Hardware Data Sheet

ET1100

EtherCAT® Slave Controller

Section I – Technology
(Online at http://www.beckhoff.com)

Section II – Register Description
(Online at http://www.beckhoff.com)

Section III – Hardware Description

Pinout, Interface description, electrical and mechanical specification, ET1100 features and registers

Version 2.0
Date: 2017-02-21

BECKHOFF

DOCUMENT ORGANIZATION

The Beckhoff EtherCAT Slave Controller (ESC) documentation covers the following Beckhoff ESCs:

- ET1200
- ET1100
- EtherCAT IP Core for Intel® FPGAs
- EtherCAT IP Core for Xilinx® FPGAs
- ESC20

The documentation is organized in three sections. Section I and section II are common for all Beckhoff ESCs, Section III is specific for each ESC variant.

The latest documentation is available at the Beckhoff homepage (http://www.beckhoff.com).

Section I – Technology (All ESCs)

Section I deals with the basic EtherCAT technology. Starting with the EtherCAT protocol itself, the frame processing inside EtherCAT slaves is described. The features and interfaces of the physical layer with its two alternatives Ethernet and EBUS are explained afterwards. Finally, the details of the functional units of an ESC like FMMU, SyncManager, Distributed Clocks, Slave Information Interface, Interrupts, Watchdogs, and so on, are described.

Since Section I is common for all Beckhoff ESCs, it might describe features which are not available in a specific ESC. Refer to the feature details overview in Section III of a specific ESC to find out which features are available.

Section II – Register Description (All ESCs)

Section II contains detailed information about all ESC registers. This section is also common for all Beckhoff ESCs, thus registers, register bits, or features are described which might not be available in a specific ESC. Refer to the register overview and to the feature details overview in Section III of a specific ESC to find out which registers and features are available.

Section III – Hardware Description (Specific ESC)

Section III is ESC specific and contains detailed information about the ESC features, implemented registers, configuration, interfaces, pinout, usage, electrical and mechanical specification, and so on. Especially the Process Data Interfaces (PDI) supported by the ESC are part of this section.

Additional Documentation

Application notes and utilities like pinout configuration tools for ET1100 can also be found at the Beckhoff homepage.

DOCUMENT HISTORY

Version	Comment
0.6	Editorial Changes
0.7	• Synchronous µController Interface LSB/MSB clarification table added • EEPROM_LOADED pull-down recommendation added • Chip label updated • $VCC_{I/O}$/$GND_{I/O}$ pins adjacent to LDO indicated • Frame processing order example corrected • I^2C EEPROM interface description added • MII management interface description added • Corrected Process RAM size in Register Overview • P_CONF does not correspond with physical ports. See new port configuration tables for details. • Revision/Build information added
0.8	• CLK25OUT1/2 availability completed • Recommendations for unused input pins added (should not be left open) • EEPROM_SIZE description corrected from Kbyte to Kbit, possible EEPROM sizes range from 16 Kbit to 4 Mbit • RoHS compliance added • Autonegotiation is mandatory for ESCs • Description of power supply options added • Electrical characteristics added/revised • SPI_IRQ delay added, support for SPI masters with 2 or 4 bytes added • TX Shift timing diagram and description added • Internal 27 kΩ PU/PD resistors at EBUS-RX pins added • LED polarity depending on configuration pin setting described • Recommendation for voltage stabilization capacitors added • Description of Digital I/O behavior on watchdog expiration enhanced • 8 bit asynchronous µController PDI connection added • EBUS ports are open failsafe • Reset example schematic added • Ethernet PHY requirements and PHY connection schematic added • MI_DATA pull-up requirement added • µController PDI: DATA bus signal direction corrected • Pin/Signal description overview added • PERR(x) LEDs are only for testing/debugging • Editorial changes
1.0	• RUN, LINKACT/x) and PERR(x) LED activity level corrected: active high if pulled down, active low if pulled up • DC Characteristics enhanced: added $V_{Reset\ Core}$, V_{ID}, V_{IC} • Synchronous µController interface: timing characteristics enhanced • Note on RBIAS if no EBUS ports/only MII ports are used • DC SYNC/LATCH signal description and timing characteristics added • MII Interface chapter and MII timing characteristics added • EBUS Interface chapter added • Frame processing order, PHY requirements, EEPROM Interface description and MII Management Interface description moved to Section I • TX Shift description moved to MII Interface chapter • Ambient temperature range instead of junction temperature range • Editorial changes

Version	Comment
1.1	• Port configurations with 2 ports: P_CONF[3] erroneously named P_MODE[3] • Clarified I/O voltage with respect to I/O power supply (only 3.3V I/O with $V_{CCI/O}$=3.3V, and no 5V input tolerance unless $V_{CCI/O}$=5V) • Update to ET1100 stepping 1 • Added/revised OSC_IN, CLK25OUT1/2, and MII TX signal timings • Added soldering profile • PHY address configuration changed • Added feature detail overview, removed redundant feature details • PDI and DC SYNC/LATCH signals are not driven until EEPROM is loaded • Synchronous 8/16 bit µController interface: clarified that clock is CPU_CLK_IN • Editorial changes
1.2	• PHY address configuration chapter added, configuration revised • Enhanced link detection for MII available depending on PHY address configuration • Ethernet Management Interface: read and write times were interchanged • Reserved pins are input pins • Editorial changes
1.3	• Added reset timing figure and power-on value sample time • Distributed Clocks SYNC/LATCH signals are configurable and unidirectional • Information on CLK25OUT/CPU_CLK clock output during reset added • Description of internal PU/PD resistors at EBUS_RX pins enhanced • Added t_{Diff} timing characteristic • Power supply example schematic clarified • Enhanced package information: MSL, ball's material, and solder joint recommendation • Digital I/O PDI: added SOF/OUTVALID description, dispensable timings removed • Editorial changes
1.4	• Register 0x0980 is only available if DC Sync Unit is enabled (0x0140.10=1) • Updated solder joint recommendation • OSC_IN/OSC_OUT pin capacitance added, crystal connection note extended • Release Notes added • Timing requirement for asynchronous µController PDI ($t_{ADR\_BHE\_setup}$) relaxed • Input threshold voltage for OSC_IN added • Example schematic for transparent mode added • Renamed Err(x) LED to PERR(x) • Digital I/O PDI: OE_CONF functionality in bidirectional mode corrected • Digital I/O PDI: output event description corrected (EOF mode and WD_TRIG mode) • SPI PDI: access error if SPI_DI not 1 in the last read byte (not SPI_DO) • Async./sync. µC PDI: access error with A(0)=1 and nBHE=1 (not nBHE=0), timing requirements and diagrams clarified • Async. µC PDI: timing requirement for asynchronous µController PDI ($t_{ADR\_BHE\_setup}$) relaxed • AC timing: forwarding delay figures enhanced • Editorial changes
1.5	• Reset timing figure corrected • Maximum soldering profile added • SPI PDI updated • SII EEPROM interface is a point-to-point connection • Editorial changes
1.6	• Update to ET1100-0002 • Editorial changes
1.7	• µC PDI timing updated • Editorial changes

Version	Comment
1.8	• Enhanced Link Detection must not be activated if EBUS ports are used • Enhanced Link Detection for MII ports requires PHY address offset = 0 • Digital Output principle schematic updated • Chip label updated • Editorial changes
1.9	• Update to ET1100-0003 • Enhanced Link Detection for MII ports supports PHY address offset 0 and 16 • Enhanced Link Detection for MII ports can be disabled at any time • Enhanced Link Detection for EBUS ports is always disabled • MII management interface issues additional MCLK cycle after write accesses • Remote link down signalling time configurable 0x0100[22] • Editorial changes
2.0	• Added thermal characteristics • Added note on enabling transparent mode (disables Link Polarity configuration to active high) • Added RoHS 2 compliance including amendment "COMMISSION DELEGATED DIRECTIVE (EU) 2015/863" • Clarified soldering temperature and time • Updated PCB recommendations • Updated recommended power supply options • Editorial changes

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

(x)	Physical Port x
[y]	Bit y
µC	Microcontroller
ADR	Address
AL	Application Layer
BD	Bidirectional
BGA	Ball Grid Array
BHE	Bus High Enable
CMD	Command
CS	Chip Select
DC	Distributed Clock
Dir.	Pin direction
DL	Data Link Layer
ECAT	EtherCAT
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EOF	End of Frame
ESC	EtherCAT Slave Controller
ESI	EtherCAT Slave Information
FMMU	Fieldbus Memory Management Unit
GPI	General Purpose Input
GPO	General Purpose Output
I	Input
I/O	Input or Output
IRQ	Interrupt Request
LDO	Low Drop-Out regulator
LI-	LVDS RX-
LI+	LVDS RX+
LO-	LVDS TX-
LO+	LVDS TX+
MAC	Media Access Controller
MDIO	Management Data Input / Output
MI	(PHY) Management Interface
MII	Media Independent Interface
MISO	Master In – Slave Out
MOSI	Master Out – Slave In
n.a.	not available
n.c.	not connected
O	Output
PD	Pull-down
PDI	Process Data Interface
PLL	Phase Locked Loop
PU	Pull-up
QFN	Quad Flat package No leads
RD	Read
SII	Slave Information Interface
SM	SyncManager
SOF	Start of Frame
SPI	Serial Peripheral Interface
TA	Transfer Acknowledge
TFBGA	Thin-profile Fine-pitch BGA
TS	Transfer Start
UI	Unused Input (PDI: PD, others: GND)
WD	Watchdog
WPD	Weak Pull-down, sufficient only for configuration signals
WPU	Weak Pull-up, sufficient only for configuration signals
WR	Write

1 Overview

The ET1100 ASIC is an EtherCAT Slave Controller (ESC). It takes care of the EtherCAT communication as an interface between the EtherCAT fieldbus and the slave application. The ET1100 supports a wide range of applications. For example, it may be used as a 32 bit Digital I/O node without external logic using Distributed clocks, or as a part of a complex µController design with up to 4 EtherCAT communication ports.

Table 1: ET1100 Main Features

Feature	ET1100
Ports	2-4 ports (each EBUS or MII)
FMMUs	8
SyncManagers	8
RAM	8 Kbyte
Distributed Clocks	Yes, 64 bit (power saving options with SII EEPROM configuration)
Process Data Interfaces	• 32 Bit Digital I/O (unidirectional/bidirectional) • SPI Slave • 8/16 asynchronous/synchronous µController
Power supply	Integrated voltage regulator (LDO) for logic core/PLL (5V/3.3V to 2.5V), optional external power supply for logic core/PLL.
I/O	3.3V compatible I/O
Package	BGA128 (10x10 mm^2)
Other features	• Internal 1GHz PLL • Clock output for external devices (10, 20, 25 MHz)

The general functionality of the ET1100 EtherCAT Slave Controller (ESC) is shown in Figure 1:

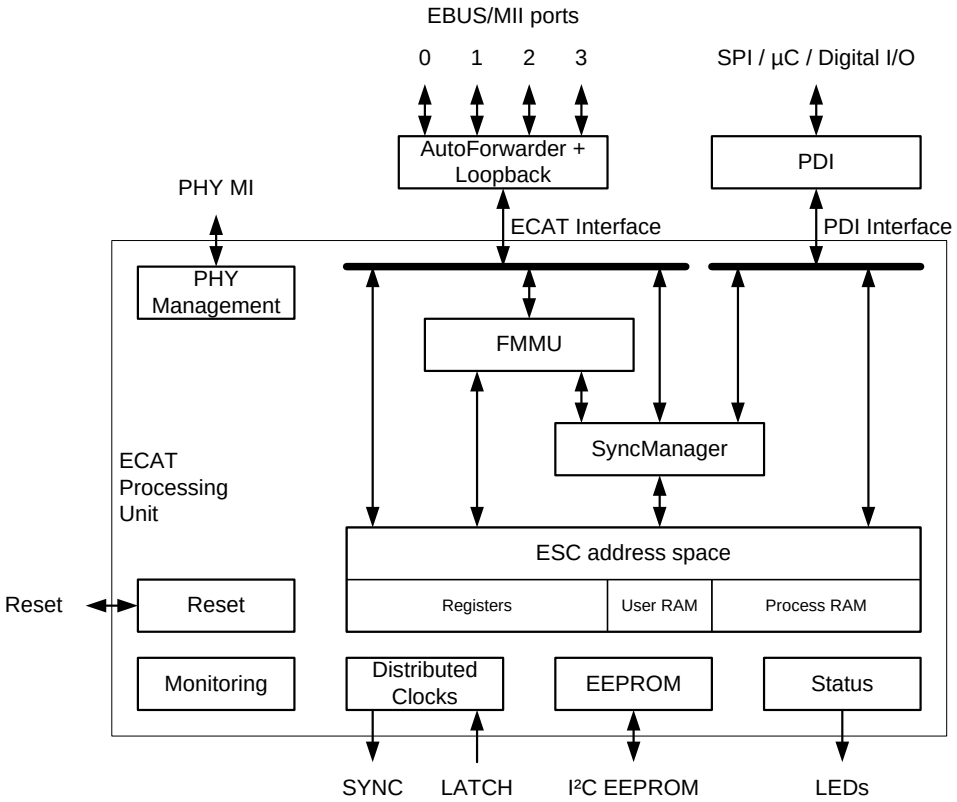

Figure 1: ET1100 Block Diagram

1.1 Frame processing order

The frame processing order of the ET1100 depends on the number of ports and the chip mode (logical port numbers are used):

Table 2: Frame Processing Order

Number of Ports	Frame processing order
2	0→EtherCAT Processing Unit→1 / 1→0
3	0→EtherCAT Processing Unit→1 / 1→2 / 2→0 (log. ports 0,1, and 2) or 0→EtherCAT Processing Unit→3 / 3→1 / 1→0 (log. ports 0,1, and 3)
4	0→EtherCAT Processing Unit→3 / 3→1 / 1→2 / 2→0

Figure 2 shows the frame processing in general:

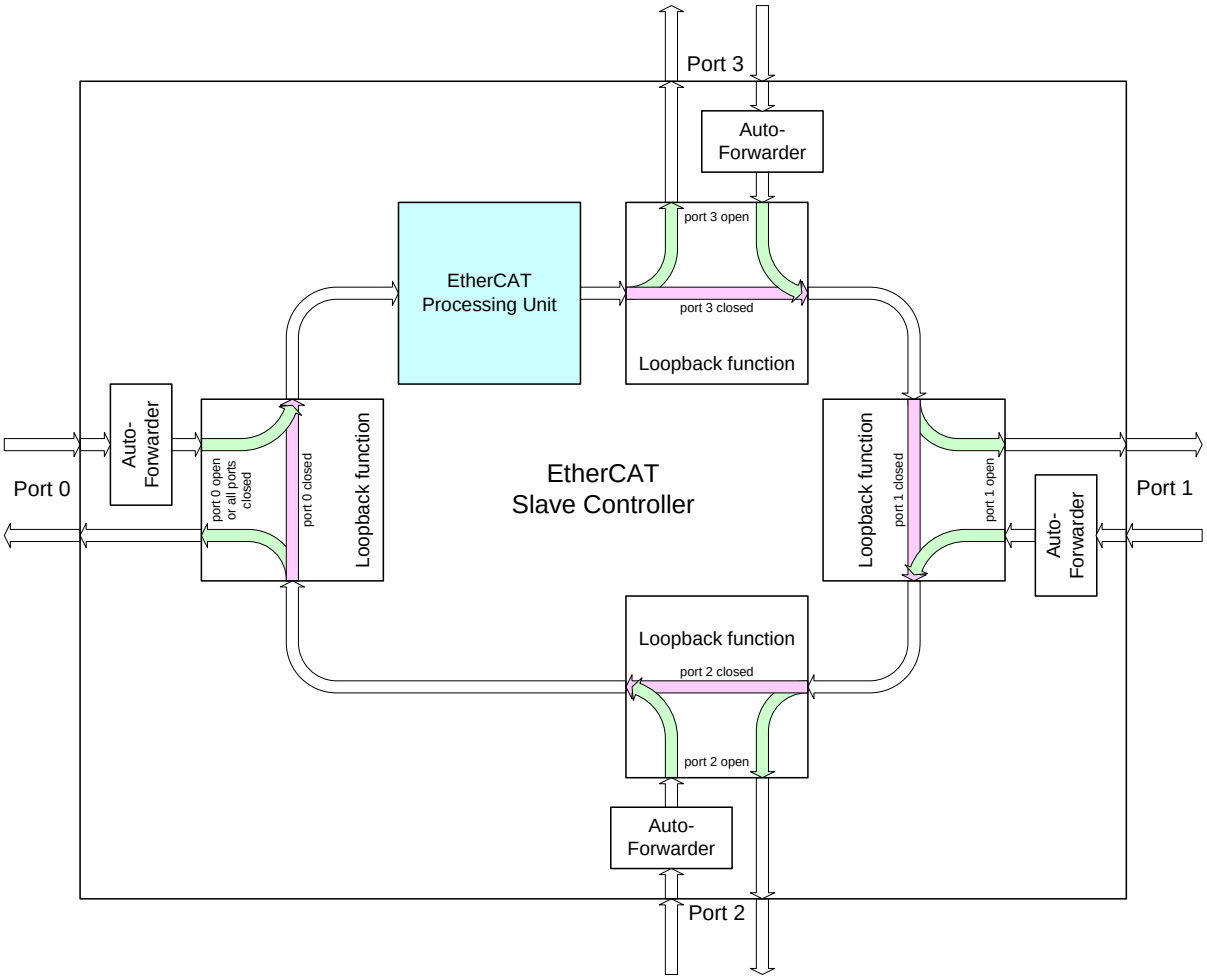

Figure 2: Frame Processing

1.2 Scope of this document

This documentation refers to stepping ET1100-0003.

1.3 Revision/Build History

Table 3: Revision/Build History

Revision Register 0x0001	Build Register 0x0002:0x0003	Stepping
0x00	0x0000	ET1100-0000 or ET1100-0001
0x00	0x0002	ET1100-0002
0x00	0x0003	ET1100-0003

The stepping code is printed on the devices, do not confuse the stepping code with the ordering codes.

2 Features and Registers

2.1 Features

Table 4: ET1100 Feature Details

Feature	ET1100-0003
EtherCAT Ports	**2-4**
Permanent ports	2-4
Optional Bridge port 3 (EBUS or MII)	-
EBUS ports	0-4
MII ports	0-4
RMII ports	-
RGMII ports	-
Port 0	-
Ports 0, 1	x
Ports 0, 1, 2	x
Ports 0, 1, 3	x
Ports 0, 1, 2, 3	x
EtherCAT mode	Direct
Slave Category	Full Slave
Position addressing	x
Node addressing	x
Logical addressing	x
Broadcast addressing	x
Physical Layer General Features	
FIFO Size configurable (0x0100[18:16])	x
FIFO Size default from SII EEPROM	-
Auto-Forwarder checks CRC and SOF	x
Forwarded RX Error indication, detection and Counter (0x0308:0x030B)	x
Lost Link Counter (0x0310:0x0313)	x
Prevention of circulating frames	x
Fallback: Port 0 opens if all ports are closed	x
VLAN Tag and IP/UDP support	x
Enhanced Link Detection per port configurable	-
EBUS Features	
Low Jitter	x
Enhanced Link Detection supported	-
Enhanced Link Detection compatible	x
EBUS signal validation	x
LVDS Transceiver internal	x
LVDS sample rate [MHz]	1,000
Remote link down signaling time configurable 0x0100[22]	x
General Ethernet Features (MII/RMII/RGMII)	
MII Management Interface (0x0510:0x051F)	x
Supported PHY Address Offsets	0/16
Individual port PHY addresses	-
Port PHY addresses readable	-
Link Polarity configurable	x
Enhanced Link Detection supported	x
FX PHY support (native)	-
PHY reset out signals	-
Link detection using PHY signal (LED)	x
MI link status and configuration	-
MI controllable by PDI (0x0516:0x0517)	-
MI read error (0x0510[13])	-
MI PHY configuration update status (0x0518[5])	-
MI preamble suppression	-
Additional MCLK	x
Gigabit PHY configuration	-
Gigabit PHY register 9 detection	-
FX PHY configuration	-
Transparent Mode	x
MII Features	
CLK25OUT as PHY clock source	x
Bootstrap TX Shift settings	x

Feature	ET1100-0003
Automatic TX Shift setting (with TX_CLK)	-
TX Shift not necessary (PHY TX_CLK as clock source)	-
FIFO size reduction steps	1
PDI General Features	
Increased PDI performance	-
Extended PDI Configuration (0x0152:0x0153)	x
PDI Error Counter (0x030D)	x
PDI Error Code (0x030E)	-
CPU_CLK output (10, 20, 25 MHz)	x
SOF, EOF, WD_TRIG and WD_STATE independent of PDI	-
Available PDIs and PDI features depending on port configuration	x
PDI selection at run-time (SII EEPROM)	x
PDI active immediately (SII EEPROM settings ignored)	-
PDI function acknowledge by write	-
PDI Information register 0x014E:0x014F	-
Digital I/O PDI	**x**
Digital I/O width [bits]	8/16/24/32
PDI Control register value (0x0140:0x0141)	4
Control/Status signals:	7/0[1]
LATCH_IN	x[1]
SOF	x[1]
OUTVALID	x[1]
WD_TRIG	x[1]
OE_CONF	x[1]
OE_EXT	x[1]
EEPROM_ Loaded	x[1]
WD_STATE	-
EOF	-
Granularity of direction configuration [bits]	2
Bidirectional mode	x
Output high-Z if WD expired	x
Output 0 if WD expired	x
Output with EOF	x
Output with DC SyncSignals	x
Input with SOF	x
Input with DC SyncSignals	x
SPI Slave PDI	**x**
Max. SPI clock [MHz]	20
SPI modes configurable (0x0150[1:0])	x
SPI_IRQ driver configurable (0x0150[3:2])	x
SPI_SEL polarity configurable (0x0150[4])	x
Data out sample mode configurable (0x0150[5])	x
Busy signaling	-
Wait State byte(s)	x
Number of address extension byte(s)	any
2/4 Byte SPI master support	x
Extended error detection (read busy violation)	x
SPI_IRQ delay	x
Status indication	x
EEPROM_ Loaded signal	x

[1] Availability depending on port configuration

Feature	ET1100-0003
Asynchronous µController PDI	**8/16 bit**
Extended µC configuration bits 0x0150[7:4], 0x0152:0x0153	x
ADR[15:13] available (000_b if not available)	x
EEPROM_Loaded signal	x
RD polarity configurable (0x0150[7])	x
Read BUSY delay (0x0152[0])	x
Write after first edge (0x0152[2])	-
Default BUSY state	-
Synchronous µController PDI	**8/16 bit**
EEPROM_ Loaded signal	x
On-Chip Bus PDI	**-**
EtherCAT Bridge (port 3, EBUS/MII)	**-**
General Purpose I/O	**x**
GPO bits	0-16
GPI bits	0-16
GPIO available independent of PDI or port configuration	-
GPIO available without PDI	-
Concurrent access to GPO by ECAT and PDI	x
ESC Information	
Basic Information (0x0000:0x0006)	x
Port Descriptor (0x0007)	x
ESC Features supported (0x0008:0x0009)	x
Extended ESC Feature Availability in User RAM (0x0F80 ff.)	-
Write Protection (0x0020:0x0031)	x
Data Link Layer Features	
ECAT Reset (0x0040)	x
PDI Reset (0x0041)	-
ESC DL Control (0x0100:0x0103) bytes	4
EtherCAT only mode (0x0100[0])	x
Temporary loop control (0x0100[1])	x
FIFO Size configurable (0x0100[18:16])	x
Configured Station Address (0x0010:0x0011)	x
Configured Station Alias (0x0100[24], 0x0012:0x0013)	x
Physical Read/Write Offset (0x0108:0x0109)	x
Application Layer Features	
Extended AL Control/Status bits (0x0120[15:5], 0x0130[15:5])	x
AL Status Emulation (0x0140[8])	x
AL Status Code (0x0134:0x0135)	x
Interrupts	
ECAT Event Mask (0x0200:0x0201)	x
AL Event Mask (0x0204:0x0207)	x
ECAT Event Request (0x0210:0x0211)	x
AL Event Request (0x0220:0x0223)	x
SyncManager activation changed (0x0220[4])	x
SyncManager watchdog expiration (0x0220[6])	-
Error Counters	
RX Error Counter (0x0300:0x0307)	x
Forwarded RX Error Counter (0x0308:0x030B)	x
ECAT Processing Unit Error Counter (0x030C)	x
PDI Error Counter (0x030D)	x
Lost Link Counter (0x0310:0x0313)	x
Watchdog	
Watchdog Divider configurable (0x0400:0x0401)	x
Watchdog Process Data	x
Watchdog PDI	x
Watchdog Counter Process Data (0x0442)	x
Watchdog Counter PDI (0x0443)	x
SII EEPROM Interface (0x0500:0x050F)	
EEPROM sizes supported	1 Kbyte-4 Mbyte
EEPROM size reflected in 0x0502[7]	x
EEPROM controllable by PDI	x
EEPROM Emulation by PDI	-

Feature	ET1100-0003
EEPROM Emulation CRC error 0x0502[11] PDI writable	-
Read data bytes (0x0502[6])	8
Internal Pull-Ups for EEPROM_CLK and EEPROM_DATA	x
I2C base address	0
FMMUs	8
Bit-oriented operation	x
SyncManagers	8
Watchdog trigger generation for 1 Byte Mailbox configuration independent of reading access	x
SyncManager Event Times (+0x8[7:6])	x
Buffer state (+0x5[7:6])	-
SyncManager Sequential mode	-
SyncManager deactivation delay	-
Distributed Clocks	**x**
Width	64
Sync/Latch signals	2
SyncManager Event Times (0x09F0:0x09FF)	x
DC Receive Times	x
DC Time Loop Control controllable by PDI	-
DC Sync/Latch activation (0x0140[11:10])	x
Propagation delay measurement with traffic (BWR/FPWR 0x900 detected at each port)	x
LatchSignal state in Latch Status register (0x09AE:0x09AF)	x
SyncSignal Auto-Activation (0x0981[3])	-
SyncSignal 32 or 64 bit Start Time (0x0981[4])	-
SyncSignal Late Activation (0x0981[6:5])	-
SyncSignal debug pulse (0x0981[7])	-
SyncSignal Activation State 0x0984)	-
Reset filters after writing filter depth	-
ESC Specific Registers (0x0E00:0x0EFF)	
Product and Vendor ID	-
POR Values	x
FPGA Update (online)	-
Process RAM and User RAM	
Process RAM (0x1000 ff.) [Kbyte]	8
User RAM (0x0F80:0x0FFF)	x
Extended ESC Feature Availability in User RAM	-
RAM initialization	-
Additional EEPROMs	1
SII EEPROM (I^2C)	x
FPGA configuration EEPROM	-
LED Signals	
RUN LED	x
RUN LED override	-
Link/Activity(x) LED per port	x
PERR(x) LED per port	x
Device ERR LED	-
STATE_RUN LED	-
Optional LED states	
RUN LED: Bootstrap	x
RUN LED: Booting	-
RUN LED: Device identification	-
RUN LED: loading SII EEPROM	-
Error LED: SII EEPROM loading error	-
Error LED: Invalid hardware configuration	-
Error LED: Process data watchdog timeout	-
Error LED: PDI watchdog timeout	-
Error LED: Error Indication 0x0130[4]	-
Link/Activity: port closed	-
Link/Activity: local auto-negotiation error	-
Link/Activity: remote auto-negotiation error	-
Link/Activity: unknown PHY auto-negotiation error	-
LED test	-

Feature	ET1100-0003
Clock supply	
Crystal	x
Crystal oscillator	x
TX_CLK from PHY	x
25ppm clock source accuracy	x
Internal PLL	x
Power Supply Voltages	1-2
I/O Voltage	
3.3 V	x
3.3V / 5V tolerant	-
5 V	(x)
Core Voltage	2.5V

Feature	ET1100-0003
Internal LDOs	1
LDO supply voltage	3.3V/5V
Core Voltage	x
I/O Voltage	-
Package	BGA128
Size [mm²]	10x10
Original Release date	3/2007
Configuration and Pinout calculator (XLS)	x
Register Configuration	fixed
Internal tri-strate drivers	x

Table 5: Legend

Symbol	Description
x	available
-	not available
c	configurable

2.2 Register Overview

An EtherCAT Slave Controller (ESC) has an address space of 64 Kbyte. The first block of 4 Kbyte (0x0000:0x0FFF) is dedicated for registers. The process data RAM starts at address 0x1000, its size is 8 Kbyte (end address 0x2FFF).

Table 7 gives an overview of the available registers.

Table 6: Register Overview Legend

Symbol	Description	ET1100 EEPROM setting
x	Available	
-	Not available	
SL	DC SYNC Out Unit and/or Latch In Unit enabled	0x0000[10]=1, or 0x0000[11]=1
S	DC SYNC Out Unit enabled	0x0000[10]=1
L	DC Latch In Unit enabled	0x0000[11]=1
io	Available if Digital I/O PDI is selected	

Table 7: Register Overview

Address	Length (Byte)	Description	ET1100
0x0000	1	Type	x
0x0001	1	Revision	x
0x0002:0x0003	2	Build	x
0x0004	1	FMMUs supported	x
0x0005	1	SyncManagers supported	x
0x0006	1	RAM Size	x
0x0007	1	Port Descriptor	x
0x0008:0x0009	2	ESC Features supported	x
0x0010:0x0011	2	Configured Station Address	x
0x0012:0x0013	2	Configured Station Alias	x
0x0020	1	Write Register Enable	x
0x0021	1	Write Register Protection	x
0x0030	1	ESC Write Enable	x
0x0031	1	ESC Write Protection	x
0x0040	1	ESC Reset ECAT	x
0x0041	1	ESC Reset PDI	-
0x0100:0x0101	2	ESC DL Control	x
0x0102:0x0103	2	Extended ESC DL Control	x
0x0108:0x0109	2	Physical Read/Write Offset	x
0x0110:0x0111	2	ESC DL Status	x

Address	Length (Byte)	Description	ET1100
0x0120	5 bits [4:0]	AL Control	x
0x0120:0x0121	2	AL Control	x
0x0130	5 bits [4:0]	AL Status	x
0x0130:0x0131	2	AL Status	x
0x0134:0x0135	2	AL Status Code	x
0x0138	1	RUN LED Override	-
0x0139	1	ERR LED Override	-
0x0140	1	PDI Control	x
0x0141	1	ESC Configuration	x
0x014E:0x014F	2	PDI Information	-
0x0150	1	PDI Configuration	x
0x0151	1	DC Sync/Latch Configuration	x
0x0152:0x0153	2	Extended PDI Configuration	x
0x0200:0x0201	2	ECAT Event Mask	x
0x0204:0x0207	4	PDI AL Event Mask	x
0x0210:0x0211	2	ECAT Event Request	x
0x0220:0x0223	4	AL Event Request	x
0x0300:0x0307	4x2	Rx Error Counter[3:0]	x
0x0308:0x030B	4x1	Forwarded Rx Error counter[3:0]	x
0x030C	1	ECAT Processing Unit Error Counter	x
0x030D	1	PDI Error Counter	x
0x030E	1	PDI Error Code	-
0x0310:0x0313	4x1	Lost Link Counter[3:0]	x
0x0400:0x0401	2	Watchdog Divider	x
0x0410:0x0411	2	Watchdog Time PDI	x
0x0420:0x0421	2	Watchdog Time Process Data	x
0x0440:0x0441	2	Watchdog Status Process Data	x
0x0442	1	Watchdog Counter Process Data	x
0x0443	1	Watchdog Counter PDI	x
0x0500:0x050F	16	SII EEPROM Interface	x
0x0510:0x0515	6	MII Management Interface	x
0x0516:0x0517	2	MII Management Access State	-
0x0518:0x051B	4	PHY Port Status[3:0]	-
0x0600:0x06FC	16x13	FMMU[15:0]	8
0x0800:0x087F	16x8	SyncManager[15:0]	8

Address	Length (Byte)	Description	ET1100
0x0900:0x090F	4x4	DC – Receive Times	x
0x0910:0x0917	8	DC – System Time	SL
0x0918:0x091F	8	DC – Receive Time EPU	SL
0x0920:0x0927	8	DC – System Time Offset	SL
0x0928:0x092B	4	DC – System Time Delay	SL
0x092C:0x092F	4	DC – System Time Difference	SL
0x0930:0x0931	2	DC – Speed Counter Start	SL
0x0932:0x0933	2	DC – Speed Counter Diff	SL
0x0934	1	DC – System Time Difference Filter Depth	SL
0x0935	1	DC – Speed Counter Filter Depth	SL
0x0936	1	DC – Receive Time Latch mode	-
0x0980	1	DC – Cyclic Unit Control	S
0x0981	1	DC – Activation	S
0x0982:0x0983	2	DC – Pulse length of SyncSignals	S
0x0984	1	DC – Activation Status	-
0x098E	1	DC – SYNC0 Status	S
0x098F	1	DC – SYNC1 Status	S
0x0990:0x0997	8	DC – Next Time Cyclic Operation/Next SYNC0 Pulse	S
0x0998:0x099F	8	DC – Next SYNC1 Pulse	S
0x09A0:0x09A3	4	DC – SYNC0 Cycle Time	S
0x09A4:0x09A7	4	DC – SYNC1 Cycle Time	S
0x09A8	1	DC – Latch0 Control	L
0x09A9	1	DC – Latch1 Control	L
0x09AE	1	DC – Latch0 Status	L
0x09AF	1	DC – Latch1 Status	L
0x09B0:0x09B7	8	DC – Latch0 Positive Edge	L
0x09B8:0x09BF	8	DC – Latch0 Negative Edge	L
0x09C0:0x09C7	8	DC – Latch1 Positive Edge	L
0x09C7:0x09CF	8	DC – Latch1 Negative Edge	L
0x09F0:0x09F3	4	DC – EtherCAT Buffer Change Event Time	SL
0x09F8:0x09FB	4	DC – PDI Buffer Start Event Time	SL
0x09FC:0x09FF	4	DC – PDI Buffer Change Event Time	SL
0x0E00:0x0E03	4	Power-On Values [Bits]	16
0x0E00:0x0E07	8	Product ID	-
0x0E08:0x0E0F	8	Vendor ID	-
0x0E10	1	ESC Health Status	-
0x0F00:0x0F03	4	Digital I/O Output Data	x
0x0F10:0x0F17	8	General Purpose Outputs [Byte]	2
0x0F18:0x0F1F	8	General Purpose Inputs [Byte]	2
0x0F80:0x0FFF	128	User RAM	x
0x1000:0x1003	4	Digital I/O Input Data	io
0x1000 ff.		Process Data RAM [Kbyte]	8

3 Pin Description

For pin configuration there is a table calculation file (ET1100 configuration and pinout V<version>.xls) available to make pin configuration easier. This file can be downloaded from the Beckhoff homepage (http://www.beckhoff.com). This documentation supersedes the table calculation file.

Input pins should not be left open/floating. Unused input pins (denoted with direction UI) without external or internal pull-up/pull-down resistor should not be left open. Unused configuration pins should be pulled down if the application allows this (take care of configuration signals in the PDI[39:0] area when bidirectional Digital I/O is used). Unused PDI[39:0] input pins should be pulled down, all other input pins can be connected to GND directly.

Pull-up resistors must connect to $V_{CC\ I/O}$, not to a different power source. Otherwise the ET1100 could be powered via the resistors and the internal clamping diodes as long as $V_{CC\ I/O}$ is below the other power source.

Internal pull-up/pull-down resistor values shown in the pinout tables are nominal.

