Active Note 27, Note 28	0			nS
t_{alewr}	**ALELO**, **ALEHI** Inactive to **WR** or **ENB** Active Note 28	5			nS
t_{bes}	Byte Enable Setup to **WR** or **ENB** Active	0			nS
t_{beh}	Byte Enable Hold from **WR** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 30	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 30	5			nS
t_{ds}	Data Setup to **WR** or **ENB** Active	0			nS
t_{dh}	Data Hold from **WR** or **ENB** Inactive	0			nS
t_{wrale}	**WR** or **ENB** De-assertion Time before Address Phase	13			nS
t_{wrwr}	**WR** or **ENB** De-assertion Time before Next **WR** or **ENB** Note 31	13			nS
t_{wrrd}	**WR** De-assertion Time before Next **RD** Note 31, Note 32	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{wrwa}	**WR** or **ENB** Active to **WAIT_ACK** Active			10	nS
t_{write}	**WR** or **ENB** Active to **WAIT_ACK** Inactive - 8-bit write			200	nS
	WR or **ENB** Active to **WAIT_ACK** Inactive - 16-bit write			280	nS
t_{wawr}	**WAIT_ACK** Inactive to **WR** or **ENB** Inactive	0			nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 27: Dual Phase Addressing

Note 28: Depends on **ALEHI** / **ALELO** order.

Note 29: **ALELO** and/or **ALEHI** qualified with the **CS**.

Note 30: **RD_WR** and **ENB** signaling.

Note 31: No interposed address phase.

Note 32: **RD** and **WR** signaling.

8.7.2 DEMULTIPLEXED ADDRESSING MODE TIMING REQUIREMENTS

The following figures and tables specify the timing requirements during Demultiplexed Address mode. Since timing requirements are similar across the multitude of operations (8 vs. 16-bit), many timing requirements are illustrated onto the same figures and do not necessarily represent any particular functional operation.

The following should be noted for the timing specifications in this section:

- The diagrams in this section depict active-low push-pull **WAIT_ACK** and active-high **CS**, **RD**, and **WR** signals. The polarities of these signals are selectable via the HBI WAIT_ACK Polarity, HBI Chip Select Polarity, HBI Read, Read/Write Polarity, and HBI Write, Enable Polarity bits of the PDI Configuration Register (HBI Modes), respectively. Refer to Section 8.2.2, "Control Line Polarity and Buffer Type," on page 69 for additional details.
- A read cycle may be followed by followed by a write cycle or another read cycle. A write cycle may be followed by a read cycle or another write cycle. These are shown in dashed line.

8.7.2.1 Read Timing Requirements - Legacy Mode

The following sections present timing requirements while in LAN9252 compatible Legacy mode (i.e. not in EtherCAT Direct Mapped mode). Byte Enables **BE1/BE0** are not used in Legacy mode and are not shown. If enabled for output, **WAIT_ACK** is held inactive (ACK) and is shown as such.

If **RD** and **WR** signaling is used, a host read cycle begins when **RD** is asserted with **CS** active. The cycle ends when **RD** is deasserted. **CS** may be asserted and deasserted along with **RD** but not during **RD** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host read cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a read. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.5.2, "Demultiplexed Addressing Mode Functional Timing Diagrams," on page 93 for functional descriptions.

FIGURE 8-67: DEMULTIPLEXED ADDRESSING READ CYCLE TIMING

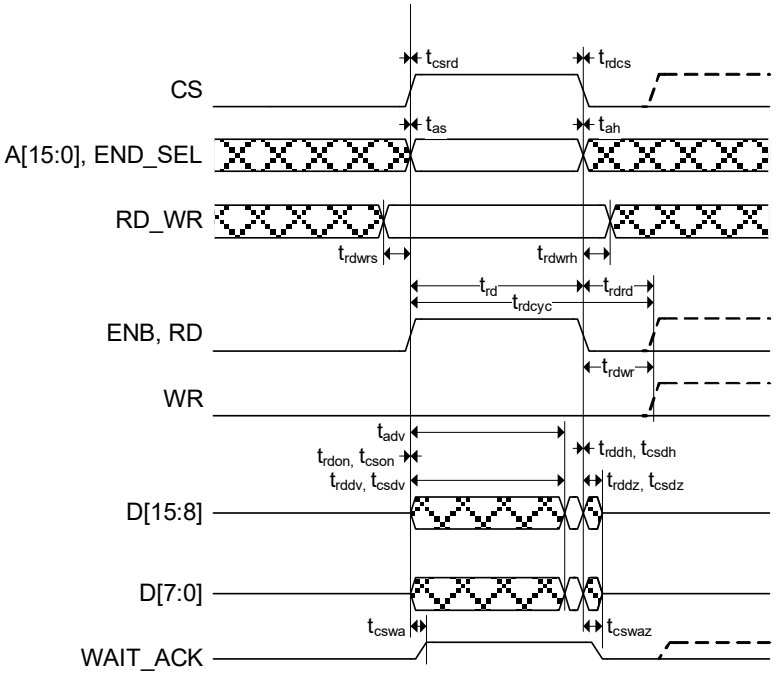


TABLE 8-9: DEMULTIPLEXED ADDRESSING READ CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{csrd}	**CS** Setup to **RD** or **ENB** Active	0			nS
t_{rdcs}	**CS** Hold from **RD** or **ENB** Inactive	0			nS
t_{as}	Address, **END_SEL** Setup to **RD** or **ENB** Active	0			nS
t_{ah}	Address, **END_SEL** Hold from **RD** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 33	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 33	5			nS
t_{rdon}	**RD** or **ENB** to Data Buffer Turn On	0			nS
t_{rddv}	**RD** or **ENB** Active to Data Valid			30	nS
t_{rddh}	Data Output Hold Time from **RD** or **ENB** Inactive	0			nS
t_{rddz}	Data Buffer Turn Off Time from **RD** or **ENB** Inactive			9	nS
t_{cson}	**CS** to Data Buffer Turn On	0			nS
t_{csdv}	**CS** Active to Data Valid			30	nS
t_{csdh}	Data Output Hold Time from **CS** Inactive	0			nS
t_{csdz}	Data Buffer Turn Off Time from **CS** Inactive			9	nS
t_{adv}	Address, **END_SEL** to Data Valid			30	nS
t_{rd}	**RD** or **ENB** Active Time	32			nS
t_{rdcyc}	**RD** or **ENB** Cycle Time	45			nS
t_{rdrd}	**RD** or **ENB** De-assertion Time before Next **RD** or **ENB**	13			nS
t_{rdwr}	**RD** De-assertion Time before Next WR Note 34	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 33: **RD_WR** and **ENB** signaling.

Note 34: **RD** and **WR** signaling.

Note: Timing values are with respect to an equivalent test load of 25 pF.

8.7.2.2 Read Timing Requirements - EtherCAT Direct Mapped Mode

The following sections present functional timing diagrams while in EtherCAT Direct Mapped mode.

If enabled for output, **WAIT_ACK** is used to indicate when data is valid to be captured, otherwise the host must assume the worst case data access time listed.

WAIT_ACK is output with **CS** active and if not already active due to a pending prior write, becomes active upon **RD** or **ENB**. If **CS** and **RD/ENB** are concurrent, **WAIT_ACK** is driven active.

If **RD** and **WR** signaling is used, a host read cycle begins when **RD** is asserted with **CS** active. The cycle ends when **RD** is deasserted. **CS** may be asserted and deasserted along with **RD** but not during **RD** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host read cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a read. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.6.2, "Demultiplexed Addressing Mode Functional Timing Diagrams," on page 115 for functional descriptions.

FIGURE 8-68: DEMULTIPLEXED ADDRESSING READ CYCLE TIMING

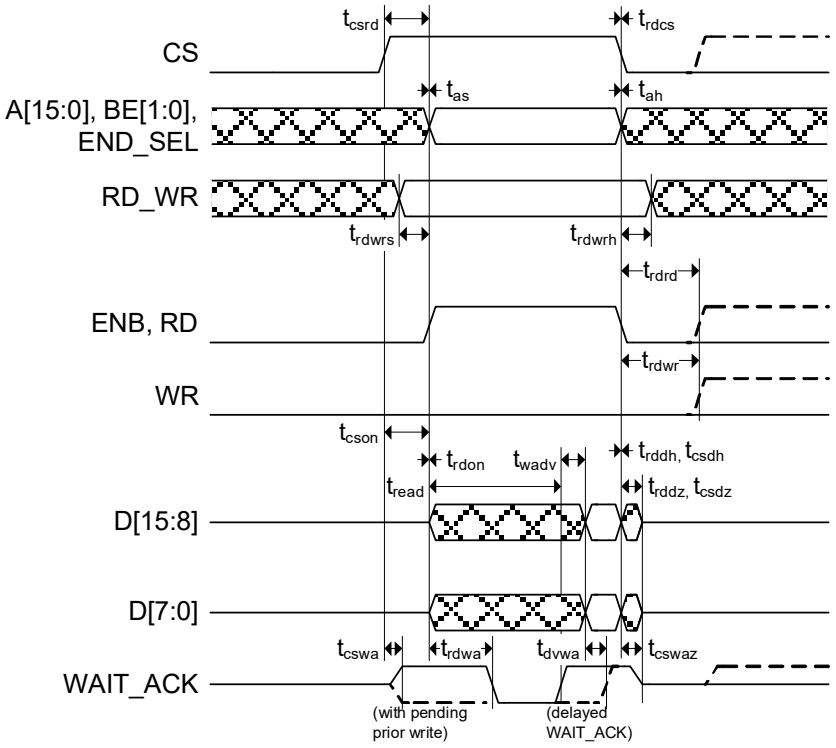


TABLE 8-10: DEMULTIPLEXED ADDRESSING READ CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{csrd}	**CS** Setup to **RD** or **ENB** Active	0			nS
t_{rdcs}	**CS** Hold from **RD** or **ENB** Inactive	0			nS
t_{as}	Address, Byte Enable, **END_SEL** Setup to **RD** or **ENB** Active	0			nS
t_{ah}	Address, Byte Enable, **END_SEL** Hold from **RD** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 35	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 35	5			nS
t_{rdon}	**RD** or **ENB** to Data Buffer Turn On	0			nS
t_{rddh}	Data Output Hold Time from **RD** or **ENB** Inactive	0			nS
t_{rddz}	Data Buffer Turn Off Time from **RD** or **ENB** Inactive			9	nS
t_{cson}	**CS** to Data Buffer Turn On	0			nS
t_{csdh}	Data Output Hold Time from **CS** Inactive	0			nS
t_{csdz}	Data Buffer Turn Off Time from **CS** Inactive			9	nS
t_{rdrd}	**RD** or **ENB** De-assertion Time before Next **RD** or **ENB**	13			nS
t_{rdwr}	**RD** De-assertion Time before Next **WR** Note 36	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{rdwa}	**RD** or **ENB** Active to **WAIT_ACK** Active			10	nS

TABLE 8-10: DEMULTIPLEXED ADDRESSING READ CYCLE TIMING VALUES (CONTINUED)

t_{read}	**RD** or **ENB** Active to **WAIT_ACK** Inactive - 8-bit read - no pending prior write Note 37			235	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 8-bit read - 8-bit pending prior write Note 37			435	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 8-bit read - 16-bit pending prior write Note 37			495	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 16-bit read - no pending prior write Note 37			315	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 16-bit read - 8-bit pending prior write Note 37			515	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 16-bit read - 16-bit pending prior write Note 37			575	nS
t_{wadv}	**WAIT_ACK** Inactive to Data Valid - normal **WAIT_ACK**			5	nS
t_{dvwa}	Data Valid Before **WAIT_ACK** Inactive - delayed **WAIT_ACK**	15			nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 35: **RD_WR** and **ENB** signaling.

Note 36: **RD** and **WR** signaling.

Note 37: Add 20nS if delayed **WAIT_ACK** is enabled.

Note: Timing values are with respect to an equivalent test load of 25 pF.

8.7.2.3 Write Timing Requirements - Legacy Mode

The following sections present timing requirements while in LAN9252 compatible Legacy mode (i.e. not in EtherCAT Direct Mapped mode). Byte Enables **BE1/BE0** are not used in Legacy mode and are not shown. If enabled for output, **WAIT_ACK** is held inactive (ACK) and is shown as such.

If **RD** and **WR** signaling is used, a host write cycle begins when **WR** is asserted with **CS** active. The cycle ends when **WR** is deasserted. **CS** may be asserted and deasserted along with **WR** but not during **WR** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host write cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a write. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.5.2, "Demultiplexed Addressing Mode Functional Timing Diagrams," on page 93 for functional descriptions.

FIGURE 8-69: DEMULTIPLEXED ADDRESSING WRITE CYCLE TIMING

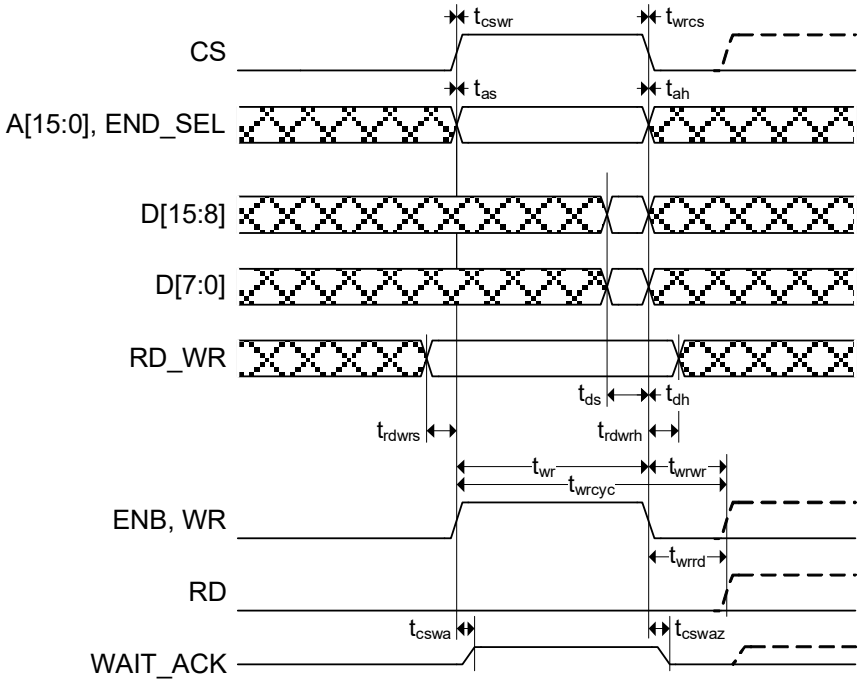


TABLE 8-11: DEMULTIPLEXED ADDRESSING WRITE CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{cswr}	**CS** Setup to **WR** or **ENB** Active	0			nS
t_{wrcs}	**CS** Hold from **WR** or **ENB** Inactive	0			nS
t_{as}	Address, **END_SEL** Setup to **WR** or **ENB** Active	0			nS
t_{ah}	Address, **END_SEL** Hold from **WR** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 38	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 38	5			nS
t_{ds}	Data Setup to **WR** or **ENB** Inactive	10			nS
t_{dh}	Data Hold from **WR** or **ENB** Inactive	0			nS
t_{wr}	**WR** or **ENB** Active Time	32			nS
t_{wrcyc}	**WR** or **ENB** Cycle Time	45			nS
t_{wrwr}	**WR** or **ENB** De-assertion Time before Next **WR** or **ENB**	13			nS
t_{wrrd}	**WR** De-assertion Time before Next **RD** Note 39	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 38: **RD_WR** and **ENB** signaling.

Note 39: **RD** and **WR** signaling.

8.7.2.4 Posted Write Timing Requirements - EtherCAT Direct Mapped Mode

The following sections present functional timing diagrams while in EtherCAT Direct Mapped mode.

If enabled for output, **WAIT_ACK** is used to indicate when the write cycle may be concluded, otherwise the host must assume the worst case write access time listed.

WAIT_ACK is output with **CS** active and will be active if there was a pending prior write. Otherwise it will be inactive. The write strobe (**WR** or **ENB**) may be made active. Once **WAIT_ACK** is inactive the write strobe may be made inactive. **WAIT_ACK** becomes active upon an inactive **WR** or **ENB**. If **CS** and **WD/ENB** become inactive concurrently, **WAIT_ACK** is released and the active state may not be seen.

If **RD** and **WR** signaling is used, a host write cycle begins when **WR** is asserted with **CS** active. The cycle ends when **WR** is deasserted. **CS** may be asserted and deasserted along with **WR** but not during **WR** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host write cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a write. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.6.2, "Demultiplexed Addressing Mode Functional Timing Diagrams," on page 115 for functional descriptions.

FIGURE 8-70: DEMULTIPLEXED ADDRESSING POSTED WRITE CYCLE TIMING

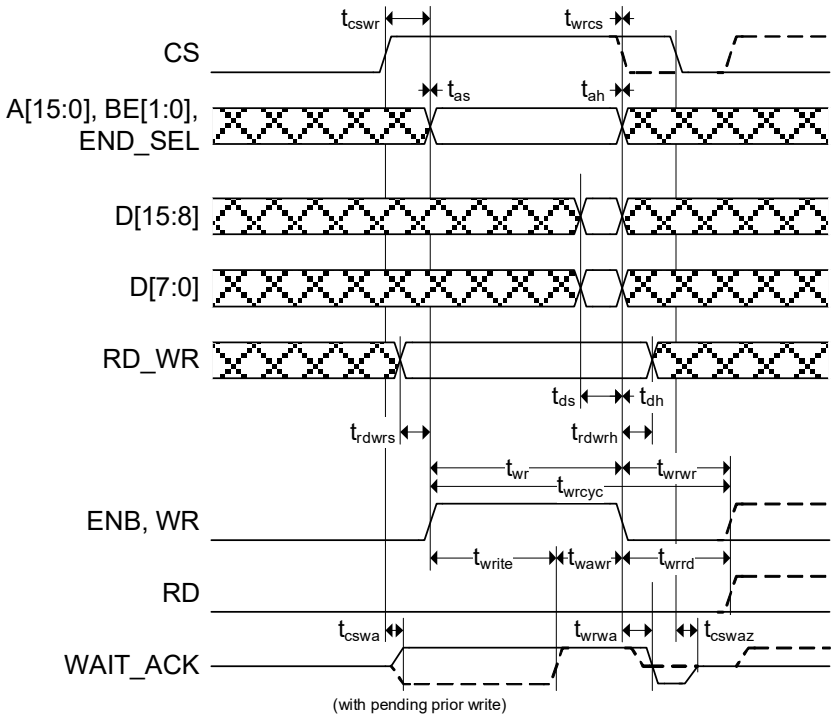


TABLE 8-12: DEMULTIPLEXED ADDRESSING POSTED WRITE CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{cswr}	**CS** Setup to **WR** or **ENB** Active	0			nS
t_{wrcs}	**CS** Hold from **WR** or **ENB** Inactive	0			nS
t_{as}	Address, Byte Enable, **END_SEL** Setup to **WR** or **ENB** Active	0			nS
t_{ah}	Address, Byte Enable, **END_SEL** Hold from **WR** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 40	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 40	5			nS
t_{ds}	Data Setup to **WR** or **ENB** Inactive	10			nS
t_{dh}	Data Hold from **WR** or **ENB** Inactive	0			nS
t_{wr}	**WR** or **ENB** Active Time - no pending prior write	32			nS
t_{wrcyc}	**WR** or **ENB** Cycle Time - no pending prior write	45			nS
t_{wrwr}	**WR** or **ENB** De-assertion Time before Next **WR** or **ENB**	13			nS
t_{wrrd}	**WR** De-assertion Time before Next **RD** Note 41	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{write}	**WR** or **ENB** Active to **WAIT_ACK** Inactive - pending prior 8-bit write			200	nS
	WR or **ENB** Active to **WAIT_ACK** Inactive - pending prior 16-bit write			280	nS
t_{wawr}	**WAIT_ACK** Inactive to **WR** or **ENB** Inactive	0			nS
t_{wrwa}	**WR** or **ENB** Inactive to **WAIT_ACK** Active Note 42			10	nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 40: **RD_WR** and **ENB** signaling.

Note 41: **RD** and **WR** signaling.

Note 42: If not three-stated first.

8.7.2.5 Non-Posted Write Timing Requirements - EtherCAT Direct Mapped Mode

The following sections present functional timing diagrams while in EtherCAT Direct Mapped mode.

If enabled for output, **WAIT_ACK** is used to indicate when the write cycle may be concluded, otherwise the host must assume the worst case write access time listed.

WAIT_ACK is output with **CS** active. It will be initially inactive and becomes active with write strobe (**WR** or **ENB**) active. Once **WAIT_ACK** is inactive the write strobe may be made inactive.

If **RD** and **WR** signaling is used, a host write cycle begins when **WR** is asserted with **CS** active. The cycle ends when **WR** is deasserted. **CS** may be asserted and deasserted along with **WR** but not during WR active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host write cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a write. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.6.2, "Demultiplexed Addressing Mode Functional Timing Diagrams," on page 115 for functional descriptions.

FIGURE 8-71: DEMULTIPLEXED ADDRESSING NON-POSTED WRITE CYCLE TIMING

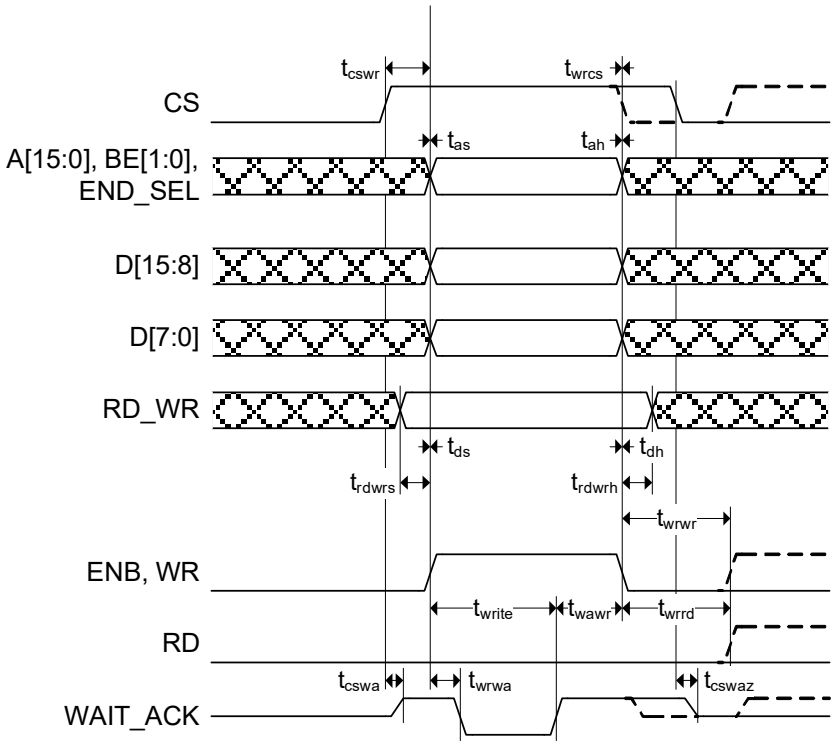


TABLE 8-13: DEMULTIPLEXED ADDRESSING NON-POSTED WRITE CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{cswr}	**CS** Setup to **WR** or **ENB** Active	0			nS
t_{wrcs}	**CS** Hold from **WR** or **ENB** Inactive	0			nS
t_{as}	Address, Byte Enable, **END_SEL** Setup to **WR** or **ENB** Active	0			nS
t_{ah}	Address, Byte Enable, **END_SEL** Hold from **WR** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 43	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 43	5			nS
t_{ds}	Data Setup to **WR** or **ENB** Active	0			nS
t_{dh}	Data Hold from **WR** or **ENB** Inactive	0			nS
t_{wrwr}	**WR** or **ENB** De-assertion Time before Next **WR** or **ENB**	13			nS
t_{wrrd}	**WR** De-assertion Time before Next **RD** Note 44	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{wrwa}	**WR** or **ENB** Active to **WAIT_ACK** Active			10	nS
t_{write}	**WR** or **ENB** Active to **WAIT_ACK** Inactive - 8-bit write			200	nS
	WR or **ENB** Active to **WAIT_ACK** Inactive - 16-bit write			280	nS
t_{wawr}	**WAIT_ACK** Inactive to **WR** or **ENB** Inactive	0			nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 43: **RD_WR** and **ENB** signaling.

Note 44: **RD** and **WR** signaling.

8.7.3 INDEXED ADDRESSING MODE TIMING REQUIREMENTS

The following figures and tables specify the timing requirements during Indexed Address mode. Since timing requirements are similar across the multitude of operations (e.g. 8 vs. 16-bit, Index vs. Configuration vs. Data registers, Index Register Bypass FIFO Access), many timing requirements are illustrated onto the same figures and do not necessarily represent any particular functional operation.

The following should be noted for the timing specifications in this section:

- The diagrams in this section depict active-low push-pull **WAIT_ACK** and active-high **CS**, **RD**, **WR**, **RD_WR**, and **ENB** signals. The polarities of these signals are selectable via the HBI WAIT_ACK Polarity, HBI Chip Select Polarity, HBI Read, Read/Write Polarity, and HBI Write, Enable Polarity bits of the PDI Configuration Register (HBI Modes), respectively. Refer to Section 8.2.2, "Control Line Polarity and Buffer Type," on page 69 for additional details.
- A read cycle may be followed by followed by a write cycle or another read cycle. A write cycle may be followed by a read cycle or another write cycle. These are shown in dashed line.

8.7.3.1 Read Timing Requirements - Legacy Mode

The following sections present timing requirements while in LAN9252 compatible Legacy mode (i.e., not in EtherCAT Direct Mapped mode). If enabled for output, **WAIT_ACK** is held inactive (ACK) and is shown as such.

If **RD** and **WR** signaling is used, a host read cycle begins when **RD** is asserted with **CS** active. The cycle ends when **RD** is deasserted. **CS** may be asserted and deasserted along with **RD** but not during **RD** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host read cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a read. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.5.3, "Indexed Address Mode Functional Timing Diagrams," on page 97 for functional descriptions.

FIGURE 8-72: INDEXED ADDRESSING READ CYCLE TIMING

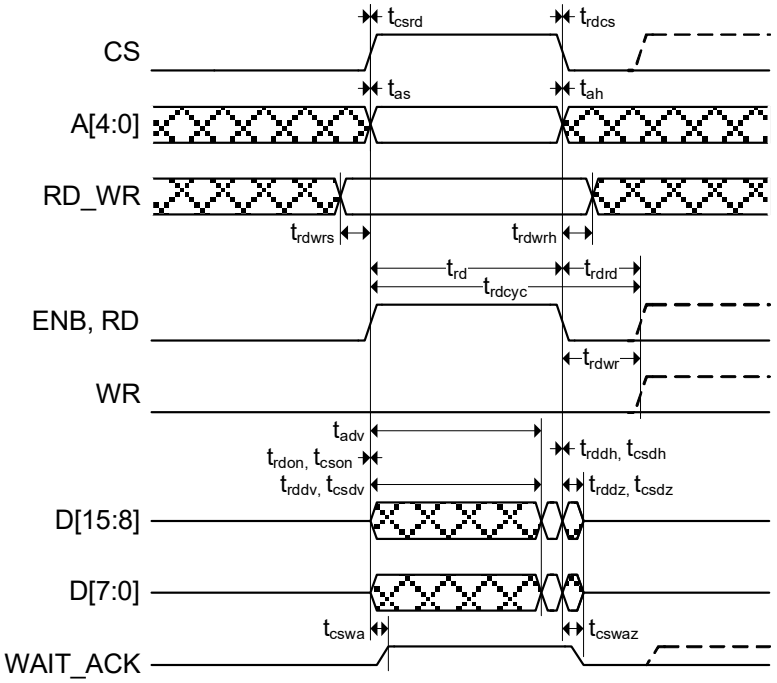


TABLE 8-14: INDEXED ADDRESSING READ CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{csrd}	**CS** Setup to **RD** or **ENB** Active	0			nS
t_{rdcs}	**CS** Hold from **RD** or **ENB** Inactive	0			nS
t_{as}	Address Setup to **RD** or **ENB** Active	0			nS
t_{ah}	Address Hold from **RD** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 45	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 45	5			nS
t_{rdon}	**RD** or **ENB** to Data Buffer Turn On	0			nS
t_{rddv}	**RD** or **ENB** Active to Data Valid			30	nS
t_{rddh}	Data Output Hold Time from **RD** or **ENB** Inactive	0			nS
t_{rddz}	Data Buffer Turn Off Time from **RD** or **ENB** Inactive			9	nS
t_{cson}	CS to Data Buffer Turn On	0			nS
t_{csdv}	CS Active to Data Valid			30	nS
t_{csdh}	Data Output Hold Time from **CS** Inactive	0			nS
t_{csdz}	Data Buffer Turn Off Time from **CS** Inactive			9	nS
t_{adv}	Address to Data Valid			30	nS
t_{rd}	**RD** or **ENB** Active Time	32			nS
t_{rdcyc}	**RD** or **ENB** Cycle Time	45			nS
t_{rdrd}	**RD** or **ENB** De-assertion Time before Next **RD** or **ENB**	13			nS
t_{rdwr}	**RD** De-assertion Time before Next **WR** Note 46	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 45: RD_WR and ENB signaling.

Note 46: RD and WR signaling.

Note: Timing values are with respect to an equivalent test load of 25 pF.

8.7.3.2 Read Timing Requirements - EtherCAT Direct Mapped Mode

The following sections present functional timing diagrams while in EtherCAT Direct Mapped mode.

If enabled for output, **WAIT_ACK** is used to indicate when data is valid to be captured, otherwise the host must assume the worst case data access time listed.

WAIT_ACK is output with **CS** active and if not already active due to a pending prior write, becomes active upon **RD** or **ENB**. If **CS** and **RD/ENB** are concurrent, **WAIT_ACK** is driven active.

If **RD** and **WR** signaling is used, a host read cycle begins when **RD** is asserted with **CS** active. The cycle ends when **RD** is deasserted. **CS** may be asserted and deasserted along with **RD** but not during **RD** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host read cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a read. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.6.3, "Indexed Address Mode Functional Timing Diagrams," on page 122 for functional descriptions.

FIGURE 8-73: INDEXED ADDRESSING READ CYCLE TIMING

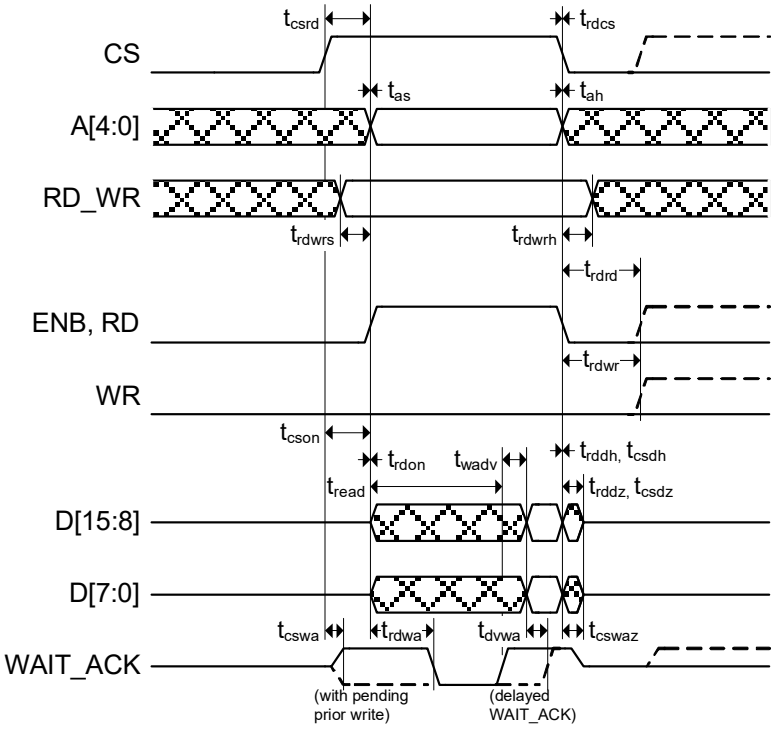


TABLE 8-15: INDEXED ADDRESSING READ CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{csrd}	**CS** Setup to **RD** or **ENB** Active	0			nS
t_{rdcs}	**CS** Hold from **RD** or **ENB** Inactive	0			nS
t_{as}	Address, **END_SEL** Setup to **RD** or **ENB** Active	0			nS
t_{ah}	Address, **END_SEL** Hold from **RD** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 47	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 47	5			nS
t_{rdon}	**RD** or **ENB** to Data Buffer Turn On	0			nS
t_{rddh}	Data Output Hold Time from **RD** or **ENB** Inactive	0			nS
t_{rddz}	Data Buffer Turn Off Time from **RD** or **ENB** Inactive			9	nS
t_{cson}	**CS** to Data Buffer Turn On	0			nS
t_{csdh}	Data Output Hold Time from **CS** Inactive	0			nS
t_{csdz}	Data Buffer Turn Off Time from **CS** Inactive			9	nS
t_{rdrd}	**RD** or **ENB** De-assertion Time before Next **RD** or **ENB**	13			nS
t_{rdwr}	**RD** De-assertion Time before Next **WR** Note 48	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{rdwa}	**RD** or **ENB** Active to **WAIT_ACK** Active			10	nS

TABLE 8-15: INDEXED ADDRESSING READ CYCLE TIMING VALUES (CONTINUED)

t_{read}	**RD** or **ENB** Active to **WAIT_ACK** Inactive - 8-bit read - no pending prior write Note 49			235	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 8-bit read - 8-bit pending prior write Note 49			435	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 8-bit read - 16-bit pending prior write Note 49			495	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 16-bit read - no pending prior write Note 49			315	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 16-bit read - 8-bit pending prior write Note 49			515	nS
	RD or **ENB** Active to **WAIT_ACK** Inactive - 16-bit read - 16-bit pending prior write Note 49			575	nS
t_{wadv}	**WAIT_ACK** Inactive to Data Valid - normal **WAIT_ACK**			5	nS
t_{dvwa}	Data Valid Before **WAIT_ACK** Inactive - delayed **WAIT_ACK**	15			nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 47: **RD_WR** and **ENB** signaling.

Note 48: **RD** and **WR** signaling.

Note 49: Add 20nS if delayed **WAIT_ACK** is enabled.

Note: Timing values are with respect to an equivalent test load of 25 pF.

8.7.3.3 Write Timing Requirements - Legacy Mode

The following sections present timing requirements while in LAN9252 compatible Legacy mode (i.e. not in EtherCAT Direct Mapped mode). If enabled for output, **WAIT_ACK** is held inactive (ACK) and is shown as such.

If **RD** and **WR** signaling is used, a host write cycle begins when **WR** is asserted with **CS** active. The cycle ends when **WR** is deasserted. **CS** may be asserted and deasserted along with **WR** but not during WR active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host write cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a write. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.5.3, "Indexed Address Mode Functional Timing Diagrams," on page 97 for functional descriptions.

FIGURE 8-74: INDEXED ADDRESSING WRITE CYCLE TIMING

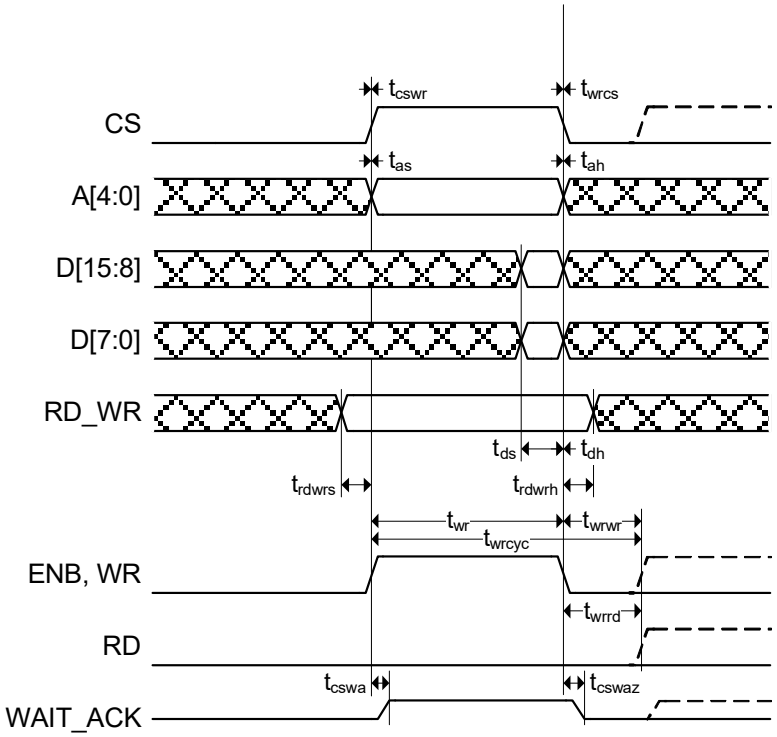


TABLE 8-16: INDEXED ADDRESSING WRITE CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{cswr}	**CS** Setup to **WR** or **ENB** Active	0			nS
t_{wrcs}	**CS** Hold from **WR** or **ENB** Inactive	0			nS
t_{as}	Address Setup to **WR** or **ENB** Active	0			nS
t_{ah}	Address Hold from **WR** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 50	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 50	5			nS
t_{ds}	Data Setup to **WR** or **ENB** Inactive	10			nS
t_{dh}	Data Hold from **WR** or **ENB** Inactive	0			nS
t_{wr}	**WR** or **ENB** Active Time	32			nS
t_{wrcyc}	**WR** or **ENB** Cycle Time	45			nS
t_{wrwr}	**WR** or **ENB** De-assertion Time before Next **WR** or **ENB**	13			nS
t_{wrrd}	**WR** De-assertion Time before Next **RD** Note 51	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 50: **RD_WR** and **ENB** signaling.

Note 51: **RD** and **WR** signaling.

8.7.3.4 Posted Write Timing Requirements - EtherCAT Direct Mapped Mode

The following sections present functional timing diagrams while in EtherCAT Direct Mapped mode.

If enabled for output, **WAIT_ACK** is used to indicate when the write cycle may be concluded, otherwise the host must assume the worst case write access time listed.

WAIT_ACK is output with **CS** active and will be active if there was a pending prior write. Otherwise it will be inactive. The write strobe (**WR** or **ENB**) may be made active. Once **WAIT_ACK** is inactive the write strobe may be made inactive. **WAIT_ACK** becomes active upon an inactive **WR** or **ENB**. If **CS** and **WD/ENB** become inactive concurrently, **WAIT_ACK** is released and the active state may not be seen.

If **RD** and **WR** signaling is used, a host write cycle begins when **WR** is asserted with **CS** active. The cycle ends when **WR** is deasserted. **CS** may be asserted and deasserted along with **WR** but not during **WR** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host write cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a write. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.6.3, "Indexed Address Mode Functional Timing Diagrams," on page 122 for functional descriptions.

FIGURE 8-75: INDEXED ADDRESSING POSTED WRITE CYCLE TIMING

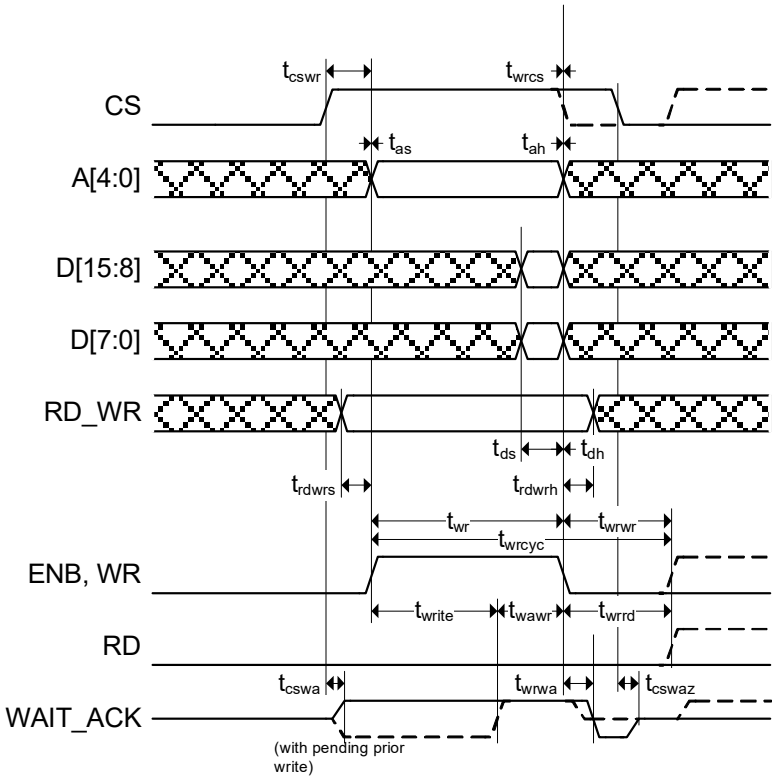


TABLE 8-17: INDEXED ADDRESSING POSTED WRITE CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{cswr}	**CS** Setup to **WR** or **ENB** Active	0			nS
t_{wrcs}	**CS** Hold from **WR** or **ENB** Inactive	0			nS
t_{as}	Address Setup to **WR** or **ENB** Active	0			nS
t_{ah}	Address Hold from **WR** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 52	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 52	5			nS
t_{ds}	Data Setup to **WR** or **ENB** Inactive	10			nS
t_{dh}	Data Hold from **WR** or **ENB** Inactive	0			nS
t_{wr}	**WR** or **ENB** Active Time - no pending prior write	32			nS
t_{wrcyc}	**WR** or **ENB** Cycle Time - no pending prior write	45			nS
t_{wrwr}	**WR** or **ENB** De-assertion Time before Next **WR** or **ENB**	13			nS
t_{wrrd}	**WR** De-assertion Time before Next RD Note 53	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{write}	**WR** or **ENB** Active to **WAIT_ACK** Inactive - pending prior 8-bit write			200	nS
	WR or **ENB** Active to **WAIT_ACK** Inactive - pending prior 16-bit write			280	nS
t_{wawr}	**WAIT_ACK** Inactive to **WR** or **ENB** Inactive	0			nS
t_{wrwa}	**WR** or **ENB** Inactive to **WAIT_ACK** Active Note 54			10	nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 52: **RD_WR** and **ENB** signaling.

Note 53: **RD** and **WR** signaling.

Note 54: If not three-stated first.

8.7.3.5 Non-Posted Write Timing Requirements - EtherCAT Direct Mapped Mode

The following sections present functional timing diagrams while in EtherCAT Direct Mapped mode.

If enabled for output, **WAIT_ACK** is used to indicate when the write cycle may be concluded, otherwise the host must assume the worst case write access time listed.

WAIT_ACK is output with **CS** active. It will be initially inactive and becomes active with write strobe (**WR** or **ENB**) active. Once **WAIT_ACK** is inactive the write strobe may be made inactive.

If **RD** and **WR** signaling is used, a host write cycle begins when **WR** is asserted with **CS** active. The cycle ends when **WR** is deasserted. **CS** may be asserted and deasserted along with **WR** but not during **WR** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host write cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a write. The cycle ends when **ENB** is deasserted. **CS** may be asserted and deasserted along with **ENB** but not during **ENB** active.

Please refer to Section 8.6.3, "Indexed Address Mode Functional Timing Diagrams," on page 122 for functional descriptions.

FIGURE 8-76: INDEXED ADDRESSING NON-POSTED WRITE CYCLE TIMING

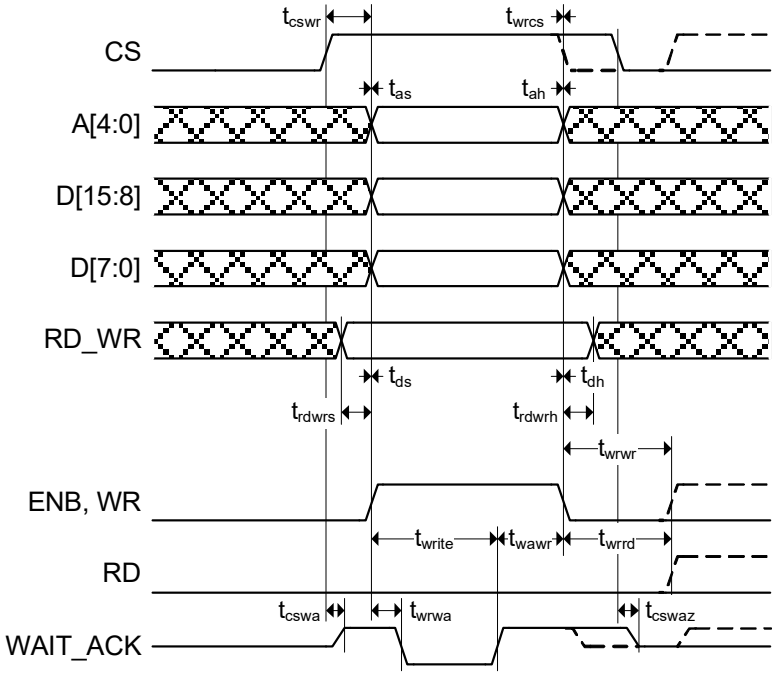


TABLE 8-18: INDEXED ADDRESSING NON-POSTED WRITE CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{cswr}	**CS** Setup to **WR** or **ENB** Active	0			nS
t_{wrcs}	**CS** Hold from **WR** or **ENB** Inactive	0			nS
t_{as}	Address Setup to **WR** or **ENB** Active	0			nS
t_{ah}	Address Hold from **WR** or **ENB** Inactive	0			nS
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 55	5			nS
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 55	5			nS
t_{ds}	Data Setup to **WR** or **ENB** Active	0			nS
t_{dh}	Data Hold from **WR** or **ENB** Inactive	0			nS
t_{wrwr}	**WR** or **ENB** De-assertion Time before Next **WR** or **ENB**	13			nS
t_{wrrd}	**WR** De-assertion Time before Next **RD** Note 56	13			nS
t_{cswa}	**CS** Active to **WAIT_ACK** Valid			10	nS
t_{wrwa}	**WR** or **ENB** Active to **WAIT_ACK** Active			10	nS
t_{write}	**WR** or **ENB** Active to **WAIT_ACK** Inactive - 8-bit write			200	nS
	WR or **ENB** Active to **WAIT_ACK** Inactive - 16-bit write			280	nS
t_{wawr}	**WAIT_ACK** Inactive to **WR** or **ENB** Inactive	0			nS
t_{cswaz}	**WAIT_ACK** Turn Off Time from **CS** Inactive			9	nS

Note 55: **RD_WR** and **ENB** signaling.

Note 56: **RD** and **WR** signaling.

9.0 SPI/SQI SLAVE

9.1 Functional Overview

The SPI/SQI Slave module provides a low pin count synchronous slave interface that facilitates communication between the device and a host system. The SPI/SQI Slave allows access to the System CSRs and internal FIFOs and memories. It supports single and multiple register read and write commands with incrementing, decrementing and static addressing. Single, Dual and Quad bit lanes are supported in SPI mode with a clock rate of up to 80 MHz. SQI mode always uses four bit lanes and also operates at up to 80 MHz.

The following is an overview of the functions provided by the SPI/SQI Slave:

- **Serial Read:** 4-wire (clock, select, data in and data out) reads at up to 30 MHz. Serial command, address and data. Single and multiple register reads with incrementing, decrementing or static addressing.
- **Fast Read:** 4-wire (clock, select, data in and data out) reads at up to 80 MHz. Serial command, address and data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- **Dual / Quad Output Read:** 4 or 6-wire (clock, select, data in / out) reads at up to 80 MHz. Serial command and address, parallel data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- **Dual / Quad I/O Read:** 4 or 6-wire (clock, select, data in / out) reads at up to 80 MHz. Serial command, parallel address and data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- **SQI Read:** 6-wire (clock, select, data in / out) writes at up to 80 MHz. Parallel command, address and data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- **Write:** 4-wire (clock, select, data in and data out) writes at up to 80 MHz. Serial command, address and data. Single and multiple register writes with incrementing, decrementing or static addressing.
- **Dual / Quad Data Write:** 4 or 6-wire (clock, select, data in / out) writes at up to 80 MHz. Serial command and address, parallel data. Single and multiple register writes with incrementing, decrementing or static addressing.
- **Dual / Quad Address / Data Write:** 4 or 6-wire (clock, select, data in / out) writes at up to 80 MHz. Serial command, parallel address and data. Single and multiple register writes with incrementing, decrementing or static addressing.
- **SQI Write:** 6-wire (clock, select, data in / out) writes at up to 80 MHz. Parallel command, address and data. Single and multiple register writes with incrementing, decrementing or static addressing.

9.1.1 ETHERCAT DIRECT MAPPED MODE PCB BACKWARDS COMPATIBILITY

While in EtherCAT Direct Mapped Mode, additional pins are required for full functionality. However with some caveats, EtherCAT Direct Mapped Mode can be made to work with PCBs designed for the Microchip LAN9252.

WAIT_ACK: The **WAIT_ACK** pin on the LAN9252 functioned as the fiber mode signal detect as well as the Port B FX-SD Enable. For copper twisted pair operation, this pin would either be tied to or pulled down to ground. **WAIT_ACK** is disabled by default to avoid a short. Without the **WAIT_ACK** connected to the host processor, the host SPI bus cycles must assume worst case cycle timing.

9.2 SPI/SQI Slave Operation

Input data on the **SIO[3:0]** pins is sampled on the rising edge of the **SCK** input clock. Output data is sourced on the **SIO[3:0]** pins with the falling edge of the clock. The **SCK** input clock can be either an active high pulse or an active low pulse. When the **SCS#** chip select input is high, the **SIO[3:0]** inputs are ignored and the **SIO[3:0]** outputs are three-stated.

In SPI mode, the 8-bit instruction is started on the first rising edge of the input clock after **SCS#** goes active. The instruction is always input serially on **SI/SIO0**.

For read and write instructions, two address bytes follow the instruction byte. Depending on the instruction, the address bytes are input either serially, or 2 or 4 bits per clock. Although some registers are accessed as DWORDs, the address field is considered a byte address. Fourteen address bits specify the address. Bits 15 and 14 of the address field specifies that the address is auto-decremented (10b) or auto-incremented (01b) for continuous accesses.

For read commands (except the READ instruction), one or two transfer length bytes follow the address bytes. Depending on the instruction, the transfer length bytes are input either serially, or 2 or 4 bits per clock. Although some registers are accessed as DWORDs, the transfer length field is considered a byte length. For the one byte transfer length format, bit 7 is low and bits 6-0 specify the length up to 127 bytes. For the two byte transfer length format, bit 7 of the first byte is high and bits 6-0 specify the lower 7 bits of the length. Bits 6-0 of the of the second byte field specify the upper 7 bits of the length with a maximum transfer length of 16,383 bytes (16K-1). The transfer length does not include any Dummy Bytes.

For read and write instructions, a programmable number of initial Dummy Byte cycles follow the address or transfer length (if applicable) bytes (preceding the first data byte).

For read and write instructions, one or more data bytes follow the Dummy Bytes (if present, else they follow the address or transfer length (if applicable) bytes). The data is input or output either serially, or 2 or 4 bits per clock.

A programmable number of Dummy Bytes (including zero) separates each data byte within a DWORD. A separately programmable number of Dummy Bytes (including zero) separates each DWORD. The device does not drive the outputs during the initial Dummy Byte cycles (that follow the address bytes) or Dummy Bytes for write instructions. It does drive the outputs during the subsequent Dummy Byte cycles for read commands. Dummy input bytes should be zero for future compatibility. Dummy output bytes are undefined. The Dummy Byte(s) are input and output either serially, or 2 or 4 bits per clock.

The number of Dummy Bytes is individually programmable per read and write command type and is set with the Set SPI Config (SPICFG) instruction. The SPICFG instruction itself does not utilize Dummy Bytes so that its format is consistent.

SQI mode is entered from SPI with the Enable Quad I/O (EQIO) instruction. Once in SQI mode, all further command, addresses, dummy bytes and data bytes are 4 bits per clock. SQI mode can be exited using the Reset Quad I/O (RSTQIO) instruction.

All instructions, addresses, transfer lengths, and data are transferred with the most-significant bit (msb) or di-bit (msd) or nibble (msn) first. Addresses are transferred with the most-significant byte (MSB) first. Transfer lengths are transferred with the most-significant byte (LSB) first. Multiple BYTE Data is transferred with the least-significant byte (LSB) first (little endian).

The SPI interface supports up to a 80 MHz input clock. Normal (non-high speed) reads instructions are limited to 30 MHz. The programmable Dummy Byte count may be used to pace the data rate or the initial read access time.

The SPI interface supports a minimum time of 50 ns between successive commands (a minimum $\overline{SCS}$ inactive time of 50 ns).

The instructions supported in SPI mode are listed in Table 9-1. SQI instructions are listed in Table 9-2. Unsupported instructions are must not be used.

TABLE 9-1: SPI INSTRUCTIONS

Instruction	Description	Bit Width Note 1	Inst. code	Addr. Bytes	Length Bytes	Initial Dummy Bytes Note 2 Note 3	Per BYTE or DWORD Dummy Bytes Note 4	Data bytes Note 5	Max Freq.
Configuration									
SETCFG	Set Configuration	1-0-1	01h	0	0	0	0	39	80 MHz
EQIO	Enable SQI	1-0-0	38h	0	0	0	0	0	80 MHz
RSTQIO	Reset SQI	1-0-0	FFh	0	0	0	0	0	80 MHz

TABLE 9-1: SPI INSTRUCTIONS (CONTINUED)

Instruction	Description	Bit Width Note 1	Inst. code	Addr. Bytes	Length Bytes	Initial Dummy Bytes Note 2 Note 3	Per BYTE or DWORD Dummy Bytes Note 4	Data bytes Note 5	Max Freq.
Read									
READ	Read	1-1-1	03h	2	0	0 to 255	0 to 255	4 to ∞ / 1 to ∞	30 MHz
FASTREAD	Read at higher speed	1-1-1	0Bh	2	1 or 2	1 to 255	0 to 255	4 to 16K / 1 to 16K	80 MHz
SDOR	SPI Dual Output Read	1-1-2	3Bh	2	1 or 2	1 to 255	0 to 255	4 to 16K / 1 to 16K	80 MHz
SDIOR	SPI Dual I/O Read	1-2-2	BBh	2	1 or 2	2 to 255	0 to 255	4 to 16K / 1 to 16K	80 MHz
SQOR	SPI Quad Output Read	1-1-4	6Bh	2	1 or 2	1 to 255	0 to 255	4 to 16K / 1 to 16K	80 MHz
SQIOR	SPI Quad I/O Read	1-4-4	EBh	2	1 or 2	4 to 255	0 to 255	4 to 16K / 1 to 16K	80 MHz
Write									
WRITE	Write	1-1-1	02h	2	0	0 to 255	0 to 255	4 to ∞ / 1 to ∞	80 MHz
SDDW	SPI Dual Data Write	1-1-2	32h	2	0	0 to 255	0 to 255	4 to ∞ / 1 to ∞	80 MHz
SDADW	SPI Dual Address / Data Write	1-2-2	B2h	2	0	0 to 255	0 to 255	4 to ∞ / 1 to ∞	80 MHz
SQDW	SPI Quad Data Write	1-1-4	62h	2	0	0 to 255	0 to 255	4 to ∞ / 1 to ∞	80 MHz
SQADW	SPI Quad Address / Data Write	1-4-4	E2h	2	0	0 to 255	0 to 255	4 to ∞ / 1 to ∞	80 MHz

Note 1: The bit width format is: instruction code bit width, address / transfer length / initial dummy bit width, data / subsequent dummy bit width.

Note 2: Although they are set as the default, the minimum values are not enforced by the hardware and should be used by software.

The minimum values support operation in LAN9252 compatibility mode for operation up to 80 MHz. In EtherCAT Direct Mapped Mode alternate minimum values should be used as described in Section 9.3, EtherCAT Direct Mapped Mode.

Note 3: Th bit width of the initial Dummy Bytes follows that of the address.

Note 4: The bit width of the inter-data Dummy Bytes follows that of the data.

Note 5: In LAN9252 compatibility mode, the minimum number of Data Bytes is 4 since all registers must be DWORD accessed. In EtherCAT Direct Mapped Mode, for non-EtherCAT core CSRs, the minimum number of Data Bytes remains as 4. For EtherCAT core CSRs or the Process RAM, the minimum number of Data Bytes is 1.

TABLE 9-2: SQI INSTRUCTIONS

Instruction	Description	Bit Width Note 6	Inst. code	Addr. Bytes	Length Bytes	Initial Dummy Bytes Note 7 Note 8	Per BYTE or DWORD Dummy Bytes Note 9	Data bytes Note 10	Max Freq.
Configuration									
SETCFG	Set Configu-ration	4-0-4	01h	0	0	0	0	39	80 MHz
RSTQIO	Reset SQI	4-0-0	FFh	0	0	0	0	0	80 MHz
Read									
FASTREAD	Read at higher speed	4-4-4	0Bh	2	1 or 2	3 to 255	0 to 255	4 to 16K / 1 to 16K	80 MHz
Write									
WRITE	Write	4-4-4	02h	2	0	0 to 255	0 to 255	4 to ∞ / 1 to ∞	80 MHz

Note 6: The bit width format is: instruction code bit width, address / transfer length / initial dummy bit width, data / subsequent dummy bit width.

Note 7: Although they are set as the default, the minimum values are not enforced by the hardware and should be used by software.

The minimum values support operation in LAN9252 compatibility mode for operation up to 80 MHz. In EtherCAT Direct Mapped Mode alternate minimum values should be used as described in Section 9.3, EtherCAT Direct Mapped Mode.

Note 8: Th bit width of the initial Dummy Bytes follows that of the address.

Note 9: The bit width of the inter-data Dummy Bytes follows that of the data.

Note 10: In LAN9252 compatibility mode, the minimum number of Data Bytes is 4 since all registers must be DWORD accessed. In EtherCAT Direct Mapped Mode, for non-EtherCAT core CSRs, the minimum number of Data Bytes remains as 4. For EtherCAT core CSRs or the Process RAM, the minimum number of Data Bytes is 1.

9.2.1 DEVICE INITIALIZATION

Until the device has been initialized to the point where the various configuration inputs are valid, the SPI/SQI interface does not respond to and is not affected by any external pin activity.

Once device initialization completes, the SPI/SQI interface will ignore the pins until a rising edge of **SCS#** is detected.

9.2.1.1 SPI/SQI Slave Read Polling for Initialization Complete

Before device initialization, the SPI/SQI interface will not return valid data. To determine when the SPI/SQI interface is functional, the Byte Order Test Register (BYTE_TEST) should be polled. Once the correct pattern is read, the interface can be considered functional. At this point, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) can be polled to determine when the device is fully configured.

Note:	The Host should only use single register reads (one data cycle per SCS# low) while polling the BYTE_TEST register.

9.2.2 ACCESS DURING AND FOLLOWING POWER MANAGEMENT

During any power management mode other than D0, reads and writes are ignored and the SPI/SQI interface does not respond to and is not affected by any external pin activity.

Once the power management mode changes back to D0, the SPI/SQI interface will ignore the pins until a rising edge of **SCS#** is detected.

To determine when the SPI/SQI interface is functional, the Byte Order Test Register (BYTE_TEST) should be polled. Once the correct pattern is read, the interface can be considered functional. At this point, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) can be polled to determine when the device is fully configured.

Note:	The Host should only use single register reads (one data cycle per **SCS#** low) while polling the BYTE_TEST register.

9.2.3 WAIT / ACKNOWLEDGMENT OPERATION

When operating in EtherCAT Direct Mapped mode (see Section 9.3, EtherCAT Direct Mapped Mode), the host system must meet the device's timing access requirements. For reads from the EtherCAT Core CSR or Process Data RAM, the read cycles must wait until read data has been internally retrieved. All write cycles must wait for any prior write access to the EtherCAT Core CSR or Process Data RAM to internally complete. The host system may either wait the specified worst case access time, or may use the **WAIT_ACK** signal.

The **WAIT_ACK** pin is enabled via the combination of the SPI WAIT_ACK Polarity and SPI WAIT_ACK Buffer Type bits of the PDI Configuration Register (SPI Modes).

WAIT_ACK is only used when operating in EtherCAT Direct Mapped mode. When not in EtherCAT Direct Mapped mode, if **WAIT_ACK** is enabled it is always inactive (showing ACK). **WAIT_ACK** may be set as an active low open drain output for wire-AND systems or as a three-stated push-pull output. This is controlled by the SPI WAIT_ACK Buffer Type bit of the PDI Configuration Register (SPI Modes). As a push-pull output, its polarity is set by the SPI WAIT_ACK Polarity bit of the PDI Configuration Register (SPI Modes). **WAIT_ACK** operation is described in Section 9.3.2.3, Wait States.

9.2.4 SPI CONFIGURATION COMMANDS

9.2.4.1 Set Configuration

The Set Configuration instruction sets the number of Dummy Bytes expected for each instruction. This instruction is supported in SPI and SQI bus protocols with clock frequencies up to 80 MHz.

The SPI/SQI slave interface is selected by first bringing **SCS#** active. The 8-bit SETCFG instruction, 01h, is input into the **SI/SIO[0]** pin, one bit per clock, in SPI mode and into the **SIO[3:0]** pins, four bits per clock, in SQI mode.

The data follows the command byte. For SPI mode, the data is input into the **SI/SIO[0]** pin starting with the msb of the first byte. For SQI mode the data is input nibble wide using **SIO[3:0]** starting with the msn of the first byte. The remaining bits/nibbles are shifted in on subsequent clock edges.

The **SCS#** input is brought inactive to conclude the cycle.

The order of the Dummy Bytes along with their defaults follow. There are three values per instruction. The first is the number of Dummy Bytes that will precede the first data byte. The second is the number of Dummy Bytes that occur between bytes within a DWORD. The third is the number of Dummy Bytes that occur between DWORDs. Note there are no Dummy Bytes after the last data byte of a command.

The hardware defaults shown support operation in LAN9252 compatibility mode for operation up to 80 MHz. In EtherCAT Direct Mapped Mode, alternate minimum values should be set as described in Section 9.3, EtherCAT Direct Mapped Mode.

TABLE 9-3: DUMMY BYTE ORDER AND DEFAULT

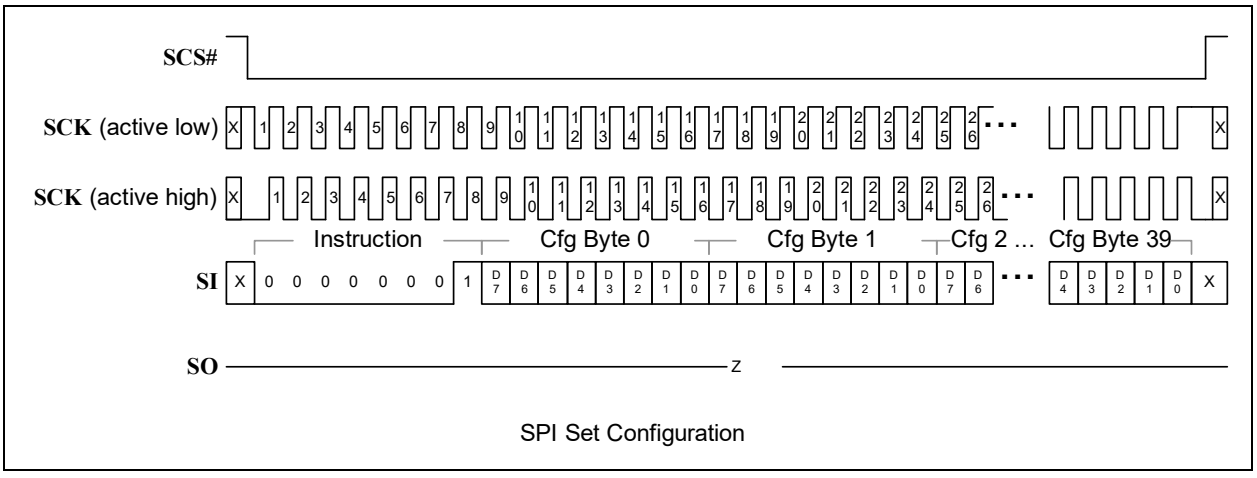
Byte Order	Instruction	Initial	Within a DWORD	Between DWORDs
		Default		
SPI				
0	READ	0	-	-
1		-	0	-
2			-	0
3	FASTREAD	1	-	-
4		-	0	-
5			-	0
6	SDOR	1	-	-
7		-	0	-
8			-	0
9	SDIOR	2	-	-
10		-	0	-
11			-	0
12	SQOR	1	-	-
13		-	0	-
14			-	0
15	SQIOR	4	-	-
16		-	0	-
17			-	0
18	WRITE	0	-	-
19		-	0	-
20			-	0
21	SDDW	0	-	-
22		-	0	-
23			-	0

TABLE 9-3: DUMMY BYTE ORDER AND DEFAULT (CONTINUED)

Byte Order	Instruction	Initial	Within a DWORD	Between DWORDs
		Default		
24	SDADW	0	-	-
25		-	0	-
26			-	0
27	SQDW	0	-	-
28		-	0	-
29			-	0
30	SQADW	0	-	-
31		-	0	-
32			-	0
SQI				
33	FASTREAD	3	-	-
34		-	0	-
35			-	0
36	WRITE	0	-	-
37		-	0	-
38			-	0

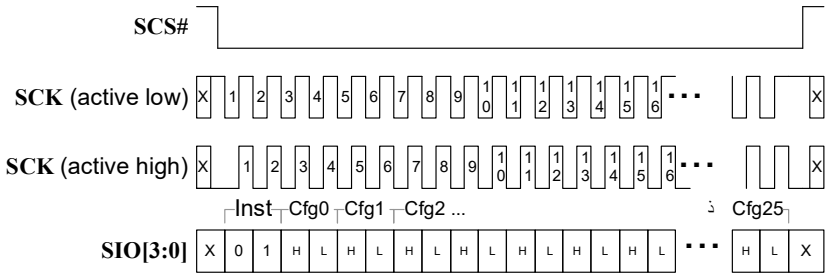
Figure 9-1 illustrates the Set Configuration instruction for SPI mode. Figure 9-2 illustrates the Set Configuration instruction for SQI mode.

FIGURE 9-1: SPI MODE SET CONFIGURATION



SPI Set Configuration

FIGURE 9-2: SQI MODE SET CONFIGURATION



SQI Set Configuration

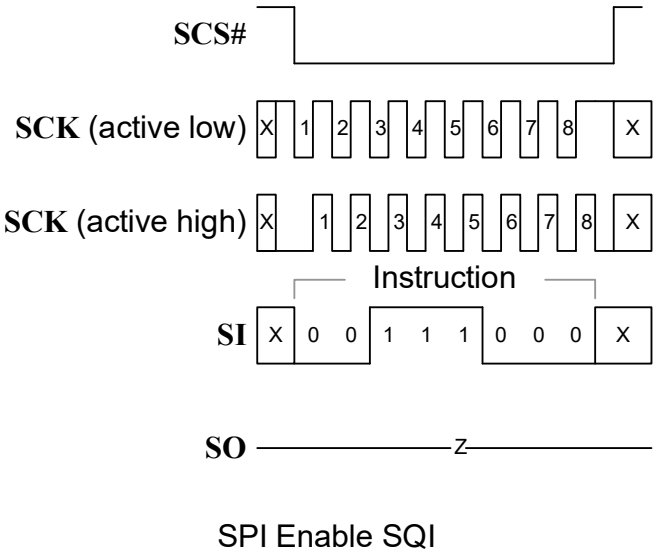
9.2.4.2 Enable SQI

The Enable SQI instruction changes the mode of operation to SQI. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit EQIO instruction, 38h, is input into the **SI/SIO[0]** pin one bit per clock. The **SCS#** input is brought inactive to conclude the cycle.

Figure 9-3 illustrates the Enable SQI instruction.

FIGURE 9-3: ENABLE SQI



SPI Enable SQI

9.2.4.3 Reset SQI

The Reset SQI instruction changes the mode of operation to SPI. This instruction is supported in SPI and SQI bus protocols with clock frequencies up to 80 MHz.

The SPI/SQI slave interface is selected by first bringing **SCS#** active. The 8-bit RSTQIO instruction, FFh, is input into the **SI/SIO[0]** pin, one bit per clock, in SPI mode and into the **SIO[3:0]** pins, four bits per clock, in SQI mode. The **SCS#** input is brought inactive to conclude the cycle.

Figure 9-4 illustrates the Reset SQI instruction for SPI mode. Figure 9-5 illustrates the Reset SQI instruction for SQI mode.

