YASKAWA

ANTAIOS
Datasheet

ANT1000/1001 | Revision 1.23

1 Overview

The ANTAIOS is a multi-fieldbus communication chip with dedicated hardware support for (isochronous) real-time Ethernet based protocols.

Based on the approved concept of the SMC1000 chip, which already provides a rich set of communication interfaces, the architecture of the ANTAIOS has been extended by several key features:

- ARM® Cortex®-A5 Host CPU, 32/32 KB Caches, 288 MHz
- 2 Port Real-Time Ethernet Switch with Integrated PHYs
- 2 Micro-Coded Protocol Processing Unit (PPU) per Ethernet Port
- 1 Additional Protocol Processing Units (PPU)
- PPU Concept: High Performance and Flexibility, Prepared for Other Protocols
- Direct Access from Ethernet Switch (PPU) to SNAP+ Master, Consistency and FIFO Interface and DDR2-SDRAM for Fast and Efficient I/O Data Exchange with Minimum CPU Interaction and System Load
- DDR2 Interface with 200 MHz for Higher Memory Bandwidth
- New NAND–Flash Controller with 16-bit ECC to Support Latest NAND-Flash Technologies
- QuadSPI Controller to Speed-Up Boot Sequence
- Advanced Host Interface for External Processor
- Configurable, FIFO Based Mailbox System for Efficient and Flexible Communication Tasks
- Consistency Interface for Hardware Based Exchange of Consistent I/O Data

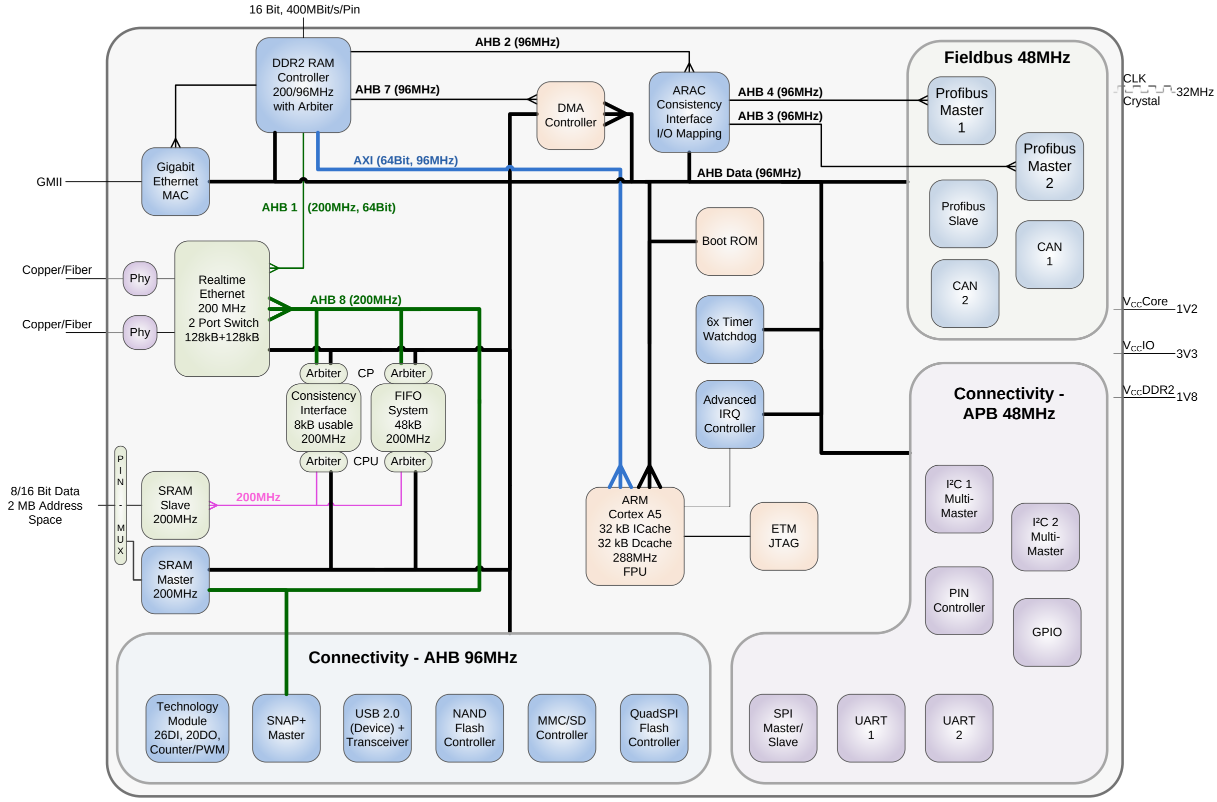
Primary focus of ANTAIOS is the efficient and flexible implementation of high-performance real-time Ethernet communication protocols.

Especially advanced protocols with demanding synchronization mechanisms require dedicated hardware support and shall be addressed with the ANTAIOS chip. These solutions will be based on a combination of micro-coded program execution inside the PPUs for all time critical protocol tasks like synchronization and I/O data exchange, and a high-level protocol stack for non-critical communication tasks (typically provided by a 3rd party cooperation partner).

Pursuing the basic concept of the SMC1000, the ANTAIOS offers several on-chip I/O functions to build block I/Os or small control applications very efficiently. In order to extend the number of I/Os or to realize a modular system concept the integrated SNAP+ Master offers an easy access to profchip's SliceBus technology.

2 Block Diagram

Figure 2-1 System Overview



3 Key IPs and Features

3.1 ARM Cortex-A5 CPU Core

- 288 MHz Core Speed
- 64-bit AXI
- 32 KByte Instruction Cache
- 32 KByte Data Cache
- JTAG Debug Interface
- ETM™ (**E**mbedded **T**race **M**acrocell™) for Real-Time Tracing
- ITM (**I**nstrumentation **T**race **M**acrocell) for Software Instrumentation
- ETB (**E**mbedded **T**race **B**uffer™)
- AHB DAP (**D**ebug **A**ccess **P**ort) for Access of Internal Memory while ARM is Running
- Little Endian Byte Ordering
- 64-bit FPU without NEON

3.2 Advanced Real-Time Ethernet Switch

- 3-Port Switch system to connect two external Ethernet ports with one internal port
- Flexible Architecture Based on a Micro-Coded 5-Core Protocol Processor Unit (PPU) Cluster
- PROFINET® IRT (profichip + Molex)
  - PROFINET IO IRT Specification v2.3
  - Conformance Class C; Real Time Class 3
  - Designed for Cycle Times down to 31.25 µs (High Performance Profile) (Software currently not available for Cycle Times of 31.25 µs)
  - designed for Master and Device
  - Stack: Molex
- MECHATROLINK-III® (Yaskawa)
  - MECHATROLINK-III Master Implementation
  - MECHATROLINK-III Slave Implementation
- EtherCAT® (profichip + Beckhoff)
  - EtherCAT Slave and EtherCAT Master Functionality
  - EtherCAT Technology License Obtained from ETG/Beckhoff
  - EtherCAT Slave Stack by Beckhoff, Support from profichip
- Other planned protocols:
  - EtherNet/IP™ (including CIP Sync™ and DLR)
  - TSN (Time Sensitive Network)
  - Ethernet Powerlink™
  - Modbus® TCP

3.3 Integrated 100Base-TX Ethernet PHYs (2x)

- 2 Integrated 100Base-TX Ethernet PHYs
- 100Base-FX and 10Base-T(e) Support
- special adaptions made to address profichip's Real-Time requirements

3.4 SNAP+ (SliceBus) Master

(to be used in Combination with SNAP+ ASIC)

- SliceBus Features
  - Single Master System
  - up to 64 Slaves (SNAP+ Modules)
  - Asynchronous, Serial Data Transmission with 48 Mbit/s via Point-to-Point LVDS Physics
- Error Detection Mechanism
  - CRC Code with Hamming Distance 4 for Every Telegram (all 3-bit Errors will be Detected)
  - Watchdog Function inside every SNAP+ Module for SNAP+ Master Observation
  - "Auto Shut Down" in Case of SNAP+ Master Malfunction
  - Retry Statistic for Early Detection of Possible Transmission Issues
- Time-Synchronisation
  - Every SNAP+ Module has its own Clock with 1 µs Resolution
  - All SNAP+ Module Clocks are Synchronized with the SNAP+ Master (Accuracy < 100ns)
  - Option for Clock Synchronization from SNAP+ Master to Fieldbus

- SNAP+ Features (SliceBus Slave ASIC)
    - Technological Functions in SNAP+ ASIC
        - Standard I/O Function: 8 DI/DO or 16 DI or 16 DO with Shift Register
        - Integrated Digital Input Filter Function
        - Asynchronous Event Signalling with µs Time Stamping for Advanced SNAP+ Modules
        - Two Advanced Counters with AB Oversampling, Latch, Reset, Output, Hysteresis, Compare Value, Repetitive/Endless Counting and Additional Time Stamp Information
        - SSI Function with Time Stamp Information (Speed Calculations: Counter Difference/Time)
        - Pulse Width Modulation with 20ns Resolution
        - Frequency Measurement Mode
        - Special Digital I/O Time Stamp Modules (ETS: Edge Time Stamp System) for Input Edge and Output Control with 1 µs Resolution (Independent from Fieldbus Cycle!)
    - SPI Interface in SNAP+ for Analog I/O, Safety I/O or Serial CP with External MCU
        - 2.6 Mbit/s SPI Interface for External Microcontroller
        - Up to 16 Byte IN / 16 Byte OUT Data for External Microcontroller
        - Up to 192 Byte of Parameter Data for External Microcontroller
        - Alarm Function and Watchdog Function

3.5 Gigabit Ethernet MAC

- 10/100/1000 Mbit/s Support
- GMII Support
- DMA Engine for Transmitting and Receiving Packets with Scatter Gather List
- Supports IP, TCP and UDP Checksum Offloads
- IEEE 802.1Q VLAN Tag Insertion for Packet Transmission, VLAN Tag Detection and Removal for Packet Reception

3.6 DDR2 SDRAM Controller (16-bit)

- 800 MByte/s maximum bandwidth
- 200 MHz Clock Rate (400 MHz Data Rate)
- 256 MByte maximum addressable[1)]
- 1 chip select

[1)] For memory configuration see chapter 4.10

3.7 Asynchronous External Interface (AEI)

- Configurable 8-bit/16-bit Master Interface:
- Setup, Hold, Access Time and Pause Time Configurable
- 2 Chip Selects with 2 MB Address Range Each and Independent Timings
- 1 Dedicated External IRQ for ARM
- 1 Dedicated External IRQ for SNAP+ Master Synchronization
- Optional WAIT Signal
- Slave Interface (FIFO / CI) 16 Bit only:
  Access Time of 70 ns in Fastest Mode

3.8 FIFO Interface

- FIFO Interface Connected to Real-Time Ethernet Switch, Internal ARM Processor and AEI Slave
- 48 KByte Total Memory, Divided into 256 FIFOs
- 255 IRQ Flags

3.9 Consistency Interface (CI)

- Direct Connection to the Real-Time Switch and the AEI Slave
- 8 KByte Input + 8 KByte Output with Consistency Control
- Byte Reorder Function, e.g.
  - Unaligned Endianness Change with Knowledge of Data Structure
  - Separate PROFINET IOPS/IOCS from I/O Data if required
  - Generate Data Areas with Different Application Update Cycles
    (e.g. 1 ms and 250 µs for IO Data of One Device)
- 8 Process Image Partitions

3.10 PROFIBUS DP Master (2x)

- 2 Independent PROFIBUS® DP Master
- Compliant with PROFIBUS Standard IEC 61158
- Supports DP-V0, DP-V1, DP-V2 (DxB, IsoM, ClockSync)
- PROFIBUS DP Master Stack Available from profichip/Candeo

3.11 VPC3+ PROFIBUS DP Slave

- PROFIBUS DP Slave with Data Rates up to 12 Mbit/s
- Compliant with PROFIBUS Standard IEC 61158
- 4 KByte Communication RAM
- Supports DP-V0, DP-V1, DP-V2 (DxB, IsoM, ClockSync)
- Hardware-PLL for DP-V2 IsoM
- Hardware Synchronization Signal to SNAP+ Master

3.12 CAN Interface (2x)

- FullCAN Controller for Data Rates up to 1 Mbit/s
- Complies with CAN Standard ISO 11898
- Up to 15 Messages Simultaneously (Each with Maximum Data Length)
- Different Message Buffers can be combined as FIFO
- Listen only Mode (Monitoring of the CAN-Bus, No Acknowledge, No Error Flags)
- Support of Clock Synchronization Between ANTAIOS Based Stations

3.13 NAND Flash Controller

- 8-bit NAND-Flash Controller
- DMA Capable in Conjunction with Main DMA Controller
- ECC: 16-bit Correctable for 512 Byte

3.14 QuadSPI Interface

- Max. 96 MHz per 4-line (max. 384 Mbit/s)
- DMA Mode
- Programmable Serial Bit Clock Polarity, Phase and Frequency
- SPI Serial Mode, Dual Mode and Quad Mode
- Additional Optional 4th Address Byte (Extend Address Space up to 4096 M)
- 2 Chip Select Lines

3.15 SD/MMC Card Controller

- Supports the MMC Bus Protocol, Version 4.3
- Compliant with the SD Memory Card Protocol Version 3.0
- Write Protect Pin
- Card Detect Pin
- Integrated DMA Controller
- Built-in Generation and Check for 7-bit and 16-bit CRC Data
- 1 KByte FIFO Buffer
- 4-bit Mode
- High Speed 25 MByte/s possible

3.16 USB 2.0 Device Controller

- USB 2.0 High Speed Device Controller (480 Mbit/s)
- 8 Endpoints
- Integrated USB PHY

3.17 Advanced IRQ Controller

- 8 Priority Levels
- Round-Robin Option for IRQs with the Same Priority
- Throttling Option for Every IRQ Channel
- All IRQs Can Be Masked
- 32-bit ISR Vector for Each IRQ
- Configurable Input Filters for External IRQs
- IRQ/FIQ Selectable for Each IRQ Channel

3.18 Main DMA Controller

- Scatter/Gather Capable with Chained Transfer (Linked List)
- 8 DMA channels
- Support for Fixed Source Address (Read from Auto-Increment-Register) to Memory
- Support for Reading from 8-bit Device and Copy to 32-bit Device

3.19 AHB/APB Bridge (2x)

- DMA channels

3.20 SPI Interface

- Master Mode with up to 80 Mbit/s
- Slave Mode with up to 24 Mbit/s
- DMA Mode in Conjunction with APB-Bridge
- Programmable Frame/Sync. Polarity
- Programmable Serial Bit Clock Polarity, Phase and Frequency
- Programmable Serial Bit Data Sequence (MSB or LSB First)
- 2 chip select lines

3.21 UART (2x)

- Standard Features (Compatible to 16C550):
  - 5/6/7/8 Data Bits
  - 1/1.5/2 Stop Bits
  - None/Odd/Even/Stick Parity
  - Register/FIFO Mode
  - Line Break Generation & Detection
  - Programmable Baud Rate Generator
  - Fully Prioritized Interrupt System Controls
  - Status Reporting Capabilities
  - Modem Control Functions
  - Loopback Mode
- Enhanced Features:
  - High Speed Mode for Higher Baud Rates up to 12 Mbit/s
  - Module Controlled Activation/Deactivation for RTS
  - 32-Byte FIFO with 16C650 DMA Behaviour
  - DMA Mode in Conjunction with APB-Bridge
  - Enable/Disable Receiver
  - IRQ Generation by Extended Timeout Control/Detection
  - IRQ Generation by Two Configurable ETX Characters
  - IRQ Generation by Receive Byte Counter
  - IRQ Generation by Transmitter with Selectable “THR Empty” or “TSR Empty”

3.22 I²C Interface

- Master or Slave for the I²C bus
- Data is Transmitted to and Received from the I²C Bus via a Buffered Interface
- Supports the Standard and Fast Modes
- Supports the 7-bit, 10-bit, and General-Call Addressing Modes
- Glitch Suppression by Debounce Circuit
- Programmable Slave Address
- Supports the Master-Transmit, Master-Receive, Slave-Transmit, and Slave-Receive Modes
- Supports the Multi-Master Mode
- General-Call Address Detection in the Slave Mode

3.23 Timer and Watchdog Module

- Timer
  - Six Independent 32-bit Timer with pre-scaler (10 ns – 80 ns Selectable)
  - Interrupt can be issued upon overflow and time-up
  - Each timer has two compare registers
  - Supports increment and decrement modes
  - Six interrupt sources, one for each counter/timer
  - Supports single-shot and free running mode
  - Automatically reloaded when reaching zero
- Watchdog
  - 32-bit Down Counter with Prescaler
  - Access Protection
  - Mode 1: System Reset or IRQ at Watchdog Event
  - Mode 2: Watchdog IRQ at First Watchdog Event, System Reset at Next Watchdog Event (Can Be Used for Debugging)
  - Option to pass information through the System Reset: Two Registers with POWER-ON-RESET Only (not Affected by Watchdog-Reset)

3.24 Boot Code

- Boot Option Selectable by two dedicated pins
- Boot from QuadSPI NOR-Flash
- Boot from NAND-Flash
- Boot from UART 1
- Boot from Parallel NOR-Flash

3.25 Technology Function Module (TechIO)

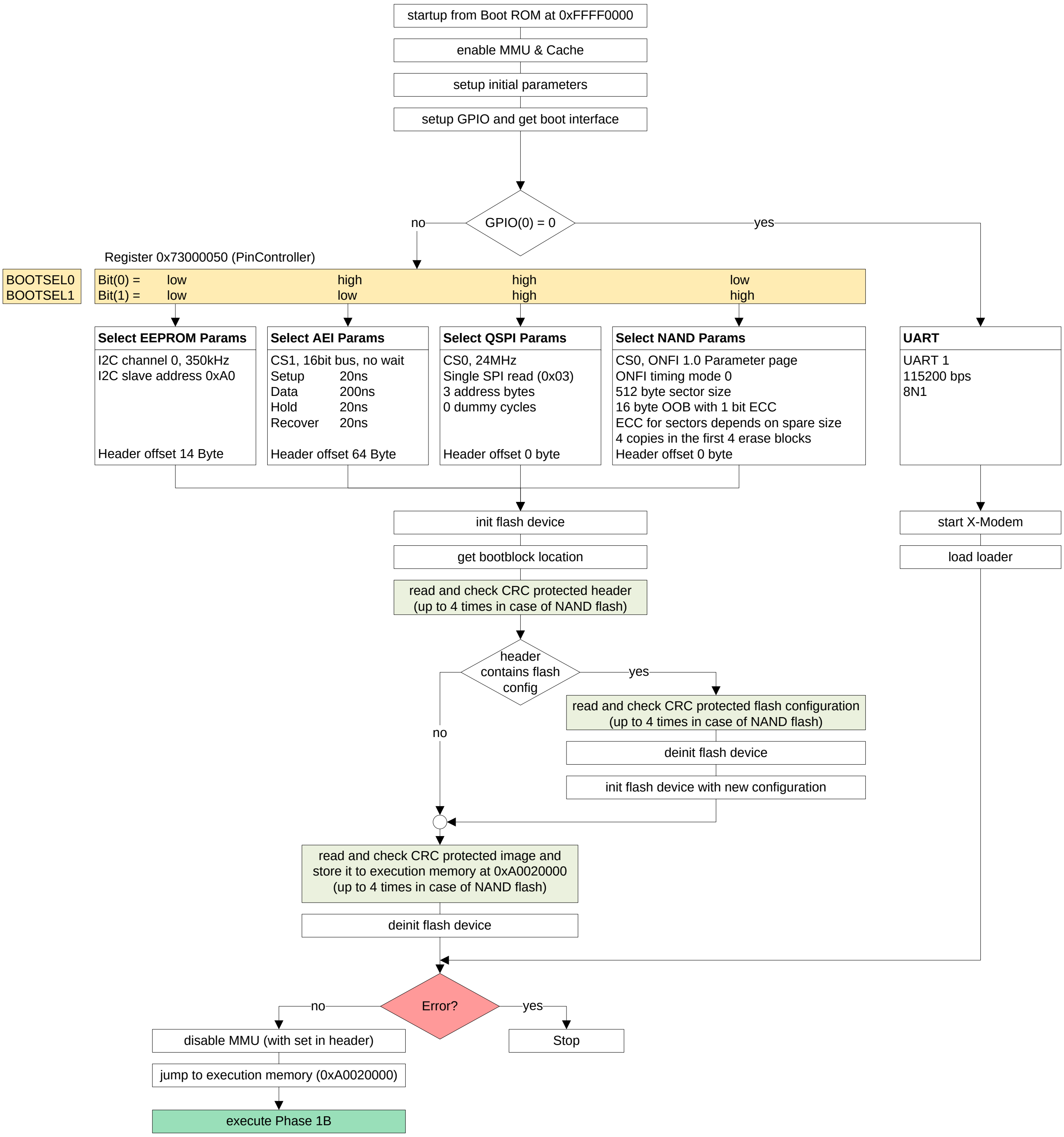
- Max. 26 Bits Input and 20 Bits Output (Shared with Other Interfaces)
- Configurable Digital Input Low Pass Filter
- Up to 4 Counter Channels with Quadruple Evaluation for Incremental Encoders
- Up to 4 PWM Channel (**P**ulse **W**idth **M**odulation)
- Up to 2 SSI Encoder Interfaces

4 Application Details

4.1 Bootloader

- Phase 1 Bootloader in internal boot ROM:
  Boot source selected by BOOTSEL0/1

Figure 4-1 Bootup-Sequence, Phase 1



- CRC protected image is copied into internal memory and executed

  The image is named Bootloader Phase 1B, if a Bootloader Phase 2 like Redboot is used which does not fit into internal memory. Therefore Bootloader Phase 1B must first initialize DDR2 ram and can then load Bootloader Phase 2.
  The destination address of Bootloader Phase 2 is user defined.

  If the image is only used to initialize the DDR2 memory, copy and execute the application and fits into internal memory it will be named Bootloader Phase 2.
  In this case no additional Bootloader Phase 1B is needed.

- Bootloader Phase 1B and Phase 2 use CRC check and are provided by profichip

4.2 eCos

- eCos is open source with modified GPL License:
  All sources of eCos and the Board Support Package (drivers for eCos) have to be publicly available, the user application has not to be disclosed!
- Support for ISR (Interrupt Service Routine, limited use of system function),
  DSR (Deferred Service Routine, more system functions possible),
  Threads (all system functions possible, could sleep, have different priorities)
- Spinlocks, semaphores, mutexes, alarms
- Timer/delay/clock functions
- IPv4 and IPv6 TCP/IP stack (BSD-Stack)
- Basic HTTP webserver
- SNMP support
- Basic IPsec support
- FAT16 and FAT32 file system support to access SDCard, eMMC or MMC
- JFFS2 file system to access QSPI NOR flash
- USB 1.1 software stack (supports also 480MBit/s)
  - With mass storage device profile
  - With serial device profile
- Initialisation of MMU is done during eCos startup

4.3 eCos Drivers (Board Support Package)

- GBit Ethernet MAC with a selection of supported external PHYs
- Timer/Watchdog initialization
- QSPI controller with a selection of supported external devices
- NAND flash controller with a selection of supported external devices
- USB 2.0 endpoint controller
- I²C controller (access to EEPROM)
- SPI master and slave controller
- SDCard/eMMC/MMC controller

4.4 Deliverables (eCos, toolchain) by profichip

- Source of eCos (also available by www.ecoscentric.com)
- Cygwin 2.5.2: enable GCC to run within Windows 7 64bit and 32bit
- GCC 5.4 cross-compiler for ARM on Cygwin (Windows 7)
- Installer for all components mentioned above
- Tool for downloading bootloader and binary by UART to QSPI flash
- Precompiled eCos library with basic settings,
  Customer can configure and compile eCos if necessary

4.5 EtherCAT Slave

4.5.1 Requirements Hardware

- ANTAIOS chip
- Magnetics and passive components for Ethernet PHYs
- DDR2 memory chip (64 MByte, 16 bit)
- QuadSPI flash (4 MByte)
- Power supply 3.3 V / 1.8 V / 1.2 V
- Optional: (for communication with external MCU)
  - SPI
  - USB (endpoint only)
  - 16-bit parallel interface

4.5.2 Requirements Software

- eCos toolchain mentioned above also includes
  the Hardware Abstraction Layer for the EtherCAT Slave Stack Code
- Customer has to acquire an EtherCAT vendor ID from www.Ethercat.org
- Customer has to download ET9300 (EtherCAT Slave Stack Code)
- Customer can generate an adapted version of EtherCAT Slave Stack Code by selecting ANTAIOS within the DropDown menu
- profichip provides application demo with
  - 1 Input SyncManager activated + FMMU 32 Byte
  - 1 Output SyncManager activated + FMMU 32 Byte
  - 2 Mailbox SyncManager + 1 FMMU
- Basic CoE dictionary
- EoE provides access to HTTP webserver
- FoE allows firmware update
- Recommended operating modes:
  - SM Synchronous: user application callback when EtherCAT frame was received
  - DC Synchronous: user application callback when SYNC0 event was activated
- Release notes for EtherCAT (PAAS1120) for more details and
  AN_ET9300 (available by ETG) for the user application interface of the EtherCAT Slave Stack

4.6 PROFINET Device

- profichip application includes the Hardware Abstraction Layer for the Molex PROFINET Device SDK
- Customer has to acquire a PROFINET vendor ID
- Customer has to acquire Ethernet MAC addresses
- Access to source code of Molex PROFINET Device SDK is subject to separate license
- profichip provides (bundled with chip)
  - default Molex PROFINET Device SDK library
  - application demo as precompiled binary
    - 2 Input
    - 2 Output
    - IRT capable
- PROFINET RT/IRT 2.3 (reference: tester bundle April 2016)
  Legacy Startup / Advanced Startup
  Conformance Class C
  Netload Class III
  Cycle time ≥ 250 µs
  1 Application Relation (AR)
  Planned feature extensions:
  - MRP
  - Shared device
  - High performance profile (31.5 µs cycle time)
- Release notes for PROFINET (PAAS1121/1122) for more details

4.7 Ethernet PHYs

The following figures show the interfaces for the internal Ethernet PHYs.