3.1 Overview

3.1.1 Pin Overview

Table 8: Pin Overview

Pin	Pin name	Dir.	Pin	Pin name	Dir.
A1	PDI[27]/RX_DV(3)/EBUS(3)-RX-	BD/LI-	D7	GND_{Core}	
A2	PDI[26]/TX_ENA(3)/EBUS(3)-TX+	BD/LO+	D8	Res. [7]	I
A3	PDI[24]/TX_D(3)[1]/EBUS(3)-TX-	BD/LO-	D9	$GND_{I/O}$	
A4	PDI[22]/TX_D(3)[3]	BD	D10	$V_{CC\ I/O}$	
A5	PDI[20]/RX_D(3)[3]	BD	D11	PDI[1]	BD
A6	PDI[18]/RX_D(3)[0]	BD	D12	PDI[0]	BD
A7	PDI[16]/RX_ERR(3)	BD	E1	TX_D(2)[1]/EBUS(2)-TX-	O/LO-
A8	PDI[14]	BD	E2	PDI[34]/TX_D(2)[0]/ CTRL_STATUS_MOVE	BD
A9	PDI[12]	BD	E3	LINKACT(2)/P_CONF[2]	BD
A10	PDI[10]	BD	E4	Res. [0]	I
A11	PDI[8]	BD	E9	$V_{CC\ I/O}$	
A12	PDI[6]	BD	E10	Res. [3]	I
B1	PDI[29]/RX_D(3)[1]/EBUS(3)-RX+	BD/LI+	E11	SYNC/LATCH[0]	BD
B2	PDI[28]/PERR(3)/TRANS(3)	BD	E12	SYNC/LATCH[1]	BD
B3	PDI[25]/TX_D(3)[0]	BD	F1	TX_ENA(2)/EBUS(2)-TX+	BD/LO+
B4	PDI[23]/TX_D(3)[2]	BD	F2	LINK_MII(2)/CLK25OUT1	BD
B5	PDI[21]/LINK_MII(3)	BD	F3	$V_{CC\ I/O}$ (T0)	
B6	PDI[19]/RX_D(3)[2]	BD	F4	Res. [6]	I
B7	PDI[17]/RX_CLK(3)	BD	F9	$GND_{I/O}$	
B8	PDI[15]	BD	F10	$V_{CC\ I/O}$	
B9	PDI[13]	BD	F11	EEPROM_DATA	BD
B10	PDI[9]	BD	F12	OSC_OUT	O
B11	PDI[7]/CPU_CLK	BD	G1	PDI[35]/RX_ERR(2)	BD
B12	PDI[4]	BD	G2	PDI[36]/RX_CLK(2)	BD
C1	PDI[31]/CLK25OUT2	BD	G3	Res. [1]	I
C2	PDI[30]/LINKACT(3)/P_CONF(3)	BD	G4	Res. [2]	I
C3	PERR(2)/TRANS(2)/PHYAD_OFF	BD	G9	GND_{PLL}	
C4	RBIAS		G10	$V_{CC\ PLL}$	
C5	$V_{CC\ I/O}$		G11	EEPROM_CLK	BD
C6	$V_{CC\ Core}$		G12	OSC_IN	I
C7	$V_{CC\ Core}$		H1	RX_DV(2)/EBUS(2)-RX-	I/LI-
C8	Res. [4]	I	H2	PDI[37]/RX_D(2)[0]	BD
C9	PDI[11]	BD	H3	TESTMODE	I
C10	PDI[5]	BD	H4	$GND_{I/O}$ (T1)	

Pin	Pin name	Dir.
C11	PDI[3]	BD
C12	PDI[2]	BD
D1	PDI[32]/TX_D(2)[3]	BD
D2	PDI[33]/TX_D(2)[2]	BD
D3	$V_{CC\ I/O}$	
D4	$GND_{I/O}$	
D5	$GND_{I/O}$	
D6	GND_{Core}	
J5	$GND_{I/O}$	
J6	GND_{Core}	
J7	GND_{Core}	
J8	$GND_{I/O}$	
J9	$GND_{I/O}$	
J10	$V_{CC\ I/O}$	
J11	PERR (0)/TRANS(0)/CLK_MODE[0]	BD
J12	LINKACT(0)/P_CONF[0]	BD
K1	PDI[39]/RX_D(2)[3]	BD
K2	PERR(1)/TRANS(1)/CLK_MODE(1)	BD
K3	LINK_MII(1)	I
K4	RX_CLK(1)	I
K5	$V_{CC\ I/O}$	
K6	$V_{CC\ Core}$	
K7	$V_{CC\ Core}$	
K8	$V_{CC\ I/O}$	
K9	$GND_{I/O}$ (T2)	
K10	RX_D(0)[0]	I
K11	MI_CLK/LINKPOL	BD
K12	MI_DATA	BD
L1	LINKACT(1)/P_CONF(1)	BD
L2	TX_D(1)[2]/P_MODE[0]	BD

Pin	Pin name	Dir.
H9	$V_{CC\ I/O}$ (T3)	
H10	Res. [5]	I
H11	RUN/EEPROM_SIZE	BD
H12	RESET	BD
J1	RX_D(2)[1]/EBUS(2)-RX+	I/LI+
J2	PDI[38]/RX_D(2)[2]	BD
J3	$V_{CC\ I/O}$	
J4	$GND_{I/O}$	
L3	TX_D(1)[0]/TRANS_MODE_ENA	BD
L4	RX_D(1)[0]	I
L5	RX_D(1)[2]	I
L6	RX_ERR(1)	I
L7	TX_D(0)[2]/C25_SHI[0]	BD
L8	TX_D(0)[0]/C25_ENA	BD
L9	LINK_MII(0)	I
L10	RX_CLK(0)	I
L11	RX_D(0)[2]	I
L12	RX_D(0)[3]	I
M1	TX_D(1)[3]/P_MODE[1]	BD
M2	TX_D(1)[1]/EBUS(1)-TX-	O/LO-
M3	TX_ENA(1)/EBUS(1)-TX+	BD/LO+
M4	RX_DV(1)/EBUS(1)-RX-	I/LI-
M5	RX_D(1)[1]/EBUS(1)-RX+	I/LI+
M6	RX_D(1)[3]	I
M7	TX_D(0)[3]/C25_SHI[1]	BD
M8	TX_D(0)[1]/EBUS(0)-TX-	O/LO-
M9	TX_ENA(0)/EBUS(0)-TX+	BD/LO+
M10	RX_ERR(0)	I
M11	RX_DV(0)/EBUS(0)-RX-	I/LI-
M12	RX_D(0)[1]/EBUS(0)-RX+	I/LI+

3.1.2 Signal Overview

Table 9: Signal Overview

Signal	Type	Dir.	Description
C25_ENA	Configuration	I	CLK25OUT2 Enable: Enable CLK25OUT2
C25_SHI[1:0]	Configuration	I	TX Shift: Shifting/phase compensation of MII TX signals
CLK_MODE[1:0]	Configuration	I	CPU_CLK configuration
CLK25OUT1/CLK25OUT2	MII	O	25 MHz clock source for Ethernet PHYs
CPU_CLK	PDI	O	Clock signal for µController
CTRL_STATUS_MOVE	Configuration	I	Move Digital I/O Control/Status signal to last available PDI byte
EBUS(3:0)-RX-	EBUS	LI-	EBUS LVDS receive signal -
EBUS(3:0)-RX+	EBUS	LI+	EBUS LVDS receive signal +
EBUS(3:0)-TX-	EBUS	LO-	EBUS LVDS transmit signal -
EBUS(3:0)-TX+	EBUS	LO+	EBUS LVDS transmit signal +
EEPROM_CLK	EEPROM	BD	EEPROM I^2C Clock
EEPROM_DATA	EEPROM	BD	EEPROM I^2C Data
EEPROM_SIZE	Configuration	I	EEPROM size configuration
PERR(3:0)	LED	O	Port receive error LED output (for testing)
GND_{Core}	Power		Core logic ground
$GND_{I/O}$	Power		I/O ground
GND_{PLL}	Power		PLL ground
LINK_MII(3:0)	MII	I	PHY signal indicating a link
LINKACT(3:0)	LED	O	Link/Activity LED output
LINKPOL	Configuration	I	LINK_MII(3:0) polarity configuration
MI_CLK	MII	O	PHY Management Interface clock
MI_DATA	MII	BD	PHY Management Interface data
OSC_IN	Clock	I	Clock source (crystal/oscillator)
OSC_OUT	Clock	O	Clock source (crystal)
P_CONF(3:0)	Configuration	I	Physical layer of logical ports
P_MODE[1:0]	Configuration	I	Number of physical ports and corresponding logical ports
PDI[39:0]	PDI	BD	PDI signal, depending on EEPROM content
PHYAD_OFF	Configuration	I	Ethernet PHY Address Offset
RBIAS	EBUS		BIAS resistor for LVDS TX current adjustment
Res. [7:0]	Reserved	I	Reserved pins
RESET	General	BD	Open collector Reset output/Reset input
RUN	LED	O	Run LED controlled by AL Status register
RX_CLK(3:0)	MII	I	MII receive clock
RX_D(3:0)[3:0]	MII	I	MII receive data
RX_DV(3:0)	MII	I	MII receive data valid
RX_ERR(3:0)	MII	I	MII receive error
SYNC/LATCH[1:0]	DC	I/O	Distributed Clocks SyncSignal output or LatchSignal input
TESTMODE	General	I	Reserved for testing, connect to GND
TRANS(3:0)	MII	I	MII interface sharing: share port enable
TRANS_MODE_ENA	Configuration	I	Enable MII interface sharing (and TRANS(3:0) signals)
TX_D(3:0)[3:0]	MII	O	MII transmit data
TX_ENA(3:0)	MII	O	MII transmit enable
$V_{CC\ Core}$	Power		Core logic power
$V_{CC\ I/O}$	Power		I/O power
$V_{CC\ PLL}$	Power		PLL power

3.1.3 PDI Signal Overview

Table 10: PDI signal overview

PDI	Signal	Dir.	Description
Digital I/O	EEPROM_LOADED	O	PDI is active, EEPROM is loaded
	I/O[31:0]	I/O/BD	Input/Output or Bidirectional data
	LATCH_IN	I	External data latch signal
	OE_CONF	I	Output Enable Configuration
	OE_EXT	I	Output Enable
	OUTVALID	O	Output data is valid/Output event
	SOF	O	Start of Frame
	WD_TRIG	O	Watchdog Trigger
SPI	EEPROM_LOADED	O	PDI is active, EEPROM is loaded
	SPI_CLK	I	SPI clock
	SPI_DI	I	SPI data MOSI
	SPI_DO	O	SPI data MISO
	SPI_IRQ	O	SPI interrupt
	SPI_SEL	I	SPI chip select
µC async.	CS	I	Chip select
	BHE	I	Byte High Enable (16 bit µController interface only)
	RD	I	Read command
	WR	I	Write command
	BUSY	O	EtherCAT device is busy
	IRQ	O	Interrupt
	EEPROM_LOADED	O	PDI is active, EEPROM is loaded
	DATA[7:0]	BD	Data bus for 8 bit µController interface
	ADR[15:0]	I	Address bus
	DATA[15:0]	BD	Data bus for 16 bit µController interface
µC sync.	ADR[15:0]	I	Address bus
	BHE	I	Byte High Enable
	CPU_CLK_IN	I	µController interface clock
	CS	I	Chip select
	DATA[15:0]	BD	Data bus for 16 Bit µController interface
	DATA[7:0]	BD	Data bus for 8 Bit µController interface
	EEPROM_LOADED	O	PDI is active, EEPROM is loaded
	IRQ	O	Interrupt
	RD/nWR	I	Read/Write access
	TA	O	Transfer Acknowledge
	TS	I	Transfer Start

3.2 Power Supply

The ET1100 supports different power supply and I/O voltage options with 3.3V I/O (or 5V I/O, not recommended) and optionally single or dual power supply.

The $V_{CCI/O}$ supply voltage directly determines the I/O voltages for all inputs and outputs, i.e., with 3.3V $V_{CCI/O}$, the inputs are 3.3V I/O compliant and they are not 5V tolerant ($V_{CCI/O}$ has to be 5V if 5V tolerant I/Os are required).

The core supply voltages $V_{CC\ Core}$/$V_{CC\ PLL}$ (nom. 2.5V) are generated from $V_{CC\ I/O}$ by an internal LDO. $V_{CC\ PLL}$ is always equal to $V_{CC\ Core}$. The internal LDO cannot be switched off, it stops operating if external supply voltage is higher than the internal LDO output voltage, thus the external supply voltage ($V_{CC\ Core}$/$V_{CC\ PLL}$) has to be higher (at least 0.1V) than the internal LDO output voltage.

Using the internal LDO increases power dissipation, and power consumption for 5V I/O voltage is significantly higher than power consumption for 3.3V I/O. It is highly recommended to use 3.3V I/O voltage and the internal LDO for $V_{CC\ Core}$/$V_{CC\ PLL}$.

Voltage stabilization capacitors at all power pairs are necessary.

Table 11: Power supply options (all voltages nominal)

$V_{CC\ I/O}$	$V_{CC\ Core}$/$V_{CC\ PLL}$	Input signals	Output signals	Comment
3.3V	Internal LDO (2.5V)	3.3V only	3.3V only	Single power supply, low power dissipation
Not recommended for future compatibility:				
3.3V	External 2.5V	3.3V only	3.3V only	Dual power supply, lowest power dissipation
5V	Internal LDO (2.5V)	5V only	5V only	Single power supply, highest power dissipation
5V	External 2.5V	5V only	5V only	Dual power supply, high power dissipation

3.2.1 I/O Power Supply

The I/O power supply pins can be connected to either 3.3V or 5.0V (5.0V not recommended), depending on the desired interface voltage. All power pins must be connected, and voltage stabilization capacitors at $V_{CCI/O}$/$GND_{I/O}$ power pairs are necessary.

Table 12: I/O power supply pins

Pin	Pin name
C5	$V_{CC\ I/O}$
D5	$GND_{I/O}$
D3	$V_{CC\ I/O}$
D4	$GND_{I/O}$
J3	$V_{CC\ I/O}$
J4	$GND_{I/O}$
K5	$V_{CC\ I/O}$
J5	$GND_{I/O}$
K8	$V_{CC\ I/O}$
J8	$GND_{I/O}$
J10	$V_{CC\ I/O}$
J9	$GND_{I/O}$
F10	$V_{CC\ I/O}$*
F9	$GND_{I/O}$*
D10	$V_{CC\ I/O}$
D9	$GND_{I/O}$
E9	$V_{CC\ I/O}$*
H4	$GND_{I/O}$ (T1)
F3	$V_{CC\ I/O}$ (T0)
K9	$GND_{I/O}$ (T2)
H9	$V_{CC\ I/O}$ (T3)

NOTE: The pins marked with * are most adjacent to the internal LDO – this should be taken into account for voltage stabilization.

The I/O power supply pins marked with T0-T3 are used as test pins for the ET1100-000x, but they must be connected to the listed supply voltages for future compatibility.

3.2.2 Logic Core Power Supply

Table 13 shows the pins for core power supply. Core supply voltage is 2.5V. The core power is either generated by the internal LDO, which is sourced by the I/O power supply, or externally. In both cases, voltage stabilization capacitors have to be connected to $V_{CC\ Core}$/GND_{Core} power pairs.

Table 13: Core Power Supply pins

Pin	Pin name
C6	$V_{CC\ Core}$
D6	GND_{Core}
K6	$V_{CC\ Core}$
J6	GND_{Core}
K7	$V_{CC\ Core}$
J7	GND_{Core}
C7	$V_{CC\ Core}$
D7	GND_{Core}

3.2.3 PLL Power Supply

Table 14 shows the pins for PLL power supply. PLL supply voltage is 2.5V. The PLL power is either generated by the internal LDO, which is sourced by the I/O power supply, or externally. In both cases, voltage stabilization capacitors have to be connected to $V_{CC\ PLL}$/GND_{PLL}.

Table 14: PLL Power Supply pins

Pin	Pin name
G10	$V_{CC\ PLL}$
G9	GND_{PLL}

3.2.4 Example schematics for power supply

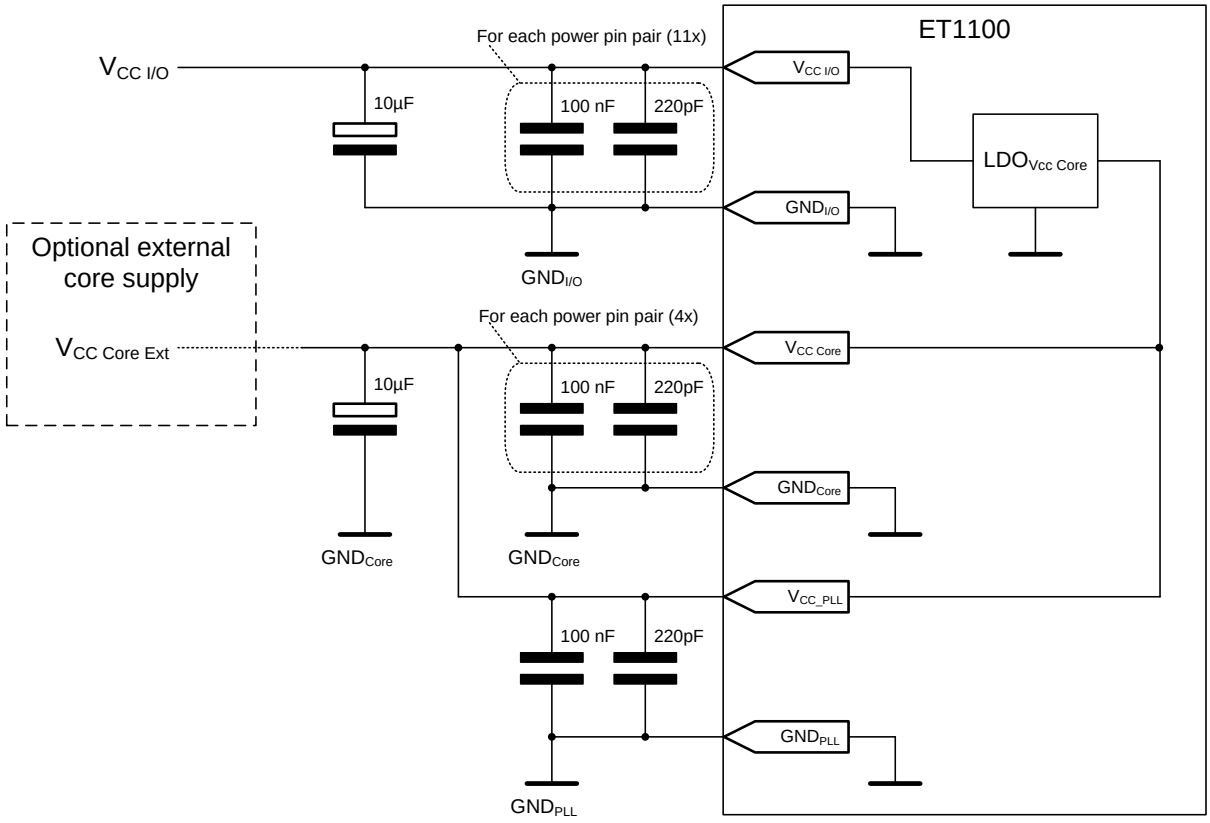

Figure 3: ET1100 power supply

Recommendation for voltage stabilization capacitors: 220pF and 100nF ceramic capacitors for each power pin pair, additional 10µF capacitor for $V_{CC\ I/O}$, and $V_{CC\ Core}$/$V_{CC\ PLL}$, i.e., a total of two 10µF capacitors.

$GND_{I/O}$, GND_{Core}, and GND_{PLL} can be connected to a single GND potential.

The internal LDO is self-deactivating if the actual $V_{CC\ Core}$/$V_{CC\ PLL}$ voltage is higher than the nominal LDO output voltage.

3.3 Clock Supply

Table 15: Clock supply pins

Pin	Pin		Signal		Configuration Signal	Internal PU/PD
	Name	Dir.	Signal	Dir.		
G12	OSC_IN	I	OSC_IN	I		
F12	OSC_OUT	O	OSC_OUT	O		

OSC_IN
Connection to external crystal or oscillator input (25 MHz). An oscillator as the clock source for both ET1100 and PHYs is mandatory if MII ports are used and CLK25OUT1/2 cannot be used as the clock source for the PHYs. The 25 MHz clock source should have an initial accuracy of 25ppm or better.

OSC_OUT
Connection to external crystal. Should be left open if an oscillator is connected to OSC_IN.

3.3.1 Example schematics for clock supply

The layout of the clock source has the biggest influence on EMC/EMI of a system design.

Although a clock frequency of 25 MHz requires not extensive design efforts, the following rules shall help to improve system performance:

- Keep clock source and ESC as close as possible close together.
- Ground Layer should be seamless in this area.
- Power supply should be of low impedance for clock source and ESC clock supply.
- Capacitors shall be used as recommended by the clock source component.
- Capacities between clock source and ESC clock supply should be in the same size (values depend upon geometrical form of board).

The initial accuracy of the ET1100 clock source has to be 25ppm or better.

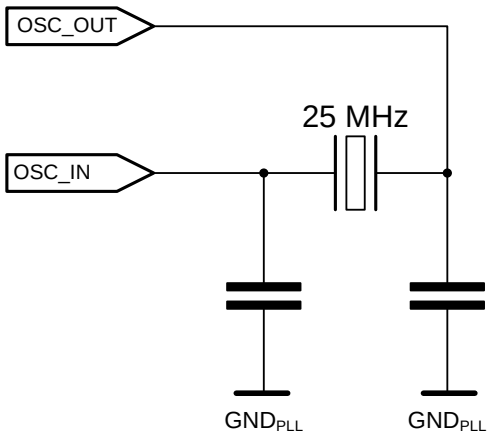

Figure 4: Quartz crystal connection

NOTE: The value of the load capacitors depends on the load capacitance of the crystal, the pin capacitance C_{OSC} of the ESC pins and the board design (typical 12pF each if C_L = 10pF).

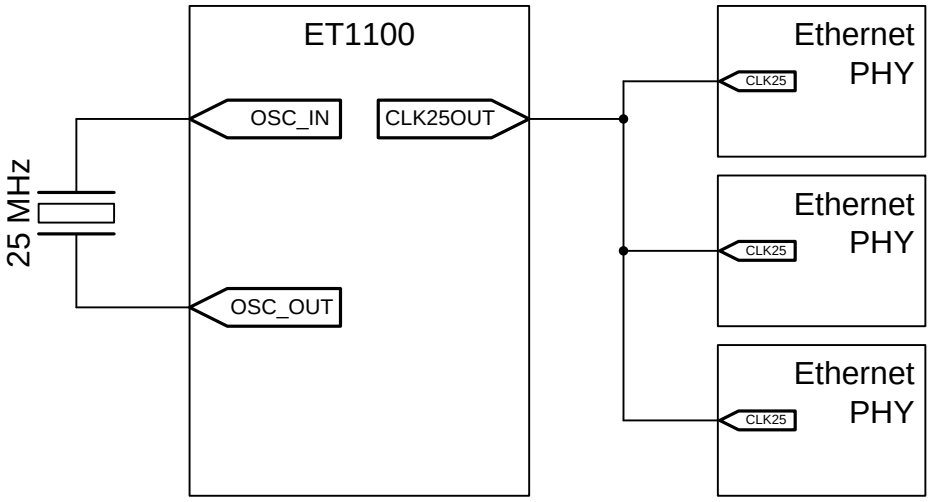

Figure 5: Quartz crystal Clock source for ET1100 and Ethernet PHYs

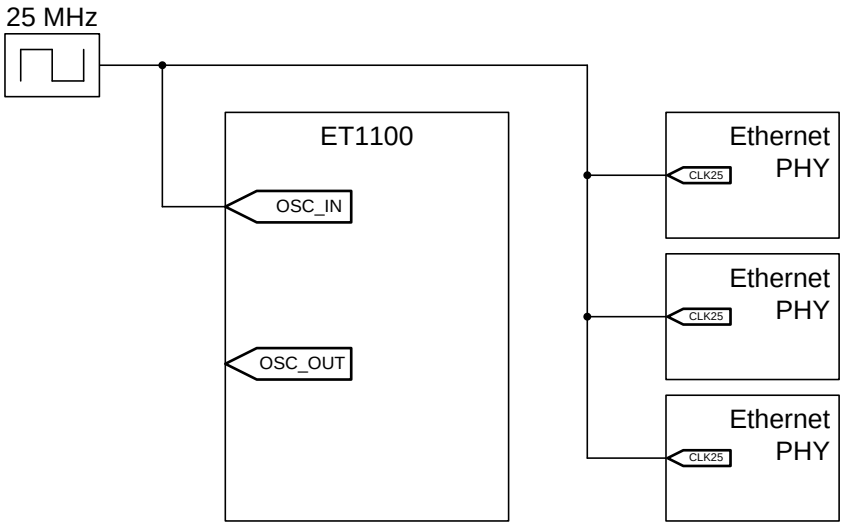

Figure 6: Oscillator clock source for ET1100 and Ethernet PHYs

3.4 Reset Pin

Table 16: Reset pin

Pin	Pin		Signal		Configuration Signal	Internal PU/PD
	Name	Dir.	Signal	Dir.		
H12	RESET	BD	RESET	BD		3.3 kΩ PU

RESET
The open collector RESET input/output (active low) signals the reset state of ET1100. The reset state is entered at power-on, if the power supply is to low, or if a reset was initiated using the reset register 0x0040. ET1100 also enters reset state if RESET pin is held low by external devices

3.4.1 Internal reset logic and Example schematic for RESET pin

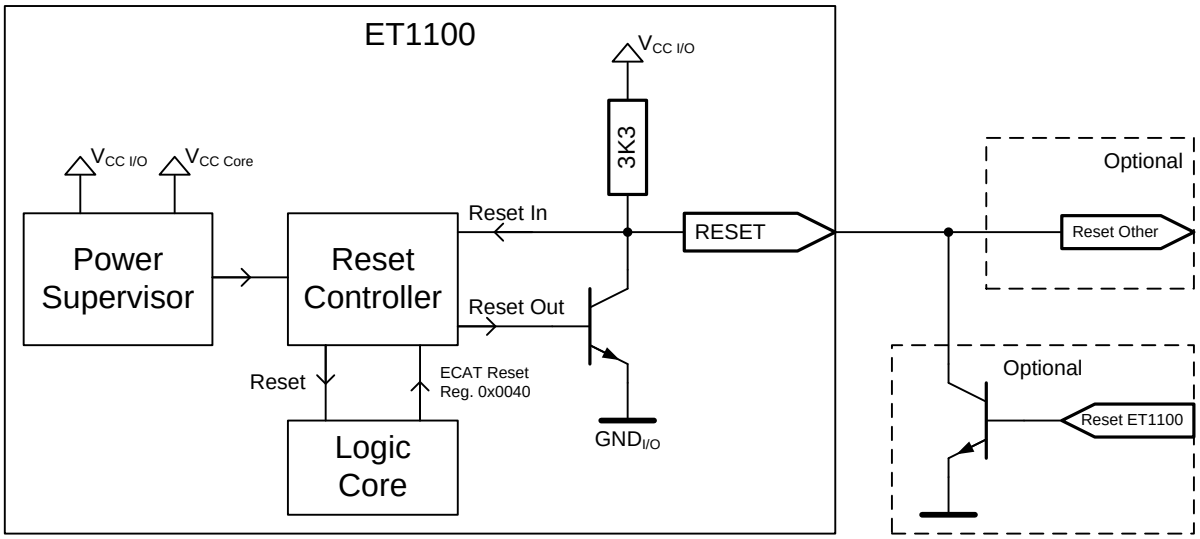

Figure 7: Reset Logic

It is recommended to connect the PHYs and the µController to the RESET pin. This makes sure that the PHYs are not communicating while the ET1200 is in reset (lost frames), and it allows for resetting the whole EtherCAT slave device via EtherCAT in case of an unintended condition.

3.5 RBIAS Pin

Table 17: RBIAS pin

Pin	Pin		Signal		Configuration Signal	Internal PU/PD
	Name	Dir.	Signal	Dir.		
C4	RBIAS		RBIAS			

RBIAS
Bias resistor for LVDS TX current adjustment, should be 11 kΩ connected to GND.

NOTE: If only MII ports are used (no EBUS at all), the RBIAS resistor can be selected in the range of 10-15 kΩ

3.5.1 Example schematic for RBIAS resistor

The LVDS RBIAS resistor should have a value of R_{BIAS}=11 kΩ.

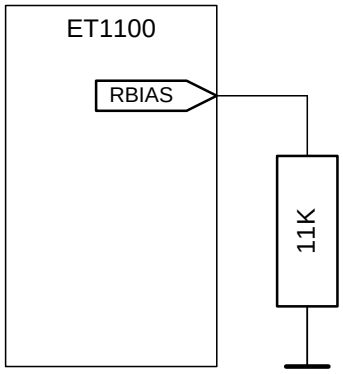

Figure 8: LVDS load resistor

.

3.6 Configuration Pins

The configuration pins are used to configure the ET1100 at power-on with pull-up or pull-down resistors. At power-on the ET1100 uses these pins as inputs to latch the configuration[2]. After power-on, the pins have their operation functionality which has been assigned to them, and therefore pin direction changes if necessary. The power-on phase finishes before the nRESET pin is released. In subsequent reset phases without power-on condition, the configuration pins still have their operation functionality, i.e., the ET1100 configuration is not latched again and output drivers remain active.

The configuration value 0 is realized by a pull-down resistor, a pull-up resistor is used for a 1. Since some configuration pins are also used as LED outputs, the polarity of the LED output depends on the configuration value.

3.6.1 Example schematics for configuration input/LED output pins

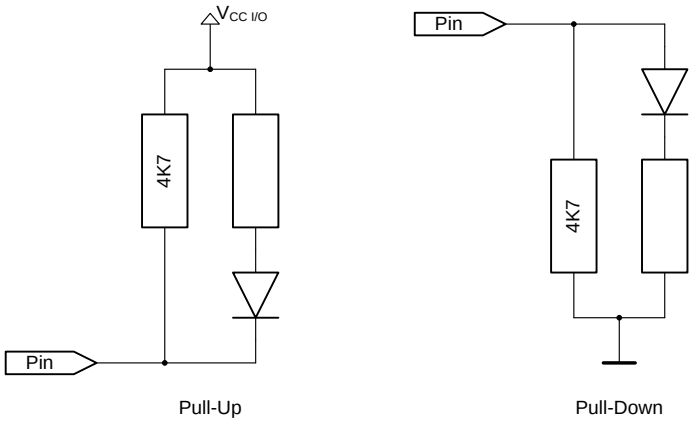

Figure 9: Dual purpose configuration input/LED output pins

[2] Take care of proper configuration: External devices attached to dual-purpose configuration pins might interfere sampling the intended configuration if they are e.g. not properly powered at the sample time (external device keeps configuration pin low although a pull-up resistor is attached). In such cases the ET1100 power-on value sampling time can be delayed by delaying power activation.

3.6.2 Port Mode

Port Mode configures the number of physical ports and the corresponding logical ports. It is shown in Table 18.

Table 18: Port Mode

Description	Config signal	Pin name	Register	P_MODE[1:0] Values
Port Mode	P_MODE[0]	TX_D(1)[2]/P_MODE[0]	0x0E00[0]	00 = 2 ports (log. ports 0 and 1) 01 = 3 ports (log. ports 0,1, and 2) 10 = 3 ports (log. ports 0,1, and 3) 11 = 4 ports (log. ports 0, 1, 2, and 3)
	P_MODE[1]	TX_D(1)[3]/P_MODE[1]	0x0E00[1]	

NOTE: The term physical port in this document is only used for grouping ET1100 interface pins. The register set as well as any master/slave software is always based on logical ports. The distinction between physical and logical ports is made in order to increase the number of available PDI pins. Each logical port is associated with exactly one physical port, and it can be configured to be either EBUS or MII.

MII ports are always assigned to the lower physical ports, then EBUS ports are assigned. If any MII ports are configured, the lowest logical MII port is always connected to physical port 0, the next higher logical MII port is connected to physical port 1, and so on. Afterwards, the lowest logical EBUS port – if configured – is connected to the next physical port following the physical MII ports, i.e. port [number of MII ports]. Without MII ports, the EBUS ports are connected beginning with physical port 0.

If only EBUS or only MII ports are used, the physical port number is the same as the logical port number for P_MODE[1:0]=00, 01 or 11. Refer to the next chapter for more details.

3.6.3 Port Configuration

P_CONF[3:0] determines the physical layer configuration (MII or EBUS). P_CONF[0] determines the physical layer of logical port 0, P_CONF[1] determines logical port 1, P_CONF[2] determines the physical layer of the next available logical port (either 3 for P_MODE[1:0]=10, else 2), and P_CONF[3] determines logical port 3. If a physical port is not used, the corresponding P_CONF configuration signal is not used.

Table 19: Port Configuration

Description	Configuration signal	Pin name	Register	Values
Port Configuration	P_CONF[0]	LINKACT(0)/P_CONF[0]	0x0E00[2]	0 = EBUS 1 = MII
	P_CONF[1]	LINKACT(1)/P_CONF(1)	0x0E00[3]	
	P_CONF[2]	LINKACT(2)/P_CONF[2]	0x0E00[4]	
	P_CONF[3]	PDI[30]/LINKACT(3)/P_CONF(3)	0x0E00[5]	

3.6.3.1 Configurations with 2 ports

For configurations with 2 ports, logical ports 0 and 1 are used. The port signals are available at physical ports 0 and 1, depending on the port configuration. P_MODE[1:0] has to be set to 00. P_CONF[1:0] determine the physical layer of logical ports (1:0). P_CONF[3:2] are not used, nevertheless, P_ CONF[2] should not be left open (connection to GND recommended). P_CONF[3] should be pulled down if possible (denoted with '-' in the table), if your application allows this.

Table 20: Configurations with 2 ports (P_MODE[1:0]=00)

Logical port		Physical port		P_CONF [3:0]
1	0	1	0	
$EBUS_{(1)}$	$EBUS_{(0)}$	$EBUS_{(1)}$	$EBUS_{(0)}$	-000
$EBUS_{(1)}$	$MII_{(0)}$	$EBUS_{(1)}$	$MII_{(0)}$	-001
$MII_{(1)}$	$EBUS_{(0)}$	$EBUS_{(0)}$	$MII_{(1)}$	-010
$MII_{(1)}$	$MII_{(0)}$	$MII_{(1)}$	$MII_{(0)}$	-011

3.6.3.2 Configurations with 3 ports

For configurations with 3 ports, either logical ports 0, 1, and 2 (P_MODE[1:0]=01) or logical ports 0, 1, and 3 (P_MODE[1:0]=10) are used. The port signals are available at physical ports 0, 1 and 2, depending on the port configuration. P_CONF[2:0] determine the physical layer of logical ports 2, 1, 0, or logical ports 3, 1, 0, depending on the P_MODE settings (P_CONF[2] is either used for logical port 2 or logical port 3). P_CONF[3] should be pulled down if possible (denoted with '-' in the tables), if your application allows this.