FIGURE 9-4: SPI MODE RESET SQI

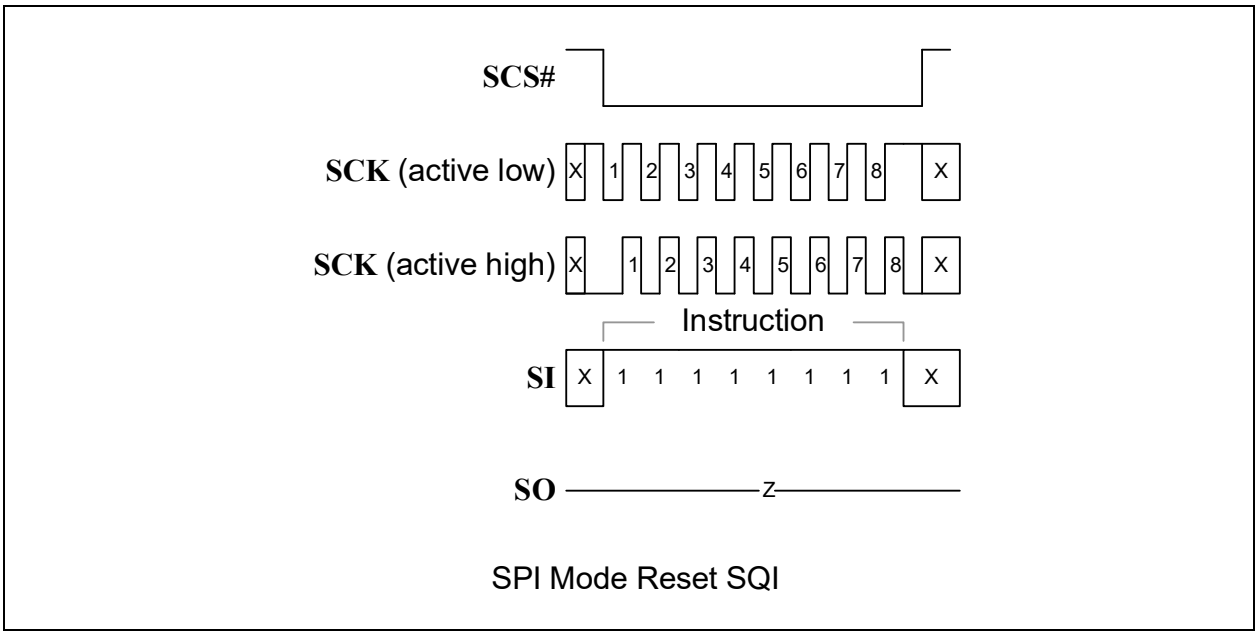
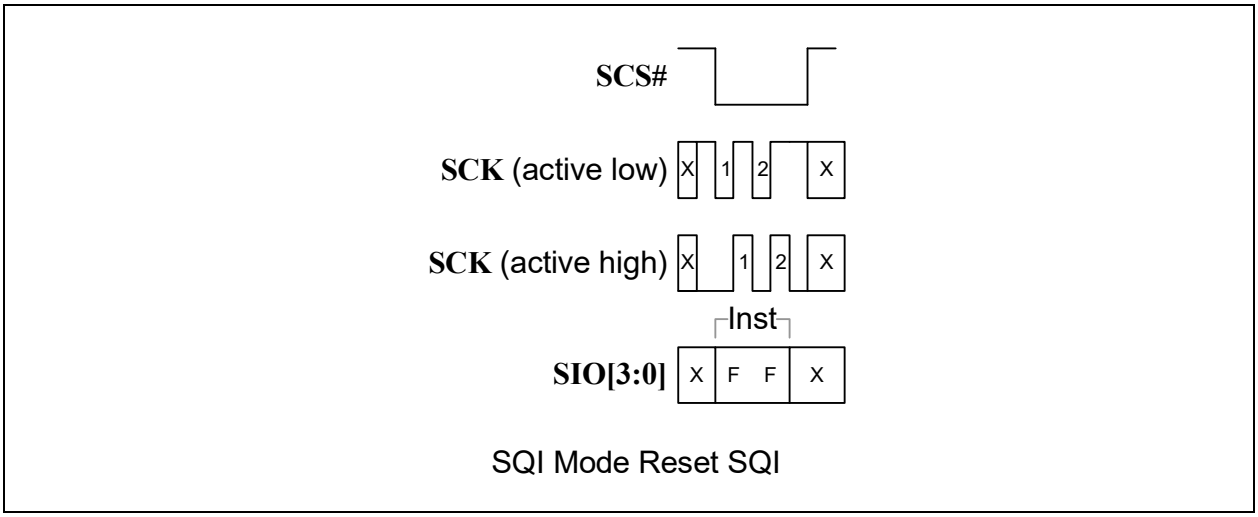


FIGURE 9-5: SQI MODE RESET SQI



9.2.5 SPI READ COMMANDS

Various read commands are support by the SPI/SQI slave. The following applies to all read commands.

MULTIPLE READS

Additional reads, beyond the first, are performed by continuing the clock pulses while SCS# is active. The upper two bits of the address specify auto-incrementing (address[15:14]=01b) or auto-decrementing (address[15:14]=10b). The internal address is incremented, decremented, or maintained based on these bits. Maintaining a fixed internal address is useful for register polling.

Towards the end of the current output shift the address is incremented or decremented, if appropriate, and another synchronized capture sequence is done.

Constant address and Auto-increment/decrement operation for multiple DWORDs operates as follows. Note that it is the DWORD address that remains constant or is incremented or decremented. The byte address within the DWORD always increments:

dec/inc=00: constant DWORD aligned address

read 4 bytes within the DWORD in incrementing byte order
internal address repeats each DWORD

starting address : yyyy
internal address : yyyy

byte address :	yyyy	+1	+2	+3
First data output :	valid	valid	valid	valid

internal address : yyyy

byte address :	yyyy	+1	+2	+3
Second data output :	valid	valid	valid	valid

internal address : yyyy

byte address :	yyyy	+1	+2	+3
Last data output :	valid	valid	valid	valid

dec/inc=01: incrementing DWORD aligned starting address

read 4 bytes within each DWORD in incrementing byte order
internal address is incremented by 4 each DWORD

starting address : yyyy
internal address : yyyy

byte address :	yyyy	+1	+2	+3
First data output :	valid	valid	valid	valid

next internal address : next DWORD

byte address :	+4	+5	+6	+7
Second data output :	valid	valid	valid	valid

next internal address : next DWORD

byte address :	+8	+9	+10	+11
Last data output :	valid	valid	valid	valid

dec/inc=10: decrementing DWORD aligned starting address

read 4 bytes within each DWORD in incrementing byte order
internal address is decremented by 4 each DWORD

starting address : yyyy
internal address : yyyy

byte address :	yyyy	+1	+2	+3
First data output :	valid	valid	valid	valid

next internal address : previous DWORD

byte address :	-4	-3	-2	-1
Second data output :	valid	valid	valid	valid
next internal address : previous DWORD				
byte address :	-8	-7	-6	-5
Last data output :	valid	valid	valid	valid

dec/inc=11: RESERVED

The above apply to full DWORD accesses starting at DWORD aligned addresses. See Section 9.3.2.1 for partial DWORD accesses or non-DWORD aligned starting addresses utilized during EtherCAT Direct Mapped mode.

READ TERMINATION

To avoid internally prefetching additional data past the last data that will be output, two methods are utilized.

For the READ instruction, the **SI** input is used. During the last output byte, the SPI master must set **SI** high (input byte = FFh), otherwise an internal prefetch will occur with the potential of loosing data.

For other read commands (FASTREAD, SDOR, SDIOR, SQOR, SQIOR), the transfer length in bytes is provided following the address.

DUMMY BYTES

In order to provide sufficient time to retrieve the register data (especially when reading from the EtherCAT Core CSRs or Process RAM while in EtherCAT Direct Mapped mode) Dummy Byte cycles may be used. The number of Dummy Bytes is set using the Set Configuration instruction and is specified per read command type.

There are three values per instruction. The first is the number of Dummy Bytes that will precede the first data byte. The second is the number of Dummy Bytes that occur between bytes within a DWORD (intra-DWORD). The third is the number of Dummy Bytes that occur between DWORDs (inter-DWORD). There are no Dummy Bytes after the last data byte of a command.

APPLICATION NOTE: The number of Dummy Bytes between DWORDs is strictly the third configuration value. It is not in addition to or paralleled with a Dummy Byte count using the second configuration value.

APPLICATION NOTE: The DWORD boundary applies to all read commands even if the command is BYTE oriented (e.g. the READ command). The intra- and inter- number of Dummy Bytes can be set the same if appropriate.

APPLICATION NOTE: For the READ command, a Dummy byte configuration which has second parameter (intra-DWORD) equal to 0 and the third parameter (inter-DWORD) not equal to 0 is not supported. Normally the second and third parameters would be the same value in this case.

APPLICATION NOTE: The DWORD boundary is based on the address of the last byte read, not on the running byte count (i.e. it is not simply every fourth byte). This is important to consider during EtherCAT Direct Mapped Mode where the starting address could be non-DWORD aligned. Refer to Section 9.3.2.1 for how the address is updated.

APPLICATION NOTE: The number of clock cycles for a Dummy Byte varies based on the bit width(s) of the instruction.

SPECIAL CSR HANDLING

Live Bits

Since data is read serially, the selected register's value is saved at the beginning of each 32-bit read to prevent the host from reading an intermediate value. The saving occurs multiple times in a multiple read sequence.

9.2.5.1 Read

The Read instruction inputs the instruction code, the address and the possible initial Dummy bytes one bit per clock and outputs the data and any subsequent Dummy Byte(s) one bit per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 30 MHz. This instruction is not supported in SQI bus protocol.

The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit READ instruction, 03h, is input into the **SI/SIO[0]** pin, followed by the two address bytes and 0 to 255 Dummy Byte(s). The address bytes specify a BYTE address within the device.

On the falling clock edge following the rising edge of the last address or Dummy bit, the **SO/SIO[1]** pin is driven starting with the msb of the LSB of the selected register. The remaining register bits are shifted out on subsequent falling clock edges. For multiple byte registers, the next byte follows the last bit of the preceding byte or Dummy Byte(s). For reads of multiple registers, the first bit of the next register follows the last bit of the preceding register or Dummy Byte(s).

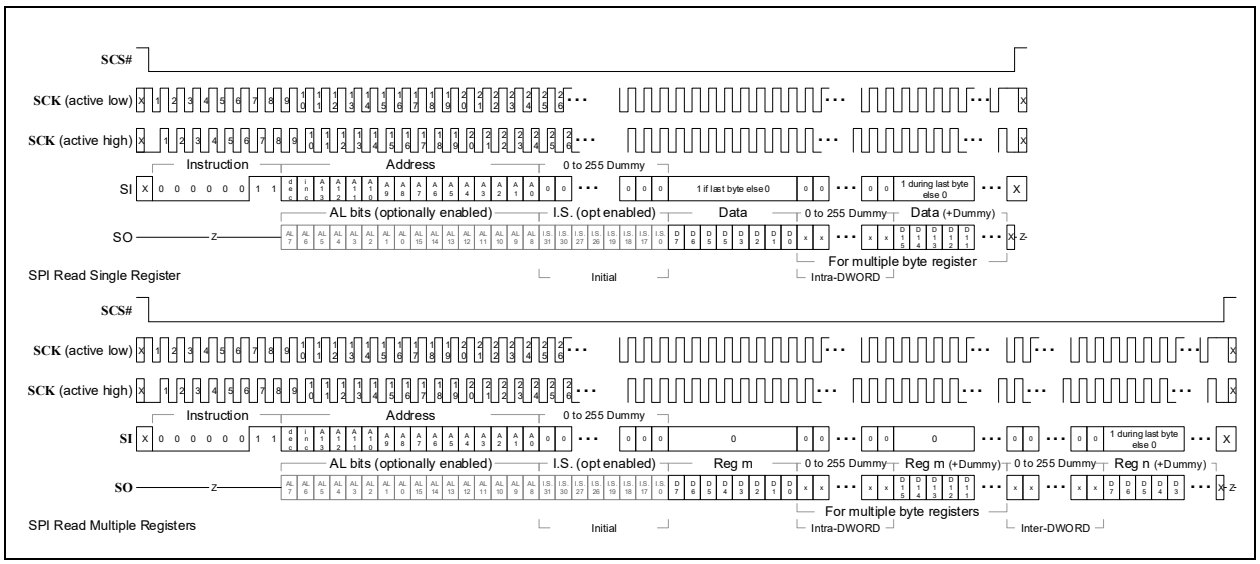
To avoid internally prefetching additional data past the last data that will be output, the **SI** input is used. During the last output byte, the SPI master must set **SI** high (input byte = FFh), otherwise an internal prefetch will occur with the potential of loosing data. For all but the last output byte, SPI master must set **SI** low (input byte = 00h).

The **SCS#** input is brought inactive to conclude the cycle. The **SO/SIO[1]** pin is three-stated at this time.

Figure 9-6 illustrates a typical single and multiple register read. A DWORD aligned full DWORD register is shown. The final DWORD is not followed by Dummy Bytes. A partial DWORD register read is also possible and would terminate after the last data BYTE without subsequent Dummy Byte(s). It is also possible for a partial DWORD read to start on a non-DWORD aligned boundary.

Application Note: A Dummy byte configuration which has second parameter (intra-DWORD) equal to 0 and the third parameter (inter-DWORD) not equal to 0 is not supported. Normally the second and third parameters would be the same value in this case.

FIGURE 9-6: SPI READ



9.2.5.2 Fast Read

The Read at higher speed instruction inputs the instruction code and the address, the transfer length, and the initial Dummy Bytes one bit per clock and outputs the data and any subsequent Dummy Byte(s) one bit per clock. In SQI mode, the instruction code, the address, the transfer length, and the initial Dummy Bytes are input four bits per clock and the data and any subsequent Dummy Byte(s) are output four bits per clock. This instruction is supported in SPI and SQI bus protocols with clock frequencies up to 80 MHz.

The SPI/SQI slave interface is selected by first bringing **SCS#** active. For SPI mode, the 8-bit FASTREAD instruction, 0Bh, is input into the **SI/SIO[0]** pin, followed by the two address bytes, one or two transfer length bytes, and 1 to 255 Dummy Byte(s). For SQI mode, the 8-bit FASTREAD instruction is input into the **SIO[3:0]** pins, followed by the two

address bytes, one or two transfer length bytes, and 3 to 255 Dummy Bytes. The address bytes specify a BYTE address within the device. The transfer length bytes specify the data length in BYTEs and does not include any Dummy Bytes. The transfer length field is either 7 or 14 bits with maximum transfer length of 127 or 16,383 bytes (16K-1).

On the falling clock edge following the rising edge of the last dummy bit (or nibble), the **SO/SIO[1]** pin is driven starting with the msb of the LSB of the selected register. For SQI mode, **SIO[3:0]** are driven starting with the msn of the LSB of the selected register. For a non-DWORD aligned starting address, 1-3 padding bytes (and possible Dummy Bytes per padding byte) occur prior to the data bytes. Padding bytes are not counted in the transfer length.

The remaining register bits are shifted out on subsequent falling clock edges. For multiple byte registers, the next byte follows the last bit (or nibble) of the preceding byte or Dummy Byte(s). For reads of multiple registers, the first bit (or nibble) of the next register follows the last bit (or nibble) of the preceding register or Dummy Byte(s).

The **SCS#** input is brought inactive to conclude the cycle. The **SO/SIO[3:0]** pins are three-stated at this time.

Figure 9-7 illustrates a typical single and multiple register fast read for SPI mode. Figure 9-8 illustrates a typical single and multiple register fast read for SQI mode. The transfer length field may be one or two bytes. A DWORD aligned full DWORD register is shown. The final DWORD is not followed by Dummy Bytes. A partial DWORD register read is also possible and would terminate after the last data BYTE without subsequent Dummy Byte(s). It is also possible for a partial DWORD read to start on a non-DWORD aligned boundary.

FIGURE 9-7: SPI FAST READ

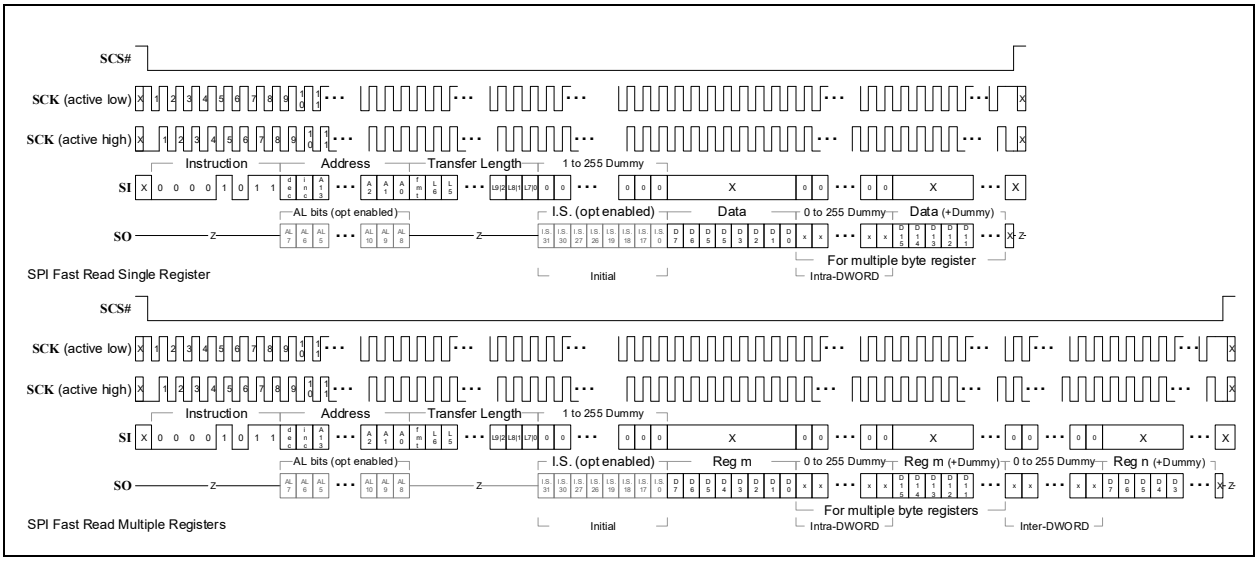
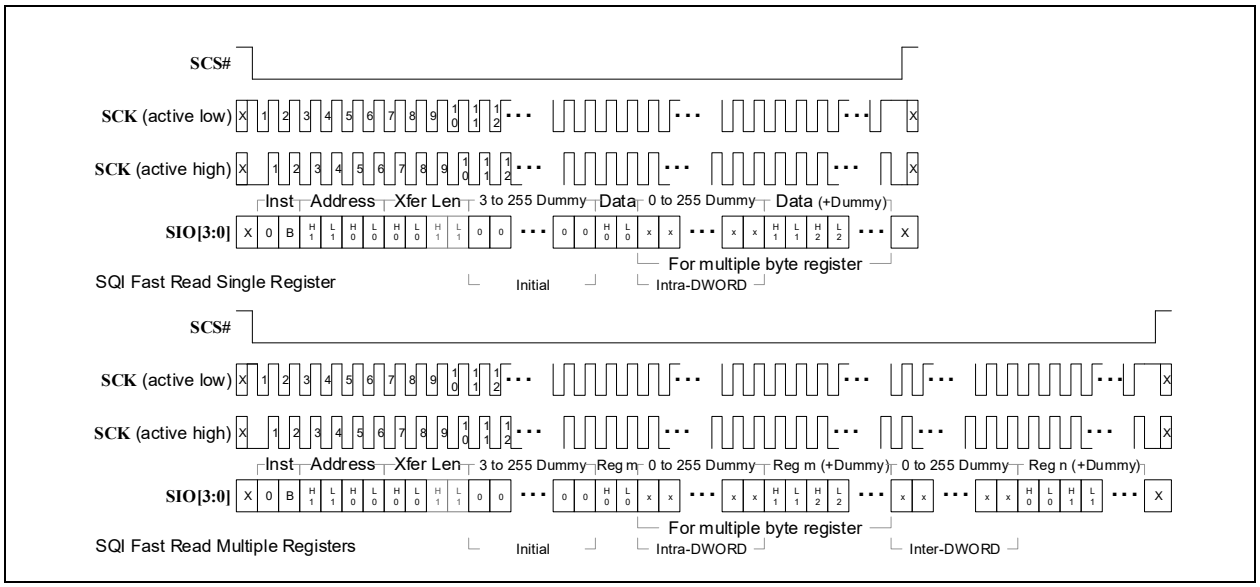


FIGURE 9-8: SQI FAST READ



9.2.5.3 Dual Output Read

The SPI Dual Output Read instruction inputs the instruction code and the address, the transfer length, and the initial Dummy Bytes one bit per clock and outputs the data and any subsequent Dummy Byte(s) two bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SDOR instruction, 3Bh, is input into the **SIO[0]** pin, followed by the two address bytes, one or two transfer length bytes, and 1 to 255 Dummy Byte(s). The address bytes specify a BYTE address within the device. The transfer length bytes specify the data length in BYTEs and does not include any Dummy Bytes. The transfer length field is either 7 or 14 bits with maximum transfer length of 127 or 16,383 bytes (16K-1).

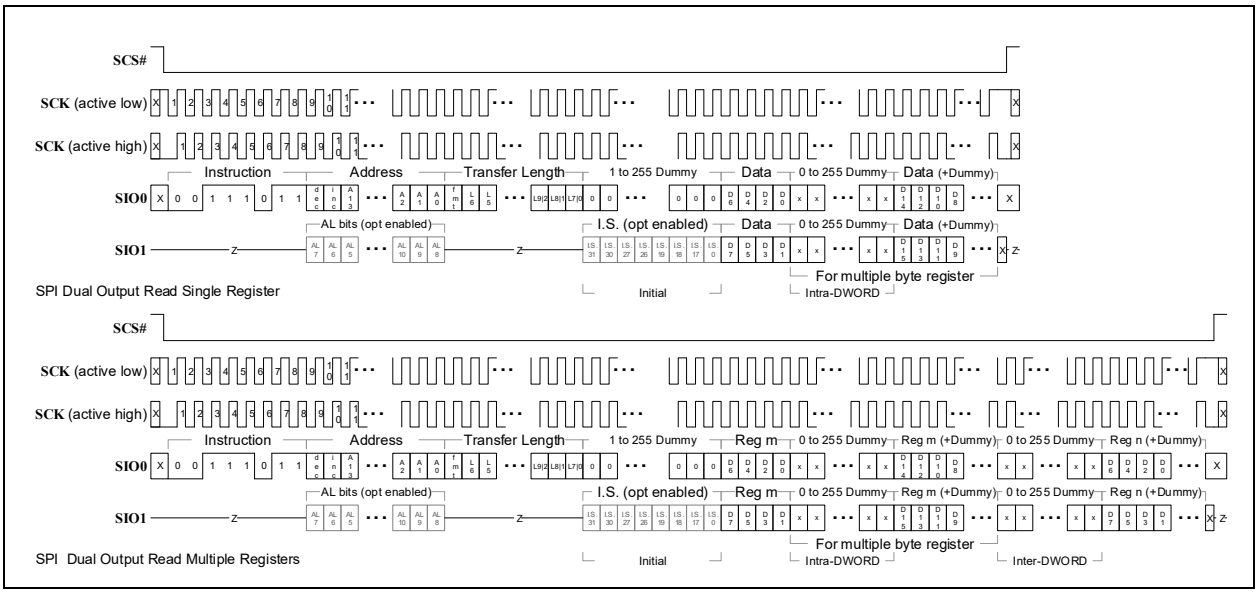
On the falling clock edge following the rising edge of the last dummy di-bit, the **SIO[1:0]** pins are driven starting with the msbs of the LSB of the selected register. For a non-DWORD aligned starting address, 1-3 padding bytes (and possible Dummy Bytes per padding byte) occur prior to the data bytes. Padding bytes are not counted in the transfer length.

The remaining register di-bits are shifted out on subsequent falling clock edges. For multiple byte registers, the next byte follows the last bit (or nibble) of the preceding byte or Dummy Byte(s). For reads of multiple registers, the first bit (or nibble) of the next register follows the last bit (or nibble) of the preceding register or Dummy Byte(s).

The **SCS#** input is brought inactive to conclude the cycle. The **SIO[1:0]** pins are three-stated at this time.

Figure 9-9 illustrates a typical single and multiple register dual output read. The transfer length field may be one or two bytes. A DWORD aligned full DWORD register is shown. The final DWORD is not followed by Dummy Bytes. A partial DWORD register read is also possible and would terminate after the last data BYTE without subsequent Dummy Byte(s). It is also possible for a partial DWORD read to start on a non-DWORD aligned boundary.

FIGURE 9-9: SPI DUAL OUTPUT READ



9.2.5.4 Quad Output Read

The SPI Quad Output Read instruction inputs the instruction code, the address, the transfer length, and the initial Dummy Bytes one bit per clock and outputs the data and any subsequent Dummy Byte(s) four bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

The SPI slave interface is selected by first bringing SCS# active. The 8-bit SQOR instruction, 6Bh, is input into the SIO[0] pin, followed by the two address bytes, one or two transfer length bytes, and 1 to 255 Dummy Byte(s). The address bytes specify a BYTE address within the device. The transfer length bytes specify the data length in BYTEs and does not include any Dummy Bytes. The transfer length field is either 7 or 14 bits with maximum transfer length of 127 or 16,383 bytes (16K-1).

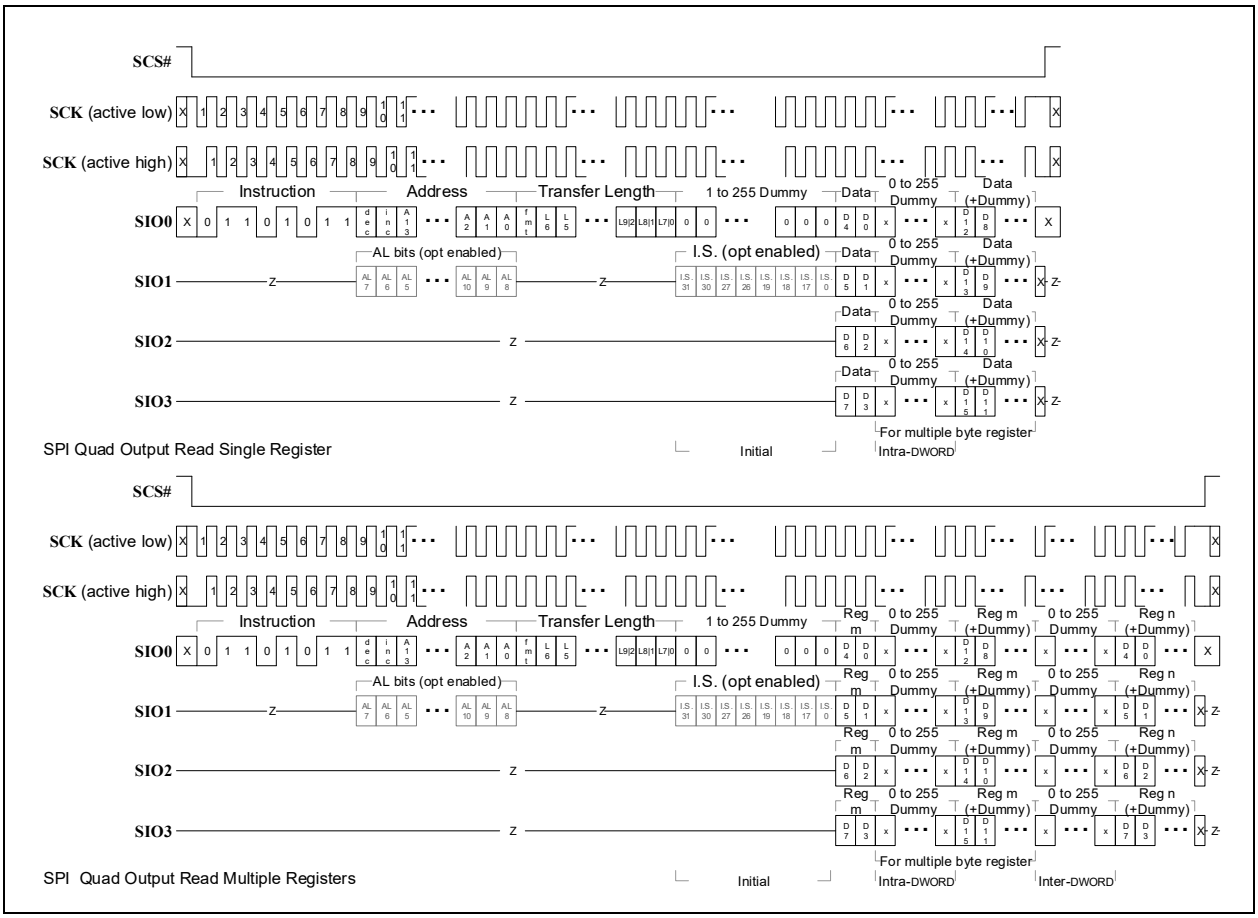
On the falling clock edge following the rising edge of the last dummy bit, the **SIO[3:0]** pins are driven starting with the msn of the LSB of the selected register. For a non-DWORD aligned starting address, 1-3 padding bytes (and possible Dummy Bytes per padding byte) occur prior to the data bytes. Padding bytes are not counted in the transfer length.

The remaining register nibbles are shifted out on subsequent falling clock edges. For multiple byte registers, the next byte follows the last bit (or nibble) of the preceding byte or Dummy Byte(s). For reads of multiple registers, the first bit (or nibble) of the next register follows the last bit (or nibble) of the preceding register or Dummy Byte(s).

The **SCS#** input is brought inactive to conclude the cycle. The **SIO[3:0]** pins are three-stated at this time.

Figure 9-10 illustrates a typical single and multiple register quad output read. The transfer length field may be one or two bytes. A DWORD aligned full DWORD register is shown. The final DWORD is not followed by Dummy Bytes. A partial DWORD register read is also possible and would terminate after the last data BYTE without subsequent Dummy Byte(s). It is also possible for a partial DWORD read to start on a non-DWORD aligned boundary.

FIGURE 9-10: SPI QUAD OUTPUT READ



9.2.5.5 Dual I/O Read

The SPI Dual I/O Read instruction inputs the instruction code one bit per clock and the address, The transfer length, and the initial Dummy Bytes two bits per clock and outputs the data and any subsequent Dummy Byte(s) two bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

The SPI slave interface is selected by first bringing SCS# active. The 8-bit SDIOR instruction, BBh, is input into the SIO[0] pin, followed by the two address bytes, one or two transfer length bytes, and 2 to 255 Dummy Bytes into the SIO[1:0] pins. The address bytes specify a BYTE address within the device. The transfer length bytes specify the data length in BYTEs and does not include any Dummy Bytes. The transfer length field is either 7 or 14 bits with maximum transfer length of 127 or 16,383 bytes (16K-1).

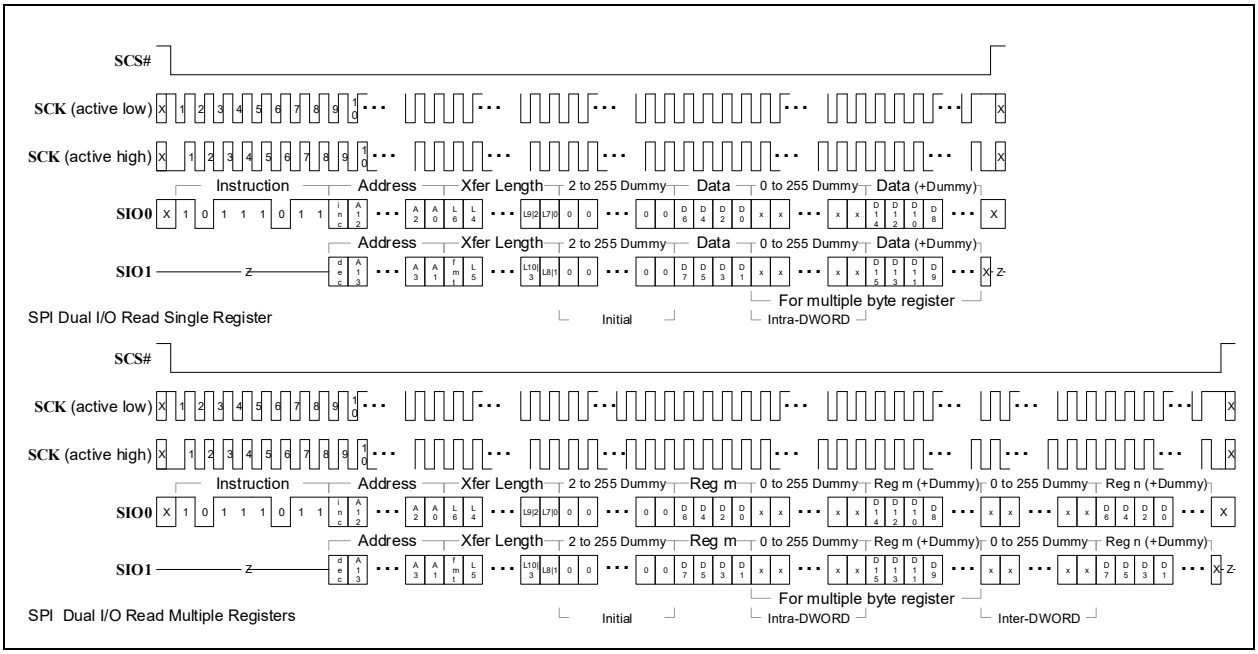
On the falling clock edge following the rising edge of the last dummy di-bit, the **SIO[1:0]** pins are driven starting with the msbs of the LSB of the selected register. For a non-DWORD aligned starting address, 1-3 padding bytes (and possible Dummy Bytes per padding byte) occur prior to the data bytes. Padding bytes are not counted in the transfer length.

The remaining register di-bits are shifted out on subsequent falling clock edges. For multiple byte registers, the next byte follows the last bit (or nibble) of the preceding byte or Dummy Byte(s). For reads of multiple registers, the first bit (or nibble) of the next register follows the last bit (or nibble) of the preceding register or Dummy Byte(s).

The **SCS#** input is brought inactive to conclude the cycle. The **SIO[1:0]** pins are three-stated at this time.

Figure 9-11 illustrates a typical single and multiple register dual I/O read. The transfer length field may be one or two bytes. A DWORD aligned full DWORD register is shown. The final DWORD is not followed by Dummy Bytes. A partial DWORD register read is also possible and would terminate after the last data BYTE without subsequent Dummy Byte(s). It is also possible for a partial DWORD read to start on a non-DWORD aligned boundary.

FIGURE 9-11: SPI DUAL I/O READ



9.2.5.6 Quad I/O Read

The SPI Quad I/O Read instruction inputs the instruction code one bit per clock and the address, the transfer length, and the initial Dummy Bytes four bits per clock and outputs the data and any subsequent Dummy Byte(s) four bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SQIOR instruction, EBh, is input into the **SIO[0]** pin, followed by the two address bytes, one or two transfer length bytes, and 4 to 255 Dummy Bytes into the **SIO[3:0]** pins. The address bytes specify a BYTE address within the device. The transfer length bytes specify the data length in BYTEs and does not include any Dummy Bytes. The transfer length field is either 7 or 14 bits with maximum transfer length of 127 or 16,383 bytes (16K-1).

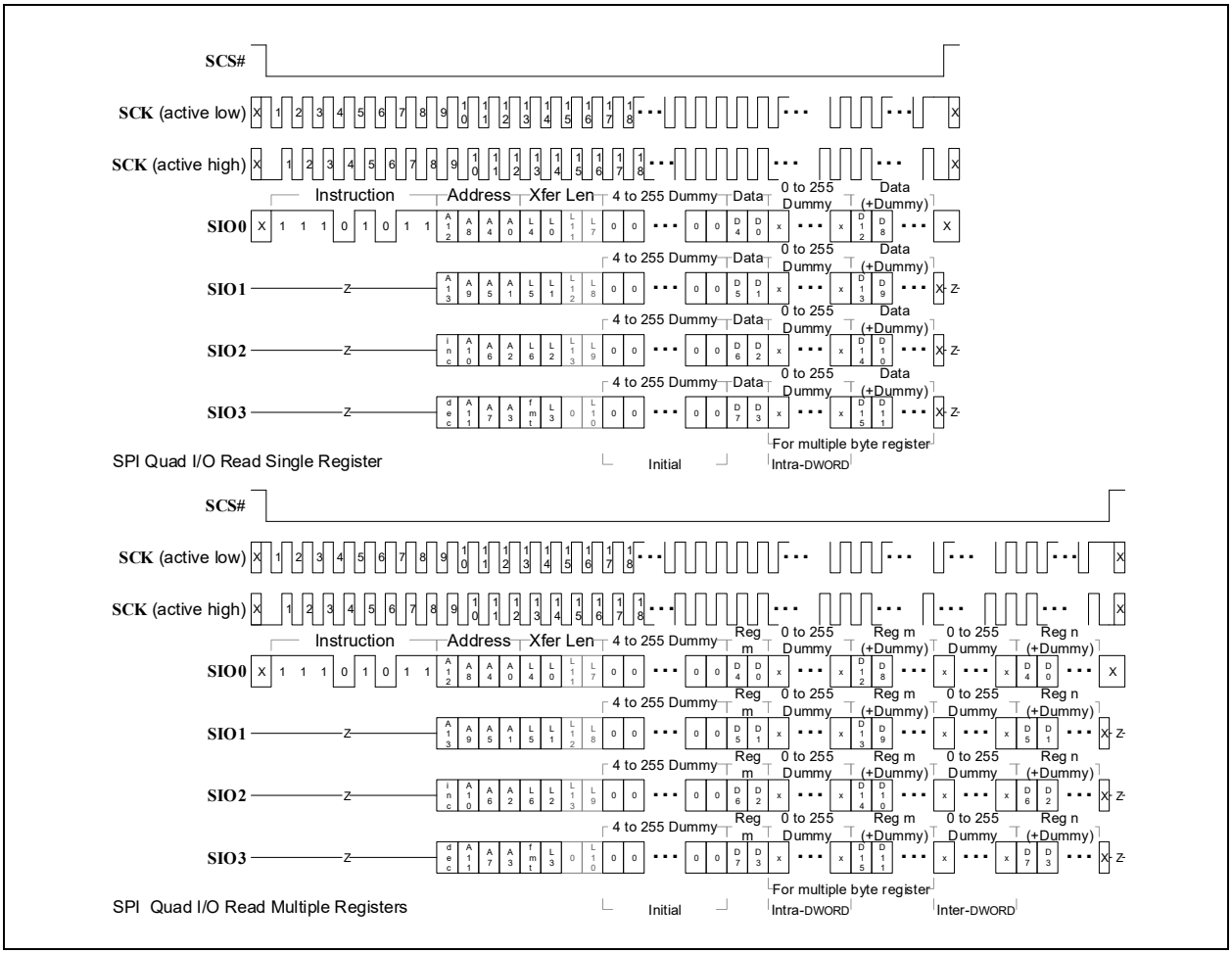
On the falling clock edge following the rising edge of the last dummy nibble, the **SIO[3:0]** pins are driven starting with the msn of the LSB of the selected register. For a non-DWORD aligned starting address, 1-3 padding bytes (and possible Dummy Bytes per padding byte) occur prior to the data bytes. Padding bytes are not counted in the transfer length.

The remaining register nibbles are shifted out on subsequent falling clock edges. For multiple byte registers, the next byte follows the last bit (or nibble) of the preceding byte or Dummy Byte(s). For reads of multiple registers, the first bit (or nibble) of the next register follows the last bit (or nibble) of the preceding register or Dummy Byte(s).

The **SCS#** input is brought inactive to conclude the cycle. The **SIO[3:0]** pins are three-stated at this time.

Figure 9-12 illustrates a typical single and multiple register quad I/O read. The transfer length field may be one or two bytes. A DWORD aligned full DWORD register is shown. The final DWORD is not followed by Dummy Bytes. A partial DWORD register read is also possible and would terminate after the last data BYTE without subsequent Dummy Byte(s). It is also possible for a partial DWORD read to start on a non-DWORD aligned boundary.

FIGURE 9-12: SPI QUAD I/O READ



9.2.6 SPI WRITE COMMANDS

Multiple write commands are support by the SPI/SQI slave. The following applies to all write commands.

MULTIPLE WRITES

Multiple writes are performed by continuing the clock pulses and input data while SCS# is active. The upper two bits of the address specify auto-incrementing (address[15:14]=01b) or auto-decrementing (address[15:14]=10b). The internal DWORD address is incremented, decremented, or maintained based on these bits. Maintaining a fixed internal address may be useful for register “bit-banging” or other repeated writes.

Constant address and Auto-increment/decrement operation for multiple DWORDs operates as follows. Note that it is the DWORD address that remains constant or is incremented or decremented. The byte address within the DWORD always increments:

dec/inc=00: constant DWORD aligned address

write 4 bytes within the DWORD in incrementing byte order
internal address repeats each DWORD

starting address : yyyy
internal address : yyyy

byte address :	yyyy	+1	+2	+3
First data input :	valid	valid	valid	valid

internal address : yyyy

byte address :	yyyy	+1	+2	+3
Second data input :	valid	valid	valid	valid

internal address : yyyy

byte address :	yyyy	+1	+2	+3
Last data input :	valid	valid	valid	valid

dec/inc=01: incrementing DWORD aligned starting address

write 4 bytes within each DWORD in incrementing byte order
internal address is incremented by 4 each DWORD

starting address : yyyy
internal address : yyyy

byte address :	yyyy	+1	+2	+3
First data input :	valid	valid	valid	valid

next internal address : next DWORD

byte address :	+4	+5	+6	+7
Second data input :	valid	valid	valid	valid

next internal address : next DWORD

byte address :	+8	+9	+10	+11
Last data input :	valid	valid	valid	valid

dec/inc=10: decrementing DWORD aligned starting address

write 4 bytes within each DWORD in incrementing byte order
internal address is decremented by 4 each DWORD

starting address : yyyy
internal address : yyyy

byte address :	yyyy	+1	+2	+3
First data input :	valid	valid	valid	valid

next internal address : previous DWORD

byte address :	-4	-3	-2	-1
Second data input :	valid	valid	valid	valid

next internal address : previous DWORD				
byte address :	-8	-7	-6	-5
Last data input :	valid	valid	valid	valid

dec/inc=11: RESERVED

The above apply to full DWORD accesses starting at DWORD aligned addresses. See Section 9.3.2.2 for partial DWORD accesses or non-DWORD aligned starting addresses utilized during EtherCAT Direct Mapped mode.

DUMMY BYTES

In order to provide sufficient time for posted writes to be internally processed (especially when writing to the EtherCAT Core CSRs or Process RAM while in EtherCAT Direct Mapped mode) Dummy Byte cycles may be used. The number of Dummy Bytes is set using the Set Configuration instruction and is specified per write command type.

There are three values per instruction. The first is the number of Dummy Bytes that will precede the first data byte. The second is the number of Dummy Bytes that occur between bytes within a DWORD (intra-DWORD). The third is the number of Dummy Bytes that occur between DWORDs (inter-DWORD). There are no Dummy Bytes after the last data byte of a command.

APPLICATION NOTE: The number of Dummy Bytes between DWORDs is strictly the third configuration value. It is not in addition to or paralleled with a Dummy Byte count using the second configuration value.

APPLICATION NOTE: The DWORD boundary applies to all write commands even if the command is using BYTE buffering mode. The intra- and inter- number of Dummy Bytes can be set the same if appropriate.

APPLICATION NOTE: For BYTE buffering mode used in EtherCAT Direct Mapped Mode for EtherCAT core accesses, a Dummy byte configuration which has second parameter (intra-DWORD) equal to 0 and the third parameter (inter-DWORD) not equal to 0 is not supported. Normally the second and third parameters would be the same value in this case.

APPLICATION NOTE: The DWORD boundary is based on the address of the last byte written, not on the running byte count (i.e. it is not simply every fourth byte). This is important to consider during EtherCAT Direct Mapped Mode where the starting address could be non-DWORD aligned. Refer to Section 9.3.2.2 for how the address is updated.

APPLICATION NOTE: The number of clock cycles for a Dummy Byte varies based on the bit width(s) of the instruction.

9.2.6.1 Write

The Write instruction inputs the instruction code, the address, the possible initial Dummy Bytes, the data and any subsequent Dummy Byte(s) one bit per clock. In SQI mode, the instruction code, the address, the possible initial Dummy Byte(s), and the data bytes are input four bits per clock. This instruction is supported in SPI and SQI bus protocols with clock frequencies up to 80 MHz.

The SPI/SQI slave interface is selected by first bringing **SCS#** active. For SPI mode, the 8-bit WRITE instruction, 02h, is input into the **SI/SIO[0]** pin, followed by the two address bytes and 0 to 255 Dummy Byte(s). For SQI mode, the 8-bit WRITE instruction, 02h, is input into the **SIO[3:0]** pins, followed by the two address bytes and 0 to 255 Dummy Byte(s). The address bytes specify a BYTE address within the device.

The data follows the address bytes. For SPI mode, the data is input into the **SI/SIO[0]** pin starting with the msb of the LSB. For SQI mode the data is input nibble wide using **SIO[3:0]** starting with the msn of the LSB. The remaining bits/nibbles are shifted in on subsequent clock edges.

While in LAN9252 compatibility mode, the data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the **SCS#** is returned high, the write is considered invalid and the register is not affected.

While in EtherCAT Direct Mapped Mode, data writes to the EtherCAT Core CSRs and Process RAM may occur after BYTEs or completed DWORDs. Data writes to the non-EtherCAT Core CSRs continue to be DWORD aligned however incomplete DWORD writes may result in a corrupted register value.

The SCS# input is brought inactive to conclude the cycle.

Figure 9-13 illustrates a typical single and multiple register write for SPI mode. Figure 9-14 illustrates a typical single and multiple register write for SQI mode. A DWORD aligned full DWORD register is shown. The final DWORD is not followed by Dummy Bytes. A partial DWORD register write is also possible and would terminate after the last data BYTE without subsequent Dummy Byte(s). It is also possible for a partial DWORD write to start on a non-DWORD aligned boundary.

FIGURE 9-13: SPI WRITE

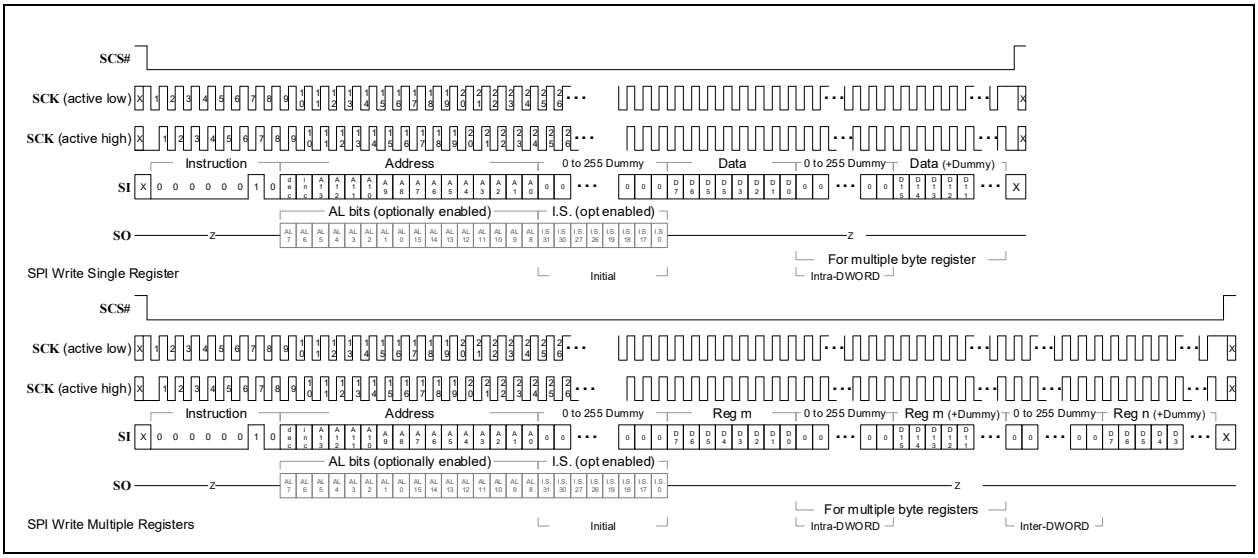
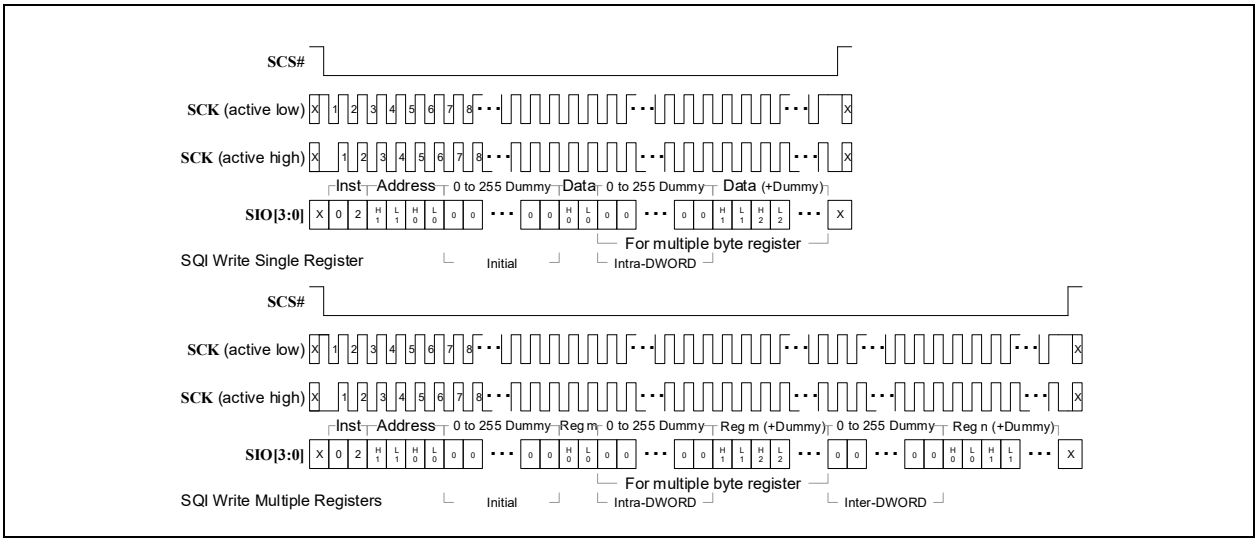


FIGURE 9-14: SQI WRITE



9.2.6.2 Dual Data Write

The SPI Dual Data Write instruction inputs the instruction code, the address and possible initial Dummy Bytes one bit per clock and inputs the data and any subsequent Dummy Byte(s) two bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SDDW instruction, 32h, is input into the **SIO[0]** pin, followed by the two address bytes and 0 to 255 Dummy Byte(s). The address bytes specify a BYTE address within the device.

The data follows the address bytes. The data is input into the **SIO[1:0]** pins starting with the msbs of the LSB. The remaining di-bits are shifted in on subsequent clock edges.

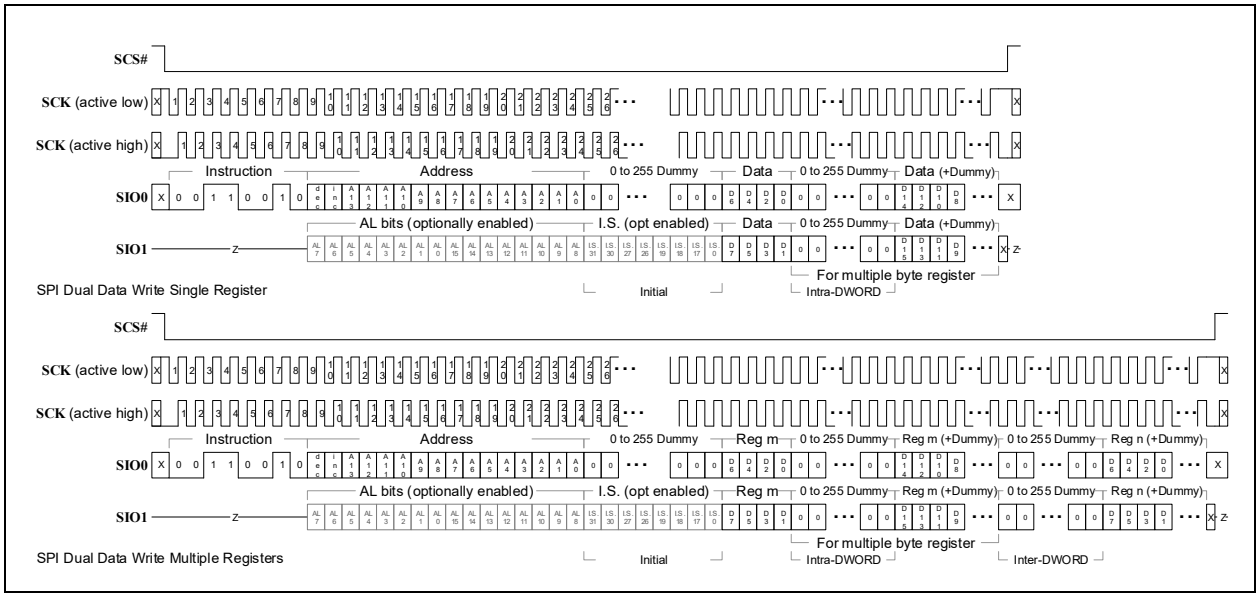
While in LAN9252 compatibility mode, the data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the **SCS#** is returned high, the write is considered invalid and the register is not affected.

While in EtherCAT Direct Mapped Mode, data writes to the EtherCAT Core CSRs and Process RAM may occur after BYTEs or completed DWORDs. Data writes to the non-EtherCAT Core CSRs continue to be DWORD aligned however incomplete DWORD writes may result in a corrupted register value.

The **SCS#** input is brought inactive to conclude the cycle.

Figure 9-15 illustrates a typical single and multiple register dual data write. A DWORD aligned full DWORD register is shown. The final DWORD is not followed by Dummy Bytes. A partial DWORD register write is also possible and would terminate after the last data BYTE without subsequent Dummy Byte(s). It is also possible for a partial DWORD write to start on a non-DWORD aligned boundary.

FIGURE 9-15: SPI DUAL DATA WRITE



9.2.6.3 Quad Data Write

The SPI Quad Data Write instruction inputs the instruction code, the address and the possible initial Dummy Bytes one bit per clock and inputs the data and any subsequent Dummy Byte(s) four bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SQDW instruction, 62h, is input into the **SIO[0]** pin, followed by the two address bytes and 0 to 255 Dummy Byte(s). The address bytes specify a BYTE address within the device.

The data follows the address bytes. The data is input into the **SIO[3:0]** pins starting with the msn of the LSB. The remaining nibbles are shifted in on subsequent clock edges.

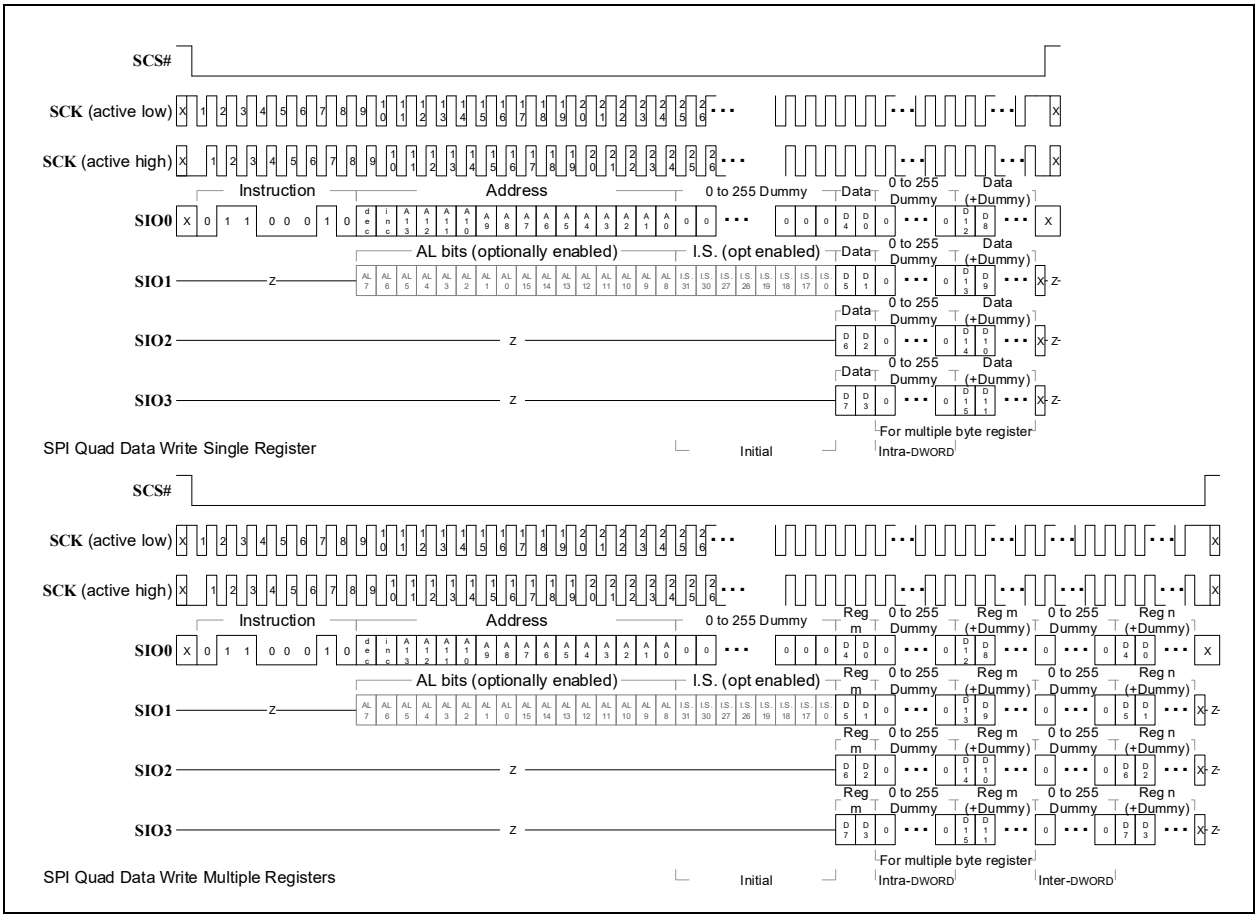
While in LAN9252 compatibility mode, the data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the **SCS#** is returned high, the write is considered invalid and the register is not affected.

While in EtherCAT Direct Mapped Mode, data writes to the EtherCAT Core CSRs and Process RAM may occur after BYTEs or completed DWORDs. Data writes to the non-EtherCAT Core CSRs continue to be DWORD aligned however incomplete DWORD writes may result in a corrupted register value.

The **SCS#** input is brought inactive to conclude the cycle.

Figure 9-16 illustrates a typical single and multiple register quad data write. A DWORD aligned full DWORD register is shown. The final DWORD is not followed by Dummy Bytes. A partial DWORD register write is also possible and would terminate after the last data BYTE without subsequent Dummy Byte(s). It is also possible for a partial DWORD write to start on a non-DWORD aligned boundary.

FIGURE 9-16: SPI QUAD DATA WRITE



9.2.6.4 Dual Address / Data Write

The SPI Dual Address / Data Write instruction inputs the instruction code one bit per clock and the address, the possible initial Dummy Bytes, the data and any subsequent Dummy Byte(s) two bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SDADW instruction, B2h, is input into the SIO[0] pin, followed by the two address bytes and 0 to 255 Dummy Byte(s) into the **SIO[1:0]** pins. The address bytes specify a BYTE address within the device.

The data follows the address bytes. The data is input into the **SIO[1:0]** pins starting with the msbs of the LSB. The remaining di-bits are shifted in on subsequent clock edges.

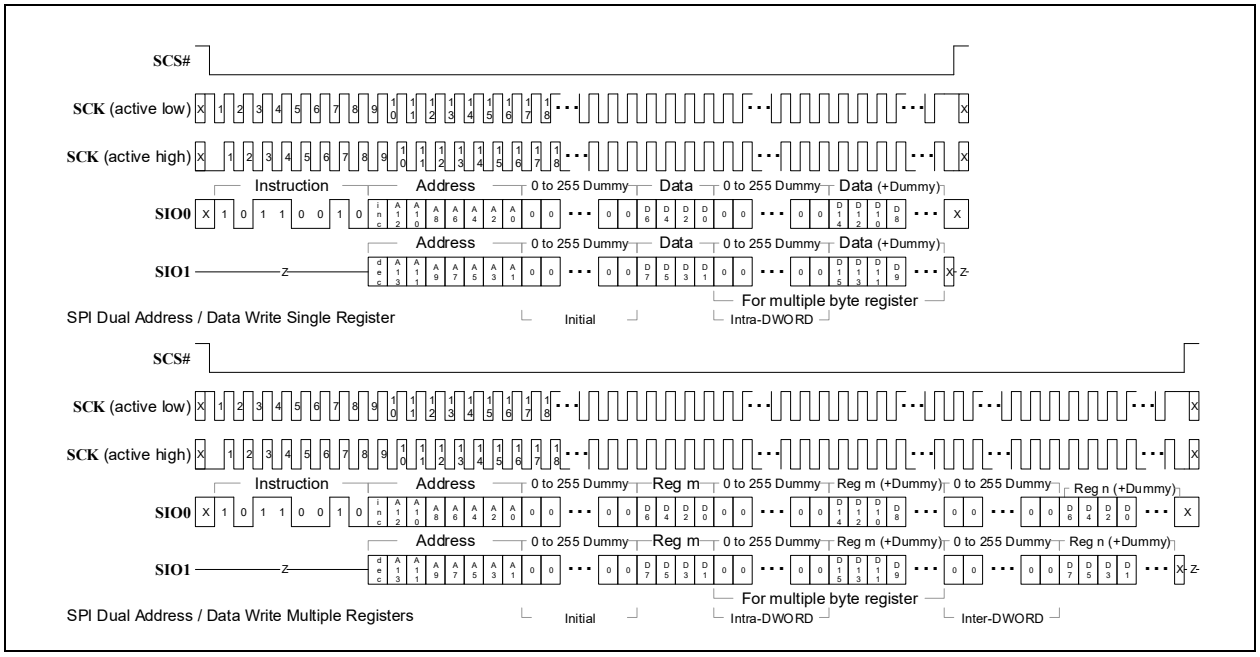
While in LAN9252 compatibility mode, the data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the **SCS#** is returned high, the write is considered invalid and the register is not affected.

While in EtherCAT Direct Mapped Mode, data writes to the EtherCAT Core CSRs and Process RAM may occur after BYTEs or completed DWORDs. Data writes to the non-EtherCAT Core CSRs continue to be DWORD aligned however incomplete DWORD writes may result in a corrupted register value.

The **SCS#** input is brought inactive to conclude the cycle.

Figure 9-17 illustrates a typical single and multiple register dual address / data write. A DWORD aligned full DWORD register is shown. The final DWORD is not followed by Dummy Bytes. A partial DWORD register write is also possible and would terminate after the last data BYTE without subsequent Dummy Byte(s). It is also possible for a partial DWORD write to start on a non-DWORD aligned boundary.

FIGURE 9-17: SPI DUAL ADDRESS / DATA WRITE



9.2.6.5 Quad Address / Data Write

The SPI Quad Address / Data Write instruction inputs the instruction code one bit per clock and the address, the possible initial Dummy Bytes, the data and any subsequent Dummy Byte(s) four bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SQADW instruction, E2h, is input into the **SIO[0]** pin, followed by the two address bytes and 0 to 255 Dummy Byte(s) into the **SIO[3:0]** pins. The address bytes specify a BYTE address within the device.

The data follows the address bytes. The data is input into the **SIO[3:0]** pins starting with the msn of the LSB. The remaining nibbles are shifted in on subsequent clock edges.

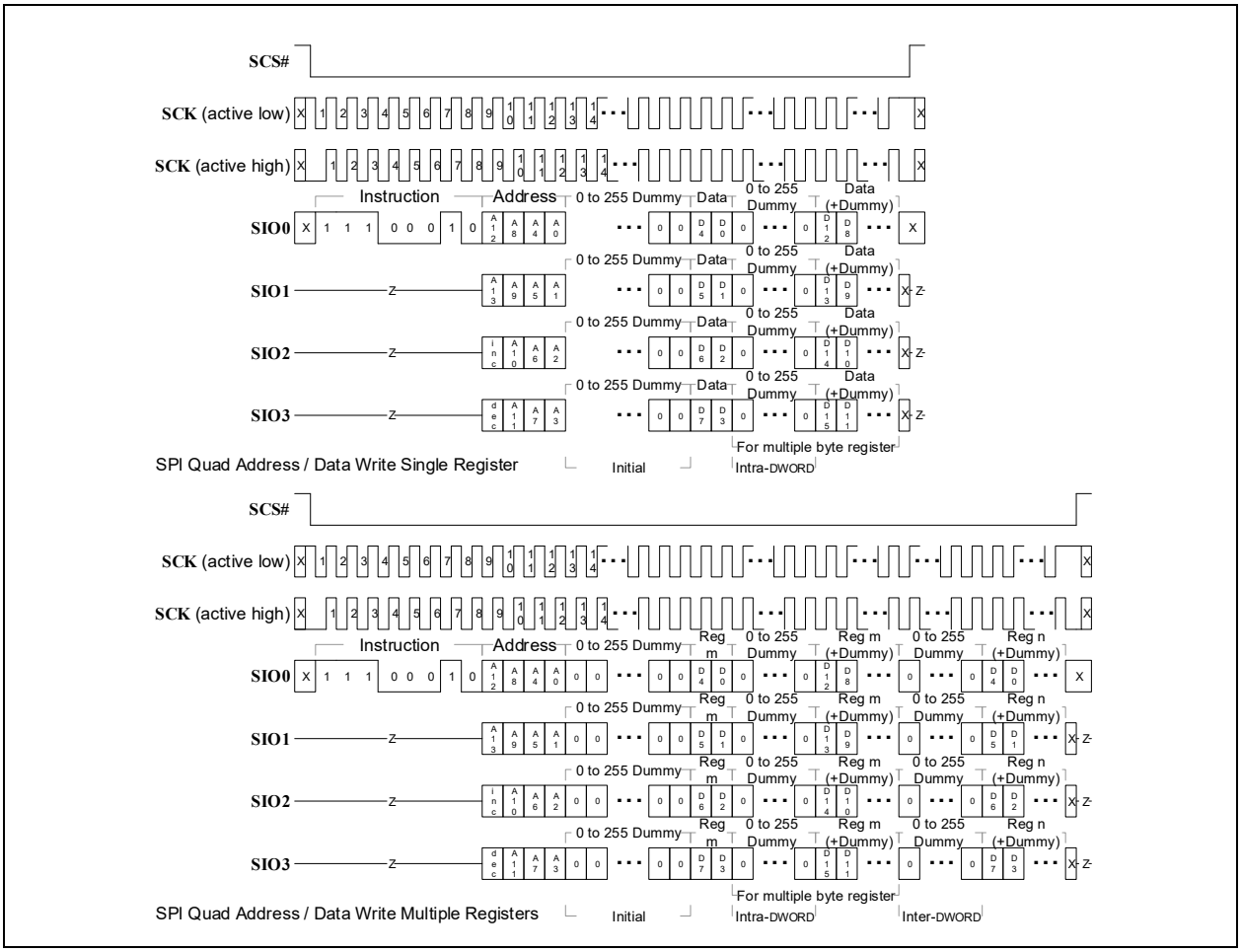
While in LAN9252 compatibility mode, the data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the **SCS#** is returned high, the write is considered invalid and the register is not affected.

While in EtherCAT Direct Mapped Mode, data writes to the EtherCAT Core CSRs and Process RAM may occur after BYTEs or completed DWORDs. Data writes to the non-EtherCAT Core CSRs continue to be DWORD aligned however incomplete DWORD writes may result in a corrupted register value.

The **SCS#** input is brought inactive to conclude the cycle.

Figure 9-18 illustrates a typical single and multiple register dual address / data write. A DWORD aligned full DWORD register is shown. The final DWORD is not followed by Dummy Bytes. A partial DWORD register write is also possible and would terminate after the last data BYTE without subsequent Dummy Byte(s). It is also possible for a partial DWORD write to start on a non-DWORD aligned boundary.

FIGURE 9-18: SPI QUAD ADDRESS / DATA WRITE



9.2.7 INTERRUPT REQUEST (AL EVENT REQUEST) AND INTERRUPT STATUS (INST_STS) REGISTER OUTPUT

During the address phase of the SPI Read, Fast Read, Dual Output Read, Quad Output Read, Write, Dual Data Write and Quad Data Write commands, the lower 16 bits of the AL Event Request register may be shifted out onto the **SO/SIO[1]** pin. The order is LSB first. This function is shown in the above diagrams.

Output of this data is enabled by the SPI AL Event Request & INT_STS Enable bit in the PDI Configuration Register. If not enabled, the SPI interface remains three-stated during those times.

During the first Dummy Byte (if present) following the address phase of the SPI Read, Fast Read, Dual Output Read, Quad Output Read, Write, Dual Data Write and Quad Data Write commands, bits 31:30, 27:26, 19:17 and 0 of the Interrupt Status Register (INT_STS) may be shifted out onto the **SO/SIO[1]** pin. This function is shown in the above diagrams.

9.3 EtherCAT Direct Mapped Mode

In EtherCAT Direct Mapped mode, the device address space is split between the EtherCAT Core CSRs and Process RAM (addresses 0h to 2FFFh) and the non-EtherCAT Core CSRs (addresses 3000h through 3FFFh). Access to the non-EtherCAT Core CSRs remains to be through the existing CSR bus interface. Access to the EtherCAT Core CSRs and Process RAM is no longer through the various EtherCAT CSR and Process RAM Command, Address, Length and Data registers but instead it is directly between the SPI slave and the EtherCAT Core.

SPI EtherCAT Direct Mapped mode is selected by the PDI Control Register per the values in Table 13-2, "PDI Mode Selection".

9.3.1 NON-ETHERCAT CORE CSRS

The non-EtherCAT Core CSRs continue to be accessed as they would be in LAN9252 compatibility mode.

SPI commands must not be started if there is a pending prior EtherCAT Core write. Upon the activation of **SCS#**, in the absence of a pending prior write, **WAIT_ACK** will indicate acknowledge (not busy). If an EtherCAT Core write operation is internally pending, the **WAIT_ACK** will initially indicate wait. The host may deactivate **SCS#** and retry the cycle at a later time, or it may simply wait until **WAIT_ACK** indicates not busy. Once not busy, **WAIT_ACK** will indicate acknowledge through the rest of the cycle.