Figure 4-2 Twisted Pair interface for 100Base-TX and 10Base-T operation

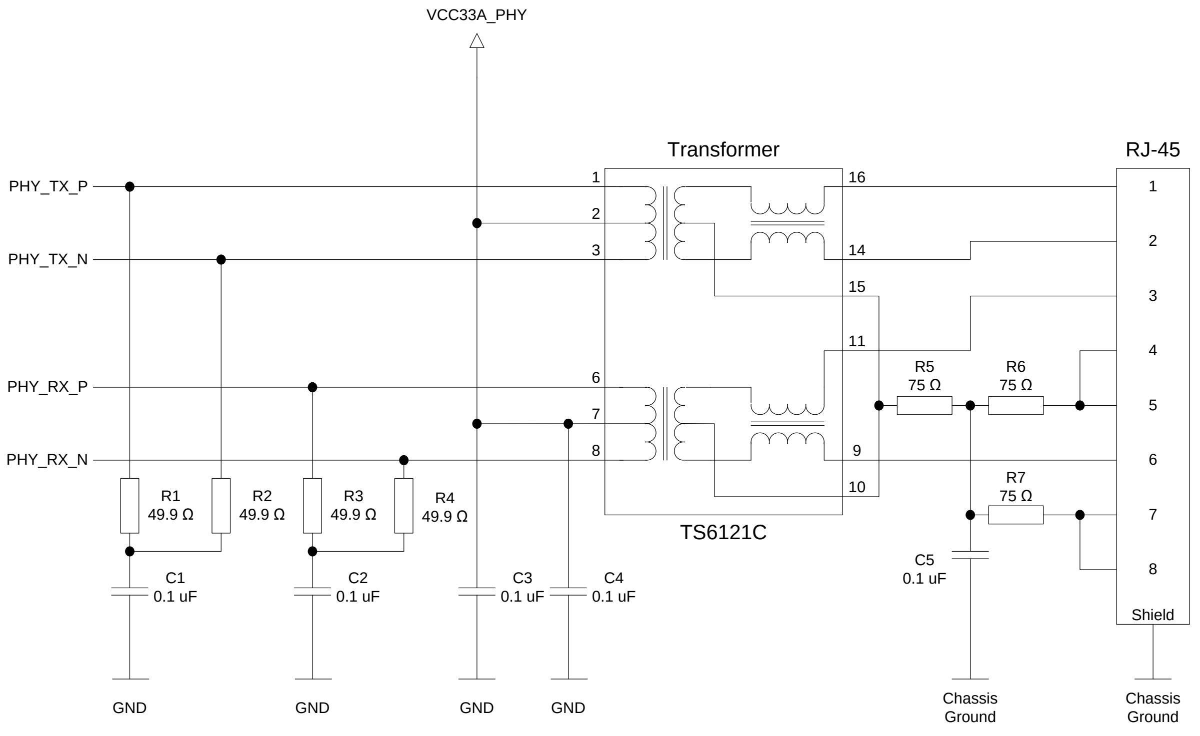
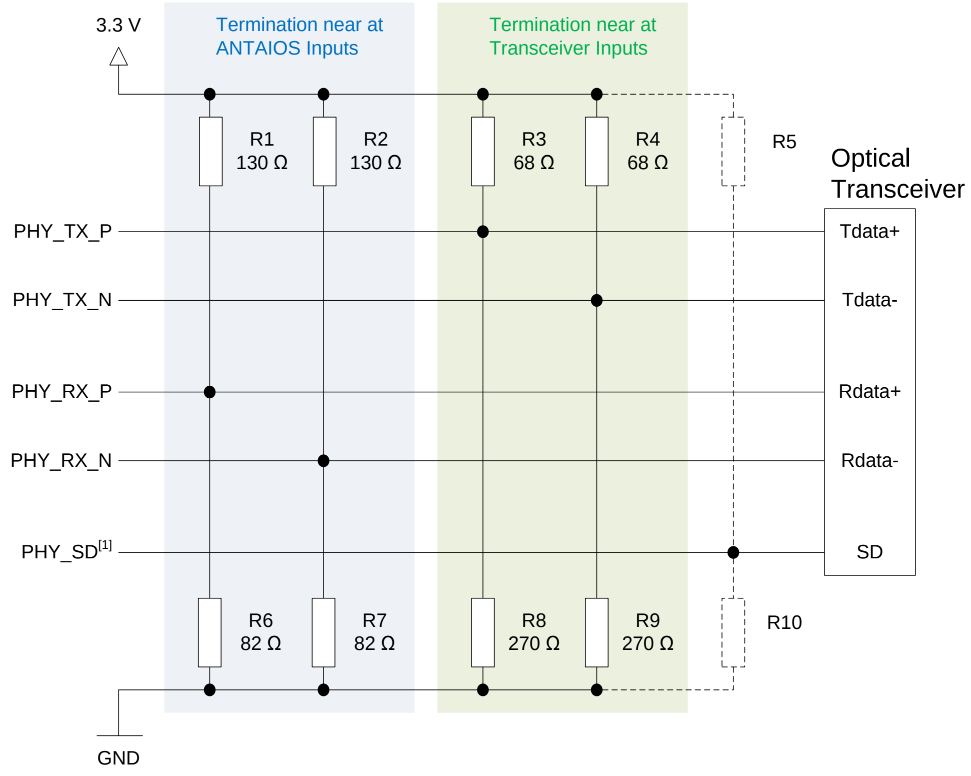


Figure 4-3 Optical Interface



[1] The voltage levels of PHY_SD must be respected (see Table 7-15 Characteristics of internal PHYs). The usage of a voltage divider depends on the used optical transceiver.

4.7.1 Qualified Chips

- 10/100Base-T isolation transformer: Universal Microelectronics UE-TS6121C
- 10/100Base-T isolation transformer: Pulse H1102

4.8 SliceBus Interface

4.8.1 Qualified Chips

- High-speed differential line driver/receiver: Texas Instruments SN65LVDxx

4.9 Gigabit Ethernet Interface

4.9.1 Qualified Chips

- 1000 Mbit PHY: Marvell 88E1119R

4.10 DDR2 Memory

4.10.1 64 MByte Configuration (1 chip)

- Single chip 512 Mbit DDR2-800 x16 (800 MBytes/s usable bandwidth)
- Memory chip: row address 13 bit (8K); column address 10 bit (1K); 4 banks (2 bit); x16
- Configuration: row address 13 bit (8K); column address 10 bit (1K); 4 banks (2 bit)
- Differential clock; differential DQS

4.10.2 128 MByte Configuration (2 chips)

- 2 chips 512 Mbit DDR2-800 x8 (800 MBytes/s usable bandwidth)
- Memory chip: row address 14 bit (16K); column address 10 bit (1K); 4 banks (2 bit); x8
- Configuration: row address 14 bit (16K); column address 10 bit (1K); 4 banks (2 bit)
- Differential clock; differential DQS

4.10.3 128 MByte Configuration (1 chip)

- Single chip 2048 Mbit DDR2-800 x16 (800 MBytes/s usable bandwidth) (only half size is used, keep BA2 of memory chip at GND level)
- Memory chip: row address 14 bit (16K); column address 10 bit (1K); 8 banks (3 bit); x16
- Configuration: row address 14 bit (16K); column address 10 bit (1K); 4 banks (2 bit)
- Differential clock; differential DQS

4.10.4 256 MByte Configuration (2 chips)

- 2 chips 2048 Mbit DDR2-800 x8 (800 MBytes/s usable bandwidth) (only half size is used; not all AHB master can access to full memory range (> 128 MByte))
- Memory chip: row address 15 bit (32K); column address 10 bit (1K); 8 banks (3 bit); x8
- Configuration: row address 15 bit (32K); column address 10 bit (1K); 8 banks (3 bit)
- Differential clock; differential DQS

4.10.5 Qualified Chips

- Alliance Memory AS4C16M16D2-25BIN (256 Mbit, 1.8 V)
- Intelligent Memory IM5116D2DABG-25I (512 Mbit, 1.8 V)
- Micron MT47H64M16NF-25E:M (1024 Mbit, 1.8 V)
- Nanya Technology NT5TU32M16EG-ACI (512 Mbit, 1.8 V)
- Winbond W9751G6NB-25I (512 Mbit, 1.8 V)
- Winbond W9751G6KB-25I (512 Mbit, 1.8 V), EoL

4.11 Asynchronous External Interface (AEI)

4.11.1 SRAM Master Mode: Port D Mode V1

4.11.1.1 Qualified Chips

- nvSRAM: Cypress CY14B108N-BA25XI (8 Mbit, 3.0 V)

4.11.2 SRAM Slave Interface: Port D Mode V2

- Internally Connected to CI and FIFO Interface
- Always 16 Bit data bus necessary
- AEI_S_A_00 up to AEI_S_A_07 (256 Byte address range each) for access of the CI/FIFO Registers
- AEI_S_A_20 necessary to select between CI (1) and FIFO(0)
- All other address lines (AEI_S_A_08 up to AEI_S_A_19) have to be tied to ‘0’
- Timing: Chapter 0
- Pinning: Chapter 5.2.9.5
- Only one chip select
- Byte Select necessary
- Wait Signal is optional
- IRQ Output available

4.12 Asynchronous/SDR NAND Flash

4.12.1 Qualified Chips

- Micron MT29F8G08ABACAH4 (8 Gbit, 3.3 V)

4.13 QuadSPI NOR Flash

4.13.1 Up to 128 Mbit Configuration

- Single chip up to 128 Mbit (3-Byte address mode)

4.13.2 Beyond 128 Mbit Configuration

- Single chip beyond 128 Mbit (4-Byte address mode)
- SPI flash memory has to start operation in 3-Byte address mode after power cycle because of boot-loader routine inside the internal boot ROM of ANTAIOS

4.13.3 2 Chip Configuration

- Different sizes of flash memories possible
- Same address handling as mentioned above

4.13.4 Qualified Chips

- Adesto AT25SF321-MHD (32 Mbit, 2.5 V – 3.6 V)
- Cypress (Spansion) S25FL132K0XMFI01 (32 Mbit, 2.7 V – 3.6 V)
- Integrated Silicon Solution IS25LP064A (64 Mbit, 2.3 V – 3.6 V)
- Integrated Silicon Solution IS25LP128-JBLE (128 Mbit, 2.3 V – 3.6 V)
- Macronix MX25L3233FZNI-08G (32 Mbit, 2.65 V – 3.6 V)
- Macronix MX25L6433F (64 Mbit, 2.65 V – 3.6 V)
- Winbond W25Q32FVZPIG (32 Mbit, 2.7 V – 3.6 V)

Qualified, but not recommended for new designs:

- Micron N25Q064A13EF640 (64 Mbit, 2.7 V – 3.6 V), EoL

4.14 SPI Slave Connection

- 2 SPI slaves can be connected (use of different chip select signals)
- more SPI slaves can be connected if they support daisy-chaining

4.15 SPI Master Connection

- chip select 0 is used for SPI slave mode

4.16 I²C Interface

4.16.1 Qualified Chips

- EEPROM: CAT24WC64 (64 Kbit, 1.8 V – 6 V)
- RTC: RTC-8564JE (1.8 V – 5.5 V)

4.17 JTAG Debug Interface

- used for in-circuit debugger for on-chip debugging, e.g.
    - Lauterbach POWER DEBUG II
    - SEGGER J-Link

  debugger scripts are included in PAA1100
- supports Instrumentation Trace Macrocell (ITM)
- supports AHB access through Debug Access Port (DAP) in order to read and write internal memory and registers while ARM is running
- max. JTAG frequency is 48 MHz

4.18 ETM Trace Output

- CoreSight™ Embedded Trace Macrocell (ETMv3)
    - 4 KByte Embedded Trace Buffer (ETB) to store the compressed trace information. (The trace is read out at low speed using the JTAG interface when the trace capture is complete.)
    - Trace Port to export the compressed trace information to an external Trace Port Analyzer.
- used for real-time monitoring of instruction and data flow
- complete code coverage
- successfully tested with Lauterbach POWER DEBUG II (in-circuit debugger) + POWER TRACE II (trace hardware with trace memory) + AUTOFOCUS II (pre-processor); use of AUTOFOCUS II pre-processor is mandatory, because of its improved sampling capabilities
- trace clock frequency is 144 MHz

4.18.1 16-bit Output

16-bit ETM output is the default configuration; no additional port signals are required and therefore the use of other interfaces is not restricted

4.18.2 32-bit Output

32-bit ETM output is very challenging on signal integrity and power supply due to 144 MHz ETM clock with 288 Mbit/s data rate on each of the 32 ETM output pins. But it provides the capability to trace instruction and data of the Cortex-A5 almost without FIFO overflows and provides very advanced debugging and analysing capabilities.
The 16-bit output (mentioned above) is extended by 16 additional signals of port E.

5 Pin Description

5.1 Pinout

The ANTAIOS is available in two package versions: TFBGA-380 or TFBGA-385. Several pins are sharing different functions. Which pin function actually applies depends on the configuration of the Pin Controller. Please see the following chapters for details.
Details about package outlines and dimensions are listed in chapter Package Specifications.

Table 5-1 color scheme

ground		
core power supply	IO power supply	SSTL_18 (DDR2) power supply
		SSTL_18 reference voltage
reset, clock and chip test	JTAG port	ETM port
DDR2-SDRAM interface	RTGPIO	PHY 1
	GPIO	PHY 2
USB	SliceBus	(G)MII
SPI 1	UART 1	
Port A	Port B	Port C
Port D	Port E	

5.1.1 TFBGA-380

The ball maps below show the pin assignments on the TFBGA-380 package in four sections (upper left, upper right, lower left and lower right).

Figure 5-1 TFBGA-380 Ball Map - Upper Left Corner (Top View)

	1	2	3	4	5	6	7	8	9	10	11
A	PORT_E_0	DBG_STB	GMAC_TXD_2	GMAC_COL	GMAC_RX_ER	GMAC_TX_CK	GMAC_RXD_2	RT_GPIO_0	RT_GPIO_4	PORT_A_0	PORT_A_4
B	PORT_E_2	PORT_E_1	GMAC_CRS	GMAC_TX_ER	GMAC_RXD_1	GMAC_MDIO	GMAC_RXD_3	RT_GPIO_1	RT_GPIO_5	PORT_A_1	PORT_A_5
C	PORT_E_4	PORT_E_3	GND	GMAC_TXD_1	GMAC_RX_DV	GMAC_MDC	GMAC_RXD_0	RT_GPIO_3	RT_GPIO_7	PORT_A_3	PORT_A_7
D	PORT_E_8	PORT_E_7	PORT_E_5	BOOT SEL0	GMAC_TXD_0	GMAC_TXD_3	GMAC_RX_CK	GMAC_CLK_IN 125	RT_GPIO_2	PORT_A_2	PORT_A_6
E	PORT_E_12	PORT_E_11	PORT_E_6	DBG_CLK		GMAC_TX_EN	VCC_CORE	GND	VCC3IO	RT_GPIO_6	PORT_A_10
F	PORT_E_14	PORT_E_13	PORT_E_10	PORT_E_9	VCC_CORE						
G	PORT_D_0	PORT_E_15	PORT_D_3	VCC3IO	GND						
H	PORT_D_4	PORT_D_1	PORT_D_7	PORT_D_2	VCC3IO				GND	VCC_CORE	GND
J	GND	GND	PORT_D_5	PORT_D_10	PORT_D_6			GND	GND	VCC_CORE	GND
K	USB_DN	USB_DP	VCC33A_USB	VCC_CORE	GND			VCC_CORE	VCC_CORE		
L	GND	VCC33A_USB	GND	PORT_D_11	PORT_D_8			GND	GND		

Figure 5-2 TFBGA-380 Ball Map - Upper Right Corner (Top View)

12	13	14	15	16	17	18	19	20	21	22	
PORT_A_8	PORT_A_12	PORT_B_1	PORT_B_5	PORT_B_9	PORT_B_13	PORT_B_17	PORT_B_21	PORT_B_25	PORT_B_29	BOOT SEL1	**A**
PORT_A_9	PORT_B_0	PORT_B_4	PORT_B_8	PORT_B_12	PORT_B_16	PORT_B_20	PORT_B_24	PORT_B_28	PORT_B_31	PORT_B_32	**B**
PORT_A_11	PORT_A_14	PORT_B_3	PORT_B_7	PORT_B_11	PORT_B_15	PORT_B_19	PORT_B_22	PORT_B_23	PORT_B_27	PORT_B_34	**C**
PORT_A_13	PORT_B_6	PORT_B_10	PORT_B_14	PORT_B_18	VCC3IO	VCC_CORE	PORT_B_30	PORT_B_26	PORT_B_35	JTAG_SEL	**D**
PORT_B_2	VCC3IO	VCC_CORE	GND	VCC_CORE	GND		PORT_B_33	GND	SRST_N	JTAG_TDO	**E**
						VCC3IO	ON_CHIP_TESTER	JTAG_TCK	JTAG_TDI	JTAG_TMS	**F**
						GND	JTAG_TRST_N	ETM_CLK	ETM_DATA_6	ETM_DATA_7	**G**
VCC_CORE	GND	VCC_CORE				VCC_CORE	ETM_DATA_4	ETM_DATA_5	ETM_DATA_15	ETM_DATA_14	**H**
VCC_CORE	GND	VCC_CORE	VCC_CORE			GND	VCC3IO	ETM_DATA_3	ETM_DATA_13	ETM_DATA_12	**J**
		GND	GND			ETM_DATA_0	ETM_DATA_2	ETM_DATA_1	ETM_DATA_11	ETM_DATA_10	**K**
		VCC_CORE	VCC_CORE			UART1_CTS_N	UART1_RTS_N	ETM_CTL	ETM_DATA_9	ETM_DATA_8	**L**

Figure 5-3 TFBGA-380 Ball Map - Lower Left Corner (Top View)

	1	2	3	4	5	6	7	8	9	10	11
M	PORT_D_12	PORT_D_9	PORT_D_15	PORT_D_18	PORT_D_14			VCC_CORE	VCC_CORE		
N	PORT_D_16	PORT_D_13	PORT_D_19	PORT_D_22	TEST			GND	GND		
P	PORT_D_20	PORT_D_17	PORT_D_23	PORT_D_26	PORT_D_30			VCC_CORE	VCC_CORE	GND	VCC_CORE
R	PORT_D_24	PORT_D_21	PORT_D_27	PORT_D_31	VCC3IO				VCC_CORE	GND	VCC_CORE
T	PORT_D_28	PORT_D_25	GND	PORT_D_33	VCC3IO						
U	PORT_D_32	PORT_D_29	VCC3IO	PORT_D_34	VCC_CORE						
V	PORT_D_36	VCC3IO	PORT_D_35	PORT_D_43		VCC12D_PHY1	GND	VCC_CORE	GND	VCC3IO	VCC_CORE
W	PORT_D_38	PORT_D_39	PORT_D_37	VCC3IO	VCC33A_PHY1	PHY1_BIAS	PHY2_BIAS	PHY2_SD	SBUS_MDLO	PORT_C_7	PORT_C_5
Y	PORT_D_40	PORT_D_41	GND	VCC12A_PHY1	PHY1_SD	VCC12D_PHY2	VCC12A_PHY2	VCC33A_PHY2	SBUS_ALARM_N	PORT_C_6	PORT_C_4
AA	PORT_D_42	VCC12A_PHY1	PHY1_TX_P	PHY1_RX_P	VCC12A_PHY1	PHY2_RX_P	PHY2_TX_P	VCC12A_PHY2	SBUS_NDLI	PORT_C_2	PORT_C_0
AB	PORT_D_44	GND	PHY1_TX_N	PHY1_RX_N	GND	PHY2_RX_N	PHY2_TX_N	GND	PORT_C_3	PORT_C_1	GPIO_2

Figure 5-4 TFBGA-380 Ball Map - Lower Right Corner (Top View)

		GND	GND			VCC3IO	SPI_CS0_N	UART1_RXD	SPI_RXD	UART1_TXD	**M**
		VCC_CORE	VCC_CORE			GND	SPI_CS1_N	SPI_TXD	SPI_CLK	SPI_CLKIN	**N**
GND	VCC_CORE	GND	GND			VREF_SSTL18_1	VCC18O_DDR	DDR_DQ_15	DDR_DQ_10	DDR_DQ_11	**P**
GND	VCC_CORE	GND				VCC12A_DLL_DDRD	GND	DDR_DQ_14	DDR_UDQS_N	DDR_UDQS	**R**
						VCC18O_DDR	DDR_DQ_08	DDR_DQ_13	DDR_DQ_09	DDR_UDM	**T**
						VCC_CORE	DDR_DQ_02	VCC18O_DDR	DDR_DQ_03	DDR_DQ_12	**U**
VCC_CORE	VCC12AD_PLL1	VCC12AD_PLL2	GND	VCC18O_DDR	GND		VREF_SSTL18_0	GND	DDR_DQ_06	DDR_DQ_07	**V**
GPIO_4	VCC3IO	GND	DDR_A_08	VCC18O_DDR	DDR_BA_1	DDR_RAS	DDR_CAS	DDR_DQ_00	DDR_LDQS_N	DDR_LDQS	**W**
GPIO_5	GPIO_3	DDR_CKE	DDR_A_11	DDR_A_05	DDR_A_02	DDR_A_00	VCC18O_DDR	GND	DDR_DQ_01	DDR_LDM	**Y**
GPIO_0	RESET_N	DDR_A_14	DDR_A_12	DDR_A_06	DDR_A_01	DDR_CLK_N	DDR_BA_0	DDR_WE	DDR_DQ_04	DDR_DQ_05	**AA**
GPIO_1	CLK32	DDR_A_09	DDR_A_07	DDR_A_04	DDR_A_03	DDR_CLK	DDR_A_10	DDR_CS	DDR_ODT	DDR_A_13	**AB**
12	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	

5.1.2 TFBGA-385

The ball maps below show the pin assignments on the TFBGA-385 package in four sections (upper left, upper right, lower left and lower right).