Table 21: Configurations with 3 ports (ports 0,1, and 2; P_MODE[1:0]=01)

Logical port			Physical port			P_CONF [3:0]
2	1	0	2	1	0	
$EBUS_{(2)}$	$EBUS_{(1)}$	$EBUS_{(0)}$	$EBUS_{(2)}$	$EBUS_{(1)}$	$EBUS_{(0)}$	-000
$EBUS_{(2)}$	$EBUS_{(1)}$	$MII_{(0)}$	$EBUS_{(2)}$	$EBUS_{(1)}$	$MII_{(0)}$	-001
$EBUS_{(2)}$	$MII_{(1)}$	$EBUS_{(0)}$	$EBUS_{(2)}$	$EBUS_{(0)}$	$MII_{(1)}$	-010
$EBUS_{(2)}$	$MII_{(1)}$	$MII_{(0)}$	$EBUS_{(2)}$	$MII_{(1)}$	$MII_{(0)}$	-011
$MII_{(2)}$	$EBUS_{(1)}$	$EBUS_{(0)}$	$EBUS_{(1)}$	$EBUS_{(0)}$	$MII_{(2)}$	-100
$MII_{(2)}$	$EBUS_{(1)}$	$MII_{(0)}$	$EBUS_{(1)}$	$MII_{(2)}$	$MII_{(0)}$	-101
$MII_{(2)}$	$MII_{(1)}$	$EBUS_{(0)}$	$EBUS_{(0)}$	$MII_{(2)}$	$MII_{(1)}$	-110
$MII_{(2)}$	$MII_{(1)}$	$MII_{(0)}$	$MII_{(2)}$	$MII_{(1)}$	$MII_{(0)}$	-111

Table 22: Configurations with 3 ports (ports 0, 1, and 3; P_MODE[1:0]=10)

Logical port			Physical port			P_CONF [3:0]
3	1	0	2	1	0	
$EBUS_{(3)}$	$EBUS_{(1)}$	$EBUS_{(0)}$	$EBUS_{(3)}$	$EBUS_{(1)}$	$EBUS_{(0)}$	-000
$EBUS_{(3)}$	$EBUS_{(1)}$	$MII_{(0)}$	$EBUS_{(3)}$	$EBUS_{(1)}$	$MII_{(0)}$	-001
$EBUS_{(3)}$	$MII_{(1)}$	$EBUS_{(0)}$	$EBUS_{(3)}$	$EBUS_{(0)}$	$MII_{(1)}$	-010
$EBUS_{(3)}$	$MII_{(1)}$	$MII_{(0)}$	$EBUS_{(3)}$	$MII_{(1)}$	$MII_{(0)}$	-011
$MII_{(3)}$	$EBUS_{(1)}$	$EBUS_{(0)}$	$EBUS_{(1)}$	$EBUS_{(0)}$	$MII_{(3)}$	-100
$MII_{(3)}$	$EBUS_{(1)}$	$MII_{(0)}$	$EBUS_{(1)}$	$MII_{(3)}$	$MII_{(0)}$	-101
$MII_{(3)}$	$MII_{(1)}$	$EBUS_{(0)}$	$EBUS_{(0)}$	$MII_{(3)}$	$MII_{(1)}$	-110
$MII_{(3)}$	$MII_{(1)}$	$MII_{(0)}$	$MII_{(3)}$	$MII_{(1)}$	$MII_{(0)}$	-111

3.6.3.3 Configurations with 4 ports

For configurations with 4 ports, logical ports 0 to 3 are used. The port signals are available at physical ports 0 to 3, depending on the port configuration. P_MODE[1:0] has to be set to 11. P_CONF[3:0] determine the physical layer of logical ports (3:0).

Table 23: Configurations with 4 ports (P_MODE[1:0]=01)

Logical port				Physical port				P_CONF [3:0]
3	2	1	0	3	2	1	0	
EBUS(3)	EBUS(2)	EBUS(1)	EBUS(0)	EBUS(3)	EBUS(2)	EBUS(1)	EBUS(0)	0000
EBUS(3)	EBUS(2)	EBUS(1)	MII(0)	EBUS(3)	EBUS(2)	EBUS(1)	MII(0)	0001
EBUS(3)	EBUS(2)	MII(1)	EBUS(0)	EBUS(3)	EBUS(2)	EBUS(0)	MII(1)	0010
EBUS(3)	EBUS(2)	MII(1)	MII(0)	EBUS(3)	EBUS(2)	MII(1)	MII(0)	0011
EBUS(3)	MII(2)	EBUS(1)	EBUS(0)	EBUS(3)	EBUS(1)	EBUS(0)	MII(2)	0100
EBUS(3)	MII(2)	EBUS(1)	MII(0)	EBUS(3)	EBUS(1)	MII(2)	MII(0)	0101
EBUS(3)	MII(2)	MII(1)	EBUS(0)	EBUS(3)	EBUS(0)	MII(2)	MII(1)	0110
EBUS(3)	MII(2)	MII(1)	MII(0)	EBUS(3)	MII(2)	MII(1)	MII(0)	0111
MII(3)	EBUS(2)	EBUS(1)	EBUS(0)	EBUS(2)	EBUS(1)	EBUS(0)	MII(3)	1000
MII(3)	EBUS(2)	EBUS(1)	MII(0)	EBUS(2)	EBUS(1)	MII(3)	MII(0)	1001
MII(3)	EBUS(2)	MII(1)	EBUS(0)	EBUS(2)	EBUS(0)	MII(3)	MII(1)	1010
MII(3)	EBUS(2)	MII(1)	MII(0)	EBUS(2)	MII(3)	MII(1)	MII(0)	1011
MII(3)	MII(2)	EBUS(1)	EBUS(0)	EBUS(1)	EBUS(0)	MII(3)	MII(2)	1100
MII(3)	MII(2)	EBUS(1)	MII(0)	EBUS(1)	MII(3)	MII(2)	MII(0)	1101
MII(3)	MII(2)	MII(1)	EBUS(0)	EBUS(0)	MII(3)	MII(2)	MII(1)	1110
MII(3)	MII(2)	MII(1)	MII(0)	MII(3)	MII(2)	MII(1)	MII(0)	1111

3.6.4 CPU_CLK MODE

CLK_MODE is used to provide a clock signal to an external microcontroller. If CLK_MODE is not 00, CPU_CLK is available on PDI[7], thus this pin is not available for PDI signals anymore. For µController PDIs, PDI[7] is ADR[15], which is treated to be 0 if CPU_CLK is selected. The CPU_CLK MODE is shown in Table 24.

Table 24: CPU_CLK Mode

Description	Config signal	Pin name	Register	Values
CPU_CLK_MODE	CLK_MODE[0]	PERR(0)/ TRANS(0)/ CLK_MODE[0]	0x0E00[6]	00 = off, PDI[7]/CPU_CLK available for PDI 01 = 25 MHz clock output at PDI[7]/CPU_CLK 10 = 20 MHz clock output at PDI[7]/CPU_CLK 11 = 10 MHz clock output at PDI[7]/CPU_CLK
	CLK_MODE[1]	PERR(1)/ TRANS(1)/ CLK_MODE(1)	0x0E00[7]	

3.6.5 TX Shift

Phase shift (0/10/20/30ns) of MII TX signals (TX_ENA, TX_D[3:0]) can be attained via the C25_SHI[x] signals. TX-Shift is explained in Table 25. It is recommended to support all C25_SHI[1:0] configurations by hardware options to enable later adjustments.

Table 25: TX Shift

Description	Config signal	Pin name	Register	Values
TX Shift	C25_SHI[0]	TX_D(0)[2]/C25_SHI[0]	0x0E01[0]	00 = MII TX signals not delayed 01 = MII TX signals delayed by 10 ns 10 = MII TX signals delayed by 20 ns 11 = MII TX signals delayed by 30 ns
	C25_SHI[1]	TX_D(0)[3]/C25_SHI[1]	0x0E01[1]	

3.6.6 CLK25OUT2 Enable

A 25MHz clock for Ethernet PHYs can be made available by the ET1100 on PDI[31]/CLK25OUT2 pin. This is only relevant if three MII ports are used. In cases with less than 3 MII ports, pin LINK_MII(2)/CLK25OUT1 provides CLK25OUT anyway, because LINK_MII(2) is not used. If 4 MII ports are used, PDI[31]/CLK25OUT2 provides CLK25OUT2 regardless of CLK25OUT2 Enable. CLK25OUT2 Enable is explained in Table 26.

Table 26: CLK25OUT2 Enable

Description	Config signal	Pin name	Register	Values
CLK25OUT2 Enable	C25_ENA	TX_D(0)[0]/C25_ENA	0x0E01[2]	0 = disable, PDI[31]/CLK25OUT2 is available for PDI 1 = enable, PDI[31]/CLK25OUT2 is 25 MHz clock output

3.6.7 Transparent Mode Enable

The ET1100 is capable of sharing the MII interfaces with other MACs on a per port basis. Typically, the Transparent mode is disabled, and the ET1100 has exclusive access to the MII interfaces of the PHYs. With the Transparent mode turned on, the MII interfaces can be assigned either to the ET1100 or to other MACs, e.g., µControllers with integrated MACs. Reassignment is not meant to be done whilst network traffic is processed.

The Transparent mode primarily affects the PERR(x)/TRANS(x) signals. If Transparent mode is enabled, PERR(x)/TRANS(x) becomes TRANS(x) (active low), which controls the transparent state of each port. PERR(x) is not available in Transparent mode.

TRANS(x) does only affect the TX_ENA(x)/TX_D(x) signals of the same port as well as MI_CLK/MI_DATA. RX_CLK(x), RX_DV(x), RX_D(x), and RX_ERR(x) are connected to both ET1100 and the other MAC.

Each MII interface behaves as usual as long as TRANS(x) is high, and the ET1100 controls the MII interface. If TRANS(x) is low, the port becomes transparent (or isolated), i.e., the ET1100 will no longer drive TX_ENA(x)/TX_D(x) actively, thus, the other MAC can drive these signals.

The Link/Act(x) LED will still be driven by the ET1100, because it samples RX_DV(x) and TX_ENA(x) (which becomes an input while a port is transparent) for detection of activity.

As long as at least one MII interface is not transparent, the ET1100 is in control of the MII management interface. With the Transparent mode turned on, the PHY management interface of the ET1100 can be accessed via the PDI interface, so a µController gets access to the management interface. If all MII interfaces are transparent, the ET1100 releases MI_CLK and MI_DATA drivers, so they can be driven by the other MAC.

Refer to example schematic for more details.

Table 27: Transparent Mode Enable

Description	Config signal	Pin name	Register	Values
Transparent Mode Enable	TRANS_MODE_ENA	TX_D(1)[0]/ TRANS_MODE_ENA	0x0E01[3]	0 = normal mode/Transparent mode disabled. ET1100 uses PHY exclusively 1 = Transparent mode enabled, ET1100 can share PHY with other MACs

NOTE: Enabling transparent mode disables Link Polarity configuration to active high.

3.6.7.1 Example schematic for Transparent Mode

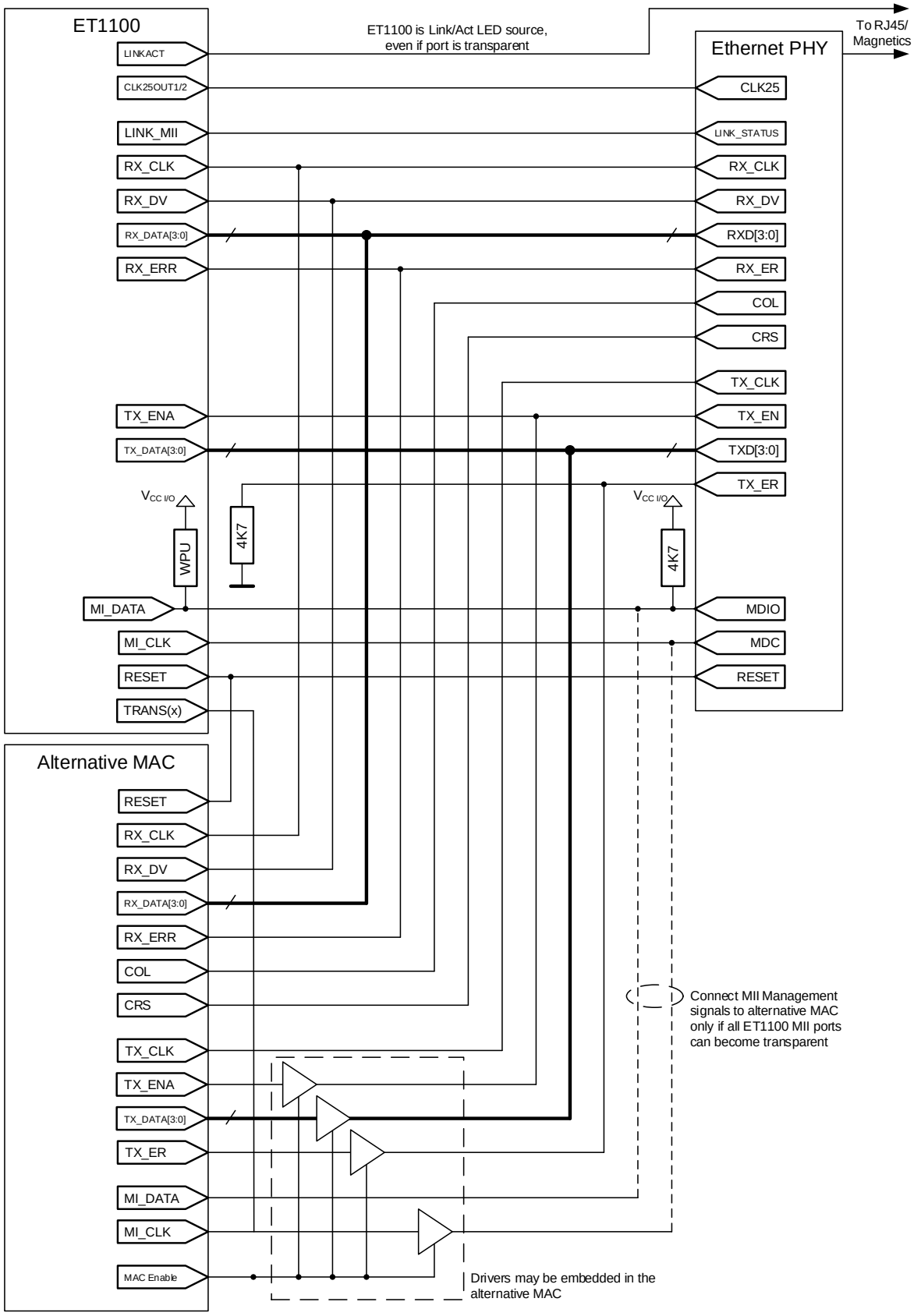

Figure 10: Transparent Mode

NOTE: MI_DATA outputs of alternative MAC have to be high-Z if ET1100 is controlling PHY management interface, otherwise add driver (like MI_CLK). Check alternative MAC's TX timings when extra drivers are used.

3.6.8 Digital Control/Status Move

If more than 2 MII ports are used (PDI[39:32] are not available for PDI use), the Digital I/O PDI control and status signals can be made available at the highest available PDI byte with CTRL_STATUS_MOVE.

Digital Control/Status Move is explained in Table 28:

Table 28: Digital Control/Status Move

Description	Config signal	Pin name	Register	Values
Digital Control/ Status Move	CTRL_STATUS_MOVE	PDI[34]/TX_D(2)[0]/ CTRL_STATUS_MOVE	0x0E01[4]	0 = Digital I/O control/status signals are not moved: they are available at PDI[39:32] if less than 3 MII ports are used, otherwise they are not available 1 = Digital I/O control/status signals moved to last PDI byte if PDI[39:32] is used for MII(2). Digital I/O control/status signals are available in any configuration.

3.6.9 PHY Address Offset

The ET1100 supports two PHY address offset configurations, either 0 or 16. Refer to chapter 4.2 for details on PHY address configuration.

PHY Address Offset is explained in Table 29:

Table 29: PHY Address Offset

Description	Config signal	Pin name	Register	Values
PHY Address Offset	PHYAD_OFF	PERR(2)/TRANS(2)/ PHYAD_OFF	0x0E01[5]	0 = PHY address offset 0 1 = PHY address offset 16

3.6.10 Link Polarity

Ethernet PHYs signal a 100 Mbit/s Full (Duplex Link(to the ET1100 by asserting LINK_MII(x). The polarity can be selected with LINKPOL.

Link Polarity is explained in Table 30:

Table 30: Link Polarity

Description	Config signal	Pin name	Register	Values
Link Polarity	LINKPOL	MI_CLK/LINKPOL	0x0E01[6]	0 = LINK_MII(x) is active low 1 = LINK_MII(x) is active high
Reserved	RESERVED	PDI[28]/PERR(3)/TRANS(3)	0x0E01[7]	reserved

NOTE: Enabling transparent mode disables Link Polarity configuration to active high

3.6.11 SII EEPROM Size

EEPROM_SIZE determines the size of the EEPROM (and the number of I²C address bytes). EEPROM_SIZE is sampled at the beginning of the EEPROM access. EEPROM_SIZE is shown in Table 31:

Table 31: SII EEPROM_SIZE

Description	Config signal	Pin name	Register	Values
E²PROM Size	EEPROM_SIZE	RUN/EEPROM_SIZE	0x0502[7]	0 = 1 address byte (1 Kbit to 16 Kbit EEPROM) 1 = 2 address bytes (32 Kbit to 4 Mbit EEPROM)

3.6.12 Reserved Configuration Pins

The reserved configuration pin should be pulled down when 4 ports are used. Otherwise it should be left open. It is shown in Table 32:

Table 32: Reserved

Description	Config signal	Pin name	Register	Values
Reserved	RESERVED	PDI[28]/PERR(3)/TRANS(3)	0x0E01[7]	0 for 4 port configurations

3.7 SII EEPROM Interface Pins

Table 33: SII EEPROM pins

Pin	Pin		Signal		Configuration Signal	Internal PU/PD
	Name	Dir.	Signal	Dir.		
G11	EEPROM_CLK	BD	EEPROM_CLK	BD		3.3 kΩ PU
F11	EEPROM_DATA	BD	EEPROM_DATA	BD		3.3 kΩ PU

EEPROM_CLK
EEPROM I²C clock signal (open collector output).

EEPROM_DATA
EEPROM I²C data signal (open collector output).

3.8 MII Management Pins

The MII Management signals are only used if at least one MII port is configured.

Table 34: MII Management pins

Pin	Pin		No MII port used		MII port(s) used		Configuration Signal	Internal PU/PD
	Name	Dir.	Signal	Dir.	Signal	Dir.		
K11	MI_CLK/LINKPOL	BD		UI	MI_CLK	O	LINKPOL	WPD
K12	MI_DATA	BD		UI	MI_DATA	BD		WPU

MI_CLK/LINKPOL
During power on LINK Polarity configuration during power-up, PHY Management Interface clock afterwards.

MI_DATA
PHY Management Interface Data.

NOTE: MI_DATA must have a pull-up resistor (4.7 kΩ recommended for ESCs).

3.9 Distributed Clocks SYNC/LATCH Pins

Table 35: DC SYNC/LATCH pins

Pin	Pin		Signal		Configuration Signal	Internal PU/PD
	Name	Dir.	Signal	Dir.		
E11	SYNC/LATCH[0]	BD	SYNC[0]/ LATCH[0]	O/ I		
E12	SYNC/LATCH[1]	BD	SYNC[1]/ LATCH[1]	O/ I		

SYNC/LATCH[x]
Distributed Clocks SyncSignal output or LatchSignal input, depending on SII EEPROM configuration. SYNC/LATCH signals are not driven (high impedance) until the EEPROM is loaded.

3.10 LED Signals

All LED signals are also used as configuration signals. The polarity of each LED signal depends on the configuration: LED is active high if pin is pulled down for configuration, and active low if pin is pulled up. Refer to the chapter 3.6.1 for LED connection details.

Table 36: LED pins

Pin	Pin		Signal		Configuration Signal	Internal PU/PD
	Name	Dir.	Signal	Dir.		
H11	RUN/EEPROM_SIZE	BD	RUN	O	EEPROM_SIZE	

NOTE: The pin locations for LINKACT(x) and PERR(x)/TRANS(x) are described in the Physical Port 0-3 chapters.

RUN/EEPROM_SIZE
SII EEPROM Size configuration (either 1 Kbit-16 Kbit or 32 KBit-4 Mbit) sampled at the beginning of the EEPROM access. Otherwise RUN LED signal. RUN is active high if pin is pulled down, and active low if pin is pulled up. Refer to chapter 3.6.1 for connection details. RUN LED should be green.

LINKACT(x)
Link/Activity LED output (off=no link, on=link without activity, blinking=link and activity) for physical port x. LINKACT(x) is active high if pin is pulled down, and active low if pin is pulled up. Refer to chapter 3.6.1 for connection details. Link/Activity LED should be green.

PERR(x)/TRANS(x)
Error LED output of physical port x for EBUS ports, and for MII ports if TRANS_MODE_ENA=0. If TRANS_MODE_ENA=1, PERR(x)/TRANS(x) is used as TRANS(x) for MII physical port x, which puts port x into isolate/transparent operation. PERR(x) is not available in this case. PERR(x) is active high if pin is pulled down, and active low if pin is pulled up. Refer to chapter 3.6.1 for connection details.

NOTE: PERR(x) LEDs are not part of the EtherCAT indicator specification. They are only intended for testing and debugging. The PERR(x) LED flashes once if a physical layer receive error occurs. Do not confuse PERR(x) LEDs with application layer ERR LED, this is not supported by the ESCs and has to be controlled by a µController.

3.11 Physical Ports and PDI Pins

The ET1100 pin out is optimized in order to achieve an optimum of size and features. To obtain this, there is a number of pins where either communication or PDI functionality can be assigned to. Number and type of the communication ports might reduce/exclude one or more PDI possibilities.

The physical communication ports are numbered from port 0 to port 3. Port 0 and port 1 do not interfere with PDI pins, while port 2 and port 3 might overlap with PDI[39:16] and therefore limit the number of choices for the PDI.

Pin configuration for ports will overwrite pin configuration for PDI. Therefore, number and type of ports should be configured first.

The ET1100 has 40 PDI pins, PDI[39:0]. They are structured in 4 groups: PDI[15:0] (PDI byte 0/1), PDI[16:23] (PDI byte 2), PDI[24:31] (PDI byte 3), and PDI[32:39] (PDI byte 4).

Possible Physical Port / PDI combinations

Table 37: Combinations of physical ports and PDI

	Async. µC	Sync. µC	SPI	Digital I/O with CTLR_STATUS_MOVE=	
				0	1
2 ports or 3 ports with min. 1xEBUS	8 Bit 16Bit	8 Bit 16Bit	SPI +32 Bit GPI/O	32Bit I/O +control/status signals	
3xMII, 0xEBUS	8Bit	8Bit	SPI +24 Bit GPI/O	32Bit I/O	24Bit I/O + control/status signals
4 ports, min. 2xEBUS	-	-	SPI +16Bit GPI/O	24Bit I/O + control/status signals	
3xMII, 1xEBUS	-	-	SPI +16Bit GPI/O	24 Bit I/O	16Bit I/O + control/status signals
4xMII	-	-	SPI +8Bit GPI/O	16Bit I/O	8Bit I/O + control/status signals

3.11.1 MII Signals

LINK_MII(x)
Input signal provided by the PHY if a 100 Mbit/s (Full Duplex) link is established. LINK_MII(x) polarity is configurable.

RX_CLK(x)
MII Receive Clock

RX_DV(x)
MII receive data valid.

RX_D(x)[3:0]
MII receive data.

RX_ERR(x)
MII receive error.

TX_ENA(x)
MII transmit enable output. Used as MII transmit enable input for controlling the Link/Activity LED if port is in transparent mode (TRANS_MODE_ENA=1 and TRANS(x)=0).

TX_D(x)[3:0]
MII transmit data.

3.11.1.1 CLK25OUT1/2 Signals

The ET1100 has to provide the Ethernet PHYs with a 25 MHz clock signal (CLK25OUT) if a 25 MHz crystal is used for clock generation. In case a 25 MHz oscillator is used, CLK25OUT is not necessary, because Ethernet PHYs and ET1100 can share the oscillator output. Depending on the port configuration and C25_ENA, CLK25OUT is available at different pins:

Table 38: CLK25OUT1/2 signal output

Conf.	C25_ENA=0	C25_ENA=1
0-2xMII	LINK_MII(2)/CLK25OUT1 provides CLK25OUT (PDI[31]/CLK25OUT2 also provides CLK25OUT if 4 ports are used)	LINK_MII(2)/CLK25OUT1 and PDI[31]/CLK25OUT2 provide CLK25OUT
3xMII	CLK25OUT not available, oscillator is mandatory	PDI[31]/CLK25OUT2 provides CLK25OUT
4xMII	PDI[31]/CLK25OUT2 provides CLK25OUT	

NOTE: Unused CLK25OUT pins should not be connected to reduce driver load.

The CLK25OUT pins provide a clock signal – if configured – during external or ECAT reset, clock output is only turned off during power-on reset.

3.11.1.2 Example schematic for MII connection

Refer to chapter 3.11.1 for more information on special markings (!). Take care of proper configuration of TX Shift, LINK_POL, and PHY addresses.

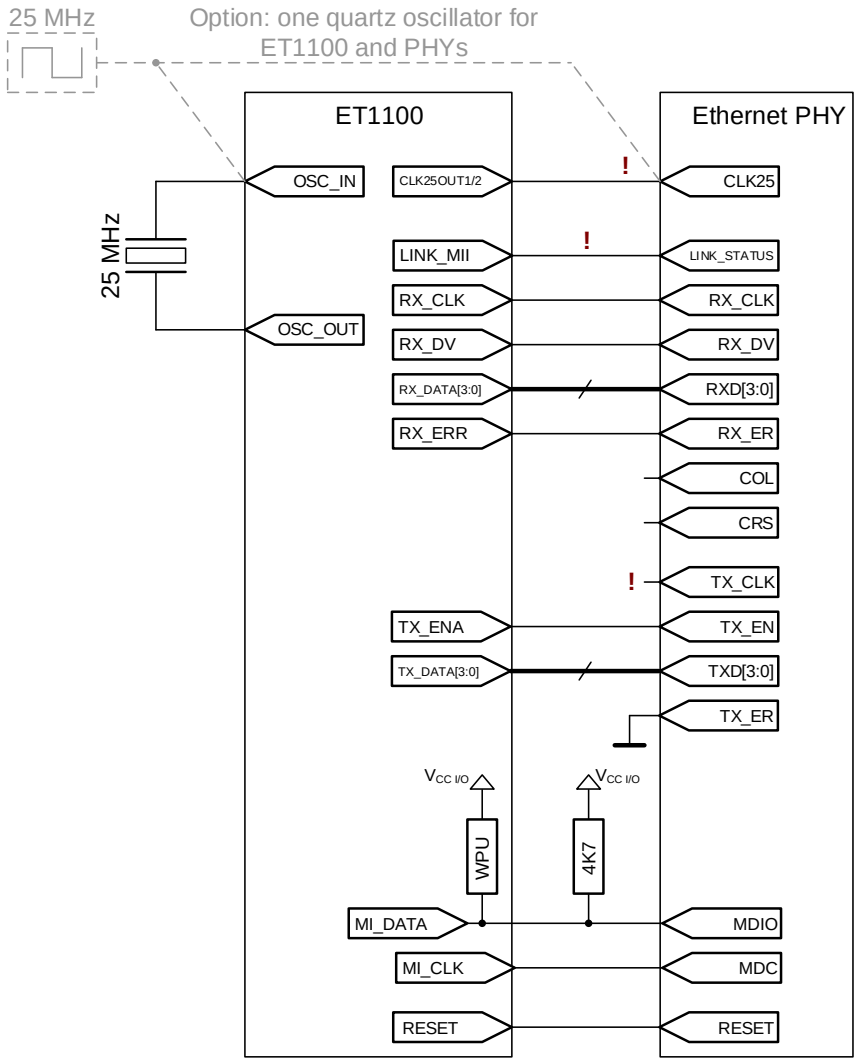

Figure 11: PHY Connection

3.11.2 EBUS Signals

The EBUS ports of the ET1100 are open failsafe, i.e., the ET1100 detects if an EBUS port is unconnected and closes the port internally (no physical link).

EBUS(x)-RX+/EBUS(x)-RX-
EBUS LVDS receive signals. EBUS_RX+ pins incorporate a pull-down resistor R_{LI+} and EBUS_RX- pins incorporate a pull-up resistor R_{LI-}, even if the pins are not configured for EBUS.

EBUS(x)-TX+/EBUS(x)-TX-
EBUS LVDS transmit signals.

3.11.2.1 Example schematic for EBUS termination

The LVDS termination with an impedance of 100 Ω is typically achieved by a resistor R_L=100 Ω. It is only necessary for EBUS ports and should be placed adjacent to the EBUS_RX inputs.

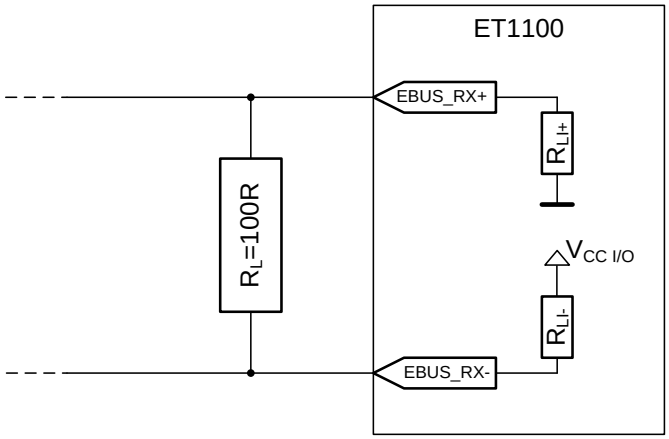

Figure 12: LVDS termination

3.11.3 PDI Pins

PDI[x]
The function of PDI[x] signals depends on the configuration stored in the device SII EEPROM. PDI signals are not driven (high impedance) until the EEPROM is loaded. This has to be taken into account especially for Digital Outputs.
PDI signals are not driven (high impedance) if no PDI is configured (PDI Control register 0x0140=0x00).

CPU_CLK
The ET1100 can provide a clock signal for µControllers on pin PDI[7]/CPU_CLK. The CPU_CLK output setting is controlled by the CLK_MODE configuration pin. If CPU_CLK is enabled, PDI[7] is not available for the PDI, i.e., ADR[15] cannot be used by µController PDIs (ADR[15] is treated to be 0 internally), and I/O[7] is not available for Digital I/O PDIs.

CPU_CLK provides a clock signal – if configured – during external or ECAT reset, clock output is only turned off during power-on reset.

3.11.4 Physical Port 0

Table 39 shows the pins for physical port 0. It can be configured as MII or EBUS and is always available. Use of this port does in no case clash with pins needed for PDI.

Table 39: Physical Port 0

Pin	Pin		MII		EBUS		Configuration Signal	Internal PU/PD
	Name	Dir.	Signal	Dir.	Signal	Dir.		
M9	TX_ENA(0)/ EBUS(0)-TX+	BD/LO+	TX_ENA(0)	O/I	EBUS(0)-TX+	LO+		
L8	TX_D(0)[0]/ C25_ENA	BD	TX_D(0)[0]	O			C25_ENA	
M8	TX_D(0)[1]/ EBUS(0)-TX-	O/LO-	TX_D(0)[1]	O	EBUS(0)-TX-	LO-		
L7	TX_D(0)[2]/ C25_SHI[0]	BD	TX_D(0)[2]	O			C25_SHI[0]	
M7	TX_D(0)[3]/ C25_SHI[1]	BD	TX_D(0)[3]	O			C25_SHI[1]	
K10	RX_D(0)[0]	I	RX_D(0)[0]	I		UI		
M12	RX_D(0)[1]/ EBUS(0)-RX+	I/LI+	RX_D(0)[1]	I	EBUS(0)-RX+	LI+		27 kΩ PD
L11	RX_D(0)[2]	I	RX_D(0)[2]	I		UI		
L12	RX_D(0)[3]	I	RX_D(0)[3]	I		UI		
M11	RX_DV(0)/ EBUS(0)-RX-	I/LI-	RX_DV(0]	I	EBUS(0)-RX-	LI-		27 kΩ PU
M10	RX_ERR(0)	I	RX_ERR(0)	I		UI		
L10	RX_CLK(0)	I	RX_CLK(0)	I		UI		
L9	LINK_MII(0)	I	LINK_MII(0)	I		UI		
J11	PERR(0)/ TRANS(0)/ CLK_MODE[0]	BD	PERR(0)/ TRANS(0)	O/ I	PERR(0)	O	CLK_MODE[0]	
J12	LINKACT(0)/ P_CONF[0]	BD	LINKACT(0)	O	LINKACT(0)	O	P_CONF[0]	

3.11.5 Physical Port 1

Table 40 shows the pins for physical port 1. It can be configured as MII or EBUS and is always available. Use of this port does in no case clash with pins needed for PDI.

Table 40: Physical Port 1

Pin	Pin		MII		EBUS		Configuration Signal	Internal PU/PD
	Name	Dir.	Signal	Dir.	Signal	Dir.		
M3	TX_ENA(1)/ EBUS(1)-TX+	BD/LO+	TX_ENA(1)	O/I	EBUS(1)-TX+	LO+		
L3	TX_D(1)[0]/ TRANS-MODE-ENA	BD	TX_D(1)[0]	O			TRANS_ MODE_ENA	
M2	TX_D(1)[1]/ EBUS(1)-TX-	O/LO-	TX_D(1)[1]	O	EBUS(1)-TX-	LO-		
L2	TX_D(1)[2]/ P_MODE[0]	BD	TX_D(1)[2]	O			P_MODE[0]	
M1	TX_D(1)[3]/ P_MODE[1]	BD	TX_D(1)[3]	O			P_MODE[1]	
L4	RX_D(1)[0]	I	RX_D(1)[0]	I		UI		
M5	RX_D(1)[1]/ EBUS(1)-RX+	I/LI+	RX_D(1)[1]	I	EBUS(1)-RX+	LI+		27 kΩ PD
L5	RX_D(1)[2]	I	RX_D(1)[2]	I		UI		
M6	RX_D(1)[3]	I	RX_D(1)[3]	I		UI		
M4	RX_DV(1)/ EBUS(1)-RX-	I/LI-	RX_DV(1)	I	EBUS(1)-RX-	LI-		27 kΩ PU
L6	RX_ERR(1)	I	RX_ERR(1)	I		UI		
K4	RX_CLK(1)	I	RX_CLK(1)	I		UI		
K3	LINK_MII(1)	I	LINK_MII(1)	I		UI		
K2	PERR(1)/ TRANS(1)/ CLK_MODE(1)	BD	PERR(1)/ TRANS(1)	O/ I	PERR(1)	O	CLK_MODE[1]	
L1	LINKACT(1)/ P_CONF(1)	BD	LINKACT(1)	O	LINKACT(1)	O	P_CONF[1]	

3.11.6 Physical Port 2 / PDI byte 4

Table 41 shows the pins for physical port 2 or for PDI byte 4 (PDI[39:32]). If used as communication port it can be configured as MII or EBUS.

Table 41: Physical Port 2/PDI byte 4

	Pin	Pin		PDI		MII		EBUS		Configu-ration Signal	Int. PU/PD
		Name	Dir.	Signal	Dir.	Signal	Dir.	Signal	Dir.		
PDI Byte 4	D1	PDI[32]/ TX_D(2)[3]	BD	PDI[32]	BD	TX_D(2)[3]	O	PDI[32]	BD		
	D2	PDI[33]/ TX_D(2)[2]	BD	PDI[33]	BD	TX_D(2)[2]	O	PDI[33]	BD		
	E2	PDI[34]/ TX_D(2)[0]/ CTRL_STATUS_MOVE	BD	PDI[34]	BD	TX_D(2)[0]	O	PDI[34]	BD	CTRL_STATUS_MOVE	
	G1	PDI[35]/ RX_ERR(2)	BD	PDI[35]	BD	RX_ERR(2)	I	PDI[35]	BD		
	G2	PDI[36]/ RX_CLK(2)	BD	PDI[36]	BD	RX_CLK(2)	I	PDI[36]	BD		
	H2	PDI[37]/ RX_D(2)[0]	BD	PDI[37]	BD	RX_D(2)[0]	I	PDI[37]	BD		
	J2	PDI[38]/ RX_D(2)[2]	BD	PDI[38]	BD	RX_D(2)[2]	I	PDI[38]	BD		
	K1	PDI[39]/ RX_D(2)[3]	BD	PDI[39]	BD	RX_D(2)[3]	I	PDI[39]	BD		

Table 42: Physical Port 2

Pin	Pin		Only 2 ports		MII		EBUS		Configu-ration Signal	Int. PU/PD
	Name	Dir.	Signal	Dir.	Signal	Dir.	Signal	Dir.		
F1	TX_ENA(2)/ EBUS(2)-TX+	BD/ LO+		UI	TX_ENA(2)	O/I	EBUS(2)-TX+	LO+		
E1	TX_D(2)[1]/ EBUS(2)-TX-	O/ LO-		n.c.	TX_D(2)[1]	O	EBUS(2)-TX-	LO-		
H1	RX_DV(2)/ EBUS(2)-RX-	I/LI-		UI	RX_DV(2)	I	EBUS(2)-RX-	LI-		27 kΩ PU
J1	RX_D(2)[1]/ EBUS(2)-RX+	I/LI+		UI	RX_D(2)[1]	I	EBUS(2)-RX+	LI+		27 kΩ PD
C3	PERR(2)/ TRANS(2)/ PHYAD_OFF	BD		O	PERR(2)/ TRANS(2)	O/ I	PERR(2)	O	PHYAD_OFF	
E3	LINKACT(2)/ P_CONF[2]	BD		O	LINKACT(2)	O	LINKACT(2)	O	P_CONF[2]	
F2	LINK_MII(2)/ CLK25OUT1	BD	CLK25OUT1	O	LINK_MII(2)	I	CLK25OUT1	O		

3.11.7 Physical Port 3 / PDI Bytes 2/3

Table 43 shows the pins for physical port 3 or for PDI bytes 2/3 (PDI[23:16], PDI[3117]). If used as communication port it can be configured as MII or EBUS.