Note:	Non-EtherCAT Core CSR accesses do not create wait states using the wait indication on **WAIT_ACK**. However, the minimum Initial Dummy Byte values listed in Table 9-1 and Table 9-2 must still be used. These are the default values per Table 9-3. Higher values may be used if Dummy Bytes are used for EtherCAT Core CSRs and Process RAM accesses per the Time or Dummy Cycle Based section. Waiting the equivalent time in lieu of using the Initial Dummy bytes is also not acceptable.

Registers are DWORD aligned and are a DWORD in length. Non- DWORD aligned / non-DWORD length access is not supported. The address provided to the CSRs and interface logic is forced to be DWORD aligned.

9.3.1.1 Reads

The data read(s) from the register(s) occur(s) initially and then after every 32-bits. Registers that are affected by a read operation are updated every 32-bits (once the current DWORD output shift has started). Multiple reads are DWORD oriented.

To avoid internally prefetching additional data past the last data that will be output, two methods are utilized. For the READ instruction, the **SI** input is used as described in Section 9.2.5.1. For other read commands (FASTREAD, SDOR, SDIOR, SQOR, SQIOR), the data length in bytes is used as described in Section 9.2.5.2 through Section 9.2.5.6.

Auto-Increment / Decrement

Auto-increment/decrement operation for multiple DWORDs remains as described in Section 9.2.5.

9.3.1.2 Writes

The data write(s) to the register(s) occur(s) after each 32-bits are shifted in. In the event that 32-bits are not written when **SCS#** is returned high, a write may occur and the register may be corrupted. Multiple writes are DWORD oriented.

Auto-Increment / Decrement

Auto-increment/decrement operation for multiple DWORDs remains as described in Section 9.2.6.

9.3.2 ETHERCAT CORE CSRS AND PROCESS RAM

Registers / Process RAM accesses may have any alignment and length (up to the limit of the read transfer length field if applicable).

9.3.2.1 Reads

Depending on the read command used, the data reads from the EtherCAT Core occur every 8-bits or every DWORD.

It is the responsibility of the Host to only shift the output data when it is available, as described below.

READ Command:

The initial data request occurs once the address portion of the command has been shifted in. Subsequent data requests occur when the previous data shift output starts. The READ instruction lacks the information to indicate how many bytes of a DWORD the host will shift out. Therefore for the READ instruction, data is requested one byte at a time.

To avoid requesting additional data past the last data that will be output, the **SI** input is used as a read termination as described in Section 9.2.5.1.

FASTREAD, SDOR, SDIOR, SQOR, SQIOR Commands:

The initial data request occurs once the transfer length portion of the command has been shifted in. In order to provide a consistent time to pre-fetch the next DWORD, 0 to 3 padding bytes (and potentially Dummy Bytes per padding byte) are pre-pended before the first valid data, achieving a DWORD aligned output. The number of bytes is based on the starting address. Subsequent data requests occur when the current data or the first padding byte shift output starts.

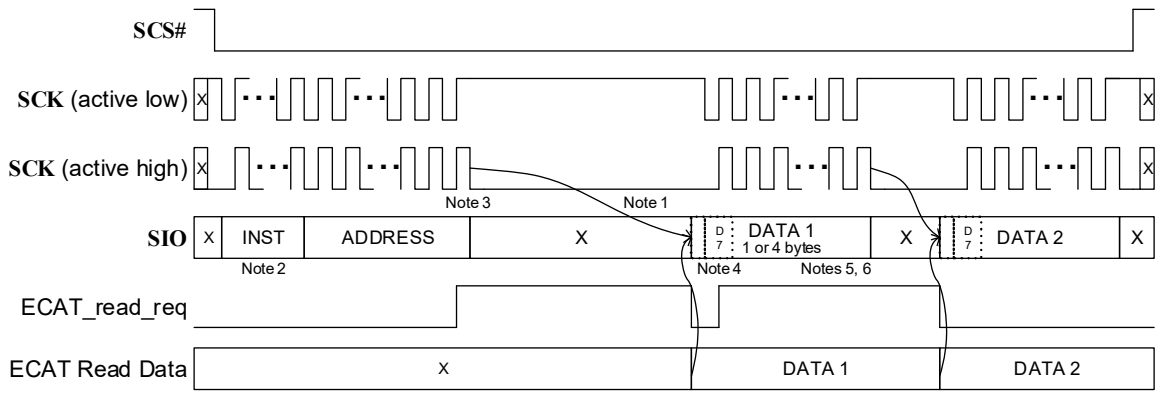
The FASTREAD, SDOR, SDIOR, SQOR and SQIOR instructions contain the transfer length and, along with the starting address, is used to calculate the number of bytes initially requested. The remaining byte count is used to calculate the number of bytes subsequently requested, avoiding requesting additional data past the last data that will be output.

Special Handling of First Bit(s):

An active high SCK consists of a series of pulses and will stop low (on a falling edge) following the last address, transfer length or Dummy Byte input. This falling edge corresponds to that which outputs the first data bit(s). However, it is likely that valid data has not yet been read from the EtherCAT Core.

To account for this, the first bit(s) are output asynchronously once data is read from the EtherCAT Core. This is illustrated in Figure 9-19.

FIGURE 9-19: ACTIVE HIGH SCK FIRST DATA BIT(S)



Note 1: Active high SCK - First data bit is output asynchronously since falling edge already occurred.

Note 2: INST is either READ, FASTREAD, SDOR, SDIOR, SQOR or SQIOR.

Note 3: ADDRESS could be followed by the transfer length and / or Dummy bytes. The last clock before the data output is the one of concern.

Note 4: BYTE access shown (READ command). DWORD access (FASTREAD, SDOR, SDIOR, SQOR and SQIOR commands) could be pre-pended by pad bytes but the same falling edge is still the one of concern.

Note 5: Falling edge of concern occurs per byte for READ command and per DWORD for FASTREAD, SDOR, SDIOR, SQOR and SQIOR commands.

Note 6: DATA could be followed by Dummy bytes. The last clock before the next byte (READ) or DWORD (others) output is the one of concern.

Auto-Increment / Decrement

Constant address and Auto-increment/decrement read operations are either BYTE or DWORD oriented depending on the command used (READ vs FASTREAD, SDOR, SDIOR, SQOR, SQIOR).

For DWORD oriented, the DWORD address remains constant or is incremented or decremented while the byte address within the DWORD always increments. For Auto-increment operation non-DWORD alignment and partial DWORDs are supported. For Auto-decrement and Constant address operation only DWORD aligned starting addresses and full DWORD transfers are supported.

READ Command (BYTE oriented):

dec/inc=00: constant BYTE address

read BYTE
internal address repeats each BYTE

starting address : yyyy
internal address : yyyy
byte address : yyyy
First data output : valid

next internal address : yyyy
byte address : yyyy
Second data output : valid

next internal address : yyyy
byte address : yyyy
Last data output : valid

dec/inc=01: incrementing BYTE address

read BYTE
internal address is incremented to next BYTE

starting address : yyyy
internal address : yyyy
byte address : yyyy
First data output : valid

next internal address : next BYTE
byte address : +1
Second data output : valid

next internal address : next BYTE
byte address : +2
Last data output : valid

dec/inc=10: decrementing BYTE address
read BYTE
internal address is decremented to previous BYTE

starting address : yyyy
internal address : yyyy
byte address : yyyy
First data output : valid

next internal address : previous BYTE
byte address : -1
Second data output : valid

next internal address : previous BYTE
byte address : -2
Last data output : valid

dec/inc=11: RESERVED

FASTREAD, SDOR, SDIOR, SQOR, SQIOR Commands (DWORD oriented):

dec/inc=00: constant DWORD aligned address
read 4 bytes within the DWORD in incrementing byte order
internal address repeats each DWORD

starting address : yyyy

internal address : yyyy				
byte address :	yyyy	+1	+2	+3
First data output :	valid	valid	valid	valid
internal address : yyyy				
byte address :	yyyy	+1	+2	+3
Second data output :	valid	valid	valid	valid
internal address : yyyy				
byte address :	yyyy	+1	+2	+3
Last data output :	valid	valid	valid	valid

dec/inc=01: incrementing DWORD address
read up to 4 bytes within each DWORD in incrementing byte order
internal address is incremented to next DWORD each DWORD

starting address : yyyy

internal address : DWORD address of yyyy

byte address* :	yyyy	+1	+2	+3
		yyyy	+1	+2
			yyyy	+1
				yyyy
First data output* :	valid	valid	valid	valid
	pad*	valid	valid	valid
	pad*	pad*	valid	valid
	pad*	pad*	pad*	valid

next internal address : next DWORD

byte address :	+4	+5	+6	+7
	+3	+4	+5	+6
	+2	+3	+4	+5
	+1	+2	+3	+4
Second data output :	valid	valid	valid	valid

next internal address : next DWORD

byte address :	+8	+9	+10	+11
	+7	+8	+9	+10
	+6	+7	+8	+9
	+5	+6	+7	+8
Last data output** :	valid	valid/na**	valid/na**	valid/na**

*first data output could start with pad bytes if non-DWORD aligned starting address
-- transfer length does not include any pad bytes
**last data output could be less than 4 bytes

dec/inc=10: decrementing DWORD aligned starting address

read 4 bytes within each DWORD in incrementing byte order
internal address is decremented by 4 each DWORD

starting address : yyyy
internal address : yyyy

byte address :	yyyy	+1	+2	+3
First data output :	valid	valid	valid	valid

next internal address : previous DWORD

byte address :	-4	-3	-2	-1
Second data output :	valid	valid	valid	valid

next internal address : previous DWORD

byte address :	-8	-7	-6	-5
Last data output :	valid	valid	valid	valid

dec/inc=11: RESERVED

9.3.2.2 Writes

The data write(s) to the EtherCAT Core occur either every 8-bits or after the last BYTE of a DWORD is shifted in. The latter achieves better throughput but requires more post command processing time, potentially delaying the next command.

Multiple writes can be BYTE or DWORD oriented.

Based on the SPI command's increment / decrement bits, the SPI interface can dynamically accumulate and write up to 32 bits, multiple times, in one SPI bus cycle. In the case where less than 4 bytes are shifted in before the bus cycle ends, the SPI interface commits whatever amount of data it received.

BYTE buffering is used normally.

DWORD buffering is indicated by using the normally reserved auto decrement / increment value of 'b11.

It is the responsibility of the Host to only shift in the next data when the device is ready. This is described in the following subsection.

Auto-Increment / Decrement

Constant address and Auto-increment/decrement write operations are either BYTE or DWORD oriented depending on the buffering type used (BYTE buffering vs DWORD buffering).

For DWORD buffering, the DWORD address remains constant or is incremented or decremented while the byte address within the DWORD always increments. Only Auto-increment operation is supported with DWORD buffering.

dec/inc=00: constant BYTE address

write BYTE
internal address repeats each BYTE

starting address : yyyy
internal address : yyyy
byte address : yyyy
First data input : valid

next internal address : yyyy
byte address : yyyy
Second data input : valid

next internal address : yyyy
byte address : yyyy
Last data input : valid

dec/inc=01: incrementing BYTE address

write BYTE
internal address is incremented to next BYTE

starting address : yyyy
internal address : yyyy
byte address : yyyy
First data input : valid

next internal address : next BYTE
byte address : +1
Second data input : valid

next internal address : next BYTE
byte address : +2
Last data input : valid

dec/inc=10: decrementing BYTE address

write BYTE
internal address is decremented to previous BYTE

starting address : yyyy
internal address : yyyy
byte address : yyyy
First data input : valid

next internal address : previous BYTE
byte address : -1
Second data input : valid

next internal address : previous BYTE
byte address : -2
Last data input : valid

dec/inc=11: incrementing DWORD address

write up to 4 bytes within each DWORD in incrementing byte order
internal address is incremented to next DWORD each DWORD

starting address : yyyy
internal address : DWORD address of yyyy

byte address* :	yyyy	+1	+2	+3
	yyyy	+1	+2	na*
	yyyy	+1	na*	na*
	yyyy	na*	na*	na*
First data input* :	valid	valid	valid	valid
	valid	valid	valid	na*
	valid	valid	na*	na*
	valid	na*	na*	na*

next internal address : next DWORD

byte address :	+4	+5	+6	+7
	+3	+4	+5	+6
	+2	+3	+4	+5
	+1	+2	+3	+4
Second data input :	valid	valid	valid	valid

next internal address : next DWORD

byte address :	+8	+9	+10	+11
	+7	+8	+9	+10
	+6	+7	+8	+9
	+5	+6	+7	+8
Last data input** :	valid	valid/na**	valid/na**	valid/na**

*first data could be less than 4 bytes if non-DWORD aligned starting address
**last data output could be less than 4 bytes

9.3.2.3 Wait States

EtherCAT Direct Mapped mode requires the host bus to obey the access arbitration controlled by the EtherCAT Core PDI.

Subsequent SPI commands must not be started if there is a pending prior EtherCAT Core write.

Read cycles are first arbitrated and data is returned. This requires a delay for all reads. In order to guarantee valid data is returned, the shifting out of the read data must not occur before the worst case access time. Sequential data is prefetched during the initial read data shift. however the shifting out of the subsequent read data must not occur before the worst case access time.

Write cycles are posted and executed after the data has been shifted in. For multiple writes, subsequent write cycles may start to be shifted in but the bus speed must be limited such that the subsequent write cycle does not complete until after the worst case write arbitration time. When DWORD Buffering is used, BYTES within the same DWORD may be shifted in without any delay. However with DWORD Buffering, in the case of a partial DWORD being written on the final write, **SCS#** must not return high until the prior write has completed.

Besides simply waiting the required time or using a sufficiently slow SPI clock speed, two other methods are available for the host. Dummy Byte cycles are available for all read and write commands to pace the data rate.The wait / acknowledge (**WAIT_ACK**) pin may be used as an indication when read data may be shifted out or when the write data shift in may complete.

Time or Dummy Cycle Based

Dummy Byte cycles are available for all read and write commands. A programmable number of Dummy Bytes can be set per command type as well as for the first, per BYTE or per DWORD of data. The number of Dummy Bytes to be used depends on the SPI clock speed as well as the SPI command used.

Note:	Dummy Byte cycles may coexist with the Wait / Acknowledge operation described in the following subsection. It is intended that the Dummy Bytes are shifted while Wait is indicated. The Dummy Bytes are a means to fill the required time indicated by the Wait indication. Waiting for Wait to be negated before shifting the Dummy Bytes would nullify the need for the them.

The following timing constraints must be followed:

Note:	The timing values in this section are preliminary and subject to change.

SPI commands must not be started if there is a pending prior EtherCAT Core write. The worst case internal write cycle time is 250ns for a single byte plus 80ns per additional byte.

Read first BYTE - From rising edge of **SCK** that samples last address bit (or last transfer length bit if applicable) to data ready - READ command 370ns, other read commands 610ns. A falling edge on **SCK** may occur prior to this time in which case data is output when it becomes ready. See Section 9.3.2.1.

Note:	The “other” read commands are optimized for multiple BYTE reads. Although they have a longer initial access time as well as the overhead of the transfer length byte(s) they may outperform the standard READ command.

Read next BYTE - From rising edge of **SCK** that follows the falling edge of **SCK** that outputs first data bit of current BYTE to next BYTE ready - READ command 370ns, other read commands 0ns (“read next DWORD” timing still must be meet). Some or all of this wait time might overlap with the time it takes to shift out the current BYTE.

Read next DWORD - From rising edge of **SCK** that follows the falling edge of **SCK** that outputs first data bit of current DWORD to next DWORD ready - READ command n/a, other read commands 610ns. Some or all of this wait time might overlap with the time it takes to shift out the current DWORD.

Write first BYTE (BYTE Buffering mode) - to rising edge of **SCK** that samples last data bit - 0ns.

Write first DWORD (DWORD Buffering mode) - to rising edge of **SCK** that samples last data bit - 0ns.

Write next BYTE (BYTE Buffering mode) - from rising edge of **SCK** that samples last data bit of current BYTE to rising edge of **SCK** that samples last data bit of next BYTE - 250ns. Some or all of this wait time might overlap with the time it takes to shift in the next BYTE.

Write next full DWORD (DWORD Buffering mode) - from rising edge of **SCK** that samples last data bit of current DWORD to rising edge of **SCK** that samples last data bit of next full DWORD - 490ns. Some or all of this wait time might overlap with the time it takes to shift in the next DWORD,

Write last partial DWORD (DWORD Buffering mode) - from rising edge of **SCK** that samples last data bit of current DWORD to rising edge of **SCS#** which forces write of last partial DWORD - 490ns. Some or all of this wait time might overlap with the time it takes to shift in the partial DWORD.

Wait / Acknowledge Operation

The host system may either wait the specified above worst case access time, or may use the **WAIT_ACK** signal.

Note:	Wait / Acknowledge may coexist with the Dummy Byte cycles operation described above.

Read and write cycles start with the leading edge of **SCS#**, which enables the **WAIT_ACK** output.

If an EtherCAT Core write operation is internally pending, the **WAIT_ACK** will initially indicate wait. In the absence of a pending prior write, **WAIT_ACK** will initially indicate acknowledge (not busy).

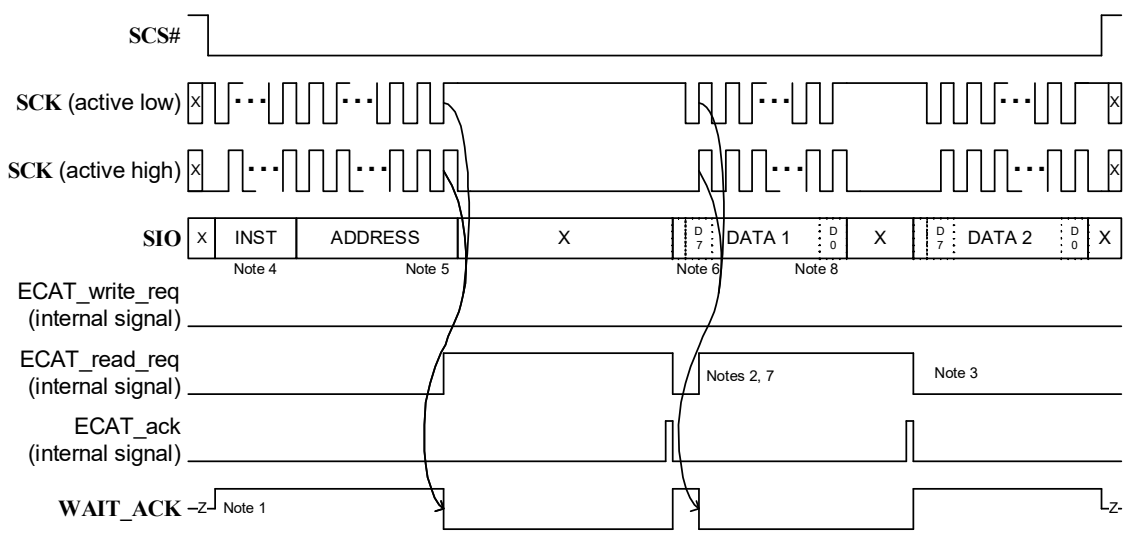
SPI commands must not be started if there is a pending prior EtherCAT Core write. The worst case internal write cycle time is 250ns for a single byte plus 80ns per additional byte. The host may deactivate **SCS#** and retry the cycle at a later time, or it may simply wait until **WAIT_ACK** indicates not busy.

For reads from the EtherCAT Core CSRs and Process RAM (addresses 0h to 2FFFh), following the shift in of the address, **WAIT_ACK** will change to indicate wait as the SPI interface retrieves the read data from the EtherCAT Core. Once the read data is available, **WAIT_ACK** will indicate acknowledge and the host may shift out the data.

Note:	Per “Special Handling of First Bit(s)” in Section 9.3.2.1, the first falling SCK edge of each data may occur before data is ready, in which case data will be output asynchronously when ready.

Following the shift out of the first bit of data (on the rising edge following the falling edge that outputs the data), **WAIT_ACK** will indicate wait once again, as the SPI interface pre-fetches the next read data.(assuming there is a next pre-fetch otherwise **WAIT_ACK** will remain indicating acknowledge).

FIGURE 9-20: SPI WAIT ACK ETHERCAT CORE READ



Note 1: **WAIT_ACK** initially high due to no pending internal write.

Note 2: Next read request along with next **WAIT_ACK** starts on rising edge after first data shift started.

Note 3: 3rd read request not performed, therefore **WAIT_ACK** remains inactive.

Note 4: INST is either READ, FASTREAD, SDOR, SDIOR, SQOR or SQIOR.

Note 5: ADDRESS could be followed by the transfer length and / or Dummy bytes. **WAIT_ACK** becomes active on the last address or transfer length input. Any Dummy bytes occur in parallel with **WAIT_ACK**.

Note 6: BYTE access shown (READ command). DWORD access (FASTREAD, SDOR, SDIOR, SQOR and SQIOR commands) could be pre -pended by pad bytes.

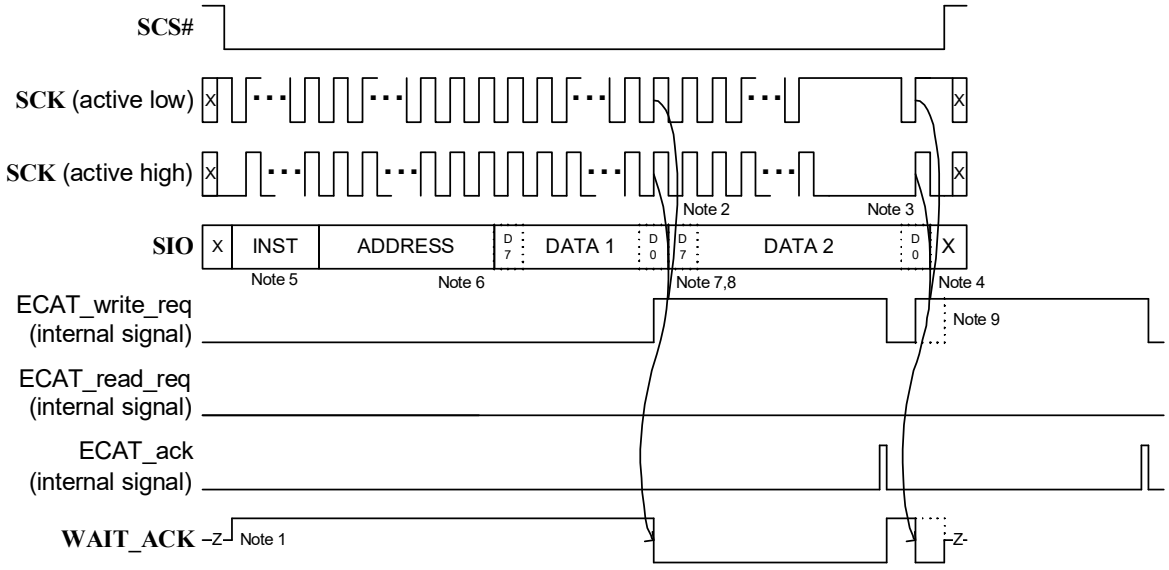
Note 7: **WAIT_ACK** occurs per byte (READ) or per DWORD (others).

Note 8: DATA could be followed by Dummy bytes which would occur in parallel with **WAIT_ACK**.

Note: Active High Push-Pull **WAIT_ACK** shown.

For writes to the EtherCAT Core CSRs and Process RAM (addresses 0h to 2FFFh), depending on the write buffering mode, **WAIT_ACK** will change to indicate wait following the shift in of the last data bit of each BYTE or of each (potentially partial) DWORD. Following the last data bit shift in, an internal write operation is initiated. Once the internal write operation has completed, **WAIT_ACK** will indicate acknowledge. The host may start the next data input during the internal write wait time, however it may only complete the BYTE / DWORD when acknowledge is indicated.

FIGURE 9-21: SPI WAIT ACK ETHERCAT CORE WRITE



Note 1: **WAIT_ACK** initially high due to no pending internal write.

Note 2: Host can start next data write shift during internal write.

Note 3: Host must wait before finishing next data write.

Note 4: Host can finish SPI cycle with final internal write pending.

Note 5: INST is either WRITE, SDDW, SDADW, DQDW or SQADW.

Note 6: ADDRESS could be followed by Dummy bytes.

Note 7: DATA could be followed by Dummy bytes which would occur in parallel with **WAIT_ACK**.

Note 8: BYTE Buffering mode shown. DWORD Buffering mode is the same except internal write and WAIT occurs at the end of a DWO RD (bit D24).

Note 9: For DWORD Buffering mode, internal write occurs on **SCS#** inactive if last bit is not D24.

Note: Active High Push -Pull **WAIT_ACK** shown.

WAIT_ACK becomes undriven with the negation of **SCS#**.

9.4 Controller Access Errors

The following access errors are detected by the SPI interface:

- A SPI command started while a prior write was pending (EtherCAT Direct Mapped Mode only).
- For non-EtherCAT Core CSR access, the number of clock cycles during the data phase of the transfer does not align to a DWORD multiple (incomplete DWORDs were transferred).
- For EtherCAT Core CSR and Process RAM access, the number of clock cycles during the data phase of the transfer does not align to a byte multiple (incomplete BYTEs were transferred) (EtherCAT Direct Mapped Mode only).
- For a READ instruction, the data phase was not terminated by setting **SI** high for the last byte.
- For a READ instruction, additional bytes were read after setting **SI** high for the last byte.
- For the FASTREAD, SDOR, SDIOR, SQOR, SQIOR instructions, the incorrect number of bytes were read (not matching the byte length provided).
- For a EtherCAT Core CSR and Process RAM read access (EtherCAT Direct Mapped Mode only):

  a) For the initial access, a rising SCK following the last address, transfer length or Dummy Byte input occurred while the interface was busy fetching the data.

  b) For subsequent accesses, a rising SCK following the last current data or Dummy Byte output occurred while the interface was busy fetching the next data.

  (Testing on the rising **SCK** allows for a falling edge on the address, transfer length, Dummy Bytes or last bit of the current data before the next data is ready for the active high SCK case - see Section 9.3.2.1).
- For a write access, the last data clock cycle of the current BYTE (BYTE Buffering) or DWORD (DWORD Buffering) occurred while a prior write was pending (EtherCAT Direct Mapped Mode only).
- For a write access, **SCS#** became inactive (committing the current partial DWORD write) while a prior write was pending (EtherCAT Direct Mapped Mode only).

The PDI Error Counter Register will be incremented and the reason of the access error can be read in the PDI Error Code Register. The PDI Error Code Register bit definitions are detailed in Section 11.16.40.

Note:	A transfer may contain multiple errors. Sequential errors are counted individually as they occur. The error status is updated as each error occurs, overwriting any previous error status. Simultaneous errors that occur at the end of the transfer (“Incomplete BYTE or DWORD”, “Read finished without setting **SI** high for the last byte” and “Actual read length did not match byte length provided”) cause the PDI Error Counter Register to increment only once more. The error status will still overwrite any previous error status. however, for multiple simultaneous errors, the error status may indicate multiple conditions.

9.5 SPI/SQI Timing Requirements

SPI/SQI interface pin timing is described below. Data access time for EtherCAT Direct Mapped Mode is described in Section 9.3.2.3, Wait States.

FIGURE 9-22: SPI/SQI INPUT TIMING

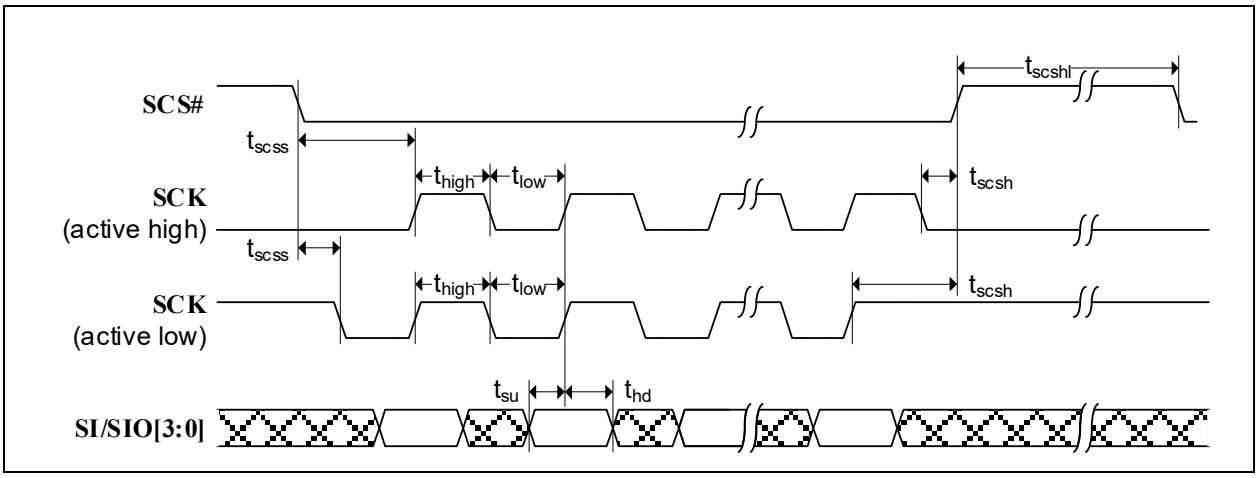


FIGURE 9-23: SPI/SQI OUTPUT TIMING

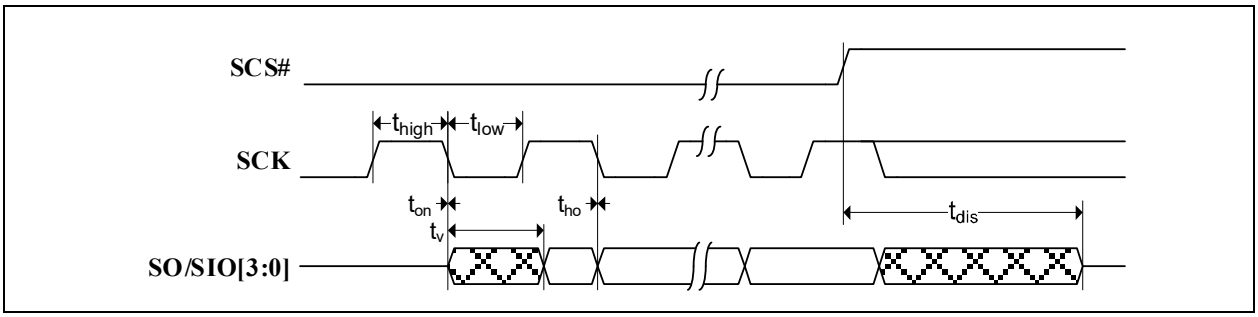


FIGURE 9-24: SPI/SQI WAIT_ACK TIMING

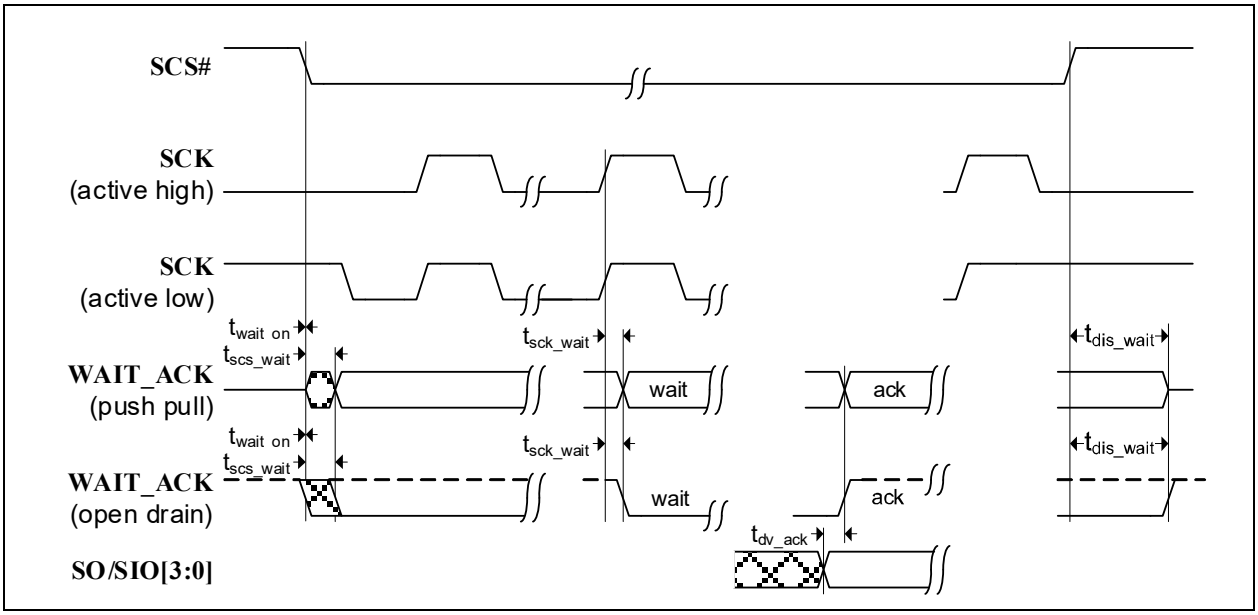


TABLE 9-4: SPI/SQI TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
f_{sck}	**SCK** clock frequency Note 11			30 / 80	MHz
t_{high}	**SCK** high time	5.5			ns
t_{low}	**SCK** low time	5.5			ns
t_{scss}	**SCS#** setup time to **SCK**	5			ns
t_{scsh}	**SCS#** hold time from **SCK**	5			ns
t_{scshl}	**SCS#** inactive time	50			ns
t_{su}	Data input setup time to **SCK**	3			ns
t_{hd}	Data input hold time from **SCK**	4			ns
t_{on}	Data output turn on time from **SCK**	0			ns
t_v	Data output valid time from **SCK** Note 12 **Application Note:** Depending on the clock frequency and pulse width, data may not be valid until following the next rising edge of **SCK**. This is normal with high speed SPI slaves (SPI flashes, etc.). The host SPI controller would need to delay the sampling of the data by either a fixed time or by using the falling edge of **SCK**.			3.3V **VDDIO** 30pF: 9 10pF: 8.5	ns
				1.8V **VDDIO** 30pF: 13 10pF: 12	
t_{ho}	Data output hold time from **SCK**	0			ns
t_{dis}	Data output disable time from **SCS#** inactive			20	ns
t_{wait_on}	**WAIT_ACK** turn on time from **SCS#** active	0			ns
t_{scs_wait}	**WAIT_ACK** valid from **SCS#** active			15	ns
t_{dis_wait}	**WAIT_ACK** disable time from **SCS#** inactive			20	ns
t_{sck_wait}	**WAIT_ACK** assertion from **SCK**			15	ns
t_{dv_ack}	Read Data asynchronous output valid before **WAIT_ACK** de-assertion	15			ns

Note 11: The Read instruction is limited to 30 MHz maximum.

Note 12: Depends on loading and supply voltage.

10.0 ETHERNET PHYS

10.1 Functional Overview

The device contains PHYs A and B, which are identical in functionality. PHY A connects to either port 0 or 2 of the EtherCAT Core. PHY B connects to port 1 of the EtherCAT core. These PHYs interface with their respective MAC via an internal MII interface.

The PHYs comply with the IEEE 802.3 Physical Layer for Twisted Pair Ethernet specification and can be configured for full/half duplex 100 Mbps (100BASE-TX) or 10 Mbps (10BASE-T) Ethernet operation. However, only full duplex, 100BASE-TX operation is used for EtherCAT. All PHY registers follow the IEEE 802.3 (clause 22.2.4) specified MII management register set and are fully configurable.

10.1.1 PHY ADDRESSING

Each individual PHY is assigned a default PHY address. The address for PHY A is set to 0 or 2, based on the device mode. The address for PHY B is fixed to 1.

In addition, the addresses for the PHYs can be changed via the PHY Address (PHYADD) field in the PHY x Special Modes Register (PHY_SPECIAL_MODES_x).

10.2 PHYs A & B

The device integrates two IEEE 802.3 PHY functions. The PHYs are configured for 100 Mbps copper (100BASE-TX) Ethernet operation and include Auto-Negotiation and HP Auto-MDIX.

Note:	Because PHYs A and B are functionally identical, this section will describe them as “PHY x”, or simply “PHY”. Wherever a lowercase “x” has been appended to a port or signal name, it can be replaced with “A” or “B” to indicate the PHY A or PHY B respectively. In some instances, a “1” or a “2” may be appropriate instead. All references to “PHY” in this section can be used interchangeably for both the PHYs A and B.

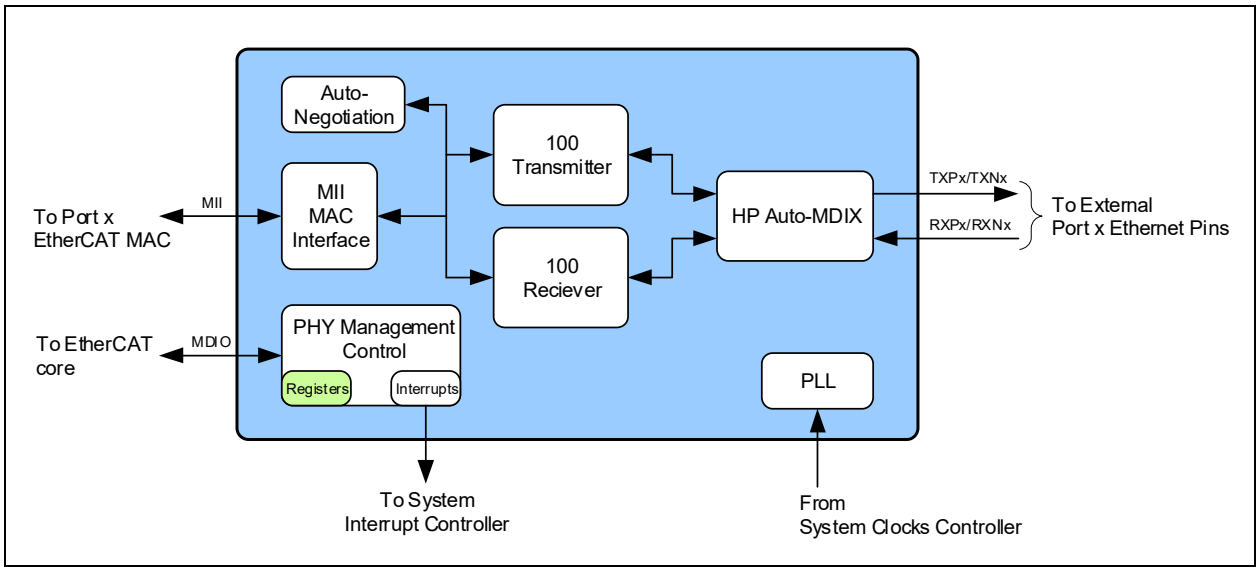
10.2.1 FUNCTIONAL DESCRIPTION

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

- 100BASE-TX Transmit and 100BASE-TX Receive
- Auto-Negotiation
- HP Auto-MDIX
- PHY Management Control and PHY Interrupts
- PHY Power-Down Modes
- Wake on LAN (WoL)
- Resets
- Link Integrity Test
- Cable Diagnostics
- Loopback Operation

A block diagram of the main components of each PHY can be seen in Figure 10-1.

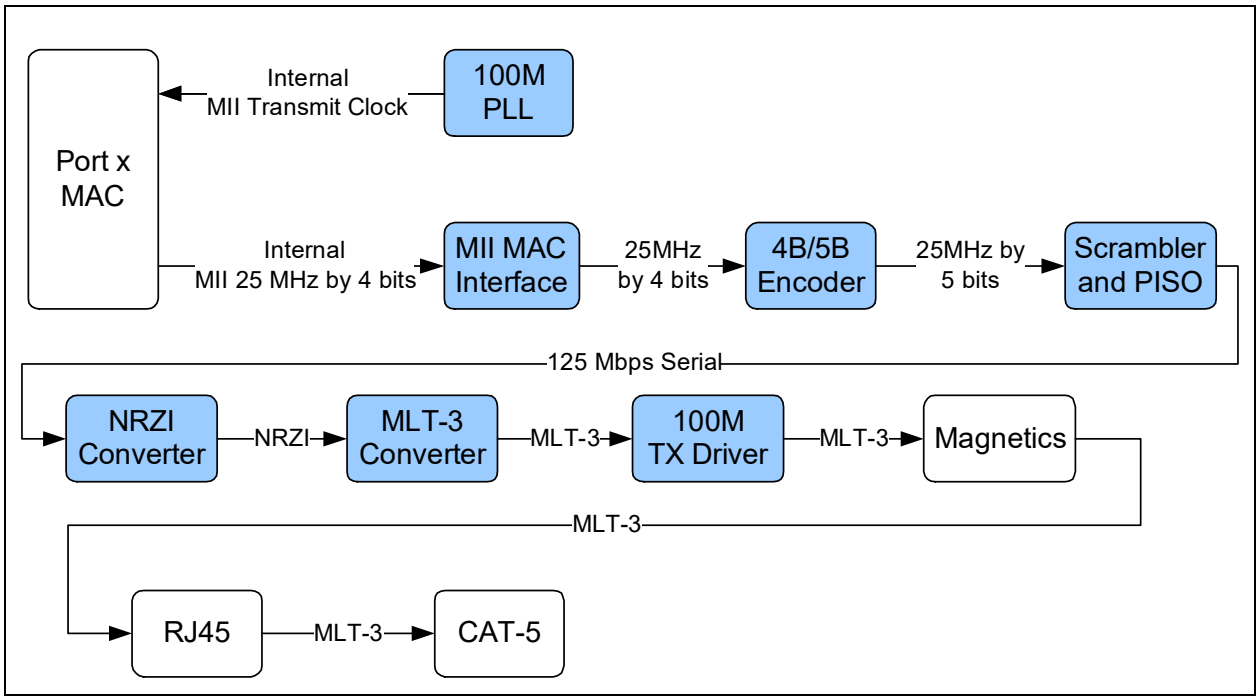
FIGURE 10-1: PHY BLOCK DIAGRAM



10.2.2 100BASE-TX TRANSMIT

The 100BASE-TX transmit data path is shown in Figure 10-2. Shaded blocks are those which are internal to the PHY. Each major block is explained in the following sections.

FIGURE 10-2: 100BASE-TX TRANSMIT DATA PATH



10.2.2.1 100BASE-TX Transmit Data Across the Internal MII Interface

For a transmission, the EtherCAT Core MAC drives the transmit data onto the internal MII TXD bus and asserts the internal MII TXEN to indicate valid data. The data is in the form of 4-bit wide 25 MHz data.

10.2.2.2 4B/5B Encoder

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 10-1. 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.

TABLE 10-1: 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	
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 the MII Transmitter Enable signal (TXEN) is received		
11000	/J/	First nibble of SSD, translated to “0101” following IDLE, else MII Receive Error (RXER)			Sent for rising MII Transmitter Enable signal (TXEN)		
10001	/K/	Second nibble of SSD, translated to “0101” following J, else MII Receive Error (RXER)			Sent for rising MII Transmitter Enable signal (TXEN)		
01101	/T/	First nibble of ESD, causes de-assertion of CRS if followed by /R/, else assertion of MII Receive Error (RXER)			Sent for falling MII Transmitter Enable signal (TXEN)		
00111	/R/	Second nibble of ESD, causes de-assertion of CRS if following /T/, else assertion of MII Receive Error (RXER)			Sent for falling MII Transmitter Enable signal (TXEN)		
00100	/H/	Transmit Error Symbol			Sent for rising MII Transmit Error (TXER)		

TABLE 10-1: 4B/5B CODE TABLE (CONTINUED)

Code Group	Sym	Receiver Interpretation	Transmitter Interpretation
00110	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
11001	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
00000	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
00001	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
00010	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
00011	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
00101	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
01000	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
01100	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
10000	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID

10.2.2.3 Scrambler and PISO

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 seed for the scrambler is generated from the PHY address, ensuring that each PHY will have its own scrambler sequence. For more information on PHY addressing, refer to Section 10.1.1, "PHY Addressing".

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

10.2.2.4 NRZI and MLT-3 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 then encoded to MLT-3. MLT-3 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”.

10.2.2.5 100M Transmit Driver

The MLT-3 data is then passed to the analog transmitter, which drives the differential MLT-3 signal on output pins TXPx and TXNx, to the twisted pair media across a 1:1 ratio isolation transformer. The transmitter drives into the 100 Ω impedance of the CAT-5 cable. Cable termination and impedance matching require external components.

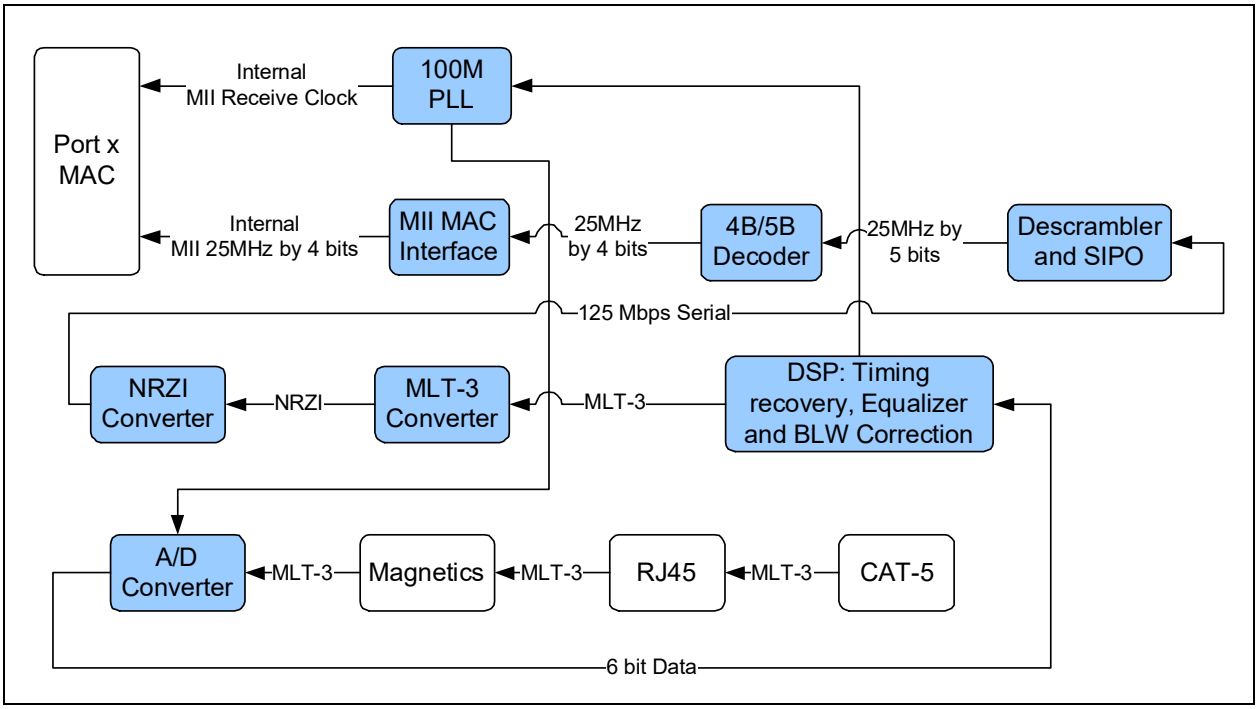
10.2.2.6 100M Phase Lock Loop (PLL)

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

10.2.3 100BASE-TX RECEIVE

The 100BASE-TX receive data path is shown in Figure 10-3. Shaded blocks are those which are internal to the PHY. Each major block is explained in the following sections.

FIGURE 10-3: 100BASE-TX RECEIVE DATA PATH



10.2.3.1 100M Receive Input

The MLT-3 data from the cable is fed into the PHY on inputs RXPx and RXNx 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 quantizer, 6 digital bits are generated 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.

10.2.3.2 Equalizer, BLW Correction and Clock/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 100m.

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.

10.2.3.3 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.

10.2.3.4 Descrambler

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 transceiver by searching for IDLE symbols within a window of 4000 bytes (40 us). 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 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.

10.2.3.5 5B/4B Decoding

The 5-bit code-groups are translated into 4-bit data nibbles according to the 4B/5B table. The translated data is presented on the internal MII RXD[3:0] signal lines. The SSD, /J/K/, is translated to “0101 0101” as the first 2 nibbles of the MAC preamble. Reception of the SSD causes the transceiver to assert the receive data valid signal, indicating that valid data is available on the 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 transceiver to de-assert carrier sense and receive data valid signal.

Note: These symbols are not translated into data.

10.2.3.6 Receive Data Valid Signal

The internal MII’s Receive Data Valid signal (RXDV) indicates that recovered and decoded nibbles are being presented on the RXD[3:0] outputs synchronous to RXCLK. RXDV becomes active after the /J/K/ delimiter has been recognized and RXD is aligned to nibble boundaries. It remains active until either the /T/R/ delimiter is recognized or link test indicates failure or SIGDET becomes false.

RXDV is asserted when the first nibble of translated /J/K/ is ready for transfer over the Media Independent Interface.

10.2.3.7 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 RXER signal is asserted and arbitrary data is driven onto the internal MII’s RXD[3:0] lines. Should an error be detected during the time that the /J/K/ delimiter is being decoded (bad SSD error), RXER is asserted true and the value 1110b is driven onto the RXD[3:0] lines. Note that the internal MII’s data valid signal (RXDV) is not yet asserted when the bad SSD occurs.

10.2.3.8 100M Receive Data Across the Internal MII Interface

For reception, the 4-bit data nibbles are sent to the MII MAC Interface block. These data nibbles are clocked to the controller at a rate of 25 MHz. RXCLK is the output clock for the internal MII bus. It is recovered from the received data to clock the RXD bus. If there is no received signal, it is derived from the system reference clock.

10.2.4 10BASE-T TRANSMIT

10BASE-T is not used for EtherCAT.

10.2.5 AUTO-NEGOTIATION

The purpose of the Auto-Negotiation function is to automatically configure the transceiver 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 and is enabled by setting the Auto-Negotiation Enable (PHY_AN) of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x).

The advertised capabilities of the PHY are stored in the PHY x Auto-Negotiation Advertisement Register (PHY_AN_ADV_x). The transceiver supports “Next Page” capability which is used to negotiate Energy Efficient Ethernet functionality as well as to support software controlled pages. Many of the default advertised capabilities of the PHY are determined via configuration straps as shown in Section 10.2.18.5, "PHY x Auto-Negotiation Advertisement Register (PHY_AN_ADV_x)," on page 233. Refer to Section 3.3, "Configuration Straps," on page 36 for additional details on how to use the device configuration straps.

Once Auto-Negotiation has completed, information about the resolved link and the results of the negotiation process are reflected in the Speed Indication bits in the PHY x Special Control/Status Register (PHY_SPECIAL_CONTROL_STATUS_x), as well as the PHY x Auto-Negotiation Link Partner Base Page Ability Register (PHY_AN_LP_BASE_ABILITY_x). The Auto-Negotiation protocol is a purely physical layer activity and proceeds independently of the MAC controller.

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 any of the following events:

- Power-On Reset (POR)
- Hardware reset (RST#)
- PHY Software reset (via Reset Control Register (RESET_CTL), or bit 15 of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x))
- PHY Power-down reset (Section 10.2.9, "PHY Power-Down Modes," on page 212)
- PHY Link status down (bit 2 of the PHY x Basic Status Register (PHY_BASIC_STATUS_x) is cleared)
- Setting the PHY x Basic Control Register (PHY_BASIC_CONTROL_x), bit 9 high (auto-neg restart)
- EtherCAT System Reset

Note: Refer to Section 6.2, "Resets," on page 46 for information on these and other system resets.

On detection of one of these events, the transceiver begins Auto-Negotiation by transmitting bursts of Fast Link Pulses (FLP). These are bursts of link pulses from the 10M TX Driver. 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 transceiver 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 the PHY x Auto-Negotiation Advertisement Register (PHY_AN_ADV_x).

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

- 100M Full Duplex (Highest priority)
- 100M Half Duplex (Not used for EtherCAT)
- 10M Full Duplex (Not used for EtherCAT)
- 10M Half Duplex (Lowest priority) (Not used for EtherCAT)

If the full capabilities of the transceiver 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 mode.

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 the PHY x Auto-Negotiation Advertisement Register (PHY_AN_ADV_x) bits [8:5] allows software control of the capabilities advertised by the transceiver. Writing the PHY x Auto-Negotiation Advertisement Register (PHY_AN_ADV_x) does not automatically re-start Auto-Negotiation. The Restart Auto-Negotiation (PHY_RST_AN) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) must be set before the new abilities will be advertised. Auto-Negotiation can also be disabled via software by clearing the Auto-Negotiation Enable (PHY_AN) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x).

10.2.5.1 Pause Flow Control

Pause flow control is not used for EtherCAT.

10.2.5.2 Parallel Detection

Parallel detection is not used for EtherCAT.

10.2.5.3 Restarting Auto-Negotiation

Auto-Negotiation can be re-started at any time by setting the Restart Auto-Negotiation (PHY_RST_AN) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x). 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 setting the Restart Auto-Negotiation (PHY_RST_AN) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x), the device will respond by stopping all transmission/receiving operations. Once the internal break_link_time is completed in the Auto-Negotiation state-machine (approximately 1200ms), Auto-Negotiation will re-start. In this case, the link partner will have also dropped the link due to lack of a received signal, so it too will resume Auto-Negotiation.

10.2.5.4 Disabling Auto-Negotiation

Auto-Negotiation can be disabled by clearing the Auto-Negotiation Enable (PHY_AN) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x). The transceiver will then force its speed of operation to reflect the information in the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) (Speed Select LSB (PHY_SPEED_SEL_LSB) and Duplex Mode (PHY_DUPLEX)). These bits are ignored when Auto-Negotiation is enabled.

10.2.5.5 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 carrier sense signal, CRS, responds to both transmit and receive activity. If data is received while the transceiver is transmitting, a collision results.

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

EtherCAT requires the use of full-duplex operation.

10.2.6 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 10-4, the transceiver is capable of configuring the TXPx/TXNx and RXPx/RXNx twisted pair pins for correct transceiver operation.

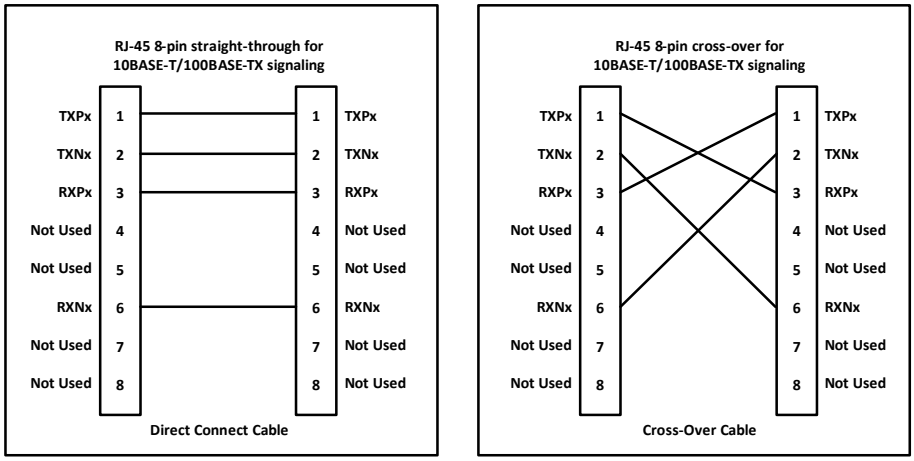
The internal logic of the device detects the TX and RX pins of the connecting device. Since the RX and TX line pairs are interchangeable, special PCB design considerations are needed to accommodate the symmetrical magnetics and termination of an Auto-MDIX design.

The default enable state of Auto-MDIX is controlled by the AMDIX Disable PHY A and AMDIX Disable PHY B bits in the Hardware Configuration Register (HW_CFG).

Software based control of the Auto-MDIX function may be performed using the Auto-MDIX Control (AMDIXCTRL) bit of the PHY x Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND_x). When AMDIXCTRL is set to 1, the Auto-MDIX capability is determined by the Auto-MDIX Enable (AMDIXEN) and Auto-MDIX State (AMDIXSTATE) bits of the PHY x Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND_x).

Note:	When operating in 10BASE-T or 100BASE-TX manual modes, the Auto-MDIX crossover time can be extended via the Extend Manual 10/100 Auto-MDIX Crossover Time bit of the PHY x EDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG_x). Refer to Section 10.2.18.12, on page 242 for additional information. When Energy Detect Power-Down is enabled, the Auto-MDIX crossover time can be extended via the EDPD Extend Crossover bit of the PHY x EDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG_x). Refer to Section 10.2.18.12, on page 242 for additional information

FIGURE 10-4: DIRECT CABLE CONNECTION VS. CROSS-OVER CABLE CONNECTION



10.2.7 PHY MANAGEMENT CONTROL

The PHY Management Control block is responsible for the management functions of the PHY, including register access and interrupt generation. A Serial Management Interface (SMI) is used to support registers as required by the IEEE 802.3 (Clause 22), as well as the vendor specific registers allowed by the specification. The SMI interface consists of the MII Management Data (MDIO) signal and the MII Management Clock (MDC) signal. These signals allow access to all PHY registers. Refer to Section 10.2.18, "PHY Registers," on page 225 for a list of all supported registers and register descriptions. Non-supported registers will be read as FFFFh.

10.2.8 PHY INTERRUPTS

The PHY contains the ability to generate various interrupt events. Reading the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x) shows the source of the interrupt. The PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x) enables or disables each PHY interrupt.

The PHY Management Control block aggregates the enabled interrupts status into an internal signal which is sent to the System Interrupt Controller and is reflected via the PHY A Interrupt Event (PHY_INT_A) and PHY B Interrupt Event (PHY_INT_B) bits of the Interrupt Status Register (INT_STS). For more information on the device interrupts, refer to Section 7.0, "System Interrupts," on page 60.

The PHY interrupt system provides two modes, a Primary interrupt mode and an Alternative interrupt mode. Both modes will assert the internal interrupt signal sent to the System Interrupt Controller when the corresponding mask bit is set. These modes differ only in how they de-assert the internal interrupt signal. These modes are detailed in the following subsections.

Note:	The Primary interrupt mode is the default interrupt mode after a power-up or hard reset. The Alternative interrupt mode requires setup after a power-up or hard reset.

10.2.8.1 Primary Interrupt Mode

The Primary interrupt mode is the default interrupt mode. The Primary interrupt mode is always selected after power-up or hard reset. In this mode, to enable an interrupt, set the corresponding mask bit in the PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x) (see Table 10-2). When the event to assert an interrupt is true, the internal interrupt signal will be asserted. When the corresponding event to de-assert the interrupt is true, the internal interrupt signal will be de-asserted.

TABLE 10-2: INTERRUPT MANAGEMENT TABLE

Mask	Interrupt Source Flag		Interrupt Source		Event to Assert interrupt	Event to De-assert interrupt
30.9	29.9	Link Up	LINKSTAT See Note 1	Link Status	Rising LINK-STAT	Falling LINKSAT or Reading register 29
30.8	29.8	Wake on LAN	WOL_INT See Note 2	Enabled WOL event	Rising WOL_INT	Falling WOL_INT or Reading register 29
30.7	29.7	ENERGYON	17.1	ENERGYON	Rising 17.1 (Note 3)	Falling 17.1 or Reading register 29
30.6	29.6	Auto-Negotia-tion complete	1.5	Auto-Negoti-ate Com-plete	Rising 1.5	Falling 1.5 or Reading register 29
30.5	29.5	Remote Fault Detected	1.4	Remote Fault	Rising 1.4	Falling 1.4, or Reading register 1 or Reading register 29
30.4	29.4	Link Down	1.2	Link Status	Falling 1.2	Reading register 1 or Reading register 29
30.3	29.3	Auto-Negotia-tion LP Acknowl-edge	5.14	Acknowl-edge	Rising 5.14	Falling 5.14 or Reading register 29
30.2	29.2	Parallel Detec-tion Fault	6.4	Parallel Detection Fault	Rising 6.4	Falling 6.4 or Reading register 6, or Reading register 29, or Re-Auto Negotiate or Link down
30.1	29.1	Auto-Negotia-tion Page Received	6.1	Page Received	Rising 6.1	Falling 6.1 or Reading register 6, or Reading register 29, or Re-Auto Negotiate, or Link down.

Note 1: LINKSTAT is the internal link status and is not directly available in any register bit.

Note 2: WOL_INT is defined as bits 7:4 in the PHY x Wakeup Control and Status Register (PHY_WUCSR_x) ANDed with bits 3:0 of the same register, with the resultant 4 bits OR'ed together.

Note 3: If the mask bit is enabled and the internal interrupt signal has been de-asserted while ENERGYON is still high, the internal interrupt signal will assert for 256 ms, approximately one second after ENERGYON goes low when the Cable is unplugged. To prevent an unexpected assertion of the internal interrupt signal, the ENERGYON interrupt mask should always be cleared as part of the ENERGYON interrupt service routine.

Note: The Energy On (ENERGYON) bit in the PHY x Mode Control/Status Register (PHY_MODE_CONTROL_STATUS_x) is defaulted to a '1' at the start of the signal acquisition process, therefore the INT7 bit in the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x) will also read as a '1' at power-up. If no signal is present, then both Energy On (ENERGYON) and INT7 will clear within a few milliseconds.

10.2.8.2 Alternate Interrupt Mode

The Alternate interrupt mode is enabled by setting the ALTINT bit of the PHY x Mode Control/Status Register (PHY_MODE_CONTROL_STATUS_x) to "1". In this mode, to enable an interrupt, set the corresponding bit of the in the PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x) (see Table 10-3). To clear an interrupt, clear the interrupt source and write a '1' to the corresponding Interrupt Source Flag. Writing a '1' to the Interrupt Source Flag will cause the state machine to check the Interrupt Source to determine if the Interrupt Source Flag should clear or stay as a '1'. If the condition to de-assert is true, then the Interrupt Source Flag is cleared and the internal interrupt signal is also deasserted. If the condition to de-assert is false, then the Interrupt Source Flag remains set, and the internal interrupt signal remains asserted.

TABLE 10-3: ALTERNATIVE INTERRUPT MODE MANAGEMENT TABLE

Mask	Interrupt Source Flag		Interrupt Source		Event to Assert interrupt	Condition to De-assert	Bit to Clear interrupt
30.9	29.9	Link Up	LINKSTAT See Note 4	Link Status	Rising LINKSTAT	LINKSTAT low	29.9
30.8	29.8	Wake on LAN	WOL_INT See Note 5	Enabled WOL event	Rising WOL_INT	WOL_INT low	29.8
30.7	29.7	ENERGYON	17.1	ENERGYON	Rising 17.1	17.1 low	29.7
30.6	29.6	Auto-Negotiation complete	1.5	Auto-Negotiate Complete	Rising 1.5	1.5 low	29.6
30.5	29.5	Remote Fault Detected	1.4	Remote Fault	Rising 1.4	1.4 low	29.5
30.4	29.4	Link Down	1.2	Link Status	Falling 1.2	1.2 high	29.4
30.3	29.3	Auto-Negotiation LP Acknowledge	5.14	Acknowledge	Rising 5.14	5.14 low	29.3
30.2	29.2	Parallel Detection Fault	6.4	Parallel Detection Fault	Rising 6.4	6.4 low	29.2
30.1	29.1	Auto-Negotiation Page Received	6.1	Page Received	Rising 6.1	6.1 low	29.1

Note 4: LINKSTAT is the internal link status and is not directly available in any register bit.

Note 5: WOL_INT is defined as bits 7:4 in the PHY x Wakeup Control and Status Register (PHY_WUCSR_x) ANDed with bits 3:0 of the same register, with the resultant 4 bits OR'ed together.

Note: The Energy On (ENERGYON) bit in the PHY x Mode Control/Status Register (PHY_MODE_CONTROL_STATUS_x) is defaulted to a '1' at the start of the signal acquisition process, therefore the INT7 bit in the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x) will also read as a '1' at power-up. If no signal is present, then both Energy On (ENERGYON) and INT7 will clear within a few milliseconds.

10.2.9 PHY POWER-DOWN MODES

There are two PHY power-down modes: General Power-Down Mode and Energy Detect Power-Down Mode. These modes are described in the following subsections.

Note: For more information on the various power management features of the device, refer to Section 6.3, "Power Management," on page 51.

The power-down modes of each PHY are controlled independently.

The PHY power-down modes do not reload or reset the PHY registers.

10.2.9.1 General Power-Down

This power-down mode is controlled by the Power Down (PHY_PWR_DWN) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x). In this mode the entire transceiver, except the PHY management control interface, is powered down. The transceiver will remain in this power-down state as long as the Power Down (PHY_PWR_DWN) bit is set. When the Power Down (PHY_PWR_DWN) bit is cleared, the transceiver powers up and is automatically reset.

10.2.9.2 Energy Detect Power-Down

This power-down mode is enabled by setting the Energy Detect Power-Down (EDPWRDOWN) bit of the PHY x Mode Control/Status Register (PHY_MODE_CONTROL_STATUS_x). In this mode, when no energy is present on the line, the entire transceiver is powered down (except for the PHY management control 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 Energy On (ENERGYON) bit in the PHY x Mode Control/Status Register (PHY_MODE_CONTROL_STATUS_x) signal is low, the transceiver is powered down and nothing is transmitted. When energy is received, via link pulses or packets, the Energy On (ENERGYON) bit goes high, and the transceiver powers up. The transceiver automatically resets itself into the state prior to power-down, and asserts the INT7 bit of the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x). The first and possibly second packet to activate ENERGYON may be lost.

When the Energy Detect Power-Down (EDPWRDOWN) bit of the PHY x Mode Control/Status Register (PHY_MODE_CONTROL_STATUS_x) is low, energy detect power-down is disabled.

When in EDPD mode, the device's NLP characteristics may be modified. The device can be configured to transmit NLPs in EDPD via the EDPD TX NLP Enable bit of the PHY x EDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG_x). When enabled, the TX NLP time interval is configurable via the EDPD TX NLP Interval Timer Select field of the PHY x EDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG_x). When in EDPD mode, the device can also be configured to wake on the reception of one or two NLPs. Setting the EDPD RX Single NLP Wake Enable bit of the PHY x EDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG_x) will enable the device to wake on reception of a single NLP. If the EDPD RX Single NLP Wake Enable bit is cleared, the maximum interval for detecting reception of two NLPs to wake from EDPD is configurable via the EDPD RX NLP Max Interval Detect Select field of the PHY x EDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG_x).

The energy detect power down feature is part of the broader power management features of the device and can be used to trigger the power management event output pin (**PME**) or the general interrupt request pin (**IRQ**). This is accomplished by enabling the energy detect power-down feature of the PHY as described above, and setting the corresponding energy detect enable (bit 14 for PHY A, bit 15 for PHY B) of the Power Management Control Register (PMT_CTRL). Refer to Power Management for additional information.

10.2.10 WAKE ON LAN (WOL)

The PHY supports layer 2 WoL event detection of Perfect DA, Broadcast, Magic Packet, and Wakeup frames.

Each type of supported wake event (Perfect DA, Broadcast, Magic Packet, or Wakeup frames) may be individually enabled via Perfect DA Wakeup Enable (PFDA_EN), Broadcast Wakeup Enable (BCST_EN), Magic Packet Enable (MPEN), and Wakeup Frame Enable (WUEN) bits of the PHY x Wakeup Control and Status Register (PHY_WUCSR_x), respectively. The WoL event is indicated via the INT8 bit of the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x).

The WoL feature is part of the broader power management features of the device and can be used to trigger the power management event output pin (**PME**) or the general interrupt request pin (**IRQ**). This is accomplished by enabling the WoL feature of the PHY as described above, and setting the corresponding WoL enable (bit 14 for PHY A, bit 15 for PHY B) of the Power Management Control Register (PMT_CTRL). Refer to Section 6.3, "Power Management," on page 51 for additional information.

The PHY x Wakeup Control and Status Register (PHY_WUCSR_x) also provides a WoL Configured bit, which may be set by software after all WoL registers are configured. Because all WoL related registers are not affected by software resets, software can poll the WoL Configured bit to ensure all WoL registers are fully configured. This allows the software to skip reprogramming of the WoL registers after reboot due to a WoL event.

The following subsections detail each type of WoL event. For additional information on the main system interrupts, refer to Section 7.0, "System Interrupts," on page 60.

10.2.10.1 Perfect DA (Destination Address) Detection

When enabled, the Perfect DA detection mode allows the detection of a frame with the destination address matching the address stored in the PHY x MAC Receive Address A Register (PHY_RX_ADDRA_x), PHY x MAC Receive Address B Register (PHY_RX_ADDRB_x), and PHY x MAC Receive Address C Register (PHY_RX_ADDRC_x). The frame must also pass the FCS and packet length check.

As an example, the Host system must perform the following steps to enable the device to detect a Perfect DA WoL event:

1. Set the desired MAC address to cause the wake event in the PHY x MAC Receive Address A Register (PHY_RX_ADDRA_x), PHY x MAC Receive Address B Register (PHY_RX_ADDRB_x), and PHY x MAC Receive Address C Register (PHY_RX_ADDRC_x).
2. Set the Perfect DA Wakeup Enable (PFDA_EN) bit of the PHY x Wakeup Control and Status Register (PHY_WUCSR_x) to enable Perfect DA detection.
3. Set bit 8 (WoL event indicator) in the PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x) to enable WoL events.

When a match is triggered, bit 8 of the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x) will be set, and the Perfect DA Frame Received (PFDA_FR) bit of the PHY x Wakeup Control and Status Register (PHY_WUCSR_x) will be set.

10.2.10.2 Broadcast Detection

When enabled, the Broadcast detection mode allows the detection of a frame with the destination address value of FF FF FF FF FF FF. The frame must also pass the FCS and packet length check.

As an example, the Host system must perform the following steps to enable the device to detect a Broadcast WoL event:

1. Set the Broadcast Wakeup Enable (BCST_EN) bit of the PHY x Wakeup Control and Status Register (PHY_WUCSR_x) to enable Broadcast detection.
2. Set bit 8 (WoL event indicator) in the PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x) to enable WoL events.