Figure 5-5 TFBGA-385 Ball Map - Upper Left Corner (Top View)

	1	2	3	4	5	6	7	8	9	10	11	12
A	DBG_CLK	DBG_STB	GMAC_TXD_0	GMAC_TX_EN	GMAC_TXD_3	GMAC_RX_ER	GMAC_RXD_0	RT_GPIO_3	RT_GPIO_0	RT_GPIO_2	PORT_A_0	PORT_A_4
B	PORT_E_0	PORT_E_1	GMAC_CRS	GMAC_TXD_2	GMAC_RX_DV	GMAC_MDC	GMAC_TX_CK	GMAC_CLK_IN 125	RT_GPIO_1	RT_GPIO_4	PORT_A_1	PORT_A_5
C	PORT_E_6	PORT_E_5	PORT_E_9	BOOT SEL0	GMAC_TX_ER	GMAC_COL	GMAC_RX_CK	GMAC_RXD_2	RT_GPIO_7	RT_GPIO_6	PORT_A_2	PORT_A_6
D	PORT_E_2	PORT_E_10	PORT_E_3	GND	GMAC_TXD_1	GMAC_RXD_1	GMAC_MDIO	GMAC_RXD_3	VCC3IO	RT_GPIO_5	PORT_A_3	PORT_A_7
E	PORT_E_7	PORT_E_4	PORT_E_8	VCC3IO	GND	GND	VCC3IO				GND	VCC3IO
F	PORT_D_10	PORT_D_0	PORT_E_12	PORT_E_11	GND							
G	PORT_D_4	PORT_D_2	PORT_E_14	PORT_E_13	VCC3IO							
H	GND	GND	PORT_D_3	PORT_E_15								
J	USB_DN	USB_DP	VCC33A_USB	PORT_D_7					GND	VCC_CORE	VCC_CORE	GND
K	GND	VCC33A_USB	PORT_D_1	VCC3IO					VCC_CORE	GND	GND	GND
L	PORT_D_11	PORT_D_8	PORT_D_6	PORT_D_5	GND				VCC_CORE	GND	GND	GND
M	PORT_D_14	PORT_D_13	PORT_D_12	PORT_D_9	VCC3IO				GND	GND	GND	GND

Figure 5-6 TFBGA-385 Ball Map - Upper Right Corner (Top View)

13	14	15	16	17	18	19	20	21	22	23	
PORT_A_10	PORT_A_12	PORT_B_2	PORT_B_1	PORT_B_4	PORT_B_7	PORT_B_11	PORT_B_18	PORT_B_21	PORT_B_25	PORT_B_30	**A**
PORT_A_11	PORT_A_13	PORT_B_0	PORT_B_3	PORT_B_5	PORT_B_9	PORT_B_15	PORT_B_19	PORT_B_24	PORT_B_28	PORT_B_29	**B**
PORT_A_8	PORT_A_14	PORT_B_6	PORT_B_8	PORT_B_12	PORT_B_16	PORT_B_20	PORT_B_22	PORT_B_31	PORT_B_23	PORT_B_33	**C**
PORT_A_9	VCC3IO	PORT_B_10	PORT_B_14	PORT_B_13	PORT_B_17	VCC3IO	GND	PORT_B_26	PORT_B_35	PORT_B_34	**D**
GND				VCC3IO	GND	GND	BOOT SEL1	PORT_B_27	SRST_N	JTAG_TDO	**E**
						GND	PORT_B_32	JTAG_SEL	JTAG_TCK	JTAG_TMS	**F**
						VCC3IO	ON_CHIP_TESTER	ETM_CLK	JTAG_TDI	JTAG_TRST_N	**G**
							ETM_DATA_6	ETM_DATA_7	ETM_DATA_4	ETM_DATA_5	**H**
VCC_CORE	VCC_CORE	GND					VCC3IO	ETM_DATA_3	ETM_DATA_15	ETM_DATA_14	**J**
GND	GND	VCC_CORE					GND	ETM_DATA_2	ETM_DATA_13	ETM_DATA_0	**K**
GND	GND	VCC_CORE				GND	ETM_DATA_1	ETM_DATA_8	ETM_DATA_12	ETM_DATA_11	**L**
GND	GND	GND				VCC3IO	UART1_RTS_N	ETM_CTL	ETM_DATA_10	ETM_DATA_9	**M**

Figure 5-7 TFBGA-385 Ball Map - Lower Left Corner (Top View)

N	PORT_D_15	PORT_D_16	PORT_D_18	PORT_D_17	GND				VCC_CORE	GND	GND	GND
P	PORT_D_19	PORT_D_20	TEST	GND					VCC_CORE	GND	GND	GND
R	PORT_D_21	PORT_D_22	PORT_D_23	VCC3IO					GND	VCC_CORE	VCC_CORE	GND
T	PORT_D_24	PORT_D_25	PORT_D_26	PORT_D_27								
U	PORT_D_28	PORT_D_29	PORT_D_30	PORT_D_31	VCC3IO							
V	PORT_D_32	PORT_D_33	PORT_D_34	PORT_D_35	GND							
W	PORT_D_36	PORT_D_38	PORT_D_39	VCC3IO	GND	GND	GND				GND	GND
Y	PORT_D_37	PORT_D_40	GND	PHY1_SD	PHY1_BIAS	PHY2_BIAS	PHY2_SD	GND	SBUS_MDLO	PORT_C_7	PORT_C_5	VCC3IO
AA	PORT_D_41	PORT_D_42	GND	VCC33A_PHY1	VCC12D_PHY1	GND	VCC12D_PHY2	VCC33A_PHY2	SBUS_ALARM_N	PORT_C_6	PORT_C_4	GPIO_5
AB	PORT_D_43	GND	PHY1_TX_P	PHY1_RX_P	VCC12A_PHY1	PHY2_RX_P	PHY2_TX_P	VCC12A_PHY2	SBUS_NDLI	PORT_C_2	PORT_C_0	GPIO_4
AC	PORT_D_44	VCC12A_PHY1	PHY1_TX_N	PHY1_RX_N	GND	PHY2_RX_N	PHY2_TX_N	GND	PORT_C_3	PORT_C_1	GPIO_2	GPIO_0
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**

Figure 5-8 TFBGA-385 Ball Map - Lower Right Corner (Top View)

GND	GND	VCC_CORE				GND	UART1_TXD	SPI_CS0_N	UART1_RXD	UART1_CTS_N	**N**
GND	GND	VCC12A_DLL_DDRD					SPI_CLKIN	SPI_TXD	SPI_CS1_N	SPI_RXD	**P**
VCC12AD_PLL1	VCC12AD_PLL2	VCC_CORE					VCC3IO	DDR_DQ_10	SPI_CLK	DDR_DQ_11	**R**
							VREF_SSTL18_1	DDR_UDM	DDR_DQ_14	DDR_DQ15	**T**
						VCC18O_DDR	VCC18O_DDR	DDR_DQ_09	DDR_UDQS_N	DDR_UDQS	**U**
						GND	GND	DDR_DQ_12	DDR_DQ_13	DDR_DQ_08	**V**
VCC3IO				VCC18O_DDR	GND	GND	VREF_SSTL18_0	DDR_DQ_07	DDR_DQ_02	DDR_DQ_03	**W**
GND	VCC18O_DDR	DDR_A_08	DDR_A_04	VCC18O_DDR	VCC18O_DDR	DDR_CS	DDR_DQ_04	DDR_DQ_06	DDR_LDQS_N	DDR_LDQS	**Y**
GPIO_3	DDR_A_14	DDR_A_11	DDR_A_06	DDR_A_02	GND	DDR_RAS	DDR_ODT	VCC18O_DDR	DDR_DQ_01	DDR_LDM	**AA**
RESET_N	DDR_CKE	DDR_A_12	DDR_A_07	DDR_A_01	DDR_CLK_N	DDR_A_10	DDR_BA_0	DDR_WE	GND	DDR_DQ_00	**AB**
GPIO_1	CLK32	DDR_A_09	DDR_A_05	DDR_A_03	DDR_CLK	DDR_A_00	DDR_BA_1	DDR_CAS	DDR_A_13	DDR_DQ_05	**AC**
13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	

5.2 Pin Assignment

5.2.1 General statements and notes

Table 5-2 Ball Characteristics

Symbol	Description
VCC_CORE	+ 1.2 V
VCC_DDR	+ 1.8 V
VCC_IO	+ 3.3 V
GND	0 V
I	LVTTL Input
O	LVTTL Output, Push / Pull
Oe	LVTTL Output, Tristate capable
(S)	Input with Schmitt-Trigger characteristic
(8)	Output can source / sink 8 mA
(12)	Output can source / sink 12 mA
USB	USB data line
PHY	Ethernet PHY data line
Analog	Analog ball
PU	Internal Pull Up resistor (75 kΩ)
PD	Internal Pull Down resistor (75 kΩ)

Table 5-3 Output States

Symbol	Description
X	undefined state: 1 or 0
Z	Tristate (Output inactive)
H	Tristate with internal Pull Up resistor
L	Tristate with internal Pull Down resistor
1	Output drives V_{OH}
0	Output drives V_{OL}

5.2.2 Power / Ground

Table 5-4 Ground

Ball 380	Ball 385	Pin Name	Type	Description
C3	D4	GND	GND	
E8	D20	GND	GND	
E15	E5	GND	GND	
E17	E6	GND	GND	
E20	E11	GND	GND	
G5	E13	GND	GND	
G18	E18	GND	GND	

Ball 380	Ball 385	Pin Name	Type	Description
H9	E19	GND	GND	
H11	F5	GND	GND	
H13	F19	GND	GND	
J1	H1	GND	GND	
J2	H2	GND	GND	
J8	J9	GND	GND	
J9	J12	GND	GND	
J11	J15	GND	GND	
J13	K1	GND	GND	
J18	K10 .. K14	GND	GND	
K5	K20	GND	GND	
K14	L5	GND	GND	
K15	L10 .. L14	GND	GND	
L1	L19	GND	GND	
L3	M9 .. M15	GND	GND	
L8	N5	GND	GND	
L9	N10 .. N14	GND	GND	
M14	N19	GND	GND	
M15	P4	GND	GND	
N8	P10 .. P14	GND	GND	
N9	R9	GND	GND	
N18	R12	GND	GND	
P10	V5	GND	GND	
P12	V19	GND	GND	
P14	V20	GND	GND	
P15	W5 .. W7	GND	GND	
R10	W11	GND	GND	
R12	W12	GND	GND	
R14	W18	GND	GND	
R19	W19	GND	GND	
T3	Y3	GND	GND	
V7	Y8	GND	GND	
V9	Y13	GND	GND	
V15	AA3	GND	GND	
V17	AA6	GND	GND	
V20	AA18	GND	GND	
W14	AB2	GND	GND	
Y3	AB22	GND	GND	
Y20	AC5	GND	GND	
AB2	AC8	GND	GND	
AB5		GND	GND	
AB8		GND	GND	

Table 5-5 Vcc Core, Vcc I/O and special Vcc

Ball 380	Ball 385	Pin Name	Type	Description
D18	J10	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
E7	J11	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
E14	J13	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
E16	J14	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
F5	K9	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
H10	K15	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
H12	L9	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
H14	L15	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
H18	N9	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
J10	N15	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
J12	P9	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
J14	R10	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
J15	R11	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
K4	R15	VCC_CORE	VCC_CORE	Vcc Core 1.2 V
K8		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
K9		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
L14		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
L15		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
M8		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
M9		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
N14		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
N15		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
P11		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
P13		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
P8		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
P9		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
R9		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
R11		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
R13		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
U5		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
U18		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
V8		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
V11		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
V12		VCC_CORE	VCC_CORE	Vcc Core 1.2 V
R18	P15	VCC12A_DLL_DDRD	VCC_CORE	1.2 V analog power supply for DLL (separate filtering required)
AA2	AB5	VCC12A_PHY1	VCC_CORE	1.2 V Vcc PHY 1 for analog part
AA5	AC2	VCC12A_PHY1	VCC_CORE	
Y4		VCC12A_PHY1	VCC_CORE	
V6	AA5	VCC12D_PHY1	VCC_CORE	
AA8	AB8	VCC12A_PHY2	VCC_CORE	1.2 V Vcc PHY 2 for analog part
Y7		VCC12A_PHY2	VCC_CORE	
Y6	AA7	VCC12D_PHY2	VCC_CORE	
V13	R13	VCC12AD_PLL1	VCC_CORE	1.2 V Vcc PLL 1 (separate filtering required)
V14	R14	VCC12AD_PLL2	VCC_CORE	1.2 V Vcc PLL 2 (separate filtering required)

Ball 380	Ball 385	Pin Name	Type	Description
V16	Y17	VCC18O_DDR	VCC_DDR	1.8 V DDR2 I/O power supply
P19	U19	VCC18O_DDR	VCC_DDR	
T18	U20	VCC18O_DDR	VCC_DDR	
U20	W17	VCC18O_DDR	VCC_DDR	
W16	Y14	VCC18O_DDR	VCC_DDR	
Y19	AA21	VCC18O_DDR	VCC_DDR	
	Y18	VCC18O_DDR	VCC_DDR	
D17	D9	VCC3IO	VCC_IO	3.3 V Vcc I/O cells
E9	D14	VCC3IO	VCC_IO	
E13	D19	VCC3IO	VCC_IO	
F18	E4	VCC3IO	VCC_IO	
G4	E7	VCC3IO	VCC_IO	
H5	E12	VCC3IO	VCC_IO	
J19	E17	VCC3IO	VCC_IO	
M18	G5	VCC3IO	VCC_IO	
R5	G19	VCC3IO	VCC_IO	
T5	J20	VCC3IO	VCC_IO	
U3	K4	VCC3IO	VCC_IO	
V2	M5	VCC3IO	VCC_IO	
W4	M19	VCC3IO	VCC_IO	
V10	R4	VCC3IO	VCC_IO	
W13	R20	VCC3IO	VCC_IO	
	U5	VCC3IO	VCC_IO	
	W4	VCC3IO	VCC_IO	
	W13	VCC3IO	VCC_IO	
	Y12	VCC3IO	VCC_IO	
W5	AA4	VCC33A_PHY1	VCC_IO	3.3 V Vcc PHY 1 for analog part
Y8	AA8	VCC33A_PHY2	VCC_IO	3.3 V Vcc PHY 2 for analog part
K3	J3	VCC33A_USB	VCC_IO	3.3 V Vcc USB for analog part of transceiver
L2	K2	VCC33A_USB	VCC_IO	

5.2.3 Basic and Reserved Signals

Table 5-6 Reset, System Clock and Chip Test

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
AB13	AC14	CLK32	I(S)	main input clock: 32 MHz crystal oscillator		
AA13	AB13	RESET_N	I(S)	active-low master reset	PU	
N5	P3	TEST	I(S)	(leave unconnected)	PD	
D4	C4	BOOTSEL0	I(S)	selection of boot source	PD	
A22	E20	BOOTSEL1	I(S)	selection of boot source	PD	
F19	G20	ON_CHIP_TESTER	I(S)	(leave unconnected)	PD	

5.2.4 Development Interfaces

Table 5-7 JTAG Interface for ARM Cortex-A5

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
D22	F21	JTAG_SEL	I(S)		PD	
G19	G23	JTAG_TRST_N	I(S)	active-low test port reset: connect to RESET_N with 10 kΩ series resistor	PU	
E21	E22	SRST_N	I(S)	active-low system reset: controlled by JTAG port	PU	
F20	F22	JTAG_TCK	I(S)	test clock	PD	
F21	G22	JTAG_TDI	I(S)	test data input	PD	
E22	E23	JTAG_TDO	O(8)	test data output		0
F22	F23	JTAG_TMS	I(S)	test mode select	PD	

Table 5-8 Embedded Trace Macrocell (ETM)

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
G20	G21	ETM_CLK	O(12)			0
L20	M21	ETM_CTL	O(8)			1
K18	K23	ETM_DATA_00	O(8)	ETM read data		0
K20	L20	ETM_DATA_01	O(8)			0
K19	K21	ETM_DATA_02	O(8)			0
J20	J21	ETM_DATA_03	O(8)			0
H19	H22	ETM_DATA_04	O(8)			0
H20	H23	ETM_DATA_05	O(8)			0
G21	H20	ETM_DATA_06	O(8)			0
G22	H21	ETM_DATA_07	O(8)			0
L22	L21	ETM_DATA_08	O(8)			0
L21	M23	ETM_DATA_09	O(8)			0
K22	M22	ETM_DATA_10	O(8)			0
K21	L23	ETM_DATA_11	O(8)			0
J22	L22	ETM_DATA_12	O(8)			0
J21	K22	ETM_DATA_13	O(8)			0
H22	J23	ETM_DATA_14	O(8)			0
H21	J22	ETM_DATA_15	O(8)			0

Table 5-9 Debug ports

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
E4	A1	DBG_CLK	Oe	PPU trace clock (leave unconnected)		Z
A2	A2	DBG_STB	Oe	PPU trace strobe (leave unconnected)		Z

5.2.5 DDR2-SDRAM Interface

Table 5-10 SSTL_18, 1.8 V

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
Y18	AC19	DDR_A_00	OUT	address bus		X
AA17	AB17	DDR_A_01	OUT			X
Y17	AA17	DDR_A_02	OUT			X
AB17	AC17	DDR_A_03	OUT			X
AB16	Y16	DDR_A_04	OUT			X
Y16	AC16	DDR_A_05	OUT			X
AA16	AA16	DDR_A_06	OUT			X
AB15	AB16	DDR_A_07	OUT			X
W15	Y15	DDR_A_08	OUT			X
AB14	AC15	DDR_A_09	OUT			X
AB19	AB19	DDR_A_10	OUT			X
Y15	AA15	DDR_A_11	OUT			X
AA15	AB15	DDR_A_12	OUT			X
AB22	AC22	DDR_A_13	OUT			X
AA14	AA14	DDR_A_14	OUT			X
AA19	AB20	DDR_BA_0	OUT	bank address bus		X
W17	AC20	DDR_BA_1	OUT			X
W19	AC21	DDR_CAS_N	OUT	active-low column address strobe		X
AB18	AC18	DDR _CLK	OUT	clock		X
AA18	AB18	DDR_CLK_N	OUT	clock (reversed)		X
Y14	AB14	DDR_CKE	OUT	clock enable control signal		0
AB20	Y19	DDR_CS_N	OUT	active-low chip enable		X
Y22	AA23	DDR_LDM	OUT	input data mask for lower byte		Z
T22	T21	DDR_UDM	OUT	input data mask for upper byte		Z
W20	AB23	DDR_DQ_0	INOUT	data bus		Z
Y21	AA22	DDR_DQ_01	INOUT			Z
U19	W22	DDR_DQ_02	INOUT			Z
U21	W23	DDR_DQ_03	INOUT			Z
AA21	Y20	DDR_DQ_04	INOUT			Z
AA22	AC23	DDR_DQ_05	INOUT			Z
V21	Y21	DDR_DQ_06	INOUT			Z
V22	W21	DDR_DQ_07	INOUT			Z
T19	V23	DDR_DQ_08	INOUT			Z
T21	U21	DDR_DQ_09	INOUT			Z
P21	R21	DDR_DQ_10	INOUT			Z
P22	R23	DDR_DQ_11	INOUT			Z
U22	V21	DDR_DQ_12	INOUT			Z
T20	V22	DDR_DQ_13	INOUT			Z
R20	T22	DDR_DQ_14	INOUT			Z
P20	T23	DDR_DQ_15	INOUT			Z
W22	Y23	DDR_LDQS	INOUT	data strobe for lower byte		Z
R22	U23	DDR_UDQS	INOUT	data strobe for upper byte		Z
W21	Y22	DDR_LDQS_N	INOUT	data strobe for lower byte (reversed)		Z
R21	U22	DDR_UDQS_N	INOUT	data strobe for upper byte (reversed)		Z

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
AB21	AA20	DDR_ODT	OUT	on-die termination control signal		0
W18	AA19	DDR_RAS_N	OUT	active-low row address strobe		X
AA20	AB21	DDR_WE_N	OUT	active-low write enable		X
V19	W20	VREF_SSTL18_0	A	reference voltage for the receivers		
P18	T20	VREF_SSTL18_1	A	reference voltage for the receivers		

5.2.6 General Purpose I/Os

Table 5-11 General Purpose I/Os (GPIO)

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
AA12	AC12	GPIO_00	IOe(S8)	special bootstrap pin: select UART-boot option	PU	H
AB12	AC13	GPIO_01	IOe(S8)		PU	H
AB11	AC11	GPIO_02	IOe(S8)		PU	H
Y13	AA13	GPIO_03	IOe(S8)		PU	H
W12	AB12	GPIO_04	IOe(S8)	I²C 1: serial clock (if Port F V2 used)	PU	H
Y12	AA12	GPIO_05	IOe(S8)	I²C 1: serial data (if Port F V2 used)	PU	H

Table 5-12 Real-Time General Purpose I/Os (RTGPIO) of Ethernet Switch

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
A8	A9	RT_GPIO_0	IOe(S8)		PD	L
B8	B9	RT_GPIO_1	IOe(S8)		PD	L
D9	A10	RT_GPIO_2	IOe(S8)		PD	L
C8	A8	RT_GPIO_3	IOe(S8)		PD	L
A9	B10	RT_GPIO_4	IOe(S8)		PD	L
B9	D10	RT_GPIO_5	IOe(S8)		PD	L
E10	C10	RT_GPIO_6	IOe(S8)		PD	L
C9	C9	RT_GPIO_7	IOe(S8)		PD	L

5.2.7 Ethernet

The table below shows the pin connection for a third Ethernet port with up to 100 Mbit/s transfer rate. This port could also be used for Gigabit Ethernet, if the additional lines at port B are used. Therefore port B has to be configured with version 1 or 2 inside the pin controller.

Table 5-13 (Gigabit) Ethernet (G)MII

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
D8	B8	GMAC_CLK_IN125	I(S)	clock input 125 MHz for Gigabit Ethernet	PD	
A4	C6	GMAC_COL	I(S)	collision detect	PD	
B3	B3	GMAC_CRS	I(S)	carrier sense	PD	
C6	B6	GMAC_MDC	O(8)	clock of PHY management		0
B6	D7	GMAC_MDIO	IOe(S8)	data input/output of PHY management	PD	L
D7	C7	GMAC_RX_CK	I(S)	GMII/RGMII receive clock	PD	
C5	B5	GMAC_RX_DV	I(S)	receive data valid for the GMII/MII modes	PD	
A5	A6	GMAC_RX_ER	I(S)	receive error	PD	
C7	A7	GMAC_RXD_0	I(S)	receive data	PD	
B5	D6	GMAC_RXD_1	I(S)		PD	
A7	C8	GMAC_RXD_2	I(S)		PD	
B7	D8	GMAC_RXD_3	I(S)		PD	
A6	B7	GMAC_TX_CK	I(S)	MII transmit clock	PD	
E6	A4	GMAC_TX_EN	O(8)	transmit enable for the GMII/MII modes		0
B4	C5	GMAC_TX_ER	O(8)	transmit error		0
D5	A3	GMAC_TXD_0	O(8)	transmit data		0
C4	D5	GMAC_TXD_1	O(8)			0
A3	B4	GMAC_TXD_2	O(8)			0
D6	A5	GMAC_TXD_3	O(8)			0

Table 5-14 Ethernet Switch Interface (10Base-T, 100Base-TX/FX)

Ball 380	Ball 385	Pin Name	Type	Description		Pull Up / Pull Down	Reset State
W6	Y5	PHY1_BIAS	Analog OUT	PHY1	off-chip bias resistor (connect a resistor of 12.3 kΩ ±1% to the PCB analog ground)		0
AB4	AC4	PHY1_RX_N	PHY IN		differential received signal pair		
AA4	AB4	PHY1_RX_P	PHY IN				
Y5	Y4	PHY1_SD	Analog IN		fiber mode signal detect pin (fiber mode: connect to the fiber transceiver)		
AB3	AC3	PHY1_TX_N	PHY OUT		differential transmitted signal pair		0
AA3	AB3	PHY1_TX_P	PHY OUT				0
W7	Y6	PHY2_BIAS	Analog OUT	PHY2	off-chip bias resistor (connect a resistor of 12.3 kΩ ±1% to the PCB analog ground)		0
AB6	AC6	PHY2_RX_N	PHY IN		differential received signal pair		
AA6	AB6	PHY2_RX_P	PHY IN				
W8	Y7	PHY2_SD	Analog IN		fiber mode signal detect pin (fiber mode: connect to the fiber transceiver)		
AB7	AC7	PHY2_TX_N	PHY OUT		differential transmitted signal pair		0
AA7	AB7	PHY2_TX_P	PHY OUT				0

5.2.8 Serial Interfaces

Table 5-15 SliceBus

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
Y9	AA9	SBUS_ALARM_N	I(S)	alarm line (level shifter required)	PU	
W9	Y9	SBUS_MDLO	O(8)	master data line output (LVTTL to LVDS driver required)		0
AA9	AB9	SBUS_NDLI	I(S)	master data line input (LVTTL to LVDS driver required)	PD	

Table 5-16 SPI

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
N21	R22	SPI_CLK	O(8)	serial clock output (master mode)		0
N22	P20	SPI_CLKIN	I(S)	serial clock input (slave mode)	PD	
M19	N21	SPI_CS_0_N	IO(S8)	chip select 0 (master mode: out; slave mode: in)	PU	1
N19	P22	SPI_CS_1_N	O(8)	chip select 1 (only used at master mode)		1
M21	P23	SPI_RXD	I(S)	receive data	PD	
N20	P21	SPI_TXD	Oe(8)	transmit data		Z

Table 5-17 UART1 (Bootstrap UART)

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
L18	N23	UART1_CTS_N	I(S)	active-low clear to send	PU	
L19	M20	UART1_RTS_N	O(8)	active-low request to send		1
M20	N22	UART1_RXD	I(S)	receive data	PD	
M22	N20	UART1_TXD	O(8)	transmit data		1

Table 5-18 USB 2.0 Device Controller

Ball 380	Ball 385	Pin Name	Type	Description	Pull Up / Pull Down	Reset State
K1	J1	USB_DN	USB	USB D- line		0
K2	J2	USB_DP	USB	USB D+ line		0

The signal USB_INT is available on port B version 3 and indicates if a cable is connected. If USB port is not used, USB_DN and USB_DP can be left open (floating). In this scenario set bit GOSUSP in Main Control Register of USB controller to enable power saving.