Table 43: Physical Port 3 / PDI

	Pin	Pin		PDI		MII		EBUS		Configuration Signal	Int. PU/PD
		Name	Dir.	Signal	Dir.	Signal	Dir.	Signal	Dir.		
PDI Byte 2	A7	PDI[16]/ RX_ERR(3)	BD	PDI[16]	BD	RX_ERR(3)	I	PDI[16]	BD		
	B7	PDI[17]/ RX_CLK(3)	BD	PDI[17]	BD	RX_CLK(3)	I	PDI[17]	BD		
	A6	PDI[18]/ RX_D(3)[0]	BD	PDI[18]	BD	RX_D(3)[0]	I	PDI[18]	BD		
	B6	PDI[19]/ RX_D(3)[2]	BD	PDI[19]	BD	RX_D(3)[2]	I	PDI[19]	BD		
	A5	PDI[20]/ RX_D(3)[3]	BD	PDI[20]	BD	RX_D(3)[3]	I	PDI[20]	BD		
	B5	PDI[21]/ LINK_MII(3)	BD	PDI[21]	BD	LINK_MII(3)	I	PDI[21]	BD		
	A4	PDI[22]/ TX_D(3)[3]	BD	PDI[22]	BD	TX_D(3)[3]	O	PDI[22]	BD		
	B4	PDI[23]/ TX_D(3)[2]	BD	PDI[23]	BD	TX_D(3)[2]	O	PDI[23]	BD		
PDI Byte 3	A3	PDI[24]/ TX_D(3)[1]/ EBUS(3)-TX-	BD/ LO-	PDI[24]	BD	TX_D(3)[1]	O	EBUS(3)-TX-	LO-		
	B3	PDI[25]/ TX_D(3)[0]	BD	PDI[25]	BD	TX_D(3)[0]	O		UI		
	A2	PDI[26]/ TX_ENA(3)/ EBUS(3)-TX+	BD/ LO+	PDI[26]	BD	TX_ENA(3)	O/I	EBUS(3)-TX+	LO+		
	A1	PDI[27]/ RX_DV(3)/ EBUS(3)-RX-	BD/ LI-	PDI[27]	BD	RX_DV(3)	I	EBUS(3)-RX-	LI-		27 kΩ PU
	B2	PDI[28]/ PERR(3)/ TRANS(3)	BD	PDI[28]	BD	PERR(3)/ TRANS(3)	O/ I	PERR(3)	O	RESER-VED	
	B1	PDI[29]/ RX_D(3)[1]/ EBUS(3)-RX+	BD/ LI+	PDI[29]	BD	RX_D(3)[1]	I	EBUS(3)-RX+	LI+		27 kΩ PD
	C2	PDI[30]/ LINKACT(3)/ P_CONF(3)	BD	PDI[30]	BD	LINKACT(3)	O	LINKACT(3)	O	P_CONF [3]	
	C1	PDI[31]/ CLK25OUT2	BD	PDI[31]/ CLK25 OUT2	BD	CLK25OUT2	O	CLK25OUT2	O		

3.11.8 PDI Bytes 0/1

Table 44 shows PDI byte 0 and byte 1 (PDI[15:0]).

The direction of all PDI pins depends on the PDI configuration stored in the SII EEPROM.

Table 44: PDI pins

		Pin	Pin		PDI, CLK_MODE=00		PDI, CLK_MODE/=00	
			Name	Dir.	Signal	Dir.	Signal	Dir.
PDI Byte O	PDI[7:0]	D12	PDI[0]	BD	PDI[0]	BD	PDI[0]	BD
		D11	PDI[1]	BD	PDI[1]	BD	PDI[1]	BD
		C12	PDI[2]	BD	PDI[2]	BD	PDI[2]	BD
		C11	PDI[3]	BD	PDI[3]	BD	PDI[3]	BD
		B12	PDI[4]	BD	PDI[4]	BD	PDI[4]	BD
		C10	PDI[5]	BD	PDI[5]	BD	PDI[5]	BD
		A12	PDI[6]	BD	PDI[6]	BD	PDI[6]	BD
		B11	PDI[7]/CPU_CLK	BD	PDI[7]	BD	CPU_CLK	O
PDI Byte 1	PDI[8:15]	A11	PDI[8]	BD	PDI[8]	BD	PDI[8]	BD
		B10	PDI[9]	BD	PDI[9]	BD	PDI[9]	BD
		A10	PDI[10]	BD	PDI[10]	BD	PDI[10]	BD
		C9	PDI[11]	BD	PDI[11]	BD	PDI[11]	BD
		A9	PDI[12]	BD	PDI[12]	BD	PDI[12]	BD
		B9	PDI[13]	BD	PDI[13]	BD	PDI[13]	BD
		A8	PDI[14]	BD	PDI[14]	BD	PDI[14]	BD
		B8	PDI[15]	BD	PDI[15]	BD	PDI[15]	BD

3.12 PDI Signal Pinout depending on selected PDI

The PDI signal pinout depends on the selected PDI (SII EEPROM). The PDI selection and PDI signal pinout is subject to restrictions introduced by the port configuration. Digital I/O and SPI PDI are available in any configuration – although the I/O width can be reduced depending on the configuration. The µController PDIs are only available with up to 3 ports, the data bus width can be reduced depending on the configuration.

Refer to PDI descriptions for further PDI and PDI signal descriptions.

The SPI PDI supports additional general purpose I/O signals, which are not part of the SPI PDI description:

GPO[x]
General purpose output signals.

GPI[x]
General purpose input signals.

3.12.1 Digital I/O Pin Out

Table 45: Mapping of Digital I/O Interface (1)

	Digital IO	PDI signal	2 ports, or 3 ports with min. 1xEBUS		3xMII, 0xEBUS CTRL_STATUS_MOVE=			
					0		1	
			Signal	Dir.	Signal	Dir.	Signal	Dir.
PDI Byte 0	PDI[15:0]	PDI[0]	I/O[0]	I/O/BD	I/O[0]	I/O/BD	I/O[0]	I/O/BD
		PDI[1]	I/O[1]	I/O/BD	I/O[1]	I/O/BD	I/O[1]	I/O/BD
		PDI[2]	I/O[2]	I/O/BD	I/O[2]	I/O/BD	I/O[2]	I/O/BD
		PDI[3]	I/O[3]	I/O/BD	I/O[3]	I/O/BD	I/O[3]	I/O/BD
		PDI[4]	I/O[4]	I/O/BD	I/O[4]	I/O/BD	I/O[4]	I/O/BD
		PDI[5]	I/O[5]	I/O/BD	I/O[5]	I/O/BD	I/O[5]	I/O/BD
		PDI[6]	I/O[6]	I/O/BD	I/O[6]	I/O/BD	I/O[6]	I/O/BD
		PDI[7]/ CPU_CLK	I/O[7]/ CPU_CLK	I/O/BD/ O	I/O[7]/ CPU_CLK	I/O/BD/ O	I/O[7]/ CPU_CLK	I/O/BD/ O
PDI Byte 1		PDI[8]	I/O[8]	I/O/BD	I/O[8]	I/O/BD	I/O[8]	I/O/BD
		PDI[9]	I/O[9]	I/O/BD	I/O[9]	I/O/BD	I/O[9]	I/O/BD
		PDI[10]	I/O[10]	I/O/BD	I/O[10]	I/O/BD	I/O[10]	I/O/BD
		PDI[11]	I/O[11]	I/O/BD	I/O[11]	I/O/BD	I/O[11]	I/O/BD
		PDI[12]	I/O[12]	I/O/BD	I/O[12]	I/O/BD	I/O[12]	I/O/BD
		PDI[13]	I/O[13]	I/O/BD	I/O[13]	I/O/BD	I/O[13]	I/O/BD
		PDI[14]	I/O[14]	I/O/BD	I/O[14]	I/O/BD	I/O[14]	I/O/BD
		PDI[15]	I/O[15]	I/O/BD	I/O[15]	I/O/BD	I/O[15]	I/O/BD
PDI Byte 2	PDI[23:16]/ MII(3)	PDI[16]	I/O[16]	I/O/BD	I/O[16]	I/O/BD	I/O[16]	I/O/BD
		PDI[17]	I/O[17]	I/O/BD	I/O[17]	I/O/BD	I/O[17]	I/O/BD
		PDI[18]	I/O[18]	I/O/BD	I/O[18]	I/O/BD	I/O[18]	I/O/BD
		PDI[19]	I/O[19]	I/O/BD	I/O[19]	I/O/BD	I/O[19]	I/O/BD
		PDI[20]	I/O[20]	I/O/BD	I/O[20]	I/O/BD	I/O[20]	I/O/BD
		PDI[21]	I/O[21]	I/O/BD	I/O[21]	I/O/BD	I/O[21]	I/O/BD
		PDI[22]	I/O[22]	I/O/BD	I/O[22]	I/O/BD	I/O[22]	I/O/BD
		PDI[23]	I/O[23]	I/O/BD	I/O[23]	I/O/BD	I/O[23]	I/O/BD
PDI Byte 3	PDI[31:24]/ MII(3)/ EBUS(3)	PDI[24]	I/O[24]	I/O/BD	I/O[24]	I/O/BD	SOF	O
		PDI[25]	I/O[25]	I/O/BD	I/O[25]	I/O/BD	OE_EXT	I
		PDI[26]	I/O[26]	I/O/BD	I/O[26]	I/O/BD	OUTVALID	O
		PDI[27]	I/O[27]	I/O/BD	I/O[27]	I/O/BD	WD_TRIG	O
		PDI[28]	I/O[28]	I/O/BD	I/O[28]	I/O/BD	LATCH_IN	I
		PDI[29]	I/O[29]	I/O/BD	I/O[29]	I/O/BD	OE_CONF	I
		PDI[30]	I/O[30]	I/O/BD	I/O[30]	I/O/BD	EEPROM_ LOADED	O
		PDI[31]/ CLK25OUT2	I/O[31]/ CLK25OUT2	I/O/BD/ O	I/O[31]/ CLK25OUT2	I/O/BD/ O	--/ CLK25OUT2	--/ O
PDI Byte 4	PDI[39:32]/ MII(2)	PDI[32]	SOF	O	MII(2)		MII(2)	
		PDI[33]	OE_EXT	I				
		PDI[34]	OUTVALID	O				
		PDI[35]	WD_TRIG	O				
		PDI[36]	LATCH_IN	I				
		PDI[37]	OE_CONF	I				
		PDI[38]	EEPROM_LOADED	O				
		PDI[39]	--	--				

Table 46: Mapping of Digital I/O Interface (2)

	Digital IO	PDI signal	4 ports, min. 2x EBUS		3xMII, 1xEBUS CTRL_STATUS_MOVE			
					0		1	
			Signal	Dir.	Signal	Dir.	Signal	Dir.
PDI Byte 0	PDI[15:0]	PDI[0]	I/O[0]	I/O/BD	I/O[0]	I/O/BD	I/O[0]	I/O/BD
		PDI[1]	I/O[1]	I/O/BD	I/O[1]	I/O/BD	I/O[1]	I/O/BD
		PDI[2]	I/O[2]	I/O/BD	I/O[2]	I/O/BD	I/O[2]	I/O/BD
		PDI[3]	I/O[3]	I/O/BD	I/O[3]	I/O/BD	I/O[3]	I/O/BD
		PDI[4]	I/O[4]	I/O/BD	I/O[4]	I/O/BD	I/O[4]	I/O/BD
		PDI[5]	I/O[5]	I/O/BD	I/O[5]	I/O/BD	I/O[5]	I/O/BD
		PDI[6]	I/O[6]	I/O/BD	I/O[6]	I/O/BD	I/O[6]	I/O/BD
		PDI[7]/ CPU_CLK	I/O[7]/ CPU_CLK	I/O/BD/ O	I/O[7]/ CPU_CLK	I/O/BD/ O	I/O[7]/ CPU_CLK	I/O/BD/ O
PDI Byte 1		PDI[8]	I/O[8]	I/O/BD	I/O[8]	I/O/BD	I/O[8]	I/O/BD
		PDI[9]	I/O[9]	I/O/BD	I/O[9]	I/O/BD	I/O[9]	I/O/BD
		PDI[10]	I/O[10]	I/O/BD	I/O[10]	I/O/BD	I/O[10]	I/O/BD
		PDI[11]	I/O[11]	I/O/BD	I/O[11]	I/O/BD	I/O[11]	I/O/BD
		PDI[12]	I/O[12]	I/O/BD	I/O[12]	I/O/BD	I/O[12]	I/O/BD
		PDI[13]	I/O[13]	I/O/BD	I/O[13]	I/O/BD	I/O[13]	I/O/BD
		PDI[14]	I/O[14]	I/O/BD	I/O[14]	I/O/BD	I/O[14]	I/O/BD
		PDI[15]	I/O[15]	I/O/BD	I/O[15]	I/O/BD	I/O[15]	I/O/BD
PDI Byte 2	PDI[23:16]/ MII(3)	PDI[16]	I/O[16]	I/O/BD	I/O[16]	I/O/BD	SOF	O
		PDI[17]	I/O[17]	I/O/BD	I/O[17]	I/O/BD	OE_EXT	I
		PDI[18]	I/O[18]	I/O/BD	I/O[18]	I/O/BD	OUTVALID	O
		PDI[19]	I/O[19]	I/O/BD	I/O[19]	I/O/BD	WD_TRIG	O
		PDI[20]	I/O[20]	I/O/BD	I/O[20]	I/O/BD	LATCH_IN	I
		PDI[21]	I/O[21]	I/O/BD	I/O[21]	I/O/BD	OE_CONF	I
		PDI[22]	I/O[22]	I/O/BD	I/O[22]	I/O/BD	EEPROM_ LOADED	O
		PDI[23]	I/O[23]	I/O/BD	I/O[23]	I/O/BD	--	--
PDI Byte 3	PDI[31:24]/ MII(3)/ EBUS(3)	PDI[24]	EBUS(3)		EBUS(3)		EBUS(3)	
		PDI[25]						
		PDI[26]						
		PDI[27]						
		PDI[28]						
		PDI[29]						
		PDI[30]						
		PDI[31]/ CLK25OUT2						
PDI Byte 4	PDI[39:32]/ MII(2)	PDI[32]	SOF	O	MII(2)		MII(2)	
		PDI[33]	OE_EXT	I				
		PDI[34]	OUTVALID	O				
		PDI[35]	WD_TRIG	O				
		PDI[36]	LATCH_IN	I				
		PDI[37]	OE_CONF	I				
		PDI[38]	EEPROM_LOADED	O				
		PDI[39]	--	--				

Table 47: Mapping of Digital I/O Interface (3)

	Digital IO	PDI signal	4xMII CTRL_STATUS_MOVE=			
			0		1	
			Signal	Dir.	Signal	Dir.
PDI Byte 0	PDI[15:0]	PDI[0]	I/O[0]	I/O/BD	I/O[0]	I/O/BD
		PDI[1]	I/O[1]	I/O/BD	I/O[1]	I/O/BD
		PDI[2]	I/O[2]	I/O/BD	I/O[2]	I/O/BD
		PDI[3]	I/O[3]	I/O/BD	I/O[3]	I/O/BD
		PDI[4]	I/O[4]	I/O/BD	I/O[4]	I/O/BD
		PDI[5]	I/O[5]	I/O/BD	I/O[5]	I/O/BD
		PDI[6]	I/O[6]	I/O/BD	I/O[6]	I/O/BD
		PDI[7]/CPU_CLK	I/O[7]/CPU_CLK	I/O/BD/O	I/O[7]/CPU_CLK	I/O/BD/O
PDI Byte 1		PDI[8]	I/O[8]	I/O/BD	SOF	O
		PDI[9]	I/O[9]	I/O/BD	OE_EXT	I
		PDI[10]	I/O[10]	I/O/BD	OUTVALID	O
		PDI[11]	I/O[11]	I/O/BD	WD_TRIG	O
		PDI[12]	I/O[12]	I/O/BD	LATCH_IN	I
		PDI[13]	I/O[13]	I/O/BD	OE_CONF	I
		PDI[14]	I/O[14]	I/O/BD	EEPROM_LOADED	O
		PDI[15]	I/O[15]	I/O/BD	--	--
PDI Byte 2	PDI[23:16]/ MII(3)	PDI[16]	MII(3)		MII(3)	
		PDI[17]				
		PDI[18]				
		PDI[19]				
		PDI[20]				
		PDI[21]				
		PDI[22]				
		PDI[23]				
PDI Byte 3	PDI[31:24]/ MII(3)/ EBUS(3)	PDI[24]	MII(3)		MII(3)	
		PDI[25]				
		PDI[26]				
		PDI[27]				
		PDI[28]				
		PDI[29]				
		PDI[30]				
		PDI[31]/CLK25OUT2				
PDI Byte 4	PDI[39:32]/ MII(2)	PDI[32]	MII(2)		MII(2)	
		PDI[33]				
		PDI[34]				
		PDI[35]				
		PDI[36]				
		PDI[37]				
		PDI[38]				
		PDI[39]				

3.12.2 8/16 Bit asynchronous µController

Figure 13: Mapping of asynchronous µC Interface to Port

	Async. µC	PDI signal	2 ports, or 3 ports with min. 1xEBUS				3xMII, 0xEBUS	
			8 bit		16 bit		8 bit	
			Signal	Dir.	Signal	Dir.	Signal	Dir.
PDI Byte 0	PDI[15:0]	PDI[0]	CS	I	CS	I	CS	I
		PDI[1]	RD	I	RD	I	RD	I
		PDI[2]	WR	I	WR	I	WR	I
		PDI[3]	BUSY	O	BUSY	O	BUSY	O
		PDI[4]	IRQ	O	IRQ	O	IRQ	O
		PDI[5]	BHE	I	BHE	I	BHE	I
		PDI[6]	EEPROM_LOADED	O	EEPROM_LOADED	O	EEPROM_LOADED	O
		PDI[7]/ CPU_CLK	ADR[15]/ CPU_CLK	I/ O	ADR[15]/ CPU_CLK	I/ O	ADR[15]/ CPU_CLK	I/ O
PDI Byte 1		PDI[8]	ADR[14]	I	ADR[14]	I	ADR[14]	I
		PDI[9]	ADR[13]	I	ADR[13]	I	ADR[13]	I
		PDI[10]	ADR[12]	I	ADR[12]	I	ADR[12]	I
		PDI[11]	ADR[11]	I	ADR[11]	I	ADR[11]	I
		PDI[12]	ADR[10]	I	ADR[10]	I	ADR[10]	I
		PDI[13]	ADR[9]	I	ADR[9]	I	ADR[9]	I
		PDI[14]	ADR[8]	I	ADR[8]	I	ADR[8]	I
		PDI[15]	ADR[7]	I	ADR[7]	I	ADR[7]	I
PDI Byte 2	PDI[23:16]/ MII(3)	PDI[16]	ADR[6]	I	ADR[6]	I	ADR[6]	I
		PDI[17]	ADR[5]	I	ADR[5]	I	ADR[5]	I
		PDI[18]	ADR[4]	I	ADR[4]	I	ADR[4]	I
		PDI[19]	ADR[3]	I	ADR[3]	I	ADR[3]	I
		PDI[20]	ADR[2]	I	ADR[2]	I	ADR[2]	I
		PDI[21]	ADR[1]	I	ADR[1]	I	ADR[1]	I
		PDI[22]	ADR[0]	I	ADR[0]	I	ADR[0]	I
		PDI[23]	DATA[0]	BD	DATA[0]	BD	DATA[0]	BD
PDI Byte 3	PDI[31:24]/ MII(3)/ EBUS(3)	PDI[24]	DATA[1]	BD	DATA[1]	BD	DATA[1]	BD
		PDI[25]	DATA[2]	BD	DATA[2]	BD	DATA[2]	BD
		PDI[26]	DATA[3]	BD	DATA[3]	BD	DATA[3]	BD
		PDI[27]	DATA[4]	BD	DATA[4]	BD	DATA[4]	BD
		PDI[28]	DATA[5]	BD	DATA[5]	BD	DATA[5]	BD
		PDI[29]	DATA[6]	BD	DATA[6]	BD	DATA[6]	BD
		PDI[30]	DATA[7]	BD	DATA[7]	BD	DATA[7]	BD
		PDI[31]/ CLK25OUT2	--/ CLK25OUT2	--/ O	--/ CLK25OUT2	--/ O	--/ CLK25OUT2	--/ O
PDI Byte 4	PDI[39:32]/ MII(2)	PDI[32]	--	--	DATA[8]	BD	MII(2)	
		PDI[33]	--	--	DATA[9]	BD		
		PDI[34]	--	--	DATA[10]	BD		
		PDI[35]	--	--	DATA[11]	BD		
		PDI[36]	--	--	DATA[12]	BD		
		PDI[37]	--	--	DATA[13]	BD		
		PDI[38]	--	--	DATA[14]	BD		
		PDI[39]	--	--	DATA[15]	BD		

3.12.3 8/16 Bit synchronous µController

Table 48: Mapping of synchronous µC Interface to Port

	Sync. µC	PDI signal	2 ports, or 3 ports with min. 1xEBUS				3xMII, 0xEBUS	
			8 bit		16 bit		8 bit	
			Signal	Dir.	Signal	Dir.	Signal	Dir.
PDI Byte 0	PDI[15:0]	PDI[0]	CS	I	CS	I	CS	I
		PDI[1]	TS	I	TS	I	TS	I
		PDI[2]	RD/nWR	I	RD/nWR	I	RD/nWR	I
		PDI[3]	TA	O	TA	O	TA	O
		PDI[4]	IRQ	O	IRQ	O	IRQ	O
		PDI[5]	BHE	I	BHE	I	BHE	I
		PDI[6]	EEPROM_LOADED	O	EEPROM_LOADED	O	EEPROM_LOADED	O
		PDI[7]/ CPU_CLK	ADR[15]/ CPU_CLK	I/ O	ADR[15]/ CPU_CLK	I/O	ADR[15]/ CPU_CLK	I/ O
PDI Byte 1		PDI[8]	ADR[14]	I	ADR[14]	I	ADR[14]	I
		PDI[9]	ADR[13]	I	ADR[13]	I	ADR[13]	I
		PDI[10]	ADR[12]	I	ADR[12]	I	ADR[12]	I
		PDI[11]	ADR[11]	I	ADR[11]	I	ADR[11]	I
		PDI[12]	ADR[10]	I	ADR[10]	I	ADR[10]	I
		PDI[13]	ADR[9]	I	ADR[9]	I	ADR[9]	I
		PDI[14]	ADR[8]	I	ADR[8]	I	ADR[8]	I
		PDI[15]	ADR[7]	I	ADR[7]	I	ADR[7]	I
PDI Byte 2	PDI[23:16]/ MII(3)	PDI[16]	ADR[6]	I	ADR[6]	I	ADR[6]	I
		PDI[17]	ADR[5]	I	ADR[5]	I	ADR[5]	I
		PDI[18]	ADR[4]	I	ADR[4]	I	ADR[4]	I
		PDI[19]	ADR[3]	I	ADR[3]	I	ADR[3]	I
		PDI[20]	ADR[2]	I	ADR[2]	I	ADR[2]	I
		PDI[21]	ADR[1]	I	ADR[1]	I	ADR[1]	I
		PDI[22]	ADR[0]	I	ADR[0]	I	ADR[0]	I
		PDI[23]	DATA[0]	BD	DATA[0]	BD	DATA[0]	BD
PDI Byte 3	PDI[31:24]/ MII(3)/ EBUS(3)	PDI[24]	DATA[1]	BD	DATA[1]	BD	DATA[1]	BD
		PDI[25]	DATA[2]	BD	DATA[2]	BD	DATA[2]	BD
		PDI[26]	DATA[3]	BD	DATA[3]	BD	DATA[3]	BD
		PDI[27]	DATA[4]	BD	DATA[4]	BD	DATA[4]	BD
		PDI[28]	DATA[5]	BD	DATA[5]	BD	DATA[5]	BD
		PDI[29]	DATA[6]	BD	DATA[6]	BD	DATA[6]	BD
		PDI[30]	DATA[7]	BD	DATA[7]	BD	DATA[7]	BD
		PDI[31]	CPU_CLK_IN	I	CPU_CLK_IN	I	CPU_CLK_IN	I
PDI Byte 4	PDI[39:32]/ MII(2)	PDI[32]	--	--	DATA[8]	BD	MII(2)	
		PDI[33]	--	--	DATA[9]	BD		
		PDI[34]	--	--	DATA[10]	BD		
		PDI[35]	--	--	DATA[11]	BD		
		PDI[36]	--	--	DATA[12]	BD		
		PDI[37]	--	--	DATA[13]	BD		
		PDI[38]	--	--	DATA[14]	BD		
		PDI[39]	--	--	DATA[15]	BD		

3.12.4 SPI Pin Out

Figure 14: Mapping of SPI Interface to Port (1)

	SPI	PDI signal	2 ports, or 3 ports with min. 1xEBUS		3xMII, 0xEBUS	
			Signal	Dir.	Signal	Dir.
PDI Byte 0	PDI[15:0]	PDI[0]	SPI_CLK	I	SPI_CLK	I
		PDI[1]	SPI_SEL	I	SPI_SEL	I
		PDI[2]	SPI_DI	I	SPI_DI	I
		PDI[3]	SPI_DO	O	SPI_DO	O
		PDI[4]	SPI_IRQ	O	SPI_IRQ	O
		PDI[5]	--	--	--	--
		PDI[6]	EEPROM_LOADED	O	EEPROM_LOADED	O
		PDI[7]/CPU_CLK	--/CPU_CLK	--/O	--/CPU_CLK	--/O
PDI Byte 1		PDI[8]	GPO[0]	O	GPO[0]	O
		PDI[9]	GPO[1]	O	GPO[1]	O
		PDI[10]	GPO[2]	O	GPO[2]	O
		PDI[11]	GPO[3]	O	GPO[3]	O
		PDI[12]	GPI[0]	I	GPI[0]	I
		PDI[13]	GPI[1]	I	GPI[1]	I
		PDI[14]	GPI[2]	I	GPI[2]	I
		PDI[15]	GPI[3]	I	GPI[3]	I
PDI Byte 2	PDI[23:16]/ MII(3)	PDI[16]	GPO[4]	O	GPO[4]	O
		PDI[17]	GPO[5]	O	GPO[5]	O
		PDI[18]	GPO[6]	O	GPO[6]	O
		PDI[19]	GPO[7]	O	GPO[7]	O
		PDI[20]	GPI[4]	I	GPI[4]	I
		PDI[21]	GPI[5]	I	GPI[5]	I
		PDI[22]	GPI[6]	I	GPI[6]	I
		PDI[23]	GPI[7]	I	GPI[7]	I
PDI Byte 3	PDI[31:24]/ MII(3)/ EBUS(3)	PDI[24]	GPO[8]	O	GPO[8]	O
		PDI[25]	GPO[9]	O	GPO[9]	O
		PDI[26]	GPO[10]	O	GPO[10]	O
		PDI[27]	GPO[11]	O	GPO[11]	O
		PDI[28]	GPI[8]	I	GPI[8]	I
		PDI[29]	GPI[9]	I	GPI[9]	I
		PDI[30]	GPI[10]	I	GPI[10]	I
		PDI[31]/CLK25OUT2	GPI[11]/CLK25OUT2	I/O	GPI[11]/CLK25OUT2	I/O
PDI Byte 4	PDI[39:32]/ MII(2)	PDI[32]	GPO[12]	O	MII(2)	
		PDI[33]	GPO[13]	O		
		PDI[34]	GPO[14]	O		
		PDI[35]	GPO[15]	O		
		PDI[36]	GPI[12]	I		
		PDI[37]	GPI[13]	I		
		PDI[38]	GPI[14]	I		
		PDI[39]	GPI[15]	I		

Table 49: Mapping of SPI Interface to Port (2)

	SPI	PDI signal	4 ports, min. 2x EBUS		3xMII, 1xEBUS		4xMII	
			Signal	Dir.	Signal	Dir.	Signal	Dir.
PDI Byte 0	PDI[15:0]	PDI[0]	SPI_CLK	I	SPI_CLK	I	SPI_CLK	I
		PDI[1]	SPI_SEL	I	SPI_SEL	I	SPI_SEL	I
		PDI[2]	SPI_DI	I	SPI_DI	I	SPI_DI	I
		PDI[3]	SPI_DO	O	SPI_DO	O	SPI_DO	O
		PDI[4]	SPI_IRQ	O	SPI_IRQ	O	SPI_IRQ	O
		PDI[5]	--	--	--	--	--	--
		PDI[6]	EEPROM _LOADED	O	EEPROM _LOADED	O	EEPROM _LOADED	O
		PDI[7]/CPU_CLK	--/CPU_CLK	--/O	--/CPU_CLK	--/O	--/CPU_CLK	--/O
PDI Byte 1		PDI[8]	GPO[0]	O	GPO[0]	O	GPO[0]	O
		PDI[9]	GPO[1]	O	GPO[1]	O	GPO[1]	O
		PDI[10]	GPO[2]	O	GPO[2]	O	GPO[2]	O
		PDI[11]	GPO[3]	O	GPO[3]	O	GPO[3]	O
		PDI[12]	GPI[0]	I	GPI[0]	I	GPI[0]	I
		PDI[13]	GPI[1]	I	GPI[1]	I	GPI[1]	I
		PDI[14]	GPI[2]	I	GPI[2]	I	GPI[2]	I
		PDI[15]	GPI[3]	I	GPI[3]	I	GPI[3]	I
PDI Byte 2	PDI[23:16]/ MII(3)	PDI[16]	GPO[4]	O	GPO[4]	O	MII(3)	
		PDI[17]	GPO[5]	O	GPO[5]	O		
		PDI[18]	GPO[6]	O	GPO[6]	O		
		PDI[19]	GPO[7]	O	GPO[7]	O		
		PDI[20]	GPI[4]	I	GPI[4]	I		
		PDI[21]	GPI[5]	I	GPI[5]	I		
		PDI[22]	GPI[6]	I	GPI[6]	I		
		PDI[23]	GPI[7]	I	GPI[7]	I		
PDI Byte 3	PDI[31:24]/ MII(3)/ EBUS(3)	PDI[24]	EBUS(3)		EBUS(3)		MII(3)	
		PDI[25]						
		PDI[26]						
		PDI[27]						
		PDI[28]						
		PDI[29]						
		PDI[30]						
		PDI[31]/CLK25OUT2						
PDI Byte 4	PDI[39:32]/ MII(2)	PDI[32]	GPO[12]	O	MII(2)		MII(2)	
		PDI[33]	GPO[13]	O				
		PDI[34]	GPO[14]	O				
		PDI[35]	GPO[15]	O				
		PDI[36]	GPI[12]	I				
		PDI[37]	GPI[13]	I				
		PDI[38]	GPI[14]	I				
		PDI[39]	GPI[15]	I				

3.13 TESTMODE Pin

Table 50: TESTMODE pin

Pin	Pin		Signal		Configuration Signal	Internal PU/PD
	Name	Dir.	Signal	Dir.		
H3	TESTMODE	I	TESTMODE	I		WPD

TESTMODE
Reserved for testing, should be connected to GND.

3.14 Reserved Pins

Table 51 shows reserved Pins which are not used on the ET1100 and have to be connected to $GND_{I/O}$.

Table 51: Reserved Pins

Pin	Pin name	Dir.	Connect to
E4	Res. [0]	I	$GND_{I/O}$
G3	Res. [1]	I	$GND_{I/O}$
G4	Res. [2]	I	$GND_{I/O}$
E10	Res. [3]	I	$GND_{I/O}$
C8	Res. [4]	I	$GND_{I/O}$
H10	Res. [5]	I	$GND_{I/O}$
F4	Res. [6]	I	$GND_{I/O}$
D8	Res. [7]	I	$GND_{I/O}$

4 MII Interface

The ET1100 is connected with Ethernet PHYs using the MII interfaces. The MII interfaces of the ET1100 are optimized for low processing/forwarding delays by omitting a transmit FIFO. To allow this, the ET1100 has additional requirements to Ethernet PHYs, which are easily accomplished by several PHY vendors.

Refer to "Section I – Technology" for Ethernet PHY requirements.

Additional information regarding the ET1100:

- The clock source of the PHYs is either CLK25OUT1/2 of the ET1100, or the clock signal that is connected to OSC_IN if a quartz oscillator is used.
- The TX_CLK signal of the PHYs is not connected to the ET1100. The ET1100 does not use the MII interface for link detection or link configuration.

For details about the ESC MII Interface refer to Section I.

4.1 MII Interface Signals

The MII interface of the ET1100 has the following signals:

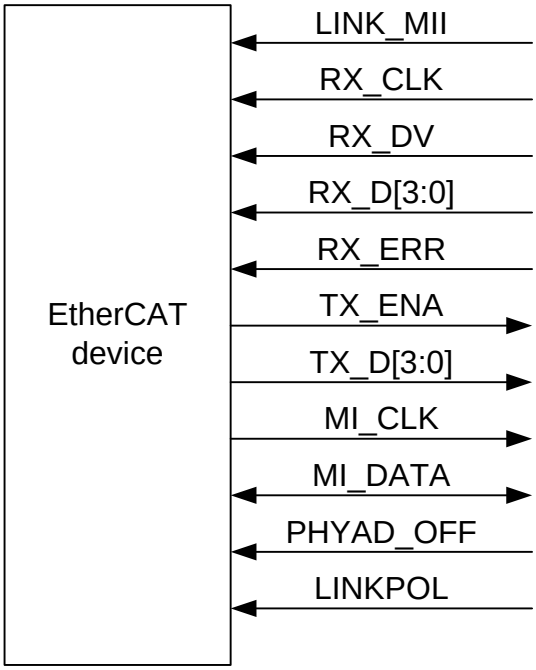

Figure 15: MII Interface signals

Table 52: MII Interface signals

Signal	Direction	Description
LINK_MII	IN	Input signal provided by the PHY if a 100 Mbit/s (Full Duplex) link is established
RX_CLK	IN	Receive Clock
RX_DV	IN	Receive data valid
RX_D[3:0]	IN	Receive data (alias RXD)
RX_ERR	IN	Receive error (alias RX_ER)
TX_ENA	OUT	Transmit enable (alias TX_EN)
TX_D[3:0]	OUT	Transmit data (alias TXD)
MI_CLK	OUT	Management Interface clock (alias MCLK)
MI_DATA	BIDIR	Management Interface data (alias MDIO)
PHYAD_OFF	IN	Configuration: PHY address offset
LINKPOL	IN	Configuration: LINK_MII polarity

MI_DATA should have an external pull-up resistor (4.7 kΩ recommended for ESCs). MI_CLK is driven rail-to-rail, idle value is High.

4.2 PHY Address Configuration

The ET1100 addresses Ethernet PHYs using logical port number (or PHY address register value) plus PHY address offset. Typically, the Ethernet PHY addresses should correspond with the logical port number, so PHY addresses 0-3 are used.

A PHY address offset of 16 can be applied which moves the PHY addresses to 16-19 by inverting the MSB of the PHY address internally.

If both alternatives cannot be used, the PHYs should be configured to use an actual PHY address offset of 1, i.e., PHY addresses 1-4. The PHY address offset configuration of the ET1100 remains 0.

Refer to Section I for more details about PHY addressing.