When a match is triggered, bit 8 of the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x) will be set, and the Broadcast Frame Received (BCAST_FR) bit of the PHY x Wakeup Control and Status Register (PHY_WUCSR_x) will be set.

10.2.10.3 Magic Packet Detection

When enabled, the Magic Packet detection mode allows the detection of a Magic Packet frame. A Magic Packet is a frame addressed to the device - either a unicast to the programmed address, or a broadcast - which contains the pattern 48'h FF_FF_FF_FF_FF_FF after the destination and source address field, followed by 16 repetitions of the desired MAC address (loaded into the PHY x MAC Receive Address A Register (PHY_RX_ADDRA_x), PHY x MAC Receive Address

B Register (PHY_RX_ADDRB_x), and PHY x MAC Receive Address C Register (PHY_RX_ADDRC_x)) without any breaks or interruptions. In case of a break in the 16 address repetitions, the logic scans for the 48'h FF_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 frame must also pass the FCS check and packet length checking.

As an example, if the desired address is 00h 11h 22h 33h 44h 55h, then the logic 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

…FCS

As an example, the Host system must perform the following steps to enable the device to detect a Magic Packet WoL event:

1. Set the desired MAC address to cause the wake event in the PHY x MAC Receive Address A Register (PHY_RX-_ADDRA_x), PHY x MAC Receive Address B Register (PHY_RX_ADDRB_x), and PHY x MAC Receive Address C Register (PHY_RX_ADDRC_x).
2. Set the Magic Packet Enable (MPEN) bit of the PHY x Wakeup Control and Status Register (PHY_WUCSR_x) to enable Magic Packet detection.
3. Set bit 8 (WoL event indicator) in the PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x) to enable WoL events.

When a match is triggered, bit 8 of the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x) will be set, and the Magic Packet Received (MPR) bit of the PHY x Wakeup Control and Status Register (PHY_WUCSR_x) will be set.

10.2.10.4 Wakeup Frame Detection

When enabled, the Wakeup Frame detection mode allows the detection of a pre-programmed Wakeup Frame. Wakeup Frame detection provides a way for system designers to detect a customized pattern within a packet via a programmable wake-up frame filter. The filter has a 128-bit byte mask that indicates which bytes of the frame should be compared by the detection logic. A CRC-16 is calculated over these bytes. The result is then compared with the filter's respective CRC-16 to determine if a match exists. When a wake-up pattern is received, the Remote Wakeup Frame Received (WUFR) bit of the PHY x Wakeup Control and Status Register (PHY_WUCSR_x) is set.

If enabled, the filter can also include a comparison between the frame's destination address and the address specified in the PHY x MAC Receive Address A Register (PHY_RX_ADDRA_x), PHY x MAC Receive Address B Register (PHY_RX_ADDRB_x), and PHY x MAC Receive Address C Register (PHY_RX_ADDRC_x). The specified address can be a unicast or a multicast. If address matching is enabled, only the programmed unicast or multicast address will be considered a match. Non-specific multicast addresses and the broadcast address can be separately enabled. The address matching results are logically OR'd (i.e., specific address match result OR any multicast result OR broadcast result).

Whether or not the filter is enabled and whether the destination address is checked is determined by configuring the PHY x Wakeup Filter Configuration Register A (PHY_WUF_CFGA_x). Before enabling the filter, the application program must provide the detection logic with the sample frame and corresponding byte mask. This information is provided by writing the PHY x Wakeup Filter Configuration Register A (PHY_WUF_CFGA_x), PHY x Wakeup Filter Configuration Register B (PHY_WUF_CFGB_x), and PHY x Wakeup Filter Byte Mask Registers (PHY_WUF_MASK_x). The starting offset within the frame and the expected CRC-16 for the filter is determined by the Filter Pattern Offset and Filter CRC-16 fields, respectively.

If remote wakeup mode is enabled, the remote wakeup function checks each frame against the filter and recognizes the frame as a remote wakeup frame if it passes the filter's address filtering and CRC value match.

The pattern 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 with the pattern offset, should be checked. If bit j in the byte mask is set, the detection logic checks the byte (pattern offset + j) in the frame, otherwise byte (pattern offset + j) is ignored.

At the completion of the CRC-16 checking process, the CRC-16 calculated using the pattern offset and byte mask is compared to the expected CRC-16 value associated with the filter. If a match occurs, a remote wake-up event is signaled. The frame must also pass the FCS check and packet length checking.

Table 10-4 indicates the cases that produce a wake-up event. All other cases do not generate a wake-up event.

TABLE 10-4: WAKEUP GENERATION CASES

Filter Enabled	Frame Type	CRC Matches	Address Match Enabled	Any Mcast Enabled	Bcast Enabled	Frame Address Matches
Yes	Unicast	Yes	No	X	X	X
Yes	Unicast	Yes	Yes	X	X	Yes
Yes	Multicast	Yes	X	Yes	X	X
Yes	Multicast	Yes	Yes	No	X	Yes
Yes	Broadcast	Yes	X	X	Yes	X

As an example, the Host system must perform the following steps to enable the device to detect a Wakeup Frame WoL event:

Declare Pattern:

1. Update the PHY x Wakeup Filter Byte Mask Registers (PHY_WUF_MASK_x) to indicate the valid bytes to match.
2. Calculate the CRC-16 value of valid bytes offline and update the PHY x Wakeup Filter Configuration Register B (PHY_WUF_CFGB_x). CRC-16 is calculated as follows:

   At the start of a frame, CRC-16 is initialized with the value FFFFh. CRC-16 is updated when the pattern offset and mask indicate the received byte is part of the checksum calculation. The following algorithm is used to update the CRC-16 at that time:

   Let:

   ^ denote the exclusive or operator.

   Data [7:0] be the received data byte to be included in the checksum.

   CRC[15:0] contain the calculated CRC-16 checksum.

   F0 … F7 be intermediate results, calculated when a data byte is determined to be part of the CRC-16.

   Calculate:

   F0 = CRC[15] ^ Data[0]

   F1 = CRC[14] ^ F0 ^ Data[1]

   F2 = CRC[13] ^ F1 ^ Data[2]

   F3 = CRC[12] ^ F2 ^ Data[3]

   F4 = CRC[11] ^ F3 ^ Data[4]

   F5 = CRC[10] ^ F4 ^ Data[5]

   F6 = CRC[09] ^ F5 ^ Data[6]

   F7 = CRC[08] ^ F6 ^ Data[7]

   The CRC-32 is updated as follows:

   CRC[15] = CRC[7] ^ F7

   CRC[14] = CRC[6]

   CRC[13] = CRC[5]

   CRC[12] = CRC[4]

   CRC[11] = CRC[3]

CRC[10] = CRC[2]

CRC[9] = CRC[1] ^ F0

CRC[8] = CRC[0] ^ F1

CRC[7] = F0 ^ F2

CRC[6] = F1 ^ F3

CRC[5] = F2 ^ F4

CRC[4] = F3 ^ F5

CRC[3] = F4 ^ F6

CRC[2] = F5 ^ F7

CRC[1] = F6

CRC[0] = F7

3. Determine the offset pattern with offset 0 being the first byte of the destination address. Update the offset in the Filter Pattern Offset field of the PHY x Wakeup Filter Configuration Register A (PHY_WUF_CFGA_x).

Determine Address Matching Conditions:

4. Determine the address matching scheme based on Table 10-4 and update the Filter Broadcast Enable, Filter Any Multicast Enable, and Address Match Enable bits of the PHY x Wakeup Filter Configuration Register A (PHY_WUF_CFGA_x) accordingly.
5. If necessary (see step 4), set the desired MAC address to cause the wake event in the PHY x MAC Receive Address A Register (PHY_RX_ADDRA_x), PHY x MAC Receive Address B Register (PHY_RX_ADDRB_x), and PHY x MAC Receive Address C Register (PHY_RX_ADDRC_x).
6. Set the Filter Enable bit of the PHY x Wakeup Filter Configuration Register A (PHY_WUF_CFGA_x) to enable the filter.

Enable Wakeup Frame Detection:

7. Set the Wakeup Frame Enable (WUEN) bit of the PHY x Wakeup Control and Status Register (PHY_WUCSR_x) to enable Wakeup Frame detection.
8. Set bit 8 (WoL event indicator) in the PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x) to enable WoL events.

When a match is triggered, the Remote Wakeup Frame Received (WUFR) bit of the PHY x Wakeup Control and Status Register (PHY_WUCSR_x) will be set. To provide additional visibility to software, the Filter Triggered bit of the PHY x Wakeup Filter Configuration Register A (PHY_WUF_CFGA_x) will be set.

10.2.11 RESETS

In addition to the chip-level hardware reset (**RST#**), EtherCAT system reset, and Power-On Reset (POR), the PHY supports three block specific resets. These are discussed in the following sections. For detailed information on all device resets and the reset sequence refer to Section 6.2, "Resets," on page 46.

Note:	Only a hardware reset (**RST#**), Power-On Reset (POR) or EtherCAT system reset will automatically reload the configuration strap values into the PHY registers. The Digital Reset (DIGITAL_RST) bit in the Reset Control Register (RESET_CTL) does not reset the PHYs. For all other PHY resets, PHY registers will need to be manually configured via software.

10.2.11.1 PHY Software Reset via RESET_CTL

The PHYs can be reset via the Reset Control Register (RESET_CTL). These bits are self clearing after approximately 102 us. This reset does not reload the configuration strap values into the PHY registers.

10.2.11.2 PHY Software Reset via PHY_BASIC_CTRL_x

The PHY can also be reset by setting the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x). This bit is self clearing and will return to 0 after the reset is complete. This reset does not reload the configuration strap values into the PHY registers.

10.2.11.3 PHY Power-Down Reset

After the PHY has returned from a power-down state, a reset of the PHY is automatically generated. The PHY power-down modes do not reload or reset the PHY registers. Refer to Section 10.2.9, "PHY Power-Down Modes," on page 212 for additional information.

10.2.12 LINK INTEGRITY TEST

The device performs the link integrity test as outlined in the IEEE 802.3u (clause 24-15) Link Monitor state diagram. The link status is multiplexed with the 10 Mbps link status to form the Link Status bit in the PHY x Basic Status Register (PHY_BASIC_STATUS_x) and to drive the LINK LED functions.

The DSP indicates a valid MLT-3 waveform present on the RXPx and RXNx signals as defined by the ANSI X3.263 TP-PMD standard, to the Link Monitor state-machine, using the internal DATA_VALID signal. When DATA_VALID is asserted, the control logic moves into a Link-Ready state and waits for an enable from the auto-negotiation block. When received, the Link-Up state is entered, and the Transmit and Receive logic blocks become active. Should auto-negotiation be disabled, the link integrity logic moves immediately to the Link-Up state when the DATA_VALID is asserted.

To allow the line to stabilize, the link integrity logic will wait a minimum of 330 ms from the time DATA_VALID is asserted until the Link-Ready state is entered. Should the DATA_VALID input be negated at any time, this logic will immediately negate the Link signal and enter the Link-Down state.

10.2.13 CABLE DIAGNOSTICS

The PHYs provide cable diagnostics which allow for open/short and length detection of the Ethernet cable. The cable diagnostics consist of two primary modes of operation:

- Time Domain Reflectometry (TDR) Cable Diagnostics
  TDR cable diagnostics enable the detection of open or shorted cabling on the TX or RX pair, as well as cable length estimation to the open/short fault.
- Matched Cable Diagnostics
  Matched cable diagnostics enable cable length estimation on 100 Mbps-linked cables.

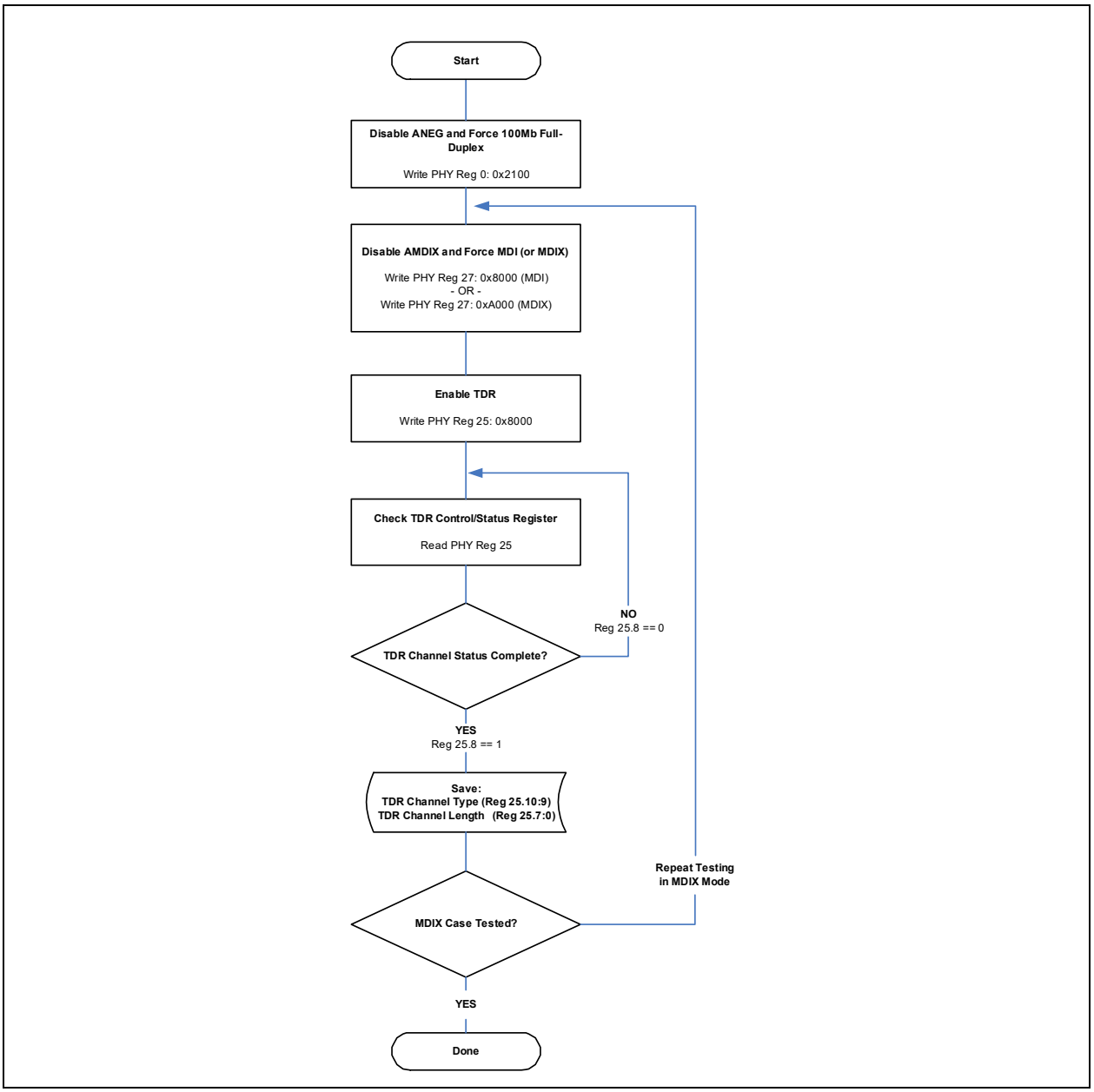
Refer to the following sub-sections for details on proper operation of each cable diagnostics mode.

Note:	Cable diagnostics are not used for 100BASE-FX mode.

10.2.13.1 Time Domain Reflectometry (TDR) Cable Diagnostics

The PHYs provide TDR cable diagnostics which enable the detection of open or shorted cabling on the TX or RX pair, as well as cable length estimation to the open/short fault. To utilize the TDR cable diagnostics, Auto-MDIX and Auto Negotiation must be disabled, and the PHY must be forced to 100 Mbps full-duplex mode. These actions must be performed before setting the TDR Enable bit in the PHY x TDR Control/Status Register (PHY_TDR_CONTROL_STAT_x). With Auto-MDIX disabled, the TDR will test the TX or RX pair selected by register bit 27.13 (Auto-MDIX State (AMDIX-STATE)). Proper cable testing should include a test of each pair. When TDR testing is complete, prior register settings may be restored. Figure 10-5 provides a flow diagram of proper TDR usage.

FIGURE 10-5: TDR USAGE FLOW DIAGRAM



The TDR operates by transmitting pulses on the selected twisted pair within the Ethernet cable (TX in MDI mode, RX in MDIX mode). If the pair being tested is open or shorted, the resulting impedance discontinuity results in a reflected signal that can be detected by the PHY. The PHY measures the time between the transmitted signal and received reflection and indicates the results in the TDR Channel Length field of the PHY x TDR Control/Status Register (PHY_TDR_CONTROL_STAT_x). The TDR Channel Length field indicates the “electrical” length of the cable, and can be multiplied by the appropriate propagation constant in Table 10-5 to determine the approximate physical distance to the fault.

Note:	The TDR function is typically used when the link is inoperable. However, an active link will drop when operating the TDR.

Since the TDR relies on the reflected signal of an improperly terminated cable, there are several factors that can affect the accuracy of the physical length estimate. These include:

1. ***Cable Type (CAT 5, CAT5e, CAT6):*** The electrical length of each cable type is slightly different due to the twists-per-meter of the internal signal pairs and differences in signal propagation speeds. If the cable type is known, the length estimate can be calculated more accurately by using the propagation constant appropriate for the cable type (see Table 10-5). In many real-world applications the cable type is unknown, or may be a mix of different cable types and lengths. In this case, use the propagation constant for the "unknown" cable type.
2. ***TX and RX Pair:*** For each cable type, the EIA standards specify different twist rates (twists-per-meter) for each signal pair within the Ethernet cable. This results in different measurements for the RX and TX pair.
3. ***Actual Cable Length:*** The difference between the estimated cable length and actual cable length grows as the physical cable length increases, with the most accurate results at less than approximately 100 m.
4. ***Open/Short Case:*** The Open and Shorted cases will return different TDR Channel Length values (electrical lengths) for the same physical distance to the fault. Compensation for this is achieved by using different propagation constants to calculate the physical length of the cable.

   For the Open case, the estimated distance to the fault can be calculated as follows:

   Distance to Open fault in meters ≅ TDR Channel Length * P_{OPEN}
   Where: P_{OPEN} is the propagation constant selected from Table 10-5

   For the Shorted case, the estimated distance to the fault can be calculated as follows:

   Distance to Open fault in meters ≅ TDR Channel Length * P_{SHORT}
   Where: P_{SHORT} is the propagation constant selected from Table 10-5

TABLE 10-5: TDR PROPAGATION CONSTANTS

TDR Propagation Constant	Cable Type			
	Unknown	CAT 6	CAT 5E	CAT 5
P_{OPEN}	0.769	0.745	0.76	0.85
P_{SHORT}	0.793	0.759	0.788	0.873

The typical cable length measurement margin of error for Open and Shorted cases is dependent on the selected cable type and the distance of the open/short from the device. Table 10-6 and Table 10-7 detail the typical measurement error for Open and Shorted cases, respectively.

TABLE 10-6: TYPICAL MEASUREMENT ERROR FOR OPEN CABLE (+/- METERS)

Physical Distance to Fault	Selected Propagation Constant			
	P_{OPEN} = Unknown	P_{OPEN} = CAT 6	P_{OPEN} = CAT 5E	P_{OPEN} = CAT 5
CAT 6 Cable, 0-100 m	9	6		
CAT 5E Cable, 0-100 m	5		5	
CAT 5 Cable, 0-100 m	13			3
CAT 6 Cable, 101-160 m	14	6		
CAT 5E Cable, 101-160 m	8		6	
CAT 5 Cable, 101-160 m	20			6

TABLE 10-7: TYPICAL MEASUREMENT ERROR FOR SHORTED CABLE (+/- METERS)

PHYSICAL DISTANCE TO FAULT	SELECTED PROPAGATION CONSTANT			
	P_{SHORT} = Unknown	P_{SHORT} = CAT 6	P_{SHORT} = CAT 5E	P_{SHORT} = CAT 5
CAT 6 Cable, 0-100 m	8	5		
CAT 5E Cable, 0-100 m	5		5	
CAT 5 Cable, 0-100 m	11			2
CAT 6 Cable, 101-160 m	14	6		
CAT 5E Cable, 101-160 m	7		6	
CAT 5 Cable, 101-160 m	11			3

10.2.13.2 Matched Cable Diagnostics

Matched cable diagnostics enable cable length estimation on 100 Mbps-linked cables of up to 120 meters. If there is an active 100 Mb link, the approximate distance to the link partner can be estimated using the PHY x Cable Length Register (PHY_CABLE_LEN_x). If the cable is properly terminated, but there is no active 100 Mb link (the link partner is disabled, nonfunctional, the link is at 10 Mb, etc.), the cable length cannot be estimated and the PHY x Cable Length Register (PHY_CABLE_LEN_x) should be ignored. The estimated distance to the link partner can be determined via the Cable Length (CBLN) field of the PHY x Cable Length Register (PHY_CABLE_LEN_x) using the lookup table provided in Table 10-8. The typical cable length measurement margin of error for a matched cable case is +/- 20 m. The matched cable length margin of error is consistent for all cable types from 0 to 120 m.

TABLE 10-8: MATCH CASE ESTIMATED CABLE LENGTH (CBLN) LOOKUP

CBLN Field Value	Estimated Cable Length
0 - 3	0
4	6
5	17
6	27
7	38
8	49
9	59
10	70
11	81
12	91
13	102
14	113
15	123

Note:	For a properly terminated cable (Match case), there is no reflected signal. In this case, the TDR Channel Length field is invalid and should be ignored.

10.2.14 SIGNAL QUALITY INDEX

10.2.14.1 Background

MLT-3 modulation is used for data transmission in 100BASE-TX, with logical signal values of {-1, 0, +1}. These logic levels correspond to maximum positive and negative DSP slicer reference levels. Positive and negative compare thresholds are set between these maximums and zero, such that the analog received data samples may be quantized to -1, 0 and +1.

Ideally, each receive data sample would be the maximum distance from the compare thresholds, with error values of 0. But because of noise and imperfection in real applications, the sampled data may be off from its ideal. The closer to the compare threshold, the worse the signal quality.

The slicer error is a measurement of how far the processed data off from its ideal location. The largest instantaneous slicer error 100BASE-TX is +/-32.

A higher absolute slicer error means a degraded signal receiving condition.

Note:	The following register sequence must be issued to precondition PHYs A and B to enable the slicer error into the Signal Quality Index logic.
	
	// writing bit[10] of reg-20 from 1 -> 0 -> 1 will enable the testmode in PHY
	Register PHY_TSTCNTL_A/B Data = 0x0400
	Register PHY_TSTCNTL_A/B Data = 0x0000
	Register PHY_TSTCNTL_A/B Data = 0x0400
	
	// Maps err_nm1_real[5:0] to co_testbus_out[5:0] and rxclk125 to co_clk_out
	Register PHY_TSTWRITE_A/B Data = 0x7200
	
	// Enables the test bus outputs in PHY
	Register PHY_TSTCNTL_A/B Data = 0x4401

10.2.14.2 OPEN Alliance TC1 DCQ Mean Square Error

Note:	All register referenced in this section are in MMD Device Address 30. Refer to Section 10.2.18, "PHY Registers" for additional information.

This section defines the implementation of section 6.1.1 of the TC1 specification. The PHY can provide detailed information of the dynamic signal quality by means of a MSE value. This mode is enabled by setting the sqi_enable bit, in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x).

With this method, the slicer error is converted into a squared value and then filtered by a programmable low pass filter. This is similar to taking the average of absolute slicer error over a long moving time window.

For each data sample, the difference between the absolute slicer error and the current filtered value is added back into the current filtered value.

The sqi_squ_mode_en bit in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x) must be set to choose square mode.

The sqi_kp field in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x) sets the weighting of the add back as a divide by $2^{\wedge sqi\_kp}$, effectively setting the filter bandwidth. As the sqi_kp value is increased, the weighing is decreased, and the mean slicer error value takes a longer time to settle to a stable value. Also as the sqi_kp value is increased, there will be less variation in the mean slicer error value reported.

The scale611 field in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x) is used to set a divide by factor (divide by 2^scale611) such that the MSE value is linearly scaled to the range of 0 to 511. If the divide by factor is too small, the MSE value is capped at a maximum of 511.

The filtered error value is saved every 65,536 symbols (524,288 ns).

In order to capture the MSE Value, the DCQ Read Capture bit in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x) needs to be written as a high. The DCQ Read Capture bit will immediately self-clear and the result will be available in the PHY x DCQ Mean Square Error Register (PHY_DCQ_MSE_x).

In addition to the current MSE Value the worst case MSE value since the last read of PHY x DCQ Mean Square Error Register (PHY_DCQ_MSE_x) is stored in PHY x DCQ Mean Square Error Worst Case Register (PHY_DCQ_MSE_WC_x).

10.2.14.3 OPEN Alliance TC1 DCQ Signal Quality Index

Note:	All register referenced in this section are in MMD Device Address 30. Refer to Section 10.2.18, "PHY Registers" for additional information.

This section defines the implementation of section 6.1.2 of the TC1 specification.

This mode builds upon the above DCQ Mean Square Error method by mapping the MSE value onto a simple quality index.

This mode is enabled by setting the sqi_enable bit, in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x).

Note:	As above, the sqi_squ_mode_en bit in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x) must be set to choose square mode. As above, the scale611 field in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x) is used to set the divide by factor (divide by 2^scale611) such that the MSE value is linearly scaled to the range of 0 to 511.

The MSE value is compared to the thresholds set in the PHY x DCQ SQI Table Registers (PHY_DCQ_SQI_TBL_x) to provide a Signal Quality Index value between 0 (worst value) and 7 (best value) as follows:

TABLE 10-9: MSE TO SIGNAL QUALITY INDEX MAPPING

MSE Value		Signal Quality Index Value
Greater Than	**Less Than or Equal To**	
	SQI_TBL7.SQI_VALUE	7
SQI_TBL7.SQI_VALUE	SQI_TBL6.SQI_VALUE	6
SQI_TBL6.SQI_VALUE	SQI_TBL5.SQI_VALUE	5
SQI_TBL5.SQI_VALUE	SQI_TBL4.SQI_VALUE	4
SQI_TBL4.SQI_VALUE	SQI_TBL3.SQI_VALUE	3
SQI_TBL3.SQI_VALUE	SQI_TBL2.SQI_VALUE	2
SQI_TBL2.SQI_VALUE	SQI_TBL1.SQI_VALUE	1
SQI_TBL1.SQI_VALUE		0

In order to capture the SQI value, the DCQ Read Capture bit in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x) needs to be written as a high. The DCQ Read Capture bit will immediately self-clear and the result will be available in the PHY x DCQ SQI Register (PHY_DCQ_SQI_x).

In addition to the current SQI the worst case (lowest) Signal Quality Index since the last read is available in the SQI Worst Case field.

The correlation between the Signal Quality Index values stored in the PHY x DCQ SQI Register (PHY_DCQ_SQI_x) and an according signal to noise ratio (SNR) based on AWG noise (bandwidth of 80MHz @ 100Mbps) is shown in Table 10-10. The bit error rates to be expected in the case of white noise as interference signal is shown in the table as well for information purposes.

A link loss only occurs if the Signal Quality Index value is 0.

TABLE 10-10: SIGNAL QUALITY INDEX VALUE CORRELATION

SQI Value	SNR Value @ MDI - AWG Noise	Recommended BER for AWG Noise Model
0	< 18 dB	BER>10^-10
1	18 dB <= SNR < 19 dB	
2	19 dB <= SNR < 20 dB	
3	20 dB <= SNR < 21 dB	BER<10^-10
4	21 dB <= SNR < 22 dB	
5	22 dB <= SNR < 23 dB	
6	23 dB <= SNR < 24 dB	
7	SNR <= 24 dB	

10.2.14.4 OPEN Alliance TC1 DCQ Peak MSE value (pMSE)

Note: All register referenced in this section are in MMD Device Address 30. Refer to Section 10.2.18, "PHY Registers" for additional information.

This section defines the implementation of section 6.1.3 of the TC1 specification.

The peakMSE value is intended to identify transient disturbances which are typically in the microsecond range.

This mode is enabled by setting the sqi_enable bit, in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x).

With this method, the slicer error is converted into a squared value and then filtered by a programmable low pass filter. This is similar to taking the average of absolute slicer error over a moving time window.

For each data sample, the difference between the absolute slicer error and the current filtered value is added back into the current filtered value.

The sqi_squ_mode_en *bit in the* PHY x DCQ Configuration Register (PHY_DCQ_CFG_x) *must be set to choose square mode.*

The sqi_kp3 field in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x) sets the weighting of the add back as a divide by $2^{\wedge(sqi\_kp3)}$, effectively setting the filter bandwidth. As the sqi_kp3 value is increased, the weighing is decreased, and the mean slicer error value takes a longer time to settle to a stable value.

Every 65,536 symbols (524,288 ns), the highest filtered value over those previous 65,536 symbols is saved.

The scale613 field in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x) is used to set a divide by factor (divide by $2^{\wedge scale613+3)}$ such that the peak MSE value is linearly scaled to the range of 0 to 63. If the divided by factor is too small, the peak MSE value is capped at a maximum of 63.

In order to capture the Peak MSE Value, the DCQ Read Capture bit in the PHY x DCQ Configuration Register (PHY_DCQ_CFG_x) needs to be written as a high. The DCQ Read Capture bit will immediately self-clear and the result will be available in the PHY x DCQ Peak MSE Register (PHY_DCQ_PEAK_MSE_x).

In addition to the current Peak MSE Value the worst case Peak MSE value since the last read of PHY x DCQ Peak MSE Register (PHY_DCQ_PEAK_MSE_x) is stored in the same register.

This method is similar to the above DCQ Mean Square Error method except that this method saves the peak value every 1.0ms and uses different “kp” and “scale” parameters.

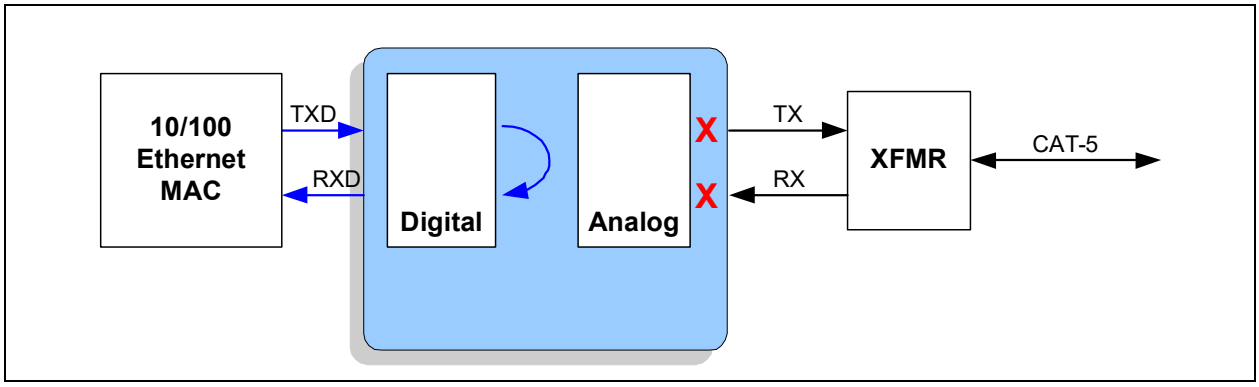
10.2.15 LOOPBACK OPERATION

The PHYs may be configured for near-end loopback and connector loopback. These loopback modes are detailed in the following subsections.

10.2.15.1 Near-end Loopback

Near-end loopback mode sends the digital transmit data back out the receive data signals for testing purposes, as indicated by the blue arrows in Figure 10-6. The near-end loopback mode is enabled by setting the Loopback (PHY_LOOPBACK) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) to “1”. A large percentage of the digital circuitry is operational in near-end loopback mode because data is routed through the PCS and PMA layers into the PMD sublayer before it is looped back. The COL signal will be inactive in this mode, unless Collision Test Mode (PHY_COL_TEST) is enabled in the PHY x Basic Control Register (PHY_BASIC_CONTROL_x). The transmitters are powered down regardless of the state of the internal MII TXEN signal.

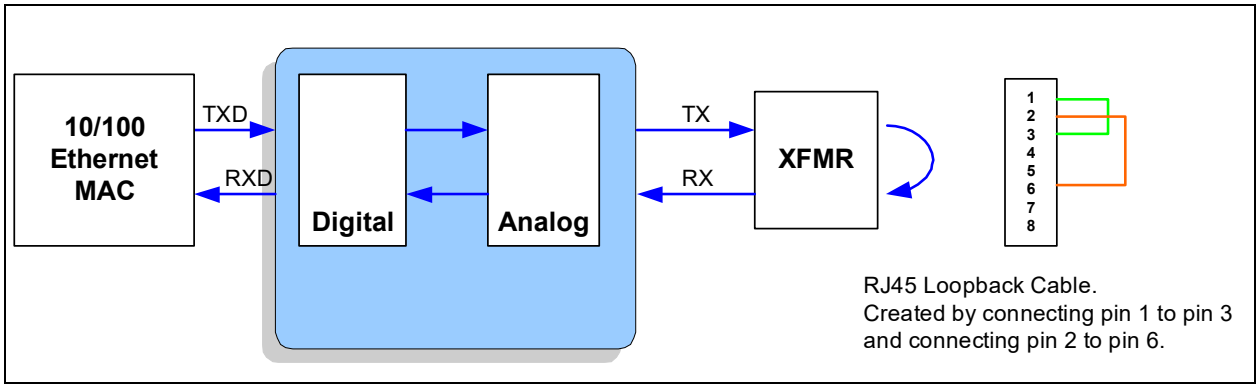
FIGURE 10-6: NEAR-END LOOPBACK BLOCK DIAGRAM



10.2.15.2 Connector Loopback

The device maintains reliable transmission over very short cables and can be tested in a connector loopback as shown in Figure 10-7. An RJ45 loopback cable can be used to route the transmit signals from the output of the transformer back to the receiver inputs. The loopback works at both 10 and 100 Mbps.

FIGURE 10-7: CONNECTOR LOOPBACK BLOCK DIAGRAM



10.2.16 REQUIRED ETHERNET MAGNETICS (100BASE-TX)

The magnetics selected for use with the device should be an Auto-MDIX style magnetic, which is widely available from several vendors. Please review the SMSC/Microchip Application note 8.13 “Suggested Magnetics” for the latest qualified and suggested magnetics. A list of vendors and part numbers are provided within the application note.

10.2.17 ETHERCAT P

EtherCAT P is an enhancement to EtherCAT combining power and the EtherCAT data transmission on the same four wires.

The four wires utilized for EtherCAT and the four wires used for powering U_S (for logic) and U_P (for the output peripherals) are shown in x.

TABLE 10-11: ETHERCAT P SIGNAL AND POWER WIRES

(Typical) Wire Color	Yellow	Orange	White	Blue
EtherCAT	TX+	TX-	RX+	RX-
Power	GND_S	U_S	GND_P	U_P

EtherCAT P is implemented on the PCB design by using an LC low pass filter to pass the DC power to the power supply circuits and by using a series capacitor high pass filter to pass the communications signal to the Ethernet transformer.

Due to the MLT3-Coding, a specific bit pattern leads to a DC offset in the physical transmission. When a stream of data pattern known as “killer packets” (long periods of +1 or -1) is sent through the EtherCAT cable and the Ethernet transformer, it causes the MLT-3 signals to accumulate a significant DC offset since the transformer is high pass in nature. This causes the signals to wander away from the baseline (referred to as baseline wander or BLW). On the receive side, this signal results in bit error.

10.2.18 PHY REGISTERS

PHYs A and B are comparable in functionality and have an identical set of non-memory mapped registers. The PHY registers are indirectly accessed through the EtherCAT module.

Because PHY A and B registers are functionally identical, their register descriptions have been consolidated. A lowercase “x” has been appended to the end of each PHY register name in this section, where “x” hold be replaced with “A” or “B” for the PHY A or PHY B registers respectively. In some instances, a “1” or a “2” may be appropriate instead.

A list of the MII serial accessible Control and Status registers and their corresponding register index numbers is included in Table 10-12. Each individual PHY is assigned a unique PHY address as detailed in Section 10.1.1, "PHY Addressing," on page 201.

In addition to the MII serial accessible Control and Status registers, a set of indirectly accessible registers provides support for the *IEEE 802.3 Section 45.2 MDIO Manageable Device (MMD) Registers*. A list of these registers and their corresponding register index numbers is included in Table 10-15.

Control and Status Registers

Table 10-12 provides a list of supported registers. Register details, including bit definitions, are provided in the following subsections.

Unless otherwise specified, reserved fields must be written with zeros if the register is written.

TABLE 10-12: PHY A AND B MII SERIALLY ACCESSIBLE CONTROL AND STATUS REGISTERS

Index	Register Name (SYMBOL)	Group
0	PHY x Basic Control Register (PHY_BASIC_CONTROL_x)	Basic

TABLE 10-12: PHY A AND B MII SERIALLY ACCESSIBLE CONTROL AND STATUS REGISTERS

Index	Register Name (SYMBOL)	Group
1	PHY x Basic Status Register (PHY_BASIC_STATUS_x)	Basic
2	PHY x Identification MSB Register (PHY_ID_MSB_x)	Extended
3	PHY x Identification LSB Register (PHY_ID_LSB_x)	Extended
4	PHY x Auto-Negotiation Advertisement Register (PHY_AN_ADV_x)	Extended
5	PHY x Auto-Negotiation Link Partner Base Page Ability Register (PHY_AN_LP_BASE_ABILITY_x)	Extended
6	PHY x Auto-Negotiation Expansion Register (PHY_AN_EXP_x)	Extended
7	PHY x Auto Negotiation Next Page TX Register (PHY_AN_NP_TX_x)	Extended
8	PHY x Auto Negotiation Next Page RX Register (PHY_AN_NP_RX_x)	Extended
13	PHY x MMD Access Control Register (PHY_MMD_ACCESS)	Extended
14	PHY x MMD Access Address/Data Register (PHY_MMD_ADDR_DATA)	Extended
16	PHY x EDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG_x)	Vendor-specific
17	PHY x Mode Control/Status Register (PHY_MODE_CONTROL_STATUS_x)	Vendor-specific
18	PHY x Special Modes Register (PHY_SPECIAL_MODES_x)	Vendor-specific
24	PHY x TDR Patterns/Delay Control Register (PHY_TDR_PAT_DELAY_x)	Vendor-specific
25	PHY x TDR Control/Status Register (PHY_TDR_CONTROL_STAT_x)	Vendor-specific
26	PHY x Symbol Error Counter Register	Vendor-specific
27	PHY x Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND_x)	Vendor-specific
28	PHY x Cable Length Register (PHY_CABLE_LEN_x)	Vendor-specific
29	PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x)	Vendor-specific
30	PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x)	Vendor-specific
31	PHY x Special Control/Status Register (PHY_SPECIAL_CONTROL_STATUS_x)	Vendor-specific

10.2.18.1 PHY x Basic Control Register (PHY_BASIC_CONTROL_x)

Index (decimal): 0 Size: 16 bits

This read/write register is used to configure the PHY.

Bits	Description	Type	Default
15	**Soft Reset (PHY_SRST)** When set, this bit resets all the PHY registers to their default state, except those marked as NASR type. This bit is self clearing. **0: Normal operation** **1: Reset**	R/W SC	0b
14	**Loopback (PHY_LOOPBACK)** This bit enables/disables the loopback mode. When enabled, transmissions are not sent to network. Instead, they are looped back into the PHY. **0: Loopback mode disabled (normal operation)** **1: Loopback mode enabled**	R/W	0b
13	**Speed Select LSB (PHY_SPEED_SEL_LSB)** This bit is used to set the speed of the PHY when the Auto-Negotiation Enable (PHY_AN) bit is disabled. **0: 10 Mbps** **1: 100 Mbps**	R/W	1b
12	**Auto-Negotiation Enable (PHY_AN)** This bit enables/disables Auto-Negotiation. When enabled, the Speed Select LSB (PHY_SPEED_SEL_LSB) and Duplex Mode (PHY_DUPLEX) bits are overridden. **0: Auto-Negotiation disabled** **1: Auto-Negotiation enabled**	R/W	Note 6
11	**Power Down (PHY_PWR_DWN)** This bit controls the power down mode of the PHY. **0: Normal operation** **1: General power down mode**	R/W	0b
10	**RESERVED**	RO	-
9	**Restart Auto-Negotiation (PHY_RST_AN)** When set, this bit restarts the Auto-Negotiation process. **0: Normal operation** **1: Auto-Negotiation restarted**	R/W SC	0b
8	**Duplex Mode (PHY_DUPLEX)** This bit is used to set the duplex when the Auto-Negotiation Enable (PHY_AN) bit is disabled. **0: Half Duplex** **1: Full Duplex**	R/W	1b

Bits	Description	Type	Default
7	**Collision Test Mode (PHY_COL_TEST)** This bit enables/disables the collision test mode of the PHY. When set, the collision signal is active during transmission. It is recommended that this feature be used only in loopback mode. **0: Collision test mode disabled** **1: Collision test mode enabled**	R/W	0b
6:0	**RESERVED**	RO	-

Note 6: The default is the value of this field is determined by the ANEG Disable PHY A/ANEG Disable PHY B bits in the Hardware Configuration Register (HW_CFG), which are in turn controlled by configuration straps. EtherCAT normally uses auto-negotiate, however special use cases may require it to be disabled.

10.2.18.2 PHY x Basic Status Register (PHY_BASIC_STATUS_x)

Index (decimal): 1 Size: 16 bits

This register is used to monitor the status of the PHY.

Bits	Description	Type	Default
15	**100BASE-T4** This bit displays the status of 100BASE-T4 compatibility. **0: PHY not able to perform 100BASE-T4** **1: PHY able to perform 100BASE-T4**	RO	0b
14	**100BASE-X Full Duplex** This bit displays the status of 100BASE-X full duplex compatibility. **0: PHY not able to perform 100BASE-X full duplex** **1: PHY able to perform 100BASE-X full duplex**	RO	1b
13	**100BASE-X Half Duplex** This bit displays the status of 100BASE-X half duplex compatibility. **0: PHY not able to perform 100BASE-X half duplex** **1: PHY able to perform 100BASE-X half duplex**	RO	1b
12	**10BASE-T Full Duplex** This bit displays the status of 10BASE-T full duplex compatibility. **0: PHY not able to perform 10BASE-T full duplex** **1: PHY able to perform 10BASE-T full duplex**	RO	1b
11	**10BASE-T Half Duplex** This bit displays the status of 10BASE-T half duplex compatibility. **0: PHY not able to perform 10BASE-T half duplex** **1: PHY able to perform 10BASE-T half duplex**	RO	1b
10	**100BASE-T2 Full Duplex** This bit displays the status of 100BASE-T2 full duplex compatibility. **0: PHY not able to perform 100BASE-T2 full duplex** **1: PHY able to perform 100BASE-T2 full duplex**	RO	0b
9	**100BASE-T2 Half Duplex** This bit displays the status of 100BASE-T2 half duplex compatibility. **0: PHY not able to perform 100BASE-T2 half duplex** **1: PHY able to perform 100BASE-T2 half duplex**	RO	0b
8	**Extended Status** This bit displays whether extended status information is in register 15 (per IEEE 802.3 clause 22.2.4). **0: No extended status information in Register 15** **1: Extended status information in Register 15**	RO	0b

Bits	Description	Type	Default
7	**Unidirectional Ability** This bit indicates whether the PHY is able to transmit regardless of whether the PHY has determined that a valid link has been established. **0: Can only transmit when a valid link has been established** **1: Can transmit regardless**	RO	0b
6	**MF Preamble Suppression** This bit indicates whether the PHY accepts management frames with the preamble suppressed. **0: Management frames with preamble suppressed not accepted** **1: Management frames with preamble suppressed accepted**	RO	0b
5	**Auto-Negotiation Complete** This bit indicates the status of the Auto-Negotiation process. **0: Auto-Negotiation process not completed** **1: Auto-Negotiation process completed**	RO	0b
4	**Remote Fault** This bit indicates if a remote fault condition has been detected. **0: No remote fault condition detected** **1: Remote fault condition detected**	RO/LH	0b
3	**Auto-Negotiation Ability** This bit indicates the PHY's Auto-Negotiation ability. **0: PHY is unable to perform Auto-Negotiation** **1: PHY is able to perform Auto-Negotiation**	RO	1b
2	**Link Status** This bit indicates the status of the link. **0: Link is down** **1: Link is up**	RO/LL	0b
1	**Jabber Detect** This bit indicates the status of the jabber condition. **0: No jabber condition detected** **1: Jabber condition detected**	RO/LH	0b
0	**Extended Capability** This bit indicates whether extended register capability is supported. **0: Basic register set capabilities only** **1: Extended register set capabilities**	RO	1b

10.2.18.3 PHY x Identification MSB Register (PHY_ID_MSB_x)

Index (decimal): 2 Size: 16 bits

This read/write register contains the MSB of the Organizationally Unique Identifier (OUI) for the PHY. The LSB of the PHY OUI is contained in the PHY x Identification LSB Register (PHY_ID_LSB_x).

Bits	Description	Type	Default
15:0	**PHY ID** This field is assigned to the 3rd through 18th bits of the OUI, respectively (OUI = 00800Fh).	R/W	0007h

10.2.18.4 PHY x Identification LSB Register (PHY_ID_LSB_x)

Index (decimal): 3 Size: 16 bits

This read/write register contains the LSB of the Organizationally Unique Identifier (OUI) for the PHY. The MSB of the PHY OUI is contained in the PHY x Identification MSB Register (PHY_ID_MSB_x).

Bits	Description	Type	Default
15:10	**PHY ID** This field is assigned to the 19th through 24th bits of the PHY OUI, respectively. (OUI = 00800Fh).	R/W	C170h
9:4	**Model Number** This field contains the 6-bit manufacturer's model number of the PHY.	R/W	
3:0	**Revision Number** This field contain the 4-bit manufacturer's revision number of the PHY.	R/W	

Note: The default value of the Revision Number field may vary dependent on the silicon revision number.

10.2.18.5 PHY x Auto-Negotiation Advertisement Register (PHY_AN_ADV_x)

Index (decimal): 4 Size: 16 bits

This read/write register contains the advertised ability of the PHY and is used in the Auto-Negotiation process with the link partner.

Bits	Description	Type	Default
15	**Next Page** **0 = No next page ability** **1 = Next page capable**	R/W	0b
14	**RESERVED**	RO	-
13	**Remote Fault** This bit determines if remote fault indication will be advertised to the link partner. **0: Remote fault indication not advertised** **1: Remote fault indication advertised**	R/W	0b
12	**Extended Next Page** **Note:** This bit should be written as 0.	R/W	0b
11	**Asymmetric Pause** This bit determines the advertised asymmetric pause capability. **0: No Asymmetric PAUSE toward link partner advertised** **1: Asymmetric PAUSE toward link partner advertised**	R/W	0b Note 7
10	**Symmetric Pause** This bit determines the advertised symmetric pause capability. **0: No Symmetric PAUSE toward link partner advertised** **1: Symmetric PAUSE toward link partner advertised**	R/W	0b Note 7
9	**RESERVED**	RO	-
8	**100BASE-X Full Duplex** This bit determines the advertised 100BASE-X full duplex capability. **0: 100BASE-X full duplex ability not advertised** **1: 100BASE-X full duplex ability advertised**	R/W	1b Note 8
7	**100BASE-X Half Duplex** This bit determines the advertised 100BASE-X half duplex capability. **0: 100BASE-X half duplex ability not advertised** **1: 100BASE-X half duplex ability advertised**	R/W	0b Note 8
6	**10BASE-T Full Duplex** This bit determines the advertised 10BASE-T full duplex capability. **0: 10BASE-T full duplex ability not advertised** **1: 10BASE-T full duplex ability advertised**	R/W	0b Note 8

Bits	Description	Type	Default
5	**10BASE-T Half Duplex** This bit determines the advertised 10BASE-T half duplex capability. **0: 10BASE-T half duplex ability not advertised** **1: 10BASE-T half duplex ability advertised**	R/W	0b Note 8
4:0	**Selector Field** This field identifies the type of message being sent by Auto-Negotiation. **00001: IEEE 802.3**	R/W	00001b

Note 7: Pause flow control is not used for EtherCAT.

Note 8: EtherCAT only uses 100Mbps, full-duplex.

10.2.18.6 PHY x Auto-Negotiation Link Partner Base Page Ability Register (PHY_AN_LP_BASE_ABILITY_x)

Index (decimal): 5 Size: 16 bits

This read-only register contains the advertised ability of the link partner's PHY and is used in the Auto-Negotiation process between the link partner and the PHY.

Bits	Description	Type	Default
15	**Next Page** This bit indicates the link partner PHY page capability. **0: Link partner PHY does not advertise next page capability** **1: Link partner PHY advertises next page capability**	RO	0b
14	**Acknowledge** This bit indicates whether the link code word has been received from the partner. **0: Link code word not yet received from partner** **1: Link code word received from partner**	RO	0b
13	**Remote Fault** This bit indicates whether a remote fault has been detected. **0: No remote fault** **1: Remote fault detected**	RO	0b
12	**Extended Next Page** **0: Link partner PHY does not advertise extended next page capability** **1: Link partner PHY advertises extended next page capability**	RO	0b
11	**Asymmetric Pause** This bit indicates the link partner PHY asymmetric pause capability. **0: No Asymmetric PAUSE toward link partner** **1: Asymmetric PAUSE toward link partner**	RO	0b
10	**Pause** This bit indicates the link partner PHY symmetric pause capability. **0: No Symmetric PAUSE toward link partner** **1: Symmetric PAUSE toward link partner**	RO	0b
9	**100BASE-T4** This bit indicates the link partner PHY 100BASE-T4 capability. **0: 100BASE-T4 ability not supported** **1: 100BASE-T4 ability supported**	RO	0b
8	**100BASE-X Full Duplex** This bit indicates the link partner PHY 100BASE-X full duplex capability. **0: 100BASE-X full duplex ability not supported** **1: 100BASE-X full duplex ability supported**	RO	0b

Bits	Description	Type	Default
7	**100BASE-X Half Duplex** This bit indicates the link partner PHY 100BASE-X half duplex capability. **0: 100BASE-X half duplex ability not supported** **1: 100BASE-X half duplex ability supported**	RO	0b
6	**10BASE-T Full Duplex** This bit indicates the link partner PHY 10BASE-T full duplex capability. **0: 10BASE-T full duplex ability not supported** **1: 10BASE-T full duplex ability supported**	RO	0b
5	**10BASE-T Half Duplex** This bit indicates the link partner PHY 10BASE-T half duplex capability. **0: 10BASE-T half duplex ability not supported** **1: 10BASE-T half duplex ability supported**	RO	0b
4:0	**Selector Field** This field identifies the type of message being sent by Auto-Negotiation. **00001: IEEE 802.3**	RO	00001b

10.2.18.7 PHY x Auto-Negotiation Expansion Register (PHY_AN_EXP_x)

Index (decimal): 6 Size: 16 bits

This read/write register is used in the Auto-Negotiation process between the link partner and the PHY.

Bits	Description	Type	Default
15:7	**RESERVED**	RO	-
6	**Receive Next Page Location Able** 0 = Received next page storage location is not specified by bit 6.5 1 = Received next page storage location is specified by bit 6.5	RO	1b
5	**Received Next Page Storage Location** **0 = Link partner next pages are stored in the** PHY x Auto-Negotiation Link Partner Base Page Ability Register (PHY_AN_LP_BASE_ABILITY_x) **(PHY register 5)** **1 = Link partner next pages are stored in the** PHY x Auto Negotiation Next Page RX Register (PHY_AN_NP_RX_x) **(PHY register 8)**	RO	1b
4	**Parallel Detection Fault** This bit indicates whether a Parallel Detection Fault has been detected. **0: A fault hasn't been detected via the Parallel Detection function** **1: A fault has been detected via the Parallel Detection function**	RO/LH	0b
3	**Link Partner Next Page Able** This bit indicates whether the link partner has next page ability. **0: Link partner does not contain next page capability** **1: Link partner contains next page capability**	RO	0b
2	**Next Page Able** This bit indicates whether the local device has next page ability. **0: Local device does not contain next page capability** **1: Local device contains next page capability**	RO	1b
1	**Page Received** This bit indicates the reception of a new page. **0: A new page has not been received** **1: A new page has been received**	RO/LH	0b
0	**Link Partner Auto-Negotiation Able** This bit indicates the Auto-Negotiation ability of the link partner. **0: Link partner is not Auto-Negotiation able** **1: Link partner is Auto-Negotiation able**	RO	0b

10.2.18.8 PHY x Auto Negotiation Next Page TX Register (PHY_AN_NP_TX_x)

Index (In Decimal): 7 Size: 16 bits

Bits	Description	Type	Default
15	**Next Page** 0 = No next page ability 1 = Next page capable	R/W	0b
14	**RESERVED**	RO	-
13	**Message Page** 0 = Unformatted page 1 = Message page	R/W	1b
12	**Acknowledge 2** 0 = Device cannot comply with message. 1 = Device will comply with message.	R/W	0b
11	**Toggle** 0 = Previous value was HIGH. 1 = Previous value was LOW.	RO	0b
10:0	**Message Code** Message/Unformatted Code Field	R/W	000 0000 0001b

10.2.18.9 PHY x Auto Negotiation Next Page RX Register (PHY_AN_NP_RX_x)

Index (In Decimal): 8 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15	**Next Page** **0 = No next page ability** **1 = Next page capable**	RO	0b
14	**Acknowledge** **0 = Link code word not yet received from partner** **1 = Link code word received from partner**	RO	0b
13	**Message Page** **0 = Unformatted page** **1 = Message page**	RO	0b
12	**Acknowledge 2** **0 = Device cannot comply with message.** **1 = Device will comply with message.**	RO	0b
11	**Toggle** **0 = Previous value was HIGH.** **1 = Previous value was LOW.**	RO	0b
10:0	**Message Code** Message/Unformatted Code Field	RO	000 0000 0000b

10.2.18.10 PHY x MMD Access Control Register (PHY_MMD_ACCESS)

Index (In Decimal): 13 Size: 16 bits

This register in conjunction with the PHY x MMD Access Address/Data Register (PHY_MMD_ADDR_DATA) provides indirect access to the MDIO Manageable Device (MMD) registers. Refer to the MDIO Manageable Device (MMD) Registers on page 256 for additional details.

Bits	Description	Type	Default
15:14	**MMD Function** This field is used to select the desired MMD function: **00 = Address** **01 = Data, no post increment** **10 = RESERVED** **11 = RESERVED**	R/W	00b
13:5	**RESERVED**	RO	-
4:0	**MMD Device Address (DEVAD)** This field is used to select the desired MMD device address. (3 = PCS, 7 = auto-negotiation)	R/W	0h

10.2.18.11 PHY x MMD Access Address/Data Register (PHY_MMD_ADDR_DATA)

Index (In Decimal): 14 Size: 16 bits

This register in conjunction with the PHY x MMD Access Control Register (PHY_MMD_ACCESS) provides indirect access to the MDIO Manageable Device (MMD) registers. Refer to the MDIO Manageable Device (MMD) Registers on page 256 for additional details.

Bits	Description	Type	Default
15:0	**MMD Register Address/Data** If the MMD Function field of the PHY x MMD Access Control Register (PHY_MMD_ACCESS) is “00”, this field is used to indicate the MMD register address to read/write of the device specified in the MMD Device Address (DEVAD) field. Otherwise, this register is used to read/write data from/to the previously specified MMD address.	R/W	0000h

10.2.18.12 PHY x EDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG_x)

Index (decimal): 16 Size: 16 bits

This register is used to Enable EEE functionality and control NLP pulse generation and the Auto-MDIX Crossover Time of the PHY.

Bits	Description	Type	Default
15	**EDPD TX NLP Enable** Enables the generation of a Normal Link Pulse (NLP) with a selectable interval while in Energy Detect Power-Down. 0=disabled, 1=enabled. The Energy Detect Power-Down (EDPWRDOWN) bit in the PHY x Mode Control/Status Register (PHY_MODE_CONTROL_STATUS_x) needs to be set in order to enter Energy Detect Power-Down mode and the PHY needs to be in the Energy Detect Power-Down state in order for this bit to generate the NLP. The EDPD TX NLP Independent Mode bit of this register also needs to be set when setting this bit.	R/W NASR Note 9	0b
14:13	**EDPD TX NLP Interval Timer Select** Specifies how often a NLP is transmitted while in the Energy Detect Power-Down state. **00b: 1 s** **01b: 768 ms** **10b: 512 ms** **11b: 256 ms**	R/W NASR Note 9	00b
12	**EDPD RX Single NLP Wake Enable** When set, the PHY will wake upon the reception of a single Normal Link Pulse. When clear, the PHY requires two link pluses, within the interval specified below, in order to wake up. Single NLP Wake Mode is recommended when connecting to “Green” network devices.	R/W NASR Note 9	0b
11:10	**EDPD RX NLP Max Interval Detect Select** These bits specify the maximum time between two consecutive Normal Link Pulses in order for them to be considered a valid wake up signal. **00b: 64 ms** **01b: 256 ms** **10b: 512 ms** **11b: 1 s**	R/W NASR Note 9	00b
9:4	**RESERVED**	RO	-
3	**EDPD TX NLP Independent Mode** When set, each PHY port independently detects power down for purposes of the EDPD TX NLP function (via the EDPD TX NLP Enable bit of this register). When cleared, both ports need to be in a power-down state in order to generate TX NLPs during energy detect power-down. Normally set this bit when setting EDPD TX NLP Enable.	R/W NASR Note 9	0b

Bits	Description	Type	Default
2	**PHY Energy Efficient Ethernet Enable (PHYEEEEN)** When set, enables Energy Efficient Ethernet (EEE) operation in the PHY. When cleared, EEE operation is disabled. EEE is not used for EtherCAT.	R/W NASR Note 9	0b Note 10
1	**EDPD Extend Crossover** When in Energy Detect Power-Down (EDPD) mode (Energy Detect Power-Down (EDPWRDOWN) = 1), setting this bit to 1 extends the crossover time by 2976 ms. **0 = Crossover time extension disabled** **1 = Crossover time extension enabled (2976 ms)**	R/W NASR Note 9	0b
0	**Extend Manual 10/100 Auto-MDIX Crossover Time** When Auto-Negotiation is disabled, setting this bit extends the Auto-MDIX crossover time by 32 sample times (32 * 62 ms = 1984 ms). This allows the link to be established with a partner PHY that has Auto-Negotiation enabled. When Auto-Negotiation is enabled, this bit has no affect. It is recommended that this bit is set when disabling AN with Auto-MDIX enabled.	R/W NASR Note 9	1b

Note 9: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

Note 10: EEE is not used for EtherCAT.

10.2.18.13 PHY x Mode Control/Status Register (PHY_MODE_CONTROL_STATUS_x)

Index (decimal): 17 Size: 16 bits

This read/write register is used to control and monitor various PHY configuration options.

Bits	Description	Type	Default
15:14	**RESERVED**	RO	-
13	**Energy Detect Power-Down (EDPWRDOWN)** This bit controls the Energy Detect Power-Down mode. **0: Energy Detect Power-Down is disabled** **1: Energy Detect Power-Down is enabled** **Note:** When in EDPD mode, the device's NLP characteristics can be modified via the PHY x EDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG_x).	R/W	0b
12:7	**RESERVED**	RO	-
6	**ALTINT** Alternate Interrupt Mode: **0 = Primary interrupt system enabled (Default)** **1 = Alternate interrupt system enabled** Refer to Section 10.2.8, "PHY Interrupts," on page 209 for additional information.	R/W NASR Note 11	0b
5:2	**RESERVED**	RO	-
1	**Energy On (ENERGYON)** Indicates whether energy is detected. This bit transitions to "0" if no valid energy is detected within 256 ms (1500 ms if auto-negotiation is enabled). It is reset to "1" by a hardware reset and by a software reset if auto-negotiation was enabled or will be enabled via strapping. Refer to Section 10.2.9.2, "Energy Detect Power-Down," on page 212 for additional information.	RO	1b
0	**RESERVED**	RO	-

Note 11: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

10.2.18.14 PHY x Special Modes Register (PHY_SPECIAL_MODES_x)

Index (decimal): 18 Size: 16 bits

This read/write register is used to control the special modes of the PHY.

Bits	Description	Type	Default
15:11	**RESERVED**	RO	-
10	**RESERVED** **Note:** This but must always be written low.	R/W	0b
9:8	**RESERVED**	RO	-
7:5	**PHY Mode (MODE[2:0])** This field controls the PHY mode of operation. Refer to Table 10-13 for a definition of each mode. **Note:** This field must always be written with its read value. Device strapping must be used to set the desired more.	R/W NASR Note 12	011b or 100b Note 13
4:0	**PHY Address (PHYADD)** The PHY Address field determines the MMI address to which the PHY will respond and is also used for initialization of the cipher (scrambler) key. Each PHY must have a unique address. Refer to Section 10.1.1, "PHY Addressing," on page 201 for additional information. **Note:** No check is performed to ensure that this address is unique from the other PHY addresses (PHY A, PHY B).	R/W NASR Note 12	Note 14

Note 12: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

Note 13: The default value of this field is determined by the ANEG Disable PHY A/ANEG Disable PHY B bits in the Hardware Configuration Register (HW_CFG), which are in turn controlled by configuration strap pins.

Note 14: The default value of this field is determined per Section 10.1.1, "PHY Addressing," on page 201.

TABLE 10-13: MODE[2:0] DEFINITIONS

MODE[2:0]	Mode Definitions
000	10BASE-T Half Duplex. Auto-Negotiation disabled.
001	10BASE-T Full Duplex. Auto-Negotiation disabled.
010	100BASE-TX Half Duplex. Auto-Negotiation disabled. CRS is active during Transmit & Receive.
011	100BASE-TX Full Duplex. Auto-Negotiation disabled. CRS is active during Receive.
100	100BASE-TX Full Duplex is advertised. Auto-Negotiation enabled. CRS is active during Receive.
101	RESERVED
110	Power Down mode.
111	All capable. Auto-Negotiation enabled.

10.2.18.15 PHY x TDR Patterns/Delay Control Register (PHY_TDR_PAT_DELAY_x)

Index (In Decimal): 24 Size: 16 bits

Bits	Description	Type	Default
15	**TDR Delay In** **0 = Line break time is 2 ms.** **1 = The device uses TDR Line Break Counter to increase the line break time before starting TDR.**	R/W NASR Note 15	1b
14:12	**TDR Line Break Counter** When TDR Delay In is 1, this field specifies the increase in line break time in increments of 256 ms, up to 2 seconds.	R/W NASR Note 15	001b
11:6	**TDR Pattern High** This field specifies the data pattern sent in TDR mode for the high cycle.	R/W NASR Note 15	101110b
5:0	**TDR Pattern Low** This field specifies the data pattern sent in TDR mode for the low cycle.	R/W NASR Note 15	011101b

Note 15: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

10.2.18.16 PHY x TDR Control/Status Register (PHY_TDR_CONTROL_STAT_x)

Index (In Decimal): 25 Size: 16 bits

Bits	Description	Type	Default
15	**TDR Enable** **0 = TDR mode disabled** **1 = TDR mode enabled** **Note:** This bit self clears when TDR completes (TDR Channel Status goes high)	R/W NASR SC Note 16	0b
14	**TDR Analog to Digital Filter Enable** **0 = TDR analog to digital filter disabled** **1 = TDR analog to digital filter enabled (reduces noise spikes during TDR pulses)**	R/W NASR Note 16	0b
13:11	**RESERVED**	RO	-
10:9	**TDR Channel Cable Type** Indicates the cable type determined by the TDR test. **00 = Default** **01 = Shorted cable condition** **10 = Open cable condition** **11 = Match cable condition**	R/W NASR Note 16	00b
8	**TDR Channel Status** When high, this bit indicates that the TDR operation has completed. This bit will stay high until reset or the TDR operation is restarted (TDR Enable = 1)	R/W NASR Note 16	0b
7:0	**TDR Channel Length** This eight bit value indicates the TDR channel length during a short or open cable condition. Refer to Section 10.2.13.1, "Time Domain Reflectometry (TDR) Cable Diagnostics," on page 217 for additional information on the usage of this field. **Note:** This field is not valid during a match cable condition. The PHY x Cable Length Register (PHY_CABLE_LEN_x) must be used to determine cable length during a non-open/short (match) condition. Refer to Section 10.2.13, "Cable Diagnostics," on page 217 for additional information.	R/W NASR Note 16	00h

Note 16: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

10.2.18.17 PHY x Symbol Error Counter Register

Index (In Decimal): 26 Size: 16 bits

Bits	Description	Type	Default
15:0	**Symbol Error Counter (SYM_ERR_CNT)** This 100BASE-TX receiver-based error counter increments when an invalid code symbol is received, including IDLE symbols. The counter is incremented only once per packet, even when the received packet contains more than one symbol error. This field counts up to 65,536 and rolls over to 0 if incremented beyond its maximum value. **Note:** This register is cleared on reset, but is not cleared by reading the register. It does not increment in 10BASE-T mode.	RO	0000h

10.2.18.18 PHY x Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND_x)

Index (decimal): 27 Size: 16 bits

This read/write register is used to control various options of the PHY.

Bits	Description	Type	Default
15	**Auto-MDIX Control (AMDIXCTRL)** This bit is responsible for determining the source of Auto-MDIX control for Port x. **0: Port x Auto-MDIX enabled** **1: Port x Auto-MDIX determined by bits 14 and 13**	R/W NASR Note 17	0b
14	**Auto-MDIX Enable (AMDIXEN)** When the AMDIXCTRL bit of this register is set, this bit is used in conjunction with the AMDIXSTATE bit to control the Port x Auto-MDIX functionality as shown in Table 10-14.	R/W NASR Note 17	0b
13	**Auto-MDIX State (AMDIXSTATE)** When the AMDIXCTRL bit of this register is set, this bit is used in conjunction with the AMDIXEN bit to control the Port x Auto-MDIX functionality as shown in Table 10-14.	R/W NASR Note 17	0b
12	**RESERVED**	RO	-
11	**SQE Test Disable (SQEOFF)** This bit controls the disabling of the SQE test (Heartbeat). SQE test is enabled by default. **0: SQE test enabled** **1: SQE test disabled**	R/W NASR Note 17	0b
10:5	**RESERVED**	RO	-
4	**10BASE-T Polarity State (XPOL)** This bit shows the polarity state of the 10BASE-T. **0: Normal Polarity** **1: Reversed Polarity**	RO	0b
3:0	**RESERVED**	RO	-

Note 17: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

TABLE 10-14: AUTO-MDIX ENABLE AND AUTO-MDIX STATE BIT FUNCTIONALITY

Auto-MDIX Enable	Auto-MDIX State	Mode
0	0	Manual mode, no crossover
0	1	Manual mode, crossover
1	0	Auto-MDIX mode
1	1	RESERVED (do not use this state)

10.2.18.19 PHY x Cable Length Register (PHY_CABLE_LEN_x)

Index (In Decimal): 28 Size: 16 bits

Bits	Description	Type	Default
15:12	**Cable Length (CBLN)** This four bit value indicates the cable length. Refer to Section 10.2.13.2, "Matched Cable Diagnostics," on page 220 for additional information on the usage of this field. **Note:** This field indicates cable length for 100BASE-TX linked devices that do not have an open/short on the cable. To determine the open/short status of the cable, the PHY x TDR Patterns/Delay Control Register (PHY_TDR_PAT_DELAY_x) and PHY x TDR Control/Status Register (PHY_TDR_CONTROL_STAT_x) must be used. Cable length is not supported for 10BASE-T links. Refer to Section 10.2.13, "Cable Diagnostics," on page 217 for additional information.	RO	0000b
11:0	**RESERVED - Write as 100000000000b, ignore on read**	R/W	-

10.2.18.20 PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x)

Index (decimal): 29 Size: 16 bits

This read-only register is used to determine to source of various PHY interrupts. All interrupt source bits in this register are read-only and latch high upon detection of the corresponding interrupt (if enabled). A read of this register clears the interrupts. These interrupts are enabled or masked via the PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x).

Bits	Description	Type	Default
15:9	**RESERVED**	RO	-
9	**INT9** This interrupt source bit indicates a Link Up (link status asserted). **0: Not source of interrupt** **1: Link Up (link status asserted)**	RO/LH	0b
8	**INT8** **0: Not source of interrupt** **1: Wake on LAN (WoL) event detected**	RO/LH	0b
7	**INT7** This interrupt source bit indicates when the Energy On (ENERGYON) bit of the PHY x Mode Control/Status Register (PHY_MODE_CONTROL_STATUS_x) has been set. **0: Not source of interrupt** **1: ENERGYON generated**	RO/LH	0b
6	**INT6** This interrupt source bit indicates Auto-Negotiation is complete. **0: Not source of interrupt** **1: Auto-Negotiation complete**	RO/LH	0b
5	**INT5** This interrupt source bit indicates a remote fault has been detected. **0: Not source of interrupt** **1: Remote fault detected**	RO/LH	0b
4	**INT4** This interrupt source bit indicates a Link Down (link status negated). **0: Not source of interrupt** **1: Link Down (link status negated)**	RO/LH	0b
3	**INT3** This interrupt source bit indicates an Auto-Negotiation LP acknowledge. **0: Not source of interrupt** **1: Auto-Negotiation LP acknowledge**	RO/LH	0b

Bits	Description	Type	Default
2	**INT2** This interrupt source bit indicates a Parallel Detection fault. **0: Not source of interrupt** **1: Parallel Detection fault**	RO/LH	0b
1	**INT1** This interrupt source bit indicates an Auto-Negotiation page received. **0: Not source of interrupt** **1: Auto-Negotiation page received**	RO/LH	0b
0	**RESERVED**	RO	-

10.2.18.21 PHY x Interrupt Mask Register (PHY_INTERRUPT_MASK_x)

Index (decimal): 30 Size: 16 bits

This read/write register is used to enable or mask the various PHY interrupts and is used in conjunction with the PHY x Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE_x).