5.2.9 PinCtrl-Ports

Due to a limited number of available signal pins some interfaces are sharing the same physical pins and some restrictions apply regarding the concurrent usability of interfaces. The pin sharing matrix can be configured by a register set inside the integrated pin controller.

Table 5-19 PinCtrl Configurations

Port	V1	V2	V3	V4	V5
A	NAND Flash	SD/MMC QuadSPI			
B	MII1 MII2 GPIO[7:6]	GMII TechIO output[3:0] TechIO input[0:11] GPIO[15:6]	GMII TechIO output[3:0] TechIO input[0:5] USB IRQ GPIO[14:6]	TechIO output[12:0] TechIO input[0:19] TechIO output[15:13]	
C	VPC I²C 2 GPIO[16]	UART 2 I²C 2 GPIO[17:16]	CAN 1 CAN 2 I²C 2 GPIO[17:16]	PBM 1 PBM 2	GPIO[23:16]
D	AEI Master	AEI Slave	GPIO[16:31] CAN 1 CAN 2 VPC UART 2	TechIO output[0:19] TechIO input[0:24]	
E	PPU GPIO[15:0]	Debug: - ETM[16:31]	TechIO output[16:19] TechIO input[20:25] SSI1 SSI2		
F	GPIO[4:5]	I²C 1			

All port pins are bidirectional pins. Before the ports are configured all pins are switched to tristate mode.

Table 5-20 Port A

Port A								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
A10	A11	PORT_A_00	NAND_BUSY_N	V1	I(S)	NAND Flash: active-low busy signal	PU	H
			SD_MMC_CD_N	V2	I(S)	SD/MMC: active-low card detect		
B10	B11	PORT_A_01	NAND_WP_N	V1	O(8)	NAND Flash: active-low write protect	PU	H
			SD_MMC_WP	V2	I(S)	SD/MMC: card write protect		
D10	C11	PORT_A_02	NAND_WE_N	V1	O(8)	NAND Flash: active-low write enable	PU	H
			SD_MMC_CLK	V2	O(8)	SD/MMC: serial clock		
C10	D11	PORT_A_03	NAND_RE_N	V1	O(8)	NAND Flash: active-low read enable	PU	H
			SD_MMC_DATA_0	V2	IOe(S8)	SD/MMC: IO data[0]		
A11	A12	PORT_A_04	NAND_CE_N	V1	O(8)	NAND Flash: active-low chip enable	PU	H
			SD_DATA_1	V2	IOe(S8)	SD/MMC: IO data[1]		
B11	B12	PORT_A_05	NAND_CLE	V1	O(8)	NAND Flash: command latch enable	PU	H
			SD_DATA_2	V2	IOe(S8)	SD/MMC: IO data[2]		
D11	C12	PORT_A_06	NAND_ALE	V1	O(8)	NAND Flash: address latch enable	PU	H
			SD_DATA_3	V2	IOe(S8)	SD/MMC: IO data[3]		
C11	D12	PORT_A_07	NAND_DATA_0	V1	IOe(S8)	NAND Flash: IO data[0]	PU	H
			SD_MMC_COM	V2	IOe(S8)	SD/MMC: command line		

Port A								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
A12	C13	PORT_A_08	NAND_DATA_1	V1	IOe(S12)	NAND Flash: IO data[1]	PU	H
			QUADSPI_CS_0_N	V2	O(12)	QuadSPI: active-low chip select 0		
B12	D13	PORT_A_09	NAND_DATA_2	V1	IOe(S12)	NAND Flash: IO data[2]	PU	H
			QUADSPI_CS_1_N	V2	O(12)	QuadSPI: active-low chip select 1		
E11	A13	PORT_A_10	NAND_DATA_3	V1	IOe(S12)	NAND Flash: IO data[3]	PU	H
			AUADSPI_SCK_OUT	V2	O(12)	QuadSPI: serial clock		
C12	B13	PORT_A_11	NAND_DATA_4	V1	IOe(S12)	NAND Flash: IO data[4]	PU	H
			QUADSPI_TX	V2	IOe(S12)	QuadSPI: transmit data / IO data[0]		
A13	A14	PORT_A_12	NAND_DATA_5	V1	IOe(S12)	NAND Flash: IO data[5]	PU	H
			QUADSPI_RX	V2	IOe(S12)	QuadSPI: receive data / IO data[1]		
D12	B14	PORT_A_13	NAND_DATA_6	V1	IOe(S12)	NAND Flash: IO data[6]	PU	H
			QUADSPI_WP_N	V2	IOe(S12)	QuadSPI: IO data[2]		
C13	C14	PORT_A_14	NAND_DATA_7	V1	IOe(S12)	NAND Flash: IO data[7]	PU	H
			QUADSPI_HOLD_N	V2	IOe(S12)	QuadSPI: IO data[3]		

Table 5-21 Port B

Port B								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Down**	**Reset State**
B13	B15	PORT_B_00	MII1_TX_CK	V1	I(S)	MII 1: transmit clock	PU	H
			GMII_GTXCLK	V2	O(8)	GMII: transmit clock		
				V3				
			TECH_OUT_12	V4	O(8)	TechIO: output data[12]		
A14	A16	PORT_B_01	MII1_RXD_0	V1	I(S)	MII 1: receive data[0]	PU	H
			GMII_RXD_4	V2	I(S)	GMII: receive data[4]		
				V3				
			TECH_OUT_11	V4	O(8)	TechIO: output data[11]		
E12	A15	PORT_B_02	MII1_RXD_1	V1	I(S)	MII 1: receive data[1]	PU	H
			GMII_RXD_5	V2	I(S)	GMII: receive data[5]		
				V3				
			TECH_OUT_10	V4	O(8)	TechIO: output data[10]		
C14	B16	PORT_B_03	MII1_RXD_2	V1	I(S)	MII 1: receive data[2]	PU	H
			GMII_RXD_6	V2	I(S)	GMII: receive data[6]		
				V3				
			TECH_OUT_09	V4	O(8)	TechIO: output data[9]		
B14	A17	PORT_B_04	MII1_RXD_3	V1	I(S)	MII 1: receive data[3]	PU	H
			GMII_RXD_7	V2	I(S)	GMII: receive data[7]		
				V3				
			TECH_OUT_08	V4	O(8)	TechIO: output data[8]		
A15	B17	PORT_B_05	MII1_TXD_3	V1	O(8)	MII 1: transmit data[3]	PU	H
			GMII_TXD_7	V2	O(8)	GMII: transmit data[7]		
				V3				
			TECH_OUT_07	V4	O(8)	TechIO: output data[7]		

Port B								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
D13	C15	PORT_B_06	MII1_TXD_2	V1	O(8)	MII 1: transmit data[2]	PU	H
			GMII_TXD_6	V2, V3	O(8)	GMII: transmit data[6]		
			TECH_OUT_06	V4	O(8)	TechIO: output data[6]		
C15	A18	PORT_B_07	MII1_TXD_1	V1	O(8)	MII 1: transmit data[1]	PU	H
			GMII_TXD_5	V2, V3	O(8)	GMII: transmit data[5]		
			TECH_OUT_05	V4	O(8)	TechIO: output data[5]		
B15	C16	PORT_B_08	MII1_TXD_0	V1	O(8)	MII 1: transmit data[0]	PU	H
			GMII_TXD_4	V2, V3	O(8)	GMII: transmit data[4]		
			TECH_OUT_04	V4	O(8)	TechIO: output data[4]		
A16	B18	PORT_B_09	MII1_TX_ER	V1	O(8)	MII 1: transmit error	PU	H
			TECH_OUT_03	V2, V3, V4	O(8)	TechIO: output data[3]		
D14	D15	PORT_B_10	MII1_TX_EN	V1	O(8)	MII 1: transmit enable	PU	H
			TECH_OUT_02	V2, V3, V4	O(8)	TechIO: output data[2]		
C16	A19	PORT_B_11	MII1_RX_ER	V1	I(S)	MII 1: receive error detected	PU	H
			TECH_OUT_01	V2, V3, V4	O(8)	TechIO: output data[1]		

Port B

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
B16	C17	PORT_B_12	MII1_RX_CK	V1	I(S)	MII 1: receive clock	PU	H
			TECH_OUT_00	V2, V3, V4	O(8)	TechIO: output data[0]		
A17	D17	PORT_B_13	MII1_RX_DV	V1	I(S)	MII 1: receive data valid	PU	H
			TECH_IN_00	V2, V3, V4	I(S)	TechIO: input data[0]		
D15	D16	PORT_B_14	MII1_CRS	V1	I(S)	MII 1: carrier sense	PU	H
			TECH_IN_01	V2, V3, V4	I(S)	TechIO: input data[1]		
C17	B19	PORT_B_15	MII1_COL	V1	I(S)	MII 1: collision detect	PU	H
			TECH_IN_02	V2, V3, V4	I(S)	TechIO: input data[2]		
B17	C18	PORT_B_16	MII2_TX_CK	V1	I(S)	MII 2: transmit clock	PU	H
			TECH_IN_03	V2, V3, V4	I(S)	TechIO: input data[3]		
A18	D18	PORT_B_17	MII2_RXD_0	V1	I(S)	MII 2: receive data[0]	PU	H
			TECH_IN_04	V2, V3, V4	I(S)	TechIO: input data[4]		

Port B								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Down**	**Reset State**
D16	A20	PORT_B_18	MII2_RXD_1	V1	I(S)	MII 2: receive data[1]	PU	H
			TECH_IN_05	V2	I(S)	TechIO: input data[5]		
				V3				
				V4				
C18	B20	PORT_B_19	MII2_RXD_2	V1	I(S)	MII 2: receive data[2]	PU	H
			TECH_IN_06	V2	I(S)	TechIO: input data[6]		
				V3		reserved (leave unconnected)		
			TECH_IN_06	V4	I(S)	TechIO: input data[6]		
B18	C19	PORT_B_20	MII2_RXD_3	V1	I(S)	MII 2: receive data[3]	PU	H
			TECH_IN_07	V2	I(S)	TechIO: input data[7]		
			USB_INT	V3	I(S)	USB: cable connected		
			TECH_IN_07	V4	I(S)	TechIO: input data[7]		
A19	A21	PORT_B_21	MII2_TXD_3	V1	O(8)	MII 2: transmit data[3]	PU	H
			TECH_IN_08	V2	I(S)	TechIO: input data[8]		
				V3		reserved (leave unconnected)		
			TECH_IN_08	V4	I(S)	TechIO: input data[8]		
C19	C20	PORT_B_22	MII2_TXD_2	V1	O(8)	MII 2: transmit data[2]	PU	H
			TECH_IN_09	V2	I(S)	TechIO: input data[9]		
				V3		reserved (leave unconnected)		
			TECH_IN_09	V4	I(S)	TechIO: input data[9]		
C20	C22	PORT_B_23	MII2_TXD_1	V1	O(8)	MII 2: transmit data[1]	PU	H
			TECH_IN_10	V2	I(S)	TechIO: input data[10]		
				V3		reserved (leave unconnected)		
			TECH_IN_10	V4	I(S)	TechIO: input data[10]		

Port B								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Down**	**Reset State**
B19	B21	PORT_B_24	M22_TXD_0	V1	O(8)	MII 2: transmit data[0]	PU	H
			TECH_IN_11	V2	I(S)	TechIO: input data[11]		
				V3		reserved (leave unconnected)		
			TECH_IN_11	V4	I(S)	TechIO: input data[11]		
A20	A22	PORT_B_25	MII2_TX_ER	V1	O(8)	MII 2: tansmit error	PU	H
				V2		reserved (leave unconnected)		
				V3				
			TECH_IN_12	V4	I(S)	TechIO: input data[12]		
D20	D21	PORT_B_26	MII2_TX_EN	V1	O(8)	MII 2: transmit enable	PU	H
			GPIO_15	V2	IOe(S8)	GPIO: I/O data[15]		
				V3		reserved (leave unconnected)		
			TECH_IN_13	V4	I(S)	TechIO: input data[13]		
C21	E21	PORT_B_27	MII2_RX_ER	V1	I(S)	MII 2: receive error detected	PU	H
			GPIO_14	V2	IOe(S8)	GPIO: I/O data[14]		
				V3				
			TECH_IN_14	V4	I(S)	TechIO: input data[14]		
B20	B22	PORT_B_28	MII2_RX_CK	V1	I(S)	MII 2: receive clock	PU	H
			GPIO_13	V2	IOe(S8)	GPIO: I/O data[13]		
				V3				
			TECH_IN_15	V4	I(S)	TechIO: input data[15]		
A21	B23	PORT_B_29	MII_RX_DV	V1	I(S)	MII 2: receive data valid	PU	H
			GPIO_12	V2	IOe(S8)	GPIO: I/O data[12]		
				V3				
			TECH_IN_16	V4	I(S)	TechIO: input data[16]		

Port B								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Down**	**Reset State**
D19	A23	PORT_B_30	MII2_CRS	V1	I(S)	MII 2: carrier sense	PU	H
			GPIO_11	V2 V3	IOe(S8)	GPIO: I/O data[11]		
			TECH_IN_17	V4	I(S)	TechIO: input data[17]		
B21	C21	PORT_B_31	MII2_COL	V1	I(S)	MII 2: collision detect	PU	H
			GPIO_10	V2 V3	IOe(S8)	GPIO: I/O data[10]		
			TECH_IN_18	V4	I(S)	TechIO: input data[18]		
B22	F20	PORT_B_32	MII_MDC	V1	O(8)	MII 2: management data clock	PU	H
			GPIO_09	V2 V3	IOe(S8)	GPIO: I/O data[9]		
			TECH_IN_19	V4	I(S)	TechIO: input data[19]		
E19	C23	PORT_B_33	MII_MDIO	V1	IOe(S8)	MII 2: management data I/O	PU	H
			GPIO_08	V2 V3	IOe(S8)	GPIO I/O data[8]		
			TECH_OUT_15	V4	O(8)	TechIO: output data[15]		
C22	D23	PORT_B_34	GPIO_07	V1	IOe(S8)	MII 2: link done (used as IN port)	PU	H
				V2 V3		GPIO: I/O data[7]		
			TECH_OUT_14	V4	O(8)	TechIO: output data[14]		
D21	D22	PORT_B_35	GPIO_06	V1	IOe(S8)	MII 1: link done (used as IN port)	PU	H
				V2 V3		GPIO: I/O data[6]		
			TECH_OUT_13	V4	O(8)	TechIO: output data[13]		

Table 5-22 Port C

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
AA11	AB11	PORT_C_00	VPC_RXD	V1	I(S)	PROFIBUS slave: receive data	PU	H
			UART2_RXD	V2	I(S)	UART 2: receive data		
			CAN1_RX	V3	I(S)	CAN 1: receive data		
			PBM1_RXD	V4	I(S)	PROFIBUS master 1: receive data		
			GPIO_23	V5	IOe(S8)	GPIO: I/O data[23]		
AB10	AC10	PORT_C_01	VPC_CTS_N	V1	I(S)	PROFIBUS slave: active-low clear to send	PU	H
			UART2_CTS_N	V2	I(S)	UART 2: active-low clear to send		
			CAN1_TX	V3	O(8)	CAN 1: transmit data		
			PBM1_CTS_N	V4	I(S)	PROFIBUS master 1: active-low clear to send		
			GPIO_22	V5	IOe(S8)	GPIO: I/O data[22]		
AA10	AB10	PORT_C_02	VPC_RTS	V1	O(8)	PROFIBUS slave: request to send	PU	H
			UART2_RTS_N	V2	O(8)	UART 2: active-low request to send		
			CAN2_RX	V3	I(S)	CAN 2: receive data		
			PBM1_RTS	V4	O(8)	PROFIBUS master 1: request to send		
			GPIO_21	V5	IOe(S8)	GPIO: I/O data[21]		
AB9	AC9	PORT_C_03	VPC_TXD	V1	Oe(8)	PROFIBUS slave: transmit data	PU	H
			UART2_TXD	V2	O(8)	UART 2: transmit data		
			CAN2_TX	V3	O(8)	CAN 2: transmit data		
			PBM1_TXD	V4	Oe(8)	PROFIBUS master 1: transmit data		
			GPIO_20	V5	IOe(S8)	GPIO: I/O data[20]		
Y11	AA11	PORT_C_04	IIC2_SCL	V1	IOe(S8)	I²C 2: serial clock	PU	H
			IIC2_SCL	V2	IOe(S8)	I²C 2: serial clock		
			IIC2_SCL	V3	IOe(S8)	I²C 2: serial clock		
			PBM2_RXD	V4	I(S)	PROFIBUS master 2: receive data		
			GPIO_19	V5	IOe(S8)	GPIO: I/O data[19]		

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
						Port C		
W11	Y11	PORT_C_05	IIC2_SDA	V1	IOe(S8)	I²C 2: serial data	PU	H
			IIC2_SDA	V2	IOe(S8)	I²C 2: serial data		
			IIC2_SDA	V3	IOe(S8)	I²C 2: serial data		
			PBM2_CTS_N	V4	I(S)	PROFIBUS master 2: active-low clear to send		
			GPIO_18	V5	IOe(S8)	GPIO: I/O data[18]		
Y10	AA10	PORT_C_06	VPC_DATAEXCH_N	V1	O(8)	PROFIBUS slave: indicate state DATA-EXCH	PU	H
			GPIO_17	V2	IOe(S8)	GPIO: I/O data[17]		
			GPIO_17	V3	IOe(S8)	GPIO: I/O data[17]		
			PBM2_RTS	V4	O(8)	PROFIBUS master 2: request to send		
			GPIO_17	V5	IOe(S8)	GPIO: I/O data[17]		
W10	Y10	PORT_C_07	GPIO_16	V1	IOe(S8)	GPIO: I/O data[16]	PU	H
			GPIO_16	V2	IOe(S8)	GPIO: I/O data[16]		
			GPIO_16	V3	IOe(S8)	GPIO: I/O data[16]		
			PBM2_TXD	V4	Oe(8)	PROFIBUS master 2: transmit data		
			GPIO_16	V5	IOe(S8)	GPIO: I/O data[16]		

Table 5-23 Port D

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
G1	F2	PORT_D_00	AEI_M_D_15	V1	IOe(S8)	AEI master: data[15]	PU	H
			AEI_S_D_15	V2	IOe(S8)	AEI slave: data[15]		
			GPIO_16	V3	IOe(S8)	GPIO: I/O data[16]		
			TECH_OUT_00	V4	O(8)	TechIO: output data[0]		
H2	K3	PORT_D_01	AEI_M_D_14	V1	IOe(S8)	AEI master: data[14]	PU	H
			AEI_S_D_14	V2	IOe(S8)	AEI slave: data[14]		
			GPIO_17	V3	IOe(S8)	GPIO: I/O data[17]		
			TECH_OUT_01	V4	O(8)	TechIO: output data[1]		
H4	G2	PORT_D_02	AEI_M_D_13	V1	IOe(S8)	AEI master: data[13]	PU	H
			AEI_S_D_13	V2	IOe(S8)	AEI slave: data[13]		
			GPIO_18	V3	IOe(S8)	GPIO: I/O data[18]		
			TECH_OUT_02	V4	O(8)	TechIO: output data[2]		
G3	H3	PORT_D_03	AEI_M_D_12	V1	IOe(S8)	AEI master: data[12]	PU	H
			AEI_S_D_12	V2	IOe(S8)	AEI slave: data[12]		
			GPIO_19	V3	IOe(S8)	GPIO: I/O data[19]		
			TECH_OUT_03	V4	O(8)	TechIO: output data[3]		
H1	G1	PORT_D_04	AEI_M_D_11	V1	IOe(S8)	AEI master: data[11]	PU	H
			AEI_S_D_11	V2	IOe(S8)	AEI slave: data[11]		
			GPIO_20	V3	IOe(S8)	GPIO: I/O data[20]		
			TECH_OUT_04	V4	O(8)	TechIO: output data[4]		
J3	L4	PORT_D_05	AEI_M_D_10	V1	IOe(S8)	AEI master: data[10]	PU	H
			AEI_S_D_10	V2	IOe(S8)	AEI slave: data[10]		
			GPIO_21	V3	IOe(S8)	GPIO: I/O data[21]		
			TECH_OUT_05	V4	O(8)	TechIO: output data[5]		

Port D

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
J5	L3	PORT_D_06	AEI_M_D_09	V1	IOe(S8)	AEI master: data[9]	PU	H
			AEI_S_D_09	V2	IOe(S8)	AEI slave: data[9]		
			GPIO_22	V3	IOe(S8)	GPIO: I/O data[22]		
			TECH_OUT_06	V4	O(8)	TechIO: output data[6]		
H3	J4	PORT_D_07	AEI_M_D_08	V1	IOe(S8)	AEI master: data[8]	PU	H
			AEI_S_D_08	V2	IOe(S8)	AEI slave: data[8]		
			GPIO_23	V3	IOe(S8)	GPIO: I/O data[23]		
			TECH_OUT_07	V4	O(8)	TechIO: output data[7]		
L5	L2	PORT_D_08	AEI_M_D_07	V1	IOe(S8)	AEI master: data[7]	PU	H
			AEI_S_D_07	V2	IOe(S8)	AEI slave: data[7]		
			GPIO_24	V3	IOe(S8)	GPIO: I/O data[24]		
			TECH_OUT_08	V4	O(8)	TechIO: output data[8]		
M2	M4	PORT_D_09	AEI_M_D_06	V1	IOe(S8)	AEI master: data[6]	PU	H
			AEI_S_D_06	V2	IOe(S8)	AEI slave: data[6]		
			GPIO_25	V3	IOe(S8)	GPIO: I/O data[25]		
			TECH_OUT_09	V4	O(8)	TechIO: output data[9]		
J4	F1	PORT_D_10	AEI_M_D_05	V1	IOe(S8)	AEI master: data[5]	PU	H
			AEI_S_D_05	V2	IOe(S8)	AEI slave: data[5]		
			GPIO_26	V3	IOe(S8)	GPIO: I/O data[26]		
			TECH_OUT_10	V4	O(8)	TechIO: output data[10]		
L4	L1	PORT_D_11	AEI_M_D_04	V1	IOe(S8)	AEI master: data[4]	PU	H
			AEI_S_D_04	V2	IOe(S8)	AEI slave: data[4]		
			GPIO_27	V3	IOe(S8)	GPIO: I/O data[27]		
			TECH_OUT_11	V4	O(8)	TechIO: output data[11]		

Port D

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
M1	M3	PORT_D_12	AEI_M_D_03	V1	IOe(S8)	AEI master: data[3]	PU	H
			AEI_S_D_03	V2	IOe(S8)	AEI slave: data[3]		
			GPIO_28	V3	IOe(S8)	GPIO: I/O data[28]		
			TECH_OUT_12	V4	O(8)	TechIO: output data[12]		
N2	M2	PORT_D_13	AEI_M_D_02	V1	IOe(S8)	AEI master: data[2]	PU	H
			AEI_S_D_02	V2	IOe(S8)	AEI slave: data[2]		
			GPIO_29	V3	IOe(S8)	GPIO: I/O data[29]		
			TECH_OUT_13	V4	O(8)	TechIO: output data[13]		
M5	M1	PORT_D_14	AEI_M_D_01	V1	IOe(S8)	AEI master: data[1]	PU	H
			AEI_S_D_01	V2	IOe(S8)	AEI slave: data[1]		
			GPIO_30	V3	IOe(S8)	GPIO: I/O data[30]		
			TECH_OUT_14	V4	O(8)	TechIO: output data[14]		
M3	N1	PORT_D_15	AEI_M_D_00	V1	IOe(S8)	AEI master: data[0]	PU	H
			AEI_S_D_00	V2	IOe(S8)	AEI slave: data[0]		
			GPIO_31	V3	IOe(S8)	GPIO: I/O data[31]		
			TECH_OUT_15	V4	O(8)	TechIO: output data[15]		
N1	N2	PORT_D_16	AEI_M_A_00	V1	IOe(S8)	AEI master: address[0]	PU	H
			AEI_S_A_00	V2	I(S)	AEI slave: address[0]		
			CAN1_RX	V3	I(S)	CAN 1: receive data		
			TECH_OUT_16	V4	O(8)	TechIO: output data[16]		
P2	N4	PORT_D_17	AEI_M_A_01	V1	O(8)	AEI master: address[1]	PU	H
			AEI_S_A_01	V2	I(S)	AEI slave: address[1]		
			CAN1_TX	V3	O(8)	CAN 1: transmit data		
			TECH_OUT_17	V4	O(8)	TechIO: output data[17]		