4.3 TX Shift Compensation

Since ET1100 and the Ethernet PHY share the same clock source, TX_CLK from the PHY has a fixed phase relation to TX_ENA/TX_D[3:0] from the ET1100. Thus, TX_CLK is not connected and the delay of a TX FIFO inside the ET1100 is saved. The phase shift between TX_CLK and TX_ENA/TX_D[3:0] can be compensated by an appropriate value for TX Shift, which will delay TX_ENA/TX_D[3:0] by 0, 10, 20, or 30 ns.

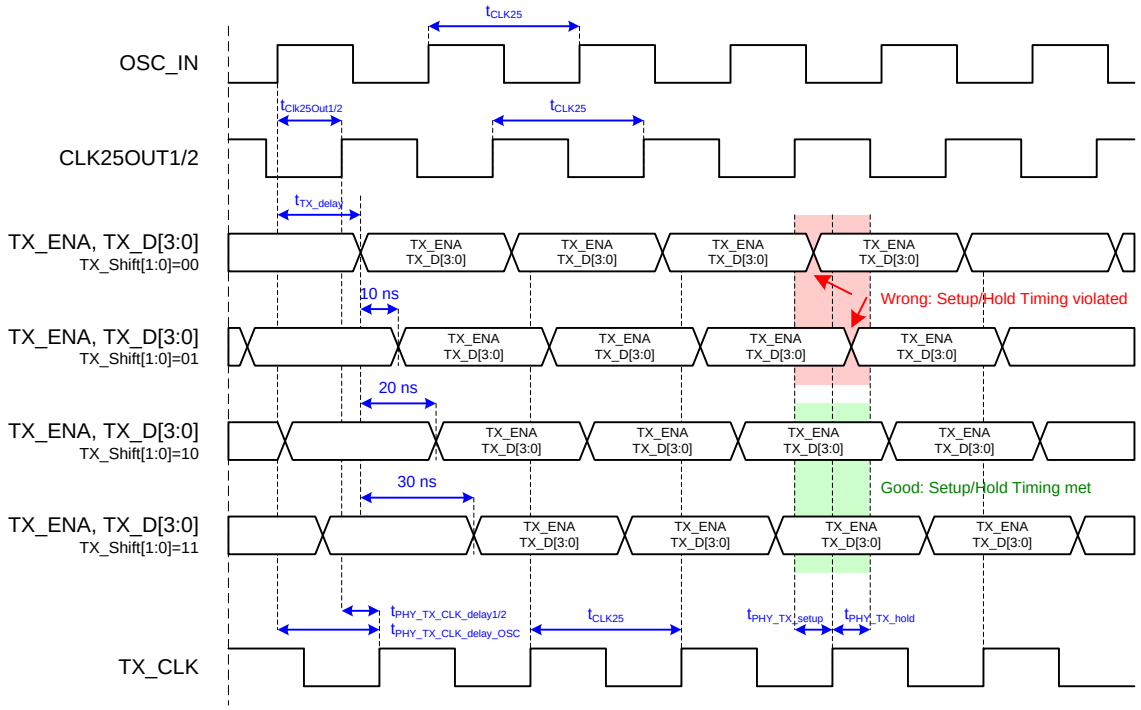

Figure 16: TX Shift Timing Diagram

Table 53: TX Shift Timing characteristics

Parameter	Comment
t_{CLK25}	25 MHz clock source period (OSC_IN, see f_{CLK25})
$t_{CLK25OUT1/2}$	CLK25OUT1/2 delay after OSC_IN (refer to AC characteristics)
t_{TX_delay}	TX_ENA/TX_D[3:0] delay after rising edge of OSC_IN (refer to AC characteristics)
$t_{PHY\_TX\_CLK\_delay1/2}$	Delay between PHY clock source CLK25OUT1/2 and TX_CLK output of the PHY, PHY dependent.
$t_{PHY\_TX\_CLK\_delay\_OSC}$	Delay between PHY clock source OSC_IN and TX_CLK output of the PHY, PHY dependent.
$t_{PHY\_TX\_setup}$	PHY setup requirement: TX_ENA/TX_D[3:0] with respect to TX_CLK. (PHY dependent, IEEE802.3 limit is 15 ns)
$t_{PHY\_TX\_hold}$	PHY hold requirement: TX_ENA/TX_D[3:0] with respect to TX_CLK. (PHY dependent, IEEE802.3 limit is 0 ns)

NOTE: TX Shift can be adjusted by displaying TX_CLK of a PHY and TX_ENA/TX_D[3:0] on an oscilloscope. TX_ENA/TX_D is allowed to change between 0 ns and 25 ns after a rising edge of TX_CLK (according to IEEE802.3 – check your PHY's documentation, it may contain relaxed timing requirements). Configure TX Shift so that TX_ENA/TX_D[3:0] change near the middle of this range. It is sufficient to check just one of the TX_ENA/TX_D[3:0] signals, because they are generated nearly at the same time.

4.4 Timing specifications

Table 54: MII timing characteristics

Parameter	Min	Typ	Max	Comment
t_{RX_CLK}	40 ns ± 100 ppm			RX_CLK period (100 ppm with maximum FIFO Size only)
t_{RX_setup}	9 ns			RX_DV/RX_DATA/RX_D[3:0] valid before rising edge of RX_CLK
t_{RX_hold}	3 ns			RX_DV/RX_DATA/RX_D[3:0] valid after rising edge of RX_CLK
t_{Clk}		~ 1.44 µs		MI_CLK period (f_{Clk} ≈ 700 kHz)
t_{Write}		~ 92.16 µs		MI Write access time
t_{Read}		~ 91.44 us		MI Read access time

NOTE: For MI timing diagrams refer to Section I.

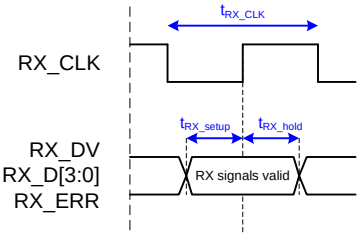

Figure 17: MII timing RX signals

5 EBUS/LVDS Interface

For details about the ESC EBUS Interface refer to Section I.

5.1 EBUS Interface Signals

The EBUS interface of the ET1100 has the following signals:

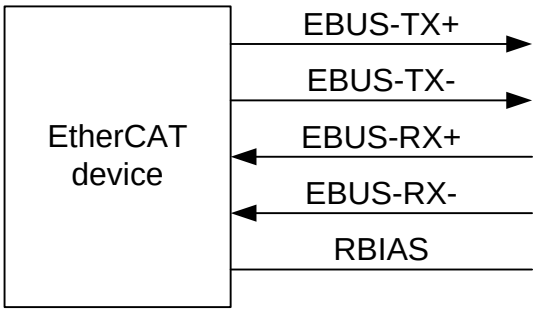

Figure 18: EBUS Interface Signals

Table 55: EBUS Interface signals

Signal	Direction	Description
EBUS-TX+ EBUS-TX-	OUT	EBUS/LVDS transmit signals
EBUS-RX+ EBUS-RX-	IN	EBUS/LVDS receive signals
RBIAS		BIAS resistor for EBUS-TX current adjustment

NOTE: An external LVDS termination with an impedance of 100 Ω between EBUS-RX+ and EBUS-RX- is necessary for EBUS ports. EBUS-RX+ incorporates a pull-down resistor and EBUS-RX- incorporated a pull-up resistor.

6 PDI Description

Table 56: Available PDIs for ET1100

PDI number (PDI Control register 0x0140[7:0])	PDI name	ET1100
0	Interface deactivated	x
4	Digital I/O	x
5	SPI Slave	x
7	EtherCAT Bridge (port 3)	
8	16 Bit async. µC	x
9	8 Bit async. µC	x
10	16 Bit sync. µC	x
11	8 Bit sync. µC	x
16	32 Digital Input/0 Digital Output	
17	24 Digital Input/8 Digital Output	
18	16 Digital Input/16 Digital Output	
19	8 Digital Input/24 Digital Output	
20	0 Digital Input/32 Digital Output	
128	On-chip bus (Avalon or OPB)	
Others	Reserved	

6.1 PDI Deactivated

The PDI is deactivated with PDI type 0x00. The PDI pins are not driven (high impedance).

6.2 Digital I/O Interface

6.2.1 Interface

The Digital I/O PDI is selected with PDI type 0x04. The signals of the Digital I/O interface are:

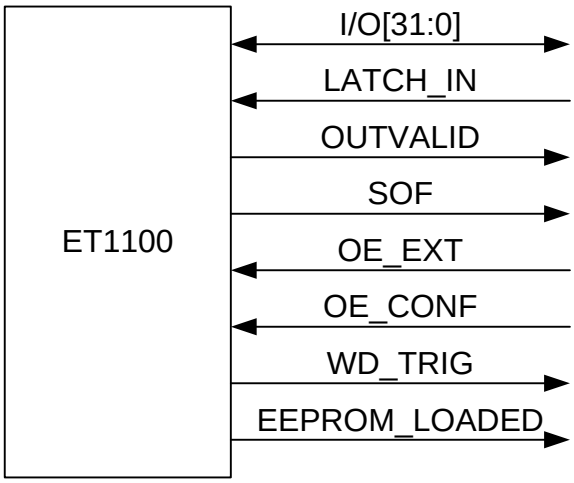

Figure 19: ET1100 Digital I/O signals

Table 57: ET1100 Digital I/O signals

Signal	Direction	Description	Signal polarity
I/O[31:0]	IN/OUT/BIDIR	Input/Output or Bidirectional data	
LATCH_IN	IN	External data latch signal	act. high
OUTVALID	OUT	Output data is valid/Output event	act. high
SOF	OUT	Start of Frame	act. high
OE_EXT	IN	Output Enable	act. high
OE_CONF	IN	Output Enable Configuration	
WD_TRIG	OUT	Watchdog Trigger	act. high
EEPROM_LOADED	OUT	PDI is active, EEPROM is loaded	act. high

6.2.2 Configuration

The Digital I/O interface is selected with PDI type 0x04 in the PDI control register 0x0140. It supports different configurations, which are located in registers 0x0150 – 0x0153.

6.2.3 Digital Inputs

Digital input values appear in the process memory at address 0x1000:0x1003. EtherCAT devices use Little Endian byte ordering, so I/O[7:0] can be read at 0x1000 etc. Digital inputs are written to the process memory by the Digital I/O PDI using standard PDI write operations.

Digital inputs can be configured to be sampled by the ESC in four ways:

- Digital inputs are sampled at the start of each Ethernet frame, so that EtherCAT read commands to address 0x1000:0x1003 will present digital input values sampled at the start of the same frame. The SOF signal can be used externally to update the input data, because the SOF is signaled before input data is sampled.
- The sample time can be controlled externally by using the LATCH_IN signal. The input data is sampled by the ESC each time a rising edge of LATCH_IN is recognized.
- Digital inputs are sampled at Distributed Clocks SYNC0 events.
- Digital inputs are sampled at Distributed Clocks SYNC1 events.

For Distributed Clock SYNC input, SYNC generation must be activated (register 0x0981). SYNC output is not necessary (register 0x0151). SYNC pulse length (registers 0x0982:0x0983) should not be set to 0, because acknowledging of SYNC events is not possible with Digital I/O PDI. Sample time is the beginning of the SYNC event.

6.2.4 Digital Outputs

Digital Output values have to be written to register 0x0F00:0x0F03 (register 0x0F00 controls I/O[7:0] etc.). Digital Output values are not read by the Digital I/O PDI using standard read commands, instead, there is a direct connection for faster response times.

The process data watchdog (register 0x0440) has to be either active or disabled; otherwise digital outputs will not be updated. Digital outputs can be configured to be updated in four ways:

- Digital Outputs are updated at the end of each EtherCAT frame (EOF mode).
- Digital outputs are updated with Distributed Clocks SYNC0 events (DC SYNC0 mode).
- Digital outputs are updated with Distributed Clocks SYNC1 events (DC SYNC1 mode).
- Digital Outputs are updated at the end of an EtherCAT frame which triggered the Process Data Watchdog (with typical SyncManager configuration: a frame containing a write access to at least one of the registers 0x0F00:0x0F03). Digital Outputs are only updated if the EtherCAT frame was correct (WD_TRIG mode).

For Distributed Clock SYNC output, SYNC generation must be activated (register 0x0981). SYNC output is not necessary (register 0x0151). SYNC pulse length (registers 0x0982:0x0983) should not be set to 0, because acknowledging of SYNC events is not possible with Digital I/O PDI. Output time is the beginning of the SYNC event.

An output event is always signaled by a pulse on OUTVALID even if the digital outputs remain unchanged.

For output data to be visible on the I/O signals, the following conditions have to be met:

- SyncManager watchdog must be either active (triggered) or disabled.
- OE_EXT (Output enable) must be high.
- Output values have to be written to the registers 0x0F00:0x0F03 within a valid EtherCAT frame.
- The configured output update event must have occurred.

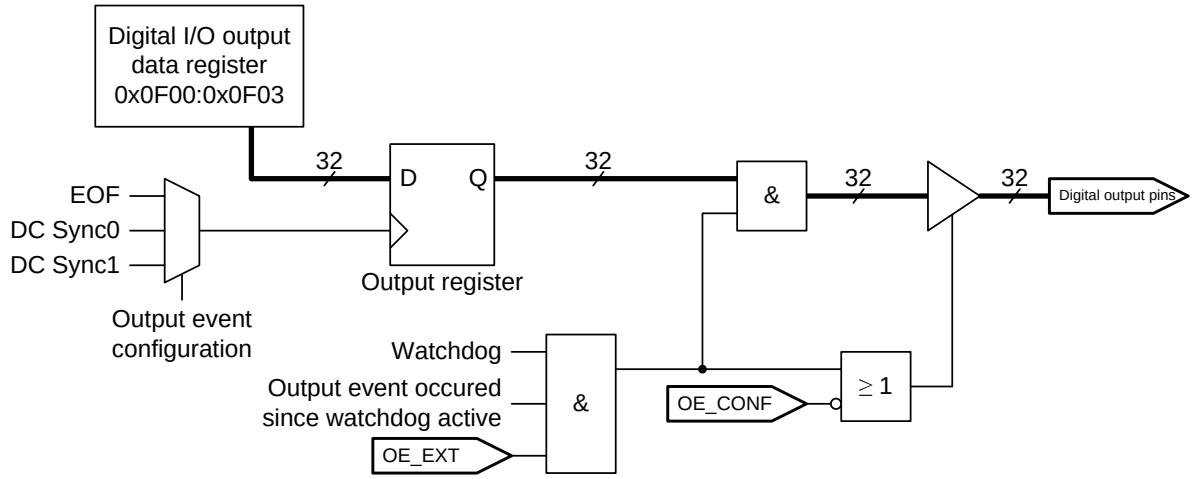

Figure 20: Digital Output Principle Schematic

NOTE: The Digital Outputs are not driven (high impedance) until the EEPROM is loaded. Depending on the configuration, the Digital Outputs are also not driven if the Watchdog is expired or if the outputs are disabled. This behavior has to be taken into account when using digital output signals.

6.2.5 Bidirectional mode

In bidirectional mode, all DATA signals are bidirectional (individual input/output configuration is ignored). Input signals are connected to the ESC via series resistors, output signals are driven actively by the ESC. Output signals are permanently available if they are latched with OUTVALID (Flip-Flop or Latch).

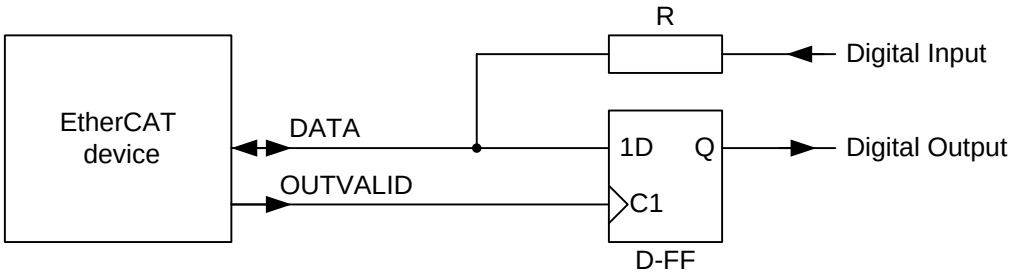

Figure 21: Bidirectional mode: Input/Output connection (R=4.7 kΩ recommended)

Input sample event and output update event can be configured as described in the Digital Inputs/Digital Outputs chapter.

An output event is signaled by a pulse on OUTVALID even if the digital outputs remain unchanged. Overlapping input and output events will lead to corrupt input data.

6.2.6 Output Enable/Output Configuration

The ET1100 has an Output Enable signal OE_EXT and an Output Configuration signal OE_CONF. With the OE_EXT signal, the I/O signals can be cleared/put into a high impedance state. OE_CONF controls the output driver's behavior after the output enable signal OE_EXT is set to low or the SyncManager Watchdog is expired (and not disabled).

Table 58: Output Enable/Output Configuration combinations

OE_CONF	OE_EXT	
	0	1
0	I/O driver: ON I/O: 0	I/O driver: ON I/O: 0 if WD is expired, else output data
1	I/O driver: OFF	I/O driver: OFF if WD is expired or output event has not occurred since WD was last activated I/O: 0 if WD is expired, else output data

OE_CONF is ignored in bidirectional mode, I/O will be driven low during output events if OE_EXT is 0 or the watchdog is expired.

NOTE: I/O drivers are off until the EEPROM is loaded regardless of OE_CONF, OE_EXT, and watchdog.

6.2.7 SyncManager Watchdog

The SyncManager watchdog (registers 0x0440:0x0441) must be either active (triggered) or disabled for output values to appear on the I/O signals. The SyncManager Watchdog is triggered by an EtherCAT write access to the output data registers.

If the output data bytes are written independently, a SyncManager with a length of 1 byte is used for each byte of 0x0F00:0x0F03 containing output bits (SyncManager N configuration: buffered mode, EtherCAT write/PDI read, and Watchdog Trigger enabled: 0x44 in register 0x0804+N*8). Alternatively, if all output data bits are written together in one EtherCAT command, one SyncManager with a length of 1 byte is sufficient (SyncManager N configuration: buffered mode, EtherCAT write/PDI read, and Watchdog Trigger enabled: 0x44 in register 0x0804+N*8). The start address of the SyncManager should be one of the 0x0F00:0x0F03 bytes containing output bits, e.g., the last byte containing output bits.

The SyncManager Watchdog can also be disabled by writing 0 into registers 0x0440:0x0441.

The Watchdog Mode configuration bit is used to configure 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 latter case is especially relevant for Distributed Clock SYNC output events, because any output change will occur at the configured SYNC event.

Immediate output reset after watchdog timeout is not available if OUTVALID mode set to watchdog trigger (0x0150[1]=1).

For external watchdog implementations, the WD_TRIG (watchdog trigger) signal can be used. A WD_TRIG 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 OE_EXT and OE_CONF to reset the I/O signals if the watchdog is expired. For devices without the WD_TRIG signal, OUTVALID can be configured to reflect WD_TRIG.

6.2.8 SOF

SOF indicates the start of an Ethernet/EtherCAT frame. It is asserted shortly after RX_DV=1 or EBUS SOF. Input data is sampled in the time interval between $t_{SOF\_to\_DATA\_setup}$ and $t_{SOF\_to\_DATA\_setup}$ after the SOF signal is asserted.

6.2.9 OUTVALID

A pulse on the OUTVALID signal indicates an output event. If the output event is configured to be the end of a frame, OUTVALID is issued shortly after RX_DV=0 or EBUS EOF, right after the CRC has been checked and the internal registers have taken their new values. OUTVALID is issued independent of actual output data values, i.e., it is issued even if the output data does not change.

6.2.10 EEPROM_LOADED

The EEPROM_LOADED signal indicates that the Digital I/O Interface is operational. Attach a pull-down resistor for proper function, since the PDI pin will not be driven until the EEPROM is loaded.

6.2.11 Timing specifications

Table 59: Digital I/O timing characteristics ET1100

Parameter	Min	Max	Comment
t_{DATA_setup}	7 ns		Input data valid before LATCH_IN
t_{DATA_hold}	3 ns		Input data valid after LATCH_IN
t_{LATCH_IN}	8 ns		LATCH_IN high time
t_{SOF}	35 ns	45 ns	SOF high time
$t_{SOF\_to\_DATA\_setup}$		1,2 µs	Input data valid after SOF, so that Inputs can be read in the same frame
$t_{SOF\_to\_DATA\_hold}$	1,6 µs		Input data invalid after SOF
$t_{input\_event\_delay}$	440 ns		Time between consecutive input events
$t_{OUTVALID}$	75 ns	85 ns	OUTVALID high time
$t_{DATA\_to\_OUTVALID}$	65 ns		Output data valid before OUTVALID
t_{WD_TRIG}	35 ns	45 ns	WD_TRIG high time
$t_{DATA\_to\_WD\_TRIG}$		35 ns	Output data valid after WD_TRIG
$t_{OE\_EXT\_to\_DATA\_invalid}$	0 ns	15 ns	Outputs zero or Outputs high impedance after OE_EXT set to low
$t_{output\_event\_delay}$	320 ns		Time between consecutive output events
$t_{BIDIR\_DATA\_valid}$	65 ns		Bidirectional mode: I/O valid before OUTVALID
$t_{BIDIR\_DATA\_invalid}$	65 ns		Bidirectional mode: I/O invalid after OUTVALID
$t_{BIDIR\_event\_delay}$	440 ns		Bidirectional mode: time between consecutive input and output events

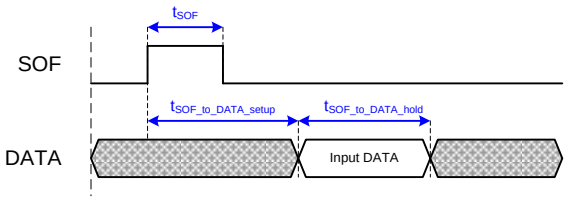

Figure 22: Digital Input: Input data sampled at SOF, I/O can be read in the same frame

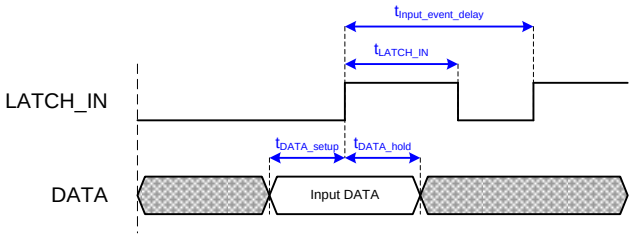

Figure 23: Digital Input: Input data sampled with LATCH_IN

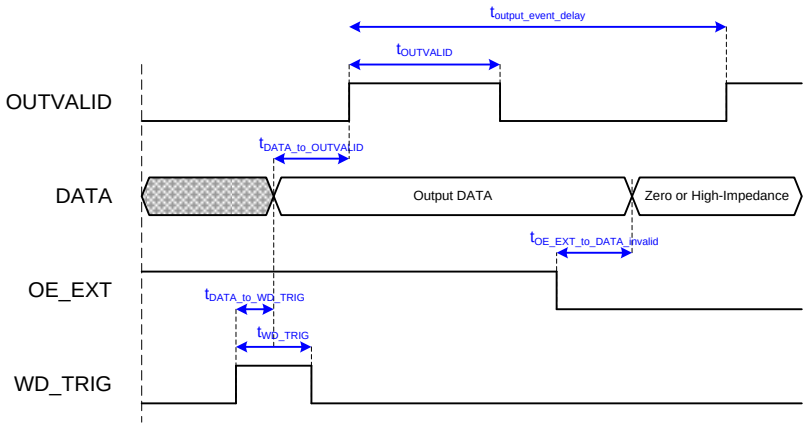

Figure 24: Digital Output timing

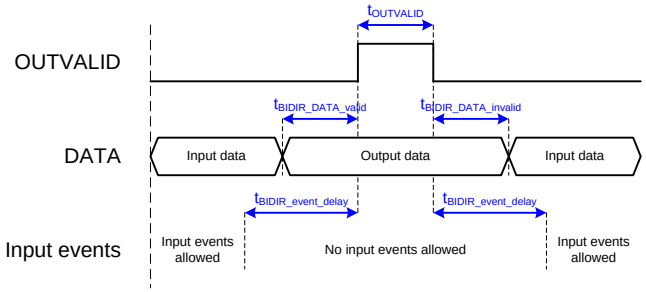

Figure 25: Bidirectional Mode timing

6.3 SPI Slave Interface

6.3.1 Interface

An EtherCAT device with PDI type 0x05 is an SPI slave. The SPI has 5 signals: SPI_CLK, SPI_DI (MOSI), SPI_DO (MISO), SPI_SEL and SPI_IRQ:

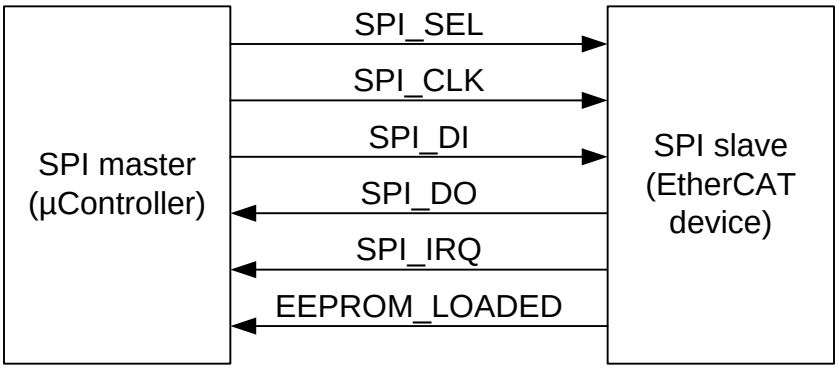

Figure 26: SPI master and slave interconnection

Table 60: SPI signals

Signal	Direction		Description	Signal polarity
SPI_SEL	IN	(master → slave)	SPI chip select	Typical: act. low
SPI_CLK	IN	(master → slave)	SPI clock	
SPI_DI	IN	(master → slave)	SPI data MOSI	act. high
SPI_DO	OUT	(slave → master)	SPI data MISO	act. high
SPI_IRQ	OUT	(slave → master)	SPI interrupt	Typical: act. low
EEPROM_LOADED	OUT	(slave → master)	PDI is active, EEPROM is loaded	act. high

6.3.2 Configuration

The SPI slave interface is selected with PDI type 0x05 in the PDI control register 0x0140. It supports different timing modes and configurable signal polarity for SPI_SEL and SPI_IRQ. The SPI configuration is located in register 0x0150.

6.3.3 SPI access

Each SPI access is separated into an address phase and a data phase. In the address phase, the SPI master transmits the first address to be accessed and the command. In the data phase, read data is presented by the SPI slave (read command) or write data is transmitted by the master (write command). The address phase consists of 2 or 3 bytes depending on the address mode. The number of data bytes for each access may range from 0 to N bytes. The slave internally increments the address for the following bytes after reading or writing the start address. The bits of both address/command and data are transmitted in byte groups.

The master starts an SPI access by asserting SPI_SEL and terminates it by taking back SPI_SEL (polarity determined by configuration). While SPI_SEL is asserted, the master has to cycle SPI_CLK eight times for each byte transfer. In each clock cycle, both master and slave transmit one bit to the other side (full duplex). The relevant edges of SPI_CLK for master and slave can be configured by selecting SPI mode and Data Out sample mode.

The most significant bit of a byte is transmitted first, the least significant bit last, the byte order is low byte first. EtherCAT devices use Little Endian byte ordering.

6.3.4 Commands

The command CMD0 in the second address/command byte may be READ, READ with following Wait State bytes, WRITE, NOP, or Address Extension. The command CMD1 in the third address/command byte may have the same values:

Table 61: SPI commands CMD0 and CMD1

CMD[2]	CMD[1]	CMD[0]	Command
0	0	0	NOP (no operation)
0	0	1	reserved
0	1	0	Read
0	1	1	Read with following Wait State bytes
1	0	0	Write
1	0	1	reserved
1	1	0	Address Extension (3 address/command bytes)
1	1	1	reserved

6.3.5 Address modes

The SPI slave interface supports two address modes, 2 byte addressing and 3 byte addressing. With two byte addressing, the lower 13 address bits A[12:0] are selected by the SPI master, while the upper 3 bits A[15:13] are assumed to be 000b inside the SPI slave, thus only the first 8 Kbyte in the EtherCAT slave address space can be accessed. Three byte addressing is used for accessing the whole 64 Kbyte address space of an EtherCAT slave.

For SPI masters which do only support consecutive transfers of more than one byte, additional Address Extension commands can be inserted.

Table 62: Address modes without (Read access without Wait state byte)

Byte	2 Byte address mode		3 Byte address mode	
0	A[12:5]	address bits [12:5]	A[12:5]	address bits [12:5]
1	A[4:0] CMD0[2:0]	address bits [4:0] read/write command	A[4:0] CMD0[2:0]	address bits [4:0] 3 byte addressing: 110b
2	D0[7:0]	data byte 0	A[15:13] CMD1[2:0] res[1:0]	address bits [15:13] read/write command two reserved bits, set to 00b
3	D1[7:0]	data byte 1	D0[7:0]	data byte 0
4 ff.	D2[7:0]	data byte 2	D1[7:0]	data byte 1

Table 63: Address modes for Read access with Wait state byte

Byte	2 Byte address mode		3 Byte address mode	
0	A[12:5]	address bits [12:5]	A[12:5]	address bits [12:5]
1	A[4:0] CMD0[2:0]	address bits [4:0] read command: 011b	A[4:0] CMD0[2:0]	address bits [4:0] 3 byte addressing: 110b
2	0xFF	wait state byte	A[15:13] CMD1[2:0] res[1:0]	address bits [15:13] read command: 011b two reserved bits, set to 00b
3	D0[7:0]	data byte 0	0xFF	wait state byte
4	D1[7:0]	data byte 1	D0[7:0]	data byte 0
5 ff.	D2[7:0]	data byte 2	D1[7:0]	data byte 1

6.3.6 Interrupt request register (AL Event register)

During the address phase, the SPI slave transmits the PDI interrupt request registers 0x0220-0x0221 (2 byte address mode), and additionally register 0x0222 for 3 byte addressing on SPI_DO (MISO):

Table 64: Interrupt request register transmission

Byte	2 Byte address mode			3 Byte address mode		
	SPI_DI (MOSI)	SPI_DO (MISO)		SPI_DI (MOSI)	SPI_DO (MISO)	
0	A[12:5]	I0[7:0]	interrupt request register 0x0220	A[12:5]	I0[7:0]	interrupt request register 0x0220
1	A[4:0] CMD0[2:0]	I1[7:0]	interrupt request register 0x0221	A[4:0] CMD0[2:0]	I1[7:0]	interrupt request register 0x0221
2	(Data phase)			A[15:13] CMD1[2:0]	I2[7:0]	interrupt request register 0x0222

6.3.7 Write access

In the data phase of a write access, the SPI master sends the write data bytes to the SPI slave (SPI_DI/MOSI). The write access is terminated by taking back SPI_SEL after the last byte. The SPI_DO signal (MISO) is undetermined during the data phase of write accesses.

6.3.8 Read access

In the data phase of a read access, the SPI slave sends the read data bytes to the SPI master (SPI_DO/MISO).

6.3.8.1 Read Wait State

Between the last address phase byte and the first data byte of a read access, the SPI master has to wait for the SPI slave to fetch the read data internally. Subsequent read data bytes are prefetched automatically, so no further wait states are necessary.

The SPI master can choose between these possibilities:

- The SPI master may either wait for the specified worst case internal read time t_{read} after the last address/command byte and before the first clock cycle of the data phase.
- The SPI master inserts one Wait State byte after the last address/command byte. The Wait State byte must have a value of 0xFF transferred on SPI_DI.

6.3.8.2 Read Termination

The SPI_DI signal (MOSI) is used for termination of the read access by the SPI master. For the last data byte, the SPI master has to set SPI_DI to high (Read Termination byte = 0xFF), so the slave will

not prefetch the next read data internally. If SPI_DI is low during a data byte transfer, at least one more byte will be read by the master afterwards.

6.3.9 SPI access errors and SPI status flag

The following reasons for SPI access errors are detected by the SPI slave:

- The number of clock cycles recognized while SPI_SEL is asserted is not a multiple of 8 (incomplete bytes were transferred).
- For a read access, a clock cycle occurred while the slave was busy fetching the first data byte.
- For a read access, the data phase was not terminated by setting SPI_DI to high for the last byte.
- For a read access, additional bytes were read after termination of the access.

A wrong SPI access will have these consequences:

- Registers will not accept write data (nevertheless, RAM will be written).
- Special functions are not executed (e.g., SyncManager buffer switching).
- The PDI error counter 0x030D will be incremented.
- A status flag will indicate the error until the next access (not for SPI mode 0/2 with normal data out sample)

A status flag, which indicates if the last access had an error, is available in any mode except for SPI mode 0/2 with normal data out sample. The status flag is presented on SPI_DO (MISO) after the slave is selected (SPI_SEL) and until the first clock cycle occurs. So the status can be read either between two accesses by assertion of SPI_SEL without clocking, or at the beginning of an access just before the first clock cycle. The status flag will be high for a good access, and low for a wrong access.

6.3.10 EEPROM_LOADED

The EEPROM_LOADED signal indicates that the SPI Interface is operational. Attach a pull-down resistor for proper function, since the PDI pin will not be driven until the EEPROM is loaded.

6.3.11 2 Byte and 4 Byte SPI Masters

Some SPI masters do not allow an arbitrary number of bytes per access, the number of bytes per access must be a multiple of 2 or 4 (maybe even more). The SPI slave interface supports such masters. The length of the data phase is in control of the master and can be set to the appropriate length, the length of the address phase has to be extended. The address phase of a read access can be set to a multiple of 2/4 by using the 3 byte address mode and a wait state byte. The address phase of a write access can be enhanced to 4 bytes using 3 byte address mode and an additional address extension byte (byte 2) according to Table 65.

Table 65: Write access for 2 and 4 Byte SPI Masters

Byte	2 Byte SPI master		4 Byte SPI master	
0	A[12:5]	address bits [12:5]	A[12:5]	address bits [12:5]
1	A[4:0] CMD0[2:0]	address bits [4:0] write command: 100b	A[4:0] CMD0[2:0]	address bits [4:0] 3 byte addressing: 110b
2	D0[7:0]	data byte 0	A[15:13] CMD1[2:0] res[1:0]	address bits [15:13] 3 byte addressing: 110b two reserved bits, set to 00b
3	D1[7:0]	data byte 1	A[15:13] CMD2[2:0] res[1:0]	address bits [15:13] write command: 100b two reserved bits, set to 00b
4	D2[7:0]	data byte 2	D0[7:0]	data byte 0
5	D3[7:0]	data byte 3	D1[7:0]	data byte 1
6	D4[7:0]	data byte 4	D2[7:0]	data byte 2
7	D5[7:0]	data byte 5	D3[7:0]	data byte 3

NOTE: The address phase of a write access can be further extended by an arbitrary number of address extension bytes containing 110b as the command. The address phase of a read access can also be enhanced with additional address extension bytes (the read wait state has to be maintained anyway). The address portion of the last address extension byte is used for the access.

6.3.12 Timing specifications

Table 66: SPI timing characteristics ET1100

Parameter	Min	Max	Comment
t_{CLK}	50 ns		SPI_CLK frequency (f_{CLK} ≤ 20 MHz)
$t_{SEL\_to\_CLK}$	6 ns		First SPI_CLK cycle after SPI_SEL asserted
$t_{CLK\_to\_SEL}$	a) 5 ns b) t_{CLK}/2+5 ns		Deassertion of SPI_SEL after last SPI_CLK cycle a) SPI mode 0/2, SPI mode 1/3 with normal data out sample b) SPI mode 1/3 with late data out sample
t_{read}	240 ns		Only for read access between address/command and first data byte. Can be ignored if Wait State Bytes are used.
$t_{SEL\_to\_DO\_valid}$		15 ns	Status/Interrupt Byte 0 bit 7 valid after SPI_SEL asserted
$t_{SEL\_to\_DO\_invalid}$	0 ns		Status/Interrupt Byte 0 bit 7 invalid after SPI_SEL de-asserted
t_{STATUS_valid}	12 ns		Time until status of last access is valid. Can be ignored if status is not used.
t_{access_delay}	a) 15 ns b) 240 ns		Delay between SPI accesses a) typical b) If last access was shorter than 2 bytes, otherwise Interrupt Request Register value I0_[7:0] will not be valid.
t_{DI_setup}	9 ns		SPI_DI valid before SPI_CLK edge
t_{DI_hold}	3 ns		SPI_DI valid after SPI_CLK edge
$t_{CLK\_to\_DO\_valid}$		15 ns	SPI_DO valid after SPI_CLK edge
$t_{CLK\_to\_DO\_invalid}$	0 ns		SPI_DO invalid after SPI_CLK edge
$t_{EEPROM\_LOADED\_to\_access}$	0 ns		Time between EEPROM_LOADED and first access
t_{IRQ_delay}	160 ns		Internal delay between AL event and SPI_IRQ output to enable correct reading of the interrupt registers.