Bits	Description	Type	Default
15:10	**RESERVED**	RO	-
9	**INT9_MASK** This interrupt mask bit enables/masks the Link Up (link status asserted) interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
8	**INT8_MASK** This interrupt mask bit enables/masks the WoL interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
7	**INT7_MASK** This interrupt mask bit enables/masks the ENERGYON interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
6	**INT6_MASK** This interrupt mask bit enables/masks the Auto-Negotiation interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
5	**INT5_MASK** This interrupt mask bit enables/masks the remote fault interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
4	**INT4_MASK** This interrupt mask bit enables/masks the Link Down (link status negated) interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
3	**INT3_MASK** This interrupt mask bit enables/masks the Auto-Negotiation LP acknowledge interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b

Bits	Description	Type	Default
2	**INT2_MASK** This interrupt mask bit enables/masks the Parallel Detection fault interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
1	**INT1_MASK** This interrupt mask bit enables/masks the Auto-Negotiation page received interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
0	**RESERVED**	RO	-

10.2.18.22 PHY x Special Control/Status Register (PHY_SPECIAL_CONTROL_STATUS_x)

Index (decimal): 31 Size: 16 bits

This read/write register is used to control and monitor various options of the PHY.

Bits	Description	Type	Default
15:13	**RESERVED**	RO	-
12	**Autodone** This bit indicates the status of the Auto-Negotiation on the PHY. **0: Auto-Negotiation is not completed, is disabled, or is not active** **1: Auto-Negotiation is completed**	RO	0b
11:5	**RESERVED - Write as 0000010b, ignore on read**	R/W	0000010b
4:2	**Speed Indication** This field indicates the current PHY speed configuration. **STATE** / **DESCRIPTION** 000 RESERVED 001 10BASE-T Half-duplex 010 100BASE-TX Half-duplex 011 RESERVED 100 RESERVED 101 10BASE-T Full-duplex 110 100BASE-TX Full-duplex 111 RESERVED	RO	XXXb
1:0	**RESERVED**	RO	0b

MDIO Manageable Device (MMD) Registers

The device MMD registers adhere to the *IEEE 802.3-2008 45.2 MDIO Interface Registers* specification. The MMD registers are not memory mapped. These registers are accessed indirectly via the PHY x MMD Access Control Register (PHY_MMD_ACCESS) and PHY x MMD Access Address/Data Register (PHY_MMD_ADDR_DATA). The supported MMD device addresses are 3 (PCS), 7 (Auto-Negotiation), and 30 (Vendor Specific). Table 10-15, "MMD Registers" details the supported registers within each MMD device.

TABLE 10-15: MMD REGISTERS

MMD DEVICE ADDRESS (IN DECIMAL)	INDEX (IN DECIMAL)	REGISTER NAME
3 (PCS)	0	PHY x PCS Control 1 Register (PHY_PCS_CTL1_x)
	1	PHY x PCS Status 1 Register (PHY_PCS_STAT1_x)
	5	PHY x PCS MMD Devices Present 1 Register (PHY_PCS_MMD_PRESENT1_x)
	6	PHY x PCS MMD Devices Present 2 Register (PHY_PCS_MMD_PRESENT2_x)
	20	PHY x EEE Capability Register (PHY_EEE_CAP_x)
	22	PHY x EEE Wake Error Register (PHY_EEE_WAKE_ERR_x)
	32784	PHY x Wakeup Control and Status Register (PHY_WUCSR_x)
	32785	PHY x Wakeup Filter Configuration Register A (PHY_WUF_CFGA_x)
	32786	PHY x Wakeup Filter Configuration Register B (PHY_WUF_CFGB_x)
	32801	PHY x Wakeup Filter Byte Mask Registers (PHY_WUF_MASK_x)
	32802	
	32803	
	32804	
	32805	
	32806	
	32807	
	32808	
	32865	PHY x MAC Receive Address A Register (PHY_RX_ADDRA_x)
	32866	PHY x MAC Receive Address B Register (PHY_RX_ADDRB_x)
	32867	PHY x MAC Receive Address C Register (PHY_RX_ADDRC_x)
7 (Auto-Negotiation)	5	PHY x Auto-Negotiation MMD Devices Present 1 Register (PHY_AN_MMD_PRESENT1_x)
	6	PHY x Auto-Negotiation MMD Devices Present 2 Register (PHY_AN_MMD_PRESENT2_x)
	60	PHY x EEE Advertisement Register (PHY_EEE_ADV_x)
	61	PHY x EEE Link Partner Advertisement Register (PHY_EEE_LP_ADV_x)

TABLE 10-15: MMD REGISTERS (CONTINUED)

MMD DEVICE ADDRESS (IN DECIMAL)	INDEX (IN DECIMAL)	REGISTER NAME
30 (Vendor Specific)	2	PHY x Vendor Specific MMD 1 Device ID 1 Register (PHY_VEND_SPEC_MMD1_DEVID1_x)
	3	PHY x Vendor Specific MMD 1 Device ID 2 Register (PHY_VEND_SPEC_MMD1_DEVID2_x)
	5	PHY x Vendor Specific MMD 1 Devices Present 1 Register (PHY_VEND_SPEC_MMD1_PRESENT1_x)
	6	PHY x Vendor Specific MMD 1 Devices Present 2 Register (PHY_VEND_SPEC_MMD1_PRESENT2_x)
	8	PHY x Vendor Specific MMD 1 Status Register (PHY_VEND_SPEC_MMD1_STAT_x)
	14	PHY x Vendor Specific MMD 1 Package ID 1 Register (PHY_VEND_SPEC_MMD1_PKG_ID1_x)
	15	PHY x Vendor Specific MMD 1 package ID 2 Register (PHY_VEND_SPEC_MMD1_PKG_ID2_x)
	16	PHY x Mean Slicer Error Register (PHY_MEAN_ERR_x)
	17	PHY x DCQ Mean Square Error Register (PHY_DCQ_MSE_x)
	18	PHY x DCQ Mean Square Error Worst Case Register (PHY_DCQ_MSE_WC_x)
	19	PHY x DCQ SQI Register (PHY_DCQ_SQI_x)
	20	PHY x DCQ Peak MSE Register (PHY_DCQ_PEAK_MSE_x)
	21	PHY x DCQ Control Register (PHY_DCQ_CTL_x)
	22	PHY x DCQ Configuration Register (PHY_DCQ_CFG_x)
	23-29	PHY x DCQ SQI Table Registers (PHY_DCQ_SQI_TBL_x)

To read or write an MMD register, the following procedure must be observed:

1. Write the PHY x MMD Access Control Register (PHY_MMD_ACCESS) with 00b (address) for the MMD Function field and the desired MMD device (3 for PCS, 7 for Auto-Negotiation) for the MMD Device Address (DEVAD) field.
2. Write the PHY x MMD Access Address/Data Register (PHY_MMD_ADDR_DATA) with the 16-bit address of the desired MMD register to read/write within the previously selected MMD device (PCS or Auto-Negotiation).
3. Write the PHY x MMD Access Control Register (PHY_MMD_ACCESS) with 01b (data) for the MMD Function field and choose the previously selected MMD device (3 for PCS, 7 for Auto-Negotiation) for the MMD Device Address (DEVAD) field.
4. If reading, read the PHY x MMD Access Address/Data Register (PHY_MMD_ADDR_DATA), which contains the selected MMD register contents. If writing, write the PHY x MMD Access Address/Data Register (PHY_MMD_ADDR_DATA) with the register contents intended for the previously selected MMD register.

Unless otherwise specified, reserved fields must be written with zeros if the register is written.

10.2.18.23 PHY x PCS Control 1 Register (PHY_PCS_CTL1_x)

Index (In Decimal): 3.0 Size: 16 bits

Bits	Description	Type	Default
15:0	**RESERVED**	RO	-

10.2.18.24 PHY x PCS Status 1 Register (PHY_PCS_STAT1_x)

Index (In Decimal): 3.1 Size: 16 bits

Bits	Description	Type	Default
15:0	**RESERVED**	RO	-

10.2.18.25 PHY x PCS MMD Devices Present 1 Register (PHY_PCS_MMD_PRESENT1_x)

Index (In Decimal): 3.5 Size: 16 bits

Bits	Description	Type	Default
15:8	**RESERVED**	RO	-
7	**Auto-Negotiation Present** **0 = Auto-negotiation not present in package** **1 = Auto-negotiation present in package**	RO	1b
6	**TC Present** **0 = TC not present in package** **1 = TC present in package**	RO	0b
5	**DTE XS Present** **0 = DTE XS not present in package** **1 = DTE XS present in package**	RO	0b
4	**PHY XS Present** **0 = PHY XS not present in package** **1 = PHY XS present in package**	RO	0b
3	**PCS Present** **0 = PCS not present in package** **1 = PCS present in package**	RO	1b
2	**WIS Present** **0 = WIS not present in package** **1 = WIS present in package**	RO	0b
1	**PMD/PMA Present** **0 = PMD/PMA not present in package** **1 = PMD/PMA present in package**	RO	0b
0	**Clause 22 Registers Present** **0 = Clause 22 registers not present in package** **1 = Clause 22 registers present in package**	RO	0b

10.2.18.26 PHY x PCS MMD Devices Present 2 Register (PHY_PCS_MMD_PRESENT2_x)

Index (In Decimal): 3.6 Size: 16 bits

Bits	Description	Type	Default
15	**Vendor Specific Device 2 Present** **0 = Vendor specific device 2 not present in package** **1 = Vendor specific device 2 present in package**	RO	0b
14	**Vendor Specific Device 1 Present** **0 = Vendor specific device 1 not present in package** **1 = Vendor specific device 1 present in package**	RO	1b
13	**Clause 22 Extension Present** **0 = Clause 22 extension not present in package** **1 = Clause 22 extension present in package**	RO	0b
12:0	**RESERVED**	RO	-

10.2.18.27 PHY x EEE Capability Register (PHY_EEE_CAP_x)

Index (In Decimal): 3.20 Size: 16 bits

Bits	Description	Type	Default
15:7	**RESERVED**	RO	-
6	**10GBASE-KR EEE** **0 = EEE is not supported for 10GBASE-KR** **1 = EEE is supported for 10GBASE-KR** **Note:** The device does not support this mode.	RO	0b
5	**10GBASE-KX4 EEE** **0 = EEE is not supported for 10GBASE-KX4** **1 = EEE is supported for 10GBASE-KX4** **Note:** The device does not support this mode.	RO	0b
4	**10GBASE-KX EEE** **0 = EEE is not supported for 10GBASE-KX** **1 = EEE is supported for 10GBASE-KX** **Note:** The device does not support this mode.	RO	0b
3	**10GBASE-T EEE** **0 = EEE is not supported for 10GBASE-T** **1 = EEE is supported for 10GBASE-T** **Note:** The device does not support this mode.	RO	0b
2	**1000BASE-T EEE** **0 = EEE is not supported for 1000BASE-T** **1 = EEE is supported for 1000BASE-T** **Note:** The device does not support this mode.	RO	0b
1	**100BASE-TX EEE** **0 = EEE is not supported for 100BASE-TX** **1 = EEE is supported for 100BASE-TX**	RO	0b
0	**RESERVED**	RO	-

10.2.18.28 PHY x EEE Wake Error Register (PHY_EEE_WAKE_ERR_x)

Index (In Decimal): 3.22 Size: 16 bits

Bits	Description	Type	Default
15:0	**EEE Wake Error Counter** This counter is cleared to zeros on read and is held to all ones on overflow.	RO/RC	0000h

10.2.18.29 PHY x Wakeup Control and Status Register (PHY_WUCSR_x)

Index (In Decimal): 3.32784 Size: 16 bits

Bits	Description	Type	Default
15:9	**RESERVED**	RO	-
8	**WoL Configured** This bit may be set by software after the WoL registers are configured. This sticky bit (and all other WoL related register bits) is reset only via a power cycle or a pin reset, allowing software to skip programming of the WoL registers in response to a WoL event. **Note:** Refer to Section 10.2.10, "Wake on LAN (WoL)," on page 213 for additional information.	R/W/ NASR Note 18	0b
7	**Perfect DA Frame Received (PFDA_FR)** The PHY sets this bit upon receiving a valid frame with a destination address that matches the physical address.	R/W1C/ NASR Note 18	0b
6	**Remote Wakeup Frame Received (WUFR)** The PHY sets this bit upon receiving a valid remote Wakeup Frame.	R/W1C/ NASR Note 18	0b
5	**Magic Packet Received (MPR)** The PHY sets this bit upon receiving a valid Magic Packet.	R/W1C/ NASR Note 18	0b
4	**Broadcast Frame Received (BCAST_FR)** The PHY Sets this bit upon receiving a valid broadcast frame.	R/W1C/ NASR Note 18	0b
3	**Perfect DA Wakeup Enable (PFDA_EN)** When set, remote wakeup mode is enabled and the PHY is capable of waking up on receipt of a frame with a destination address that matches the physical address of the device. The physical address is stored in the PHY x MAC Receive Address A Register (PHY_RX_ADDRA_x), PHY x MAC Receive Address B Register (PHY_RX_ADDRB_x) and PHY x MAC Receive Address C Register (PHY_RX_ADDRC_x).	R/W/ NASR Note 18	0b
2	**Wakeup Frame Enable (WUEN)** When set, remote wakeup mode is enabled and the PHY is capable of detecting Wakeup Frames as programmed in the Wakeup Filter.	R/W/ NASR Note 18	0b
1	**Magic Packet Enable (MPEN)** When set, Magic Packet wakeup mode is enabled.	R/W/ NASR Note 18	0b
0	**Broadcast Wakeup Enable (BCST_EN)** When set, remote wakeup mode is enabled and the PHY is capable of waking up from a broadcast frame.	R/W/ NASR Note 18	0b

Note 18: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

10.2.18.30 PHY x Wakeup Filter Configuration Register A (PHY_WUF_CFGA_x)

Index (In Decimal): 3.32785 Size: 16 bits

Bits	Description	Type	Default
15	**Filter Enable** **0 = Filter disabled** **1 = Filter enabled**	R/W/ NASR Note 19	0b
14	**Filter Triggered** **0 = Filter not triggered** **1 = Filter triggered**	R/W1C/ NASR Note 19	0b
13:11	**RESERVED**	RO	-
10	**Address Match Enable** When set, the destination address must match the programmed address. When cleared, any unicast packet is accepted. Refer to Section 10.2.10.4, "Wakeup Frame Detection," on page 214 for additional information.	R/W/ NASR Note 19	0b
9	**Filter Any Multicast Enable** When set, any multicast packet other than a broadcast will cause an address match. Refer to Section 10.2.10.4, "Wakeup Frame Detection," on page 214 for additional information. **Note:** This bit has priority over bit 10 of this register.	R/W/ NASR Note 19	0b
8	**Filter Broadcast Enable** When set, any broadcast frame will cause an address match. Refer to Section 10.2.10.4, "Wakeup Frame Detection," on page 214 for additional information. **Note:** This bit has priority over bit 10 of this register.	R/W/ NASR Note 19	0b
7:0	**Filter Pattern Offset** Specifies the offset of the first byte in the frame on which CRC checking begins for Wakeup Frame recognition. Offset 0 is the first byte of the incoming frame's destination address.	R/W/ NASR Note 19	00h

Note 19: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

10.2.18.31 PHY x Wakeup Filter Configuration Register B (PHY_WUF_CFGB_x)

Index (In Decimal): 3.32786 Size: 16 bits

Bits	Description	Type	Default
15:0	**Filter CRC-16** This field specifies the expected 16-bit CRC value for the filter that should be obtained by using the pattern offset and the byte mask programmed for the filter. This value is compared against the CRC calculated on the incoming frame, and a match indicates the reception of a Wakeup Frame.	R/W/ NASR Note 20	0000h

Note 20: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

10.2.18.32 PHY x Wakeup Filter Byte Mask Registers (PHY_WUF_MASK_x)

Index (In Decimal): 3.32801 Size: 16 bits

Bits	Description	Type	Default
15:0	**Wakeup Filter Byte Mask [127:112]**	R/W/ NASR Note 21	0000h

Index (In Decimal): 3.32802 Size: 16 bits

Bits	Description	Type	Default
15:0	**Wakeup Filter Byte Mask [111:96]**	R/W/ NASR Note 21	0000h

Index (In Decimal): 3.32803 Size: 16 bits

Bits	Description	Type	Default
15:0	**Wakeup Filter Byte Mask [95:80]**	R/W/ NASR Note 21	0000h

Index (In Decimal): 3.32804 Size: 16 bits

Bits	Description	Type	Default
15:0	**Wakeup Filter Byte Mask [79:64]**	R/W/ NASR Note 21	0000h

Index (In Decimal): 3.32805 Size: 16 bits

Bits	Description	Type	Default
15:0	**Wakeup Filter Byte Mask [63:48]**	R/W/ NASR Note 21	0000h

Index (In Decimal): 3.32806 Size: 16 bits

Bits	Description	Type	Default
15:0	**Wakeup Filter Byte Mask [47:32]**	R/W/ NASR Note 21	0000h

Index (In Decimal): 3.32807 Size: 16 bits

Bits	Description	Type	Default
15:0	**Wakeup Filter Byte Mask [31:16]**	R/W/ NASR Note 21	0000h

Index (In Decimal): 3.32808 Size: 16 bits

Bits	Description	Type	Default
15:0	**Wakeup Filter Byte Mask [15:0]**	R/W/ NASR Note 21	0000h

Note 21: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

10.2.18.33 PHY x MAC Receive Address A Register (PHY_RX_ADDRA_x)

Index (In Decimal): 3.32865 Size: 16 bits

Bits	Description	Type	Default
15:0	**Physical Address [47:32]**	R/W/ NASR Note 22	FFFFh

Note 22: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

10.2.18.34 PHY x MAC Receive Address B Register (PHY_RX_ADDRB_x)

Index (In Decimal): 3.32866 Size: 16 bits

Bits	Description	Type	Default
15:0	**Physical Address [31:16]**	R/W/ NASR Note 23	FFFFh

Note 23: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

10.2.18.35 PHY x MAC Receive Address C Register (PHY_RX_ADDRC_x)

Index (In Decimal): 3.32867 Size: 16 bits

Bits	Description	Type	Default
15:0	**Physical Address [15:0]**	R/W/ NASR Note 24	FFFFh

Note 24: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY x Basic Control Register (PHY_BASIC_CONTROL_x) is set.

10.2.18.36 PHY x Auto-Negotiation MMD Devices Present 1 Register (PHY_AN_MMD_PRESENT1_x)

Index (In Decimal): 7.5 Size: 16 bits

Bits	Description	Type	Default
15:8	**RESERVED**	RO	-
7	**Auto-Negotiation Present** **0 = Auto-negotiation not present in package** **1 = Auto-negotiation present in package**	RO	1b
6	**TC Present** **0 = TC not present in package** **1 = TC present in package**	RO	0b
5	**DTE XS Present** **0 = DTE XS not present in package** **1 = DTE XS present in package**	RO	0b
4	**PHY XS Present** **0 = PHY XS not present in package** **1 = PHY XS present in package**	RO	0b
3	**PCS Present** **0 = PCS not present in package** **1 = PCS present in package**	RO	1b
2	**WIS Present** **0 = WIS not present in package** **1 = WIS present in package**	RO	0b
1	**PMD/PMA Present** **0 = PMD/PMA not present in package** **1 = PMD/PMA present in package**	RO	0b
0	**Clause 22 Registers Present** **0 = Clause 22 registers not present in package** **1 = Clause 22 registers present in package**	RO	0b

10.2.18.37 PHY x Auto-Negotiation MMD Devices Present 2 Register (PHY_AN_MMD_PRESENT2_x)

Index (In Decimal): 7.6 Size: 16 bits

Bits	Description	Type	Default
15	**Vendor Specific Device 2 Present** **0 = Vendor specific device 2 not present in package** **1 = Vendor specific device 2 present in package**	RO	0b
14	**Vendor Specific Device 1 Present** **0 = Vendor specific device 1 not present in package** **1 = Vendor specific device 1 present in package**	RO	1b
13	**Clause 22 Extension Present** 0 = Clause 22 extension not present in package 1 = Clause 22 extension present in package	RO	0b
12:0	**RESERVED**	RO	-

10.2.18.38 PHY x EEE Advertisement Register (PHY_EEE_ADV_x)

Index (In Decimal): 7.60 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:2	**RESERVED**	RO	-
1	**100BASE-TX EEE** **0 = Do not advertise EEE capability for 100BASE-TX.** **1 = Advertise EEE capability for 100BASE-TX.**	Note 25	0b Note 26
0	**RESERVED**	RO	-

Note 25: This bit is read/write (R/W). However, the user must not set this bit if EEE is disabled.

Note 26: EtherCAT does not use EEE.

10.2.18.39 PHY x EEE Link Partner Advertisement Register (PHY_EEE_LP_ADV_x)

Index (In Decimal): 7.61 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:7	**RESERVED**	RO	-
6	**10GBASE-KR EEE** **0 = Link partner does not advertise EEE capability for 10GBASE-KR.** **1 = Link partner advertises EEE capability for 10GBASE-KR.** **Note:** This device does not support this mode.	RO	0b
5	**10GBASE-KX4 EEE** **0 = Link partner does not advertise EEE capability for 10GBASE-KX4.** **1 = Link partner advertises EEE capability for 10GBASE-KX4.** **Note:** This device does not support this mode.	RO	0b
4	**10GBASE-KX EEE** **0 = Link partner does not advertise EEE capability for 10GBASE-KX.** **1 = Link partner advertises EEE capability for 10GBASE-KX.** **Note:** This device does not support this mode.	RO	0b
3	**10GBASE-T EEE** **0 = Link partner does not advertise EEE capability for 10GBASE-T.** **1 = Link partner advertises EEE capability for 10GBASE-T.** **Note:** This device does not support this mode.	RO	0b
2	**1000BASE-T EEE** **0 = Link partner does not advertise EEE capability for 1000BASE-T.** **1 = Link partner advertises EEE capability for 1000BASE-T.** **Note:** This device does not support this mode.	RO	0b
1	**100BASE-TX EEE** **0 = Link partner does not advertise EEE capability for 100BASE-TX.** **1 = Link partner advertises EEE capability for 100BASE-TX.**	RO	0b
0	**RESERVED**	RO	-

10.2.18.40 PHY x Vendor Specific MMD 1 Device ID 1 Register (PHY_VEND_SPEC_MMD1_DEVID1_x)

Index (In Decimal): 30.2 Size: 16 bits

Bits	Description	Type	Default
15:0	**RESERVED**	RO	0000h

10.2.18.41 PHY x Vendor Specific MMD 1 Device ID 2 Register (PHY_VEND_SPEC_MMD1_DEVID2_x)

Index (In Decimal): 30.3 Size: 16 bits

Bits	Description	Type	Default
15:0	**RESERVED**	RO	0000h

10.2.18.42 PHY x Vendor Specific MMD 1 Devices Present 1 Register (PHY_VEND_SPEC_MMD1_PRESENT1_x)

Index (In Decimal): 30.5 Size: 16 bits

Bits	Description	Type	Default
15:8	**RESERVED**	RO	-
7	**Auto-Negotiation Present** **0 = Auto-negotiation not present in package** **1 = Auto-negotiation present in package**	RO	1b
6	**TC Present** **0 = TC not present in package** **1 = TC present in package**	RO	0b
5	**DTE XS Present** **0 = DTE XS not present in package** **1 = DTE XS present in package**	RO	0b
4	**PHY XS Present** **0 = PHY XS not present in package** **1 = PHY XS present in package**	RO	0b
3	**PCS Present** **0 = PCS not present in package** **1 = PCS present in package**	RO	1b
2	**WIS Present** **0 = WIS not present in package** **1 = WIS present in package**	RO	0b
1	**PMD/PMA Present** **0 = PMD/PMA not present in package** **1 = PMD/PMA present in package**	RO	0b
0	**Clause 22 Registers Present** **0 = Clause 22 registers not present in package** **1 = Clause 22 registers present in package**	RO	0b

10.2.18.43 PHY x Vendor Specific MMD 1 Devices Present 2 Register (PHY_VEND_SPEC_MMD1_PRESENT2_x)

Index (In Decimal): 30.6 Size: 16 bits

Bits	Description	Type	Default
15	**Vendor Specific Device 2 Present** **0 = Vendor specific device 2 not present in package** **1 = Vendor specific device 2 present in package**	RO	0b
14	**Vendor Specific Device 1 Present** **0 = Vendor specific device 1 not present in package** **1 = Vendor specific device 1 present in package**	RO	1b
13	**Clause 22 Extension Present** **0 = Clause 22 extension not present in package** **1 = Clause 22 extension present in package**	RO	0b
12:0	**RESERVED**	RO	-

10.2.18.44 PHY x Vendor Specific MMD 1 Status Register (PHY_VEND_SPEC_MMD1_STAT_x)

Index (In Decimal): 30.8 Size: 16 bits

Bits	Description	Type	Default
15:14	**Device Present** **00 = No device responding at this address** **01 = No device responding at this address** **10 = Device responding at this address** **11 = No device responding at this address**	RO	10b
13:0	**RESERVED**	RO	-

10.2.18.45 PHY x Vendor Specific MMD 1 Package ID 1 Register (PHY_VEND_SPEC_MMD1_PKG_ID1_x)

Index (In Decimal): 30.14 Size: 16 bits

Bits	Description	Type	Default
15:0	**RESERVED**	RO	0000h

10.2.18.46 PHY x Vendor Specific MMD 1 package ID 2 Register (PHY_VEND_SPEC_MMD1_PKG_ID2_x)

Index (In Decimal): 30.15 Size: 16 bits

Bits	Description	Type	Default
15:0	**RESERVED**	RO	0000h

10.2.18.47 PHY x Mean Slicer Error Register (PHY_MEAN_ERR_x)

Index (In Decimal): 30.16 Size: 16 bits

Bits	Description	Type	Default
15:0	**Mean Slicer Error** This field provides the current mean error value. Either absolute or square mode values can be provided. **Note:** This field is updated when the DCQ Read Capture bit in the PHY x DCQ Control Register (PHY_DCQ_CTL_x) is written as a 1.	RO	0000h

10.2.18.48 PHY x DCQ Mean Square Error Register (PHY_DCQ_MSE_x)

Index (In Decimal): 30.17 Size: 16 bits

Bits	Description	Type	Default
15:10	**RESERVED**	RO	-
9	**MSE Value Valid** This field provides the mean square error valid indication. 1 = invalid 0 = valid **Note:** This field is updated when the DCQ Read Capture bit in the PHY x DCQ Control Register (PHY_DCQ_CTL_x) is written as a 1.	RO	0b
8:0	**MSE Value** This field provides the current mean square error value. **Note:** This field is updated when the DCQ Read Capture bit in the PHY x DCQ Control Register (PHY_DCQ_CTL_x) is written as a 1.	RO	000h

10.2.18.49 PHY x DCQ Mean Square Error Worst Case Register (PHY_DCQ_MSE_WC_x)

Index (In Decimal): 30.18 Size: 16 bits

Bits	Description	Type	Default
15:10	**RESERVED**	RO	-
9	**MSE Worst Case Value Valid** This field provides the worst case mean square error valid indication. 1 = invalid 0 = valid **Note:** This field is updated when the DCQ Read Capture bit in the PHY x DCQ Control Register (PHY_DCQ_CTL_x) is written as a 1.	RO	0b
8:0	**MSE Worst Case Value** This field provides the worst case mean square error value since the last time the channel was captured for reading. **Note:** This field is updated when the DCQ Read Capture bit in the PHY x DCQ Control Register (PHY_DCQ_CTL_x) is written as a 1.	RO	000h

10.2.18.50 PHY x DCQ SQI Register (PHY_DCQ_SQI_x)

Index (In Decimal): 30.19 Size: 16 bits

Bits	Description	Type	Default
15:8	**RESERVED**	RO	-
7:5	**SQI Worst Case** This field indicates the worst case SQI value since the last time the channel was captured for reading. **Note:** This field is updated when the DCQ Read Capture bit in the PHY x DCQ Control Register (PHY_DCQ_CTL_x) is written as a 1.	RO	000b
4	**RESERVED**	RO	-
3:1	**SQI** This field indicates the current SQI value. **Note:** This field is updated when the DCQ Read Capture bit in the PHY x DCQ Control Register (PHY_DCQ_CTL_x) is written as a 1.	RO	000b
0	**RESERVED**	RO	-

10.2.18.51 PHY x DCQ Peak MSE Register (PHY_DCQ_PEAK_MSE_x)

Index (In Decimal): 30.20 Size: 16 bits

Bits	Description	Type	Default
15:8	**Peak MSE Worst Case** This field indicates the worst case peak MSE value since the last time the channel was captured for reading. 0-63 = Peak MSE 64-254 = Invalid 255 = measurement not ready **Note:** This field is updated when the DCQ Read Capture bit in the PHY x DCQ Control Register (PHY_DCQ_CTL_x) is written as a 1.	RO	00h
7:0	**Peak MSE Value** This field provides the current peak MSE value. 0-63 = Peak MSE 64-254 = Invalid 255 = measurement not ready **Note:** This field is updated when the DCQ Read Capture bit in the PHY x DCQ Control Register (PHY_DCQ_CTL_x) is written as a 1.	RO	00h

10.2.18.52 PHY x DCQ Control Register (PHY_DCQ_CTL_x)

Index (In Decimal): 30.21 Size: 16 bits

Bits	Description	Type	Default
15	**DCQ Read Capture** When this bit is set the DCQ values are captured.	R/W SC	0b
14:2	**RESERVED**	R/W	-
1:0	**RESERVED**	R/W	00b

10.2.18.53 PHY x DCQ Configuration Register (PHY_DCQ_CFG_x)

Index (In Decimal): 30.22 Size: 16 bits

This register is used to configure the Signal Quality Index function.

Bits	Description	Type	Default
15:14	**scale613** Scaling factor for peak MSE. **Note:** This field only takes effect when sqi_enable changes from a 0 to a 1.	R/W	00b
13:10	**sqi_kp3** LPF bandwidth control.for peak MSE. **Note:** This field only takes effect when sqi_enable changes from a 0 to a 1.	R/W	101b
9:8	**scale611** Scaling factor for MSE and SQI. **Note:** This field only takes effect when sqi_enable changes from a 0 to a 1.	R/W	00b
7	**sqi_reset** When set the SQI logic is reset. **Note:** This bit does not self-clear.	R/W	0b
6	**sqi_squ_mode_en** 0 = Absolute mode 1 = Square mode. **Note:** This field only takes effect when sqi_enable changes from a 0 to a 1.	R/W	1b
5	**sqi_enable** When set SQI measurements are enabled.	R/W	1b
4:0	**sqi_kp** LPF bandwidth control for MSE and SQI. **Note:** This field only takes effect when sqi_enable changes from a 0 to a 1.	R/W	0Eh

10.2.18.54 PHY x DCQ SQI Table Registers (PHY_DCQ_SQI_TBL_x)

Index (In Decimal): 30.23-29 Size: 16 bits

Bits	Description	Type	Default
15:9	**RESERVED**	RO	-
8:0	**SQI_VALUE** Lookup table utilized for implement of SQI. These registers set the thresholds to map the error value to a SQI level. **Note:** This field only takes effect when sqi_enable changes from a 0 to a 1.	R/W	Table 10-16

TABLE 10-16: SQI VALUE DEFAULTS

Register	Default (Hexadecimal)
SQI_TBL1.SQI_VALUE	A3h
SQI_TBL2.SQI_VALUE	82h
SQI_TBL3.SQI_VALUE	67h
SQI_TBL4.SQI_VALUE	52h
SQI_TBL5.SQI_VALUE	41h
SQI_TBL6.SQI_VALUE	34h
SQI_TBL7.SQI_VALUE	29h

11.0 ETHERCAT

11.1 EtherCAT Functional Overview

The EtherCAT module implements a 3 port EtherCAT slave controller with 8K bytes of Dual Port memory (DPRAM), 8 SyncManagers, 8 Fieldbus Memory Management Units (FMMUs) and a 64-bit Distributed Clock.

Each port receives an Ethernet frame, performs frame checking and forwards it to the next port. Time stamps of received frames are generated when they are received. The Loop-back function of each port forwards Ethernet frames to the next logical port if there is either no link at a port, or if the port is not available, or if the loop is closed for that port. The Loop-back function of port 0 forwards the frames to the EtherCAT Processing Unit. The loop settings can be controlled by the EtherCAT master.

Packets are forwarded in the following order: Port 0->EtherCAT Processing Unit->Port 1->Port 2.

The EtherCAT Processing Unit (EPU) receives, analyses and processes the EtherCAT data stream. The main purpose of the EtherCAT Processing unit is to enable and coordinate access to the internal registers and the memory space of the ESC, which can be addressed both from the EtherCAT master and from the local application. Data exchange between master and slave application is comparable to a dual-ported memory (process memory), enhanced by special functions e.g. for consistency checking (SyncManager) and data mapping (FMMU).

Each FMMU performs the task of bitwise mapping of logical EtherCAT system addresses to physical addresses of the device.

SyncManagers are responsible for consistent data exchange and mailbox communication between EtherCAT master and slaves. Each SyncManager's direction and mode of operation is configured by the EtherCAT master. Two modes of operation are available: buffered mode or mailbox mode. In the buffered mode, both the local microcontroller and EtherCAT master can write to the device concurrently. The buffer within the LAN9254 will always contain the latest data. If newer data arrives before the old data can be read out, the old data will be dropped. In mailbox mode, access to the buffer by the local microcontroller and the EtherCAT master is performed using handshakes, guaranteeing that no data will be dropped.

Distributed Clocks (DC) allow for precisely synchronized generation of output signals and input sampling, as well as time stamp generation of events.

The EtherCAT chapter consists of the following main sections:

The EtherCAT module consists of these major blocks:

- EtherCAT Core - These registers provide access to various Switch Fabric parameters for configuration and monitoring.
- EtherCAT CSR and Process Data RAM Access - This block interfaces the EtherCAT Core CSRs to the rest of the device.
- EtherCAT CSR and Process Data RAM Access Registers (Directly Addressable) - These provide the directly addressable System CSRs which are related to the EtherCAT Core.

Refer to Figure 2-2 for an overview of the interconnection of the EtherCAT module within the device.

11.2 Distributed Clocks

The device supports 64-bit distributed clocks as detailed in the following sub-sections.

11.2.1 SYNC/LATCH PIN MULTIPLEXING

The EtherCAT Core provides two input pins (**LATCH0** and **LATCH1**) which are used for time stamping of external events. Both rising edge and falling edge time stamps are recorded. These pins are shared with the **SYNC0** and **SYNC1** output pins, respectively, which are used to indicate the occurrence of time events. The functions of the **SYNC0/LATCH0** and **SYNC1/LATCH1** pins are determined by the SYNC0/LATCH0 Configuration and SYNC1/LATCH1 Configuration bits of the Sync/Latch PDI Configuration Register, respectively.

When set for **SYNC0/SYNC1** functionality, the output type (Push-Pull vs. Open Drain/Source) and output polarity are determined by the SYNC0 Output Driver/Polarity and SYNC1 Output Driver/Polarity bits of the Sync/Latch PDI Configuration Register.

When **SYNC0** is selected, **LATCH0** is potentially remapped onto an alternate pin (Table 3-1) depending on the device configuration. Similarly, when **SYNC1** is selected, **LATCH1** is potentially remapped onto an alternate pin (Table 3-1) depending on the device configuration. This mapping is detailed in Table 11-1.

Note:	If not remapped to an alternate pin, **LATCH0/LATCH1** is held low when **SYNC0/SYNC1** are enabled as outputs.

TABLE 11-1: LATCH0/1 MAPPING

Signal	HBI Indexed Mode Pin Name	HBI Multiplexed Mode Pin Name	SPI with GPIO Mode Pin Name	SPI with MII Mode Pin Name	Digital I/O Mode Pin Name
LATCH0	**LATCH0** remapped				
LATCH1	**LATCH1** remapped				

Note:	The Sync/Latch PDI Configuration Register is initialized from the contents of EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

11.2.2 SYNC IRQ MAPPING

The **SYNC0** and **SYNC1** states can be mapped into the State of DC SYNC0 and State of DC SYNC1 bits of the AL Event Request Register, respectively. The mapping of the **SYNC0** and **SYNC1** states is enabled by the SYNC0 Map and SYNC1 Map bits of the Sync/Latch PDI Configuration Register, respectively.

Note:	The Sync/Latch PDI Configuration Register is initialized from the contents of EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

11.2.3 SYNC PULSE LENGTH

The **SYNC0** and **SYNC1** pulse length is controlled via the Pulse Length of SyncSignals Register. The Pulse Length of SyncSignals Register is initialized from the contents of EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

11.2.4 SYNC/LATCH I/O TIMING REQUIREMENTS

This section specifies the **SYNC0/LATCH0** and **SYNC1/LATCH1** input and output timings.

FIGURE 11-1: ETHERCAT SYNC/LATCH TIMING DIAGRAM

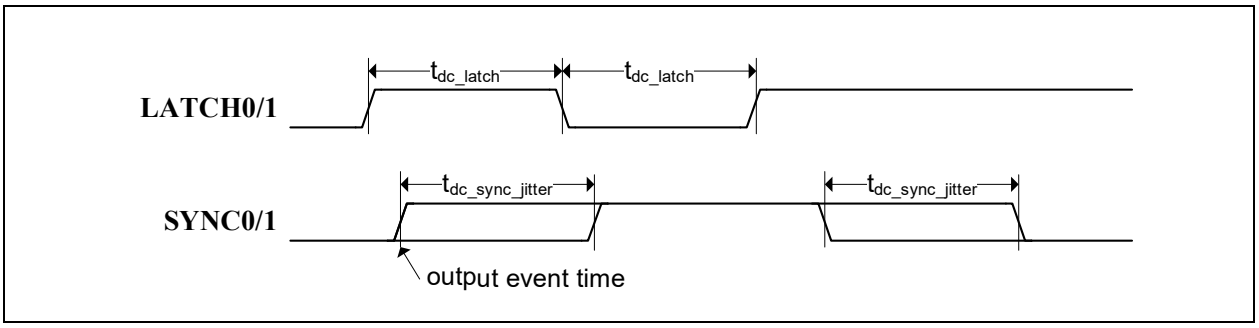


TABLE 11-2: ETHERCAT SYNC/LATCH TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{dc_latch}	Time between **LATCH0** or **LATCH1** events	15	-	-	ns
$t_{dc\_sync\_jitter}$	**SYNC0** or **SYNC1** output jitter	-	-	15	ns

11.3 PDI Selection and Configuration

The Process Data Interface (PDI) used by the device is indicated via the PDI Control Register. The available PDIs are:

- **04h:** Digital I/O PDI
- **05h:** Beckhoff SPI Controller
- **80h, 82h:** SPI Interface
- **88h, 8Ah, 8Ch, 8Eh:** Host Bus Interface (HBI)
  (8-bit, HBI Multiplexed/Demultiplexed/Indexed 1/2 Phase, LAN9252 Compatible)
- **89h, 8Bh, 8Dh, 8Fh:** Host Bus Interface (HBI)
  (16-bit, HBI Multiplexed/Demultiplexed/Indexed 1/2 Phase, LAN9252 Compatible)
- **90h, 92h, 94h, 96h:** Host Bus Interface (HBI)
  (8-bit, HBI Multiplexed/Demultiplexed/Indexed 1/2 Phase, EtherCAT Direct Map)
- **91h, 93h, 95h, 97h:** Host Bus Interface (HBI)
  (16-bit, HBI Multiplexed/Demultiplexed/Indexed 1/2 Phase, EtherCAT Direct Map)

Note: The PDI Control Register can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

The PDI Control Register is initialized from the contents of EEPROM.

The configuration of the enabled PDI is controlled via the PDI Configuration Register and Extended PDI Configuration Register. The definition of these registers depends on the selected mode of operation. However, only one register set exists.

11.4 Digital I/O PDI

The Digital I/O PDI provides 32 configurable digital I/Os (**DIGIO[31:0]**) to be used for simple systems without a host controller. The Digital I/O Output Data Register is used to control the output values, while the Digital I/O Input Data Register is used to read the input values. Each 2-bit pair of the digital I/Os is configurable as an input or output. The direction is selected by the Extended PDI Configuration Register, which is configured via EEPROM (Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information). The Digital I/Os can also be configured to bi-directional mode, where the outputs are driven and latched externally and then released so that the input data can be sampled. Bi-directional operation is selected via the Unidirectional/Bidirectional Mode bit of the PDI Configuration Register. The PDI Configuration Register is initialized from the contents of EEPROM.

11.4.1 OUTPUT WATCHDOG BEHAVIOR

The watchdog control of the digital outputs can be configured to specify if the expiration of the SyncManager Watchdog will have an immediate effect on the I/O signals (output reset immediately after watchdog timeout) or if the effect is delayed until the next output event (output reset with next output event). The choice is determined by the Watchdog Behavior bit of the PDI Configuration Register. The PDI Configuration Register is initialized from the contents of EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

11.4.2 OE_EXT OUTPUT WATCHDOG BEHAVIOR

For external watchdog implementations, the **WD_TRIG** (watchdog trigger) pin can be used. A pulse is generated if the SyncManager Watchdog is triggered. In this case, the internal SyncManager Watchdog should be disabled, and the external watchdog may use the **OE_EXT** pin to reset the I/O signals if the watchdog is expired.

The OUTVALID Mode bit of the PDI Configuration Register controls if **WD_TRIG** is mapped onto the **OUTVALID** pin. The PDI Configuration Register is initialized from the contents of EEPROM. Since there is a dedicated **WD_TRIG** pin, this bit is normally set to 0 in the EEPROM.

11.4.3 INPUT DATA SAMPLING

Digital inputs can be configured to be sampled in four ways, at the start of each Ethernet frame, at the rising edge of the **LATCH_IN** pin, at Distributed Clocks **SYNC0** events or at Distributed Clocks **SYNC1** events. The choice of sampling mode is determined by the Input Data Sample Selection bits of the PDI Configuration Register. The PDI Configuration Register is initialized from the contents of EEPROM.

11.4.4 OUTPUT DATA UPDATING

Digital outputs can be configured to be update four ways, at the end of each Ethernet frame, with Distributed Clocks **SYNC0** events, with Distributed Clocks **SYNC1** events or at the end of an EtherCAT frame which triggered the Process Data Watchdog. The choice of sampling mode is determined by the Output Data Sample Selection bits of the PDI Configuration Register. The PDI Configuration Register is initialized from the contents of EEPROM.

11.4.5 OUTVALID POLARITY

The output polarity of the **OUTVALID** pin is determined by the OUTVALID Polarity bit of the PDI Configuration Register. The PDI Configuration Register is initialized from the contents of EEPROM.

11.4.6 DIGITAL I/O TIMING REQUIREMENTS

This section specifies the **DIGIO[31:0]**, **LATCH_IN** and **SOF** input and output timings.

FIGURE 11-2: ETHERCAT DIGITAL I/O INPUT TIMING DIAGRAM

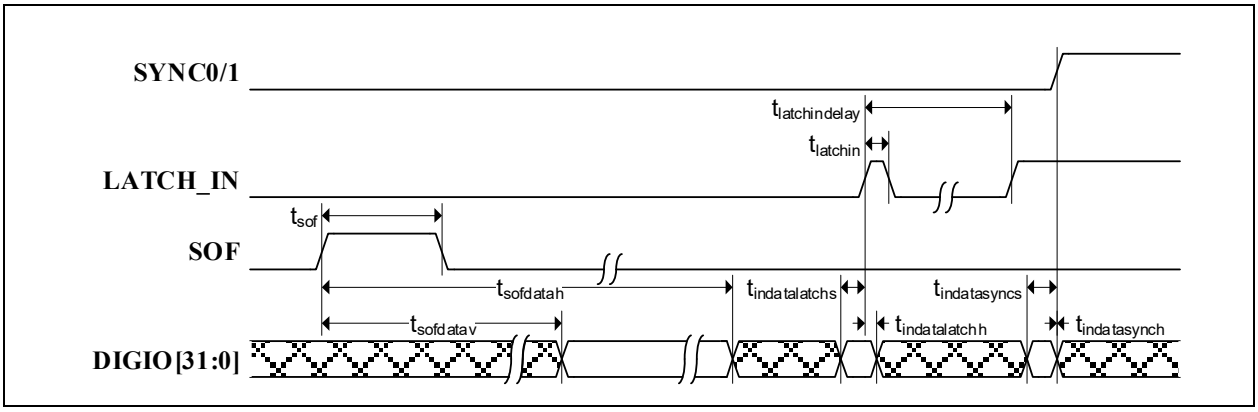


FIGURE 11-3: ETHERCAT DIGITAL I/O OUTPUT TIMING DIAGRAM

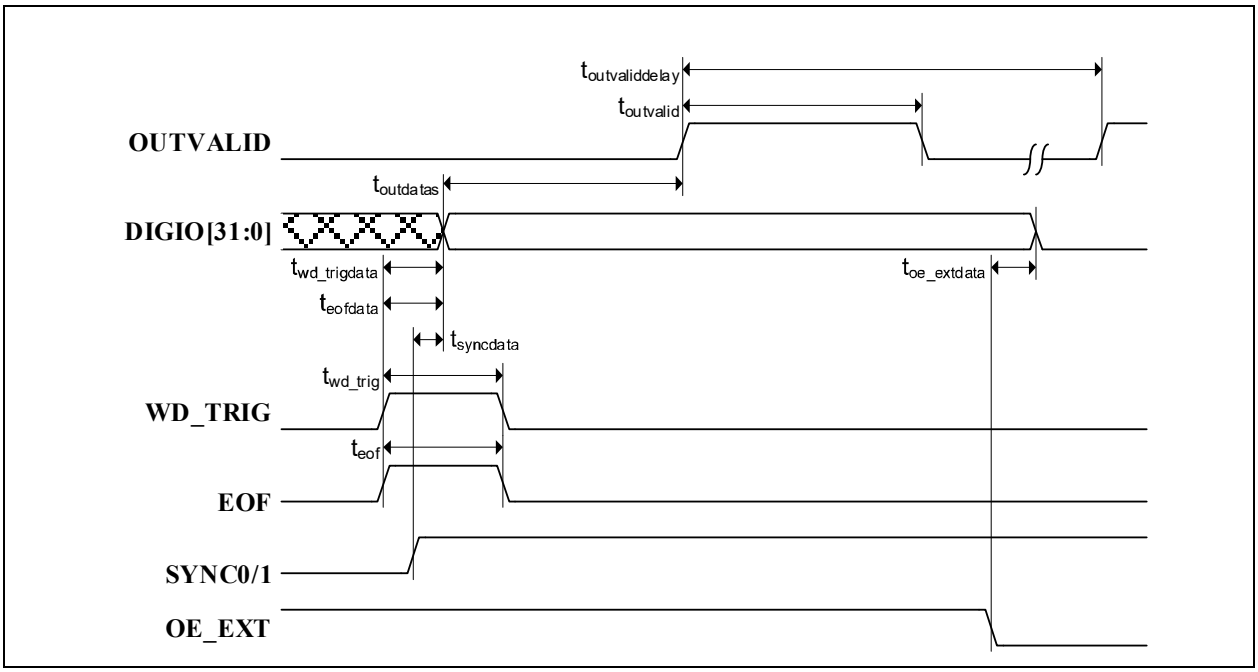


FIGURE 11-4: ETHERCAT DIGITAL I/O BI-DIRECTIONAL TIMING DIAGRAM

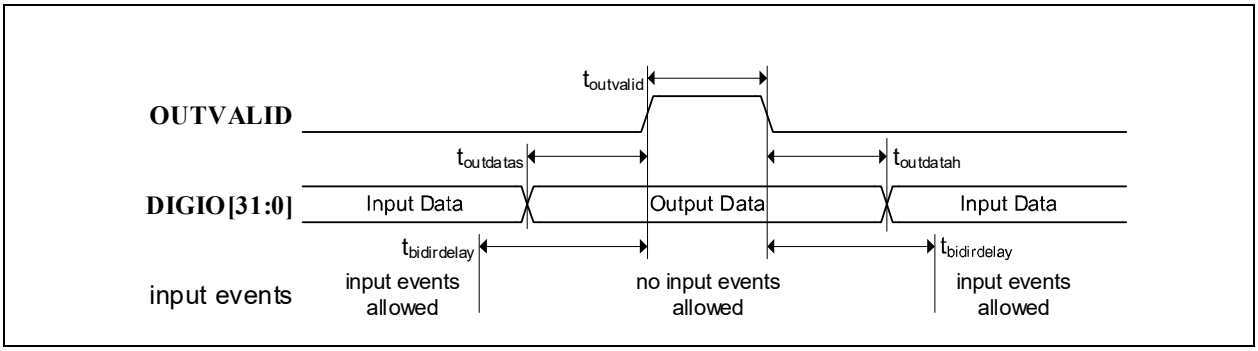


TABLE 11-3: ETHERCAT DIGITAL I/O TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
$t_{indatasyncs}$	Input data setup to **SYNC0/1** rising	10	-	-	ns
$t_{indatasynch}$	Input data hold from **SYNC0/1** rising	0	-	-	ns
$t_{indatalatchs}$	Input data setup to **LATCH_IN** rising	8	-	-	ns
$t_{indatalatchh}$	Input data hold from **LATCH_IN** rising	4	-	-	ns
$t_{latchin}$	**LATCH_IN** high time	8	-	-	ns
$t_{latchindelay}$	time between consecutive input events	440	-	-	ns
t_{sof}	**SOF** high time	35	-	45	ns
$t_{sofdatav}$	Input data valid after **SOF** active, so that input data can be read in the same frame	-	-	1.2	µs
$t_{sofdatah}$	Input data hold after **SOF** active, so that input data can be read in the same frame	1.6	-	-	µs
$t_{outdatas}$	Output data setup to **OUTVALID** rising	65	-	-	ns
$t_{outdatah}$	Output data hold from **OUTVALID** falling	65	-	-	ns
$t_{outvalid}$	**OUTVALID** high time	75	-	85	ns
$t_{outvaliddelay}$	time between consecutive output events	320	-	-	ns
t_{eof}	**EOF** high time	35	-	45	ns
$t_{eofdata}$	Output data valid after **EOF**	-	-	35	ns
t_{wd_trig}	**WD_TRIG** high time	35	-	45	ns
$t_{wd\_trigdata}$	Output data valid after **WD_TRIG**	-	-	35	ns
$t_{syncdata}$	Output data valid after **SYNC0/1**	-	-	25	ns
$t_{oe\_extdata}$	**OE_EXT** to data low	0	-	15	ns
$t_{bidirdelay}$	time between consecutive input or output events	440	-	-	ns

11.5 SPI Interface

This interface is used for systems with a host controller that provides a compatible SPI host interface. It supports both a LAN9252 compatibility mode as well as an EtherCAT Direct Mapped mode.

The values in the PDI Configuration Register and the Extended PDI Configuration Register reflect the value from EEPROM. The value in the PDI Configuration Register is used to configure the SPI interface. The value in the Extended PDI Configuration Register is used if GPIOs are enabled (SPI w/GPIO).

The PDI Configuration Register and Extended PDI Configuration Register are initialized from the contents of the EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

11.6 Host Bus Interface (HBI)

This interface is used for systems with a Host Bus Interface. It supports both a LAN9252 compatibility mode as well as an EtherCAT Direct Mapped mode.

The values in the PDI Configuration Register and the Extended PDI Configuration Register reflect the value from EEPROM. The values in the PDI Configuration Register are used to configure the HBI. The values in the Extended PDI Configuration Register are used to further configure the HBI while in EtherCAT Direct Mapped mode.

The PDI Configuration Register and Extended PDI Configuration Register are initialized from the contents of the EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

11.7 Beckhoff SPI Controller

The Beckhoff SPI interface is used as an alternate SPI interface.

The value in the PDI Configuration Register reflects the value from EEPROM. The value in the PDI Configuration Register is used to configure the SPI. The value in the Extended PDI Configuration Register is used if GPIOs are enabled (SPI w/GPIO).

The PDI Configuration Register and Extended PDI Configuration Register are initialized from the contents of the EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

11.8 GPIOs

The EtherCAT Core provides 16 General Purpose Inputs (**GPI[15:0]**) and 16 General Purpose Outputs (**GPO[15:0]**) The General Purpose Output Register is used to control the output value. The General Purpose Input Register is used to read the input value.

Note:	When GPIOs are not available due to chip configuration, the General Purpose Output Register remains R/W, but has no effect. When GPIOs are not available due to chip configuration, the General Purpose Input Register will return zeros.

Each 2-bit pair is configurable as input, push-pull output or open-drain output. The direction and buffer type are determined by the Extended PDI Configuration Register. Bits 7:0 control the direction of the pairs (bit 0 for GPIO[1:0], bit 1 for GPIO[3:2], etc.). A value of 1 selects the output direction. Bits 15:8 control the output type (bit 8 for GPIO[1:0], bit 9 for GPIO[3:2], etc.). A value of 1 selects the open-drain. The Extended PDI Configuration Register is initialized from the contents of the EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

Note:	The Extended PDI Configuration Register is also used for the Digital I/O PDI direction. However, GPIOs are not used during Digital I/O mode.

11.9 User RAM

A 128 byte user RAM is located at 0F80h-0FFFh. The default values within this RAM are undefined for all addresses.

11.10 EEPROM Configurable Registers

The following registers are configurable via EEPROM. Refer to the corresponding register definition for details on each bit function.

Note: Reserved bits must be written as 0 unless otherwise noted.

TABLE 11-4: ETHERCAT CORE EEPROM CONFIGURABLE REGISTERS

Register	Bits	EEPROM Word / [Bits]
PDI Control Register (0140h)	[7:0] Process Data Interface (PDI_SELECT)	0 / [7:0]
ESC Configuration Register (0141h)	[7] (unused)	0 / [15]
	[6] Enhanced Link Port 2	0 / [14]
	[5] Enhanced Link Port 1	0 / [13]
	[4] Enhanced Link Port 0	0 / [12]
	[3] Distributed Clocks Latch In Unit **Note:** Bit 3 is NOT set by EEPROM	-
	[2] Distributed Clocks SYNC Out Unit **Note:** Bit 2 is NOT set by EEPROM	-
	[1] Enhanced Link Detection All Ports	0 / [9]
	[0] Device Emulation (control of AL Status Register)	0 / [8]
PDI Configuration Register (0150h) Digital I/O Mode	[7:6] Output Data Sample Selection	1 / [7:6]
	[5:4] Input Data Sample Selection	1 / [5:4]
	[3] Watchdog Behavior	1 / [3]
	[2] Unidirectional/Bidirectional Mode	1 / [2]
	[1] OUTVALID Mode	1 / [1]
	[0] OUTVALID Polarity	1 / [0]
PDI Configuration Register (0150h) SPI Mode	[7:3] RESERVED	1 / [7:3]
	[2] SPI AL Event Request & INT_STS Enable	1 / [2]
	[1] SPI WAIT_ACK Polarity	1 / [1]
	[0] SPI WAIT_ACK Buffer Type	1 / [0]
PDI Configuration Register (0150h) Beckhoff SPI Mode	[7:6] RESERVED	1 / [7:6]
	[5] Data Out Sample Mode	1 / [5]
	[4] SCS# Polarity	1 / [4]
	[3:2] RESERVED	1 / [3:2]
	[1:0] SPI Mode	1 / [1:0]
PDI Configuration Register (0150h) HBI Mode	[7] HBI ALE Qualification	1 / [7]
	[6] HBI Read/Write Mode	1 / [6]
	[5] HBI Chip Select Polarity	1 / [5]
	[4] HBI Read, Read/Write Polarity	1 / [4]
	[3] HBI Write, Enable Polarity	1 / [3]
	[2] HBI ALE Polarity	1 / [2]
	[1] HBI WAIT_ACK Polarity	1 / [1]
	[0] HBI WAIT_ACK Buffer Type	1 / [0]

TABLE 11-4: ETHERCAT CORE EEPROM CONFIGURABLE REGISTERS (CONTINUED)

Register	Bits	EEPROM Word / [Bits]
Sync/Latch PDI Configuration Register (0151h)	[7] SYNC1 Map	1 / [15]
	[6] SYNC1/LATCH1 Configuration	1 / [14]
	[5:4] SYNC1 Output Driver/Polarity	1 / [13:12]
	[3] SYNC0 Map	1 / [11]
	[2] SYNC0/LATCH0 Configuration	1 / [10]
	[1:0] SYNC0 Output Driver/Polarity	1 / [9:8]
Pulse Length of SyncSignals Register (0982h-0983h)	[15:0] Pulse length of SyncSignals	2 / [15:0]
Extended PDI Configuration Register (0152h-0153h) Digital I/O Mode	[15:8] I/O 31-16 Direction	3 / [15:8]
	[7:0] I/O 15-0 Direction	3 / [7:0]
Extended PDI Configuration Register (0152h-0153h) SPI w/GPIO Mode	[15:8] I/O 15-0 Buffer Type	3 / [15:8]
	[7:0] I/O 15-0 Direction	3 / [7:0]
Extended PDI Configuration Register (0152h-0153h) HBI Modes	[15:3] RESERVED	3 / [15:3]
	[2] HBI BE1/BE0 Polarity	3 / [2]
	[1] Perform Internal Write	3 / [1]
	[0] Read WAIT_ACK Removal Delay	3 / [0]
Configured Station Alias Register (0012h-0013h)	[15:0] Configured Station Alias Address	4 / [15:0]
MII Management Control/Status Register (0510h-0511h)	[2] MI Link Detection	5 / [15]
ASIC Configuration Register (0142h-0143h)	[15] MI Link Detection	
	[14] ERRLED Enable	5 / [14]
	[13:8] RESERVED	5 / [13:8]
	[7] MI Write Gigabit Register 9 Enable	5 / [7]
	[6] STATE_RUNLED Mode Select	5 / [6]
	[5:0] RESERVED	5 / [5:0]
RESERVED Register (0144h-0145h)	[15:0] RESERVED	6 / [15:0]

11.11 Port Interfaces

11.11.1 PORTS 0 AND 2 (INTERNAL PHY A OR EXTERNAL MII)

Port 0 of the EtherCAT Slave is connected to internal PHY A when the **CHIP_MODE[1:0]** configuration straps are not equal to 11b (2 port mode or 3 port downstream mode). Port 0 is connected to the MII pins when the **CHIP_MODE[1:0]** configuration straps are equal to 11b (3 port upstream mode).

Port 2 of the EtherCAT Slave is connected to internal PHY A when the **CHIP_MODE[1:0]** configuration straps are equal to 11b (3 port upstream mode). Port 2 is connected to the MII pins when the **CHIP_MODE[1:0]** configuration straps are equal to 10b (3 port downstream mode).

11.11.1.1 EXTERNAL MII PHY CONNECTION

An external PHY is connected to the MII port as shown in Figure 11-5. The clock source for the Ethernet PHY and the EtherCAT Slave must be the same. A 25 MHz output (**MII_CLK25**) is provided to be used as the reference clock for the PHY. TX_CLK from the PHY is not connected since the EtherCAT Slave does not incorporate a TX FIFO. The TX signals from the EtherCAT Slave may be delayed with respect to the CLK25 output by using TX shift compensation so that they align properly as if they were driven by the PHY's TX_CLK. MII timing is described in Section 11.11.7, "External PHY Timing".

The Ethernet PHY should be connected to the EtherCAT Slave **RST#** pin so that the PHY is held in reset until the EtherCAT Slave is ready. Otherwise, the far end Link Partner would detect valid link signals from the PHY and would "open" its port assuming that the local EtherCAT Slave was ready.

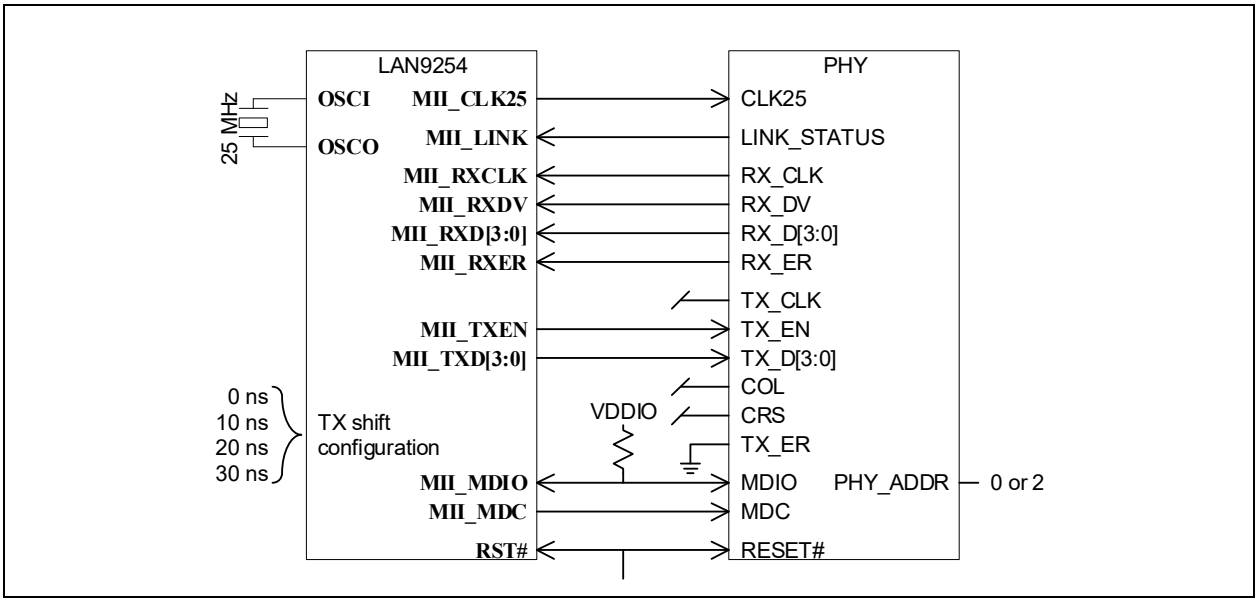
The **MII_MDC** and **MII_MDIO** signals are connected between the EtherCAT slave and the PHY. **MII_MDIO** requires an external pull-up. The management address of the external PHY must be set to 0 when the **CHIP_MODE[1:0]** configuration straps are equal to 11b (3 port upstream mode) and to 2 when the **CHIP_MODE[1:0]** configuration straps are equal to 10b (3 port downstream mode).

LINK_STATUS from the PHY is an LED output which indicates that a 100 Mbit/s, Full Duplex link is active. The polarity of the **MII_LINK** input of the EtherCAT slave is configurable.

The COL and CRS outputs from the PHY are not connected since EtherCAT operates in full-duplex mode.

The TX_ER input to the PHY is tied to system ground since the EtherCAT Slave never generates transmit errors.

FIGURE 11-5: ETHERCAT EXTERNAL PHY CONNECTION



11.11.1.2 BACK-TO-BACK CONNECTION

Two EtherCAT Slave devices can be connected using a back-to-back MII connection as shown in Figure 11-6. One device is placed in 3 port upstream mode and the other in 3 port downstream mode.

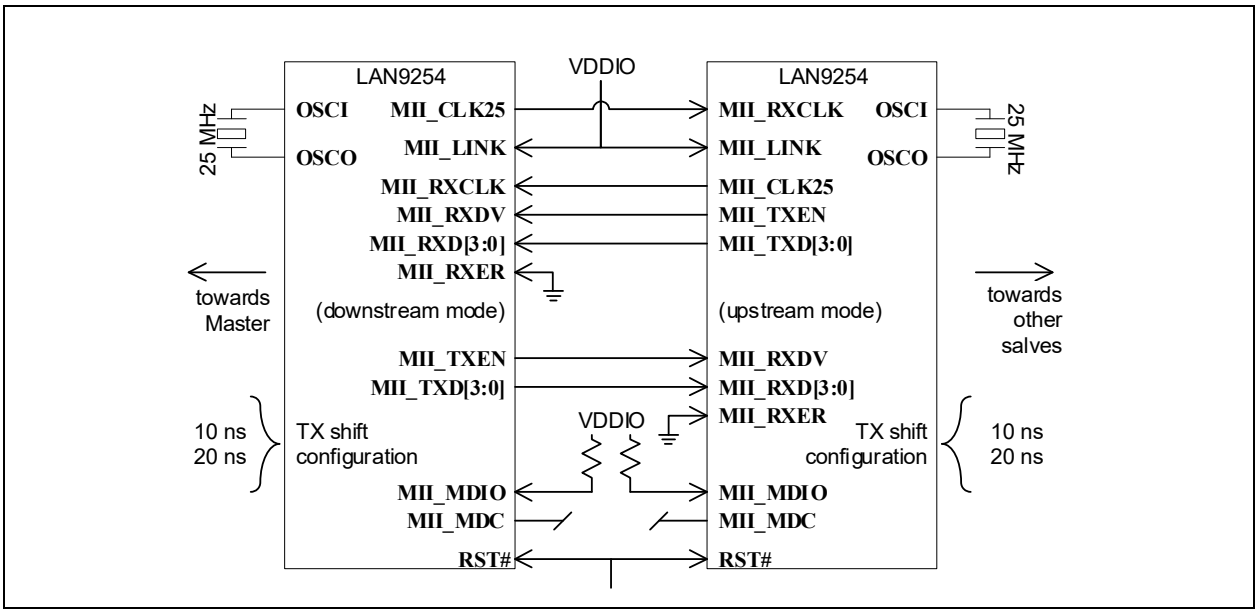
The clock sources of each EtherCAT Slave may be different. The 25 MHz output (**MII_CLK25**) is provided to be used as the **RX_CLK** input to the other device. The TX signals from each EtherCAT Slave may be delayed with respect to the CLK25 output by using TX shift compensation so that they align properly to meet the RX timing requirement of the other device. Back-to-back MII timing is described in Section 11.11.7, "External PHY Timing".

The **MII_RXER** signals are not used since the EtherCAT Slaves never generate errors.

The **MII_MDIO** and **MII_MDC** signals are not used since neither device contains a PHY register set. The **MII_MDIO** pins require (separate) pull-ups so that a high value is returned when PHY register reads are attempted.

MII_LINK may be tied active, if the two EtherCAT slaves are released from reset at about the same time. Otherwise, **MII_LINK** can be used to indicate to the partner that the device is not ready.

FIGURE 11-6: ETHERCAT BACK-TO-BACK MII CONNECTION



11.11.1.3 2 PORT OPERATION

When configured for two port mode (**CHIP_MODE[1:0]** equal to 00b or 01b), port 2 is disabled. The port status is also shown in the Port 2 Configuration bits of the Port Descriptor Register and is set to a 01b (Not configured) when the device is configured for two port operation.

11.11.2 PORT 1 (INTERNAL PHY B)

Port 1 of the EtherCAT core is always connected to internal PHY B.

11.11.3 PHY CONFIGURATION

By default, the internal PHYs are configured for 100Mbps, full-duplex operation with Auto-Negotiation enabled. The EtherCAT Core will also check and update the configuration if necessary. Fixed 100Mbps full-duplex can be enabled, as detailed in Section 11.11.6, Fixed 100Mbps Full-Duplex Support.

By default, the external PHY is configured for 100Mbps, full-duplex operation with Auto-Negotiation enabled. The EtherCAT Core will check and update the configuration if necessary.

11.11.4 PHY LINK STATUS

The link status originates from the PHY's link signal (internal or external). The EtherCAT Core also checks the PHY status to determine a proper link. By cyclically polling the PHYs, it checks that Auto-negotiation registers are configure properly, if a link is established, if Auto-Negotiation has finished successfully, and if the link partner also used Auto-Negotiation. Fixed 100Mbps full-duplex can be enabled, as detailed in Section 11.11.6, Fixed 100Mbps Full-Duplex Support.

Link checking through the MII Management Interface (MI) is enabled via EEPROM and reflected in the MII Management Control/Status Register.

Note:	MI link detection and configuration is disabled until the device is successfully configured from the EEPROM. The EEPROM setting for MI link detection and configuration is only taken at the first EEPROM loading after power-on or reset. Changing the EEPROM and manually reloading it will not affect the MI link detection enable status, even if the EEPROM could not be read initially.

As shown in Table 11-4, "EtherCAT Core EEPROM Configurable Registers", bit 7 of the ASIC Configuration Register is used to enable writes to PHY register 9 for PHYs which use this register per IEEE 802.3. The ASIC Configuration Register is initialized from the contents of the EEPROM.

11.11.4.1 MI LINK DETECTION AND CONFIGURATION STATE MACHINE

The MI Link Detection and Configuration state machine operates as follows:

- Check that auto-negotiation is enabled
- Check that only 100BASE-X full-duplex is advertised
- Check that 1000BASE-T is not advertised
- Check that auto-negotiation is completed
- Check that link partner is 100BASE-X full-duplex
- Otherwise, set the registers as needed and restart auto-negotiation

11.11.5 ENHANCED LINK DETECTION

The EtherCAT Core supports the enhanced link detection feature and the enable is controlled by the EEPROM. With this, the EtherCAT Core will disconnect a link if at least 32 RX errors (RX_ER) occur in a fixed interval of time (~10 us). Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

11.11.6 FIXED 100MBPS FULL-DUPLEX SUPPORT

In order to support special use cases, Auto-Negotiation may be disabled on the internal PHYs, with the speed set to 100Mbps and the duplex set to full. This is determined by the ANEG Disable PHY A and ANEG Disable PHY B bits in the Hardware Configuration Register (HW_CFG), which are in turn controlled by the **100FD_A** and **100FD_B** configuration strap pins.