Port D								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
M4	N3	PORT_D_18	AEI_M_A_02	V1	O(8)	AEI master: address[2]	PU	H
			AEI_S_A_02	V2	I(S)	AEI slave: address[2]		
			CAN2_RX	V3	I(S)	CAN 2: receive data		
			TECH_OUT_18	V4	O(8)	TechIO: output data[18]		
N3	P1	PORT_D_19	AEI_M_A_03	V1	O(8)	AEI master: address[3]	PU	H
			AEI_S_A_03	V2	I(S)	AEI slave: address[3]		
			CAN2_TX	V3	O(8)	CAN 2: transmit data		
			TECH_OUT_19	V4	O(8)	TechIO: output data[19]		
P1	P2	PORT_D_20	AEI_M_A_04	V1	O(8)	AEI master: address[4]	PU	H
			AEI_S_A_04	V2	I(S)	AEI slave: address[4]		
			VPC_RXD	V3	I(S)	PROFIBUS slave: receive data		
			TECH-IN_00	V4	I(S)	TechIO: input data[0]		
R2	R1	PORT_D_21	AEI_M_A_05	V1	O(8)	AEI master: address[5]	PU	H
			AEI_S_A_05	V2	I(S)	AEI slave: address[5]		
			VPC_CTS_N	V3	I(S)	PROFIBUS slave: active-low clear to send		
			TECH-IN_01	V4	I(S)	TechIO: input data[1]		
N4	R2	PORT_D_22	AEI_M_A_06	V1	O(8)	AEI master: address[6]	PU	H
			AEI_S_A_06	V2	I(S)	AEI slave: address[6]		
			VPC_RTS	V3	O(8)	PROFIBUS slave: request to send		
			TECH_IN_02	V4	I(S)	TechIO: input data[2]		
P3	R3	PORT_D_23	AEI_M_A_07	V1	O(8)	AEI master: address[7]	PU	H
			AEI_S_A_07	V2	I(S)	AEI slave: address[7]		
			VPC_DATAEXCH_N	V3	O(8)	PROFIBUS slave: indicate state DATA-EXCH		
			TECH_IN_03	V4	I(S)	TechIO: input data[3]		

Port D								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
R1	T1	PORT_D_24	AEI_M_A_08	V1	O(8)	AEI master: address[8]	PU	H
			AEI_S_A_08	V2	I(S)	AEI slave: address[8]		
			VPC_TXD	V3	Oe(8)	PROFIBUS slave: transmit data		
			TECH_IN_04	V4	I(S)	TechIO: input data[4]		
T2	T2	PORT_D_25	AEI_M_A_09	V1	O(8)	AEI master: address[9]	PU	H
			AEI_S_A_09	V2	I(S)	AEI slave: address[9]		
			UART2_RXD	V3	I(S)	UART 2: receive data		
			TECH_IN_05	V4	I(S)	TechIO: input data[5]		
P4	T3	PORT_D_26	AEI_M_A_10	V1	O(8)	AEI master: address[10]	PU	H
			AEI_S_A_10	V2	I(S)	AEI slave: address[10]		
			UART2_CTS_N	V3	I(S)	UART 2: active-low clear to send		
			TECH_IN_06	V4	I(S)	TechIO: input data[6]		
R3	T4	PORT_D_27	AEI_M_A_11	V1	O(8)	AEI master: address[11]	PU	H
			AEI_S_A_11	V2	I(S)	AEI slave: address[11]		
			UART2_RTS_N	V3	O(8)	UART 2: active-low request to send		
			TECH_IN_07	V4	I(S)	TechIO: input data[7]		
T1	U1	PORT_D_28	AEI_M_A_12	V1	O(8)	AEI master: address[12]	PU	H
			AEI_S_A_12	V2	I(S)	AEI slave: address[12]		
			UART2_TXD	V3	O(8)	UART 2: transmit data		
			TECH_IN_08	V4	I(S)	TechIO: input data[8]		
U2	U2	PORT_D_29	AEI_M_A_13	V1	O(8)	AEI master: address[13]	PU	H
			AEI_S_A_13	V2	I(S)	AEI slave: address[13]		
				V3		reserved (leave unconnected)		
			TECH_IN_09	V4	I(S)	TechIO: input data[9]		

Port D								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
P5	U3	PORT_D_30	AEI_M_A_14	V1	O(8)	AEI master: address[14]	PU	H
			AEI_S_A_14	V2	I(S)	AEI slave: address[14]		
				V3		reserved (leave unconnected)		
			TECH_IN_10	V4	I(S)	TechIO: input data[10]		
R4	U4	PORT_D_31	AEI_M_A_15	V1	O(8)	AEI master: address[15]	PU	H
			AEI_S_A_15	V2	I(S)	AEI slave: address[15]		
				V3		reserved (leave unconnected)		
			TECH_IN_11	V4	I(S)	TechIO: input data[11]		
U1	V1	PORT_D_32	AEI_M_A_16	V1	O(8)	AEI master: address[16]	PU	H
			AEI_S_A_16	V2	I(S)	AEI slave: address[16]		
				V3		reserved (leave unconnected)		
			TECH_IN_12	V4	I(S)	TechIO: input data[12]		
T4	V2	PORT_D_33	AEI_M_A_17	V1	O(8)	AEI master: address[17]	PU	H
			AEI_S_A_17	V2	I(S)	AEI slave: address[17]		
				V3		reserved (leave unconnected)		
			TECH_IN_13	V4	I(S)	TechIO: input data[13]		
U4	V3	PORT_D_34	AEI_M_A_18	V1	O(8)	AEI master: address[18]	PU	H
			AEI_S_A_18	V2	I(S)	AEI slave: address[18]		
				V3		reserved (leave unconnected)		
			TECH_IN_14	V4	I(S)	TechIO: input data[14]		
V3	V4	PORT_D_35	AEI_M_A_19	V1	O(8)	AEI master: address[19]	PU	H
			AEI_S_A_19	V2	I(S)	AEI slave: address[19]		
				V3		reserved (leave unconnected)		
			TECH_IN_15	V4	I(S)	TechIO: input data[15]		

Port D								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
V1	W1	PORT_D_36	AEI_M_A_20	V1	O(8)	AEI master: address[20]	PU	H
			AEI_S_A_20	V2	I(S)	AEI slave: address[20]		
				V3		reserved (leave unconnected)		
			TECH_IN_16	V4	I(S)	TechIO: input data[16]		
W3	Y1	PORT_D_37	AEI_M_CS_0	V1	O(8)	AEI master: active-low chip select[0]	PU	H
			AEI_S_CS_0_N	V2	I(S)	AEI slave: active-low chip select[0]		
				V3		reserved (leave unconnected)		
			TECH_IN_17	V4	I(S)	TechIO: input data[17]		
W1	W2	PORT_D_38	AEI_M_CS_1	V1	O(8)	AEI master: active-low chip select[1]	PU	H
				V2		reserved (leave unconnected)		
				V3		reserved (leave unconnected)		
			TECH_IN_18	V4	I(S)	TechIO: input data[18]		
W2	W3	PORT_D_39	AEI_M_BS_1_N	V1	O(8)	AEI master: active-low high-byte select	PU	H
			AEI_S_BS_1_N	V2	I(S)	AEI slave: active-low high-byte select		
				V3		reserved (leave unconnected)		
			TECH_IN_19	V4	I(S)	TechIO: input data[19]		
Y1	Y2	PORT_D_40	AEI_M_RE_N	V1	O(8)	AEI master: active-low read enable	PU	H
			AEI_S_RE_N	V2	I(S)	AEI slave: active-low read enable		
				V3		reserved (leave unconnected)		
			TECH_IN_20	V4	I(S)	TechIO: input data[20]		
Y2	AA1	PORT_D_41	AEI_M_WE_N	V1	O(8)	AEI master: active-low write enable	PU	H
			AEI_S_WE_N	V2	I(S)	AEI slave: active-low write enable		
				V3		reserved (leave unconnected)		
			TECH_IN_21	V4	I(S)	TechIO: input data[21]		

Port D								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Down	Reset State
AA1	AA2	PORT_D_42	AEI_M_WAIT	V1	I(S)	AEI master: wait signal	PU	H
			AEI_S_WAIT	V2	O(8)	AEI slave: wait signal		
				V3		reserved (leave unconnected)		
			TECH_IN_22	V4	I(S)	TechIO: input data[22]		
V4	AB1	PORT_D_43	AEI_M_INT_0	V1	I(S)	AEI master: 1st IRQ line, can be used for SNAP+ synchronization	PU	H
			AEI_S_INT_0	V2	O(8)	reserved (leave unconnected)		
				V3		reserved (leave unconnected)		
			TECH_IN_23	V4	I(S)	TechIO: input data[23]		
AB1	AC1	PORT_D_44	AEI_M_INT_1	V1	I(S)	AEI master: 2nd IRQ line	PU	H
			AEI_S_INT_1	V2	O(8)	AEI slave: IRQ line		
				V3		reserved (leave unconnected)		
			TECH_IN_24	V4	I(S)	TechIO: input data[24]		

Table 5-24 Port E

Port E									
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Debug Select**	**Description**	**Pull Up / Down**	**Reset State**
A1	B1	PORT_E_00	PPUGPIO_00	V1	IOe(S8)		Ethernet Switch: I/O data[0]	PD	L
A1	B1	PORT_E_00	DBG_D_00	V2	O(8)	0	ETM: debug data[16]	PD	L
A1	B1	PORT_E_00	TECH_OUT_16	V3	O(8)		TechIO: output data[16]	PD	L
B2	B2	PORT_E_01	PPUGPIO_01	V1	IOe(S8)		Ethernet Switch: I/O data[1]	PD	L
B2	B2	PORT_E_01	DBG_D_01	V2	O(8)	0	ETM: debug data[17]	PD	L
B2	B2	PORT_E_01	TECH_OUT_17	V3	O(8)		TechIO: output data[17]	PD	L
B1	D1	PORT_E_02	PPUGPIO_02	V1	IOe(S8)		Ethernet Switch: I/O data[2]	PD	L
B1	D1	PORT_E_02	DBG_D_02	V2	O(8)	0	ETM: debug data[18]	PD	L
B1	D1	PORT_E_02	TECH_OUT_18	V3	O(8)		TechIO: output data[18]	PD	L
C2	D3	PORT_E_03	PPUGPIO_03	V1	IOe(S8)		Ethernet Switch: I/O data[3]	PD	L
C2	D3	PORT_E_03	DBG_D_03	V2	O(8)	0	ETM: debug data[19]	PD	L
C2	D3	PORT_E_03	TECH_OUT_19	V3	O(8)		TechIO: output data[19]	PD	L
C1	E2	PORT_E_04	PPUGPIO_04	V1	IOe(S8)		Ethernet Switch: I/O data[4]	PD	L
C1	E2	PORT_E_04	DBG_D_04	V2	O(8)	0	ETM: debug data[20]	PD	L
C1	E2	PORT_E_04	TECH_IN_20	V3	I(S)		TechIO: input data[20]	PD	L
D3	C2	PORT_E_05	PPUGPIO_05	V1	IOe(S8)		Ethernet Switch: I/O data[5]	PD	L
D3	C2	PORT_E_05	DBG_D_05	V2	O(8)	0	ETM: debug data[21]	PD	L
D3	C2	PORT_E_05	TECH_IN_21	V3	I(S)		TechIO: input data[21]	PD	L

Port E									
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Debug Select	Description	Pull Up / Down	Reset State
E3	C1	PORT_E_06	PPUGPIO_06	V1	IOe(S8)		Ethernet Switch: I/O data[6]	PD	L
			DBG_D_06	V2	O(8)	0	ETM: debug data[22]		
			TECH_IN_22	V3	O(8)		TechIO: input data[22]		
D2	E1	PORT_E_07	PPUGPIO_07	V1	IOe(S8)		Ethernet Switch: I/O data[7]	PD	L
			DBG_D_07	V2	O(8)	0	ETM: debug data[23]		
			TECH_IN_23	V3	I(S)		TechIO: input data[23]		
D1	E3	PORT_E_08	PPUGPIO_08	V1	IOe(S8)		Ethernet Switch: I/O data[8]	PD	L
			DBG_D_08	V2	O(8)	0	ETM: debug data[24]		
			TECH_IN_24	V3	I(S)		TechIO: input data[24]		
F4	C3	PORT_E_09	PPUGPIO_09	V1	IOe(S8)		Ethernet Switch: I/O data[9]	PD	L
			DBG_D_09	V2	O(8)	0	ETM: debug data[25]		
			TECH_IN_25	V3	I(S)		TechIO: input data[25]		
F3	D2	PORT_E_10	PPUGPIO_10	V1	IOe(S8)		Ethernet Switch: I/O data[10]	PD	L
			DBG_D_10	V2	O(8)	0	ETM: debug data[26]		
			SSI1_DIN	V3	I(S)		TechIO SSI 1: input data		
E2	F4	PORT_E_11	PPUGPIO_11	V1	IOe(S8)		Ethernet Switch: I/O data[11]	PD	L
			DBG_D_11	V2	O(8)	0	ETM: debug data[27]		
			SSI1_CLK_IN	V3	I(S)		TechIO SSI 1: clock in		

Port E

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Debug Select	Description	Pull Up / Down	Reset State
E1	F3	PORT_E_12	PPUGPIO_12	V1	IOe(S8)		Ethernet Switch: I/O data[12]	PD	L
			DBG_D_12	V2	O(8)	0	ETM: debug data[28]		
			SSI1_CLK_OUT	V3	O(8)		TechIO SSI 1: clock out		
F2	G4	PORT_E_13	PPUGPIO_13	V1	IOe(S8)		Ethernet Switch: I/O data[13]	PD	L
			DBG_D_13	V2	O(8)	0	ETM: debug data[29]		
			SSI2_DIN	V3	I(S)		TechIO SSI 2: input data		
F1	G3	PORT_E_14	PPUGPIO_14	V1	IOe(S8)		Ethernet Switch: I/O data[14]	PD	L
			DBG_D_14	V2	O(8)	0	ETM: debug data[30]		
			SSI2_CLK_IN	V3	I(S)		TechIO SSI 2: clock in		
G2	H4	PORT_E_15	PPUGPIO_15	V1	IOe(S8)		Ethernet Switch: I/O data[15]	PD	L
			DBG_D_15	V2	O(8)	0	ETM: debug data[31]		
			SSI2_CLK_OUT	V3	O(8)		TechIO SSI 2: clock out		

Table 5-25 Port F

Port F								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Down**	**Reset State**
W12	AB12	GPIO_04	GPIO_04	V1	IOe(S8)	GPIO: I/O data[4]	PU	H
			IIC1_SCL	V2	IOe(S8)	I²C 1: serial clock		
Y12	AA12	GPIO_05	GPIO_05	V1	IOe(S8)	GPIO: I/O data[5]	PU	H
			IIC1_SDA	V2	IOe(S8)	I²C 1: serial data		

5.2.9.1 Flash Memory Interfaces

Table 5-26 NAND Flash Controller

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
				Port A				
A10	A11	PORT_A_00	NAND_BUSY_N	V1	I(S)	active-low busy signal	PU	H
B10	B11	PORT_A_01	NAND_WP_N		O(8)	active-low write protect	PU	H
D10	C11	PORT_A_02	NAND_WE_N		O(8)	active-low write enable	PU	H
C10	D11	PORT_A_03	NAND_RE_N		O(8)	active-low read enable	PU	H
A11	A12	PORT_A_04	NAND_CE_N		O(8)	active-low chip enable	PU	H
B11	B12	PORT_A_05	NAND_CLE		O(8)	command latch enable	PU	H
D11	C12	PORT_A_06	NAND_ALE		O(8)	address latch enable	PU	H
C11	D12	PORT_A_07	NAND_DATA_0		IOe(S8)	IO data	PU	H
A12	C13	PORT_A_08	NAND_DATA_1		IOe(S12)		PU	H
B12	D13	PORT_A_09	NAND_DATA_2		IOe(S12)		PU	H
E11	A13	PORT_A_10	NAND_DATA_3		IOe(S12)		PU	H
C12	B13	PORT_A_11	NAND_DATA_4		IOe(S12)		PU	H
A13	A14	PORT_A_12	NAND_DATA_5		IOe(S12)		PU	H
D12	B14	PORT_A_13	NAND_DATA_6		IOe(S12)		PU	H
C13	C14	PORT_A_14	NAND_DATA_7		IOe(S12)		PU	H

Table 5-27 SD/MMC Card Controller

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
				Port A				
A10	A11	PORT_A_00	SD_MMC_CD_N	V2	I(S)	active-low card detect	PU	H
B10	B11	PORT_A_01	SD_MMC_WP		I(S)	card write protect	PU	H
D10	C11	PORT_A_02	SD_MMC_CLK		O(8)	serial clock	PU	H
C10	D11	PORT_A_03	SD_MMC_DATA_0		IOe(S8)	IO data	PU	H
A11	A12	PORT_A_04	SD_DATA_1		IOe(S8)		PU	H
B11	B12	PORT_A_05	SD_DATA_2		IOe(S8)		PU	H
D11	C12	PORT_A_06	SD_DATA_3		IOe(S8)		PU	H
C11	D12	PORT_A_07	SD_MMC_COM		IOe(S8)	command line	PU	H

Table 5-28 QuadSPI

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
				Port A				
A12	C13	PORT_A_08	QUADSPI_CS_0_N	V2	O(12)	active-low chip select 0	PU	H
B12	D13	PORT_A_09	QUADSPI_CS_1_N		O(12)	active-low chip select 1	PU	H
E11	A13	PORT_A_10	QUADSPI_SCK_OUT		O(12)	serial clock	PU	H
C12	B13	PORT_A_11	QUADSPI_TX		IOe(S12)	transmit data / IO data[0]	PU	H
A13	A14	PORT_A_12	QUADSPI_RX		IOe(S12)	receive data / IO data[1]	PU	H
D12	B14	PORT_A_13	QUADSPI_WP_N		IOe(S12)	IO data[2]	PU	H
C13	C14	PORT_A_14	QUADSPI_HOLD_N		IOe(S12)	IO data[3]	PU	H

5.2.9.2 General Purpose I/Os

Table 5-29 Ethernet Switch General Purpose I/Os

Port E								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
A1	B1	PORT_E_00	PPUGPIO_00	V1	IOe(S8)	I/O data[0]	PD	L
B2	B2	PORT_E_01	PPUGPIO_01		IOe(S8)	I/O data[1]	PD	L
B1	D1	PORT_E_02	PPUGPIO_02		IOe(S8)	I/O data[2]	PD	L
C2	D3	PORT_E_03	PPUGPIO_03		IOe(S8)	I/O data[3]	PD	L
C1	E2	PORT_E_04	PPUGPIO_04		IOe(S8)	I/O data[4]	PD	L
D3	C2	PORT_E_05	PPUGPIO_05		IOe(S8)	I/O data[5]	PD	L
E3	C1	PORT_E_06	PPUGPIO_06		IOe(S8)	I/O data[6]	PD	L
D2	E1	PORT_E_07	PPUGPIO_07		IOe(S8)	I/O data[7]	PD	L
D1	E3	PORT_E_08	PPUGPIO_08		IOe(S8)	I/O data[8]	PD	L
F4	C3	PORT_E_09	PPUGPIO_09		IOe(S8)	I/O data[9]	PD	L
F3	D2	PORT_E_10	PPUGPIO_10		IOe(S8)	I/O data[10]	PD	L
E2	F4	PORT_E_11	PPUGPIO_11		IOe(S8)	I/O data[11]	PD	L
E1	F3	PORT_E_12	PPUGPIO_12		IOe(S8)	I/O data[12]	PD	L
F2	G4	PORT_E_13	PPUGPIO_13		IOe(S8)	I/O data[13]	PD	L
F1	G3	PORT_E_14	PPUGPIO_14		IOe(S8)	I/O data[14]	PD	L
G2	H4	PORT_E_15	PPUGPIO_15		IOe(S8)	I/O data[15]	PD	L

The following tables describe the configuration of the General Purpose I/Os which are independent of the Real-Time Ethernet Switch. The GPIO_00..05 are always available

Table 5-30 General Purpose I/Os on Port F

Port F								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
W12	AB12	GPIO_04	GPIO_04	V1	IOe(S8)	I/O data[4]	PU	H
Y12	AA12	GPIO_05	GPIO_05		IOe(S8)	I/O data[5]	PU	H

Table 5-31 General Purpose I/Os on Port B

Port B								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
D20	D21	PORT_B_26	GPIO_15	V2	IOe(S8)	I/O data[15]	PU	H
C21	E21	PORT_B_27	GPIO_14	V2 V3	IOe(S8)	I/O data[14]	PU	H
B20	B22	PORT_B_28	GPIO_13	V2 V3	IOe(S8)	I/O data[13]	PU	H
A21	B23	PORT_B_29	GPIO_12	V2 V3	IOe(S8)	I/O data[12]	PU	H
D19	A23	PORT_B_30	GPIO_11	V2 V3	IOe(S8)	I/O data[11]	PU	H
B21	C21	PORT_B_31	GPIO_10	V2 V3	IOe(S8)	I/O data[10]	PU	H
B22	F20	PORT_B_32	GPIO_09	V2 V3	IOe(S8)	I/O data[9]	PU	H
E19	C23	PORT_B_33	GPIO_08	V2 V3	IOe(S8)	I/O data[8]	PU	H
C22	D23	PORT_B_34	GPIO_07	V1 V2 V3	IOe(S8)	I/O data[7]	PU	H
D21	D22	PORT_B_35	GPIO_06	V1 V2 V3	IOe(S8)	I/O data[6]	PU	H

In order to use the General Purpose I/Os 16..23 either port C or port D can be used. If both ports are configured as General Purpose I/Os, then the port D is used and the outputs of port C are disabled.