Table 67: Read/Write timing diagram symbols

Symbol	Comment
A15..A0	Address bits [15:0]
D0_7..D0_0 D1_7..D1_0	Data bits byte 0 [7:0] Data bits byte 1 [7:0]
I0_7..I0_0 I1_7..I1_0 I2_7..I2_0	Interrupt request register 0x0220 [7:0] Interrupt request register 0x0221 [7:0] Interrupt request register 0x0222 [7:0]
C0_2..C0_0 C1_2..C1_0	Command 0 [2:0] Command 1 [2:0] (3 byte addressing)
Status	0: last SPI access had errors 1: last SPI access was correct
BUSY OUT Enable	0: No Busy output, tread is relevant 1: Busy output on SPI_DO (edge sensitive)
BUSY	0: SPI slave has finished reading first byte 1: SPI slave is busy reading first byte

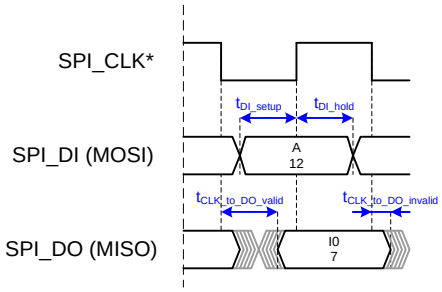

Figure 27: Basic SPI_DI/SPI_DO timing (*refer to timing diagram for relevant edges of SPI_CLK)

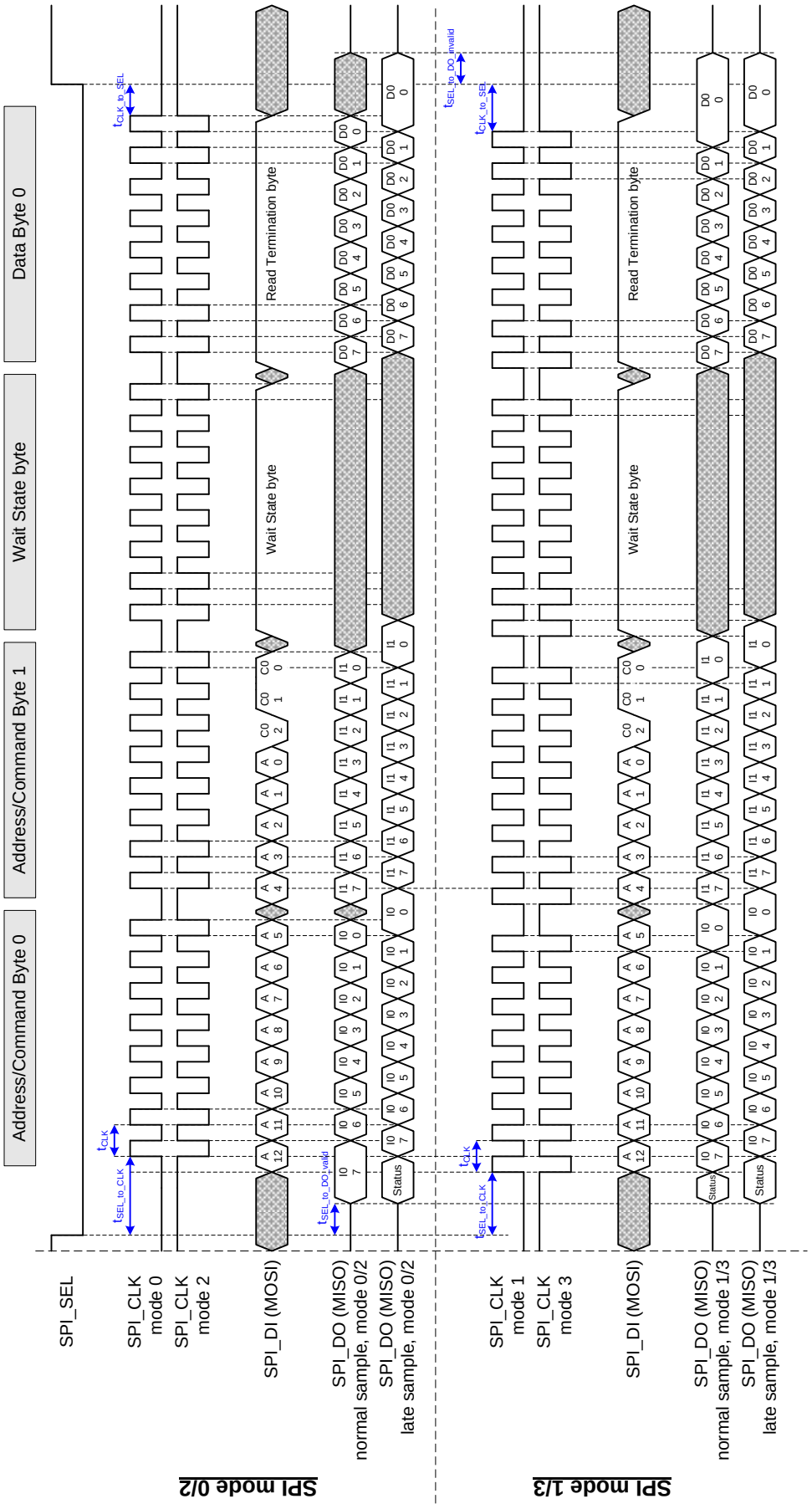

Figure 28: SPI read access (2 byte addressing, 1 byte read data) with Wait State byte

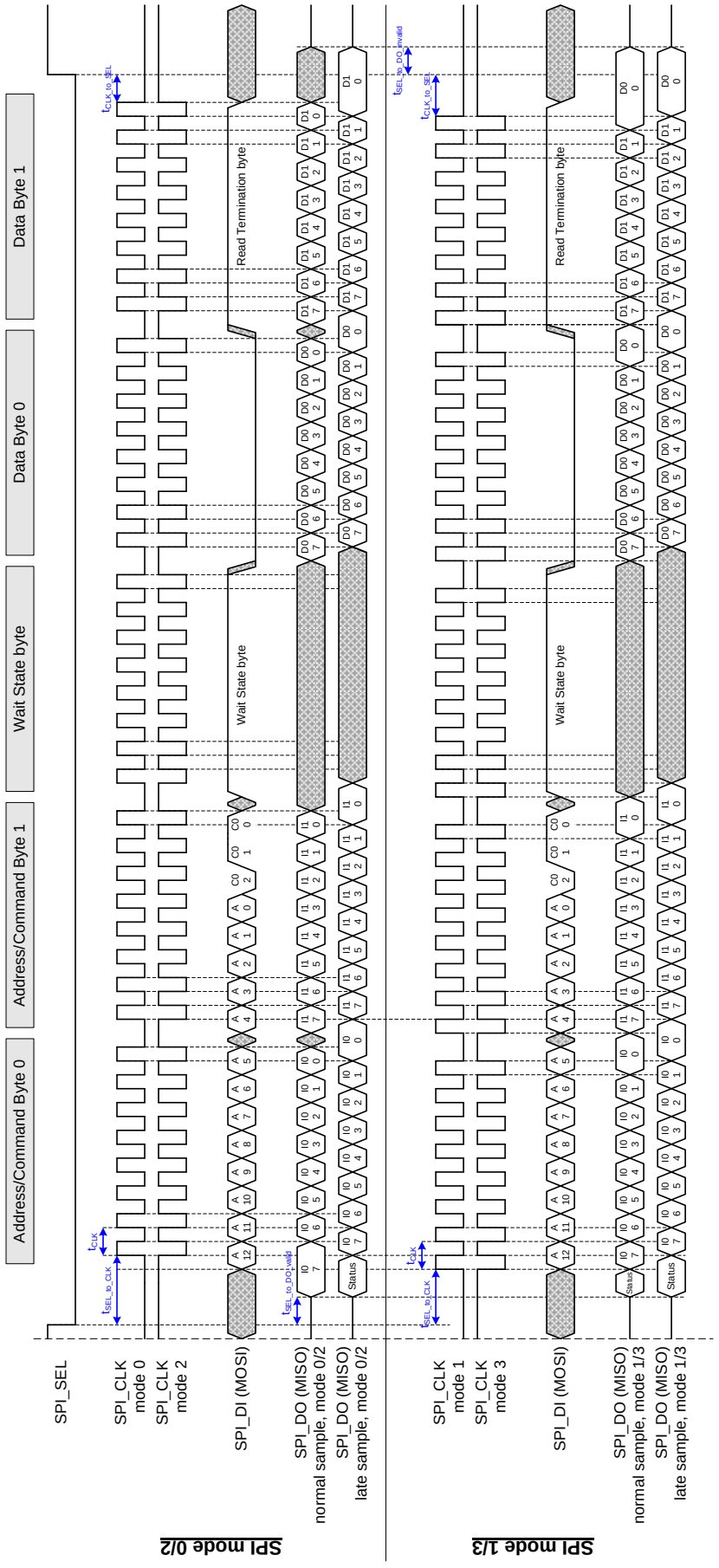

Figure 29: SPI read access (2 byte addressing, 2 byte read data) with Wait State byte

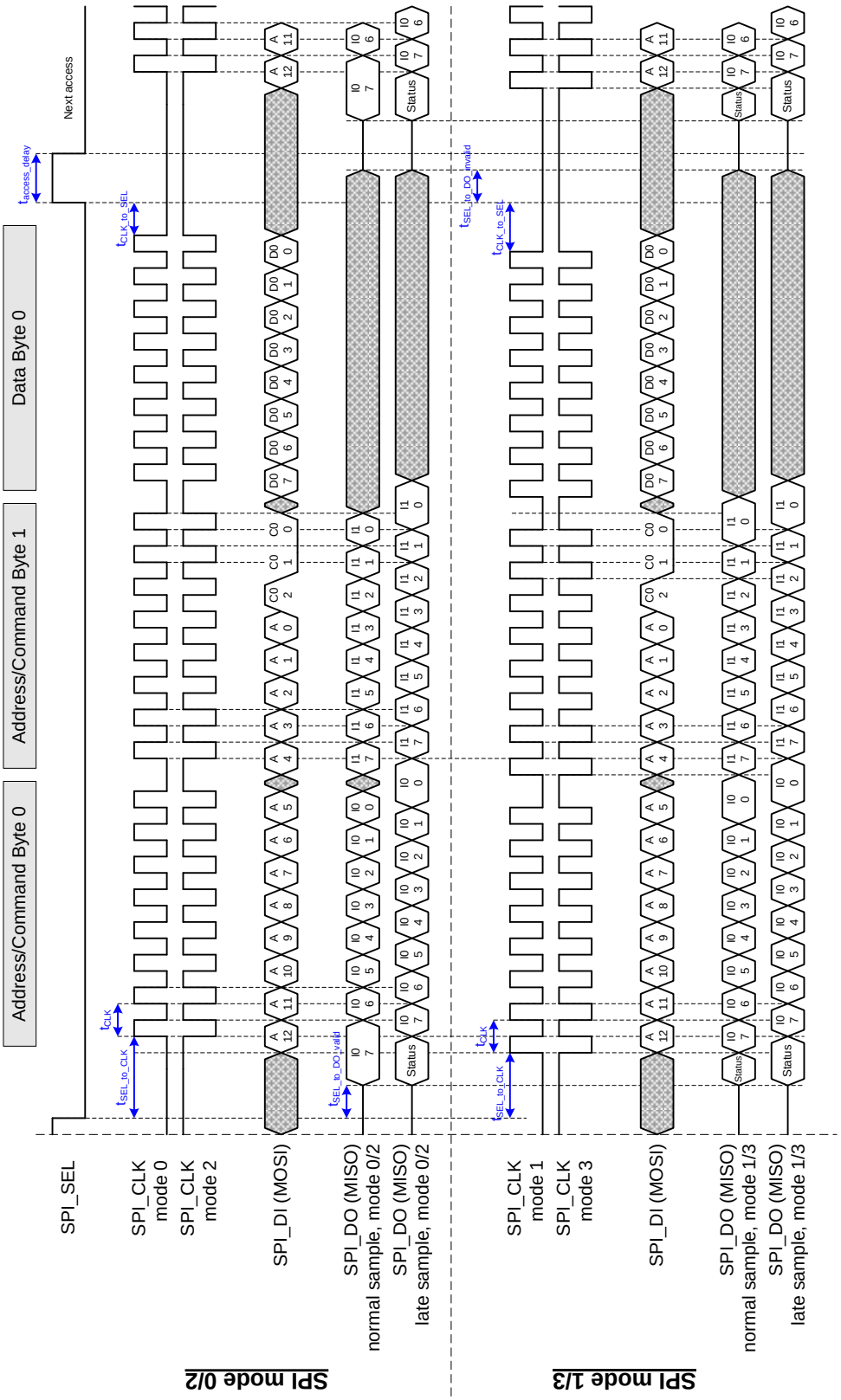

Figure 30: SPI write access (2 byte addressing, 1 byte write data)

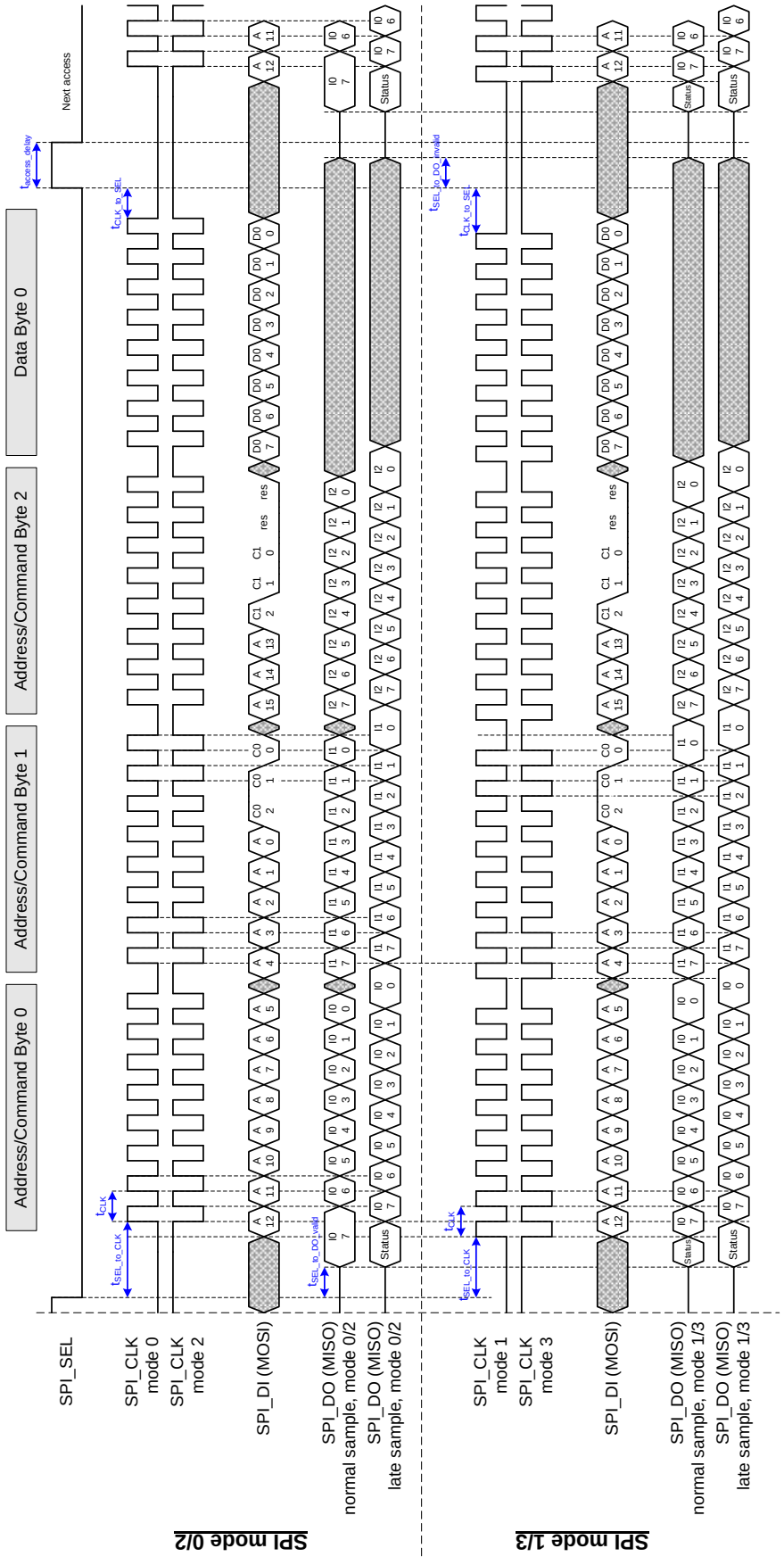

Figure 31: SPI write access (3 byte addressing, 1 byte write data)

6.4 Asynchronous 8/16 bit µController Interface

6.4.1 Interface

The asynchronous µController interface uses demultiplexed address and data busses. The bidirectional data bus can be either 8 bit or 16 bit wide. The signals of the asynchronous µController interface of EtherCAT devices are:

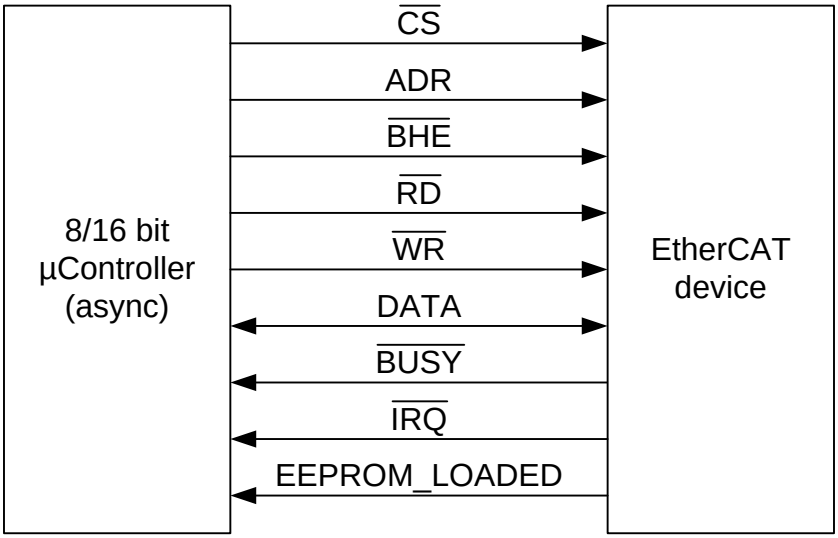

Figure 32: µController interconnection[3]

Table 68: µController signals

Signal async	Direction		Description	Signal polarity
CS	IN	(µC → ESC)	Chip select	Typical: act. low
ADR[15:0]	IN	(µC → ESC)	Address bus	Typical: act. high
BHE	IN	(µC → ESC)	Byte High Enable (16 bit µController interface only)	Typical: act. low
RD	IN	(µC → ESC)	Read command	Typical: act. low
WR	IN	(µC → ESC)	Write command	Typical: act. low
DATA[15:0]	BD	(µC ↔ ESC)	Data bus for 16 bit µController interface	act. high
DATA[7:0]	BD	(µC ↔ ESC)	Data bus for 8 bit µController interface	act. high
BUSY	OUT	(ESC → µC)	EtherCAT device is busy	Typical: act. low
IRQ	OUT	(ESC → µC)	Interrupt	Typical: act. low
EEPROM_ LOADED	OUT	(ESC → µC)	PDI is active, EEPROM is loaded	act. high

Some µControllers have a READY signal, this is the same as the BUSY signal, just with inverted polarity.

6.4.2 Configuration

The 16 bit asynchronous µController interface is selected with PDI type 0x08 in the PDI control register 0x0140, the 8 bit asynchronous µController interface has PDI type 0x09. It supports different configurations, which are located in registers 0x0150 – 0x0153.

[3] All signals are denoted with typical polarity configuration.

6.4.3 µController access

The 8 bit µController interface reads or writes 8 bit per access, the 16 bit µController interface supports both 8 bit and 16 bit read/write accesses. For the 16 bit µController interface, the least significant address bit together with Byte High Enable (BHE) are used to distinguish between 8 bit low byte access, 8 bit high byte access and 16 bit access.

EtherCAT devices use Little Endian byte ordering.

Table 69: 8 bit µController interface access types

ADR[0]	Access	DATA[7:0]
0	8 bit access to ADR[15:0] (low byte, even address)	low byte
1	8 bit access to ADR[15:0] (high byte, odd address)	high byte

Table 70: 16 bit µController interface access types

ADR[0]	BHE (act. low)	Access	DATA[15:8]	DATA[7:0]
0	0	16 bit access to ADR[15:0] and ADR[15:0]+1 (low and high byte)	high byte	low byte
0	1	8 bit access to ADR[15:0] (low byte, even address)	(RD only: copy of low byte)	low byte
1	0	8 bit access to ADR[15:0] (high byte, odd address)	high byte	(RD only: copy of high byte)
1	1	invalid access	-	-

6.4.4 Write access

A write access starts with assertion of Chip Select (CS), if it is not permanently asserted. Address, Byte High Enable and Write Data are asserted with the falling edge of WR (active low). Once the µController interface is not BUSY, a rising edge on WR completes the µController access. A write access can be terminated either by deassertion of WR (while CS remains asserted), or by deassertion or CS (while WR remains asserted), or even by deassertion of WR and CS simultaneously. Shortly after the rising edge of WR, the access can be finished by de-asserting ADR, BHE and DATA. The µController interface indicates its internal operation with the BUSY signal. Since the BUSY signal is only driven while CS is asserted, the BUSY driver will be released after CS deassertion.

Internally, the write access is performed after the rising edge of WR, this allows for fast write accesses. Nevertheless, an access following immediately will be delayed by the preceding write access (BUSY is active for a longer time).

6.4.5 Read access

A read access starts with assertion of Chip Select (CS), if it is not permanently asserted. Address and BHE have to be valid before the falling edge of RD, which signals the start of the access. The µController interface will show its BUSY state afterwards – if it is not already busy executing a preceding write access – and release BUSY when the read data are valid. The read data will remain valid until either ADR, BHE, RD or CS change. The data bus will be driven while CS and RD are asserted. BUSY will be driven while CS is asserted.

With read busy delay configuration, BUSY deassertion for read accesses can be additionally delayed for 20 ns, so external DATA setup requirements in respect to BUSY can be met.

6.4.6 µController access errors

These reasons for µController access errors are detected by the µController interface:

- Read or Write access to the 16 bit interface with A[0]=1 and BHE(act. low)=1, i.e. an access to an odd address without Byte High Enable.
- Deassertion of WR (or deassertion of CS while WR remains asserted) while the µController interface is BUSY.
- Deassertion of RD (or deassertion of CS while RD remains asserted) while the µController interface is BUSY (read has not finished).

A wrong µController access will have these consequences:

- The PDI error counter 0x030D will be incremented.
- For A[0]=1 and BHE(act. low)=1 accesses, no access will be performed internally.
- Deassertion of WR (or CS) while the µController interface is BUSY 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 de-asserted while the µController interface is BUSY (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.

6.4.7 EEPROM_LOADED

The EEPROM_LOADED signal indicates that the µController Interface is operational. Attach a pull-down resistor for proper function, since the PDI pin will not be driven until the EEPROM is loaded.

6.4.8 Connection with 16 bit µControllers without byte addressing

If the ESC is connected to 16 bit µControllers/DSPs which only support 16 bit (word) addressing, ADR[0] and BHE of the EtherCAT device have to be tied to GND, so the ESC will always perform 16 bit accesses. All other signals are connected as usual. Please note that ESC addresses have to be divided by 2 in this case.

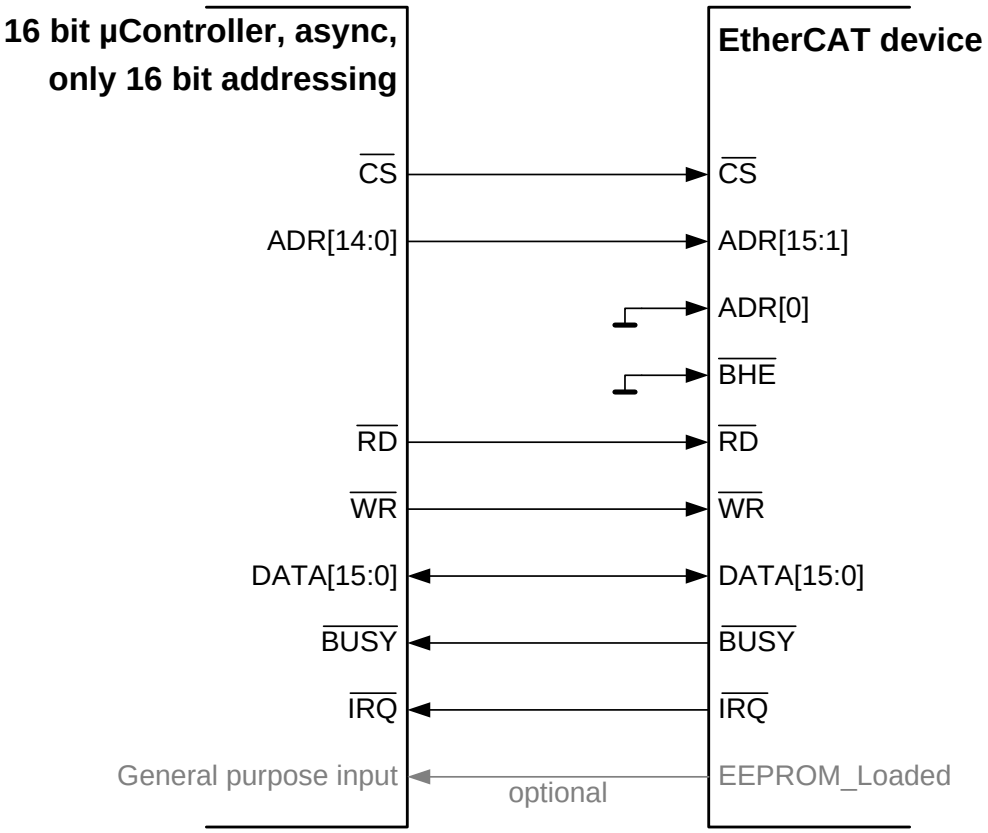

Figure 33: Connection with 16 bit µControllers without byte addressing

6.4.9 Connection with 8 bit µControllers

If the ESC is connected to 8 bit µControllers, the BHE signal as well as the DATA[15:8] signals are not used.

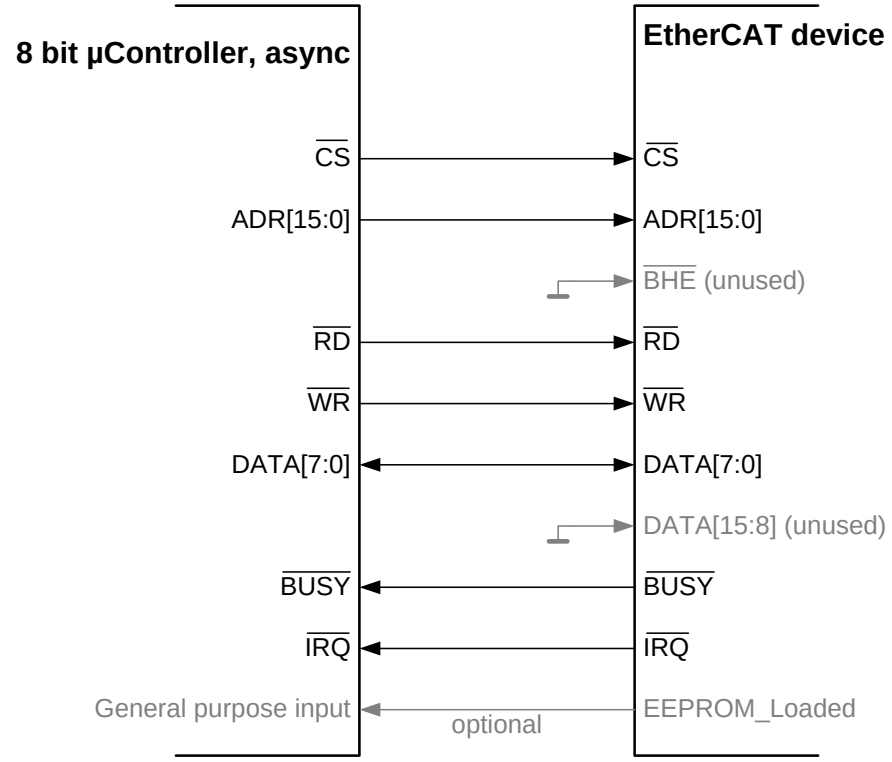

Figure 34: Connection with 8 bit µControllers (BHE and DATA[15:8] should not be left open)

6.4.10 Timing Specification

Table 71: µController timing characteristics ET1100

Parameter	Min	Max	Comment
$t_{CS\_to\_BUSY}$		15 ns	BUSY driven and valid after CS assertion
$t_{ADR\_BHE\_setup}$	-2 ns		ADR and BHE valid before RD assertion
$t_{RD\_to\_DATA\_driven}$	0 ns		DATA bus driven after RD assertion
$t_{RD\_to\_BUSY}$	0 ns	15 ns	BUSY asserted after RD assertion
t_{read}			External read time (RD assertion to BUSY deassertion) with normal read busy output (0x0152[0]). Additional 20 ns with delayed read busy output.
		a) t_{read_int} + t_{prec_write} +t_{Coll} -$t_{WR\_to\_RD}$	a) with preceding write access and $t_{WR\_to\_RD} < t_{prec\_write} + t_{Coll}$
		b) t_{read_int}	b) without preceding write access or $t_{WR\_to\_RD} \geq t_{prec\_write} + t_{Coll}$
		c) 435 ns	c) 8 bit access, absolute worst case with preceding 8 bit write access ($t_{WR\_to\_RD}$=min, t_{prec_write} =max, t_{Coll}=max)
		d) 575 ns	d) 16 bit access, absolute worst case with preceding 16 bit write access ($t_{WR\_to\_RD}$=min, t_{prec_write} =max, t_{Coll}=0)
t_{read_int}		a) 235 ns b) 315 ns	Internal read time a) 8 bit access b) 16 bit access
t_{prec_write}		a) 180 ns b) 260 ns	Time for preceding write access a) 8 bit access b) 16 bit access
$t_{BUSY\_to\_DATA\_valid}$		a) 5 ns b) -15 ns	DATA bus valid after device BUSY is de-asserted a) normal read busy output b) delayed read busy output
$t_{ADR\_BHE\_to\_DATA\_invalid}$	0 ns		DATA invalid after ADR or BHE change
$t_{CS\_RD\_to\_DATA\_release}$	0 ns		DATA bus released after CS deassertion or RD deassertion
$t_{CS\_to\_BUSY\_release}$	0 ns	15 ns	BUSY released after CS deassertion
t_{CS_delay}	0 ns		Delay between CS deassertion an assertion
t_{RD_delay}	10 ns		Delay between RD deassertion and assertion
$t_{ADR\_BHE\_DATA\_setup}$	10 ns		ADR, BHE and Write DATA valid before WR deassertion
$t_{ADR\_BHE\_DATA\_hold}$	3 ns		ADR, BHE and Write DATA valid after WR deassertion
t_{WR_active}	10 ns		WR assertion time
$t_{BUSY\_to\_WR\_CS}$	0 ns		WR or CS deassertion after BUSY deassertion
$t_{WR\_to\_BUSY}$		15 ns	BUSY assertion after WR deassertion
t_{write}	0 ns		External write time (WR assertion to BUSY deassertion)

Parameter	Min	Max	Comment
		a) t_{write_int} -t_{WR_delay}	a) with preceding write access and $t_{WR\_delay} < t_{write\_int}$
		b) 0 ns	b) without preceding write access or $t_{WR\_delay} \geq t_{write\_int}$
		c) 200 ns	c) 8 bit access, absolute worst case with preceding 8 bit write access (t_{WR_delay}= min, t_{WR_int}=max)
		d) 280 ns	d) 16 bit access, absolute worst case with preceding 16 bit write access (t_{WR_delay}=min, t_{WR_int}=max)
t_{write_int}		a) 200 ns b) 280 ns	Internal write time a) 8 bit access b) 16 bit access
t_{WR_delay}	10 ns		Delay between WR deassertion and assertion
t_{Coll}		a) 20 ns b) 0 ns	Extra read delay a) RD access directly follows WR access with the same address (8 bit accesses or 8 bit WR and 16 bit RD) b) different addresses or 16 bit accesses
$t_{WR\_to\_RD}$	0 ns		Delay between WR deassertion and RD assertion
$t_{CS\_WR\_overlap}$	5 ns		Time both CS and WR have to be de-asserted simultaneously (only if CS is de-asserted at all)
$t_{CS\_RD\_overlap}$	5 ns		Time both CS and RD have to be de-asserted simultaneously (only if CS is de-asserted at all)
$t_{EEPROM\_LOADED\_to\_access}$	0 ns		Time between EEPROM_LOADED and first access
$t_{EEPROM\_LOADED\_to\_IRQ}$		0 ns	IRQ valid after EEPROM_LOADED

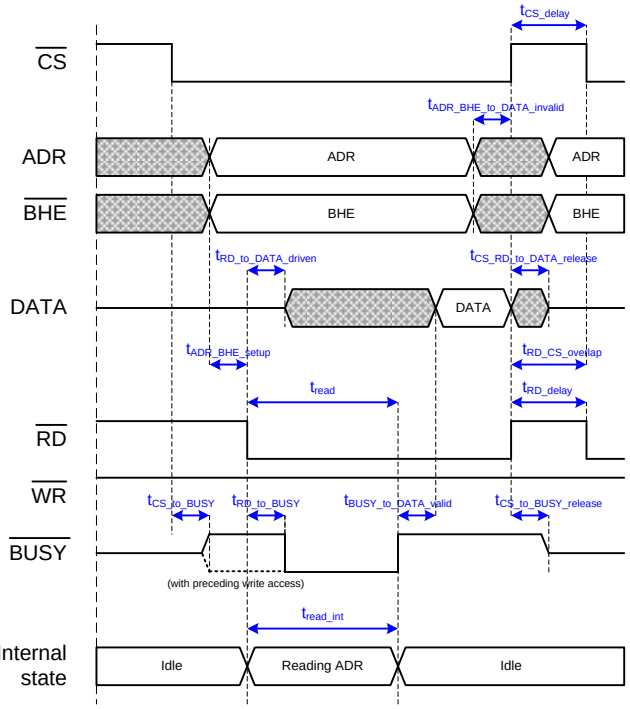

Figure 35: Read access (without preceding write access)

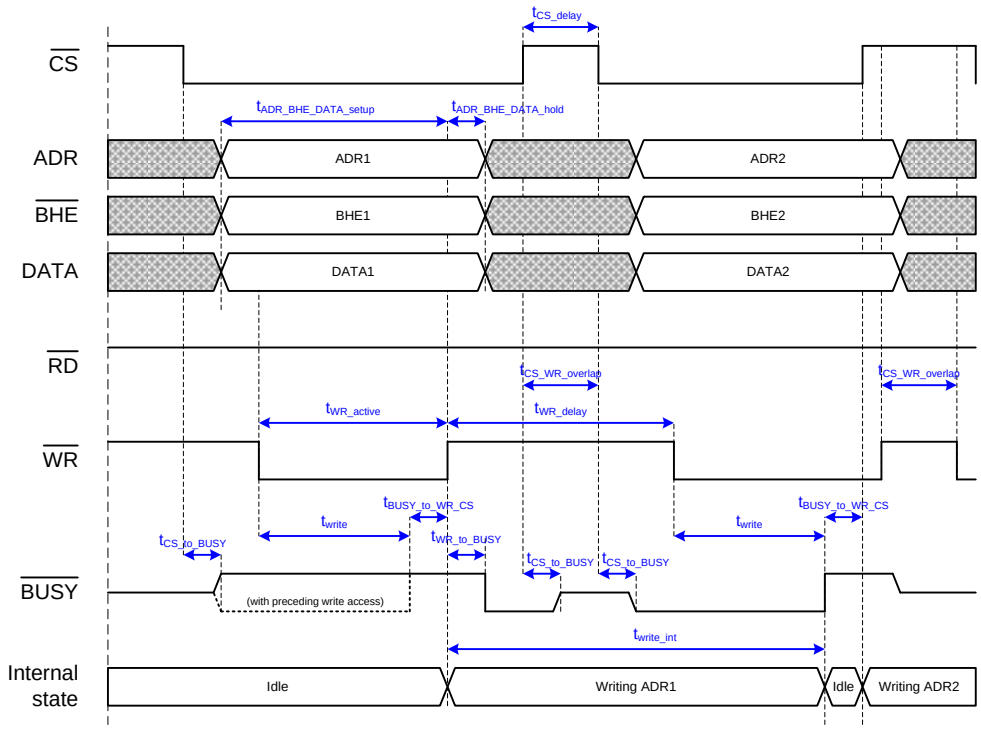

Figure 36: Write access (write after rising edge nWR, without preceding write access)

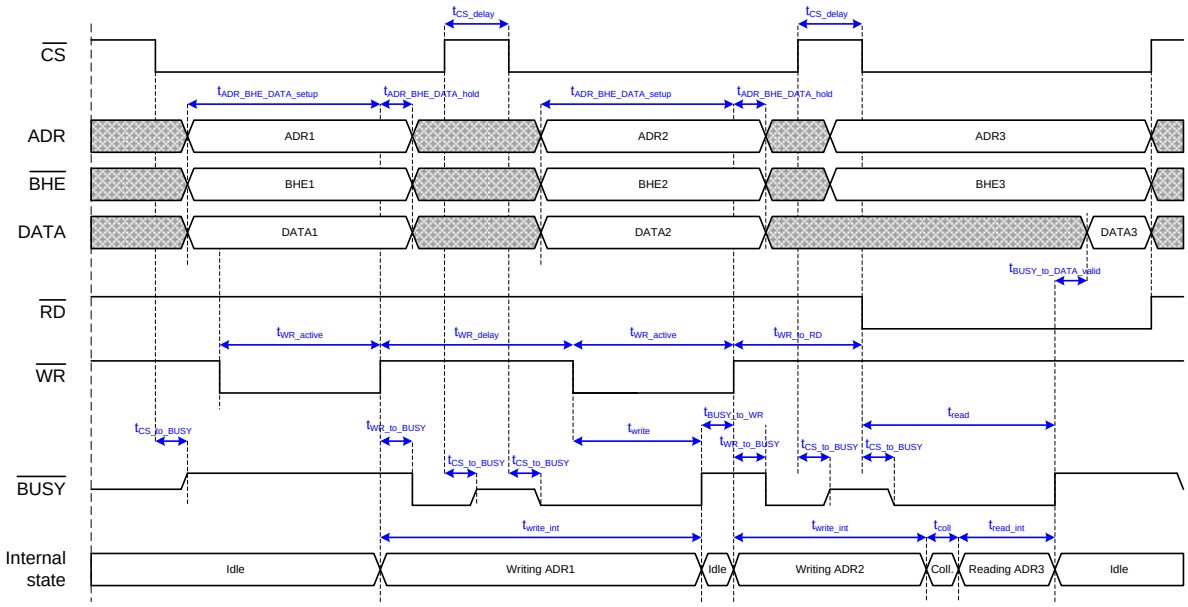

Figure 37: Sequence of two write accesses and a read access

Note: The first write access to ADR1 is performed after the first rising edge of WR. After that, the ESC is internally busy writing to ADR1. After CS is de-asserted, BUSY is not driven any more, nevertheless, the ESC is still writing to ADR1.