The MI Link Detection and Configuration function, which normally detects and enforces an auto-negotiated 100Mbps full-duplex link, will instead detect and enforce a fixed 100Mbps full-duplex link in this mode.

If Enhanced Link Detection detects an error condition, it would normally restart auto-negotiation (along with setting Auto-Negotiation Enable, Full-Duplex and Speed Select LSB) by writing a value of 0x3300 to the PHY x Basic Control Register (PHY_BASIC_CONTROL_x). With ANEG Disable PHY A/ANEG Disable PHY B set, the corresponding PHY is instead reset by setting the Soft Reset bit (along with Full-Duplex and Speed Select LSB) by writing a value of 0xA100 to the PHY x Basic Control Register (PHY_BASIC_CONTROL_x).

11.11.7 EXTERNAL PHY TIMING

Since the EtherCAT Core does not use the PHY transmit clock, proper timing must be ensured based on the common 25 MHz reference clock (which is output to the external PHY via the **MII_CLK25** pin). To aid in this, the EtherCAT Core has the TX shift feature enabled. This feature can delay the generation of the transmit signals from the EtherCAT Core by 0ns, 10ns, 20ns or 30ns. This value can be manually set using the **TX_SHIFT[1:0]** configuration straps.

11.11.7.1 MII Connection Timing

The MII interface TX and RX timing is as follows:

FIGURE 11-7: MII TX TIMING

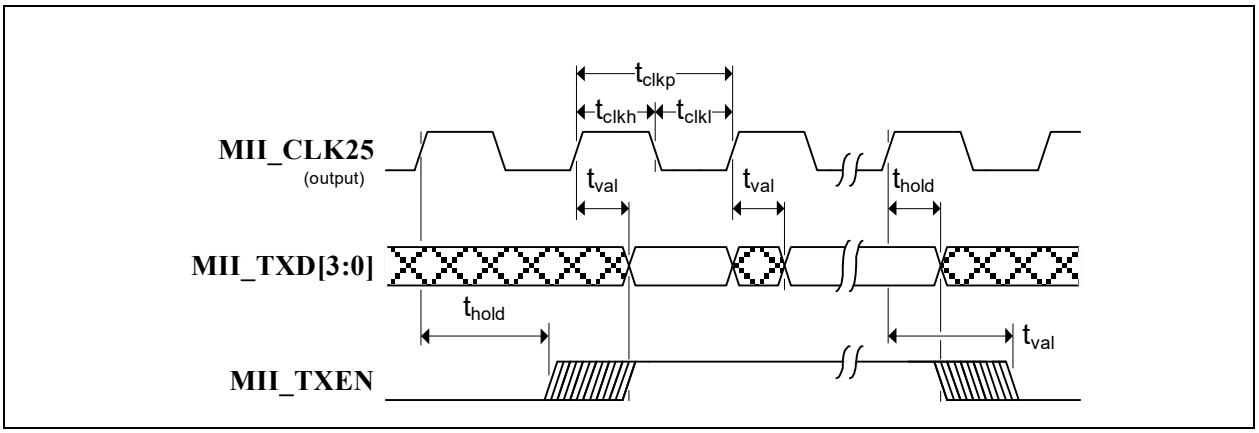


TABLE 11-5: MII TX TIMING VALUES

Symbol	Description	Min	Max	Units	Notes
t_{clkp}	**MII_CLK25** period	40	-	ns	
t_{clkh}	**MII_CLK25** high time	t_{clkp} * 0.45	t_{clkp} * 0.55	ns	
t_{clkl}	**MII_CLK25** low time	t_{clkp} * 0.45	t_{clkp} * 0.55	ns	
t_{val}	**MII_TXD[3:0]**, **MII_TXEN** output valid from rising edge of **MII_CLK25** Note 2	-	10.0	ns	Note 1
t_{hold}	**MII_TXD[3:0]**, **MII_TXEN** output hold from rising edge of **MII_CLK25** Note 2	0	-	ns	Note 1

Note 1: Timing is designed for a system load between 10 pF and 25 pF.

Note 2: Assumes TX shift value of 2, add 10 ns for each increment of TX shift (shift values of 3, 0, and 1 in order).

FIGURE 11-8: MII RX TIMING

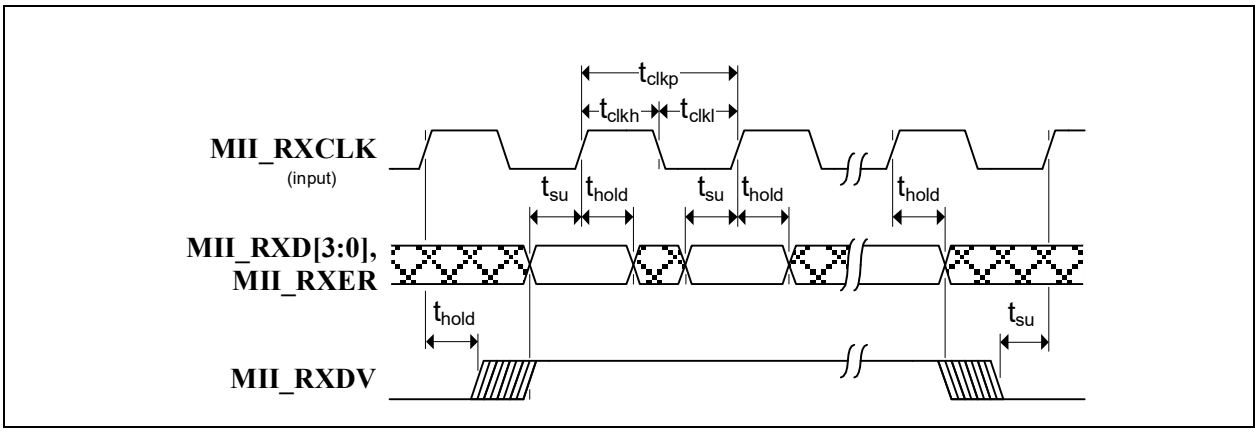


TABLE 11-6: MII RX TIMING VALUES

Symbol	Description	Min	Max	Units	Notes
t_{clkp}	**MII_RXCLK** period	40	-	ns	
t_{clkh}	**MII_RXCLK** high time	t_{clkp} * 0.4	t_{clkp} * 0.6	ns	
t_{clkl}	**MII_RXCLK** low time	t_{clkp} * 0.4	t_{clkp} * 0.6	ns	
t_{su}	**MII_RXD[3:0]**, **MII_RXER**, **MII_RXDV** setup time to rising edge of **MII_RXCLK**	5.0	-	ns	Note 3
t_{hold}	**MII_RXD[3:0]**, **MII_RXER**, **MII_RXDV** hold time after rising edge of **MII_RXCLK**	6.0	-	ns	Note 3

Note 3: Timing is designed for a system load between 10 pF and 25 pF.

11.11.7.2 Back-to-Back MII Connection Timing

With the previously listed MII TX and RX timings, back-to-back connections should use a TX shift value of 3 or 0.

11.11.7.3 Management Interface Timing

The **MII_MDIO** and **MII_MDC** timing is follows:

FIGURE 11-9: MANAGEMENT ACCESS TIMING

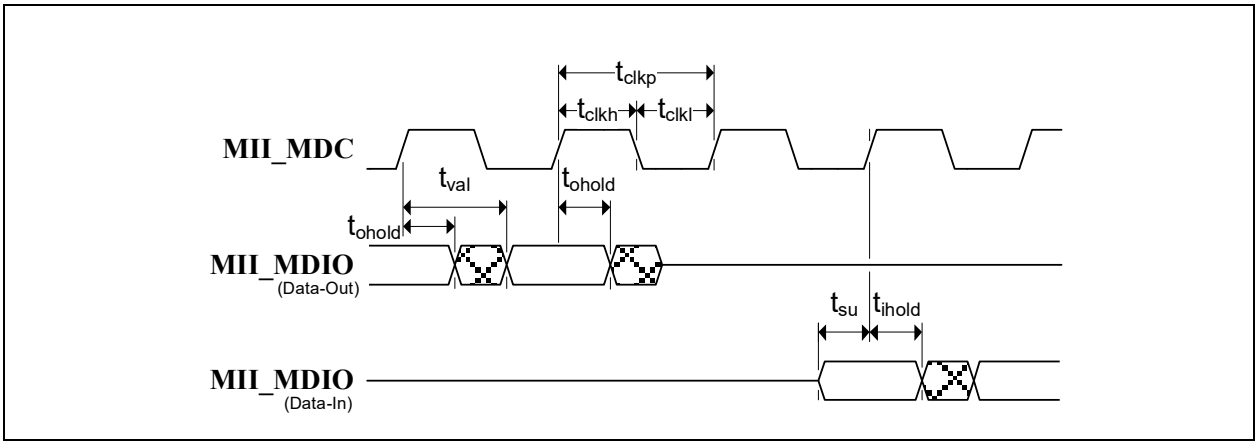


TABLE 11-7: MANAGEMENT ACCESS TIMING VALUES

Symbol	Description	Min	Max	Units	Notes
t_{clkp}	**MII_MDC** period	400	-	ns	
t_{clkh}	**MII_MDC** high time	180 (90%)	-	ns	
t_{clkl}	**MII_MDC** low time	180 (90%)	-	ns	
t_{val}	**MII_MDIO** output valid from rising edge of **MII_MDC**	-	250	ns	
t_{ohold}	**MII_MDIO** output hold from rising edge of **MII_MDC**	150	-	ns	
t_{su}	**MII_MDIO** input setup time to rising edge of **MII_MDC**	70	-	ns	
t_{ihold}	**MII_MDIO** input hold time after rising edge of **MII_MDC**	0	-	ns	

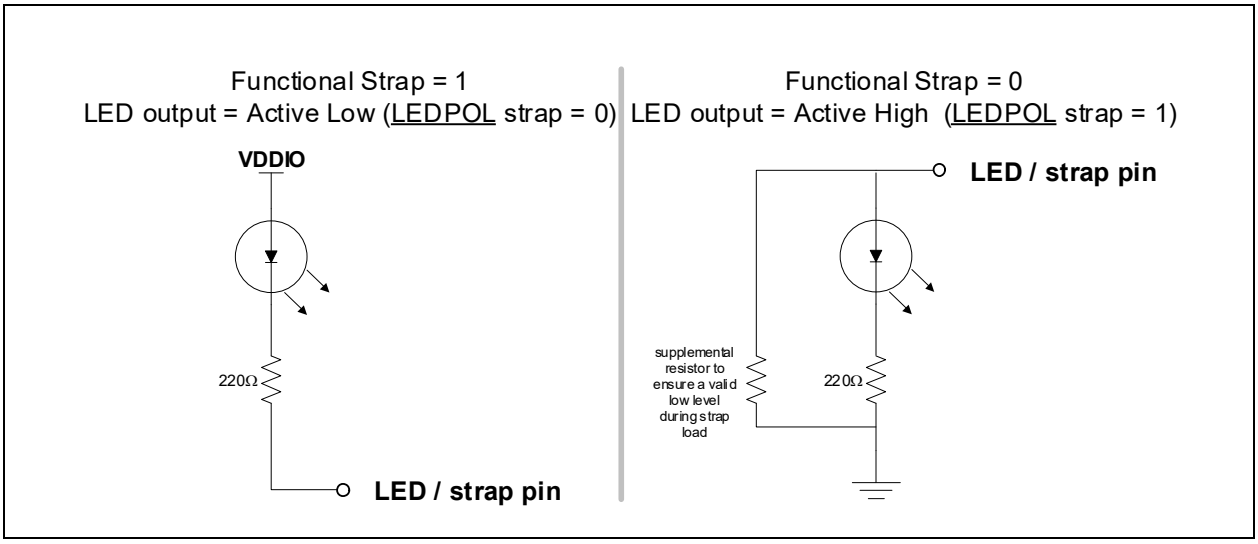
11.12 LEDs

The device includes a RUN / STATE_RUN LED (**RUNLED/STATE_RUNLED**), an error LED (**ERRLED**), and a link / activity LED per port (**LINKACTLED[0:2]**). The LED pin polarity is determined at the chip level based on the LED Polarity field in the Hardware Configuration Register (HW_CFG). The outputs are open drain or open source. The **ERRLED** also has an enable bit (ERRLED Enable) in the ASIC Configuration Register. The defaults of the LED Polarity field are controlled by the **LEDPOL[4:0]** configuration straps.

Note:	The LED pins for Port 0 and Port 2 are not swapped based on the chip mode.

When an LED pin is used as a function mode strap, the default LEF pin polarity is automatically selected based on the inverse of the strap value. An LED, via a resistor, is effectively used as a pull-up or pull-down, as detailed in Figure 11-10.

FIGURE 11-10: STRAP ON LED WITH POLARITY



The EtherCAT Core configuration provides for direct control of the RUN LED via the RUN LED Override Register, and of the ERR LED via the ERR LED Override Register.

All LED outputs may be disabled (un-driven) by setting the LED Disable (LED_DIS) bit in the Power Management Control Register (PMT_CTRL).

11.12.1 ERR LED PCB BACKWARDS COMPATIBILITY

The **ERRLED** pin on the LAN9252 functioned as the fiber mode signal detect as well as the Port A FX-SD Enable. For copper twisted pair operation, this pin would either be tied to or pulled down to ground.

Note:	Do not enable the **ERRLED** pin on a PCB design which has the pin tied to ground.

11.12.2 EEPROM ERROR AND ERR LED

Since the **ERRLED** pin is enabled via an EEPROM bit, a special situation arises in the event of an EEPROM loading error. In this case, the **ERRLED** pin is forced enabled but only if the polarity (bit 4 of the LED Polarity field in the Hardware Configuration Register (HW_CFG)) indicates the **ERRLED** is active low (implying a high strap was loaded).

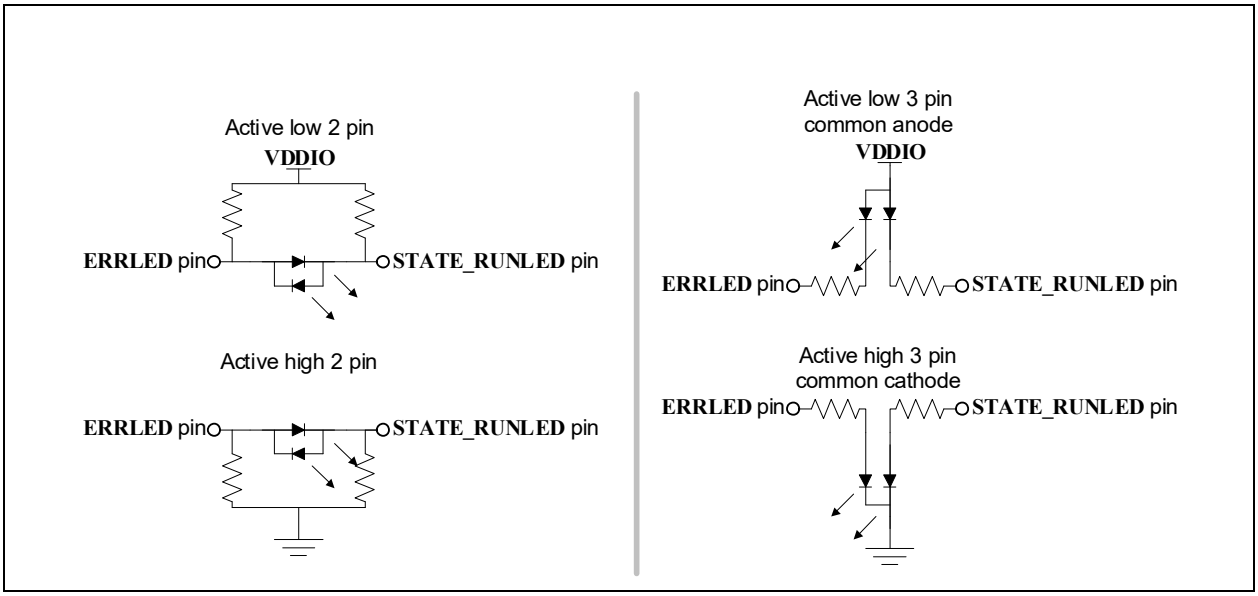
Note:	If the **ERRLED** pin is meant to be used as the PME pin, then this EEPROM loading error condition will likely trigger false PME events.

11.12.3 BI-COLOR STATE LED

A STATE LED is bi-color LED combining RUN and ERR LEDs. Since the RUN part of the STATE LED must be turned off while the ERR part is active, the **RUNLED** and **ERRLED** pins cannot be simply combined to drive the bi-color LED. Instead the **RUNLED** pin can be changed to a **STATE_RUNLED** pin, which is turned off while the **ERRLED** pin is on. **STATE_RUNLED** is selected via the STATE_RUNLED Mode Select bit of the ASIC Configuration Register. The ASIC Configuration Register is initialized from the contents of the EEPROM.

Two, three and four pin bi-color LEDs are supported. A two pin bi-color LED is supported by placing a pull-up or pull-down (depending on the signal polarity in use) on each side of the LED. A three pin bi-color LED is supported by placing a series resistor to power or to ground (depending on the signal polarity in use). The choice between common anode or common cathode also depends on the signal polarity in use. For both two and three pin options, the polarity of STATE_RUN and ERR LEDs must be the same (or the use of an inverting transistor is required). A four pin bi-color LED offers the most flexibility as each signal can have a different polarity if desired.

FIGURE 11-11: BI-COLOR STATE LED



11.13 EtherCAT CSR and Process Data RAM Access

Two methods exist for access to the EtherCAT CSR and Process Data RAM:

- EtherCAT Indirect Access Mode
- EtherCAT Direct Mapped Mode

Indirect Access Mode uses a command and data register structure that is backwards compatible with the LAN9252 and offers faster bus cycle times at the cost of more overhead. EtherCAT Direct Mapped mode reduces overhead by mapping the EtherCAT CSR and Process Data RAM into the host memory space at the cost of slower bus access and timing requirements.

The EtherCAT mode is configured via the Process Data Interface (PDI_SELECT) field in the PDI Control Register. Refer to Section 11.3, PDI Selection and Configuration for additional information.

11.13.1 ETHERCAT INDIRECT ACCESS MODE

The EtherCAT CSRs provide register level access to the various parameters of the EtherCAT Core. EtherCAT related registers can be classified into two main categories based upon their method of access: direct and indirect.

The directly accessible EtherCAT registers are part of the main system CSRs and are detailed in Section 11.15, "EtherCAT CSR and Process Data RAM Access Registers (Directly Addressable)," on page 305. These registers provide data/command registers (for access to the indirect EtherCAT Core registers).

The indirectly accessible EtherCAT Core registers reside within the EtherCAT Core and must be accessed indirectly via the EtherCAT CSR Interface Data Register (ECAT_CSR_DATA) and the EtherCAT CSR Interface Command Register (ECAT_CSR_CMD). The indirectly accessible EtherCAT Core CSRs provide full access to the many configurable parameters of the EtherCAT Core. The indirectly accessible EtherCAT Core CSRs are accessed at address 0h through 0FFFh and are detailed in Section 11.16, "EtherCAT Core CSR Registers (Indirectly Addressable)," on page 314.

The EtherCAT Core Process Data RAM can be accessed indirectly via the EtherCAT CSR Interface Data Register (ECAT_CSR_DATA) and the EtherCAT CSR Interface Command Register (ECAT_CSR_CMD), starting at 1000h. The EtherCAT Core Process Data RAM can also be accessed more efficiently using the EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA) and the EtherCAT Process RAM Write Data FIFO (ECAT_PRAM_WR_DATA). This method provides for multiple DWORDS to be transferred via a FIFO mechanism using a single command and fewer status reads.

11.13.1.1 EtherCAT CSR Reads

To perform a read of an individual EtherCAT Core register, the read cycle must be initiated by performing a single write to the EtherCAT CSR Interface Command Register (ECAT_CSR_CMD) with the CSR Busy (CSR_BUSY) bit set, the CSR Address (CSR_ADDR) field set to the desired register address, the Read/Write (R_nW) bit set and the CSR Size (CSR_SIZE) field set to the desired size.

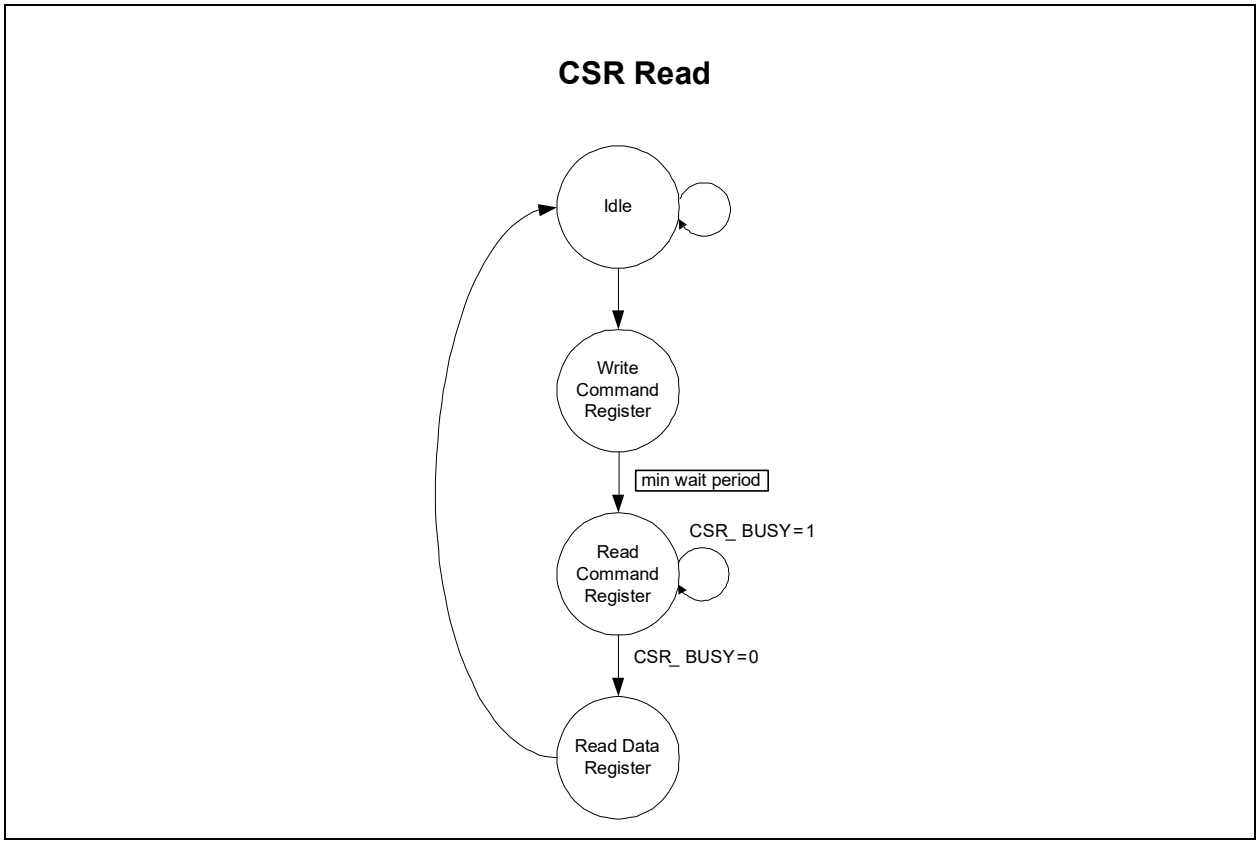
Valid data is available for reading when the CSR Busy (CSR_BUSY) bit is cleared, indicating that the data can be read from the EtherCAT CSR Interface Data Register (ECAT_CSR_DATA).

Valid data is always aligned into the lowest bits of the EtherCAT CSR Interface Data Register (ECAT_CSR_DATA).

Note:	All bytes of the EtherCAT CSR Interface Data Register (ECAT_CSR_DATA) are updated regardless of the value of CSR Size (CSR_SIZE).

Figure 11-12 illustrates the process required to perform a EtherCAT Core CSR read. The minimum wait periods as specified in Table 5-2, “Read After Write Timing Rules,” on page 43 are required where noted.

FIGURE 11-12: ETHERCAT CSR READ ACCESS FLOW DIAGRAM



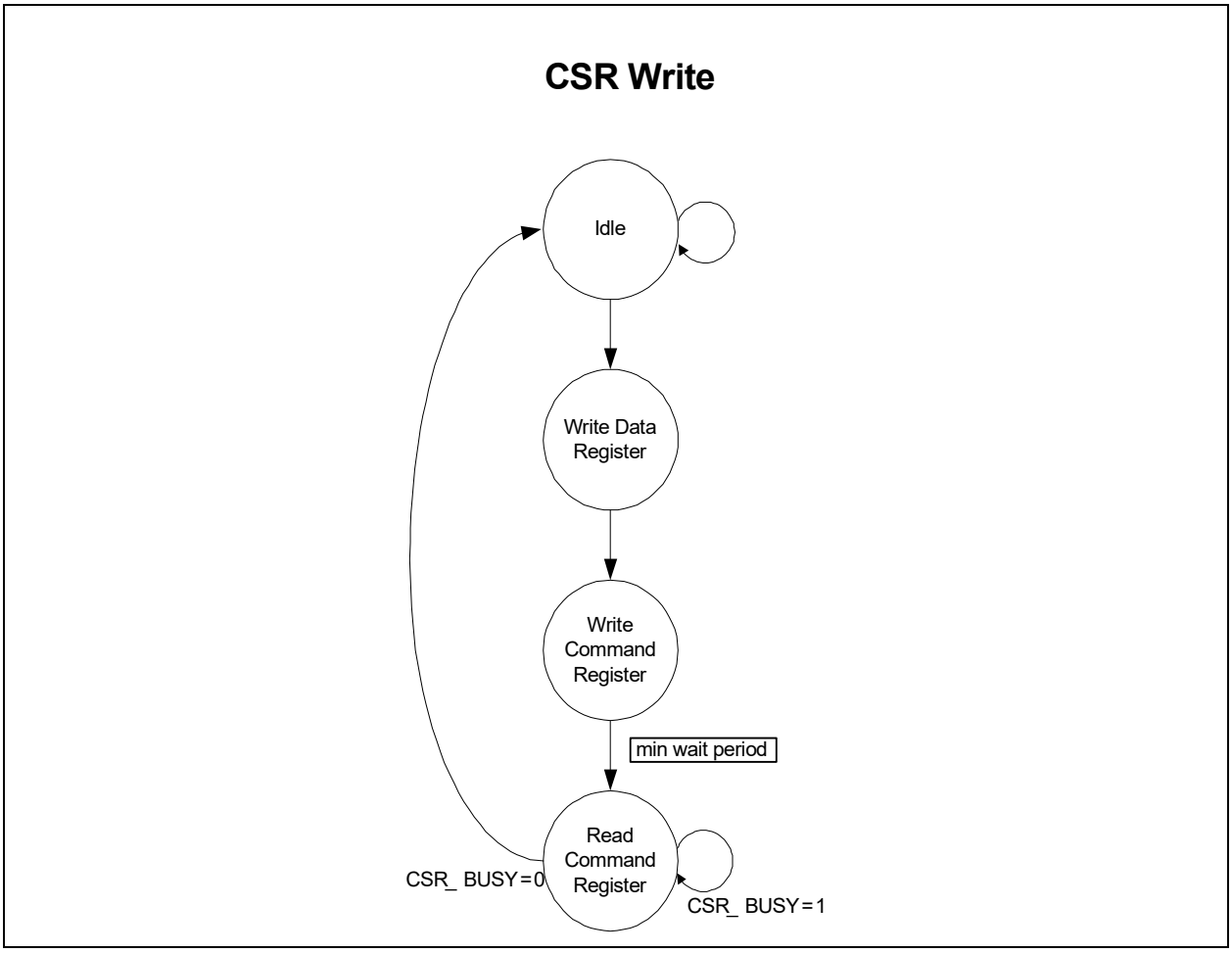
11.13.1.2 EtherCAT CSR Writes

To perform a write to an individual EtherCAT Core register, the desired data must first be written into the EtherCAT CSR Interface Data Register (ECAT_CSR_DATA). Valid data is always aligned into the lowest bits of the EtherCAT CSR Interface Data Register (ECAT_CSR_DATA).

The write cycle is initiated by performing a single write to the EtherCAT CSR Interface Command Register (ECAT_CSR_CMD) with the CSR Busy (CSR_BUSY) bit set, the CSR Address (CSR_ADDR) field set to the desired register address, the Read/Write (R_nW) bit cleared and the CSR Size (CSR_SIZE) field set to the desired size. The completion of the write cycle is indicated by the clearing of the CSR Busy (CSR_BUSY) bit.

Figure 11-13 illustrates the process required to perform a EtherCAT Core CSR write. The minimum wait periods as specified in Table 5-2, “Read After Write Timing Rules,” on page 43 are required where noted.

FIGURE 11-13: ETHERCAT CSR WRITE ACCESS FLOW DIAGRAM



11.13.1.3 EtherCAT Process RAM Reads

Process data is transferred from the EtherCAT Core through a 16 deep 32-bit wide FIFO. The FIFO has the base address of 00h, however, it is also accessible at seven additional contiguous memory locations. The Host may access the FIFO at any of these alias port locations, as they all function identically and contain the same data. This alias port addressing is implemented to allow hosts to burst through sequential addresses.

For HBI access, the Process RAM Read Data FIFO may also be accessed using FIFO Direct Selection mode. In this mode, the address input is ignored and all read accesses are directed to the Process RAM Read Data FIFO. See Section 8.3.4.1, "FIFO Direct Select Access (Multiplexed Address / Data Mode Only)," on page 75.

To perform a read of the EtherCAT Process RAM, the read cycle is initiated by first writing the EtherCAT Process RAM Read Address and Length Register (ECAT_PRAM_RD_ADDR_LEN) with the starting byte address and length (in bytes) of the desired transfer followed by a write to the EtherCAT Process RAM Read Command Register (ECAT_PRAM_RD_CMD) with the PRAM Read Busy (PRAM_READ_BUSY) bit set.

Note:	The starting byte address and length must be programmed with valid values such that all transfers are within the bounds of the Process RAM address range of 1000h to 1FFFh.

Valid data, as indicated by the PRAM Read Data Available (PRAM_READ_AVAIL) bit in the EtherCAT Process RAM Read Command Register (ECAT_PRAM_RD_CMD) is read from the FIFO through the EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA). The PRAM Read Data Available Count (PRAM_READ_AVAIL_CNT) field indicates how many reads can be performed without needing to check the status again. Following the final read of the EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA), the PRAM Read Busy (PRAM_READ_BUSY) self-clears.

Note:	The final read of the EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA) implies that all four bytes have been read, even if not all bytes are required.

As the data is transferred from the EtherCAT Core into the FIFO the PRAM Read Length (PRAM_READ_LEN) and PRAM Read Address (PRAM_READ_ADDR) are updated to show the progress.

Based on the starting address, the valid bytes in the first FIFO read are as follows:

TABLE 11-8: ETHERCAT PROCESS RAM VALID FIRST READ BYTES

Starting Address[1:0]	
00b	bytes 3, 2, 1 and 0
01b	bytes 3, 2 and 1
10b	bytes 3 and 2
11b	byte 3

Based on the starting address and length, the valid bytes in the last FIFO read are as follows:

TABLE 11-9: ETHERCAT PROCESS RAM VALID LAST READ BYTES

	Starting Length[1:0]			
Starting Address[1:0]	**01b (e.g. 5, 9, etc.)**	**10b (e.g. 6, 10, etc.)**	**11b (e.g. 7, 11, etc.)**	**00b (e.g. 8, 12, etc.)**
00b	byte 0	bytes 1 and 0	bytes 2, 1 and 0	bytes 3, 2, 1 and 0
01b	bytes 1 and 0	bytes 2, 1 and 0	bytes 3, 2, 1 and 0	byte 0
10b	bytes 2, 1 and 0	bytes 3, 2, 1 and 0	byte 0	bytes 1 and 0
11b	bytes 3, 2, 1 and 0	byte 0	bytes 1 and 0	bytes 2, 1 and 0

If the initial length is 4 bytes of less and all bytes fit into one read, the valid bytes in the only FIFO read are as follows:

TABLE 11-10: ETHERCAT PROCESS RAM VALID BYTES ONE READ

	Starting Length			
Starting Address[1:0]	**4**	**1**	**2**	**3**
00b	bytes 3, 2, 1 and 0	byte 0	bytes 1 and 0	bytes 2, 1 and 0
01b	na	byte 1	bytes 2 and 1	bytes 3, 2 and 1
10b	na	byte 2	bytes 3 and 2	na
11b	na	byte 3	na	na

Aborting a Read

If necessary, a read command can be aborted by setting the PRAM Read Abort (PRAM_READ_ABORT) bit in the EtherCAT Process RAM Read Command Register (ECAT_PRAM_RD_CMD).

11.13.1.4 EtherCAT Process RAM Writes

Process data is transferred to the EtherCAT Core through a 16 deep 32-bit wide FIFO. The FIFO has the base address of 20h, however, it is also accessible at seven additional contiguous memory locations. The Host may access the FIFO at any of these alias port locations, as they all function identically and contain the same data. This alias port addressing is implemented to allow hosts to burst through sequential addresses.

For HBI access, the Process RAM Write Data FIFO may also be accessed using FIFO Direct Selection mode. In this mode, the address input is ignored and all write accesses are directed to the Process RAM Write Data FIFO. See Section 8.3.4.1, "FIFO Direct Select Access (Multiplexed Address / Data Mode Only)," on page 75.

To perform a write to the EtherCAT Process RAM, the write cycle is initiated by first writing the EtherCAT Process RAM Write Address and Length Register (ECAT_PRAM_WR_ADDR_LEN) with the starting byte address and length (in bytes) of the desired transfer followed by a write to the EtherCAT Process RAM Write Command Register (ECAT_PRAM_WR_CMD) with the PRAM Write Busy (PRAM_WRITE_BUSY) bit set.

.

Note:	The starting byte address and length must be programmed with valid values such that all transfers are within the bounds of the Process RAM address range of 1000h to 1FFFh.

Data is transferred into the EtherCAT Core through a 16 deep 32-bit wide FIFO. The host may write data to the FIFO through the EtherCAT Process RAM Write Data FIFO (ECAT_PRAM_WR_DATA) when space is available as indicated by the PRAM Write Space Available (PRAM_WRITE_AVAIL) bit in the EtherCAT Process RAM Write Command Register (ECAT_PRAM_WR_CMD). The PRAM Write Space Available Count (PRAM_WRITE_AVAIL_CNT) field indicates how many writes can be performed without needing to check the status again. Following the final write of the data into the EtherCAT Core, the PRAM Write Busy (PRAM_WRITE_BUSY) self-clears.

Note:	The final write of the EtherCAT Process RAM Write Data FIFO (ECAT_PRAM_WR_DATA) implies that all four bytes have been written, even if not all bytes are required.

As the data is transferred to the EtherCAT Core from the FIFO the PRAM Write Length (PRAM_WRITE_LEN) and PRAM Write Address (PRAM_WRITE_ADDR) are updated to show the progress.

Based on the starting address, the valid bytes in the first FIFO write are as follows:

TABLE 11-11: ETHERCAT PROCESS RAM VALID FIRST WRITE BYTES

Starting Address[1:0]	
00b	bytes 3, 2, 1 and 0
01b	bytes 3, 2 and 1
10b	bytes 3 and 2
11b	byte 3

Based on the starting address and length, the valid bytes in the last FIFO write are as follows:

TABLE 11-12: ETHERCAT PROCESS RAM VALID LAST WRITE BYTES

	Starting Length[1:0]			
Starting address[1:0]	**01b (e.g. 5, 9, etc.)**	**10b (e.g. 6, 10, etc.)**	**11b (e.g. 7, 11, etc.)**	**00b (e.g. 8, 12, etc.)**
00b	byte 0	bytes 1 and 0	bytes 2, 1 and 0	bytes 3, 2, 1 and 0
01b	bytes 1 and 0	bytes 2, 1 and 0	bytes 3, 2, 1 and 0	byte 0
10b	bytes 2, 1 and 0	bytes 3, 2, 1 and 0	byte 0	bytes 1 and 0
11b	bytes 3, 2, 1 and 0	byte 0	bytes 1 and 0	bytes 2, 1 and 0

If the initial length is 4 bytes of less and all bytes fit into one write, the valid bytes in the only FIFO write are as follows:

TABLE 11-13: ETHERCAT PROCESS RAM VALID BYTES ONE WRITE

	Starting Length			
starting address[1:0]	**4**	**1**	**2**	**3**
00b	bytes 3, 2, 1 and 0	byte 0	bytes 1 and 0	bytes 2, 1 and 0
01b	na	byte 1	bytes 2 and 1	bytes 3, 2 and 1
10b	na	byte 2	bytes 3 and 2	na
11b	na	byte 3	na	na

Aborting a Write

If necessary, a write command can be aborted by setting the PRAM Write Abort (PRAM_WRITE_ABORT) bit in the EtherCAT Process RAM Write Command Register (ECAT_PRAM_WR_CMD).

11.13.2 ETHERCAT DIRECT MAPPED MODE

EtherCAT Direct Mapped mode places the Core CSR and Process Data RAM into host memory space. Core CSRs are mapped starting at byte address 0 through byte address FFFh and follow the address mapping of the EtherCAT core. The Process Data RAM is mapped starting at byte address 1000h through byte address 2FFFh.

EtherCAT Direct Mapped mode is enabled via the Process Data Interface (PDI_SELECT) field in the PDI Control Register. Refer to Section 11.3, PDI Selection and Configuration for additional information.

11.13.2.1 HBI Interface

During EtherCAT Direct Mapped mode, host bus access to the EtherCAT Process RAM and CSR Access Registers is disabled. The host data bus can be either 8 bit or 16 bit wide, with dynamic BYTE or WORD accesses supported in 16-bit mode. The bus width is selected by the Process Data Interface (PDI_SELECT) field in the PDI Control Register. Refer to Section 11.3, PDI Selection and Configuration for additional information.

The Asynchronous 8/16 bit microcontroller Interface PDI byte ordering is always Little Endian. Device level endianess is controlled in the HBI module. At the Asynchronous 8/16 bit microcontroller interface PDI level, access types are as follows:

TABLE 11-14: 8-BIT ASYNCHRONOUS MICROCONTROLLER INTERFACE PDI ACCESSES

A[0]	Access	DATA[7:0]
0	8 bit access to A[15:0] (low byte, even address)	low byte
1	8 bit access to A[15:0] (high byte, odd address)	high byte

TABLE 11-15: 16-BIT ASYNCHRONOUS MICROCONTROLLER INTERFACE PDI ACCESSES

A[0]	nBHE	Access	DATA[15:8]	DATA[7:0]
0	0	16 bit access to A[15:0] and A[15:0]+1 (low and high bytes)	high byte	low byte
0	1	8 bit access to A[15:0] (low byte, even address)	-	low byte
1	0	8 bit access to A[15:0] (high byte, odd address)	high byte	-
1	1	invalid access	-	-

Write Access

Write access is described in Section 10.6.3.2.

The Extended PDI Configuration Register controls the write cycle configuration for non-posted or posted operation.

Read Access

Read access is described in Section 10.6.3.3.

The Extended PDI Configuration Register controls the read cycle WAIT_ACK delay timing.

Controller Access Errors

The following access errors are detected by the microcontroller interface:

- Read or Write access in 16 bit mode with A[0]=1 and nBHE=1, i.e. an access to an odd address without Byte High Enable.
- Deassertion of WR (or deassertion of CS at the device level while WR remains asserted) while BUSY is high.
- Deassertion of RD (or deassertion of CS at the device level while RD remains asserted) while BUSY is high (read has not finished).

Access errors will have these consequences:

- The PDI Error Counter Register will be incremented.
- For A[0]=1 and nBHE (act. low) = 1 accesses, no access will be performed internally.
- Deassertion of WR (or CS) while the BUSY is low might corrupt the current and the preceding transfer (if it is not completed internally). Registers might accept write data and special functions (e.g., SyncManager buffer switching) might be performed.

- If RD (or CS) is deasserted while the BUSY is low (read has not finished), the access will be terminated internally. Although, internal byte transfers might be completed, so special functions (e.g., SyncManager buffer switching) might be performed.

The reason of the access error can be read in the PDI Error Code Register.

11.13.2.2 SPI Interface

During EtherCAT Direct Mapped mode, SPI bus access to the EtherCAT Process RAM Registers is disabled. The SPI bus interfaces directly to the OPB PDI.

Read Access

The SPI interface provides the address, byte enables and a read request. The OPB PDI returns an acknowledge and data. The SPI interface retrieves the data from the correct BYTE lane.

Write Access

The SPI interface provides the address, byte enables, data and a write request. The OPB PDI returns an acknowledge. The SPI interface provides the data onto the correct BYTE lane(s).

Controller Access Errors

Access errors are detected by the SPI interface as described in Section 13.7. The error count can be read from the PDI Error Counter Register.The reason of the access error can be read in the PDI Error Code Register.

11.14 EtherCAT Reset

After writing 0x52 (R), 0x45 (E) and 0x53 (S) into the ESC Reset ECAT Register with 3 consecutive frames or after writing 0x52 (R), 0x45 (E) and 0x53 (S) into the ESC Reset PDI Register with 3 consecutive writes, a device reset (and optional system reset) will occur, as defined in Section 6.2.1.3, "EtherCAT System Reset," on page 48.

Note:	It is likely that the last frame of the sequence will not return to the master (depending on the topology), because the links to and from the slave which is reset will go down.

11.15 EtherCAT CSR and Process Data RAM Access Registers (Directly Addressable)

This section details the directly addressable System CSRs, outside of the EtherCAT Core, which are related to the EtherCAT Core.

The EtherCAT CSR Interface Command Register (ECAT_CSR_CMD) and the EtherCAT CSR Interface Data Register (ECAT_CSR_DATA) serve as an interface to the full range of otherwise inaccessible EtherCAT Core control and status registers. For information on how to access EtherCAT registers, refer to Section 11.13, "EtherCAT CSR and Process Data RAM Access," on page 297. The EtherCAT Core registers are detailed in Section 11.16, "EtherCAT Core CSR Registers (Indirectly Addressable)," on page 314.

Note:	These registers (including the EtherCAT Process RAM Read and Write Data FIFO) are not used and are not accessible during EtherCAT Direct Mapped Mode.

TABLE 11-16: ETHERCAT PROCESS RAM AND CSR ACCESS REGISTERS

Address	EtherCAT Direct Mapped Mode	Register Name (Symbol)
000h-01Ch	N/A (unused)	EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA)
020h-03Ch	N/A (unused)	EtherCAT Process RAM Write Data FIFO (ECAT_PRAM_WR_DATA)
300h	N/A (unused)	EtherCAT CSR Interface Data Register (ECAT_CSR_DATA)
304h	N/A (unused)	EtherCAT CSR Interface Command Register (ECAT_CSR_CMD)
308h	N/A (unused)	EtherCAT Process RAM Read Address and Length Register (ECAT_PRAM_RD_ADDR_LEN)
30Ch	N/A (unused)	EtherCAT Process RAM Read Command Register (ECAT_PRAM_RD_CMD)
310h	N/A (unused)	EtherCAT Process RAM Write Address and Length Register (ECAT_PRAM_WR_ADDR_LEN)
314h	N/A (unused)	EtherCAT Process RAM Write Command Register (ECAT_PRAM_WR_CMD)

11.15.1 ETHERCAT PROCESS RAM READ DATA FIFO (ECAT_PRAM_RD_DATA)

Offset: 000h-01Ch Size: 32 bits

This read only register is used in conjunction with the EtherCAT Process RAM Read Command Register (ECAT_PRAM_RD_CMD) and the EtherCAT Process RAM Read Address and Length Register (ECAT_PRAM_RD_ADDR_LEN) to perform read operations of the EtherCAT Core Process RAM.

Data read from this register is only valid if the PRAM Read Data Available (PRAM_READ_AVAIL) bit in the EtherCAT Process RAM Read Command Register (ECAT_PRAM_RD_CMD) is a 1. The host should not read this register unless there is valid data available.

Bits	Description	Type	Default
31:0	**EtherCAT Process RAM Read Data (PRAM_RD_DATA)** This field contains the value read from the EtherCAT Core Process RAM. **Note:** Some bytes maybe invalid based on the starting address and transfer length.	RO	-

11.15.2 ETHERCAT PROCESS RAM WRITE DATA FIFO (ECAT_PRAM_WR_DATA)

Offset: 020h-03Ch Size: 32 bits

This write only register is used in conjunction with the EtherCAT Process RAM Write Command Register (ECAT_PRAM_WR_CMD) and the EtherCAT Process RAM Write Address and Length Register (ECAT_PRAM_WR_ADDR_LEN) to perform write operations to the EtherCAT Core Process RAM.

The host should not write this register unless there is available space as indicated by the PRAM Write Space Available (PRAM_WRITE_AVAIL) bit in the EtherCAT Process RAM Write Command Register (ECAT_PRAM_WR_CMD).

Bits	Description	Type	Default
31:0	**EtherCAT Process RAM Write Data (PRAM_WR_DATA)** This field contains the value written to the EtherCAT Core Process RAM. **Note:** Some bytes maybe invalid based on the starting address and transfer length.	WO	-

11.15.3 ETHERCAT CSR INTERFACE DATA REGISTER (ECAT_CSR_DATA)

Offset: 300h Size: 32 bits

This read/write register is used in conjunction with the EtherCAT CSR Interface Command Register (ECAT_CSR_CMD) to perform read and write operations with the EtherCAT Core CSRs.

Bits	Description	Type	Default
31:0	**EtherCAT CSR Data (CSR_DATA)** This field contains the value read from or written to the EtherCAT Core CSR. The EtherCAT Core CSR is selected via the CSR Address (CSR_ADDR) bits of the EtherCAT CSR Interface Command Register (ECAT_CSR_CMD). Valid data is always written to or read from the lower bits of this field. The hardware handles any required byte alignment. Upon a read, the value returned depends on the Read/Write (R_nW) bit in the EtherCAT CSR Interface Command Register (ECAT_CSR_CMD). If Read/Write (R_nW) is set, the data is from the EtherCAT Core. If Read/Write (R_nW) is cleared, the data is the value that was last written into this register.	R/W	00000000h

11.15.4 ETHERCAT CSR INTERFACE COMMAND REGISTER (ECAT_CSR_CMD)

Offset: 304h Size: 32 bits

This read/write register is used in conjunction with the EtherCAT CSR Interface Data Register (ECAT_CSR_DATA) to perform read and write operations with the EtherCAT Core CSRs.

Bits	Description	Type	Default
31	**CSR Busy (CSR_BUSY)** When a 1 is written to this bit, the read or write operation (as determined by the R_nW bit) is performed to the specified EtherCAT Core CSR in CSR Address (CSR_ADDR). This bit will remain set until the operation is complete, at which time the bit will self-clear. In the case of a read, the clearing of this bit indicates to the Host that valid data can be read from the EtherCAT CSR Interface Data Register (ECAT_CSR_DATA). Writing a 0 to this bit has no affect. The host should not modify the ETHERCAT_CSR_CMD and ETHERCAT_CSR_DATA registers unless this bit is a 0.	R/W SC	0b
30	**Read/Write (R_nW)** This bit determines whether a read or write operation is performed by the Host to the specified EtherCAT Core CSR. **0: Write** **1: Read**	R/W	0b
29:19	**RESERVED**	RO	-
18:16	**CSR Size (CSR_SIZE)** This field specifies the size of the EtherCAT Core CSR in bytes. Valid values are 1, 2 and 4. The host should not use invalid values. Note 4.	R/W	0h
15:0	**CSR Address (CSR_ADDR)** This field selects the EtherCAT Core CSR that will be accessed with a read or write operation. This is a byte address which is the format used to specify the offsets of the EtherCAT Core CSRs. Note 4.	R/W	00h

Note 4: WORD and DWORD accesses must be aligned on the proper address boundary according to the following table.

TABLE 11-17: ETHERCAT CSR ADDRESS VS. SIZE

CSR_SIZE[2:0]	CSR_ADDR[1:0]
1	00b, 01b, 10b, 11b
2	00b, 10b
4	00b

11.15.5 ETHERCAT PROCESS RAM READ ADDRESS AND LENGTH REGISTER (ECAT_PRAM_RD_ADDR_LEN)

Offset: 308h Size: 32 bits

This read/write register is used in conjunction with the EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA) and the EtherCAT Process RAM Read Command Register (ECAT_PRAM_RD_CMD) to perform read operations from the EtherCAT Core Process RAM.

Note: The starting byte address and length must be programmed with valid values such that all transfers are within the bounds of the Process RAM address range of 1000h to 2FFFh.

Bits	Description	Type	Default
31:16	**PRAM Read Length (PRAM_READ_LEN)** This field indicates the number of bytes to be read from the EtherCAT Core Process RAM. It is decremented as data is read from the EtherCAT Core and placed into the FIFO. The host should not modify this field unless the PRAM Read Busy (PRAM_READ_BUSY) bit is a low.	R/W	0000h
15:0	**PRAM Read Address (PRAM_READ_ADDR)** This field indicates the EtherCAT Core byte address to be read. It is incremented as data is read from the EtherCAT Core and placed into the FIFO. **Note:** The Process RAM starts at address 1000h. The host should not modify this field unless the PRAM Read Busy (PRAM_READ_BUSY) bit is a 0.	R/W	0000h

11.15.6 ETHERCAT PROCESS RAM READ COMMAND REGISTER (ECAT_PRAM_RD_CMD)

Offset: 30Ch Size: 32 bits

This read/write register is used in conjunction with the EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA) and the EtherCAT Process RAM Read Address and Length Register (ECAT_PRAM_RD_ADDR_LEN) to perform read operations from the EtherCAT Core Process RAM.

Bits	Description	Type	Default
31	**PRAM Read Busy (PRAM_READ_BUSY)** When a 1 is written to this bit, the read operation is started beginning at the EtherCAT Core Process RAM location specified in PRAM Read Address (PRAM_READ_ADDR) for the length specified in PRAM Read Length (PRAM_READ_LEN). This bit will remain set until the entire read operation is complete, at which time the bit will self-clear. Writing a 0 to this bit has no affect.	R/W SC	0b
30	**PRAM Read Abort (PRAM_READ_ABORT)** Writing a 1 to this bit will cause the read operation in process to be canceled. The PRAM Read Busy (PRAM_READ_BUSY) will be cleared and the Read Data FIFO, along with the status bits, will be reset. This bit will self-clear. Writing a 0 to this bit has no affect.	R/W SC	0b
29:13	**RESERVED**	RO	-
12:8	**PRAM Read Data Available Count (PRAM_READ_AVAIL_CNT)** This field indicates the number of times that the EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA) can be read without further need to check the status. This field increments as data is read from the EtherCAT Core and placed into the FIFO. This field is decremented when the a entire DWORD of data is read from the EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA).	RO	00000b
7:1	**RESERVED**	RO	-
0	**PRAM Read Data Available (PRAM_READ_AVAIL)** This field indicates that the EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA) has valid data to be read.	RO	0b

11.15.7 ETHERCAT PROCESS RAM WRITE ADDRESS AND LENGTH REGISTER (ECAT_PRAM_WR_ADDR_LEN)

Offset: 310h Size: 32 bits

This read/write register is used in conjunction with the EtherCAT Process RAM Write Data FIFO (ECAT_PRAM_WR_DATA) and the EtherCAT Process RAM Write Command Register (ECAT_PRAM_WR_CMD) to perform write operations to the EtherCAT Core Process RAM.

Note: The starting byte address and length must be programmed with valid values such that all transfers are within the bounds of the Process RAM address range of 1000h to 2FFFh.

Bits	Description	Type	Default
31:16	**PRAM Write Length (PRAM_WRITE_LEN)** This field indicates the number of bytes to be written to the EtherCAT Core Process RAM. It is decremented as data is written to the EtherCAT Core from the FIFO. The host should not modify this field unless the PRAM Write Busy (PRAM_WRITE_BUSY) bit is a low.	R/W	0000h
15:0	**PRAM Write Address (PRAM_WRITE_ADDR)** This field indicates the EtherCAT Core byte address to be written. It is incremented as data is written to the EtherCAT Core from the FIFO. **Note:** The Process RAM starts at address 1000h. The host should not modify this field unless the PRAM Write Busy (PRAM_WRITE_BUSY) bit is a 0.	R/W	0000h

11.15.8 ETHERCAT PROCESS RAM WRITE COMMAND REGISTER (ECAT_PRAM_WR_CMD)

Offset: 314h Size: 32 bits

This read/write register is used in conjunction with the EtherCAT Process RAM Write Data FIFO (ECAT_PRAM_WR_DATA) and the EtherCAT Process RAM Write Address and Length Register (ECAT_PRAM_WR_ADDR_LEN) to perform write operations to the EtherCAT Core Process RAM.

Bits	Description	Type	Default
31	**PRAM Write Busy (PRAM_WRITE_BUSY)** When a 1 is written to this bit, the write operation is started beginning at the EtherCAT Core Process RAM location specified in PRAM Write Address (PRAM_WRITE_ADDR) for the length specified in PRAM Write Length (PRAM_WRITE_LEN). This bit will remain set until the entire write operation is complete, at which time the bit will self-clear. Writing a 0 to this bit has no affect.	R/W SC	0b
30	**PRAM Write Abort (PRAM_WRITE_ABORT)** Writing a 1 to this bit will cause the write operation in process to be canceled. The PRAM Write Busy (PRAM_WRITE_BUSY) will be cleared and the Write Data FIFO, along with the status bits, will be reset. This bit will self-clear. Writing a 0 to this bit has no affect.	R/W SC	0b
29:13	**RESERVED**	RO	-
12:8	**PRAM Write Space Available Count (PRAM_WRITE_AVAIL_CNT)** This field indicates the number of times that the EtherCAT Process RAM Write Data FIFO (ECAT_PRAM_WR_DATA) can be written without further need to check the status. This field is decremented when the a entire DWORD of data is written into the EtherCAT Process RAM Write Data FIFO (ECAT_PRAM_WR_DATA). This field increments as data is read from the FIFO and placed into the EtherCAT Core.	RO	10000b
7:1	**RESERVED**	RO	-
0	**PRAM Write Space Available (PRAM_WRITE_AVAIL)** This field indicates that the EtherCAT Process RAM Write Data FIFO (ECAT_PRAM_WR_DATA) has available space for data to be written.	RO	1b

11.16 EtherCAT Core CSR Registers (Indirectly Addressable)

This section details the indirectly addressable EtherCAT Core CSRs, which are accessed via the directly addressable EtherCAT CSR Interface Data Register (ECAT_CSR_DATA) and EtherCAT CSR Interface Command Register (ECAT_CSR_CMD). For information on how to access EtherCAT registers, refer to Section 11.13, "EtherCAT CSR and Process Data RAM Access," on page 297. The directly addressable EtherCAT registers are detailed in Section 11.15, "EtherCAT CSR and Process Data RAM Access Registers (Directly Addressable)," on page 305.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

The read/write behavior of specific EtherCAT Core register bits may differ depending on how the register is accessed. Each EtherCAT Core register includes “ECAT Type” and “PDI Type” columns, which provide the bit/field type for register accesses via an EtherCAT Master Node or Process Data Interface (SPI / Host Bus), respectively.

TABLE 11-18: ETHERCAT CORE CSR REGISTERS

Address	Register Name (Symbol)
ESC Information	
0000h	Type Register
0001h	Revision Register
0002h-0003h	Build Register
0004h	FMMUs Supported Register
0005h	SyncManagers Supported Register
0006h	RAM Size Register
0007h	Port Descriptor Register
0008h-0009h	ESC Features Supported Register
Station Address	
0010h-0011h	Configured Station Register
0012h-0013h	Configured Station Alias Register
Write Protection	
0020h	Write Register Enable Register
0021h	Write Register Protection Register
0030h	ESC Write Register Enable Register
0031h	ESC Write Register Protection Register
Data Link Layer	
0040h	ESC Reset ECAT Register
0041h	ESC Reset PDI Register
0100h-0103h	ESC DL Control Register
0108h-0109h	Physical Read/Write Offset Register
0110h-0111h	ESC DL Status Register
Application Layer	
0120h-0121h	AL Control Register
0130h-0131h	AL Status Register
0134h-0135h	AL Status Code Register
0138h	RUN LED Override Register
0139h	ERR LED Override Register
PDI (Process Data Interface)	
0140h	PDI Control Register

TABLE 11-18: ETHERCAT CORE CSR REGISTERS (CONTINUED)

Address	Register Name (Symbol)
0141h	ESC Configuration Register
0142h-0143h	ASIC Configuration Register
0144h-0145h	RESERVED Register
0150h	PDI Configuration Register
0151h	Sync/Latch PDI Configuration Register
0152h-0153h	Extended PDI Configuration Register
Interrupts	
0200h-0201h	ECAT Event Mask Register
0204h-0207h	AL Event Mask Register
0210h-0211h	ECAT Event Request Register
0220h-0223h	AL Event Request Register
Error Counters	
0300h-0307h	RX Error Counter Registers
0308h-030Bh	Forwarded RX Error Counter Registers
030Ch	ECAT Processing Unit Error Counter Register
030Dh	PDI Error Counter Register
030Eh	PDI Error Code Register
0310h-0313h	Lost Link Counter Registers
Watchdogs	
0400h-0401h	Watchdog Divider Register
0410h-0411h	Watchdog Time PDI Register
0420h-0421h	Watchdog Time Process Data Register
0440h-0441h	Watchdog Status Process Data Register
0442h	Watchdog Counter Process Data Register
0443h	Watchdog Counter PDI Register
EEPROM Interface	
0500h	EEPROM Configuration Register
0501h	EEPROM PDI Access State Register
0502h-0503h	EEPROM Control/Status Register
0504h-0507h	EEPROM Address Register
0508h-050Fh	EEPROM Data Register
MII Management Interface	
0510h-0511h	MII Management Control/Status Register
0512h	PHY Address Register
0513h	PHY Register Address Register
0514h-0515h	PHY DATA Register
0516h	MII Management ECAT Access State Register
0517h	MII Management PDI Access State Register
0518h-051Bh	PHY Port Status Registers
0600h-067Fh	FMMU[2:0] Registers **(8x16 bytes)**
+0h-3h	FMMUx Logical Start Address Register
+4h-5h	FMMUx Length Register
+6h	FMMUx Logical Start Bit Register
+7h	FMMUx Logical Stop Bit Register

TABLE 11-18: ETHERCAT CORE CSR REGISTERS (CONTINUED)

Address	Register Name (Symbol)
+8h-9h	FMMUx Physical Start Address Register
+Ah	FMMUx Physical Start Bit Register
+Bh	FMMUx Type Register
+Ch	FMMUx Activate Register
+Dh-Fh	FMMUx Reserved Register
0680h-06FFh	Reserved
0800h-083Fh	SyncManager[3:0] Registers **(8x8 bytes)**
+0h-1h	SyncManager x Physical Start Address Register
+2h-3h	SyncManager x Length Register
+4h	SyncManager x Control Register
+5h	SyncManager x Status Register
+6h	SyncManager x Activate Register
+7h	SyncManager x PDI Control Register
0840h-087Fh	Reserved
0900h-09FFh	**Distributed Clocks (DC)**
Distributed Clocks - Receive Times	
0900h-0903h	Receive Time Port 0 Register
0904h-0907h	Receive Time Port 1 Register
0908h-090Bh	Receive Time Port 2 Register
090Ch-090Fh	Reserved
Distributed Clocks - Time Loop Control Unit	
0910h-0917h	System Time Register
0918h-091Fh	Receive Time ECAT Processing Unit Register
0920h-0927h	System Time Offset Register
0928h-092Bh	System Time Delay Register
092Ch-092Fh	System Time Difference Register
0930h-0931h	Speed Counter Start Register
0932h-0933h	Speed Counter Diff Register
0934h	System Time Difference Filter Depth Register
0935h	Speed Counter Filter Depth Register
Distributed Clocks - Cyclic Unit Control	
0980h	Cyclic Unit Control Register
Distributed Clocks - SYNC Out Unit	
0981h	Activation Register
0982h-0983h	Pulse Length of SyncSignals Register
0984h	Activation Status Register
098Eh	SYNC0 Status Register
098Fh	SYNC1 Status Register
0990h-0997h	Start Time Cyclic Operation Register
0998h-099Fh	Next SYNC1 Pulse Register
09A0h-09A3h	SYNC0 Cycle Time Register
09A4h-09A7h	SYNC1 Cycle Time Register
Distributed Clocks - Latch In Unit	
09A8h	LATCH0 Control Register

TABLE 11-18: ETHERCAT CORE CSR REGISTERS (CONTINUED)

Address	Register Name (Symbol)
09A9h	LATCH1 Control Register
09AEh	LATCH0 Status Register
09AFh	LATCH1 Status Register
09B0h-09B7h	LATCH0 Time Positive Edge Register
09B8h-09BFh	LATCH0 Time Negative Edge Register
09C0h-09C7h	LATCH1 Time Positive Edge Register
09C8h-09CFh	LATCH1 Time Negative Edge Register
Distributed Clocks - SyncManager Event Times	
09F0h-09F3h	EtherCAT Buffer Change Event Time Register
09F8h-09FBh	PDI Buffer Start Time Event Register
09FCh-09FFh	PDI Buffer Change Event Time Register
ESC Specific	
0E00h-0E07h	Product ID Register
0E08h-0E0Fh	Vendor ID Register
Digital Input/Output	
0F00h-0F03h	Digital I/O Output Data Register
0F10h-0F11h	General Purpose Output Register
0F18h-0F19h	General Purpose Input Register
User RAM	
0F80h-0FFFh	User RAM
Process Data RAM	
1000h-1003h	Digital I/O Input Data Register
1000h-2FFFh	Process Data RAM

11.16.1 TYPE REGISTER

Offset: 0000h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:0	**EtherCAT Controller Type** C0h = Microchip.	RO	RO	C0h

11.16.2 REVISION REGISTER

Offset: 0001h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:0	**EtherCAT Controller Revision**	RO	RO	03h

11.16.3 BUILD REGISTER

Offset: 0002h-0003h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**EtherCAT Controller Build** [7:4] = minor version [3:0] = Maintenance version	RO	RO	0000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.4 FMMUS SUPPORTED REGISTER

Offset: 0004h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:0	**Supported FMMUs** This field details the number of supported FMMU channels (or entities) of the EtherCAT slave controller. The device provides 8.	RO	RO	08h

11.16.5 SYNCMANAGERS SUPPORTED REGISTER

Offset: 0005h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:0	**Supported SyncManagers** This field details the number of supported SyncManager channels (or entities) of the EtherCAT slave controller. The device provides 8.	RO	RO	08h

11.16.6 RAM SIZE REGISTER

Offset: 0006h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:0	**Process Data RAM Size** This field details the process data RAM size included in the EtherCAT slave controller. The device provides 8KB.	RO	RO	08h

11.16.7 PORT DESCRIPTOR REGISTER

Offset: 0007h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:6	**Port 3 Configuration** This field details the Port 3 configuration. **00:** Not implemented **01:** Not configured **10:** EBUS **11:** MII/RMII	RO	RO	00b
5:4	**Port 2 Configuration** This field details the Port 2 configuration. **00:** Not implemented **01:** Not configured **10:** EBUS **11:** MII/RMII	RO	RO	11b (3-port operation) 01b (2-port operation) See **Section 13.0 "Chip Mode Configuration"**
3:2	**Port 1 Configuration** This field details the Port 1 configuration. **00:** Not implemented **01:** Not configured **10:** EBUS **11:** MII/RMII	RO	RO	11b
1:0	**Port 0 Configuration** This field details the Port 0 configuration. **00:** Not implemented **01:** Not configured **10:** EBUS **11:** MII/RMII	RO	RO	11b

11.16.8 ESC FEATURES SUPPORTED REGISTER

Offset: 0008h-0009h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:12	**RESERVED**	RO	RO	0h
11	**Fixed FMMU/SyncManager Configuration** **0:** Variable configuration **1:** Fixed configuration	RO	RO	0b
10	**EtherCAT Read/Write Command Support** **0:** Supported **1:** Not supported	RO	RO	0b
9	**EtherCAT LRW Command Support** **0:** Supported **1:** Not supported	RO	RO	0b
8	**Enhanced DC SYNC Activation** **0:** Not available **1:** Available **Note:** This feature refers to the Activation Register and Activation Status Register	RO	RO	1b
7	**Separate Handling of FCS Errors** **0:** Not supported **1:** Supported, frame with wrong FCS and additional nibble will be counted separately in Forwarded RX Counter	RO	RO	1b
6	**Enhanced Link Detection MII** **0:** Not available **1:** Available	RO	RO	1b
5	**Enhanced Link Detection EBUS** **0:** Not available **1:** Available	RO	RO	0b
4	**Low Jitter EBUS** **0:** Not available, standard jitter **1:** Available, jitter minimized	RO	RO	0b
3	**Distributed Clocks (width)** **0:** 32-bit **1:** 64-bit	RO	RO	1b
2	**Distributed Clock** **0:** Not available **1:** Available	RO	RO	1b
1	**RESERVED**	RO	RO	0b
0	**FMMU Operation** **0:** Bit oriented **1:** Byte oriented	RO	RO	0b

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.9 CONFIGURED STATION REGISTER

Offset: 0010h-0011h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**Configured Station Address** This field contains the address used for node addressing (FPxx commands)	R/W	RO	0000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.10 CONFIGURED STATION ALIAS REGISTER

Offset: 0012h-0013h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**Configured Station Alias Address** This field contains the alias address used for node addressing (FPxx commands). The use of this alias is activated by the Station Alias bit of the ESC DL Control Register. **Note:** EEPROM value is only taken over at first EEPROM load after lower-on reset.	RO	R/W	0000h Note 5

Note 5: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.11 WRITE REGISTER ENABLE REGISTER

Offset: 0020h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:1	**RESERVED** Write 0.	RO	RO	0000000b
0	**Write Register Enable** If write protection is enabled, this register must be written in the same Ethernet frame (value is a don't care) before other writes to this station are allowed. Write protection is still active after this frame (if the Write Register Protection Register is not changed)	R/W	RO	0b

11.16.12 WRITE REGISTER PROTECTION REGISTER

Offset: 0021h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:1	**RESERVED** Write 0.	RO	RO	0000000b
0	**Write Register Protection** **0:** Protection disabled **1:** Protection enabled **Note:** Registers 0000h-0F7Fh are write protected, except for 0020h and 0030h.	R/W	RO	0b

11.16.13 ESC WRITE REGISTER ENABLE REGISTER

Offset: 0030h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:1	**RESERVED** Write 0.	RO	RO	0000000b
0	**ESC Write Register Enable** If ESC write protection is enabled, this register must be written in the same Ethernet frame (value is a don't care) before other writes to this station are allowed. ESC write protection is still active after this frame (if the ESC Write Register Protection Register is not changed)	R/W	RO	0b

11.16.14 ESC WRITE REGISTER PROTECTION REGISTER

Offset: 0031h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:1	**RESERVED** Write 0.	RO	RO	0000000b
0	**ESC Write Register Protection** **0:** Protection disabled **1:** Protection enabled **Note:** All areas are write protected, except for 0030h.	R/W	RO	0b

11.16.15 ESC RESET ECAT REGISTER

Offset: 0040h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
Write				
7:0	**ESC Reset ECAT** A reset is asserted after writing 52h (”R”), 45h (“E”), and 53h (“S”) in this register with 3 consecutive commands.	R/W	RO	00h
Read				
7:2	**RESERVED**	RO	RO	000000b
1:0	**Reset Procedure Progress** **01:** After writing 52h **10:** After writing 45h (if 52h previously written) **00:** Else	R/W	RO	00b

11.16.16 ESC RESET PDI REGISTER

Offset: 0041h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
Write				
7:0	**ESC Reset PDI** A reset is asserted after writing 52h (”R”), 45h (“E”), and 53h (“S”) in this register with 3 consecutive commands.	RO	R/W	00h
Read				
7:2	**RESERVED**	RO	RO	000000b
1:0	**Reset Procedure Progress** **01:** After writing 52h **10:** After writing 45h (if 52h previously written) **00:** Else	RO	R/W	00b

11.16.17 ESC DL CONTROL REGISTER

Offset: 0100h-0103h Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:25	**RESERVED** Write 0.	RO	RO	0000000b
24	**Station Alias** **0:** Ignore station alias **1:** Alias can be used for all configured address command types (FPRD, FPWR, etc.)	R/W	RO	0b
23:20	**RESERVED** Write 0.	RO	RO	0000b
19	**EBUS Low Jitter** **0:** Normal jitter **1:** Reduced jitter	R/W	RO	0b
18:16	**RX FIFO Size/RX Delay Reduction** (ESC delays start of forwarding until FIFO is at least half full) See Note 6. EBUS / MII **000:** -50 ns / -40 ns **001:** -40 ns / -40 ns **010:** -30 ns / -40 ns **011:** -20 ns / -40 ns **100:** -10 ns / No change **101:** No change / No change **110:** No change / No change **111:** Default / Default	R/W	RO	111b
15:14	**RESERVED** Write 0.	RO	RO	00b
13:12	**Loop Port 2** **00:** Auto. **01:** Auto Close. **10:** Open. **11:** Closed.	R/W Note 7	RO	00b
11:10	**Loop Port 1** **00:** Auto. **01:** Auto Close. **10:** Open. **11:** Closed.	R/W Note 7	RO	00b
9:8	**Loop Port 0** **00:** Auto. **01:** Auto Close. **10:** Open. **11:** Closed.	R/W Note 7	RO	00b
7:2	**RESERVED** Write 0.	RO	RO	000000b

Bits	Description	ECAT Type	PDI Type	Default
1	**Temporary Use of Register 0101h Settings** **0:** Permanent Use **1:** Temporarily use for ~1 s, then revert to previous settings.	R/W	RO	0b
0	**Forwarding Rule** **0:** EtherCAT frames are processed, Non-EtherCAT frames are forwarded without processing **1:** EtherCAT frame are processed, Non-EtherCAT frames are destroyed. The source MAC address is changed for every frame (SOURCE_MAC[1] is set to 1 - locally administered address) regardless of the forwarding rule.	R/W	RO	1b

Note 6: The possibility of RX FIFO Size reduction depends on the clock source accuracy of the ESC and of every connected EtherCAT/Ethernet device (master, slave, etc.). RX FIFO Size of 111b is sufficient for 100ppm accuracy, RX FIFO Size 000b is possible with 25ppm accuracy (frame size of 1518/1522 Byte).