Table 5-32 General Purpose I/Os on Port C

Port C (variant 1)								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
AA11	AB11	PORT_C_00	GPIO_23	V5	IOe(S8)	I/O data[23]	PU	H
AB10	AC10	PORT_C_01	GPIO_22	V5	IOe(S8)	I/O data[22]	PU	H
AA10	AB10	PORT_C_02	GPIO_21	V5	IOe(S8)	I/O data[21]	PU	H
AB9	AC9	PORT_C_03	GPIO_20	V5	IOe(S8)	I/O data[20]	PU	H
Y11	AA11	PORT_C_04	GPIO_19	V5	IOe(S8)	I/O data[19]	PU	H
W11	Y11	PORT_C_05	GPIO_18	V5	IOe(S8)	I/O data[18]	PU	H
Y10	AA10	PORT_C_06	GPIO_17	V2 V3 V5	IOe(S8)	I/O data[17]	PU	H
W10	Y10	PORT_C_07	GPIO_16	V1 V2 V3 V5	IOe(S8)	I/O data[16]	PU	H

Table 5-33 General Purpose I/Os on Port D

Port D (variant 2)								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
G1	F2	PORT_D_00	GPIO_16	V3	IOe(S8)	I/O data[16]	PU	H
H2	K3	PORT_D_01	GPIO_17		IOe(S8)	I/O data[17]	PU	H
H4	G2	PORT_D_02	GPIO_18		IOe(S8)	I/O data[18]	PU	H
G3	H3	PORT_D_03	GPIO_19		IOe(S8)	I/O data[19]	PU	H
H1	G1	PORT_D_04	GPIO_20		IOe(S8)	I/O data[20]	PU	H
J3	L4	PORT_D_05	GPIO_21		IOe(S8)	I/O data[21]	PU	H
J5	L3	PORT_D_06	GPIO_22		IOe(S8)	I/O data[22]	PU	H
H3	J4	PORT_D_07	GPIO_23		IOe(S8)	I/O data[23]	PU	H
L5	L2	PORT_D_08	GPIO_24		IOe(S8)	I/O data[24]	PU	H
M2	M4	PORT_D_09	GPIO_25		IOe(S8)	I/O data[25]	PU	H
J4	F1	PORT_D_10	GPIO_26		IOe(S8)	I/O data[26]	PU	H
L4	L1	PORT_D_11	GPIO_27		IOe(S8)	I/O data[27]	PU	H
M1	M3	PORT_D_12	GPIO_28		IOe(S8)	I/O data[28]	PU	H
N2	M2	PORT_D_13	GPIO_29		IOe(S8)	I/O data[29]	PU	H
M5	M1	PORT_D_14	GPIO_30		IOe(S8)	I/O data[30]	PU	H
M3	N1	PORT_D_15	GPIO_31		IOe(S8)	I/O data[31]	PU	H

5.2.9.3 Ethernet

Table 5-34 Real-Time Ethernet Switch MII

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
			Port B					
B13	B15	PORT_B_00	MII1_TX_CK	V1	I(S)	MII 1: transmit clock	PU	H
A14	A16	PORT_B_01	MII1_RXD_0		I(S)	MII 1: receive data	PU	H
E12	A15	PORT_B_02	MII1_RXD_1		I(S)		PU	H
C14	B16	PORT_B_03	MII1_RXD_2		I(S)		PU	H
B14	A17	PORT_B_04	MII1_RXD_3		I(S)		PU	H
A15	B17	PORT_B_05	MII1_TXD_3		O(8)	MII 1: transmit data	PU	H
D13	C15	PORT_B_06	MII1_TXD_2		O(8)		PU	H
C15	A18	PORT_B_07	MII1_TXD_1		O(8)		PU	H
B15	C16	PORT_B_08	MII1_TXD_0		O(8)		PU	H
A16	B18	PORT_B_09	MII1_TX_ER		O(8)	MII 1: transmit error	PU	H
D14	D15	PORT_B_10	MII1_TX_EN		O(8)	MII 1: transmit enable	PU	H
C16	A19	PORT_B_11	MII1_RX_ER		I(S)	MII 1: receive error detected	PU	H
B16	C17	PORT_B_12	MII1_RX_CK		I(S)	MII 1: receive clock	PU	H
A17	D17	PORT_B_13	MII1_RX_DV		I(S)	MII 1: receive data valid	PU	H
D15	D16	PORT_B_14	MII1_CRS		I(S)	MII 1: carrier sense	PU	H
C17	B19	PORT_B_15	MII1_COL		I(S)	MII 1: collision detect	PU	H
B17	C18	PORT_B_16	MII2_TX_CK		I(S)	MII 2: transmit clock	PU	H
A18	D18	PORT_B_17	MII2_RXD_0		I(S)	MII 2: receive data	PU	H
D16	A20	PORT_B_18	MII2_RXD_1		I(S)		PU	H
C18	B20	PORT_B_19	MII2_RXD_2		I(S)		PU	H
B18	C19	PORT_B_20	MII2_RXD_3		I(S)		PU	H
A19	A21	PORT_B_21	MII2_TXD_3		O(8)	MII 2: transmit data	PU	H
C19	C20	PORT_B_22	MII2_TXD_2		O(8)		PU	H
C20	C22	PORT_B_23	MII2_TXD_1		O(8)		PU	H
B19	B21	PORT_B_24	MII2_TXD_0		O(8)		PU	H
A20	A22	PORT_B_25	MII2_TX_ER		O(8)	MII 2: transmit error	PU	H
D20	D21	PORT_B_26	MII2_TX_EN		O(8)	MII 2: transmit enable	PU	H
C21	E21	PORT_B_27	MII2_RX_ER		I(S)	MII 2: receive error detected	PU	H
B20	B22	PORT_B_28	MII2_RX_CK		I(S)	MII 2: receive clock	PU	H
A21	B23	PORT_B_29	MII2_RX_DV		I(S)	MII 2: receive data valid	PU	H
D19	A23	PORT_B_30	MII2_CRS		I(S)	MII 2: carrier sense	PU	H
B21	C21	PORT_B_31	MII2_COL		I(S)	MII 2: collision detect	PU	H
B22	F20	PORT_B_32	MII_MDC		O(8)	MII: management data clock	PU	H
E19	C23	PORT_B_33	MII_MDIO		IOe(S8)	MII: management data I/O	PU	H
C22	D23	PORT_B_34	GPIO_07		I(S)	MII 2: link done	PU	H
D21	D22	PORT_B_35	GPIO_06		I(S)	MII 1: link done	PU	H

Table 5-35 Gigabit Ethernet GMII (Extension from MII to GMII)

Port B								
Ball 380	**Ball** 385	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Pull**	**Reset State**
B13	B15	PORT_B_00	GMII_GTXCLK	V2	O(8)	transmit clock	PU	H
A14	A16	PORT_B_01	GMII_RXD_4		I(S)	receive data	PU	H
E12	A15	PORT_B_02	GMII_RXD_5		I(S)		PU	H
C14	B16	PORT_B_03	GMII_RXD_6		I(S)		PU	H
B14	A17	PORT_B_04	GMII_RXD_7		I(S)		PU	H
A15	B17	PORT_B_05	GMII_TXD_7		O(8)	transmit data	PU	H
D13	C15	PORT_B_06	GMII_TXD_6		O(8)		PU	H
C15	A18	PORT_B_07	GMII_TXD_5		O(8)		PU	H
B15	C16	PORT_B_08	GMII_TXD_4		O(8)		PU	H

5.2.9.4 Fieldbus Interfaces

Table 5-36 PROFIBUS Master

Port C								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
AA11	AB11	PORT_C_00	PBM1_RXD	V4	I(S)	PROFIBUS master 1: receive data	PU	H
AB10	AC10	PORT_C_01	PBM1_CTS_N		I(S)	PROFIBUS master 1: active-low clear to send	PU	H
AA10	AB10	PORT_C_02	PBM1_RTS		O(8)	PROFIBUS master 1: request to send	PU	H
AB9	AC9	PORT_C_03	PBM1_TXD		Oe(8)	PROFIBUS master 1: transmit data	PU	H
Y11	AA11	PORT_C_04	PBM2_RXD		I(S)	PROFIBUS master 2: receive data	PU	H
W11	Y11	PORT_C_05	PBM2_CTS_N		I(S)	PROFIBUS master 2: active-low clear to send	PU	H
Y10	AA10	PORT_C_06	PBM2_RTS		O(8)	PROFIBUS master 2: request to send	PU	H
W10	Y10	PORT_C_07	PBM2_TXD		Oe(8)	PROFIBUS master 2: transmit data	PU	H

In order to use the PROFIBUS slave either port C or port D can be used. If both ports are configured as PROFIBUS slave, then the port D is used and the outputs of port C are disabled.

Table 5-37 PROFIBUS Slave (variant 1)

Port C (variant 1)								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
AA11	AB11	PORT_C_00	VPC_RXD	V1	I(S)	receive data	PU	H
AB10	AC10	PORT_C_01	VPC_CTS_N		I(S)	active-low clear to send	PU	H
AA10	AB10	PORT_C_02	VPC_RTS		O(8)	request to send	PU	H
AB9	AC9	PORT_C_03	VPC_TXD		Oe(8)	transmit data	PU	H
Y10	AA10	PORT_C_06	VPC_DATAEXCH_N		O(8)	indicate state DATA-EXCH	PU	H

Table 5-38 PROFIBUS Slave (variant 2)

Port D (variant 1)								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
P1	P2	PORT_D_20	VPC_RXD	V3	I(S)	receive data	PU	H
R2	R1	PORT_D_21	VPC_CTS_N		I(S)	active-low clear to send	PU	H
N4	R2	PORT_D_22	VPC_RTS		O(8)	request to send	PU	H
P3	R3	PORT_D_23	VPC_DATAEXCH_N		O(8)	indicate state DATA-EXCH	PU	H
R1	T1	PORT_D_24	VPC_TXD		Oe(8)	transmit data	PU	H

In order to use the CAN interfaces either port C or port D can be used. If both ports are configured as PROFIBUS slave, then the port D is used and the outputs of port C are disabled.

Table 5-39 CAN (variant 1)

Port C (variant 1)								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Pull**	**Reset State**
AA11	AB11	PORT_C_00	CAN1_RX	V3	I(S)	CAN 1: receive data	PU	H
AB10	AC10	PORT_C_01	CAN1_TX		O(8)	CAN 1: transmit data	PU	H
AA10	AB10	PORT_C_02	CAN2_RX		I(S)	CAN 2: receive data	PU	H
AB9	AC9	PORT_C_03	CAN2_TX		O(8)	CAN 2: transmit data	PU	H

Table 5-40 CAN (variant 2)

Port D (variant 2)								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Pull**	**Reset State**
N1	N2	PORT_D_16	CAN1_RX	V3	I(S)	CAN 1: receive data	PU	H
P2	N4	PORT_D_17	CAN1_TX		O(8)	CAN 1: transmit data	PU	H
M4	N3	PORT_D_18	CAN2_RX		I(S)	CAN 2: receive data	PU	H
N3	P1	PORT_D_19	CAN2_TX		O(8)	CAN 2: transmit data	PU	H

5.2.9.5 Asynchronous External Interface (AEI)

Table 5-41 Master Interface

Port D								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Pull**	**Reset State**
G1	F2	PORT_D_00	AEI_M_D_15	V1	IOe(S8)	data_bus	PU	H
H2	K3	PORT_D_01	AEI_M_D_14		IOe(S8)		PU	H
H4	G2	PORT_D_02	AEI_M_D_13		IOe(S8)		PU	H
G3	H3	PORT_D_03	AEI_M_D_12		IOe(S8)		PU	H
H1	G1	PORT_D_04	AEI_M_D_11		IOe(S8)		PU	H
J3	L4	PORT_D_05	AEI_M_D_10		IOe(S8)		PU	H
J5	L3	PORT_D_06	AEI_M_D_09		IOe(S8)		PU	H
H3	J4	PORT_D_07	AEI_M_D_08		IOe(S8)		PU	H
L5	L2	PORT_D_08	AEI_M_D_07		IOe(S8)		PU	H
M2	M4	PORT_D_09	AEI_M_D_06		IOe(S8)		PU	H
J4	F1	PORT_D_10	AEI_M_D_05		IOe(S8)		PU	H
L4	L1	PORT_D_11	AEI_M_D_04		IOe(S8)		PU	H
M1	M3	PORT_D_12	AEI_M_D_03		IOe(S8)		PU	H
N2	M2	PORT_D_13	AEI_M_D_02		IOe(S8)		PU	H
M5	M1	PORT_D_14	AEI_M_D_01		IOe(S8)		PU	H
M3	N1	PORT_D_15	AEI_M_D_00		IOe(S8)		PU	H
N1	N2	PORT_D_16	AEI_M_A_00		O(8)	address bus	PU	H
P2	N4	PORT_D_17	AEI_M_A_01		O(8)		PU	H
M4	N3	PORT_D_18	AEI_M_A_02		O(8)		PU	H
N3	P1	PORT_D_19	AEI_M_A_03		O(8)		PU	H
P1	P2	PORT_D_20	AEI_M_A_04		O(8)		PU	H
R2	R1	PORT_D_21	AEI_M_A_05		O(8)		PU	H
N4	R2	PORT_D_22	AEI_M_A_06		O(8)		PU	H
P3	R3	PORT_D_23	AEI_M_A_07		O(8)		PU	H
R1	T1	PORT_D_24	AEI_M_A_08		O(8)		PU	H
T2	T2	PORT_D_25	AEI_M_A_09		O(8)		PU	H
P4	T3	PORT_D_26	AEI_M_A_10		O(8)		PU	H
R3	T4	PORT_D_27	AEI_M_A_11		O(8)		PU	H
T1	U1	PORT_D_28	AEI_M_A_12		O(8)		PU	H
U2	U2	PORT_D_29	AEI_M_A_13		O(8)		PU	H
P5	U3	PORT_D_30	AEI_M_A_14		O(8)		PU	H
R4	U4	PORT_D_31	AEI_M_A_15		O(8)		PU	H
U1	V1	PORT_D_32	AEI_M_A_16		O(8)		PU	H
T4	V2	PORT_D_33	AEI_M_A_17		O(8)		PU	H
U4	V3	PORT_D_34	AEI_M_A_18		O(8)		PU	H
V3	V4	PORT_D_35	AEI_M_A_19		O(8)		PU	H
V1	W1	PORT_D_36	AEI_M_A_20		O(8)		PU	H
W3	Y1	PORT_D_37	AEI_M_CS_0_N		O(8)	active-low chip select[0]	PU	H
W1	W2	PORT_D_38	AEI_M_CS_1_N		O(8)	active-low chip select[1]	PU	H
W2	W3	PORT_D_39	AEI_M_BS_1_N		O(8)	active-low high-byte select	PU	H
Y1	Y2	PORT_D_40	AEI_M_RE_N		O(8)	active-low read enable	PU	H
Y2	AA1	PORT_D_41	AEI_M_WE_N		O(8)	active-low write enable	PU	H

Port D								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
AA1	AA2	PORT_D_42	AEI_M_WAIT	V1	I(S)	wait signal	PU	H
V4	AB1	PORT_D_43	AEI_M_INT_0		I(S)	1st IRQ line, could be used for SNAP+ synchronization	PU	H
AB1	AC1	PORT_D_44	AEI_M_INT_1		I(S)	2nd IRQ line	PU	H

Table 5-42 Slave Interface

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description		Pull Up / Pull	Reset State
Port D									
G1	F2	PORT_D_00	AEI_S_D_15	V2	IOe(S8)	data bus		PU	H
H2	K3	PORT_D_01	AEI_S_D_14		IOe(S8)			PU	H
H4	G2	PORT_D_02	AEI_S_D_13		IOe(S8)			PU	H
G3	H3	PORT_D_03	AEI_S_D_12		IOe(S8)			PU	H
H1	G1	PORT_D_04	AEI_S_D_11		IOe(S8)			PU	H
J3	L4	PORT_D_05	AEI_S_D_10		IOe(S8)			PU	H
J5	L3	PORT_D_06	AEI_S_D_09		IOe(S8)			PU	H
H3	J4	PORT_D_07	AEI_S_D_08		IOe(S8)			PU	H
L5	L2	PORT_D_08	AEI_S_D_07		IOe(S8)			PU	H
M2	M4	PORT_D_09	AEI_S_D_06		IOe(S8)			PU	H
J4	F1	PORT_D_10	AEI_S_D_05		IOe(S8)			PU	H
L4	L1	PORT_D_11	AEI_S_D_04		IOe(S8)			PU	H
M1	M3	PORT_D_12	AEI_S_D_03		IOe(S8)			PU	H
N2	M2	PORT_D_13	AEI_S_D_02		IOe(S8)			PU	H
M5	M1	PORT_D_14	AEI_S_D_01		IOe(S8)			PU	H
M3	N1	PORT_D_15	AEI_S_D_00		IOe(S8)			PU	H
N1	N2	PORT_D_16	AEI_S_A_00		I(S)	used to address CI and FIFO registers	address bus	PU	H
P2	N4	PORT_D_17	AEI_S_A_01		I(S)			PU	H
M4	N3	PORT_D_18	AEI_S_A_02		I(S)			PU	H
N3	P1	PORT_D_19	AEI_S_A_03		I(S)			PU	H
P1	P2	PORT_D_20	AEI_S_A_04		I(S)			PU	H
R2	R1	PORT_D_21	AEI_S_A_05		I(S)			PU	H
N4	R2	PORT_D_22	AEI_S_A_06		I(S)			PU	H
P3	R3	PORT_D_23	AEI_S_A_07		I(S)			PU	H
R1	T1	PORT_D_24	AEI_S_A_08		I(S)	must be tied to '0'		PU	H
T2	T2	PORT_D_25	AEI_S_A_09		I(S)			PU	H
P4	T3	PORT_D_26	AEI_S_A_10		I(S)			PU	H
R3	T4	PORT_D_27	AEI_S_A_11		I(S)			PU	H
T1	U1	PORT_D_28	AEI_S_A_12		I(S)			PU	H
U2	U2	PORT_D_29	AEI_S_A_13		I(S)			PU	H
P5	U3	PORT_D_30	AEI_S_A_14		I(S)			PU	H
R4	U4	PORT_D_31	AEI_S_A_15		I(S)			PU	H
U1	V1	PORT_D_32	AEI_S_A_16		I(S)			PU	H
T4	V2	PORT_D_33	AEI_S_A_17		I(S)			PU	H
U4	V3	PORT_D_34	AEI_S_A_18		I(S)			PU	H
V3	V4	PORT_D_35	AEI_S_A_19		I(S)			PU	H
V1	W1	PORT_D_36	AEI_S_A_20		I(S)	select between CI ('1') and FIFO ('0')		PU	H
W3	Y1	PORT_D_37	AEI_S_CS_0_N	V2	I(S)	active-low chip select[0]		PU	H
W2	W3	PORT_D_39	AEI_S_BS_1_N		I(S)	active-low high-byte select		PU	H
Y1	Y2	PORT_D_40	AEI_S_RE_N		I(S)	active-low read enable		PU	H
Y2	AA1	PORT_D_41	AEI_S_WE_N		I(S)	active-low write enable		PU	H
AA1	AA2	PORT_D_42	AEI_S_WAIT		O(8)	wait signal		PU	H
AB1	AC1	PORT_D_44	AEI_S_INT_1		O(8)	IRQ line		PU	H

5.2.9.6 Serial Interfaces

Table 5-43 I²C 1

Port F								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Pull**	**Reset State**
W12	AB12	GPIO_04	IIC1_SCL	V2	IOe(S8)	I²C 1: serial clock	PU	H
Y12	AA12	GPIO_05	IIC1_SDA		IOe(S8)	I²C 1: serial data	PU	H

Table 5-44 I²C 2

Port C								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Pull**	**Reset State**
Y11	AA11	PORT_C_04	IIC2_SCL	V1	IOe(S8)	I²C 2: serial clock	PU	H
				V2				
				V3				
W11	Y11	PORT_C_05	IIC2_SDA	V1	IOe(S8)	I²C 2: serial data	PU	H
				V2				
				V3				

Table 5-45 USB 2.0 Device Controller

Port E								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Pull**	**Reset State**
B18	C19	PORT_B_20	USB_INT	V3	I(S)	USB: cable connected	PU	H

In order to use UART 2 either port C or port D can be used. If both ports are configured as PROFIBUS slave, then the port D is used and the outputs of port C are disabled.

Table 5-46 UART 2 (variant 1)

Port C (variant 1)								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
AA11	AB11	PORT_C_00	UART2_RXD	V2	I(S)	receive data	PU	H
AB10	AC10	PORT_C_01	UART2_CTS_N		I(S)	active-low clear to send	PU	H
AA10	AB10	PORT_C_02	UART2_RTS_N		O(8)	active-low request to send	PU	H
AB9	AC9	PORT_C_03	UART2_TXD		O(8)	transmit data	PU	H

Table 5-47 UART 2 (variant 2)

Port D (variant 2)								
Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
T2	T2	PORT_D_25	UART2_RXD	V3	I(S)	receive data	PU	H
P4	T3	PORT_D_26	UART2_CTS_N		I(S)	active-low clear to send	PU	H
R3	T4	PORT_D_27	UART2_RTS_N		O(8)	active-low request to send	PU	H
T1	U1	PORT_D_28	UART2_TXD		O(8)	transmit data	PU	H

5.2.9.7 Technology Function Module

In order to use the TechIO input and output ports either port B, D or E can be used. If port D and port B or E are configured as TechIO, then the port D is used and the outputs of port B and E are disabled.

Table 5-48 Configuration of Technology Function Module I/Os

Technology Function		Port B V2	Port B V3	Port B V4	Port D V4	Port E V3
input data	[5:0]	x	x	x	x	
	[11:6]	x		x		
	[19:12]			x	x	
	[24:20]				x	x
	[25]					x
output data	[3:0]	x	x	x	x	
	[15:3]			x	x	
	[19:16]				x	x
SSI 1						x
SSI 2						x

Table 5-49 Technology Function Module I/Os of port B

Ball 380	Ball 385	Pin Name	Signal Name	Mode	Type	Description	Pull Up / Pull	Reset State
B13	B15	PORT_B_00	TECH_OUT_12	V4	O(8)	output data[12]	PU	H
A14	A16	PORT_B_01	TECH_OUT_11	V4	O(8)	output data[11]	PU	H
E12	A15	PORT_B_02	TECH_OUT_10	V4	O(8)	output data[10]	PU	H
C14	B16	PORT_B_03	TECH_OUT_09	V4	O(8)	output data[9]	PU	H
B14	A17	PORT_B_04	TECH_OUT_08	V4	O(8)	output data[8]	PU	H
A15	B17	PORT_B_05	TECH_OUT_07	V4	O(8)	output data[7]	PU	H
D13	C15	PORT_B_06	TECH_OUT_06	V4	O(8)	output data[6]	PU	H
C15	A18	PORT_B_07	TECH_OUT_05	V4	O(8)	output data[5]	PU	H
B15	C16	PORT_B_08	TECH_OUT_04	V4	O(8)	output data[4]	PU	H
A16	B18	PORT_B_09	TECH_OUT_03	V2 V3 V4	O(8)	output data[3]	PU	H
D14	D15	PORT_B_10	TECH_OUT_02	V2 V3 V4	O(8)	output data[2]	PU	H
C16	A19	PORT_B_11	TECH_OUT_01	V2 V3 V4	O(8)	output data[1]	PU	H
B16	C17	PORT_B_12	TECH_OUT_00	V2 V3 V4	O(8)	output data[0]	PU	H
A17	D17	PORT_B_13	TECH_IN_00	V2 V3 V4	I(S)	input data[0]	PU	H

Port B								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Pull**	**Reset State**
D15	D16	PORT_B_14	TECH_IN_01	V2 V3 V4	I(S)	input data[1]	PU	H
C17	B19	PORT_B_15	TECH_IN_02	V2 V3 V4	I(S)	input data[2]	PU	H
B17	C18	PORT_B_16	TECH_IN_03	V2 V3 V4	I(S)	input data[3]	PU	H
A18	D18	PORT_B_17	TECH_IN_04	V2 V3 V4	I(S)	input data[4]	PU	H
D16	A20	PORT_B_18	TECH_IN_05	V2 V3 V4	I(S)	input data[5]	PU	H
C18	B20	PORT_B_19	TECH_IN_06	V2 V4	I(S)	input data[6]	PU	H
B18	C19	PORT_B_20	TECH_IN_07	V2 V4	I(S)	input data[7]	PU	H
A19	A21	PORT_B_21	TECH_IN_08	V2 V4	I(S)	input data[8]	PU	H
C19	C20	PORT_B_22	TECH_IN_09	V2 V4	I(S)	input data[9]	PU	H
C20	C22	PORT_B_23	TECH_IN_10	V2 V4	I(S)	input data[10]	PU	H
B19	B21	PORT_B_24	TECH_IN_11	V2 V2	I(S)	input data[11]	PU	H
A20	A22	PORT_B_25	TECH_IN_12	V4	I(S)	input data[12]	PU	H
D20	D21	PORT_B_26	TECH_IN_13	V4	I(S)	input data[13]	PU	H
C21	E21	PORT_B_27	TECH_IN_14	V4	I(S)	input data[14]	PU	H
B20	B22	PORT_B_28	TECH_IN_15	V4	I(S)	input data[15]	PU	H
A21	B23	PORT_B_29	TECH_IN_16	V4	I(S)	input data[16]	PU	H
D19	A23	PORT_B_30	TECH_IN_17	V4	I(S)	input data[17]	PU	H
B21	C21	PORT_B_31	TECH_IN_18	V4	I(S)	input data[18]	PU	H
B22	F20	PORT_B_32	TECH_IN_19	V4	I(S)	input data[19]	PU	H
E19	C23	PORT_B_33	TECH_OUT_15	V4	O(8)	output data[15]	PU	H
C22	D23	PORT_B_34	TECH_OUT_14	V4	O(8)	output data[14]	PU	H
D21	D22	PORT_B_35	TECH_OUT_13	V4	O(8)	output data[13]	PU	H