Hence, the second write access to ADR2 is delayed because the write access to ADR1 has to be completed first. So, the second rising edge of WR must not occur before BUSY is gone. After the second rising edge of WR, the ESC is busy writing to ADR2. This is reflected with the BUSY signal as long as CS is asserted.

The third access in this example is a read access. The ESC is still busy writing to ADR2 while the falling edge of RD occurs. In this case, the write access to ADR2 is finished first, and afterwards, the read access to ADR3 is performed. The ESC signals BUSY during both write and read access.

6.5 Synchronous 8/16 bit µController Interface

6.5.1 Interface

The synchronous µController interface uses demultiplexed address and data busses. The bidirectional data bus can be either 8 bit or 16 bit wide. The signals of the synchronous µController interface of EtherCAT devices are:

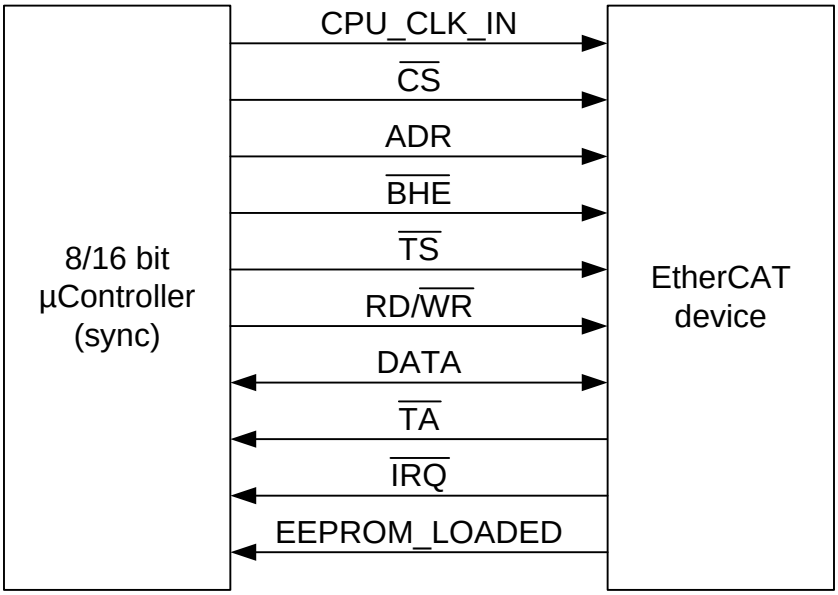

Figure 38: µController interconnection[4]

Table 72: µController signals

Signal sync I/F	Signal async I/F	Direction		Description	Signal polarity
CPU_CLK_IN	N/A	IN	(µC → ESC)	µController interface clock	
CS	CS	IN	(µC → ESC)	Chip select	Typical: act. low
ADR[15:0]	ADR[15:0]	IN	(µC → ESC)	Address bus	act. high
BHE	BHE	IN	(µC → ESC)	Byte High Enable	Typical: act. low
TS	RD	IN	(µC → ESC)	Transfer Start	Typical: act. low
RD/nWR	WR	IN	(µC → ESC)	Read/Write access	
DATA[15:0]	DATA[15:0]	BD	(µC ↔ ESC)	Data bus for 16 Bit µController interface	act. high
DATA[7:0]	DATA[7:0]	BD	(µC ↔ ESC)	Data bus for 8 Bit µController interface	act. high
TA	BUSY	OUT	(ESC → µC)	Transfer Acknowledge	Typical: act. low
IRQ	IRQ	OUT	(ESC → µC)	Interrupt	Typical: act. low
EEPROM_ LOADED	EEPROM_ LOADED	OUT	(ESC → µC)	PDI is active, EEPROM is loaded	act. high

6.5.2 Configuration

The 16 bit synchronous µController interface is selected with PDI type 0x0A in the PDI control register 0x0140, the 8 bit synchronous µController interface has PDI type 0x0B. It supports different configurations, which are located registers 0x0150 – 0x0153.

[4] All signals are denoted with typical polarity configuration.

6.5.3 µController access

The 8 bit µController interface reads or writes 8 bit per access, the 16 bit µController interface supports both 8 bit and 16 bit read/write accesses. The least significant address bit A[0] together with Byte High Enable (BHE) are used to distinguish between 8 bit low byte access, 8 bit high byte access and 16 bit access.

Table 73: 8 bit high/low byte and 16 bit access distinction

ADR[0]	BHE (act. low)	Access
0	0	16 bit access to ADR[15:0] and ADR[15:0]+1 (low and high byte)
0	1	8 bit access to ADR[15:0] (low byte, even address)
1	0	8 bit access to ADR[15:0] (high byte, odd address)
1	1	invalid access

If Byte High Enable (BHE) is used, the Byte access mode configuration bit has to be set to zero (BHE or Byte Select mode).

EtherCAT devices use Little Endian byte ordering, even with the synchronous µController interface. The conversion between Little Endian and Big Endian, depending on the register size of 8, 16, 32, or 64 bit, has to be done in software.

NOTE: A µController with 32 Bit interface is used as an example connected to the synchronous µController interface. It is also possible to use 8 or 16 Bit µControllers.

NOTE: Please compare the bit ordering ([0:31] instead of [31:0]) of your µController with that used in this document, because it might be different. The MSB/LSB notation used below will help you.

Table 74: Corresponding Bytes and Bits

Address	0	1	2	3
µController	[31:24]	[23:16]	[15:8]	[7:0]
	[MSBit:LSBit]	[MSBit:LSBit]	[MSBit:LSBit]	[MSBit:LSBit]
	MSByte	:		LSByte
ESC 8 Bit access	[7:0]			
	[MSBit:LSBit]			
ESC 16 Bit access	[7:0]	[15:8]		
	[MSBit:LSBit]	[MSBit:LSBit]		
	LSByte :	MSByte		

Table 75: Byte ordering

Addr.	ESC (Little Endian)				sync. µController (Big Endian)			
	8 bit reg.	16 bit reg.	32 bit reg.	64 bit reg.	8 bit reg.	16 bit reg.	32 bit reg.	64 bit reg.
0	Byte 0	LSB	LSB	LSB	Byte 0	MSB	MSB	MSB
1	Byte 1	MSB	|	|	Byte 1	LSB	|	|
2	Byte 2	LSB	|	|	Byte 2	MSB	|	|
3	Byte 3	MSB	MSB	|	Byte 3	LSB	LSB	|
4	Byte 4	LSB	LSB	|	Byte 4	MSB	MSB	|
5	Byte 5	MSB	|	|	Byte 5	LSB	|	|
6	Byte 6	LSB	|	|	Byte 6	MSB	|	|
7	Byte 7	MSB	MSB	MSB	Byte 7	LSB	LSB	LSB

6.5.4 µController connection using Byte Select signals (BSn)

In case the µController does not provide Byte High Enable, and Byte Select signals (BS2, and BS3 for 32 bit µController) are available, they can be used to distinguish between 8 and 16 bit accesses. The signal BS3 (active low) is equivalent to ADR[0], and BS2 (active low) is equivalent to BHE (active low).

For Byte Select mode the Byte access mode configuration bit has to be set to zero (BHE or Byte Select mode).

Table 76: Byte Select vs. A[0] and BHE

µController			EtherCAT device		Access
ADR[0]	nBS3	nBS2	ADR[0]	BHE (act. low)	
0	0	0	0	0	16 bit access (low and high byte)
0	0	1	0	1	8 bit access (low byte, even address)
1	1	0	1	0	8 bit access (high byte, odd address)
1	1	1	1	1	invalid access

The following figure shows how a 32 bit µController can be connected with the EtherCAT synchronous 16 bit µController interface using Byte Select signals:

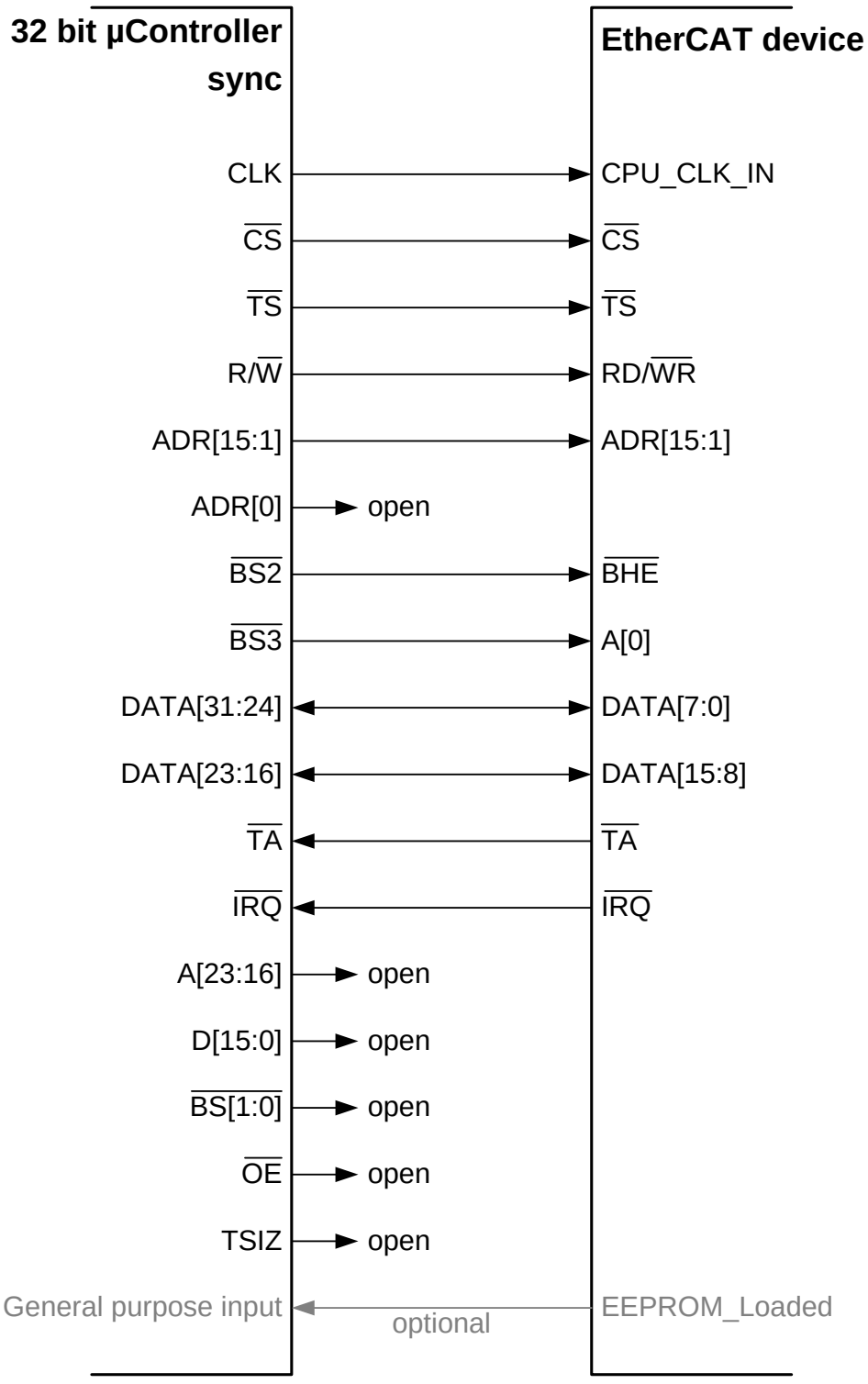

Figure 39: Synchronous 32 bit µController connection using Byte Select

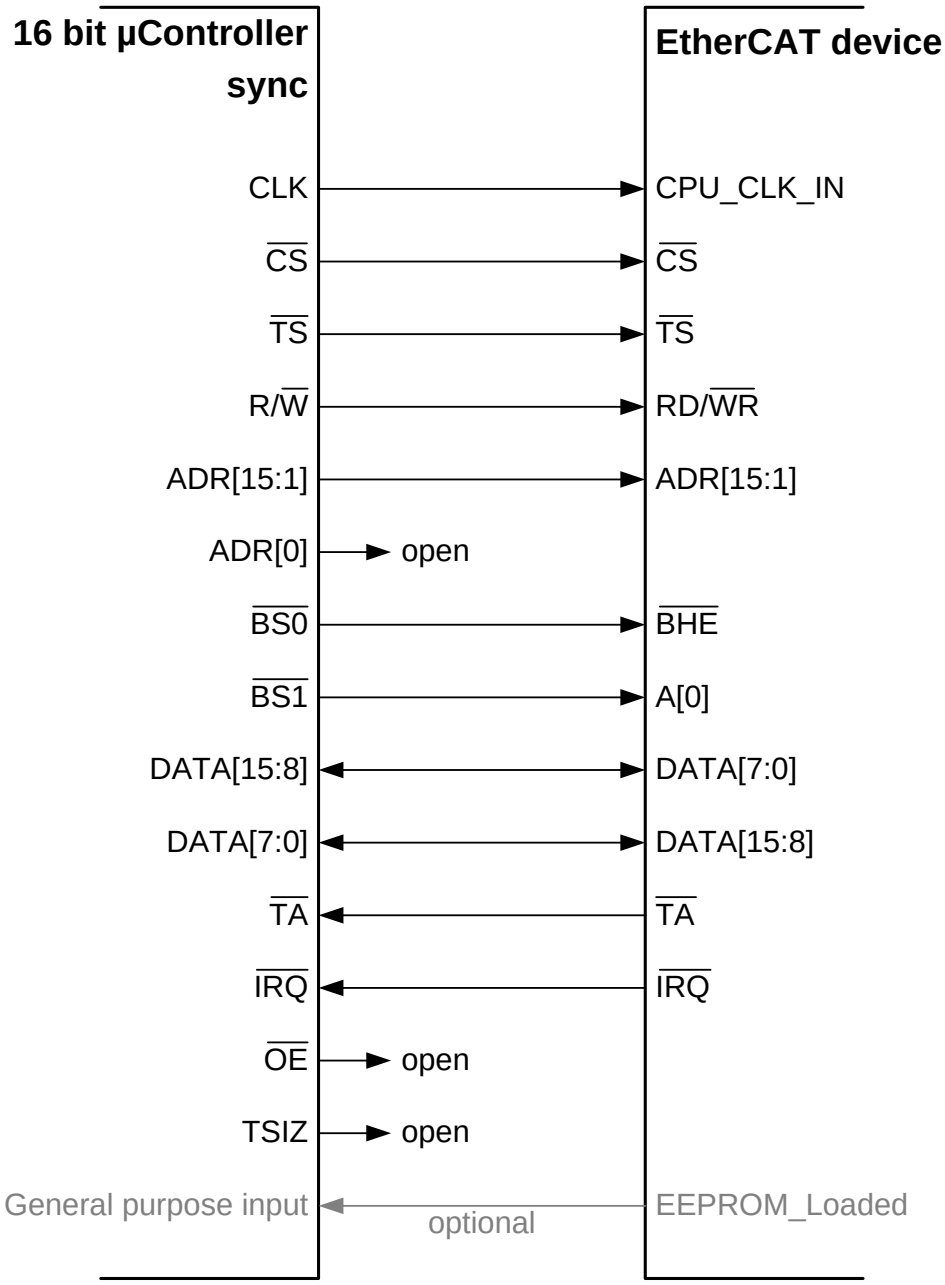

Figure 40: Synchronous 16 bit µController connection using Byte Select

6.5.5 µController connection using Transfer Size signals (SIZ)

In case the µController does not provide Byte High Enable, and Transfer Size signals (SIZ or TSIZ) are available, they can be used to distinguish between 8 and 16 bit accesses together with ADR[0]. An exclusive-or combination of ADR[0] and SIZ[0] is equivalent to BHE. This combination can be configured with Byte access mode set to one (Transfer Size mode).

Table 77: Byte Select vs. ADR[0] and BHE

µController			EtherCAT device		Access
ADR[0]	**SIZ[1:0]**	**ADR[0] xor SIZ[0]**	**ADR[0]**	**BHE (act. low)**	
0	10	0	0	0	16 bit access (low and high byte)
0	01	1	0	1	8 bit access (low byte, even address)
1	01	0	1	0	8 bit access (high byte, odd address)
0	00	0	0	0	32 bit access (splitted in two 16 bit accesses)

The following figure shows how a 32 bit µController can be connected with the EtherCAT synchronous 16 bit µController interface using SIZ signals:

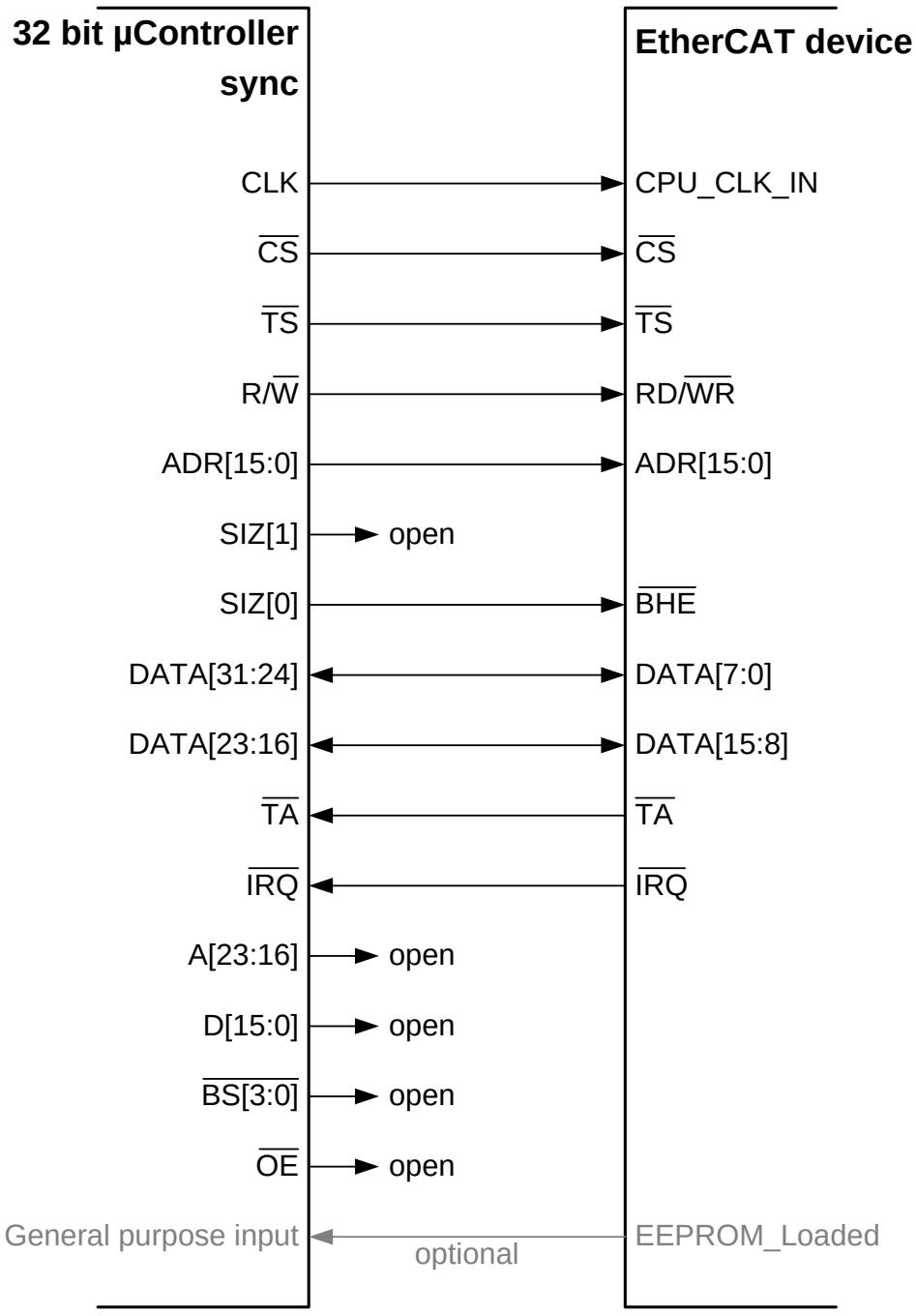

Figure 41: Synchronous 32 bit µController connection using Transfer Size

6.5.6 Write access

A write access starts with a Transfer Start (TS). Chip Select can be either together with TS or one clock cycle later (does not need to be configured). The CPU_CLK_IN edge at which CS is sampled can be configured. ADR, BHE and R/nW are valid together with TS. It is configurable if write DATA is also valid with CS or one cycle later. Once the EtherCAT device has finished the access, Transfer Acknowledge is asserted for one clock cycle. It may either be generated with the rising or falling edge of CPU_CLK_IN.

6.5.7 Read access

A read access starts with a Transfer Start (TS). Chip Select can be either together with TS or one clock cycle later (does not need to be configured). The CPU_CLK_IN edge at which CS is sampled can be configured. ADR, BHE and R/nW are valid together with TS. Once the EtherCAT device has finished the access, Transfer Acknowledge is asserted for one clock cycle together with the read DATA. TA may either be generated with the rising or falling edge of CPU_CLK_IN.

Some µControllers expect a read access always to be a 16 bit read access, regardless of the Byte Select signals. For this reason, it is configurable that the Byte Select signals are ignored and a read access is always a 16 bit access.

6.5.8 µController access errors

One reason for µController access errors is detected by the synchronous µController interface:

- Read or Write access to the 16 bit interface with A[0]=1 and BHE(act. low)=1, i.e. an access to an odd address without Byte High Enable.

Such a wrong µController access will have these consequences:

- The PDI error counter 0x030D will be incremented.
- No access will be performed internally.

6.5.9 EEPROM_LOADED

The EEPROM_LOADED signal indicates that the µController Interface is operational. Attach a pull-down resistor for proper function, since the PDI pin will not be driven until the EEPROM is loaded. EEPROM_LOADED is synchronous to CPU_CLK_IN, it will not go high if CPU_CLK_IN is not toggling.

6.5.10 Timing Specification

Table 78: µController timing characteristics ET1100

Parameter	Min	Max	Comment
t_{CLK}	25 ns		CPU_CLK_IN period ($f_{CLK} \leq 40$ MHz)
t_{IN_setup}	10 ns		Input signals valid before CPU_CLK_IN edge (TS, CS, ADR, BHE, R/nW, DATA)
t_{IN_hold}	3 ns		Input signals valid after CPU_CLK_IN edge (TS, CS, ADR, BHE, R/nW, DATA)
$t_{CLK\_to\_OUT\_valid}$		15 ns	Output signals valid after CPU_CLK_IN edge (TA, IRQ, DATA)
$t_{CLK\_to\_OUT\_invald}$	0 ns		Output signals invalid after CPU_CLK_IN edge (TA, IRQ, DATA)
t_{read}		a) t_{read_int} + t_{prec_write} +t_{Coll} b) t_{read_int} +t_{read_sync}	External read time (TS to TA) a) with preceeding write access and $t_{WR\_to\_RD} + t_{read\_sync} < t_{prec\_write} + t_{Coll}$ b) without preceeding write access or $t_{WR\_to\_RD} + t_{read\_sync} > t_{prec\_write} + t_{Coll}$
t_{read_sync}		2.5*t_{CLK} a) +0.5*t_{CLK} b) +0.5*t_{CLK} c) +t_{CLK}	Extra read synchronization delay a) extra delay if 0x0152.11=1 b) extra delay if 0x0152.10=1 b) extra delay if CS asserted one CPU_CLK_IN cycle after TS
t_{read_int}		a) 235 ns b) 315 ns	Internal read time a) 8 bit access b) 16 bit access
t_{prec_write}		a) 180 ns b) 260 ns	Time for preceding write access a) 8 bit access b) 16 bit access
t_{Coll}		a) 20 ns b) 0 ns	Extra read delay a) RD access directly follows WR access with the same address (8 bit accesses or 8 bit WR and 16 bit RD) b) different addresses or 16 bit accesses
t_{write}		a) t_{write_int} b) t_{write_sync}	External write time (TS to TA) a) with preceeding write access and $t_{WR\_delay}+t_{write\_sync} < t_{write\_int}$ b) without preceeding write access or $t_{WR\_delay}+t_{write\_sync} \geq t_{write\_int}$
t_{write_sync}		2.5*t_{CLK} a) +t_{CLK} b) +0.5*t_{CLK} c) +0.5*t_{CLK} d) +t_{CLK}	Extra write synchronization delay a) extra delay if 0x0152.8=0 b) extra delay if 0x0152.10=1 c) extra delay if 0x0152.11=1 d) extra delay if CS asserted one CPU_CLK_IN cycle after TS
t_{write_int}		a) 200 ns b) 280 ns	Internal write time a) 8 bit access b) 16 bit access
$t_{write+read}$		$t_{write}+t_{read}$	Internal write/read time for a read access following a write access
$t_{EEPROM\_LOADED\_to\_access}$	0 ns		Time between EEPROM_LOADED and first access

Parameter	Min	Max	Comment
$t_{EEPROM\_LOADED\_to\_IRQ}$		0 ns	IRQ valid after EEPROM_LOADED

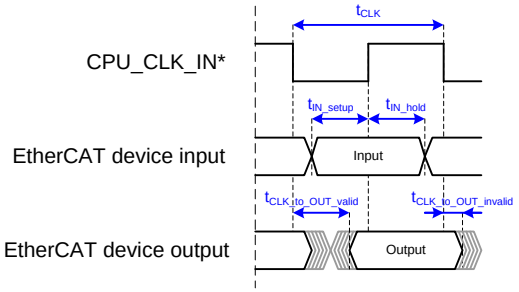

Figure 42: Basic synchronous µController interface timing (*refer to timing diagram for relevant CPU_CLK_IN edges)

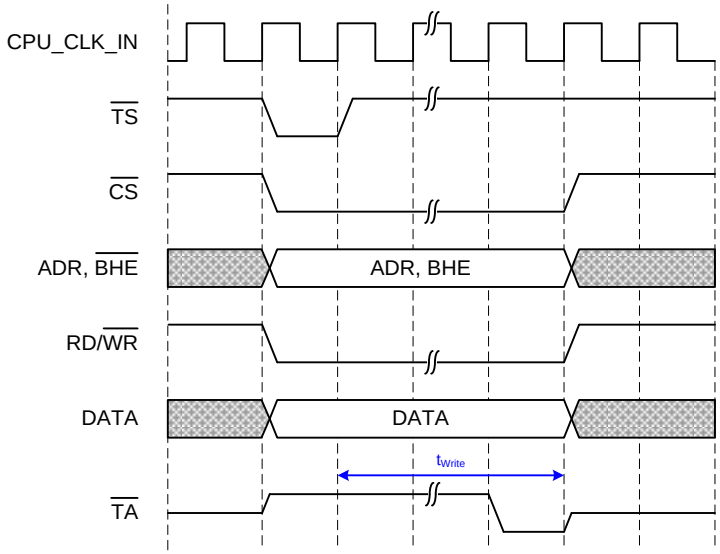

Figure 43: Write access (CS together with TS, Write DATA together with CS, CS and TA on rising edge)

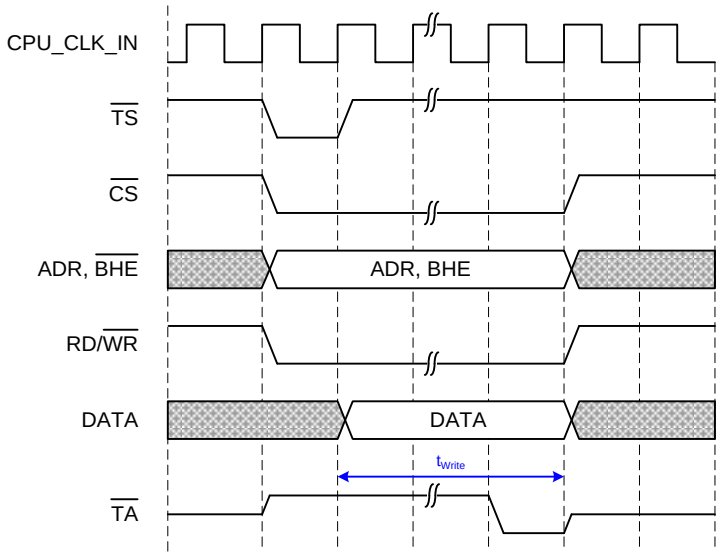

Figure 44: Write access (CS together with TS, Write DATA after CS, CS and TA on rising edge)

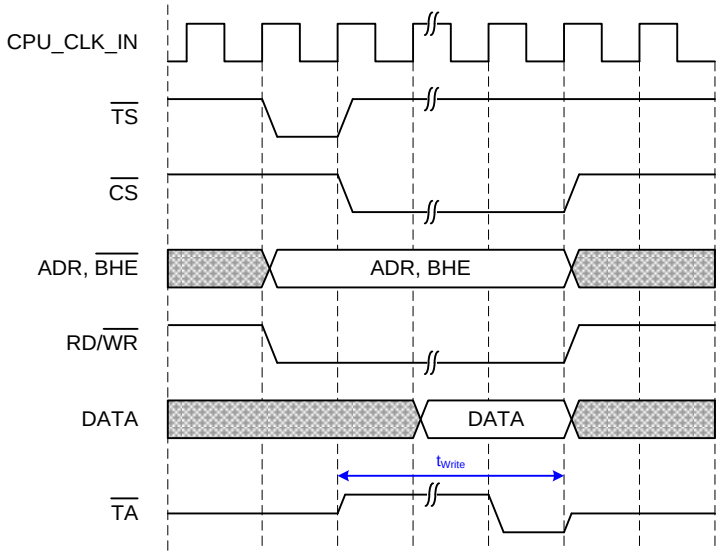

Figure 45: Write access (CS after TS, Write DATA after CS, CS and TA on rising edge)

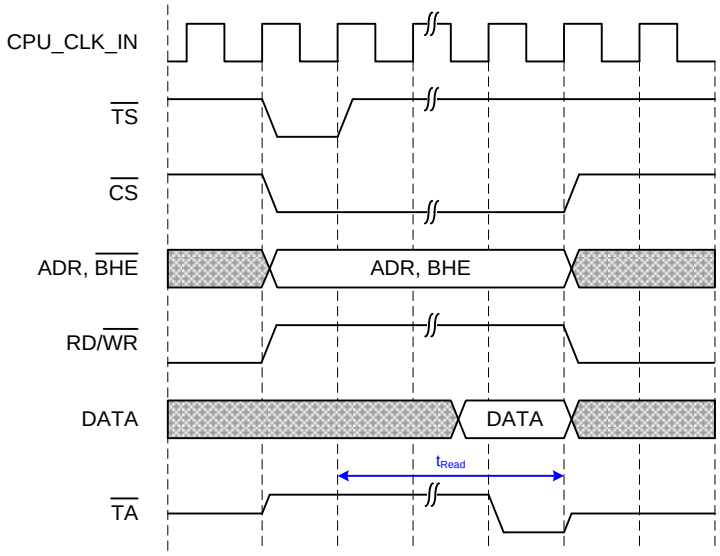

Figure 46: Read access (CS together with TS, CS and TA on rising edge)

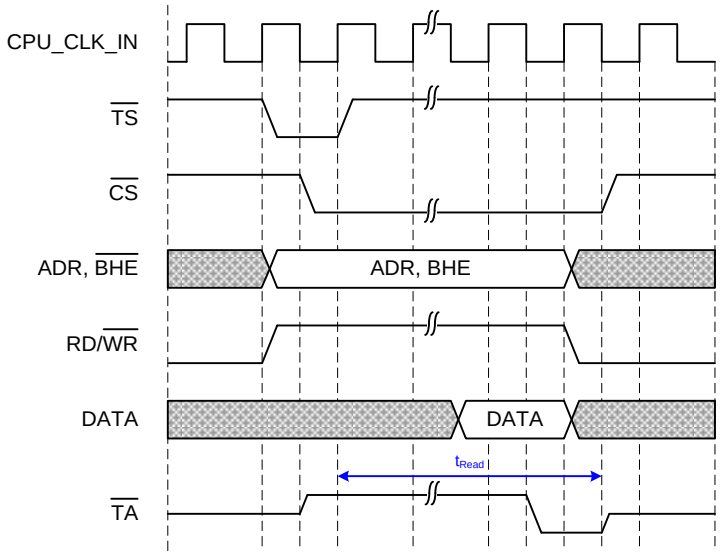

Figure 47: Read access (CS half a clock period after TS, CS and TA on falling edge)

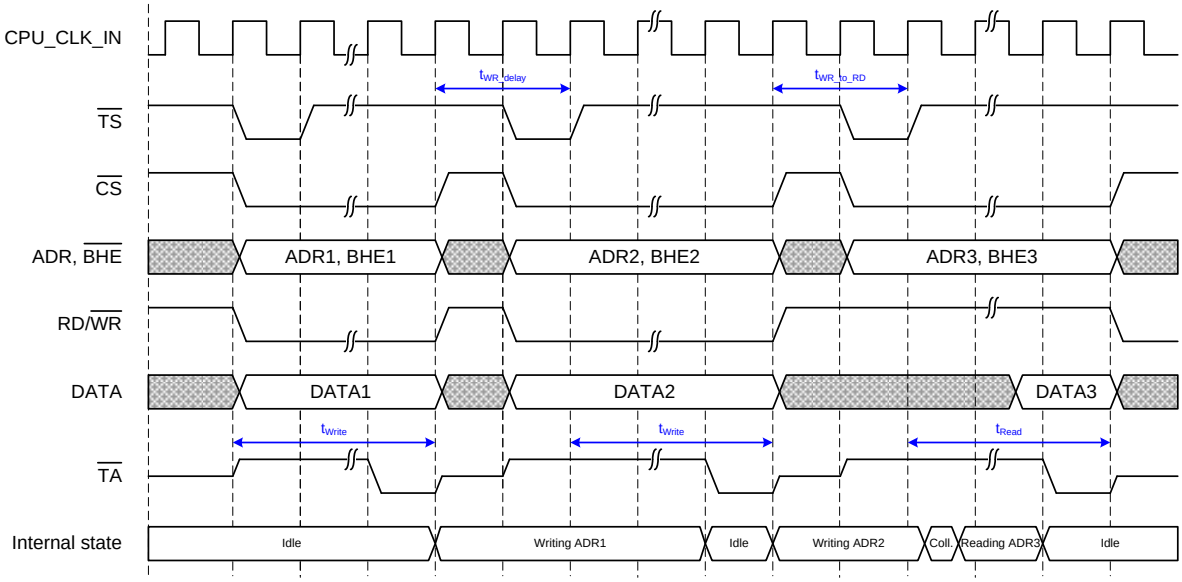

Figure 48: Sequence of two write accesses and a read access

Note: The first write access to ADR1 is performed after the first TA. After that, the ESC is internally busy writing to ADR1. After CS is de-asserted, TA is not driven any more, nevertheless, the ESC is still writing to ADR1.