Note 7: Loop configuration changes are delayed until the end of a currently received or transmitted frame at the port.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.18 PHYSICAL READ/WRITE OFFSET REGISTER

Offset: 0108h-0109h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**Physical Read/Write Offset** Offset of R/W commands (FPRW, APRW) between Read address and Write address. RD_ADR - ADR and WR_ADR = ADR + R/W-offset.	R/W	RO	0b

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.19 ESC DL STATUS REGISTER

Offset: 0110h-0111h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:14	**RESERVED**	RO	RO	00b
13	**Communication on Port 2** **0:** No stable communication **1:** Communication established	RO	RO	0b
12	**Loop Port 2** **0:** Open. **1:** Closed.	RO	RO	0b
11	**Communication on Port 1** **0:** No stable communication **1:** Communication established	RO	RO	0b
10	**Loop Port 1** **0:** Open. **1:** Closed.	RO	RO	0b
9	**Communication on Port 0** **0:** No stable communication **1:** Communication established	RO	RO	0b
8	**Loop Port 0** **0:** Open. **1:** Closed.	RO	RO	0b
7	**RESERVED**	RO	RO	0b
6	**Physical Link on Port 2** **0:** No link **1:** Link detected	RO	RO	0b
5	**Physical Link on Port 1** **0:** No link **1:** Link detected	RO	RO	0b
4	**Physical Link on Port 0** **0:** No link **1:** Link detected	RO	RO	0b
3	**RESERVED**	RO	RO	0b
2	**Enhanced Link Detection** **0:** Deactivated for all ports **1:** Activated for at least one port **Note:** EEPROM value is only taken over at first EEPROM load after power-on reset.	RO	RO	0b (until first EEPROM load, then EEPROM ADR 0000h bit 9 or 0000h[15:12])
1	**PDI Watchdog Status** **0:** Watchdog expired **1:** Watchdog reloaded	RO	RO	0b

Bits	Description	ECAT Type	PDI Type	Default
0	**PDI Operational/EEPROM Loaded Correctly** **0:** EEPROM not loaded, PDI not operational (no access to Process Data RAM) **1:** EEPROM loaded correctly, PDI operational (access to Process Data RAM)	RO	RO	0b

Note: Reading this register from ECAT clears the DL Status Event bit in the ECAT Event Request Register.

For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.20 AL CONTROL REGISTER

Offset: 0120h-0121h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:5	**RESERVED** Write as 0.	R/W Note 8	R/WC	000h
4	**Error Ind Ack** **0:** No Ack of Error Ind in AL status register **1:** Ack of Error Ind in AL status register	R/W Note 8	R/WC	0b
3:0	**Initiate State Transition of Device State Machine** **1h:** Request Init State **2h:** Request Pre-Operational State **3h:** Request Bootstrap State **4h:** Request Safe-Operational State **8h:** Request Operational State	R/W Note 8	R/WC	1h

Note 8: This register behaves like a mailbox if Device Emulation is off (Device Emulation bit of ESC Configuration Register is 0). The PDI must read this register after ECAT has written it. Otherwise, ECAT can not write again to this register. After rest, this register can be written by ECAT. Regarding mailbox functionality, both registers 0120h and 0121h are equivalent, e.g., reading 0121h is sufficient to make this register writable again. If Device Emulation is on, this register can always be written and it contents are copied to the AL Status Register. Reading this register from PDI clears all Event Requests (register 0220h bit 0).

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.21 AL STATUS REGISTER

Offset: 0130h-0131h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:5	**RESERVED** Write as 0.	RO	R/W Note 9	000h
4	**Error Ind** **0:** Device is in state as requested or Flag cleared by command **1:** Device has not entered requested state or changed state as a result of a local action	RO	R/W Note 9	0b
3:0	**Actual State of the Device State Machine** **1h:** Init State **2h:** Pre-Operational State **3h:** Bootstrap State **4h:** Safe-Operational State **8h:** Operational State	RO	R/W Note 9	1h

Note 9: This register is only writable if Device Emulation is off (Device Emulation bit of ESC Configuration Register is 0). Otherwise, this register will reflect the AL Control Register values. Reading this register from ECAT clears the AL Status Event bit in the ECAT Event Request Register.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.22 AL STATUS CODE REGISTER

Offset: 0134h-0135h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**AL Status Code**	RO	R/W	0000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.23 RUN LED OVERRIDE REGISTER

Offset: 0138h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:5	**RESERVED** Write 0.	R/W	R/W	000b
4	**RUN Override** **0:** Override disabled **1:** Override enabled	R/W	R/W	0b
3:0	**RUN LED Code** Code — FSM State **0h:** Off — 1 - Init **1h-Ch:** Flash 1x-12x — 4 - SafeOp 1x **Dh:** Blinking — 2 - PreOp **Eh:** Flickering — 3 - Bootstrap **Fh:** On — 8 - Op	R/W	R/W	0h

Note: Changes to AL Status Register with valid values will disable RUN Override (bit 4 = 0). The value read in this register always reflects the current LED output.

11.16.24 ERR LED OVERRIDE REGISTER

Offset: 0139h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:5	**RESERVED** Write 0.	R/W	R/W	000b
4	**ERR Override** **0:** Override disabled **1:** Override enabled	R/W	R/W	0b
3:0	**ERR LED Code** Code **0h:** Off **1h-Ch:** Flash 1x-12x **Dh:** Blinking **Eh:** Flickering **Fh:** On	R/W	R/W	0h

Note: New error conditions will disable ERR LED Override (bit 4 = 0). The value read in this register always reflects the current LED output.

11.16.25 PDI CONTROL REGISTER

Offset: 0140h Size: 8 bits

This register sets to Process Data Interface (PDI) of the device. Refer to Section 13.0, Chip Mode Configuration for additional information.

Bits	Description	ECAT Type	PDI Type	Default
7:0	**Process Data Interface (PDI_SELECT)** **04h:** Digital I/O **05h:** Beckhoff SPI **80h:** SPI (LAN9252 Compatibility Mode) **82h:** SPI (EtherCAT Direct Mapped Mode) **88h:** HBI Multiplexed 1 Phase 8-bit **89h:** HBI Multiplexed 1 Phase 16-bit **8Ah:** HBI Multiplexed 2 Phase 8-bit **8Bh:** HBI Multiplexed 2 Phase 16-bit **8Ch:** HBI Indexed 8-bit **8Dh:** HBI Indexed 16-bit **8Eh:** HBI Demultiplexed 8-bit **8Fh:** HBI Demultiplexed 16-bit **90h:** HBI Multiplexed 1 Phase 8-bit (EtherCAT Direct Mapped Mode) **91h:** HBI Multiplexed 1 Phase 16-bit (EtherCAT Direct Mapped Mode) **92h:** HBI Multiplexed 2 Phase 8-bit (EtherCAT Direct Mapped Mode) **93h:** HBI Multiplexed 2 Phase 16-bit (EtherCAT Direct Mapped Mode) **94h:** HBI Indexed 8-bit (EtherCAT Direct Mapped Mode) **95h:** HBI Indexed 16-bit (EtherCAT Direct Mapped Mode) **96h:** HBI Demultiplexed 8-bit (EtherCAT Direct Mapped Mode) **97h:** HBI Demultiplexed 16-bit (EtherCAT Direct Mapped Mode) **Others:** RESERVED	RO	RO	00h Note 10

Note 10: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

11.16.26 ESC CONFIGURATION REGISTER

Offset: 0141h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7	**RESERVED**	RO	RO	0b
6	**Enhanced Link Port 2** **0:** Disabled (if bit 1 = 0) **1:** Enabled	RO	RO	0b Note 11
5	**Enhanced Link Port 1** **0:** Disabled (if bit 1 = 0) **1:** Enabled	RO	RO	0b Note 11
4	**Enhanced Link Port 0** **0:** Disabled (if bit 1 = 0) **1:** Enabled	RO	RO	0b Note 11
3	**Distributed Clocks Latch In Unit** **0:** Disabled (power saving) **1:** Enabled **Note:** This bit has no affect.	RO	RO	0b
2	**Distributed Clocks SYNC Out Unit** **0:** Disabled (power saving) **1:** Enabled **Note:** This bit has no affect.	RO	RO	0b
1	**Enhanced Link Detection All Ports** **0:** Disabled (if bits [7:4] = 0) **1:** Enabled all ports	RO	RO	0b Note 11
0	**Device Emulation** (control of AL Status Register) **0:** AL Status Register must be set by PDI **1:** AL Status Register set to value written to AL Control Register **Note:** The value programmed should be 1 for Digital I/O mode and 0 for applications with a host controller.	RO	RO	0b Note 11

Note 11: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

Note: This register is initialized from the contents of the EEPROM. The EEPROM settings for Enhanced Link detection (bits 6,5,4,1) are only taken at the first EEPROM loading after power-on reset. Changing the EEPROM and manually reloading it will not affect the Enhanced link detection enable status, even if the EEPROM could not be read initially.

11.16.27 ASIC CONFIGURATION REGISTER

Offset: 0142h-0143h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15	**MI Link Detection** (Link configuration, link detection, registers PHY Port Status Registers) **0:** Not available **1:** MI Link Detection Active	RO	RO	0b Note 12
14	**ERRLED Enable** **0:** ERRLED disabled **1:** ERRLED enabled	RO	RO	0b Note 12
13:8	**RESERVED**	RO	RO	000000b Note 12
7	**MI Write Gigabit Register 9 Enable** Enables writes to PHY register 9 for PHYs which use this register per IEEE 802.3 **0:** MI writes to Gigabit register 9 disabled **1:** MI writes to Gigabit register 9 enabled	RO	RO	0b Note 12
6	**STATE_RUNLED Mode Select** **0:** STATE_RUNLED disabled **1:** STATE_RUNLED enabled	RO	RO	0b Note 12
5:0	**RESERVED**	RO	RO	0000000b Note 12

Note 12: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.28 RESERVED REGISTER

Offset: 0144h-0145h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**RESERVED**	RO	RO	0000h Note 13

Note 13: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.29 PDI CONFIGURATION REGISTER

Offset: 0150h Size: 8 bits

The bit definitions of this register are dependent on the selected PDI mode (Process Data Interface (PDI_SELECT) field in the PDI Control Register): Digital I/O Mode, SPI Mode, Beckhoff SPI Mode, or HBI Modes.

PDI Configuration Register: Digital I/O Mode

Bits	Description	ECAT Type	PDI Type	Default
7:6	**Output Data Sample Selection** **00:** End of Frame **01:** RESERVED **10:** DC SYNC0 event **11:** DC SYNC1 event **Note:** If OUTVALID Mode = 1, output DATA is updated at Process Data Watchdog trigger event (Output Data Sample Selection bit ignored)	RO	RO	00b Note 14
5:4	**Input Data Sample Selection** **00:** Start of Frame **01:** Rising edge of LATCH_IN **10:** DC SYNC0 event **11:** DC SYNC1 event	RO	RO	00b Note 14
3	**Watchdog Behavior** **0:** Outputs are reset immediately after watchdog expires **1:** Outputs are reset with next output event that follows watchdog expiration	RO	RO	0b Note 14
2	**Unidirectional/Bidirectional Mode** **0:** Unidirectional Mode: input/output direction of pins configured individually **1:** Bidirectional Mode: all I/O pins are bidirectional. Note: Direction control must be set to input.	RO	RO	0b Note 14
1	**OUTVALID Mode** **0:** Output event signaling **1:** Process Data Watchdog trigger (WD_TRIG) signaling on OUTVALID. Output data is updated if watchdog is triggered. Overrides Output Data Sample Selection bit.	RO	RO	0b Note 14
0	**OUTVALID Polarity** **0:** Active high **1:** Active low	RO	RO	0b Note 14

Note 14: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

PDI Configuration Register: SPI Mode

Bits	Description	ECAT Type	PDI Type	Default
7:3	**RESERVED** **Note:** Set EEPROM value to 0.	RO	RO	00000b Note 15
2	**SPI AL Event Request & INT_STS Enable** **0:** SPI AL Event Request & INT_STS disabled **1:** SPI AL Event Request & INT_STS enabled	RO	RO	0b Note 15
1	**SPI WAIT_ACK Polarity** **0:** Active low **1:** Active high	RO	RO	0b Note 15
0	**SPI WAIT_ACK Buffer Type** **0:** Push-Pull **1:** Open Drain	RO	RO	0b Note 15

Note 15: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

PDI Configuration Register: Beckhoff SPI Mode

Bits	Description	ECAT Type	PDI Type	Default
7:6	**RESERVED** **Note:** Set EEPROM value to 0.	RO	RO	00b Note 16
5	**Data Out Sample Mode** **0:** Normal sample (SPI_DO and SPI_DI are sampled at the same SPI_CLK edge) **1:** Late sample SPI_DO and SPI_DI are sampled at different SPI_CLK edges)	RO	RO	0b Note 16
4	**SCS# Polarity** **0:** Active low **1:** Active high	RO	RO	0b Note 16
3:2	**RESERVED** **Note:** Set EEPROM value to 0.	RO	RO	00b Note 16
1:0	**SPI Mode** **00:** SPI Mode 0 **01:** SPI Mode 1 **10:** SPI Mode 2 **11:** SPI Mode 3	RO	RO	00b Note 16

Note 16: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

PDI Configuration Register: HBI Modes

Bits	Description	ECAT Type	PDI Type	Default
7	**HBI ALE Qualification** Configures the HBI interface to qualify the **ALEHI** and **ALELO** signals with the **CS** signal. **0:** Address input is latched with **ALEHI** and **ALELO** **1:** Address input is latched with **ALEHI** and **ALELO** only when **CS** is active.	RO	RO	0b Note 17
6	**HBI Read/Write Mode** Configures the HBI interface for separate read and write signals or direction and enable signals. **0:** Read and Write **1:** Direction and Enable	RO	RO	0b Note 17
5	**HBI Chip Select Polarity** Configures the polarity of the HBI interface chip select signal. **0:** Active Low **1:** Active High	RO	RO	0b Note 17
4	**HBI Read, Read/Write Polarity** Configures the polarity of the HBI interface read signal. **0:** Active Low Read **1:** Active High Read Configures the polarity of the HBI interface read/write signal. **0:** Read when 1, write when 0 (R/nW) **1:** Write when 1, read when 0 (W/nR)	RO	RO	0b Note 17
3	**HBI Write, Enable Polarity** Configures the polarity of the HBI interface write signal. **0:** Active Low Write **1:** Active High Write Configures the polarity of the HBI interface read/write signal. **0:** Active Low Enable **1:** Active High Enable	RO	RO	0b Note 17
2	**HBI ALE Polarity** Configures the polarity of the HBI interface **ALEHI** and **ALELO** signals. **0:** Active Low Strobe (Address saved on rising edge) **1:** Active High Strobe (Address saved on falling edge)	RO	RO	0b Note 17
1	**HBI WAIT_ACK Polarity** **0:** Active low **1:** Active high	RO	RO	0b Note 17
0	**HBI WAIT_ACK Buffer Type** **0:** Push-Pull **1:** Open Drain	RO	RO	0b Note 17

Note 17: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

11.16.30 SYNC/LATCH PDI CONFIGURATION REGISTER

Offset: 0151h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7	**SYNC1 Map** SYNC1 mapped to AL Event Request Register (0220h bit 3) **0:** Disabled **1:** Enabled	RO	RO	0b Note 18
6	**SYNC1/LATCH1 Configuration** **0:** LATCH1 Input **1:** SYNC1 Output	RO	RO	0b Note 18
5:4	**SYNC1 Output Driver/Polarity** **00:** Push-Pull Active Low **01:** Open Drain (Active Low) **10:** Push-Pull Active High **11:** Open Source (Active High)	RO	RO	00b Note 18
3	**SYNC0 Map** SYNC0 mapped to AL Event Request Register (0220h bit 2) **0:** Disabled **1:** Enabled	RO	RO	0b Note 18
2	**SYNC0/LATCH0 Configuration** **0:** LATCH0 Input **1:** SYNC0 Output	RO	RO	0b Note 18
1:0	**SYNC0 Output Driver/Polarity** **00:** Push-Pull Active Low **01:** Open Drain (Active Low) **10:** Push-Pull Active High **11:** Open Source (Active High)	RO	RO	00b Note 18

Note 18: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

11.16.31 EXTENDED PDI CONFIGURATION REGISTER

Offset: 0152h-0153h Size: 16 bits

The bit definitions of this register are dependent on the selected PDI mode (Process Data Interface (PDI_SELECT) field in the PDI Control Register): Digital I/O Mode, SPI Mode, or HBI Modes.

Extended PDI Configuration Register: Digital I/O Mode

Bits	Description	ECAT Type	PDI Type	Default
15	**I/O[31:30] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
14	**I/O[29:28] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
13	**I/O[27:26] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
12	**I/O[25:24] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
11	**I/O[23:22] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
10	**I/O[21:20] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
9	**I/O[19:18] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
8	**I/O[17:16] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
7	**I/O[15:14] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19

Bits	Description	ECAT Type	PDI Type	Default
6	**I/O[13:12] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
5	**I/O[11:10] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
4	**I/O[9:8] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
3	**I/O[7:6] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
2	**I/O[5:4] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
1	**I/O[3:2] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19
0	**I/O[1:0] Direction** **0:** Input **1:** Output **Note:** Must be cleared to 0 during bidirectional mode.	RO	RO	0b Note 19

Note 19: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

Extended PDI Configuration Register: SPI Mode

Bits	Description	ECAT Type	PDI Type	Default
15	**I/O[15:14] Buffer Type** **0:** Push-Pull **1:** Open Drain	RO	RO	0b Note 20
14	**I/O[13:12] Buffer Type** **0:** Push-Pull **1:** Open Drain	RO	RO	0b Note 20
13	**I/O[11:10] Buffer Type** **0:** Push-Pull **1:** Open Drain	RO	RO	0b Note 20
12	**I/O[9:8] Buffer Type** **0:** Push-Pull **1:** Open Drain	RO	RO	0b Note 20
11	**I/O[7:6] Buffer Type** **0:** Push-Pull **1:** Open Drain	RO	RO	0b Note 20
10	**I/O[5:4] Buffer Type** **0:** Push-Pull **1:** Open Drain	RO	RO	0b Note 20
9	**I/O[3:2] Buffer Type** **0:** Push-Pull **1:** Open Drain	RO	RO	0b Note 20
8	**I/O[1:0] Buffer Type** **0:** Push-Pull **1:** Open Drain	RO	RO	0b Note 20
7	**I/O[15:14] Direction** **0:** Input **1:** Output	RO	RO	0b Note 20
6	**I/O[13:12] Direction** **0:** Input **1:** Output	RO	RO	0b Note 20
5	**I/O[11:10] Direction** **0:** Input **1:** Output	RO	RO	0b Note 20
4	**I/O[9:8] Direction** **0:** Input **1:** Output	RO	RO	0b Note 20
3	**I/O[7:6] Direction** **0:** Input **1:** Output	RO	RO	0b Note 20
2	**I/O[5:4] Direction** **0:** Input **1:** Output	RO	RO	0b Note 20
1	**I/O[3:2] Direction** **0:** Input **1:** Output	RO	RO	0b Note 20

Bits	Description	ECAT Type	PDI Type	Default
0	**I/O[1:0] Direction** **0:** Input **1:** Output	RO	RO	0b Note 20

Note 20: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

Extended PDI Configuration Register: HBI Mode

Bits	Description	ECAT Type	PDI Type	Default
15:3	**RESERVED**	RO	RO	000h Note 21
2	**HBI BE1/BE0 Polarity** **0:** BE1/BE0 pins active low **1:** BE1/BE0 pins active high **Note:** Multiplexed and Demultiplexed Modes only	RO	RO	0b Note 21
1	**Perform Internal Write** **0:** Following host write cycle (posted) **1:** Following host read cycle (non-posted)	RO	RO	0b Note 21
0	**Read WAIT_ACK Removal Delay** **0:** Normal read WAIT_ACK output **1:** Delayed read WAOT_ACK output	RO	RO	0b Note 21

Note 21: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.32 ECAT EVENT MASK REGISTER

Offset: 0200h-0201h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**ECAT Event Mask** ECAT event masking of the ECAT Event Request register Events for mapping into the ECAT event fields of EtherCAT frames. **0:** Corresponding ECAT Event Request register bit is not mapped **1:** Corresponding ECAT Event Request register bit is mapped	R/W	RO	0000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.33 AL EVENT MASK REGISTER

Offset: 0204h-0207h Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	**AL Event Mask** AL event masking of the AL Event Request register Events for mapping to the PDI IRQ signal. **0:** Corresponding AL Event Request register bit is not mapped **1:** Corresponding AL Event Request register bit is mapped	RO	R/W	00FFFF0Fh

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.34 ECAT EVENT REQUEST REGISTER

Offset: 0210h-0211h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:12	**RESERVED**	RO	RO	0000b
11	**SyncManager Status Mirror** This bit mirrors the value of the SyncManager Channel 7 Status. **0:** No Sync Channel 7 Event **1:** Sync Channel 7 Event Pending	RO	RO	0b
10	**SyncManager Status Mirror** This bit mirrors the value of the SyncManager Channel 6 Status. **0:** No Sync Channel 6 Event **1:** Sync Channel 6 Event Pending	RO	RO	0b
9	**SyncManager Status Mirror** This bit mirrors the value of the SyncManager Channel 5 Status. **0:** No Sync Channel 5 Event **1:** Sync Channel 5 Event Pending	RO	RO	0b
8	**SyncManager Status Mirror** This bit mirrors the value of the SyncManager Channel 4 Status. **0:** No Sync Channel 4 Event **1:** Sync Channel 4 Event Pending	RO	RO	0b
7	**SyncManager Status Mirror** This bit mirrors the value of the SyncManager Channel 3 Status. **0:** No Sync Channel 3 Event **1:** Sync Channel 3 Event Pending	RO	RO	0b
6	**SyncManager Status Mirror** This bit mirrors the value of the SyncManager Channel 2 Status. **0:** No Sync Channel 2 Event **1:** Sync Channel 2 Event Pending	RO	RO	0b
5	**SyncManager Status Mirror** This bit mirrors the value of the SyncManager Channel 1 Status. **0:** No Sync Channel 1 Event **1:** Sync Channel 1 Event Pending	RO	RO	0b
4	**SyncManager Status Mirror** This bit mirrors the value of the SyncManager Channel 0 Status. **0:** No Sync Channel 0 Event **1:** Sync Channel 0 Event Pending	RO	RO	0b
3	**AL Status Event** **0:** No change in AL Status **1:** AL Status Change **Note:** This bit is cleared by reading the AL Status Register from ECAT.	RO	RO	0b
2	**DL Status Event** **0:** No change in DL Status **1:** DL Status Change **Note:** This bit is cleared by reading the ESC DL Status Register from ECAT.	RO	RO	0b

Bits	Description	ECAT Type	PDI Type	Default
1	**RESERVED**	RO	RO	0b
0	**DC Latch Event** **0:** No change on DC Latch Inputs **1:** At least one change on DC Latch Inputs **Note:** This bit is cleared by reading the DC Latch event times from ECAT for ECAT controlled Latch Units, so that the LATCH0 Status Register/LATCH1 Status Register indicates no event.	RO	RO	0b

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.35 AL EVENT REQUEST REGISTER

Offset: 0220h-0223h Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:16	**RESERVED**	RO	RO	00h
15	**SyncManager 7 Interrupts** (SyncManager register offset 5h, bit 0 or 1) **0:** No SyncManager 7 Interrupt **1:** SyncManager 7 Interrupt pending	RO	RO	0b
14	**SyncManager 6 Interrupts** (SyncManager register offset 5h, bit 0 or 1) **0:** No SyncManager 6 Interrupt **1:** SyncManager 6 Interrupt pending	RO	RO	0b
13	**SyncManager 5 Interrupts** (SyncManager register offset 5h, bit 0 or 1) **0:** No SyncManager 5 Interrupt **1:** SyncManager 5 Interrupt pending	RO	RO	0b
12	**SyncManager 4 Interrupts** (SyncManager register offset 5h, bit 0 or 1) **0:** No SyncManager 4 Interrupt **1:** SyncManager 4 Interrupt pending	RO	RO	0b
11	**SyncManager 3 Interrupts** (SyncManager register offset 5h, bit 0 or 1) **0:** No SyncManager 3 Interrupt **1:** SyncManager 3 Interrupt pending	RO	RO	0b
10	**SyncManager 2 Interrupts** (SyncManager register offset 5h, bit 0 or 1) **0:** No SyncManager 2 Interrupt **1:** SyncManager 2 Interrupt pending	RO	RO	0b

Bits	Description	ECAT Type	PDI Type	Default
9	**SyncManager 1 Interrupts** (SyncManager register offset 5h, bit 0 or 1) **0:** No SyncManager 1 Interrupt **1:** SyncManager 1 Interrupt	RO	RO	0b
8	**SyncManager 0 Interrupts** (SyncManager register offset 5h, bit 0 or 1) **0:** No SyncManager 0 Interrupt **1:** SyncManager 0 Interrupt pending	RO	RO	0b
7	**RESERVED**	RO	RO	0b
6	**Watchdog Process Data** **0:** Has not expired **1:** Has expired **Note:** This bit is cleared by reading the Watchdog Status Process Data Register.	RO	RO	0b
5	**EEPROM Emulation** **0:** No command pending **1:** EEPROM command pending **Note:** This bit is cleared by acknowledging the command in EEPROM Control/Status Register from PDI.	RO	RO	0b
4	**SyncManager *x* Activation Register Changed** (SyncManager x Activate Register) **0:** No change in any SyncManager **1:** At least one SyncManager changed **Note:** This bit is cleared by reading the corresponding SyncManager x Activate Register from PDI.	RO	RO	0b
3	**State of DC SYNC1** (If Sync/Latch PDI Configuration Register bit 7 = 1) **Note:** Bit is cleared by reading SYNC1 status 0x098F.	RO	RO	0b
2	**State of DC SYNC0** (If Sync/Latch PDI Configuration Register bit 3 = 1) **Note:** Bit is cleared by reading SYNC0 status 0x098E.	RO	RO	0b
1	**DC Latch Event** **0:** No change on DC Latch Inputs **1:** At least one change on DC Latch Inputs **Note:** This bit is cleared by reading the DC Latch event times from PDI for PDI controlled Latch Units, so that the LATCH0 Status Register/LATCH1 Status Register indicates no event.	RO	RO	0b
0	**AL Control Event** **0:** No AL Control Register change **1:** AL Control Register has been written (AL control event is only generated if PDI emulation is turned off (ESC Configuration Register bit 8 = 0). **Note:** This bit is cleared by reading the AL Control Register from PDI.	RO	RO	0b

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.36 RX ERROR COUNTER REGISTERS

Offset: 0300h-0307h
Port 0: 0300h-0301h
Port 1: 0302h-0303h
Port 2: 0304h-0305h
Port 3: 0306h-0307h

Size: 16 bits

There are 4 16-bit RX Error Counter registers, each with unique address offsets as shown above. The variable "***x***" is used in the following bit descriptions to represent ports 0-3.

Bits	Description	ECAT Type	PDI Type	Default
15:8	**Port *x* RX Error Counter** Counting is stopped when FFh is reached. This is coupled directly to RX ERR of the MII/EBUS interfaces.	R/WC	RO	00h
7:0	**Port *x* Invalid Frame Counter** Counting is stopped when FFh is reached.	R/WC	RO	

Note: This register is cleared if any one of the RX Error Counter Registers is written.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

Note: Port 3 is not used.

11.16.37 FORWARDED RX ERROR COUNTER REGISTERS

Offset: 0308h-030Bh
Port 0: 0308h
Port 1: 0309h
Port 2: 030Ah
Port 3: 030Bh

Size: 8 bits

There are 4 8-bit Forwarded RX Error Counter registers, each with unique address offsets as shown above. The variable "***x***" is used in the following bit descriptions to represent ports 0-3.

Bits	Description	ECAT Type	PDI Type	Default
7:0	**Port *x* Forwarded RX Error Counter** Counting is stopped when FFh is reached. This is coupled directly to RX ERR of the MII/EBUS interfaces.	R/WC	RO	00h

Note: This register is cleared if any one of the RX Error Counter Registers is written.

Note: Port 3 is not used.

11.16.38 ECAT PROCESSING UNIT ERROR COUNTER REGISTER

Offset: 030Ch

Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:0	**ECAT Processing Unit Error Counter** Counting is stopped when FFh is reached. This field counts the errors of frames passing the Processing Unit (e.g., FCS error or datagram structure error).	R/WC	RO	00h

11.16.39 PDI ERROR COUNTER REGISTER

Offset: 030Dh

Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:0	**PDI Error Counter** Counting is stopped when FFh is reached. This field counts if a PDI access has an interface error.	R/WC	RO	00h

11.16.40 PDI ERROR CODE REGISTER

Offset: 030Eh Size: 8 bits

The bit definitions of this register are dependent on the selected PDI mode (Process Data Interface (PDI_SELECT) field in the PDI Control Register): SPI Modes or HBI Modes.

Note: This register is cleared when the PDI Error Counter Register is written.

PDI Error Codes: Beckhoff SPI Mode (PDI_SELECT = 05h)

Bits	Description	ECAT Type	PDI Type	Default
7:6	**SPI Command CMD[2:1] of access causing PDI error**	RO	RO	00b
5	**Read Termination Error** **0:** No error **1:** Error detected	RO	RO	0b
4	**Read Termination Missing** **0:** No error **1:** Error detected	RO	RO	0b
3	**Busy Violation During Read Access** **0:** No error **1:** Error detected	RO	RO	0b
2:0	**Number of SPI clock cycles of whole access (modulo 8) causing PDI error**	RO	RO	000b

PDI Error Codes: Microchip SPI Modes (PDI_SELECT = 80h or 82h)

Bits	Description	ECAT Type	PDI Type	Default
7:6	**RESERVED**	RO	RO	00b
5:4	**Read Termination Error** **00:** No error **01:** Read finished without setting SI high for the last byte. **10:** Read continued after setting SI high for the last byte. **11:** Actual read length did not match byte length provided.	RO	RO	00b
3	**Incomplete BYTE or DWORD**	RO	RO	0b
2	**SPI Command During Write** A SPI command started during prior a pending write. **Note:** This bit applies only during EtherCAT Direct Mapped Mode and is reserved in LAN9252 compatibility mode.	RO	RO	0b
1	**Read Data Not Ready** Read output started during data fetch. **Note:** This bit applies only during EtherCAT Direct Mapped Mode and is reserved in LAN9252 compatibility mode.	RO	RO	0b
0	**Write Overrun** Second write finished during prior pending write. **Note:** This bit applies only during EtherCAT Direct Mapped Mode and is reserved in LAN9252 compatibility mode.	RO	RO	0b

PDI Error Codes: HBI Modes (PDI_SELECT = 90h - 97h)

Bits	Description	ECAT Type	PDI Type	Default
7:4	**RESERVED**	RO	RO	0h
3	**Addressing error for a write access** **0:** No error **1:** Error detected	RO	RO	0b
2	**Addressing error for a read access** **0:** No error **1:** Error detected	RO	RO	0b
1	**Busy violation during a write access** **0:** No error **1:** Error detected	RO	RO	0b
0	**Busy violation during a read access** **0:** No error **1:** Error detected	RO	RO	0b

11.16.41 LOST LINK COUNTER REGISTERS

Offset: 0310h-0313h
Port 0: 0310h
Port 1: 0311h
Port 2: 0312h
Port 3: 0313h

Size: 8 bits

There are 4 8-bit Lost Link Counter registers, each with unique address offsets as shown above. The variable "***x***" is used in the following bit descriptions to represent ports 0-3.

Bits	Description	ECAT Type	PDI Type	Default
7:0	**Port *x* Lost Link Counter** Counting is stopped when FFh is reached. This counter only counts if port loop is Auto or Auto-Close. **Note:** Only lost links at open ports are counted.	R/WC	RO	00h

Note: This register is cleared if any one of the Lost Link Counter Registers is written.

Note: Port 3 is not used.

11.16.42 WATCHDOG DIVIDER REGISTER

Offset: 0400h-0401h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**Watchdog Divider** Number of 25MHz ticks (minus 2) that represents the basic watchdog increment. (default value is 100 us = 2498)	R/W	RO	09C2h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.43 WATCHDOG TIME PDI REGISTER

Offset: 0410h-0411h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**Watchdog Time PDI** Number of basic watchdog increments. (default value with Watchdog Divider of 100 us results in 100 ms watchdog.)	R/W	RO	03E8h

Note: The watchdog is disabled if Watchdog Time PDI is set to 0000h. Watchdog is restarted with every PDI access.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.44 WATCHDOG TIME PROCESS DATA REGISTER

Offset: 0420h-0421h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**Watchdog Time Process Data** Number of basic watchdog increments. (default value with Watchdog Divider of 100 us results in 100 ms watchdog.)	R/W	RO	03E8h

Note: There is one watchdog for all SyncManagers. The watchdog is disabled if Watchdog Time PDI is set to 0000h. The watchdog is restarted with every write access to the SyncManagers with the Watchdog Trigger Enable bit set.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.45 WATCHDOG STATUS PROCESS DATA REGISTER

Offset: 0440h-0441h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:1	**RESERVED**	RO	RO	0000h
0	**Watchdog Status of Process Data** (triggered by SyncManagers) **0:** Watchdog Process Data expired **1:** Watchdog Process Data is active or disabled	RO	RO	0b

Note: Reading this register clears the Watchdog Process Data bit of the AL Event Request Register.

Note: The Watchdog Status for the PDI can be read in the PDI Watchdog Status bit of the ESC DL Status Register.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.46 WATCHDOG COUNTER PROCESS DATA REGISTER

Offset: 0442h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:0	**Watchdog Counter Process Data** Counting is stopped when FFh is reached. Counts if Process Data Watchdog expires. This field is cleared if one of the Watchdog counters (0442h-0443h) is written.	R/WC	RO	00h

11.16.47 WATCHDOG COUNTER PDI REGISTER

Offset: 0443h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:0	**Watchdog PDI Counter** Counting is stopped when FFh is reached. Counts if PDI Watchdog expires. This field is cleared if one of the Watchdog counters (0442h-0443h) is written.	R/WC	RO	00h

11.16.48 EEPROM CONFIGURATION REGISTER

Offset: 0500h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:2	**RESERVED** Write 0.	RO	RO	000000b
1	**Force ECAT Access** **0:** Do not change **1:** Reset	R/W	RO	0b
0	**PDI EEPROM Control** **0:** No **1:** Yes (PDI has EEPROM control)	R/W	RO	0b

Note: EtherCAT controls the SII EEPROM interface if the PDI EEPROM Control bit of the EEPROM Configuration Register is 0 and the Access to EEPROM bit of the EEPROM PDI Access State Register is 0. Otherwise, PDI controls the EEPROM interface.

11.16.49 EEPROM PDI ACCESS STATE REGISTER

Offset: 0501h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:1	**RESERVED** Write 0.	RO	RO	0000000b
0	**Access to EEPROM** **0:** PDI releases EEPROM access **1:** PDI takes EEPROM access (PDI has EEPROM control)	RO	R/W Note 22	0b

Note 22: Write access only possible if the PDI EEPROM Control bit of the EEPROM Configuration Register is 1 and Force ECAT Access bit is 0.

Note: EtherCAT controls the SII EEPROM interface if the PDI EEPROM Control bit of the EEPROM Configuration Register is 0 and the Access to EEPROM bit of the EEPROM PDI Access State Register is 0. Otherwise, PDI controls the EEPROM interface.

11.16.50 EEPROM CONTROL/STATUS REGISTER

Offset: 0502h-0503h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15	**Busy** **0:** EEPROM interface is idle **1:** EEPROM interface is busy	RO	RO	0b
14	**Error Write Enable** **0:** No error **1:** Write Command without Write enable (See Note 23)	RO	RO	0b
13	**Error Acknowledge/Command** **0:** No error **1:** Missing EEPROM acknowledge or invalid command (See Note 23) **Note:** EEPROM emulation only: PDI writes 1 if a temporary failure has occurred.	RO	R/[W] Note 24	0b
12	**EEPROM Loading Status** **0:** EEPROM loaded, device information okay **1:** EEPROM not loaded, device information not available (EEPROM loading in-progress or finished with a failure)	RO	RO	0b
11	**Checksum Error in ESC Configuration Area** **0:** Checksum okay **1:** Checksum error	RO	R/[W] Note 24	0b
10:8	**Command Register** Write: Initiate command Read: Currently executed command **000:** No command/EEPROM idle (clear error bits) **001:** Read **010:** Write **100:** Reload **Others:** RESERVED / invalid commands (no not issue) (See Note 25)	R/W	R/[W] Note 24	000b
7	**Selected EEPROM Algorithm** **0:** 1 address byte (1Kbit - 16Kbit EEPROMs) **1:** 2 address bytes (32Kbit - 4Mbit EEPROMs)	RO	RO	Note 26
6	**Supported Number of EEPROM Bytes** **0:** 4 Bytes **1:** 8 Bytes	RO	RO	1b
5	**EEPROM Emulation** **0:** Normal operation (I^2C interface used) **1:** PDI emulates EEPROM (I^2C not used) **Note:** Must be written as 0.	RO	RO	Note 27
4:1	**RESERVED**	RO	RO	0b

Bits	Description	ECAT Type	PDI Type	Default
0	**ECAT Write Enable** **0:** Write requests are disabled **1:** Write requests are enabled (See Note 28)	R/W	RO	0b

Note 23: Error bits are cleared by writing “000” (or any valid command) to Command Register bits.

Note 24: Write access is possible if EEPROM interface is busy (Busy bit = 1). PDI acknowledges pending commands by writing a 1 into the corresponding command register bits 10:8. Errors can be indicated by writing a 1 into the error bits (11 and 13). Acknowledging clears bit 5 of the AL Event Request Register.

Note 25: The Command Register bits are self clearing after the command is executed (EEPROM Busy ends). Writing “000” to the Command Register bits will also clear the error bits 14:13. The Command Register bits are ignored if the Error Acknowledge/Command is pending.

Note 26: The default of this bit is dependent on the **E2PSIZE** configuration strap.

Note 27: The default of this bit is dependent on the **EE_EMUL2** and **EE_EMUL1** configuration straps.

Note 28: The ECAT Write Enable bit is self clearing at the SOF of the next frame.

Note: EtherCAT controls the SII EEPROM interface if the PDI EEPROM Control bit of the EEPROM Configuration Register is 0 and the Access to EEPROM bit of the EEPROM PDI Access State Register is 0. Otherwise, PDI controls the EEPROM interface.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.51 EEPROM ADDRESS REGISTER

Offset: 0504h-0507h Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	**EEPROM Address** Bit 0: First word (16-bit) Bit 1: Second word ..... **Note:** Actually used EEPROM address bits: [9:0]: EEPROM size up to 16Kbit [17:0]: EEPROM size 32Kbit - 4Mbit [31:0]: EEPROM Emulation	R/W	R/W	00000000h

Note: Write access depends upon the assignment of the EEPROM interface (ECAT/PDI). Write access is generally blocked if the EEPROM interface is busy (Busy bit of EEPROM Control/Status Register = 1)

Note: EtherCAT controls the SII EEPROM interface if the PDI EEPROM Control bit of the EEPROM Configuration Register is 0 and the Access to EEPROM bit of the EEPROM PDI Access State Register is 0. Otherwise, PDI controls the EEPROM interface.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.52 EEPROM DATA REGISTER

Offset: 0508h-050Fh Size: 64 bits

Bits	Description	ECAT Type	PDI Type	Default
63:16	**EEPROM Read Data** Data to be read from EEPROM, higher bytes	RO	RO	0000 0000 0000h
15:0	**EEPROM Read/Write Data** Data to be read from EEPROM, lower bytes or data to be written to EEPROM.	R/W	R/W	0000h

Note: Write access depends upon the assignment of the EEPROM interface (ECAT/PDI). Write access is generally blocked if the EEPROM interface is busy (Busy bit of EEPROM Control/Status Register = 1)

Note: EtherCAT controls the SII EEPROM interface if the PDI EEPROM Control bit of the EEPROM Configuration Register is 0 and the Access to EEPROM bit of the EEPROM PDI Access State Register is 0. Otherwise, PDI controls the EEPROM interface.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.53 MII MANAGEMENT CONTROL/STATUS REGISTER

Offset: 0510h-0511h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15	**Busy** **0:** MI control state machine is idle **1:** MI control state machine is active	RO	RO	0b
14	**Command Error** **0:** Last command was successful **1:** Invalid command or write command without write enable **Note:** Cleared with a valid command or by writing “00” to Command Register.	RO	RO	0b
13	**Read Error** **0:** No read error **1:** Read error occurred (PHY or register bi available) **Note:** Cleared by writing this register.	R/W Note 29	R/W Note 29	0b
12:10	**RESERVED**	RO	RO	0b
9:8	**Command Register** Write: Initiate command. Read: Currently executed command See Note 30. Commands: **00:** No command / MI Idle (clear error bits) **01:** Read **10:** Write **11:** RESERVED (do not issue)	R/W Note 29	R/W Note 29	00b
7:3	**PHY Address Offset**	RO	RO	00000b
2	**MI Link Detection** (Link configuration, link detection, registers PHY Port Status Registers) **0:** Not available **1:** MI Link Detection Active	RO	RO	0b Note 31
1	**Management Interface Control** **0:** ECAT control only **1:** MPDI control possible (MII Management ECAT Access State Register and MII Management PDI Access State Register)	RO	RO	1b
0	**Write Enable** **0:** Write Disabled **1:** Write Enabled **Note:** This bit is always 1 if PDI has MI control. (See Note 32)	R/W Note 29	RO	0b

Note 29: Write access depends upon the assignment of the MI interface (ECAT/PDI). Write access is generally blocked if the MII interface is busy (Busy bit of MII Management Control/Status Register = 1)

Note 30: Command Register bits (9:8) are self-clearing after the command is executed (Busy ends). Writing “00” to the Command Register bits will also clear the error bits 14:13 of this register. The Command Register bits (9:8) are cleared after the command is executed.

Note 31: The default value of this field can be configured via EEPROM. This bit will be 0 and MI link detection disabled until the device is successfully configured from EEPROM. The EEPROM setting for MI link detection is only taken at the first EEPROM loading after power-on reset. Changing the EEPROM and manually reloading it will not affect the MI link detection enable status, even if the EEPROM could not be read initially. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

Note 32: Write enable bit 0 is self-clearing at the SOF of the next frame (or end of the PDI access).

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.54 PHY ADDRESS REGISTER

Offset: 0512h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
6:5	**RESERVED** Write 0.	RO	RO	000b
4:0	**PHY Address**	R/W Note 33	R/W Note 33	00000b

Note 33: Write access depends upon the assignment of the MI interface (ECAT/PDI). Write access is generally blocked if the MII interface is busy (Busy bit of MII Management Control/Status Register = 1)

11.16.55 PHY REGISTER ADDRESS REGISTER

Offset: 0513h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:5	**RESERVED** Write 0.	RO	RO	000b
4:0	**Address of PHY Register to be Read/Written**	R/W Note 34	R/W Note 34	00000b

Note 34: Write access depends upon the assignment of the MI interface (ECAT/PDI). Write access is generally blocked if the MII interface is busy (Busy bit of MII Management Control/Status Register = 1)

11.16.56 PHY DATA REGISTER

Offset: 0514h-0515h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**PHY Read/Write Data**	R/W Note 35	R/W Note 35	0000h

Note 35: Write access depends upon the assignment of the MI interface (ECAT/PDI). Write access is generally blocked if the MII interface is busy (Busy bit of MII Management Control/Status Register = 1)

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.57 MII MANAGEMENT ECAT ACCESS STATE REGISTER

Offset: 0516h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:1	**RESERVED** Write 0.	RO	RO	0000000b
0	**Access to MII Management (ECAT)** **0:** ECAT enables PDI takeover of MII management control **1:** ECAT claims exclusive access to MII management	R/W Note 36	RO	0b

Note 36: Write access only possible if the Access to MII Management (PDI) bit of the MII Management PDI Access State Register is 0.

11.16.58 MII MANAGEMENT PDI ACCESS STATE REGISTER

Offset: 0517h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:2	**RESERVED** Write 0.	RO	RO	000000b
1	**Force PDI Access State** **0:** Do not change Access to MII Management (PDI) bit **1:** Reset Access to MII Management (PDI) bit	R/W	RO	0b
0	**Access to MII Management (PDI)** **0:** ECAT has access to MII management **1:** PDI has access to MII management	RO	R/W Note 37	0b

Note 37: Write access to the Access to MII Management (PDI) bit of this register is only possible if the Force PDI Access State bit of this register is 0 and the Access to MII Management (ECAT) bit of the MII Management ECAT Access State Register is 0.

11.16.59 PHY PORT STATUS REGISTERS

Offset: 0518h-051Bh Size: 8 bits
Port 0: 0518h
Port 1: 0519h
Port 2: 051Ah
Port 3: 051Bh

There are 4 8-bit PHY Port Status registers, each with unique address offsets as shown above. The variable "***x***" is used in the following bit descriptions to represent ports 0-3.

Bits	Description	ECAT Type	PDI Type	Default
7:6	**RESERVED** Write as 0.	RO	RO	00b
5	**Port *x* Lost Link Counter** **0:** No Update **1:** PHY Configuration was Updated **Note:** Cleared by writing any value to at least one of the PHY Port Status Registers.	R/WC Note 38	R/WC Note 38	0b
4	**Port *x* Link Partner Error** **0:** No Error Detected **1:** Link Partner Error	RO	RO	0b
3	**Port *x* Read Error** **0:** No Read Error Detected **1:** Read Error has Occurred **Note:** Cleared by writing any value to at least one of the PHY Port Status Registers.	R/WC Note 38	R/WC Note 38	0b

Bits	Description	ECAT Type	PDI Type	Default
2	**Port *x* Link Status Error** **0:** No Error **1:** Link Error, Link Inhibited	RO	RO	0b
1	**Port *x* Link Status** (100 Mbit/s, Full-Duplex, Auto-negotiation) **0:** No Link **1:** Link Detected	RO	RO	0b
0	**Port *x* Physical Link** (PHY Status Register 1.2) **0:** No Physical Link **1:** Physical Link Detected	RO	RO	0b

Note 38: Write access depends upon the assignment of the MI interface (ECAT/PDI).

Note: Port 3 is not used.

11.16.60 FMMU[2:0] REGISTERS

The device includes 8 FMMUs. Each FMMU is described in 16 Bytes, starting at 0600h. Table 11-19 details the base address for each FMMU. The subsequent FMMU registers will be referenced as an offset from these various base addresses. The variable "***x***" is used in the following descriptions to represent FMMUs 0 through 7.

TABLE 11-19: FMMU *X* BASE ADDRESSES

FMMU	Base Address
0	0600h
1	0610h
2	0620h
3	0630h
4	0640h
5	0650h
6	0660h
7	0670h

11.16.60.1 FMMU*x* Logical Start Address Register

Offset: FMMU*x* Base Address +0h-3h Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	**Logical Start Address** Logical start address within the EtherCAT address space.	R/W	RO	00000000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.60.2 FMMU*x* Length Register

Offset: FMMU*x* Base Address +4h-5h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**Length** Offset from the first logical FMMU byte to the last FMMU Byte + 1 (e.g., if two bytes are used, then this parameter shall contain 2).	R/W	RO	0000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.60.3 FMMU*x* Logical Start Bit Register

Offset: FMMU*x* Base Address +6h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:3	**RESERVED** Write as 0.	RO	RO	00000b
2:0	**Logical Start Bit** Logical starting bit that shall be mapped (bits are counted from least significant bit (0) to most significant bit (7)).	R/W	RO	000b

11.16.60.4 FMMU*x* Logical Stop Bit Register

Offset: FMMU*x* Base Address +7h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:3	**RESERVED** Write as 0.	RO	RO	00000b
2:0	**Logical Stop Bit** Last logical bit that shall be mapped (bits are counted from least significant bit (0) to most significant bit (7)).	R/W	RO	000b

11.16.60.5 FMMU*x* Physical Start Address Register

Offset: FMMU*x* Base Address +8h-9h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**Physical Start Address** (Mapped to logical start address)	R/W	RO	0000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.60.6 FMMU*x* Physical Start Bit Register

Offset: FMMU*x* Base Address +Ah Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:3	**RESERVED** Write as 0.	RO	RO	00000b
2:0	**Physical Start Bit** Physical starting bit as target of logical start bit mapping (bits are counted from least significant bit (0) to most significant bit (7)).	R/W	RO	000b

11.16.60.7 FMMU*x* Type Register

Offset: FMMU*x* Base Address +Bh Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:2	**RESERVED** Write as 0.	RO	RO	000000b
1	**Write Access Mapping** **0:** Ignore mapping for write accesses **1:** Use mapping for write accesses	R/W	RO	0b
0	**Read Access Mapping** **0:** Ignore mapping for read accesses **1:** Use mapping for read accesses	R/W	RO	0b

11.16.60.8 FMMU*x* Activate Register

Offset: FMMU*x* Base Address +Ch Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:1	**RESERVED** Write as 0.	RO	RO	0000000b
0	**FMMU Activation** **0:** FMMU*x* Deactivated **1:** FMMU*x* Activated. FMMU*x* checks logical addressed blocks to be mapped according to the configured mapping.	R/W	RO	0b

11.16.60.9 FMMU*x* Reserved Register

Offset: FMMU*x* Base Address +Dh-Fh Size: 24 bits

Bits	Description	ECAT Type	PDI Type	Default
23:0	**RESERVED** Write as 0.	RO	RO	000000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.61 SYNCMANAGER[3:0] REGISTERS

The device includes 8 SyncManagers. Each SyncManager is described in 8 Bytes, starting at 0800h. Table 11-20 details the base address for each SyncManager. The subsequent SyncManager registers will be referenced as an offset from these various base addresses. The variable "***x***" is used in the following descriptions to represent SyncManagers 0 through 7.

TABLE 11-20: SYNCMANAGER *X* BASE ADDRESSES

SyncManager	Base Address
0	0800h
1	0808h
2	0810h
3	0818h
4	0820h
5	0828h
6	0830h
7	0838h

11.16.61.1 SyncManager *x* Physical Start Address Register

Offset: SyncManager *x* Base Address +0h-1h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**Physical Start Address** Specifies the first byte that will be handled by SyncManager *x*.	R/W Note 39	RO	0000h

Note 39: This register can only be written if the corresponding SyncManager is disabled via the SyncManager Enable/Disable bit of the SyncManager x Activate Register.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.61.2 SyncManager *x* Length Register

Offset: SyncManager *x* Base Address +2h-3h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**Length** Number of bytes assigned to SyncManager *x*. (This field shall be greater than 1, otherwise the SyncManager is not activated. If set to 1, only Watchdog Trigger is generated, if configured.)	R/W Note 40	RO	0000h

Note 40: This register can only be written if SyncManager *x* is disabled via the SyncManager Enable/Disable bit of the SyncManager x Activate Register.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.61.3 SyncManager *x* Control Register

Offset: SyncManager *x* Base Address +4h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7	**RESERVED** Write as 0.	RO	RO	0b
6	**Watchdog Trigger Enable** **0:** Disabled **1:** Enabled	R/W Note 41	RO	0b
5	**Interrupt in PDI Event Request Register** **0:** Disabled **1:** Enabled	R/W Note 41	RO	0b
4	**Interrupt in ECAT Event Request Register** **0:** Disabled **1:** Enabled	R/W Note 41	RO	0b
3:2	**Direction** **00:** Read: ECAT read access, PDI write access **01:** Write: ECAT write access, PDI read access **10:** RESERVED **11:** RESERVED	R/W Note 41	RO	00b
1:0	**Operation Mode** **00:** Buffered (3 buffer mode) **01:** RESERVED **10:** Mailbox (single buffer mode) **11:** RESERVED	R/W Note 41	RO	00b

Note 41: This register can only be written if SyncManager *x* is disabled via the SyncManager Enable/Disable bit of the SyncManager x Activate Register.

11.16.61.4 SyncManager *x* Status Register

Offset: SyncManager *x* Base Address +5h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7	**Write Buffer in Use** (opened)	RO	RO	0b
6	**Read Buffer in Use** (opened)	RO	RO	0b
5:4	**Buffer Status (Last Written Buffer)** **Buffered Mode:** **00:** 1. buffer **01:** 2. buffer **10:** 3. buffer **11:** No buffer written **Mailbox Mode:** RESERVED	RO	RO	11b
3	**Mailbox Status** **Mailbox Mode:** **0:** Mailbox Empty **1:** Mailbox Full **Buffered Mode:** RESERVED	RO	RO	0b
2	**RESERVED** Write as 0.	RO	RO	0b
1	**Interrupt Read** **0:** Interrupt cleared after first byte of buffer was written **1:** Interrupt after buffer was completely and successfully read	RO	RO	0b
0	**Interrupt Write** **0:** Interrupt cleared after first byte of buffer was read **1:** Interrupt after buffer was completely and successfully written	RO	RO	0b

11.16.61.5 SyncManager *x* Activate Register

Offset: SyncManager *x* Base Address +6h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7	**Latch Event PDI** **0:** No **1:** Generate latch events if PDI issues a buffer exchange or if PDI accesses buffer start address.	R/W	RO	0b
6	**Latch Event ECAT** **0:** No **1:** Generate latch event if EtherCAT master issues a buffer exchange.	R/W	RO	0b
5:2	**RESERVED** Write as 0.	RO	RO	0000b
1	**Repeat Request** A toggle of Repeat Request indicates that a mailbox retry is needed (primarily used in conjunction with ECAT Read Mailbox)	R/W	RO	0b
0	**SyncManager Enable/Disable** **0:** Disable: Access to memory without SyncManager control **1:** Enable: SyncManager is active and controls memory area set in configuration.	R/W	RO	0b

Note: Reading this register from PDI in all SyncManagers which have changed activation clears the "SyncManager x Activation Register Changed" bit in the AL Event Request Register.

11.16.61.6 SyncManager *x* PDI Control Register

Offset: SyncManager *x* Base Address +7h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:2	**RESERVED** Write as 0.	RO	RO	000000b
1	**Repeat Ack** If this is set to the same value as Repeat Request, the PDI acknowledges the execution of a previous set repeat request.	RO	R/W	0b
0	**Deactivate SyncManager *x*** **Read:** **0:** Normal operation, SyncManager *x* activated **1:** SyncManager *x* deactivated and reset SyncManager *x* locks access to memory area **Write:** **0:** Activate SyncManager **1:** Request SyncManager Deactivation	RO	R/W	0b

11.16.62 RECEIVE TIME PORT 0 REGISTER

Offset: 0900h-0903h Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:8	Local time at the beginning of the last receive frame containing a write access to register 09000h.	RO	RO	Undefined
7:0	Write: A write access to register 0900h with BWR or FPWR (configured address) latches the local time of the beginning of the receive frame (start first bit of preamble) at each port. Read: Local time of the beginning of the last receive frame containing a write access to this register. **Note:** The time stamps cannot be read in the same frame in which this register was written.	R/W	RO	Undefined

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.63 RECEIVE TIME PORT 1 REGISTER

Offset: 0904h-0907h Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	Local time at the beginning of a frame (start first bit of preamble) received at port 1 containing a BWR or FPWR to the Receive Time Port 0 Register.	RO	RO	Undefined

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.64 RECEIVE TIME PORT 2 REGISTER

Offset: 0908h-090Bh Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	Local time of the beginning of a frame (start first bit of preamble) received at port 2 containing a BWR or FPWR to the Receive Time Port 0 Register.	RO	RO	Undefined

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.65 SYSTEM TIME REGISTER

Offset: 0910h-0917h Size: 64 bits

Bits	Description	ECAT Type	PDI Type	Default
63:0	ECAT Read Access: Local copy of the System Time when the frame passed the reference clock (i.e., including System Time Delay). Time latched at beginning of the frame (Ethernet SOF delimiter). PDI Read Access: Local copy of the System Time. Time latched when reading first byte (0910h).	RO	RO	00000000h 00000000h
31:0	ECAT Write Access: Written value will be compared with the local copy of the system time. The result is an input to the time control loop. PDI Write Access: Written value will be compared with LATCH0 Time Positive Edge time. The result is an input to the time control loop. **Note:** Written value will be compared at the end of the frame with the latched (SOF) local copy of the system time if at least the first byte (0910h) was written.	W Note 42	W	00000000h

Note 42: When writing via ECAT, the control loop is triggered to process the new value.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.66 RECEIVE TIME ECAT PROCESSING UNIT REGISTER

Offset: 0918h-091Fh Size: 64 bits

Bits	Description	ECAT Type	PDI Type	Default
63:0	Local time of the beginning of a frame (start first bit of preamble) received at the ECAT Processing Unit containing a write access to Receive Time Port 0 Register (0900h). **Note:** If port 0 is open, this register reflects the Receive Time Port 0 Register as a 64-bit value.	RO	RO	00000000h 00000000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.67 SYSTEM TIME OFFSET REGISTER

Offset: 0920h-0927h Size: 64 bits

Bits	Description	ECAT Type	PDI Type	Default
63:0	Difference between local time and System Time. Offset is added to local time. Local time of the beginning of a frame (start first bit of preamble) received at the ECAT Processing Unit containing a write access to Receive Time Port 0 Register (0900h). **Note:** If port 0 is open, this register reflects the Receive Time Port 0 Register as a 64-bit value.	R/W	R/W	00000000h 00000000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.68 SYSTEM TIME DELAY REGISTER

Offset: 0928h-092Bh Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	Delay between Reference Clock and the ESC.	R/W	R/W	00000000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.69 SYSTEM TIME DIFFERENCE REGISTER

Offset: 092Ch-092Fh Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31	**0:** Local copy of System Time smaller than received System Time **1:** Local copy of System Time greater than or equal to received System Time	RO	RO	0b
30:0	Mean difference between local copy of System Time and received System Time values.	RO	RO	00000000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.70 SPEED COUNTER START REGISTER

Offset: 0930h-0931h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15	**RESERVED** Write as 0.	RO	RO	0b
14:0	Bandwidth for adjustment of local copy of System Time (larger values -> smaller bandwidth and smoother adjustment). A write access resets the System Time Difference Register and Speed Counter Diff Register. Valid range: 0080h-3FFFh.	R/W	R/W	1000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.71 SPEED COUNTER DIFF REGISTER

Offset: 0932h-0933h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	Representation of the deviation between local clock period and Reference Clock's clock period (representation: two's compliment). Valid Range: +/-(Speed Counter Start Register-7Fh).	RO	RO	0000h

Note: The clock deviation after System Time Difference has settled at a low value can be calculated as follows: Deviation = Speed Counter Diff / 5(Speed Counter Start + Speed Counter Diff + 2)(Speed Counter Start - Speed Counter Diff + 2)

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.72 SYSTEM TIME DIFFERENCE FILTER DEPTH REGISTER

Offset: 0934h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:4	**RESERVED**	RO	RO	0h
3:0	Filter depth for averaging the received System Time deviation. **Note:** A write access resets the System Time Difference Register.	R/W	R/W	4h

11.16.73 SPEED COUNTER FILTER DEPTH REGISTER

Offset: 0935h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:4	**RESERVED**	RO	RO	0h
3:0	Filter depth for averaging the clock period deviation. **Note:** A write access resets the internal speed counter filter.	R/W	R/W	Ch

11.16.74 CYCLIC UNIT CONTROL REGISTER

Offset: 0980h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:6	**RESERVED** Write as 0.	RO	RO	00b
5	**Latch In Unit 1** **0:** ECAT Controlled **1:** PDI Controlled **Note:** Latch interrupt is routed to ECAT/PDI depending on this setting.	R/W	RO	0b
4	**Latch In Unit 0** **0:** ECAT Controlled **1:** PDI Controlled **Note:** Always 1 (PDI controlled) is System Time is PDI controlled. Latch interrupt is routed to ECAT/PDI depending on this setting.	R/W	RO	0b
3:1	**RESERVED** Write as 0.	RO	RO	000b
0	**Sync Out Unit Control** **0:** ECAT Controlled **1:** PDI Controlled	R/W	RO	0b

11.16.75 ACTIVATION REGISTER

Offset: 0981h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7	**SyncSignal Debug Pulse (Vasili Bit)** **0:** Deactivated **1:** Immediately generate a single debug ping on SYNC0 and SYNC1 according to bits 2 and 1 of this register.	R/W	R/W	0b
6	**Near Future Configuration (approx.)** **0:** 1/2 DC width future (2^{31} ns or 2^{63} ns) **1:** 2.1 sec future (2^{31} ns)	R/W	R/W	0b
5	**Start Time Plausibility Check** **0:** Disabled. SyncSignal generation if Start Time is reached. **1:** Immediate SyncSignal generation if Start Time is outside Near Future Configuration (approx.).	R/W	R/W	0b
4	**Extension of Start Time Cyclic Operation** (Start Time Cyclic Operation Register) **0:** No extension **1:** Extend 32-bit written Start Time to 64-bit	R/W	R/W	0b
3	**Auto-activation** (By writing Start Time Cyclic Operation Register) **0:** Disabled **1:** Auto-activation enabled. Sync Out Unit Activation is set auto-matically after Start Time is written.	R/W	R/W	0b
2	**SYNC1 Generation** **0:** Deactivated **1:** SYNC1 pulse is generated	R/W	R/W	0b
1	**SYNC0 Generation** **0:** Deactivated **1:** SYNC0 pulse is generated	R/W	R/W	0b
0	**Sync Out Unit Activation** **0:** Deactivated **1:** Activated **Note:** Write 1 after Start Time is written	R/W	R/W	0b

Note: Writes to this register depend on the Sync Out Unit Control bit of the Cyclic Unit Control Register.

11.16.76 PULSE LENGTH OF SYNCSIGNALS REGISTER

Offset: 0982h-0983h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**Pulse length of SyncSignals** (in units of 10ns) A value of 0 is used for Acknowledge Mode: SyncSignal will be cleared by reading the SYNC0 Status Register/SYNC1 Status Register.	RO	RO	0000h Note 43

Note 43: The default value of this field can be configured via EEPROM. Refer to Section 11.10, "EEPROM Configurable Registers," on page 289 for additional information.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.77 ACTIVATION STATUS REGISTER

Offset: 0984h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:3	**RESERVED**	RO	RO	00000b
2	Start Time Cyclic Operation (Start Time Cyclic Operation Register) plausibility check result when Sync Out Unit was activated. **0:** Start Time was within near future **1:** Start Time was out of near future	RO	RO	0b
1	**SYNC1 Activation State** **0:** First SYNC1 pulse is not pending **1:** First SYNC1 pulse is pending	RO	RO	0b
0	**SYNC0 Activation State** **0:** First SYNC0 pulse is not pending **1:** First SYNC0 pulse is pending	RO	RO	0b

11.16.78 SYNC0 STATUS REGISTER

Offset: 098Eh Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:1	**RESERVED**	RO	RO	0000000b
0	**SYNC0 State for Acknowledge Mode** SYNC0, in Acknowledge Mode, is cleared by reading this register from PDI. Use only in Acknowledge Mode.	RO	RO	0b

11.16.79 SYNC1 STATUS REGISTER

Offset: 098Fh Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:1	**RESERVED**	RO	RO	0000000b
0	**SYNC1 State for Acknowledge Mode** SYNC1, in Acknowledge Mode, is cleared by reading this register from PDI. Use only in Acknowledge Mode.	RO	RO	0b

11.16.80 START TIME CYCLIC OPERATION REGISTER

Offset: 0990h-0997h Size: 64 bits

Bits	Description	ECAT Type	PDI Type	Default
63:0	Write: Start time (System Time) of cyclic operation in ns. Read: System time of next SYNC0 pulse in ns.	R/W	R/W	00000000h 00000000h

Note: Writes to this register depend on the Sync Out Unit Control bit of the Cyclic Unit Control Register. It is only writable if Sync Out Unit Control is 0.

Note: When the Auto-activation bit of the Activation Register is 1: The upper 32 bits are automatically extended if only the lower 32 bits are written within one frame.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.81 NEXT SYNC1 PULSE REGISTER

Offset: 0998h-099Fh Size: 64 bits

Bits	Description	ECAT Type	PDI Type	Default
63:0	System time of next SYNC1 pulse in ns.	RO	RO	00000000h 00000000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.82 SYNC0 CYCLE TIME REGISTER

Offset: 09A0h-09A3h Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	Time between two consecutive SYNC0 pulses in ns. A value of 0 indicates Single shot mode - generate only one SYNC0 pulse.	R/W	R/W	00000000h

Note: Writes to this register depend on the Sync Out Unit Control bit of the Cyclic Unit Control Register.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.83 SYNC1 CYCLE TIME REGISTER

Offset: 09A4h-09A7h Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	Time between SYNC1 pulses and SYNC0 pulse in ns.	R/W	R/W	00000000h

Note: Writes to this register depend on the Sync Out Unit Control bit of the Cyclic Unit Control Register.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.84 LATCH0 CONTROL REGISTER

Offset: 09A8h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:2	**RESERVED** Write as 0.	RO	RO	000000b
1	**LATCH0 Negative Edge** **0:** Continuous Latch active **1:** Single Event (only first event active)	R/W	R/W	0b
0	**LATCH0 Positive Edge** **0:** Continuous Latch active **1:** Single Event (only first event active)	R/W	R/W	0b

Note: Writes to this register depend on the Latch In Unit 0 bit of the Cyclic Unit Control Register.

11.16.85 LATCH1 CONTROL REGISTER

Offset: 09A9h Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:2	**RESERVED** Write as 0.	RO	RO	000000b
1	**LATCH1 Negative Edge** **0:** Continuous Latch active **1:** Single Event (only first event active)	R/W	R/W	0b
0	**LATCH1 Positive Edge** **0:** Continuous Latch active **1:** Single Event (only first event active)	R/W	R/W	0b

Note: Writes to this register depend on the Latch In Unit 1 bit of the Cyclic Unit Control Register.

11.16.86 LATCH0 STATUS REGISTER

Offset: 09AEh Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:3	**RESERVED** Write as 0.	RO	RO	00000b
2	**LATCH0 Pin State**	RO	RO	0b
1	**Event LATCH0 Negative Edge** **0:** Negative edge not detected or continuous mode **1:** Negative edge detected in single event mode only. **Note:** Flag cleared by reading the LATCH0 Time Negative Edge Register.	RO	RO	0b
0	**Event LATCH0 Positive Edge** **0:** Positive edge not detected or continuous mode **1:** Positive edge detected in single event mode only. **Note:** Flag cleared by reading the LATCH0 Time Positive Edge Register.	RO	RO	0b

11.16.87 LATCH1 STATUS REGISTER

Offset: 09AFh Size: 8 bits

Bits	Description	ECAT Type	PDI Type	Default
7:3	**RESERVED** Write as 0.	RO	RO	00000b
2	**LATCH1 Pin State**	RO	RO	0b
1	**Event LATCH1 Negative Edge** **0:** Negative edge not detected or continuous mode **1:** Negative edge detected in single event mode only. **Note:** Flag cleared by reading the LATCH1 Time Negative Edge Register.	RO	RO	0b
0	**Event LATCH1 Positive Edge** **0:** Positive edge not detected or continuous mode **1:** Positive edge detected in single event mode only. **Note:** Flag cleared by reading the LATCH1 Time Positive Edge Register.	RO	RO	0b

11.16.88 LATCH0 TIME POSITIVE EDGE REGISTER

Offset: 09B0h-09B7h Size: 64 bits

Bits	Description	ECAT Type	PDI Type	Default
63:0	This register captures the System Time at the positive edge of the LATCH0 signal. **Note:** Reading this register clears the Event LATCH0 Positive Edge bit of the LATCH0 Status Register	RO	RO	00000000h 00000000h

Note: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Clearing the Event LATCH0 Positive Edge bit of the LATCH0 Status Register depends upon setting of the Latch In Unit 0 bit of the Cyclic Unit Control Register.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.89 LATCH0 TIME NEGATIVE EDGE REGISTER

Offset: 09B8h-09BFh Size: 64 bits

Bits	Description	ECAT Type	PDI Type	Default
63:0	This register captures the System Time at the negative edge of the LACTH0 signal. **Note:** Reading this register clears the Event LATCH0 Negative Edge bit of the LATCH0 Status Register	RO	RO	00000000h 00000000h

Note: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Clearing the Event LATCH0 Negative Edge bit of the LATCH0 Status Register depends upon setting of the Latch In Unit 0 bit of the Cyclic Unit Control Register.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.90 LATCH1 TIME POSITIVE EDGE REGISTER

Offset: 09C0h-09C7h Size: 64 bits

Bits	Description	ECAT Type	PDI Type	Default
63:0	This register captures the System Time at the positive edge of the LATCH1 signal. **Note:** Reading this register clears the Event LATCH1 Positive Edge bit of the LATCH1 Status Register	RO	RO	00000000h 00000000h

Note: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Clearing the Event LATCH1 Positive Edge bit of the LATCH1 Status Register depends upon setting of the Latch In Unit 1 bit of the Cyclic Unit Control Register.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.91 LATCH1 TIME NEGATIVE EDGE REGISTER

Offset: 09C8h-09CFh Size: 64 bits

Bits	Description	ECAT Type	PDI Type	Default
63:0	This register captures the System Time at the negative edge of the LATCH1 signal. **Note:** Reading this register clears the Event LATCH1 Negative Edge bit of the LATCH1 Status Register	RO	RO	00000000h 00000000h

Note: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Clearing the Event LATCH1 Negative Edge bit of the LATCH1 Status Register depends upon setting of the Latch In Unit 1 bit of the Cyclic Unit Control Register.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.92 ETHERCAT BUFFER CHANGE EVENT TIME REGISTER

Offset: 09F0h-09F3h Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	This register captures the local time of the beginning of the frame which causes at least one SyncManager to assert an ECAT event.	RO	RO	00000000h

Note: Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.93 PDI BUFFER START TIME EVENT REGISTER

Offset: 09F8h-09FBh Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	This register captures the local time when at least one SyncManager asserts a PDI buffer start event.	RO	RO	00000000h

Note: Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.94 PDI BUFFER CHANGE EVENT TIME REGISTER

Offset: 09FCh-09FFh Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	This register captures the local time when at least one SyncManager asserts a PDI buffer change event.	RO	RO	00000000h

Note: Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.95 PRODUCT ID REGISTER

Offset: 0E00h-0E07h Size: 64 bits

Bits	Description	ECAT Type	PDI Type	Default
63:0	**Product ID**	RO	RO	0000h ***ssss***h 9254h ***rrrr***h Note 44

Note 44: The value of "***rrrr***" is the current silicon revision. The value of "***ssss***" is a dependent on the following configuration strap values: 0, **XTAL_MODE**, **CLK_25_EN**, **100FD_B**, **MII_LINKPOL**, **TX_SHIFT[1:0]**, **100FD_A**, **EE_EMUL[2:0]**, **EE_EMUL_ALELO_POL**, **E2PSIZE**, NAND of **EE_EMUL1** and **EE_EMUL2** (if either are low, this bit is one), **CHIPMODE[1:0]**.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.96 VENDOR ID REGISTER

Offset: 0E08h-0E0Fh Size: 64 bits

Bits	Description	ECAT Type	PDI Type	Default
63:32	**RESERVED**	RO	RO	00000000h
31:0	**Vendor ID**	RO	RO	000004D8h (Microchip)

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.97 DIGITAL I/O OUTPUT DATA REGISTER

Offset: 0F00h-0F03h Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	**Output Data**	R/W	RO	0000h

Note: This register is bit-writable (using logical addressing).