Table 5-50 Technology Function Module I/Os of Port D

Port D								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Pull**	**Reset State**
G1	F2	PORT_D_00	TECH_OUT_00	V4	O(8)	output data[0]	PU	H
H2	K3	PORT_D_01	TECH_OUT_01	V4	O(8)	output data[1]	PU	H
H4	G2	PORT_D_02	TECH_OUT_02	V4	O(8)	output data[2]	PU	H
G3	H3	PORT_D_03	TECH_OUT_03	V4	O(8)	output data[3]	PU	H
H1	G1	PORT_D_04	TECH_OUT_04	V4	O(8)	output data[4]	PU	H
J3	L4	PORT_D_05	TECH_OUT_05	V4	O(8)	output data[5]	PU	H
J5	L3	PORT_D_06	TECH_OUT_06	V4	O(8)	output data[6]	PU	H
H3	J4	PORT_D_07	TECH_OUT_07	V4	O(8)	output data[7]	PU	H
L5	L2	PORT_D_08	TECH_OUT_08	V4	O(8)	output data[8]	PU	H
M2	M4	PORT_D_09	TECH_OUT_09	V4	O(8)	output data[9]	PU	H
J4	F1	PORT_D_10	TECH_OUT_10	V4	O(8)	output data[10]	PU	H
L4	L1	PORT_D_11	TECH_OUT_11	V4	O(8)	output data[11]	PU	H
M1	M3	PORT_D_12	TECH_OUT_12	V4	O(8)	output data[12]	PU	H
N2	M2	PORT_D_13	TECH_OUT_13	V4	O(8)	output data[13]	PU	H
M5	M1	PORT_D_14	TECH_OUT_14	V4	O(8)	output data[14]	PU	H
M3	N1	PORT_D_15	TECH_OUT_15	V4	O(8)	output data[15]	PU	H
N1	N2	PORT_D_16	TECH_OUT_16	V4	O(8)	output data[16]	PU	H
P2	N4	PORT_D_17	TECH_OUT_17	V4	O(8)	output data[17]	PU	H
M4	N3	PORT_D_18	TECH_OUT_18	V4	O(8)	output data[18]	PU	H
N3	P1	PORT_D_19	TECH_OUT_19	V4	O(8)	output data[19]	PU	H
P1	P2	PORT_D_20	TECH_IN_00	V4	I(S)	input data[0]	PU	H
R2	R1	PORT_D_21	TECH_IN_01	V4	I(S)	input data[1]	PU	H
N4	R2	PORT_D_22	TECH_IN_02	V4	I(S)	input data[2]	PU	H
P3	R3	PORT_D_23	TECH_IN_03	V4	I(S)	input data[3]	PU	H
R1	T1	PORT_D_24	TECH_IN_04	V4	I(S)	input data[4]	PU	H
T2	T2	PORT_D_25	TECH_IN_05	V4	I(S)	input data[5]	PU	H
P4	T3	PORT_D_26	TECH_IN_06	V4	I(S)	input data[6]	PU	H
R3	T4	PORT_D_27	TECH_IN_07	V4	I(S)	input data[7]	PU	H
T1	U1	PORT_D_28	TECH_IN_08	V4	I(S)	input data[8]	PU	H
U2	U2	PORT_D_29	TECH_IN_09	V4	I(S)	input data[9]	PU	H
P5	U3	PORT_D_30	TECH_IN_10	V4	I(S)	input data[10]	PU	H
R4	U4	PORT_D_31	TECH_IN_11	V4	I(S)	input data[11]	PU	H
U1	V1	PORT_D_32	TECH_IN_12	V4	I(S)	input data[12]	PU	H
T4	V2	PORT_D_33	TECH_IN_13	V4	I(S)	input data[13]	PU	H
U4	V3	PORT_D_34	TECH_IN_14	V4	I(S)	nput data[14]	PU	H
V3	V4	PORT_D_35	TECH_IN_15	V4	I(S)	input data[15]	PU	H
V1	W1	PORT_D_36	TECH_IN_16	V4	I(S)	input data[16]	PU	H
W3	Y1	PORT_D_37	TECH_IN_17	V4	I(S)	input data[17]	PU	H
W1	W2	PORT_D_38	TECH_IN_18	V4	I(S)	input data[18]	PU	H
W2	W3	PORT_D_39	TECH_IN_19	V4	I(S)	input data[19]	PU	H
Y1	Y2	PORT_D_40	TECH_IN_20	V4	I(S)	input data[20]	PU	H
Y2	AA1	PORT_D_41	TECH_IN_21	V4	I(S)	input data[21]	PU	H
AA1	AA2	PORT_D_42	TECH_IN_22	V4	I(S)	input data[22]	PU	H
V4	AB1	PORT_D_43	TECH_IN_23	V4	I(S)	input data[23]	PU	H
AB1	AC1	PORT_D_44	TECH_IN_24	V4	I(S)	input data[24]	PU	H

Table 5-51 Technology Function Module I/Os on Port E

Port E								
Ball 380	**Ball 385**	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Description**	**Pull Up / Pull**	**Reset State**
A1	B1	PORT_E_00	TECH_OUT_16	V3	O(8)	output data[16]	PD	L
B2	B2	PORT_E_01	TECH_OUT_17		O(8)	output data[17]	PD	L
B1	D1	PORT_E_02	TECH_OUT_18		O(8)	output data[18]	PD	L
C2	D3	PORT_E_03	TECH_OUT_19		O(8)	output data[19]	PD	L
C1	E2	PORT_E_04	TECH_IN_20		I(S)	input data[20]	PD	L
D3	C2	PORT_E_05	TECH_IN_21		I(S)	input data[21]	PD	L
E3	C1	PORT_E_06	TECH_IN_22		I(S)	input data[22]	PD	L
D2	E1	PORT_E_07	TECH_IN_23		I(S)	input data[23]	PD	L
D1	E3	PORT_E_08	TECH_IN_24		I(S)	input data[24]	PD	L
F4	C3	PORT_E_09	TECH_IN_25		I(S)	input data[25]	PD	L
F3	D2	PORT_E_10	SSI1_DIN		I(S)	SSI 1: input data	PD	L
E2	F4	PORT_E_11	SSI1_CLK_IN		I(S)	SSI 1: clock in	PD	L
E1	F3	PORT_E_12	SSI1_CLK_OUT		O(8)	SSI 1: clock out	PD	L
F2	G4	PORT_E_13	SSI2_DIN		I(S)	SSI 2: input data	PD	L
F1	G3	PORT_E_14	SSI2_CLK_IN		I(S)	SSI 2: clock in	PD	L
G2	H4	PORT_E_15	SSI2_CLK_OUT		O(8)	SSI 2: clock out	PD	L

5.2.9.8 Development Interfaces

Table 5-52 Debug Ports

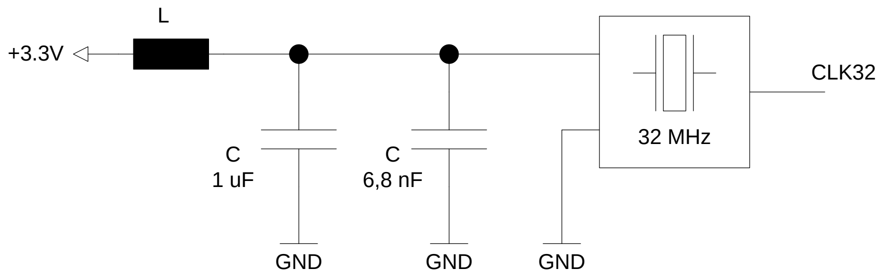
Port E									
Ball 380	**Ball** 385	**Pin Name**	**Signal Name**	**Mode**	**Type**	**Debug Select**	**Description**	**Pull Up / Pull**	**Reset State**
A1	B1	PORT_E_00	DBG_D_00	V2	O(8)	0	ETM: debug data[16]	PD	L
						others	reserved		
B2	B2	PORT_E_01	DBG_D_01	V2	O(8)	0	ETM: debug data[17]	PD	L
						others	reserved		
B1	D1	PORT_E_02	DBG_D_02	V2	O(8)	0	ETM: debug data[18]	PD	L
						others	reserved		
C2	D3	PORT_E_03	DBG_D_03	V2	O(8)	0	ETM: debug data[19]	PD	L
						others	reserved		
C1	E2	PORT_E_04	DBG_D_04	V2	O(8)	0	ETM: debug data[20]	PD	L
						others	reserved		
D3	C2	PORT_E_05	DBG_D_05	V2	O(8)	0	ETM: debug data[21]	PD	L
						others	reserved		
E3	C1	PORT_E_06	DBG_D_06	V2	O(8)	0	ETM: debug data[22]	PD	L
						others	reserved		
D2	E1	PORT_E_07	DBG_D_07	V2	O(8)	0	ETM: debug data[23]	PD	L
						others	reserved		
D1	E3	PORT_E_08	DBG_D_08	V2	O(8)	0	ETM: debug data[24]	PD	L
						others	reserved		
F4	C3	PORT_E_09	DBG_D_09	V2	O(8)	0	ETM: debug data[25]	PD	L
						others	reserved		
F3	D2	PORT_E_10	DBG_D_10	V2	O(8)	0	ETM: debug data[26]	PD	L
						others	reserved		
E2	F4	PORT_E_11	DBG_D_11	V2	O(8)	0	ETM: debug data[27]	PD	L
						others	reserved		
E1	F3	PORT_E_12	DBG_D_12	V2	O(8)	0	ETM: debug data[28]	PD	L
						others	reserved		
F2	G4	PORT_E_13	DBG_D_13	V2	O(8)	0	ETM: debug data[29]	PD	L
						others	reserved		
F1	G3	PORT_E_14	DBG_D_14	V2	O(8)	0	ETM: debug data[30]	PD	L
						others	reserved		
G2	H4	PORT_E_15	DBG_D_15	V2	O(8)	0	ETM: debug data[31]	PD	L
						others	reserved		

6 Clock and Reset Configuration

6.1 Clock

In order to reduce the EMI of PCB, a low-frequency crystal oscillator with 32 MHz is used as clock source for ANTAIOS. Two internal PLLs are used as clock generators, which synthesizes the high-frequency clocks out of this input clock.
Use a crystal oscillator for a fixed frequency instead of a programmable one, because of the lower jitter value.

Figure 6-1 Example for the Clock Source



6.1.1 Qualified Chips

- Crystal Oscillator: Seiko Epson SG-310 SCF

6.2 Reset

6.2.1 Reset Sources

The ANTAIOS offers several reset sources which can be used for a system reset of the chip. Two of them are external inputs, while another is generated by internal function block.

6.2.1.1 Power-On Reset (RESET_N)

This (active-low) input signal shall be connected to the output of a voltage supervisor chip, which checks the power supply voltages and pulls the power on reset pin low, whenever the voltages are below the minimum specified operating voltages. The power-on reset signal causes an asynchronous reset of the chip and initializes all internal registers and signals to their power on reset state. The reset shall be active for the time period till the supply voltages stays stable and the PLL is locked. Also the clock-system is reset. The power-on reset phase is internally enlarged, because of the low clocks. A communication of the debugger over the JTAG-interface isn't possible at this time.

6.2.1.2 Hardware Reset (SRST_N)

The asynchronous hardware reset is an (active low) input signal which is activated from the external debugger. During the active phase the complete internal logic is reset, but not the clock-system. During the reset phase the debugger can communicate over the JTAG-

interface with the ETM. The hardware reset phase is internally enlarged, because of the low clocks.
If not used, this ball may be left unconnected, since the input buffer is equipped with an internal pull-up resistor.

6.2.1.3 Watchdog Reset

The watchdog reset is a hardware observation to prevent system lockup due to software or hardware failures. The base for the observation is a time which is adjustable in the watchdog timer. On activation of the watchdog the time counts. If no retrigger is done within this time, the watchdog-reset is triggered. If the watchdog function is enabled the ANTAIOS is reset. In order to analyse the source of reset after a restart, the WdResetStatus register of the watchdog can be used.

6.2.1.4 JTAG Reset (JTAG_TRST_N)

There is also one input signal to reset the test port.
The asynchronous JTAG reset is an (active low) input signal which is activated from the external debugger. Only the embedded trace macrocell of the Cortex-A5 is reset. To ensure a defined state of the embedded trace macrocell without using debugger, the logic is also reset during power-on reset.

6.2.2 Reset Behavior

An order for the turning on/off of the supply voltages is not prescribed.
Only the power-on reset influences the internal clock generation, which is stopped during active-low reset. After removal of the external reset the internal reset signals remain active for $t_{RESDELAY}$.

Figure 6-2 Power-On Reset Sequence

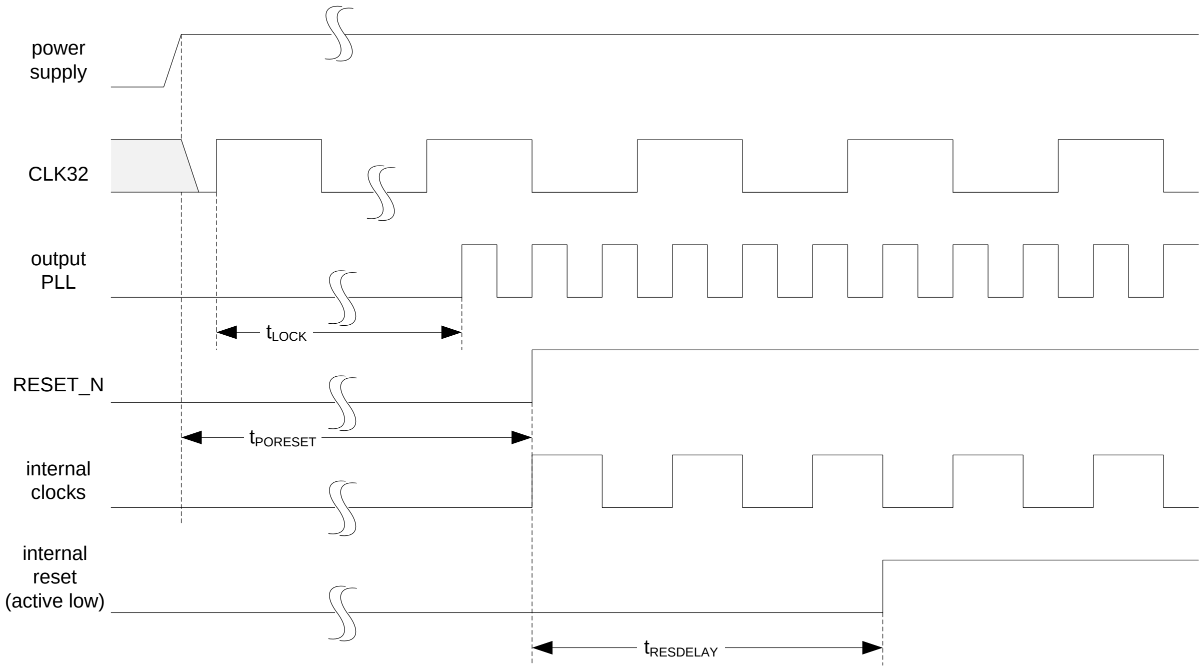


Table 6-1 Power-On Reset Timing

Parameter	Symbol	Value	Unit
PLL locking time	t_{LOCK}	< 50	µs
power-on reset	$t_{PORESET}$	> 50	µs
time between release of Power-On reset and release of internal reset signals	$t_{RESDELAY}$	< 173.6	ns

If the ANTAIOS is powered up and running the hardware reset can be used to perform a system reset. After removal of the external reset the internal reset signals remain active for $t_{RESDELAY}$.

Figure 6-3 Hardware Reset Sequence

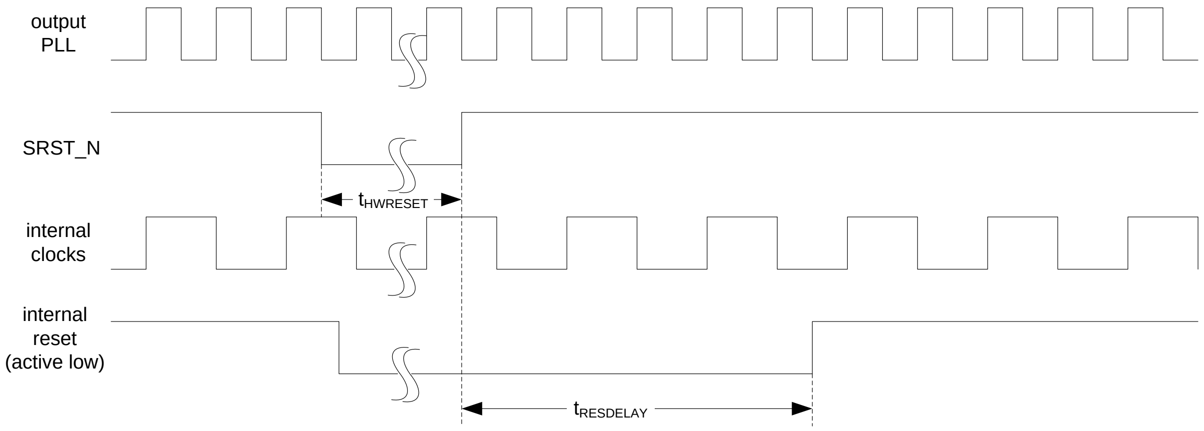


Table 6-2 Hardware Reset Timing

Parameter	Symbol	Value	Unit
hardware reset	$t_{HWRESET}$	> 10	ns
time between release of hardware reset and release of internal reset signals	$t_{RESDELAY}$	< 173.6	ns

7 Operation Specification

7.1 Absolute Maximum Ratings

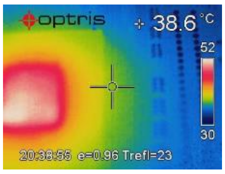
The permanent damage on a device may occur if the absolute maximum ratings are exceeded. These are only the stress ratings. The functional operations should be restricted within the conditions detailed in chapter Recommended Operating Conditions of this datasheet. Exposure to the absolute maximum rating conditions for extended period of time may affect the reliability of this device.

Table 7-1 Absolute Maximum Ratings

Parameter	Symbol	MIN	TYP	MAX	Unit
Core power supply	VCC_CORE VCC12A_DLL_DDRD VCC12A_PHY VCC12D_PHY VCC12AD_PLL	-0.5	1.2	1.6	V
IO power supply 3.3 V	VCC3IO VCC3IO-1 VCC33A_PHY VCC33A_USB	-0.5	3.3	4.6	V
Input voltage	V_{IN}	-0.5	-	4.6	V
Output short circuit current	I_{OUT}		50		mA
DC input current	I_{IN}		50		mA
Operating junction temperature	T_J	-40	25	125	°C
Storage temperature	T_{STG}	-65	-	150	°C

7.2 Thermal Characteristics

Figure 7-1 Thermal Characteristics

Parameter	Package	Symbol	Value	Unit
thermal resistance junction-to-case (for usage with heat sink)	TFBGA-380	θ_{JC}	4.0	K/W
	TFBGA-385		3.8	
thermal resistance junction-to-ambient (PCB with 6 layers, 1.6 mm thickness and dimensions of 101.5 x 45 mm) (for usage without heat sink)	TFBGA-380	θ_{JA}	16.3	K/W
	TFBGA-385		15.6	
thermal characterization parameter junction-to-top-center-of-package (for usage without heat sink)	TFBGA-380	Ψ_{JT}	0.1	K/W
	TFBGA-385		0.1	
Preliminary: ΔT ambient-to-case (P=1590 mW, still air, horizontal mounted, TFBGA-380, T_A=26 °C)	TFBGA-380	ΔT_{AC}	28.7	K
	TFBGA-385		25.7	
Preliminary: ΔT ambient-to-PCB nearby edge of ANTAIOS package  (P=1590 mW, still air, horizontal mounted, TFBGA-380, T_A=26 °C)	TFBGA-380	ΔT_{AB}	13.8	K
	TFBGA-385		12.6	

Thermal resistances θ_{JC} and θ_{JA} are intended mainly for performance comparison of the package variants. θ_{JA} should be used only for 1st order approximation of performance (i.e., T_J rise above T_A), as it highly depends on board design.

Equation 7-1 Approximation of Junction Temperature

$$T_J = T_A + \theta_{JA} \cdot P$$

θ_{JA} Theta JA is the thermal resistance between junction and ambient air

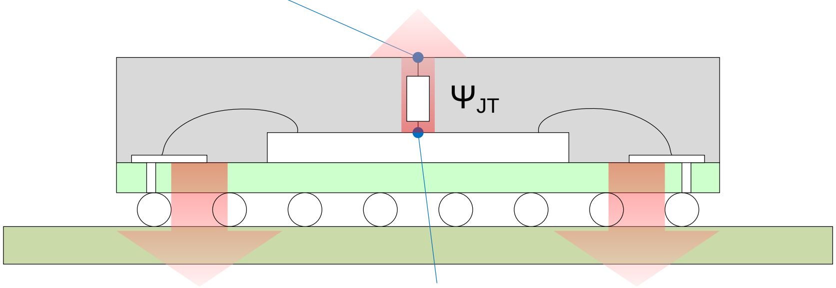
T_A ambient air temperature

T_J junction temperature

P power dissipated by device

7.2.1 Usage without Heat Sink

Figure 7-2 Heat Flow (without heat sink)



The heat caused by power consumption results in an increased temperature of the component. The heat dissipates from its source (junction J) via the chip and the case (C), and simultaneously via the balls and the board (B), to the ambient air (A). One important thing to note is that only a few heat flows through the top of the package.
When the hottest temperature at the top of the component as well as the power consumption is measured, the junction temperature can be calculated using the thermal characterization parameter Ψ_{JT}.

Equation 7-2 Junction Temperature (without heat sink)

$$T_J = T_T + \Psi_{JT} \cdot P$$

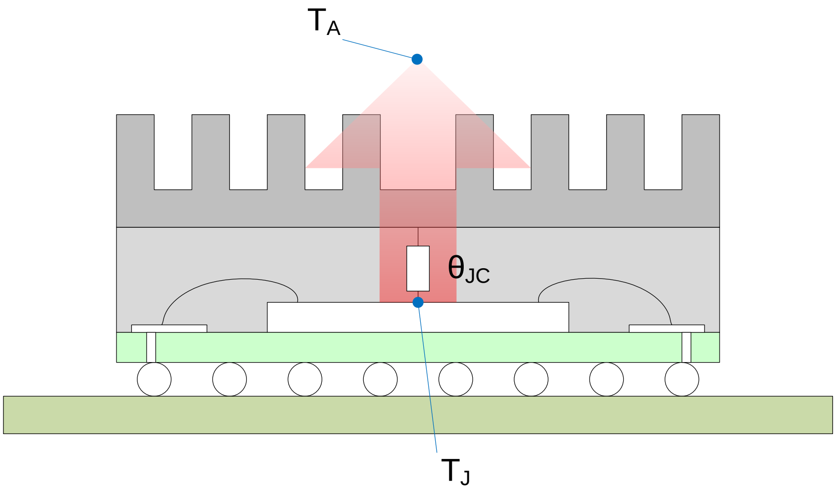
Ψ_{JT} Psi JT is the thermal characterization parameter between junction to top center of the package. It doesn't represent a thermal resistance, but instead is a characteristic parameter that can be used to convert between T_J and T_T when knowing the power dissipated by device.
T_J junction temperature
T_T top of package temperature at center
P power dissipated by device
Note that the equation uses the device's total power dissipation, it isn't necessary to know the power distribution going to the top of the package or to the board.
For example with P=1.59 W, T_A=26 °C and ΔT_{AC}=28.7 °C, the junction temperature results in

$$T_J = (T_A + \Delta T_{AC}) + \Psi_{JT} \cdot P = 26\,°C + 28.7\,K + 1.59\,W \cdot 0.1\frac{K}{W} = 54{,}86\,°C$$

T_A ambient air temperature
ΔT_{AC} difference in temperature between ambient air and top center of the package

7.2.2 Usage with Heat Sink

Figure 7-3 Heat Flow (with heat sink)



By usage of a heat sink the main heat dissipates through the case (C) and the heat sink (S) to the ambient air (A). The junction temperature can be calculated by using the thermal resistances as follows:

Equation 7-3 Junction Temperature (with heat sink)

$$T_J = T_A + (\theta_{JC} + \theta_{CS} + \theta_{SA}) \cdot P$$

θ_{JC} Theta JC is the thermal resistance from die (junction) to the top of the package (case) mounted to a heat sink

θ_{CS} Theta CS is thermal resistance of the thermal interface material (TIM) between the ANTAIOS case and heat sink

θ_{SA} Theta SA is the thermal resistance of the heat sink

T_A ambient air temperature

T_J junction temperature

P power dissipated by device

7.3 Recommended Operating Conditions

Table 7-2 Recommended Operating Conditions

Parameter	Symbol	MIN	TYP	MAX	Unit
Core power supply	VCC_CORE VCC12A_DLL_DDRD VCC12A_PHY VCC12D_PHY VCC12AD_PLL	1.14	1.2	1.26	V
IO power supply 3.3 V	VCC3IO VCC33A_PHY VCC33A_USB	3.14	3.3	3.47	V
SSTL_18 (DDR2) power supply 1.8 V	VCC18O_DDR	1.71	1.8	1.89	V
SSTL_18 (DDR2) VREF for the receivers	VREF_SSTL18	0.49 * VCC18O_DDR	0.50 * VCC18O_DDR	0.51 * VCC18O_DDR	V

7.4 Power Dissipation

Test Conditions (Very heavy workload, similar to extreme EtherCAT master usage):

- 128 MB DDR-SDRAM
- 2 internal Ethernet PHYs (full utilization)
- 5 PPUs (100 % CPU load)
- ARM (100 % CPU load)

Table 7-3 Power Consumption

Parameter	Ambient Temperature	Voltage Supply	Symbol	Value	Unit
max. chip power dissipation, measured (conditions: w/o air flow and w/o heat sink)	T_A = 85 °C	+5% Voltage	P	1.95	W
max. chip power dissipation, simulated (conditions: junction temperature 125 °C)	worst case process		P	2.2	W

Table 7-4 Current Consumption (measured values)

Parameter	Ambient Temperature	Voltage Supply	Value	Unit
core power supply	T_A = 85 °C	1.26 V	988	mA
IO (LVTTL) power supply		3.47 V	114	mA
SSTL_18 (DDR2) power supply		1.89 V	159	mA

The following chapters give an overview of power consumption depending on different application cases.