Hence, the second write access to ADR2 is delayed because the write access to ADR1 has to be completed first. After the second TA, the ESC is busy writing to ADR2.

The third access in this example is a read access. The ESC is still busy writing to ADR2 while the read access begins. In this case, the write access to ADR2 is finished first, and afterwards, the read access to ADR3 is performed. The ESC signals TA after both write and read access have finished.

7 Distributed Clocks SYNC/LATCH Signals

For details about the Distributed Clocks refer to Section I.

7.1 Signals

The Distributed Clocks unit of the ET1100 has the following external signals:

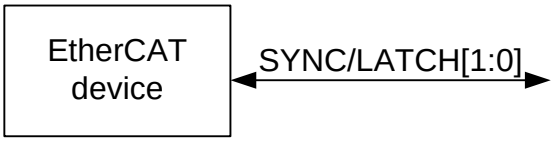

Figure 49: Distributed Clocks signals

Table 79: Distributed Clocks signals

Signal	Direction	Description
SYNC/LATCH[1:0]	OUT/IN	SyncSignal (OUT) or LatchSignal (IN), direction bitwise configurable via register 0x0151 / EEPROM.

NOTE: SYNC/LATCH signals are not driven (high impedance) until the SII EEPROM is loaded.

7.2 Timing specifications

Table 80: DC SYNC/LATCH timing characteristics ET1100

Parameter	Min	Max	Comment
t_{DC_LATCH}	15 ns		Time between Latch0/1 events
$t_{DC\_SYNC\_Jitter}$		15 ns	SYNC0/1 output jitter
$t_{DC\_SYNC\_Pulse\_IRQ}$	40 ns		Pulse length for SYNC0/1 if used as PDI interrupt in continuous mode

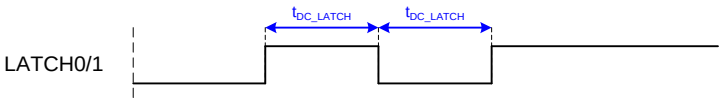

Figure 50: LatchSignal timing

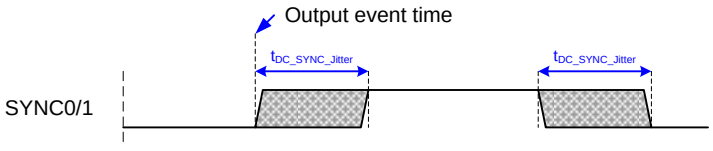

Figure 51: SyncSignal timing

8 SII EEPROM Interface (I²C)

For details about the ESC SII EEPROM Interface refer to Section I. The SII EEPROM Interface is intended to be a point-to-point interface between ET1100 and I²C EEPROM. If other I²C masters are required to access the I²C bus, the ET1100 must be held in reset state (e.g. for in-circuit-programming of the EEPROM), otherwise access collisions will be detected by the ET1100.

8.1 Signals

The EEPROM interface of the ET1100 has the following signals:

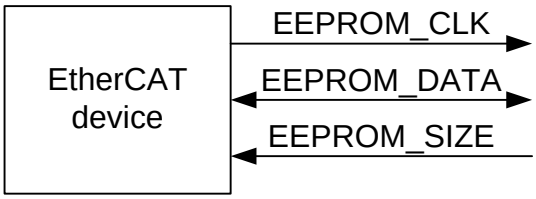

Figure 52: I²C EEPROM signals

Table 81: I²C EEPROM signals

Signal	Direction	Description
EEPROM_CLK	OUT	I²C clock
EEPROM_DATA	BIDIR	I²C data
EEPROM_SIZE	IN	EEPROM size configuration

The pull-up resistors for EEPROM_CLK and EEPROM_DATA are integrated into the ET1100. EEPROM_CLK must not be held low externally, because the ET1100 will detect this as an error.

8.2 Timing specifications

Table 82: SII EEPROM timing characteristics

Parameter	Typical		Comment
	1 Kbit-16 Kbit	32 Kbit-4 Mbit	
t_{Clk}	~ 6.72 µs		EEPROM clock period (f_{Clk} ≈ 150 kHz)
t_{Write}	~ 250 us	~ 310 µs	Write access time (without errors)
t_{Read}	a) ~ 680 µs b) ~ 1.16 ms	a) ~ 740 µs b) ~ 1.22 ms	Read access time (without errors): a) 4 words b) configuration (8 Words)
t_{Delay}	~ 168 ms		Time until configuration loading begins after Reset is gone

9 Electrical and Mechanical Specifications

9.1 Absolute Maximum Conditions

Table 83: Absolute Maximum Conditions

Symbol	Parameter	Condition	Min	Max	Units
$V_{CC\ I/O}$-V_{SS}	Supply voltage for internal LDO		-0.3	5.5	V
$I_{CC\ I/O}$	Supply current	Internal LDO for $V_{CC\ Core}$ used a) $V_{CC\ I/O}$=3.3V b) $V_{CC\ I/O}$=5V		a) 170 b) 220	mA
$I_{CC\ Core}$	Supply current	$V_{CC\ Core}$ sourced externally		150	mA
V_{ESC}	ESD protection	Human body model, according to MIL-STD-883E-3015.7 Class 1	2		kV
I_{DC_ESD}	Permanent current into ESD protection diodes	Only in case of forward biased ESD diodes. Input voltage above $V_{CCI/O}$ or below V_{SS}		2	mA

NOTE: Supply current does not include output driver current for PDIs and LEDs.

9.2 Operating Conditions

9.2.1 Power Supply

Table 84: Power Supply

Symbol	Parameter	Condition	Min	Typ	Max	Units
$V_{CC\ I/O}$	I/O power supply		3.0	3.3	5.5	V
$V_{CC\ Core}$	Logic power supply		2.25	2.5	2.75	V
$V_{CC\ PLL}$	PLL power supply		2.25	2.5	2.75	V
$V_{CC\ Core\ Ext}$	External logic power supply		2.5	2.5	2.75	V
$V_{CC\ PLL\ Ext}$	External PLL power supply		2.5	2.5	2.75	V

9.2.2 Electrical Characteristics

Table 85: DC Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
$V_{CC\ Core\ LDO}$	Internal LDO output voltage $V_{CC\ Core}$/$V_{CC\ PLL}$			2.4		V
$V_{Reset\ I/O}$	Reset threshold for $V_{CC\ I/O}$			2.8		V
$V_{Reset\ Core}$	Reset threshold for $V_{CC\ Core}$			1.6		V
V_{IL}	Input Low voltage (not OSC_IN)				0.7	V
V_{IH}	Input High voltage (not OSC_IN)	a) $V_{CC\ I/O}$=3.3V b) $V_{CC\ I/O}$=5V	2.0		a) 3.6 b) 5.5	V
$V_{IT\ OSC\_IN}$	Input threshold voltage OSC_IN (no Schmitt trigger)	a) $V_{CC\ I/O}$=3.3V b) $V_{CC\ I/O}$=5V	a) 1.4 b) 2.2	a) 1.6 b) 2.5	a) 1.8 b) 2.8	V
V_{OL}	Output Low voltage				0.4	V
V_{OH}	Output High voltage		2.4			V
V_{OD}	LVDS differential output voltage	R_L=100 Ω R_{BIAS}=11 kΩ	245	350	455	mV
ΔV_{OD}	Change in V_{OD} between 1 and 0				±50	mV
V_{OC}	LVDS common mode output voltage		1.125	1.25	1.375	V
ΔV_{OC}	Change in V_{OC} between 1 and 0				±50	mV
V_{ID}	LVDS differential input voltage		100			mV
V_{IC}	LVDS input voltage range		0		2.4	V
I_{OH}	Output High current				4	mA
I_{OL}	Output Low current				-3	mA
I_{IL}	Input leakage current (without internal pull-up/pull-down resistors)				±10	µA
I_{OL}	Output leakage current (tristate, without internal PU/PD)				±10	µA
R_{PU}	Internal pull-up resistor		1.6	3.3	7	kΩ
R_{WPU}	Weak internal pull-up resistor	a) $V_{CC\ I/O}$=3.3V b) $V_{CC\ I/O}$=5V	a) 75 b) 50	a) 110 b) 70	a) 190 b) 120	kΩ
R_{WPD}	Weak internal pull-down resistor	a) $V_{CC\ I/O}$=3.3V b) $V_{CC\ I/O}$=5V	a) 60 b) 40	a) 95 b) 60	a) 180 b) 110	kΩ
R_{LI+}	Internal LVDS input pull-down resistor at EBUS_RX+ pins		15	27	45	kΩ
R_{LI-}	Internal LVDS input pull-up resistor at EBUS_RX- pins		15	27	45	kΩ
R_{BIAS}	External LVDS BIAS resistor			11		kΩ
R_L	LVDS RX load resistor			100		Ω
C_{OSC}	OSC_IN/OSC_OUT pin capacitance			1.2		pF

NOTE: R_{WPU}/R_{WPD} cannot be used externally, their full effectiveness appears only inside the ET1100 (realized as transistors).

NOTE: Input and output characteristics without special indication apply to all non-LVDS I/O signals.

Table 86: DC Characteristics (Supply Current – Internal LDO used)

Symbol	Parameter	Condition	Typ	Units
$I_{CC\ I/O}$	Supply current examples: a) 2xMII, 1xFMMU, DC off b) 2xMII, 1xFMMU, DC S+L c) 4xMII, 8xFMMU, DC S+L d) 2xEBUS, 1xFMMU, DC off e) 2xEBUS, 1xFMMU, DC S+L f) 4xEBUS, 8xFMMU, DC S+L	$V_{CC\ I/O}$=3.3V, Internal LDO used	 a) 49 b) 63 c) 81 d) 83 e) 98 f) 149	mA
$I_{CC\ I/O}$	Supply current examples: a) 2xEBUS, 1xFMMU, DC off b) 2xEBUS, 1xFMMU, DC S+L c) 4xEBUS, 8xFMMU, DC S+L	$V_{CC\ I/O}$=5V, Internal LDO used	 a) 102 b) 117 c) 177	mA
$I_{CC\ I/O\ Base}$	Supply current calculation base	$V_{CC\ I/O}$=3.3V, Internal LDO used	32	mA
$I_{CC\ EBUS}$	Supply current add-on to $I_{CC\ I/O\ Base}$ per EBUS port		24	mA
$I_{CC\ MII}$	Supply current add-on to $I_{CC\ I/O\ Base}$ per MII port		7	mA
$I_{CC\ DC\ Cyclic}$	Supply current add-on to $I_{CC\ I/O\ Base}$ if DC Latch or Sync enabled		5	mA
$I_{CC\ DC\ Latch}$	Supply current add-on to $I_{CC\ I/O\ Base}$ if DC Latch unit enabled		4	mA
$I_{CC\ DC\ Sync}$	Supply current add-on to $I_{CC\ I/O\ Base}$ if DC Sync unit enabled		6	mA
$I_{CC\ FMMU}$	Supply current add-on to $I_{CC\ I/O\ Base}$ per FMMU		0.5	mA
$I_{CC\ Digital}$	Supply current add-on to $I_{CC\ I/O\ Base}$ if Digital I/O PDI is selected		2	mA
$I_{CC\ SPI}$	Supply current add-on to $I_{CC\ I/O\ Base}$ if SPI PDI is selected		5	mA
I_{CC_uC}	Supply current add-on to $I_{CC\ I/O\ Base}$ if µController PDI is selected		5	mA

NOTE: Supply current does not include output driver current for PDIs and LEDs.

Table 87: DC Characteristics (Supply Current – $V_{CC\ Core}$ sourced external)

Symbol	Parameter	Condition	Typ	Units
$I_{CC\ Core}$	Supply current examples (Digital I/O): a) 2xMII, 1xFMMU, DC off b) 2xMII, 1xFMMU, DC S+L c) 4xMII, 8xFMMU, DC S+L d) 2xEBUS, 1xFMMU, DC off e) 2xEBUS, 1xFMMU, DC S+L f) 4xEBUS, 8xFMMU, DC S+L	$V_{CC\ I/O}$=3.3V, $V_{CC\ Core}$=2.5V	 a) 42 b) 58 c) 70 d) 63 e) 79 f) 117	mA
$I_{CC\ I/O}$	Supply current examples (Digital I/O): a) 2xMII, 1xFMMU, DC off b) 2xMII, 1xFMMU, DC S+L c) 4xMII, 8xFMMU, DC S+L d) 2xEBUS, 1xFMMU, DC off e) 2xEBUS, 1xFMMU, DC S+L f) 4xEBUS, 8xFMMU, DC S+L	$V_{CC\ I/O}$=3.3V, $V_{CC\ Core}$=2.5V	 a) 14 b) 14 c) 14 d) 23 e) 23 f) 39	mA

NOTE: Supply current does not include output driver current for PDIs and LEDs.

9.2.3 Timing Characteristics

Table 88: Timing Characteristics

Symbol	Parameter	Min	Typ	Max	Units
f_{CLK25}	Clock source (OSC_IN) with initial accuracy		25 MHz ± 25 ppm		
$t_{CLK25OUT1}$	CLK25OUT1 rising edge after OSC_IN rising edge		5		ns
$t_{CLK25OUT2}$	CLK25OUT2 rising edge after OSC_IN rising edge		7		ns
t_{TX_delay}	TX_ENA/TX_D[3:0] edge (TX-Shift = 00) after rising edge of a) OSC_IN b) CLK25OUT1 c) CLK25OUT2		a) 5 b) 0 c) 38		ns
t_{CPU_CLK}	CPU_CLK (25 MHz) rising edge after OSC_IN rising edge		5		ns
t_{POR_Sample}	POR value sample time after power good		84		ms
t_{Driver_Enable}	Output drivers enabled after POR values sampled (not PDI and not Sync/LatchSignals)		80		ns
t_{Reset_In}	External reset input time	50			ns
t_{Reset_Out}	ET1100 Reset output time	80	84		ms
t_{Reset_Func}	ET1100 functional after RESET signal high (EEPROM not loaded, PDI not functional)			50	µs
$t_{Startup}$	Startup time (PDI operational after power good, without SII loading error)			340	ms

The timing characteristics of the PDIs, Distributed Clocks, EEPROM I^2C interface, and MII interface can be found in their respective chapters.

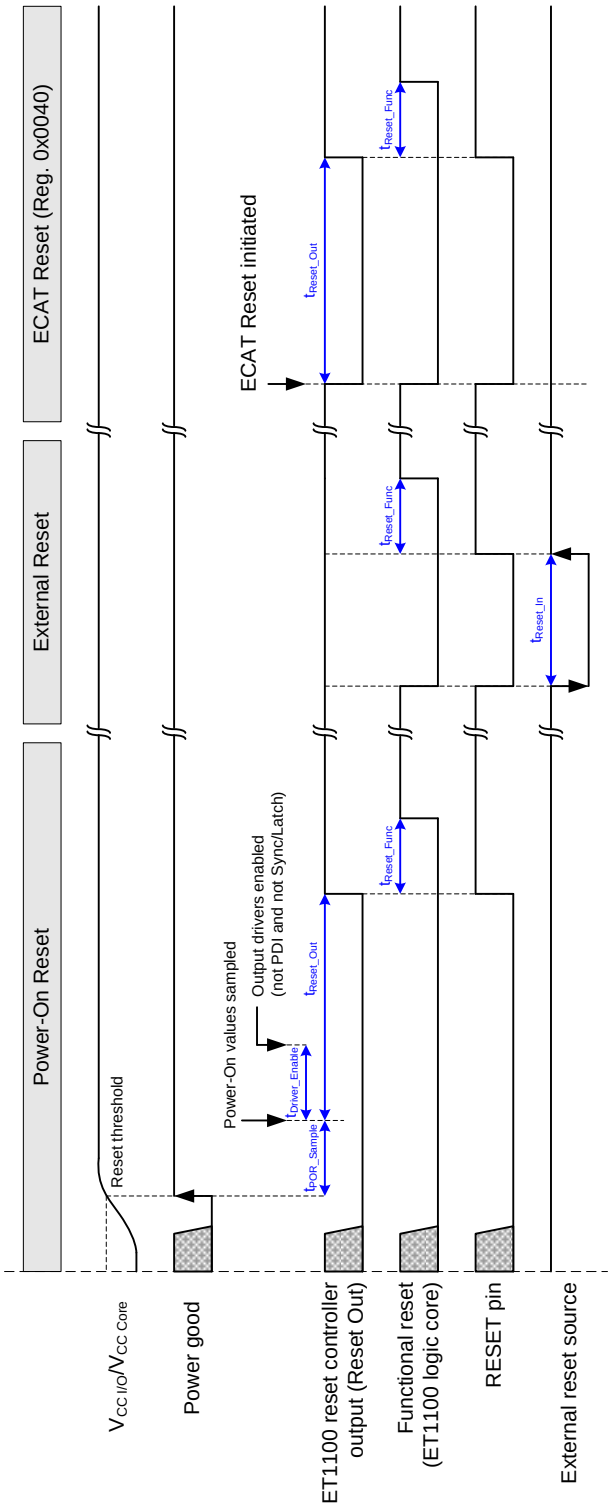

Figure 53: Reset Timing

NOTE: External clock source (quartz oscillator) is assumed to be operational at Power-good time. Otherwise t_{POR_Sampe} is delayed.

Table 89: Forwarding Delays

Symbol	Parameter	Min	Average	Max	Units
t_{Diff}	Average difference processing delay minus forwarding delay (without RX FIFO jitter) between any two ports a) at least one of the two ports is EBUS b) both ports are MII		a) 20 b) 40		ns
t_{EE}	EBUS port to EBUS port delay (FIFO size 7): a) Through ECAT Processing Unit (processing), Low Jitter off b) Alongside ECAT Processing Unit (forwarding), Low Jitter off c) Through ECAT Processing Unit (processing), Low Jitter on d) Alongside ECAT Processing Unit (forwarding), Low Jitter on	a) 140 b) 120 c) 150 d) 130	a) 150 b) 130 c) 155 d) 135	a) 160 b) 140 c) 160 d) 140	ns
t_{EM}	EBUS port to MII port delay (FIFO size 7, TX Shift=00): a) Through ECAT Processing Unit (processing), Low Jitter off b) Alongside ECAT Processing Unit (forwarding), Low Jitter off	a) 145 b) 125	a) 170 b) 150	a) 195 b) 175	ns
t_{ME}	MII port to EBUS port delay (FIFO size 7, TX Shift=00): a) Through ECAT Processing Unit (processing), Low Jitter off b) Alongside ECAT Processing Unit (forwarding), Low Jitter off c) Through ECAT Processing Unit (processing), Low Jitter on d) Alongside ECAT Processing Unit (forwarding), Low Jitter on	a) 255 b) 235 c) 265 d) 245	a) 280 b) 260 c) 290 d) 270	a) 305 b) 285 c) 315 d) 295	ns
t_{MM}	MII port to MII port delay (FIFO size 7, TX Shift=00): a) Through ECAT Processing Unit (processing), Low Jitter off b) Alongside ECAT Processing Unit (forwarding), Low Jitter off	a) 280 b) 240	a) 305 b) 265	a) 335 b) 295	ns

NOTE: Average timings are used for Distributed Clocks calculations.

9.2.4 Thermal Characteristics

Table 90: Thermal Characteristics

Symbol	Parameter	Min	Typ	Max	Units
ϑ_A	Ambient temperature	-40		85	°C
ϑ_J	Junction temperature	-40		125	°C
Θ_{JA}	Thermal resistance Theta junction to ambient		40		°C/W
Θ_{JC}	Thermal resistance Theta junction to case		9.8		°C/W
Θ_{JB}	Thermal resistance Theta junction to case		22.1-23.7		°C/W
Ψ_{JT}	Thermal resistance PSI junction to top		0.2		°C/W
Ψ_{JB}	Thermal resistance PSI junction to bottom		20.2		°C/W

Note: Modeling test board (PCB) JEDEC 2s2p

9.3 Mechanical Specifications

9.3.1 Package Information

A 10mm x 10mm TFBGA (Thin-profile Fine-pitch BGA) with 128 balls is used for the ET1100. The material of the balls is 95.5% Sn / 4% Ag / 0.5% Cu (SAC405).

The ET1100 is compliant to RoHS 2 (2011/65/EU) including amendment "COMMISSION DELEGATED DIRECTIVE (EU) 2015/863".

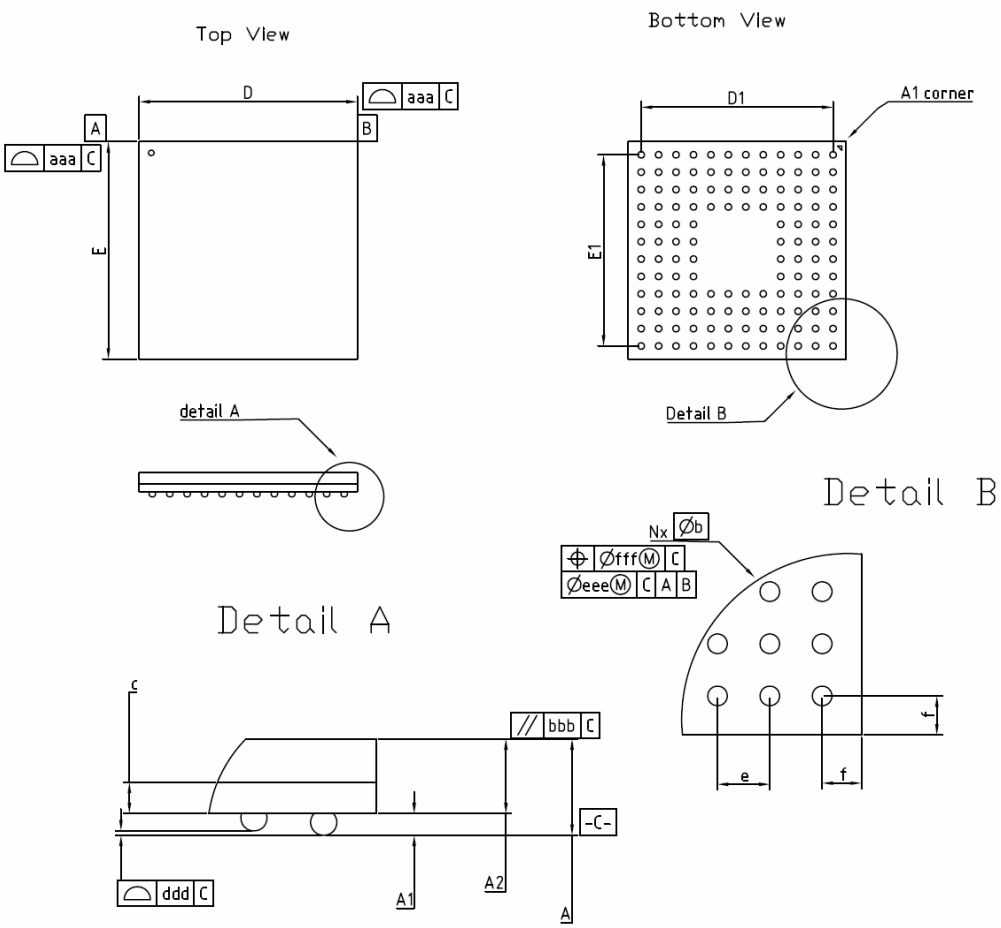

Figure 54: Package Outline

Table 91: Package Dimensions

Dimensional Ref.			
REF.	Min.	Nom	Max.
A			1.2
A1	0.2		
A2		0.82	
D		10.0	
D1		8.8	
E		10.0	
E1		8.8	
b		0.3	
c	0.28	0.32	0.36
e		0.8	
f		0.6	
m		12	
n		128	

Dimensional Tol.	
aaa	0.15
bbb	0.10
ddd	0.08
eee	0.15
fff	0.08

Notes
1. All dimensions in MM
2. 'e' represents the basic solder ball pitch
3. 'm' represents the basic solder ball matrix size. And 'n' is the number of attached solder balls
4. 'b' is measurable at the maximum solder ball diameter parallel the the primary datum -C-
5. Dimension 'aaa' is measured parallel to primary datum -C-
6. Primary datum -C- and the seating plane are defined by the spherical crowns of the solder balls
7. The package surface shall be matte finish charmilles 24 to 27
8. The over package thickness 'A' already considers collapse balls
9. Reference Jedec MO-205

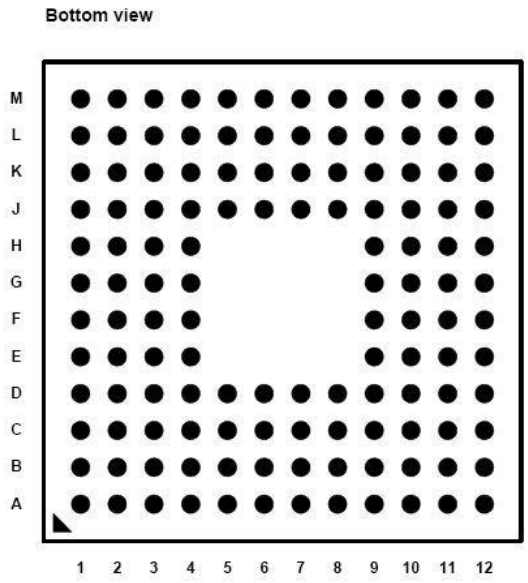

Figure 55: TFBGA 128 Pin Layout

The chip label contains the date code (X=stepping, YY=year, WW=week, optional: LLL…= lot ID).

Figure 56: Chip Label

9.3.2 Tape and Reel Information

The ET1100 is optionally available as tape on reel.

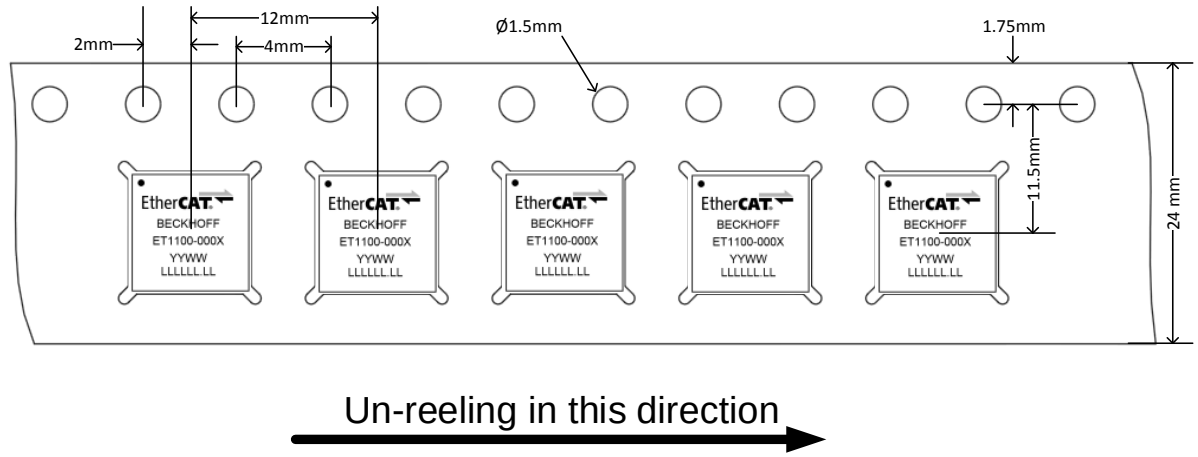

Figure 57: ET100 Tape Information

The reel is a combination of Advantek® LOKREEL® RD33008SW and RD33016SW:

Table 92: ET1100 Reel Information

Dimension	Value
Diameter	330 mm
Total width (8+16 mm)	24 mm

9.3.3 Moisture Sensitivity and Storage

The ET1100 is shipped in a sealed moisture barrier bag (dry-pack). There is a “caution” label on the dry-pack which contains all necessary information required for handling the devices. Refer to the JEDEC standards J-STD-020 and J-STD-033 for more details (http://www.jedec.org).

The information on the dry-pack takes precedence over information in this chapter.

The moisture sensitivity level of the ET1100 is MSL 3. The maximum shelf-life of the ET1100 packed in a dry-pack is one year after bag seal date. If the ET1100 is stored longer than one year, drying (baking) is required before soldering.

Drying and re-packaging can have negative effects on solderability and conducting surfaces. To minimize issues, the following steps should be taken:

- Visual inspection of the ET1100 devices
- solderability tests with some samples of the ET1100
- final test of the product using the ET1100 with focus on the ET1100 connections

Table 93: Absolute Maximum Storage Conditions

Symbol	Parameter	Min	Max	Units
$\vartheta_{Storage}$	Storage temperature	-65	150	°C

9.4 Processing

9.4.1 PCB Recommendations

PCB manufacturing technology is complex, please consult your PCB manufacturer and your PCB assembly house for advice. A few recommendations for many use cases are given here.

The pinout of the ET1100 is optimized for easy escape routing using 0.7mm/0.3mm vias inside the free center of the BGA, because the inner two ball rings are mainly used for power supply.

Non-solder mask defined pads (NSMD) with a conductive pad diameter of 300 µm and an actual solder mask opening diameter of 400 µm (after widening) are recommended. Thus, the solder ball covers the whole pad, resulting in a better connection.

Each pad (whether used or unused) should only be connected by a single trace, and the trace width should be small and identical for all pads, e.g. 125 µm. This results in an equal soldering behavior of the balls.

PCBs without plating (copper) are not recommended, because of brittle solder joints.

9.4.2 Soldering Profile

The following soldering profile is used to illustrate minimum and maximum values. For the actual soldering profile many factors have to be taken into consideration, e.g., solder paste characteristics, the PCB, plating, other components, materials, and process type.

Please consult your PCB assembly house for advice.

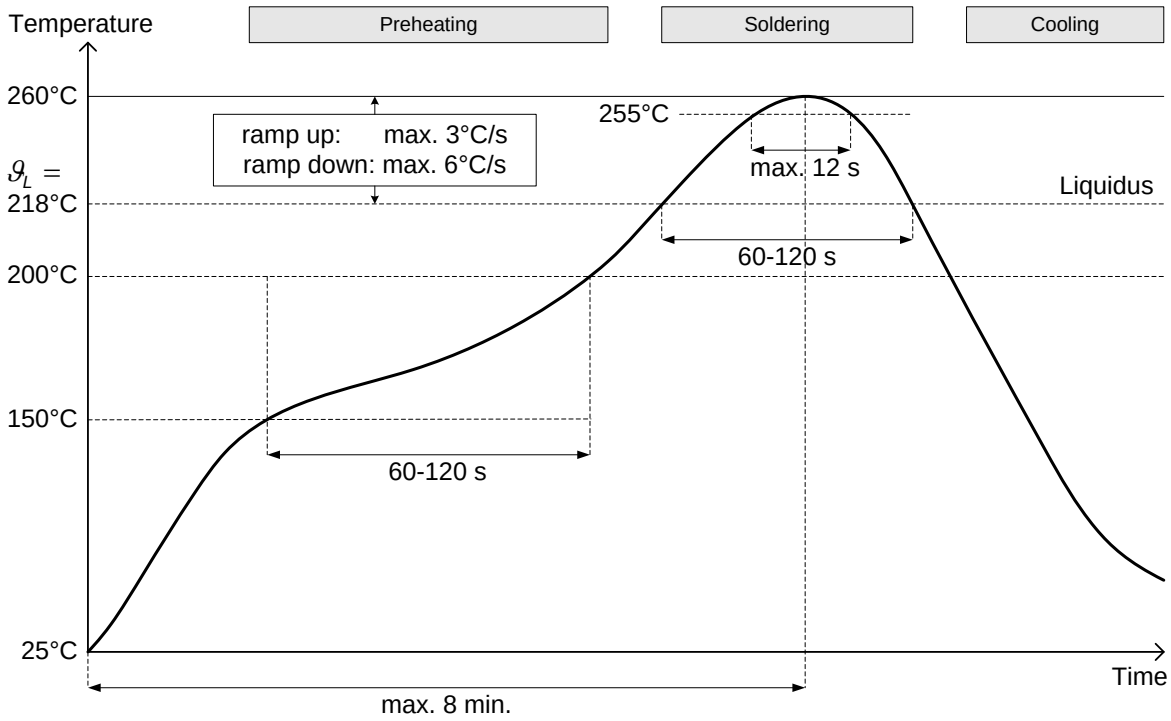

Figure 58: Soldering temperature and time

Table 94: Soldering temperature and time

Symbol	Parameter	Value	Abs. Max.	Units
ϑ_L	Liquidus temperature	218		°C
t_L	Time above ϑ_L (TAL)		120	s
ϑ_P	Peak temperature		260	°C
t_P	Time at ϑ_P		12	s
N_R	Number of reflow cycles		3	

NOTE: Recommended reading: “First Principles of Solder Reflow” by John Vivari.

10 Ordering codes

The ordering codes for the ET1100 devices are composed like this:

ET1100-0000-NNNN

The code part NNNN identifies the size of the packing unit. Do not confuse the ordering codes with the stepping code ET1100-0000. You will always get the latest stepping while the ordering codes are unchanged.

11 Appendix

11.1 Support and Service

Beckhoff and our partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

11.1.1 Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: http://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

11.2 Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG
Huelshorstweg 20
33415 Verl
Germany

Phone: +49 (0) 5246 963-0

Fax: +49 (0) 5246 963-198

E-mail: info@beckhoff.com

Web: www.beckhoff.com

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- world-wide support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline: +49 (0) 5246 963-157

Fax: +49 (0) 5246 963-9157

E-mail: support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline: +49 (0) 5246 963-460

Fax: +49 (0) 5246 963-479

E-mail: service@beckhoff.com