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.98 GENERAL PURPOSE OUTPUT REGISTER

Offset: 0F10h-0F11h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**General Purpose Output Data**	R/W	R/W	0000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.99 GENERAL PURPOSE INPUT REGISTER

Offset: 0F18h-0F19h Size: 16 bits

Bits	Description	ECAT Type	PDI Type	Default
15:0	**General Purpose Input Data**	RO	RO	0000h

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.100 USER RAM

Offset: 0F80h-0FFFh Size: 128 Bytes

Bits	Description	ECAT Type	PDI Type	Default
-	**User RAM (128 Bytes)**	R/W	R/W	Undefined

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.101 DIGITAL I/O INPUT DATA REGISTER

Offset: 1000h-1003h Size: 32 bits

Bits	Description	ECAT Type	PDI Type	Default
31:0	**Input Data**	R/W	R/W	Undefined

Note: This register is part of the Process RAM address space. The Process RAM is also directly addressable via the EtherCAT Process RAM Read Data FIFO (ECAT_PRAM_RD_DATA) and EtherCAT Process RAM Write Data FIFO (ECAT_PRAM_WR_DATA).

Note: Process Data RAM is only accessible if EEPROM was correctly loaded (PDI Operational/EEPROM Loaded Correctly bit of ESC DL Status Register = 1)

Note: Digital I/O Input Data is written into the Process Data RAM at these addresses if a Digital I/O PDI with inputs is configured.

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

11.16.102 PROCESS DATA RAM

Offset: 1000h-2FFFh Size: 8 KBytes

Bits	Description	ECAT Type	PDI Type	Default
-	**Process Data RAM (4 KBytes)**	R/W	R/W	Undefined

Note: Process Data RAM is only accessible if EEPROM was correctly loaded (PDI Operational/EEPROM Loaded Correctly bit of ESC DL Status Register = 1)

Note: For EtherCAT Core CSR registers longer than one byte, the LSB has the lowest address and the MSB the highest address.

12.0 EEPROM INTERFACE

The device contains an I2C master controller, which uses the **EESCL** and **EESDA** pins. **EESCL** and **EESDA** require an external pull-up resistor. Both 1 byte and 2 byte addressed EEPROMs are supported. The size is determined by the **E2PSIZE** configuration strap.

12.1 I2C Interface Timing Requirements

This section specifies the I2C master interface input and output timings. The I2C master interface runs in fast-mode with a rate of 148.8 kHz.

FIGURE 12-1: I2C MASTER TIMING DIAGRAM

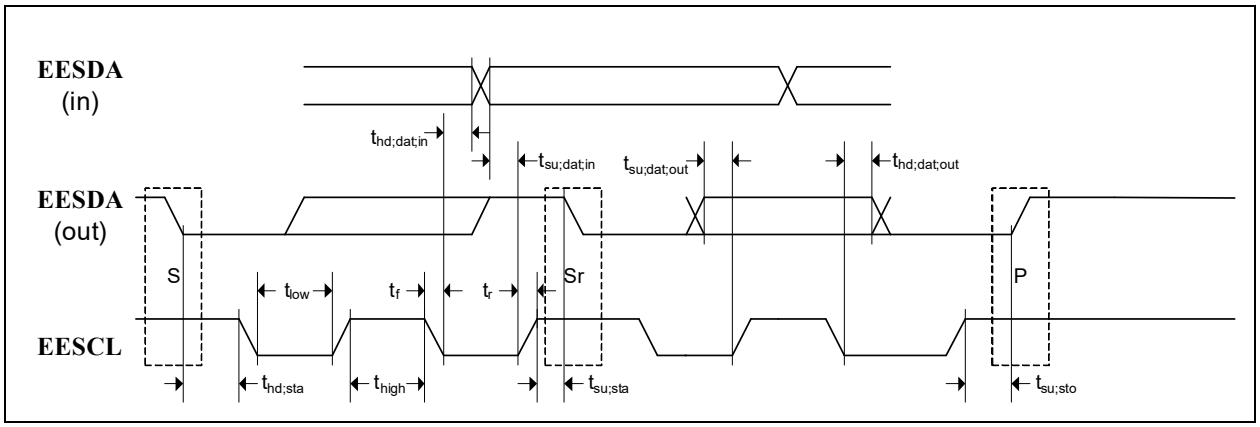


TABLE 12-1: I2C MASTER TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
f_{scl}	**EESCL** clock frequency	-	148.8	-	kHz
t_{high}	**EESCL** high time	3.0	-	-	µs
t_{low}	**EESCL** low time	3.0	-	-	µs
t_r	Rise time of **EESDA** and **EESCL**		-	300	ns
t_f	Fall time of **EESDA** and **EESCL**		-	300	ns
$t_{su;sta}$	Setup time (provided to slave) of **EESCL** high before **EESDA** output falling for repeated start condition	1000 Note 45	-	-	ns
$t_{hd;sta}$	Hold time (provided to slave) of **EESCL** after **EESDA** output falling for start or repeated start condition	1000 Note 45	-	-	ns
$t_{su;dat;in}$	Setup time (from slave) **EESDA** input before **EESCL** rising	200 Note 46	-	-	ns
$t_{hd;dat;in}$	Hold time (from slave) of **EESDA** input after **EESCL** falling	0	-	-	ns
$t_{su;dat;out}$	Setup time (provided to slave) **EESDA** output before **EESCL** rising	400 Note 47	-	-	ns
$t_{hd;dat;out}$	Hold time (provided to slave) of **EESDA** output after **EESCL** falling	400 Note 47	-	-	ns
$t_{su;sto}$	Setup time (provided to slave) of **EESCL** high before **EESDA** output rising for stop condition	1000 Note 45	-	-	ns

Note 45: These values provide 400 ns of margin compared to the I2C fast-mode specification.

Note 46: This value provides ~2100 ns of margin compared to the I2C fast-mode specification.

Note 47: These values provide 300 ns of setup margin and 400 ns of hold margin compared to the I2C fast-mode specification.

12.2 EEPROM Emulation

To reduced system cost and complexity, the device supports EEPROM Emulation.

The EEPROM Emulation mode is used in ESCs with a non-volatile memory (NVRAM) attached to a Microcontroller. The ESC configuration and the device description can be stored in the NVRAM of the Microcontroller, The Microcontroller performs the read and write actions to the NVRAM upon request of the ESC.

From the EtherCAT master's point of view, there is no difference between EEPROM Emulation mode and I^2C EEPROM the master just issues EEPROM commands and waits until the EEPROM interface is no longer busy.

In EEPROM Emulation mode, the ESC issues an interrupt to the Microcontroller if an EEPROM command is pending and it automatically sets the busy bit 0x0502[15]. While the busy bit is set, the Microcontroller can read out the command and the EEPROM address. For a write access, write data is present in the data register. For a read command, read data has to be stored in the data register by the Microcontroller.

Once the Microcontroller has finished reading/writing the EEPROM data register, it acknowledges the command by writing to the EEPROM command register bits. The Microcontroller has to write the command value it has executed into the EEPROM command register. Errors can be indicated using two of the error bits. After acknowledging the command, the EEPROM busy bit 0x0502[15] is automatically cleared, and the interrupt is released.

It is acceptable for the Microcontroller to use a cached image of the EEPROM content, to reduce access time and write cycles to the NVRAM. The NVRAM should be updated shortly after write accesses from the master are finished (e.g., after a timeout). A specific write sequence from the master should not be relied upon.

During EEPROM Emulation, a configuration reload will fetch all 16 bytes (8 bytes at a time) formatted as they would be from the ESC Configuration Area in an actual EEPROM, including a valid CRC.

EEPROM Emulation mode is enabled by externally pulling either the **EE_EMUL2** or **EE_EMUL1** configuration strap low. EEPROM Emulation mode is indicated by the EEPROM Emulation bit in the EEPROM Control/Status Register.

Note:	For normal EEPROM operation, the **EE_EMUL2** or **EE_EMUL1** configuration straps should include an external pull-up.

13.0 CHIP MODE CONFIGURATION

The mode of the chip is controlled by the **CHIP_MODE[1:0]** configuration straps as follows:

TABLE 13-1: CHIP MODE SELECTION

CHIP_MODE[1:0]	Mode
0x	2 port mode. Port 0 = PHY A, Port 1 = PHY B
10	3 port downstream mode. Port 0 = PHY A, Port 1 = PHY B, Port 2 = MII
11	3 port upstream mode. Port 0 = MII, Port 1 = PHY B, Port 2 = PHY A

Once the mode of the chip is selected, the Process Data Interface (PDI) in use is selected by the Process Data Interface (PDI_SELECT) field of the PDI Control Register. The valid choices are as follows:

TABLE 13-2: PDI MODE SELECTION

PDI_SELECT	PDI MODE	ETHERCAT DIRECT MAPPED MODE
0x04	DIG I/O	N/A
0x05	Beckhoff SPI	Yes
0x80	SPI	No
0x82	SPI	Yes
0x88	HBI Multiplexed 1 Phase 8-bit	No
0x89	HBI Multiplexed 1 Phase 16-bit	No
0x8A	HBI Multiplexed 2 Phase 8-bit	No
0x8B	HBI Multiplexed 2 Phase 16-bit	No
0x8C	HBI Indexed 8-bit	No
0x8D	HBI Indexed 16-bit	No
0x8E	HBI Demultiplexed 8-bit	No
0x8F	HBI Demultiplexed 16-bit	No
0x90	HBI Multiplexed 1 Phase 8-bit	Yes
0x91	HBI Multiplexed 1 Phase 16-bit	Yes
0x92	HBI Multiplexed 2 Phase 8-bit	Yes
0x93	HBI Multiplexed 2 Phase 16-bit	Yes
0x94	HBI Indexed 8-bit	Yes
0x95	HBI Indexed 16-bit	Yes
0x96	HBI Demultiplexed 8-bit	Yes
0x97	HBI Demultiplexed 16-bit	Yes
Others	RESERVED	-

Note: The mode of the chip as selected by the **CHIP_MODE[1:0]** configuration straps is not affected by the PDI selection.

Note: Due to pin sharing, when the device is in 3 port mode, the only usable interface is SPI.

13.1 HBI Sub-Configuration

The PDI Configuration Register and the Extended PDI Configuration Register are used for the HBI configuration straps as shown in Table 11-4, "EtherCAT Core EEPROM Configurable Registers". The PDI Configuration Register and the Extended PDI Configuration Register are initialized from the contents of the EEPROM.

13.2 SPI Sub-Configuration

The PDI Configuration Register is used for the SPI configuration straps as shown in Table 11-4, "EtherCAT Core EEPROM Configurable Registers". The PDI Configuration Register is initialized from the contents of the EEPROM.

13.3 Beckhoff SPI Sub-Configuration

The PDI Configuration Register is used for the SPI configuration straps as shown in Table 11-4, "EtherCAT Core EEPROM Configurable Registers". The PDI Configuration Registeris initialized from the contents of the EEPROM.

13.4 EEPROM Emulation Mode Device Configuration

Normally, before the ESC Configuration Area has been read and the device configured, all host interfaces (HBI and SPI) are disabled, since much of the device is configured by the EEPROM contents.

When EEPROM Emulation mode is used, a default host interface configuration / sub-configuration must be selected in order to allow the host microprocessor to access the EEPROM registers.

The following settings are used during EEPROM Emulation mode until the device has been configured:

- PDI Control Register
  The **EE_EMUL[2:0]** configuration straps select the Process Data Interface (PDI_SELECT) as shown in Table 13-3.

Note: In general, the PDI type that is provide by the host as part of the emulated EEPROM data should be the same value as the default PDI in use. In other words, the PDI type should not be changed. Changes between EtherCAT Direct Mapped mode and non-EtherCAT Direct Mapped mode within the same PDI type are allowed.

TABLE 13-3: EEPROM EMULATION PDI_SELECT

EE_EMUL[2:0]	PDI_SELECT
000	0x80 (SPI)
001	0x97 (HBI Demultiplexed 16-bit EtherCAT Direct Mapped)
010	0x91 (HBI Multiplexed 1 Phase 16-bit EtherCAT Direct Mapped)
011	0x93 (HBI Multiplexed 2 Phase 16-bit EtherCAT Direct Mapped)
100	0x82 (SPI EtherCAT Direct Mode)

TABLE 13-3: EEPROM EMULATION PDI_SELECT

EE_EMUL[2:0]	PDI_SELECT
101	0x05 (Beckhoff SPI Mode)
110	N/A (EEPROM is enabled)
111	N/A (EEPROM is enabled)

- PDI Configuration Register and Extended PDI Configuration Register
  Once the Host interface is selected, the sub-configuration defaults as shown in Table 13-4.

Note: These defaults are chosen for maximum backwards compatibility that meets the minimum functionality to load the configuration data or are a don't care. Other than the SPI SCS# polarity and the HBI CS polarity, these can be changed if needed when the device is configured. The new device configuration takes effect following the last EEPROM emulation data write. If any parameters are changed from the initial default, the SPI or HBI bus signals must change to match these parameters while the SPI SCS# or the HBI CS are inactive.

Note: The WAIT_ACK pin is not initially enabled. HBI and SPI EtherCAT Direct Mapped modes must pace the bus cycles to meet the worst case access timing.

Note: To allow the updated device configuration to take affect at the completion of EEPROM Emulation, following the last EEPROM emulation data write the host should wait at least 1.5uS before making further accesses.

TABLE 13-4: EEPROM EMULATION PDI DEFAULTS

Register	Bits	Value
PDI Configuration Register (SPI mode)	7:2 - RESERVED	00h
	2 - SPI AL Event Request and INT_STS enable	0b
	1 - SPI WAIT_ACK polarity	0b
	0 - SPI WAIT_ACK buffer type	0b
PDI Configuration Register (Beckhoff SPI mode)	7:6 - RESERVED	00b
	5 - Data Out sample mode:	EE_EMUL_SPI3
	4 - SCS# polarity	EE_EMUL_SPI2
	3:2 - RESERVED	00b
	1:0 - SPI mode	EE_EMUL_SPI[1:0]

TABLE 13-4: EEPROM EMULATION PDI DEFAULTS (CONTINUED)

Register	Bits	Value
PDI Configuration Register (HBI modes)	7 - HBI ALE qualification	0b
	6 - HBI RW mode	0b
	5 - HBI CS polarity	0b
	4 - HBI RD RD/WR polarity	0b
	3 - HBI WR EN polarity	0b
	2 - HBI ALE polarity	If **EE_EMUL[2:0]** = 10 (HBI Multiplexed 1 Phase) then **EE_EMUL_ALELO_POL**, else 0b
	1 - HBI WAIT_ACK polarity	0b
	0 - HBI WAIT_ACK buffer type	0b
Extended PDI Configuration Register (HBI modes)	15:3 - RESERVED	00h
	2 - HBI BE1/0 polarity	0b
	1 - Perform internal write	0b
	0 - Read WAIT_ACK removal delay	0b

14.0 GENERAL PURPOSE TIMER & FREE-RUNNING CLOCK

This chapter details the General Purpose Timer (GPT) and the Free-Running Clock.

14.1 General Purpose Timer

The device provides a 16-bit programmable General Purpose Timer that can be used to generate periodic system interrupts. The resolution of this timer is 100 µs.

The GPT loads the General Purpose Timer Count Register (GPT_CNT) with the value in the General Purpose Timer Pre-Load (GPT_LOAD) field of the General Purpose Timer Configuration Register (GPT_CFG) when the General Purpose Timer Enable (TIMER_EN) bit of the General Purpose Timer Configuration Register (GPT_CFG) is asserted (1). On a chip-level reset or when the General Purpose Timer Enable (TIMER_EN) bit changes from asserted (1) to de-asserted (0), the General Purpose Timer Pre-Load (GPT_LOAD) field is initialized to FFFFh. The General Purpose Timer Count Register (GPT_CNT) is also initialized to FFFFh on reset.

Once enabled, the GPT counts down until it reaches 0000h. At 0000h, the counter wraps around to FFFFh, asserts the GP Timer (GPT_INT) interrupt status bit in the Interrupt Status Register (INT_STS), asserts the IRQ interrupt (if GP Timer Interrupt Enable (GPT_INT_EN) is set in the Interrupt Enable Register (INT_EN)) and continues counting. GP Timer (GPT_INT) is a sticky bit. Once this bit is asserted, it can only be cleared by writing a 1 to the bit. Refer to Section 7.2.3, "General Purpose Timer Interrupt," on page 62 for additional information on the GPT interrupt.

Software can write a pre-load value into the General Purpose Timer Pre-Load (GPT_LOAD) field at any time (e.g., before or after the General Purpose Timer Enable (TIMER_EN) bit is asserted). The General Purpose Timer Count Register (GPT_CNT) will immediately be set to the new value and continue to count down (if enabled) from that value.

14.2 Free-Running Clock

The Free-Running Clock (FRC) is a simple 32-bit up-counter that operates from a fixed 25 MHz clock. The current FRC value can be read via the Free Running 25MHz Counter Register (FREE_RUN). On assertion of a chip-level reset, this counter is cleared to zero. On de-assertion of a reset, the counter is incremented once for every 25 MHz clock cycle. When the maximum count has been reached, the counter rolls over to zeros. The FRC does not generate interrupts.

Note: The free running counter can take up to 160 ns to clear after a reset event.

14.3 General Purpose Timer and Free-Running Clock Registers

This section details the directly addressable general purpose timer and free-running clock related System CSRs. For an overview of the entire directly addressable register map, refer to Section 5.0, "Register Map," on page 40.

TABLE 14-1: GENERAL PURPOSE TIMER AND FREE-RUNNING CLOCK REGISTERS

ADDRESS	EtherCAT Direct Mapped Mode	Register Name (SYMBOL)
08Ch	308Ch	General Purpose Timer Configuration Register (GPT_CFG)
090h	3090h	General Purpose Timer Count Register (GPT_CNT)
09Ch	309Ch	Free Running 25MHz Counter Register (FREE_RUN)

14.3.1 GENERAL PURPOSE TIMER CONFIGURATION REGISTER (GPT_CFG)

Offset: 08Ch / 308Ch Size: 32 bits

This read/write register configures the device's General Purpose Timer (GPT). The GPT can be configured to generate host interrupts at the interval defined in this register. The current value of the GPT can be monitored via the General Purpose Timer Count Register (GPT_CNT). Refer to Section 14.1, "General Purpose Timer," on page 394 for additional information.

Bits	Description	Type	Default
31:30	**RESERVED**	RO	-
29	**General Purpose Timer Enable (TIMER_EN)** This bit enables the GPT. When set, the GPT enters the run state. When cleared, the GPT is halted. On the 1 to 0 transition of this bit, the GPT_LOAD field of this register will be preset to FFFFh. **0: GPT Disabled** **1: GPT Enabled**	R/W	0b
28:16	**RESERVED**	RO	-
15:0	**General Purpose Timer Pre-Load (GPT_LOAD)** This value is pre-loaded into the GPT. This is the starting value of the GPT. The timer will begin decrementing from this value when enabled.	R/W	FFFFh

14.3.2 GENERAL PURPOSE TIMER COUNT REGISTER (GPT_CNT)

Offset: 090h / 3090h Size: 32 bits

This read-only register reflects the current general purpose timer (GPT) value. The register should be used in conjunction with the General Purpose Timer Configuration Register (GPT_CFG) to configure and monitor the GPT. Refer to Section 14.1, "General Purpose Timer," on page 394 for additional information.

Bits	Description	Type	Default
31:16	**RESERVED**	RO	-
15:0	**General Purpose Timer Current Count (GPT_CNT)** This 16-bit field represents the current value of the GPT.	RO	FFFFh

14.3.3 FREE RUNNING 25MHZ COUNTER REGISTER (FREE_RUN)

Offset: 09Ch / 309Ch Size: 32 bits

This read-only register reflects the current value of the free-running 25MHz counter. Refer to Section 14.2, "Free-Running Clock," on page 394 for additional information.

Bits	Description	Type	Default
31:0	**Free Running Counter (FR_CNT)** This field reflects the current value of the free-running 32-bit counter. At reset, the counter starts at zero and is incremented by one every 25 MHz cycle. When the maximum count has been reached, the counter will rollover to zero and continue counting. **Note:** The free running counter can take up to 160nS to clear after a reset event.	RO	00000000h

15.0 MISCELLANEOUS

This chapter describes miscellaneous functions and registers that are present in the device.

15.1 Miscellaneous System Configuration & Status Registers

This section details the remainder of the directly addressable System CSRs. These registers allow for monitoring and configuration of various device functions such as the Chip ID/revision, byte order testing, and hardware configuration.

For an overview of the entire directly addressable register map, refer to Section 5.0, "Register Map," on page 40.

TABLE 15-1: MISCELLANEOUS REGISTERS

ADDRESS	EtherCAT Direct Mapped Mode	Register Name (SYMBOL)
050h	3050h	Chip ID and Revision (ID_REV)
064h	3064h	Byte Order Test Register (BYTE_TEST)
074h	3074h	Hardware Configuration Register (HW_CFG)

15.1.1 CHIP ID AND REVISION (ID_REV)

Offset: 050h / 3050h Size: 32 bits

This read-only register contains the ID and Revision fields for the device.

Bits	Description	Type	Default
31:16	**Chip ID** This field indicates the chip ID.	RO	9254
15:0	**Chip Revision** This field indicates the design revision.	RO	Note 1

Note 1: Default value is dependent on device revision.

15.1.2 BYTE ORDER TEST REGISTER (BYTE_TEST)

Offset: 064h / 3064h Size: 32 bits

This read-only register can be used to determine the byte ordering of the current configuration. Byte ordering is a function of the host data bus width and endianess. Refer to Section 8.0, "Host Bus Interface," on page 68 for additional information on byte ordering.

The BYTE_TEST register can optionally be used as a dummy read register when assuring minimum write-to-read or read-to-read timing. Refer to Section 8.0, "Host Bus Interface," on page 68 for additional information.

For host interfaces that are disabled during the reset state, the BYTE_TEST register can be used to determine when the device has exited the reset state.

Note: This register can be read while the device is in the reset or not ready / power savings states without leaving the host interface in an intermediate state. If the host interface is in a reset state, returned data may be invalid. However, during reset, the returned data will not match the normal valid data pattern.

Note: It is not necessary to read all four BYTEs of this register. DWORD access rules do not apply to this register.

Bits	Description	Type	Default
31:0	**Byte Test (BYTE_TEST)** This field reflects the current byte ordering	RO	87654321h

15.1.3 HARDWARE CONFIGURATION REGISTER (HW_CFG)

Offset: 074h / 3074h Size: 32 bits

This register allows the configuration of various hardware features.

Note: This register can be read while the device is in the reset or not ready / power savings states without leaving the host interface in an intermediate state. If the host interface is in a reset state, returned data may be invalid.

Note: It is not necessary to read all four BYTEs of this register. DWORD access rules do not apply to this register.

Bits	Description	Type	Default
31:28	**RESERVED**	RO	-
27	**Device Ready (READY)** When set, this bit indicates that the device is ready to be accessed. Upon power-up, RST# reset, return from power savings states, EtherCAT chip level or module level reset, or digital reset, the host processor may interrogate this field as an indication that the device has stabilized and is fully active. This rising edge of this bit will assert the Device Ready (READY) bit in the Interrupt Status Register (INT_STS) and can cause an interrupt if enabled. **Note:** With the exception of the HW_CFG, PMT_CTRL, BYTE_TEST, and RESET_CTL registers, read access to any internal resources is forbidden while the READY bit is cleared. Writes to any address are invalid until this bit is set. **Note:** This bit is identical to bit 0 of the Power Management Control Register (PMT_CTRL).	RO	0b
26	**AMDIX Disable PHY B** When set, this bit disables the Auto-MDIX function and selects a non-crossed over configuration. **Note:** Auto-MDIX operation can be further overridden via the PHY x Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND_x)	R/W	Note 2
25	**AMDIX Disable PHY A** When set, this bit disables the Auto-MDIX function and selects a non-crossed over configuration. **Note:** Auto-MDIX operation can be further overridden via the PHY x Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND_x).	R/W	Note 3
24	**ANEG Disable PHY B** When set, this bit: • Sets the default mode of operation to fixed 100Mbps full-duplex. • Modifies the MI Link Detection and Configuration operation (if enabled) to check for and enforce a fixed 100Mbps full-duplex link. • Modifies the Enhanced Link Detection operation (if enabled) to reset the PHY in case of errors instead of restarting auto-negotiation. **Note:** When MI Link Detection and Configuration and Enhanced Link Detection operations are both disabled, the mode of operation can be changed via the standard PHY registers.	R/W	Note 2

Bits	Description	Type	Default
23	**ANEG Disable PHY A** When set, this bit: • Sets the default mode of operation to fixed 100Mbps full-duplex. • Modifies the MI Link Detection and Configuration operation (if enabled) to check for and enforce a fixed 100Mbps full-duplex link. • Modifies the Enhanced Link Detection operation (if enabled) to reset the PHY in case of errors instead of restarting auto-negotiation. **Note:** When MI Link Detection and Configuration and Enhanced Link Detection operations are both disabled, the mode of operation can be changed via the standard PHY registers.	R/W	Note 3
22:5	**RESERVED**	RO	-
4:0	**LED Polarity** When cleared to 0, the associated LED pin is active low. When set to 1, the associated LED pin is active high. Bit 0: **LINKACTLED0** Bit 1: **LINKACTLED1** Bit 2: **LINKACTLED2** Bit 3: **RUNLED / STATE_RUNLED** Bit 4: **ERRLED** **Note:** The polarity of these bits are opposite that of the configuration straps. **Note:** These bits do not affect the functional straps that may share the strap pins.	R/W	Note 4

Note 2: The default value of this field is determined by the **100FD_B** configuration strap. See Section 3.3, "Configuration Straps," on page 36 for more information.

Note 3: The default value of this field is determined by the **100FD_A** configuration strap. See Section 3.3, "Configuration Straps," on page 36 for more information.

Note 4: The default value of this field is determined by the ***inverse*** of the **LEDPOL[4:0]** configuration strap. See Section 3.3, "Configuration Straps," on page 36 for more information.

16.0 JTAG

16.1 JTAG

A IEEE 1149.1 compliant TAP Controller supports boundary scan and various test modes.

The device includes an integrated JTAG boundary-scan test port for board-level testing. The interface consists of four pins (**TDO**, **TDI**, **TCK** and **TMS**) and includes a state machine, data register array, and an instruction register. The JTAG pins are described in Table 3-13, "JTAG Pins," on page 35. The JTAG interface conforms to the IEEE Standard 1149.1 - 2001 *Standard Test Access Port (TAP) and Boundary-Scan Architecture*.

All input and output data is synchronous to the **TCK** test clock input. TAP input signals **TMS** and **TDI** are clocked into the test logic on the rising edge of **TCK**, while the output signal **TDO** is clocked on the falling edge.

JTAG pins are multiplexed with the GPIO/LED and EEPROM pins. The JTAG functionality is selected when the **TEST-MODE** pin is asserted.

The implemented IEEE 1149.1 instructions and their op codes are shown in Table 16-1.

TABLE 16-1: IEEE 1149.1 OP CODES

INSTRUCTION	OP CODE	COMMENT
BYPASS 0	16'h0000	Mandatory Instruction
BYPASS 1	16'hFFFF	Mandatory Instruction
SAMPLE/PRELOAD	16'hFFF8	Mandatory Instruction
EXTEST	16'hFFE8	Mandatory Instruction
CLAMP	16'hFFEF	Optional Instruction
ID_CODE	16'hFFFE	Optional Instruction
HIGHZ	16'hFFCF	Optional Instruction

Note: The JTAG device ID is 002E1445h

Note: All digital I/O pins support IEEE 1149.1 operation. Analog pins and the **OSCI** / **OSCO** pins do not support IEEE 1149.1 operation.

16.1.1 JTAG TIMING REQUIREMENTS

This section specifies the JTAG timing of the device.

FIGURE 16-1: JTAG TIMING

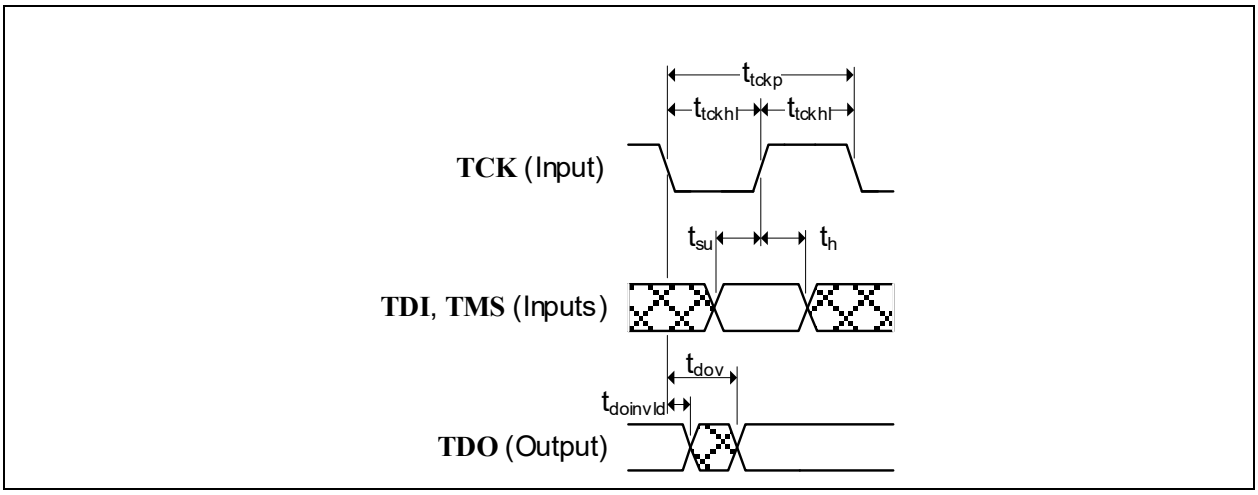


TABLE 16-2: JTAG TIMING VALUES

Symbol	Description	Min	Max	Units	Notes
t_{tckp}	**TCK** clock period	40		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	5		ns	
t_h	**TDI**, **TMS** hold from **TCK** rising edge	5		ns	
t_{dov}	**TDO** output valid from **TCK** falling edge		15	ns	
$t_{doinvld}$	**TDO** output invalid from **TCK** falling edge	0		ns	

Note: Timing values are with respect to an equivalent test load of 25 pF.

17.0 OPERATIONAL CHARACTERISTICS

17.1 Absolute Maximum Ratings*

Supply Voltage (VDD12TX1, VDD12TX2, OSCVDD12, VDDCR) (Note 1) . . . . . 0 V to +1.5 V

Supply Voltage (VDD33TXRX1, VDD33TXRX2, VDD33BIAS, VDD33, VDDIO) (Note 1) . . . . . 0 V to +3.6 V

Ethernet Magnetics Supply Voltage . . . . . -0.5 V to +3.6 V

Positive voltage on signal pins, with respect to ground (Note 2). . . . . **VDDIO** + 2.0 V

Negative voltage on signal pins, with respect to ground (Note 3). . . . . -0.5 V

Positive voltage on OSCI, with respect to ground. . . . . +3.6 V

Storage Temperature . . . . . -55°C to +150°C

Junction Temperature . . . . . +150°C

Lead Temperature Range . . . . . Refer to JEDEC Spec. J-STD-020

HBM ESD Performance . . . . . JEDEC Class 3A

Note 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 to use a clamp circuit.

Note 2: This rating does not apply to the following pins: OSCI, RBIAS

Note 3: This rating does not apply to the following pins: RBIAS

*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 17.2, "Operating Conditions**", Section 17.5, "DC Specifications", or any other applicable section of this specification is not implied. Note, device signals are *NOT* 5 volt tolerant.

17.2 Operating Conditions**

Supply Voltage (VDD12TX1, VDD12TX2, OSCVDD12, VDDCR) . . . . . +1.14 V to +1.26 V

Analog Port Supply Voltage (VDD33TXRX1, VDD33TXRX2, VDD33BIAS, VDD33). . . . . +3.0 V to +3.6 V

I/O Supply Voltage (VDDIO) (Note 1) . . . . . +1.62 V to +3.6 V

Ethernet Magnetics Supply Voltage . . . . . +2.25 V to +3.6 V

Ambient Operating Temperature in Still Air (T_A) . . . . . Note 4

Note 4: 0°C to +70°C for commercial version, -40°C to +85°C for industrial version, -40°C to +105°C for extended industrial version.

Extended industrial temperature range is supported with the following restrictions:
- External regulator required (Internal regulator disabled) and 2.5 V (typ) Ethernet magnetics voltage.

Proper operation of the device is guaranteed only within the ranges specified in this section. After the device has completed power-up, **VDDIO and the magnetics power supply must maintain their voltage level with ±10%. Varying the voltage greater than ±10% after the device has completed power-up can cause errors in device operation.

Note: Do not drive input signals without power supplied to the device.

17.3 Package Thermal Specifications

TABLE 17-1: 80-PIN SQFN PACKAGE THERMAL PARAMETERS

Parameter	Symbol	Value	Units	Comments
Thermal Resistance Junction to Ambient	Θ_{JA}	31.6	°C/W	Measured in still air
		25.7	°C/W	Airflow 1 m/s
		24.1	°C/W	Airflow 2.5 m/s
Thermal Resistance Junction to Bottom of Case	Ψ_{JT}	0.3	°C/W	Measured in still air
Thermal Resistance Junction to Top of Case	Θ_{JC}	16.1	°C/W	-

Note: Thermal parameters are measured or estimated for devices in a multi-layer 2S2P PCB per JESDN51.

TABLE 17-2: MAXIMUM POWER DISSIPATION

Mode	Maximum Power (mW)
Internal Regulator Disabled, 2.5 V Ethernet Magnetics	***TBD***
Internal Regulator Disabled, 3.3 V Ethernet Magnetics	***TBD***
Internal Regulator Enabled, 2.5 V Ethernet Magnetics	***TBD***
Internal Regulator Enabled, 3.3 V Ethernet Magnetics	***TBD***

17.4 Current Consumption and Power Consumption

This section details the device's typical supply current consumption and power dissipation for 100BASE-TX and power management modes of operation with the internal regulator enabled and disabled.

17.4.1 INTERNAL REGULATOR DISABLED

TABLE 17-3: CURRENT CONSUMPTION AND POWER DISSIPATION (REGS. DISABLED)

		3.3 V Device Current (mA) (A) Note 5, Note 7	1.2 V Device Current (mA) (B) Note 6, Note 7	TX Magnetics Current (mA) (C) Note 8	Device Power with 2.5 V Magnetics (mW) Note 9, Note 10	Device Power with 3.3 V Magnetics (mW) Note 9, Note 11
Reset (RST#)	Typ.	***TBD***	***TBD***	***TBD***	***TBD***	***TBD***
D0, 100BASE-TX with Traffic	Typ.	***TBD***	***TBD***	***TBD***	***TBD***	***TBD***
D0, 100BASE-TX Idle	Typ.	***TBD***	***TBD***	***TBD***	***TBD***	***TBD***
D0, PHY Energy Detect Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***	***TBD***
D0, PHY General Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***	***TBD***
D1, 100BASE-TX Idle	Typ.	***TBD***	***TBD***	***TBD***	***TBD***	***TBD***
D1, PHY Energy Detect Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***	***TBD***
D1, PHY General Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***	***TBD***
D2, 100BASE-TX Idle	Typ.	***TBD***	***TBD***	***TBD***	***TBD***	***TBD***
D2, PHY Energy Detect Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***	***TBD***
D2, PHY General Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***	***TBD***
D3, PHY General Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***	***TBD***

Note 5: VDD33TXRX1, VDD33TXRX2, VDD33BIAS, VDD33, VDDIO

Note 6: VDD12TX1, VDD12TX2, OSCVDD12, VDDCR

Note 7: Current measurements do not include power applied to the magnetics or the optional external LEDs.

Note 8: The Ethernet component current is independent of the supply rail voltage (2.5V or 3.3V) of the transformer. Two copper TP operation is assumed.

Note 9: This includes the power dissipated by the transmitter by way of the current through the transformer.

Note 10: 3.3*(A) + 1.2*(B) + (2.5)*(C) @ Typ

Note 11: 3.3*(A) + 1.2*(B) + (3.3)*(C) @ Typ

17.4.2 INTERNAL REGULATOR ENABLED

TABLE 17-4: CURRENT CONSUMPTION AND POWER DISSIPATION (REGS. ENABLED)

		3.3 V Device Current (mA) (A) Note 12, Note 13, Note 14	TX Magnetics Current (mA) (C) Note 15	Device Power with 2.5 V Magnetics (mW) Note 16, Note 17	Device Power with 3.3 V Magnetics (mW) Note 16, Note 18
Reset (RST#)	Typ.	***TBD***	***TBD***	***TBD***	***TBD***
D0, 100BASE-TX with Traffic	Typ.	***TBD***	***TBD***	***TBD***	***TBD***
D0, 100BASE-TX Idle	Typ.	***TBD***	***TBD***	***TBD***	***TBD***
D0, PHY Energy Detect Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***
D0, PHY General Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***
D1, 100BASE-TX Idle	Typ.	***TBD***	***TBD***	***TBD***	***TBD***
D1, PHY Energy Detect Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***
D1, PHY General Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***
D2, 100BASE-TX Idle	Typ.	***TBD***	***TBD***	***TBD***	***TBD***
D2, PHY Energy Detect Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***
D2, PHY General Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***
D3, PHY General Power Down	Typ.	***TBD***	***TBD***	***TBD***	***TBD***

Note 12: VDD33TXRX1, VDD33TXRX2, VDD33BIAS, VDD33, VDDIO

Note 13: VDD12TX1 and VDD12TX2 are driven by the internal regulator via the PCB. The current is accounted for via VDD33.

Note 14: Current measurements do not include power applied to the magnetics or the optional external LEDs.

Note 15: The Ethernet component current is independent of the supply rail voltage (2.5V or 3.3V) of the transformer. Two copper TP operation is assumed.

Note 16: This includes the power dissipated by the transmitter by way of the current through the transformer.

Note 17: 3.3*(A) + (2.5)*(C) @ Typ

Note 18: 3.3*(A) + (3.3)*(C) @ Typ

17.5 DC Specifications

TABLE 17-5: NON-VARIABLE I/O DC ELECTRICAL 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}			5.5	V	
Neg-Going Threshold	V_{ILT}	1.01	1.19	1.39	V	Schmitt
Pos-Going Threshold	V_{IHT}	1.39	1.59	1.79	V	Schmitt
Schmitt Trigger Hysteresis (V_{IHT} - V_{ILT})	V_{HYS}	336	399	459	mV	
Input Leakage (V_{IN} = VSS or VDD33)	I_{IH}	-10		10	µA	Note 19
Input Capacitance	C_{IN}			2	pF	
Pull-Up Impedance (V_{IN} = VSS)	R_{DPU}	52.7	65.9	79.7	KΩ	
Pull-Up Current (V_{IN} = VSS)	I_{DPU}	37.6	50.1	68.3	µA	
Pull-Down Impedance (V_{IN} = VDD33)	R_{DPD}	46.0	66.8	94.3	KΩ	
Pull-Down Current (V_{IN} = VDD33)	I_{DPD}	31.8	49.4	78.3	µA	
AI Type Input Buffer (RXPA/RXNA/RXPB/RXNB)						
Differential Input Level	$V_{IN-DIFF}$	0.1		**VDD33TXRX*x***	V	
Common Mode Voltage	V_{CM}	1.0	**VDD33TXRX*x*-1.3**		V	
Input Capacitance	C_{IN}			5	pF	
AI Type Input Buffer (CLK_25_EN/XTAL_MODE)						
State A Threshold	V_{THA}	-0.3		0.8	V	
State B Threshold	V_{THB}	1.2		1.7	V	
State C Threshold	V_{THC}	2.2		VDD33+0.3	V	
Pin Float Voltage	V_{FLOAT}	1.48		1.84 (Note 20)	V	

TABLE 17-5: NON-VARIABLE I/O DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	Min	Typ	Max	Units	Notes
ICLK Type Input Buffer (OSCI Input)						
Crystal Oscillator Overdrive Mode (Note 21)						
Low Input Level	V_{ILI}	-0.3		0.35	V	
High Input Level	V_{IHI}	OSCVDD12-0.35		3.6	V	
Input Leakage	I_{ILCK}	-10		10	µA	
Schmitt Trigger Input Mode						
Low Input Level	V_{ILI}	-0.3		0.8	V	
High Input Level	V_{IHI}	OSCVDD12-0.8		3.6	V	
Input Hysteresis	V_{HYS}	150			mV	
Input Leakage	I_{ILCK}	-10		10	µA	
O8 Type Buffers						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8 mA
High Output Level	V_{OH}	VDD33-0.4			V	I_{OH} = -8 mA
I/O sinking current	I_{OL}	8.28	12.64	18.42	mA	V_{OUT} = 0.4 V
I/O sinking output impedance	R_{OL}	21.71	31.64	48.32	Ω	V_{OUT} = 0.4 V
I/O sourcing current	I_{OH}	8.08	11.57	15.99	mA	V_{OUT} = VDD33-0.4 V
I/O sourcing output impedance	R_{OH}	25.02	34.57	49.5	Ω	V_{OUT} = VDD33-0.4 V
OD12 Type Buffers						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 12 mA
I/O sinking current	I_{OL}	12.11	18.44	26.72	mA	V_{OUT} = 0.4 V
I/O sinking output impedance	R_{OL}	14.97	21.69	33.02	Ω	V_{OUT} = 0.4 V
OS12 Type Buffers						
High Output Level	V_{OH}	VDD33-0.4			V	I_{OH} = -12 mA
I/O sourcing current	I_{OH}	12.43	17.61	23.99	Ω	V_{OUT} = VDD33-0.4 V
I/O sourcing output impedance	R_{OH}	16.68	22.72	32.17	mA	V_{OUT} = VDD33-0.4 V

Note 19: This specification applies to all inputs and tri-stated bi-directional pins. Internal pull-down and pull-up resistors add +/- 50 µA per-pin (typical).

Note 20: Although V_{FLOAT}(max) could exceed V_{THB}, a float is guaranteed to be detected as State B.

Note 21: OSCI can optionally be driven from a 25 MHz single-ended clock oscillator.

TABLE 17-6: VARIABLE I/O DC ELECTRICAL CHARACTERISTICS

Parameter	Symbol	Min	1.8 V Typ	3.3 V Typ	Max	Units	Notes
VIS Type Input Buffer							
Low Input Level	V_{ILI}	-0.3				V	
High Input Level	V_{IHI}				3.6	V	
Negative-Going Threshold	V_{ILT}	0.64	0.83	1.41	1.76	V	Schmitt trigger
Positive-Going Threshold	V_{IHT}	0.81	0.99	1.65	1.90	V	Schmitt trigger
Schmitt Trigger Hysteresis (V_{IHT} - V_{ILT})	V_{HYS}	102	158	138	288	mV	
Input Leakage (V_{IN} = **VSS** or **VDDIO**)	I_{IH}	-10			10	µA	Note 22
Input Capacitance	C_{IN}				2	pF	
Pull-Up Impedance (V_{IN} = VSS)	R_{DPU}	54	68	67	82	KΩ	
Pull-Up Current (V_{IN} = VSS)	I_{DPU}	20	27	49	67	µA	
Pull-Down Impedance (V_{IN} = VDDIO)	R_{DPD}	54	68	67	85	KΩ	
Pull-Down Current (V_{IN} = VDDIO)	I_{DPD}	19	26	48	66	µA	
VO8 Type Buffers							
Low Output Level	V_{OL}				0.4	V	I_{OL} = 8 mA
High Output Level	V_{OH}	VDDIO - 0.4				V	I_{OH} = -8 mA
VOD8 Type Buffer							
Low Output Level	V_{OL}				0.4	V	I_{OL} = 8 mA
VOS8 Type Buffer							
High Output Level	V_{OH}	VDDIO - 0.4				V	I_{OH} = -8 mA
VO12 Type Buffer							
Low Output Level	V_{OL}				0.4	V	I_{OL} = 12 mA
High Output Level	V_{OH}	VDDIO - 0.4				V	I_{OH} = -12 mA
VOD12 Type Buffer							
Low Output Level	V_{OL}				0.4	V	I_{OL} = 12 mA

TABLE 17-6: VARIABLE I/O DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	Min	1.8 V Typ	3.3 V Typ	Max	Units	Notes
VOS12 Type Buffer							
High Output Level	V_{OH}	VDDIO - 0.4				V	I_{OH} = -12 mA

Note 22: This specification applies to all inputs and tri-stated bi-directional pins. Internal pull-down and pull-up resistors add ±50 µA per-pin (typical).

TABLE 17-7: 100BASE-TX TRANSCEIVER CHARACTERISTICS

Parameter	Symbol	Min	Typ	Max	Units	Notes
Peak Differential Output Voltage High	V_{PPH}	950	-	1050	mVpk	Note 23
Peak Differential Output Voltage Low	V_{PPL}	-950	-	-1050	mVpk	Note 23
Signal Amplitude Symmetry	V_{SS}	98	-	102	%	Note 23
Signal Rise and Fall Time	T_{RF}	3.0	-	5.0	ns	Note 23
Rise and Fall Symmetry	T_{RFS}	-	-	0.5	ns	Note 23
Duty Cycle Distortion	D_{CD}	35	50	65	%	Note 24
Overshoot and Undershoot	V_{OS}	-	-	5	%	
Jitter	-	-	-	1.4	ns	Note 25

Note 23: Measured at line side of transformer, line replaced by 100 Ω (+/- 1%) resistor.

Note 24: Offset from 16 ns pulse width at 50% of pulse peak.

Note 25: Measured differentially.

TABLE 17-8: 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 26
Receiver Differential Squelch Threshold	V_{DS}	300	420	585	mV	

Note 26: Min/max voltages guaranteed as measured with 100 Ω resistive load.

17.6 AC Specifications

This section details the various AC timing specifications of the device.

Note: The I^2C timing adheres to the NXP *I^2C-Bus Specification*. Refer to the NXP *I^2C-Bus Specification* for detailed I^2C timing information.

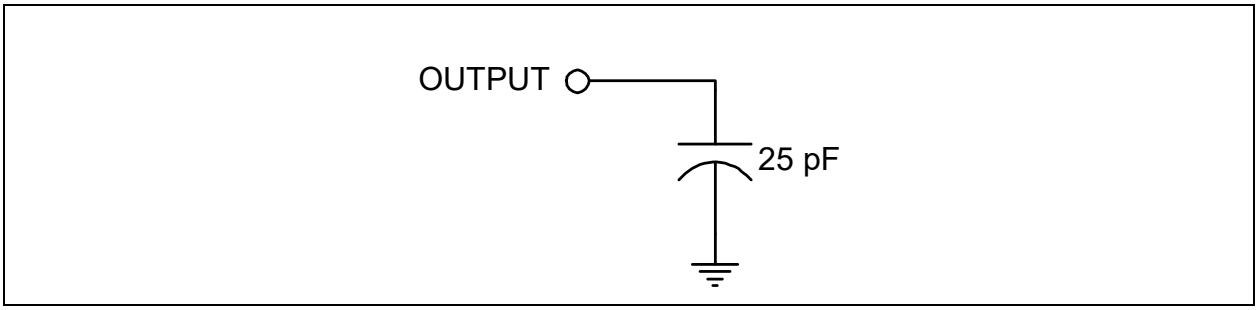
Note: The MII/SMI timing adheres to the *IEEE 802.3 Specification*.

Note: The RMII timing adheres to the RMII Consortium *RMII Specification R1.2*.

17.6.1 EQUIVALENT TEST LOAD

Output timing specifications assume the 25 pF equivalent test load, unless otherwise noted, as illustrated in Figure 17-1.

FIGURE 17-1: OUTPUT EQUIVALENT TEST LOAD



17.6.2 POWER SEQUENCING TIMING

These diagrams illustrates the device power sequencing requirements. The **VDDIO**, **VDD33**, **VDD33TXRX1**, **VDD33TXRX2**, **VDD33BIAS** and magnetics power supplies must all reach operational levels within the specified time period t_{pon}. When operating with the internal regulators disabled, **VDDCR**, **OSCVDD12**, **VDD12TX1** and **VDD12TX2** are also included into this requirement.

In addition, once the **VDDIO** power supply reaches 1.0 V, it must reach 80% of its operating voltage level (1.44 V when operating at 1.8 V, 2.0 V when operating at 2.5 V, 2.64 V when operating at 3.3 V) within an additional 15ms. This requirement can be safely ignored if using an external reset as shown in Section 17.6.3, "Reset and Configuration Strap Timing".

Device power supplies can turn off in any order provided they all reach 0 volts within the specified time period t_{poff}.

FIGURE 17-2: POWER SEQUENCE TIMING - INTERNAL REGULATORS

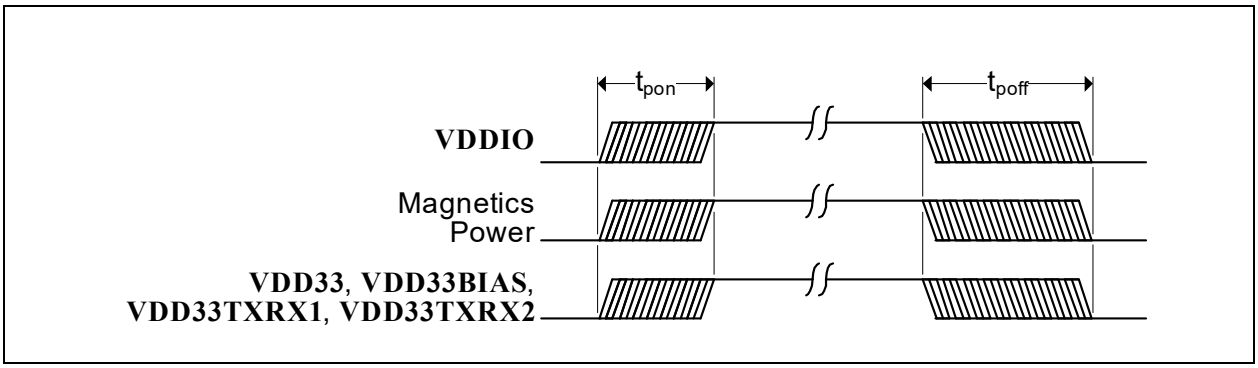


FIGURE 17-3: POWER SEQUENCE TIMING - EXTERNAL REGULATORS

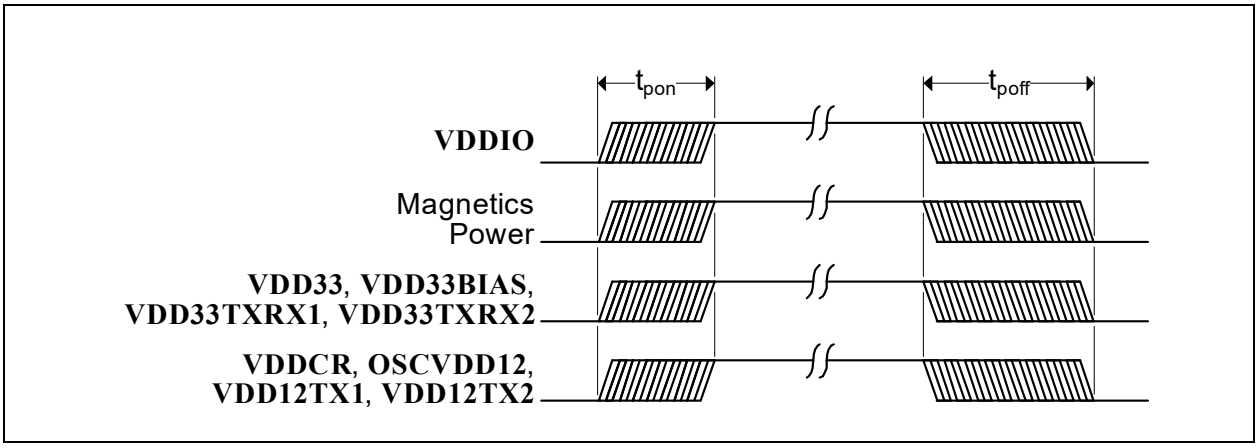


TABLE 17-9: POWER SEQUENCING TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{pon}	Power supply turn on time	-	-	50	ms
t_{poff}	Power supply turn off time	-	-	500	ms

17.6.3 RESET AND CONFIGURATION STRAP TIMING

This diagram illustrates the RST# pin timing requirements and its relation to the configuration strap pins and output drive. Assertion of RST# is not a requirement. However, if used, it must be asserted for the minimum period specified. The RST# pin can be asserted at any time, but must not be deasserted until t_{purstd} after all external power supplies have reached operational levels and if an external clock is used on the OSCI pin, it must not be deasserted until the clock input is stable. Refer to Section 6.2, "Resets," on page 46 for additional information.

FIGURE 17-4: RST# PIN CONFIGURATION STRAP LATCHING TIMING

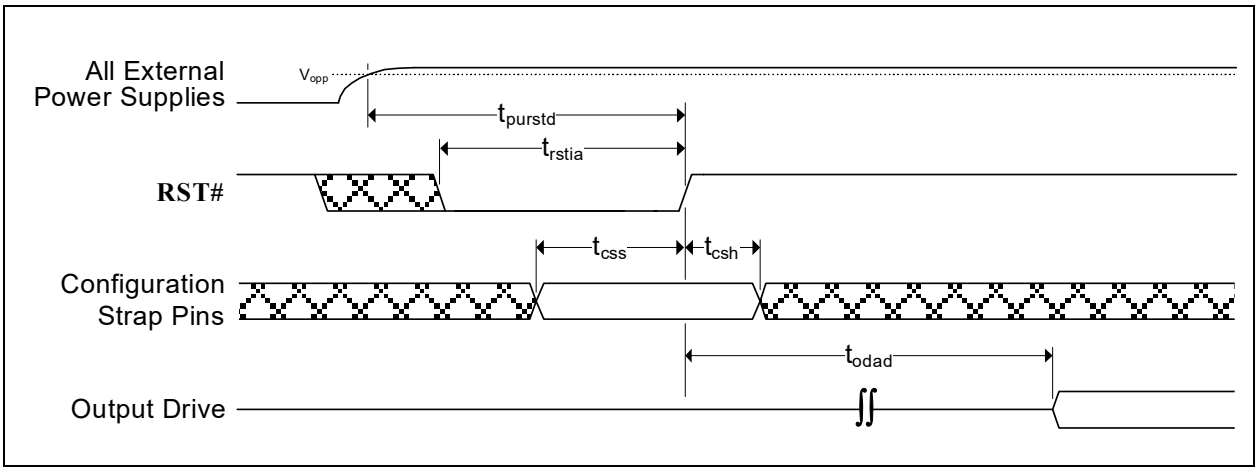


TABLE 17-10: RST# PIN CONFIGURATION STRAP LATCHING TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{purstd}	External power supplies at operational level to RST# deassertion (Note 27)	25			ms
t_{rstia}	RST# input assertion time	200	-	-	µs
t_{css}	Configuration strap pins setup to RST# deassertion	200	-	-	ns
t_{csh}	Configuration strap pins hold after RST# deassertion	10	-	-	ns
t_{odad}	Output drive after deassertion	3	-	-	µs

Note: The OSCI clock input must be stable prior to RST# deassertion.

Note: Device configuration straps, except XTAL_MODE and CLK_25_EN, are latched as a result of RST# assertion. Refer to Section 6.2.1, "Chip-Level Resets," on page 47 for details.

Note: Configuration strap latching and output drive timings shown assume that the Power-On reset has finished first otherwise the timings in Section 17.6.4, "Power-On and Configuration Strap Timing" apply.

Note: RST# does not affect the CLK_25_EN, therefore the output enable time does not apply to the CLK_25_EN configuration strap.

Note 27: When configured to be an output, CLK_25 is not driven until after the internal power-on reset time has expired. This occurs in approximately 21ms. In a daisy-chained clock configuration each device will time its power-on reset based on the CLK_25 signal from the previous device. The total system power-on reset time will therefore be the compound sum of all devices. In order to ensure all devices are released from reset concurrently, the deassertion time of RST# should be set to the number of daisy-chained times t_{purstd}.

17.6.4 POWER-ON AND CONFIGURATION STRAP TIMING

This diagram illustrates the configuration strap valid timing requirements in relation to power-on. In order for valid configuration strap values to be read at power-on, the following timing requirements must be met.

FIGURE 17-5: POWER-ON CONFIGURATION STRAP LATCHING TIMING

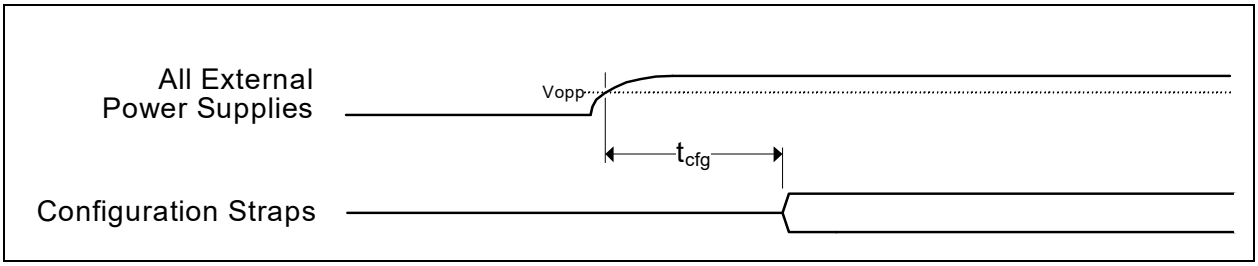


TABLE 17-11: POWER-ON CONFIGURATION STRAP LATCHING TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{cfg}	Configuration strap valid time	-	-	15	ms

Note: Configuration straps must only be pulled high or low. Configuration straps must not be driven as inputs.

Device configuration straps, except **XTAL_MODE** and **CLK_25_EN**, are also latched as a result of **RST#** assertion. Refer to Section 17.6.3, "Reset and Configuration Strap Timing" and Section 6.2.1, "Chip-Level Resets," on page 47 for additional details.

17.6.5 HOST BUS INTERFACE I/O TIMING

Timing specifications for the Host Bus Interface are given in Section 8.7, "Timing Requirements," on page 128

17.6.6 SPI/SQI SLAVE INTERFACE I/O TIMING

Timing specifications for the SPI/SQI Slave Bus Interface are given in Section 9.5, "SPI/SQI Timing Requirements," on page 199.

17.6.7 I^2C EEPROM I/O TIMING

Timing specifications for I^2C EEPROM access are given in Section 12.1, "I2C Interface Timing Requirements," on page 388.

17.6.8 ETHERCAT MII PORT MANAGEMENT ACCESS I/O TIMING

Timing specifications for the MII Port Management access are given in Section 11.11.7, "External PHY Timing," on page 293.

17.6.9 MII I/O TIMING

Timing specifications for the MII Port interface are given in Section 11.11.7, "External PHY Timing," on page 293.

17.6.10 JTAG TIMING

Timing specifications for the JTAG interface are given in Section 16.1.1, "JTAG Timing Requirements," on page 402.

17.7 Clock Circuit

The device can accept either a 25 MHz crystal or a 25 MHz single-ended clock oscillator (±50 ppm) input. If the single-ended clock oscillator method is implemented, **OSCO** should be left unconnected and **OSCI** should be driven with a clock signal that adheres to the specifications outlined throughout Section 17.0, "Operational Characteristics". See Table 17-12 and the accompanying notes for the recommended crystal specifications when using a 25 MHz crystal.

TABLE 17-12: 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	
802.3 Frequency Tolerance at 25°C	F_{tol}	-	-	±40	ppm	Note 28
802.3 Frequency Stability Over Temp	F_{temp}	-	-	±40	ppm	Note 28
802.3 Frequency Deviation Over Time	F_{age}	-	±3 to 5	-	ppm	Note 29
802.3 Total Allowable PPM Budget		-	-	±50	ppm	Note 30
EtherCAT Frequency Tolerance at 25°C	F_{tol}	-	-	±15	ppm	Note 31
EtherCAT Frequency Stability Over Temp	F_{temp}	-	-	±15	ppm	Note 31
EtherCAT Frequency Deviation Over Time	F_{age}	-	±3 to 5	-	ppm	Note 29
EtherCAT Total Allowable PPM Budget		-	-	±25	ppm	Note 32
Shunt Capacitance	C_O	-	-	7	pF	
Load Capacitance	C_L	-	-	18	pF	
Drive Level	P_W	300 Note 33	-	-	µW	
Equivalent Series Resistance	R_1	-	-	100	Ω	
Operating Temperature Range		Note 34	-	Note 35	°C	
OSCI Pin Capacitance		-	3 typ	-	pF	Note 36
OSCO Pin Capacitance		-	3 typ	-	pF	Note 36

Note 28: The maximum allowable values for frequency tolerance and frequency stability are application dependent. 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 29: Frequency Deviation Over Time is also referred to as Aging.

Note 30: The total deviation for 100BASE-TX is ±50 ppm.

Note 31: The maximum allowable values for frequency tolerance and frequency stability are application dependent. Since any particular application must meet the EtherCAT ±25 ppm Total PPM Budget, the combination of these two values must be approximately ±15 ppm (allowing for aging).

Note 32: The total deviation for EtherCAT is ±25 ppm.

Note 33: The minimum drive level requirement P_W is reduced to 100 uW with the addition of a 500 Ω series resistor, if $C_O \leq 5$ pF, $C_L \leq 12$ pF and R1≤80 Ω

Note 34: 0 °C for commercial version, -40 °C for industrial and extended industrial versions

Note 35: +70 °C for commercial version, +85 °C for industrial version, +105 °C for extended industrial version

Note 36: This number includes the pad, the bond wire and the lead frame. PCB capacitance is not included in this value. The **OSCI** pin, **OSCO** pin and PCB capacitance values are required to accurately calculate the value of the two external load capacitors. The total load capacitance must be equivalent to what the crystal expects to see in the circuit so that the crystal oscillator will operate at 25.000 MHz.

17.7.1 25 MHZ CLOCK OUTPUT

Table 17-12 specifies the **OSCI** to **CLK_25** duty cycle distortion.

TABLE 17-13: 25MHZ CLOCK OUTPUT SPECIFICATIONS

PARAMETER	MIN	NOM	MAX	NOTES
Crystal operation output duty cycle	44%	-	57%	Note 37
Crystal override mode output duty cycle distortion	0.76%	-	3.63%	
Schmitt trigger mode output duty cycle distortion	-1.03%	-	-1.36%	

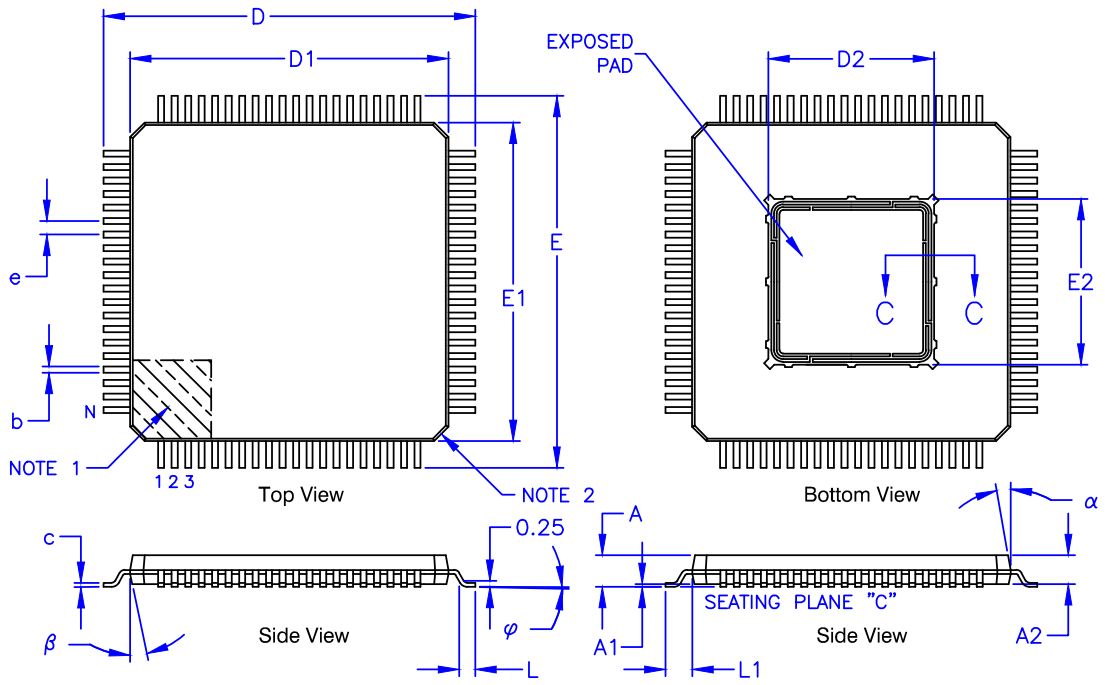
Note 37: Duty cycle at **OSCI** is approximately 50%.

Note: Specifications at an output load on **CLK_25** of 10pF.

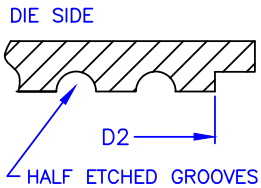
18.0 PACKAGE INFORMATION

Note: For the most current package drawings, see the Microchip Packaging Specification at http://www.microchip.com/packaging.

FIGURE 18-1: 80-TQFP-EP PACKAGE



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N		80	
Lead Pitch	e		0.50 BSC	
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	φ	0°	3.5°	7°
Overall PKG Length/Width	D/E		14.00 BSC	
PKG Body Length/Width	D1/E1		12.00 BSC	
Exposed Pad Length/Width	D2/E2	5.90	6.00	6.10
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°



Section C - C

Lead Position Tolerance (dim "b")

⌖	0.08Ⓜ	C

Lead Coplanarity Tolerance

⌓	0.08	C

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

APPENDIX A: REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision Level	Section/Figure/Entry	Correction
DS00003422A (03-18-20)	All	Initial Release

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X]	[-X]	/	XXX
Device	Tape and Reel Option	Temperature Range		Package

Device:	LAN9254	
Tape and Reel Option:	Blank = Standard packaging (Tray) T = Tape and Reel (Note 1)	
Temperature Range:	Blank = 0°C to +70°C I = -40°C to +85°C V = -40°C to +105°C	(Commercial) (Industrial) (Extended Industrial)
Package:	JRX = 80-pin TQFP-EP	

Examples:

a) LAN9254/JRX
Standard Packaging (Tray),
Commercial Temperature,
80-pin TQFP-EP

b) LAN9254-I/JRX
Standard Packaging (Tray)
Industrial Temperature,
80-pin TQFP-EP

c) LAN9254-V/JRX
Standard Packaging (Tray)
Extended Industrial Temperature,
80-pin TQFP-EP

d) LAN9254T/JRX
Tape and Reel
Commercial Temperature,
80-pin TQFP-EP

e) LAN9254T-I/JRX
Tape and Reel
Industrial Temperature,
80-pin TQFP-EP

f) LAN9254T-V/JRX
Tape and Reel
Extended Industrial Temperature,
80-pin TQFP-EP

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

ISBN: 9781522457169

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820