7.4.1 PROFINET Device

The PROFINET Device reference platform is specified as follows:

- cycle time 250 µs
- communication via 2 Ethernet ports (PHYs in 100BASE-TX mode)
- PROFINET network with 1 PROFINET Controller and 5 PROFINET Devices
- w/o USB communication
- w/o SliceBus communication
- w/o CAN communication
- w/o PROFIBUS communication
- w/o MMC/SD

Table 7-5 Power Dissipation (PROFINET Device Application maximal)

Parameter	Ambient Temperature	Voltage Supply	Symbol	Value	Unit
chip power dissipation (conditions: w/o air flow and w/o heat sink)	T_A = 85 °C	+5% Voltage	P_{max}	1.62	W
		nominal Voltage	P_{typ}	1.44	W

Additional power dissipation of DDR-SDRAM needs to be considered.

Table 7-6 Current Consumption (PROFINET Device Application maximal)

Parameter	Ambient Temperature	Voltage Supply	Value	Unit
core power supply	T_A = 85 °C	1.26 V	894	mA
		1.2 V	820	mA
IO (LVTTL) power supply		3.47 V	121	mA
		3.3 V	119	mA
SSTL_18 (DDR2) power supply		1.89 V	37	mA
		1.8 V	34	mA

Table 7-7 Power Dissipation (PROFINET Device Application typical)

Parameter	Case Temperature	Voltage Supply	Symbol	Value	Unit
chip power dissipation (conditions: with air flow and heat sink)	T_C = 35 °C	+5% Voltage	P_{max}	1.45	W
		nominal Voltage	P_{typ}	1.32	W

Additional power dissipation of DDR-SDRAM needs to be considered.

Table 7-8 Current Consumption (PROFINET Device Application typical)

Parameter	Case Temperature	Voltage Supply	Value	Unit
core power supply	T_C = 35 °C	1.26 V	749	mA
		1.2 V	703	mA
IO (LVTTL) power supply		3.47 V	126	mA
		3.3 V	126	mA
SSTL_18 (DDR2) power supply		1.89 V	36	mA
		1.8 V	33	mA

7.4.2 EtherCAT Slave

The EtherCAT Slave reference platform is specified as follows:

- cycle time 50 µs
- communication via 2 Ethernet ports (PHYs in 100BASE-TX mode)
- EtherCAT network with 1 EtherCAT Master and 4 EtherCAT Slaves
- w/o USB communication
- w/o SliceBus communication
- w/o CAN communication
- w/o PROFIBUS communication
- w/o MMC/SD

Table 7-9 Power Dissipation (EtherCAT Slave Application typical)

Parameter	Case Temperature	Voltage Supply	Symbol	Value	Unit
chip power dissipation (conditions: with air flow and heat sink)	T_C = 35 °C	+5% Voltage	P_{max}	1.53	W
		nominal Voltage	P_{typ}	1.39	W

Additional power dissipation of DDR-SDRAM needs to be considered.

Table 7-10 Current Consumption (EtherCAT Slave Application typical)

Parameter	Case Temperature	Voltage Supply	Value	Unit
core power supply	T_C = 35 °C	1.26 V	767	mA
		1.2 V	718	mA
IO (LVTTL) power supply		3.47 V	141	mA
		3.3 V	139	mA
SSTL_18 (DDR2) power supply		1.89 V	39	mA
		1.8 V	36	mA

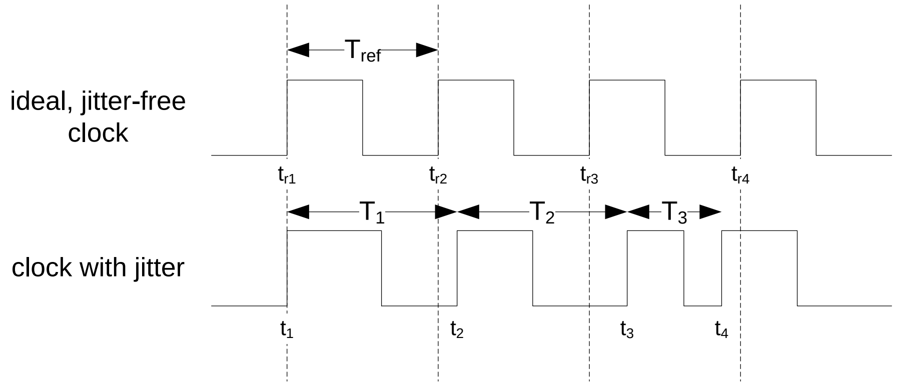
7.5 System Oscillator / PLL

The system oscillator circuit along with the internal PLL, generates all internal clocks of the ANTAIOS chip. For clock generation a crystal oscillator has to be connected to the clock input pin. Use a crystal oscillator for a fixed frequency instead of a programmable one, because of the lower jitter value.

Table 7-11 System Oscillator / PLL

Parameter		Symbol	MIN	TYP	MAX	Unit
System clock frequency		f_{CLK}	-	32	-	MHz
System clock tolerance					50	ppm
System clock duty cycle			20	-	80	%
System clock jitter	period jitter				95	ps
	cycle-to-cycle jitter				125	ps
	long term jitter (for a time period of 10 µs)				150	ps
System clock cycle time		T_{CYC}		31.25		ns
Locking time		T_{LOCK}	-	30	50	µs

Figure 7-4 Jitter Definition



Period jitter represents the difference in the clock period of the clock source from the ideal clock period.

Equation 7-4 Period Jitter

$$Period\ Jitter = T_i - T_{ref}$$

Cycle-to-cycle jitter represents the difference between two consecutive periods of the clock source.

Equation 7-5 Cycle-to-Cycle Jitter

$$Cycle - to - Cycle\ Jitter = T_{i+1} - T_i$$

Long term jitter represents the absolute difference in the clock edge from an ideal, jitter-free clock.

Equation 7-6 Long Term Jitter

$$Long\ Term\ Jitter = t_i - t_{ri}$$

7.6 Characteristics of 3.3 V LVTTL IO Cells

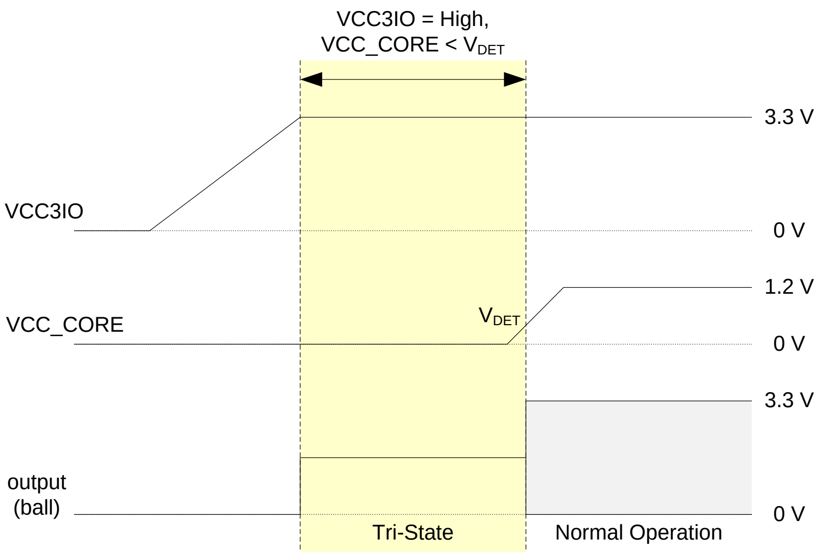
Table 7-12: Characteristics of LVTTL IO Cells

Parameter	Symbol	MIN	TYP	MAX	Unit
Input low voltage	V_{IL}	-	-	0.8	V
Input high voltage	V_{IH}	2.0	-	-	
Schmitt trigger negative-to-threshold voltage	V_{T-}	0.8	1.1	-	V
Schmitt trigger positive-to-threshold voltage	V_{T+}	-	1.6	2.0	V
Output low voltage $\|I_{OL}\|$ = 8 / 12 mA[1]	V_{OL}	-	-	0.4	V
Output high voltage $\|I_{OH}\|$ = 8 / 12 mA[1]	V_{OH}	2.4	-	-	V
Input pull-up resistance V_{IN} = 0 V	R_{PU}	40	75	190	kΩ
Input pull-down resistance V_{IN} = VCC3IO	R_{PD}	30	75	190	kΩ
Input leakage current V_{IN} = VCC3IO or 0 V	I_{IN}	-	±1.0	±10	µA
Tri-state output leakage current	I_{OZ}	-	±1.0	±10	µA
Operating junction temperature	T_J	-40	25	125	°C
Core detection voltage for power-on control VCC3IO = 3.3 V	V_{DET}	0.24	-	0.84	V
Input capacitance [2]	C_{IN}		2.73		pF

[1] driving strength of 8 or 12 mA depends on the I/O cell

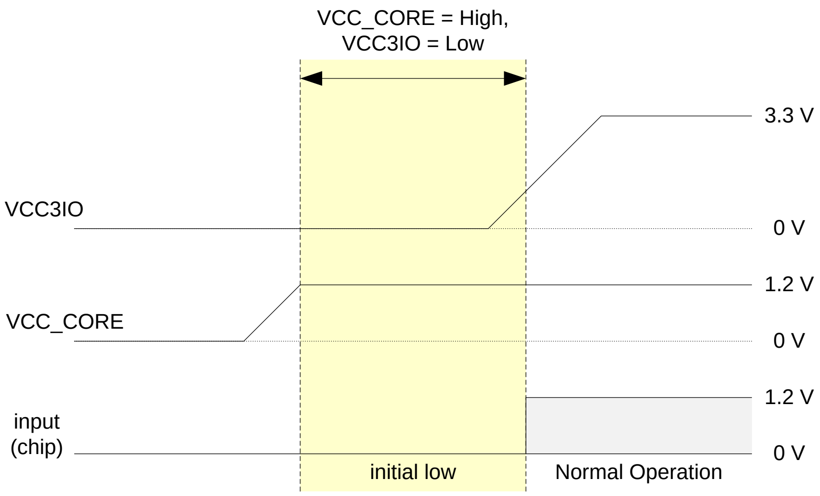
[2] C_{IN} includes the cell layout capacitance and the pad capacitance (estimated to be 0.5 pF)

Figure 7-5 Outputs: VCC3IO Ready and VCC_CORE Power-On Sequence



The LVTTL IO Cell will generate tri-state level (at the output and bidirectional IO cell pad) at the rising edge of the VCC3IO power when the VCC_CORE voltage is lower than V_{DET}. The figure above describes the output behavior of IO cells.

Figure 7-6 Inputs: VCC_CORE Ready and VCC3IO Power-On Sequence



For the input path, when the VCC3IO power supply comes after VCC_CORE, the input signal (at the input and bidirectional IO cell pad) will be initially low. The figure above describes the input behavior of the IO cells.

7.7 Characteristics of SSTL_18 (DDR2) IO Cells

Table 7-13 Characteristics of SSTL_18 IO Cells (Note: VREF=VREF_SSTL18)

Parameter	Symbol	MIN	TYP	MAX	Unit
I/O termination voltage (System)	V_{TT}	VREF - 0.04	VREF	VREF + 0.04	V
DC input high (Logic 1) voltage	V_{IH} (DC)	VREF + 0.125	-	VCC18O_DDR + 0.3	V
DC input low (Logic 0) voltage	V_{IL} (DC)	- 0.3	-	VREF – 0.125	V
AC input high (Logic 1) voltage	V_{IH} (AC)	VREF + 0.200	-	VCC18O_DDR + 0.5	V
AC input low (Logic 0) voltage	V_{IL} (AC)	-0.5	-	VREF - 0.200	V
Input differential voltage of differential input	V_{ID} (AC)	0.5	-	VCC18O_DDR	V
Cross-point of the AC differential input voltage	V_{IX} (AC)	0.5 * VCC18O_DDR – 0.175	-	0.5 * VCC18O_DDR + 0.175	V
On-Die Termination impedance value for ODTMD[1:0] = ”01” is 75 Ω[1]	R_{ODT1}	60	75	90	kΩ
On-Die Termination impedance value for ODTMD[1:0] = ”10” is 150 Ω[1]	R_{ODT2}	120	150	180	kΩ
On-Die Termination impedance value for ODTMD[1:0] = ”11” is 50 Ω[1]	R_{ODT3}	40	50	60	kΩ

[1] The measurement definition for the On-Die Termination impedance value: Apply VIH (AC) and VIL (AC) to the test pin separately and measure the current I (V_{IH} (AC)) and I (V_{IL} (AC)), respectively.
The On-Die Termination impedance value = (V_{IH} (AC) - V_{IL} (AC)) / (I (V_{IH} (AC)) – I (V_{IL} (AC)))

7.8 Characteristics of USB IO cells

Table 7-14 Characteristics of USB IO cells

Parameter		Symbol	MIN	TYP	MAX	Unit
Operating current of the VCCA_PLL domain High-Speed operation at 480 MHz		I_{VCCA_PLL}	-		10	mA
Operating current of the VCCA_HSRT domain High-Speed operation at 480 MHz		I_{VCCA_HSRT}	-	-	30	mA
Operating current of the VCC domain High-Speed operation at 480 MHz		I_{VCC}	-	-	10	mA
Operating current of the VCCA_PLL domain DP/DM with 15 kΩ to ground Pull-up resistor (1.5 kΩ) is disconnected USB PHY is in the suspend mode		$I_{SUS\_VCCA\_PLL}$	-	-	5.0	µA
Operating current of the VCCA_HSRT domain DP/DM with 15 kΩ to ground Pull-up resistor (1.5 kΩ) is disconnected USB PHY is in the suspend mode		$I_{SUS\_VCCA\_HSRT}$	-	-	5.0	µA
Operating current of the VCC domain DP/DM with 15 kΩ to ground Pull-up resistor (1.5 kΩ) is disconnected USB PHY is in the suspend mode		I_{SUS_VCC}	-	-	40	µA
USB 2.0 transceiver (HS)						
High-Speed differential input sensitivity $\lvert V_{I(DP)} - V_{I(DM)} \rvert$ measured when connected as an application circuit		V_{HSDIFF}	300	-	-	mV
High-Speed data signaling of common-mode voltage		V_{HSCM}	-50	-	500	mV
High-Speed squelch detection threshold	Squelch is detected	V_{HSSQ}	-	-	100	mV
	Squelch isn't detected		200	-	-	mV
High-Speed idle-level output voltage (differential)		V_{HSOI}	-10	-	10	mV
High-Speed low-level output voltage (differential)		V_{HSOL}	-10	-	10	mV
High-Speed high-level output voltage (differential)		V_{HSOH}	360	-	440	mV
Chirp-J output voltage (differential)		V_{CHIRPJ}	700	-	1100	mV
Chirp-K output voltage (differential)		V_{CHIRPK}	-900	-	-500	mV
Driver output impedance		R_{DRV}	40.5	45	49.5	Ω
Termination voltage with the connected pull-up resistor		V_{TERM}	3.0	-	3.6	V
USB 1.1 transceiver (FS)						
Differential input sensitivity $\lvert V_{I(DP)} - V_{I(DM)} \rvert$		V_{DI}	0.2	-	-	V
Differential common-mode voltage		V_{CM}	0.8	-	2.5	V
Single-ended receiver threshold		V_{SE}	0.8	-	2.0	V
output low voltage		V_{OL}	0	-	0.3	V
output high voltage		V_{OH}	2.8	-	3.6	V

7.9 Characteristics of PHYs

Table 7-15 Characteristics of internal PHYs

<table>
<tr><th colspan="2">Parameter</th><th>Symbol</th><th>MIN</th><th>TYP</th><th>MAX</th><th>Unit</th></tr>
<tr><td colspan="7">Total dissipative power</td></tr>
<tr><td colspan="2">100BASE-FX
(not including the TX current)</td><td></td><td>-</td><td>27</td><td>34</td><td>mW</td></tr>
<tr><td colspan="2">100BASE-TX</td><td></td><td>-</td><td>200</td><td>227</td><td>mW</td></tr>
<tr><td colspan="2">100BASE-TX
(not including the TX current)</td><td></td><td>-</td><td>70</td><td>82</td><td>mW</td></tr>
<tr><td colspan="2">10BASE-T</td><td></td><td>-</td><td>380</td><td>428</td><td>mW</td></tr>
<tr><td colspan="2">10BASE-T
(not including the TX current)</td><td></td><td>-</td><td>50</td><td>70</td><td>mW</td></tr>
<tr><td colspan="2">10BASE-Te</td><td></td><td>-</td><td>270</td><td>306</td><td>mW</td></tr>
<tr><td colspan="2">10BASE-Te
(not including the TX current)</td><td></td><td>-</td><td>50</td><td>70</td><td>mW</td></tr>
<tr><td colspan="2">Power-down mode</td><td></td><td>-</td><td>0.1</td><td>4.7</td><td>mW</td></tr>
<tr><td colspan="7">Transmitter characteristics</td></tr>
<tr><td rowspan="3">peak-to-peak differential output voltage</td><td>10BASE-T</td><td rowspan="3">$2 V_{txa}$</td><td>4.4</td><td>5.0</td><td>5.6</td><td>V</td></tr>
<tr><td>10BASE-Te</td><td>3.08</td><td>3.5</td><td>3.92</td><td>V</td></tr>
<tr><td rowspan="4">100BASE-TX</td><td>1.9</td><td>2.0</td><td>2.1</td><td>V</td></tr>
<tr><td>signal rise/fall time</td><td>t_r ; t_f</td><td>3.0</td><td>4.0</td><td>5.0</td><td>ns</td></tr>
<tr><td>output jitter
scrambled idle signal</td><td></td><td>-</td><td>-</td><td>1.4</td><td>ns</td></tr>
<tr><td>overshoot</td><td>V_{txov}</td><td>-</td><td>-</td><td>5.0</td><td>%</td></tr>
<tr><td colspan="7">Transmitter in fiber mode</td></tr>
<tr><td colspan="2">Output low voltage</td><td>V_{OL}</td><td>1.0</td><td>1.5</td><td>2.0</td><td>V</td></tr>
<tr><td colspan="2">Output high voltage</td><td>V_{OH}</td><td>2.0</td><td>2.4</td><td>2.75</td><td>V</td></tr>
<tr><td colspan="2">Differential output voltage</td><td>$|V_{OD}|$</td><td>0.52</td><td>0.83</td><td>1.3</td><td>V</td></tr>
<tr><td colspan="7">Receiver in fiber mode</td></tr>
<tr><td colspan="2">Input common-mode voltage</td><td>V_{icm}</td><td>1.67</td><td>2.0</td><td>2.33</td><td>V</td></tr>
<tr><td colspan="2">Input differential threshold voltage</td><td>$|V_{idth}|$</td><td>200</td><td>-</td><td>-</td><td>mV</td></tr>
<tr><td colspan="7">SD signals in fiber mode</td></tr>
<tr><td colspan="2">Copper mode</td><td></td><td>-</td><td>-</td><td>0.2</td><td>V</td></tr>
<tr><td colspan="2">Fiber mode without detected signal
generate far-end fault</td><td></td><td>1.0</td><td>-</td><td>1.8</td><td>V</td></tr>
<tr><td colspan="2">Fiber mode with detected signal</td><td></td><td>2.4</td><td>-</td><td>-</td><td>V</td></tr>
</table>

8 Timing Specification

8.1 AEI

Only single transfers are listed below. If transfer width is greater than data bus width, it is automatically transformed to a series of transfers. Please see 8.5 Timing Considerations of ANT1000/1001 User Manual for length calculation of the complete AEI transfer.

8.1.1 Master

8.1.1.1 Write Transfer

Figure 8-1 AEI Master – Write Transfer (Wait Signal not Used)

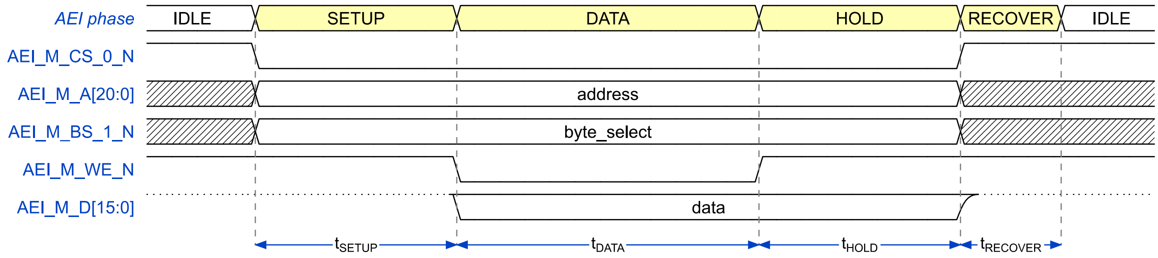


Figure 8-2 AEI Master – Write Transfer (Wait Signal Used)

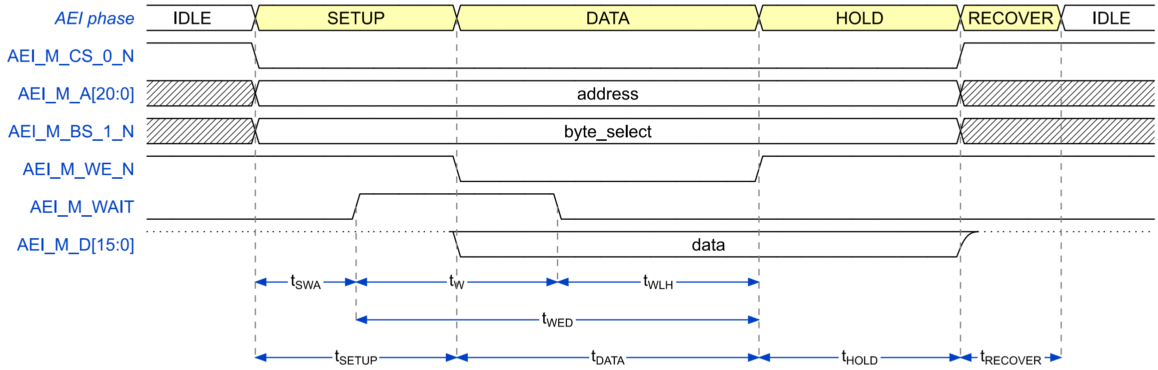


Table 8-1 AEI Master – Write Timing

Parameter	Symbol	Min	Max	Unit
Duration of AEI phases (with t_{CLK} = 1/200 MHz) (length of phase is adjustable via configuration register)	t_{SETUP} t_{DATA} t_{HOLD} $t_{RECOVER}$	1	127	t_{CLK}
AEI_M_WAIT enable to end of DATA phase; to allow detection of AEI_M_WAIT=1	t_{WED}	20	-	ns
AEI_WAIT active width	t_W	7	-	ns
Start of access to AEI_M_WAIT active	t_{SWA}	0	$t_{SETUP} + t_{DATA} -$ 20	ns
AEI_M_WAIT low to change to HOLD phase	t_{WLH}	10	20	ns

The following figures show the usage of the signals.

Figure 8-3 AEI Master – Write Transfer (Word Access, Always Even Addresses)

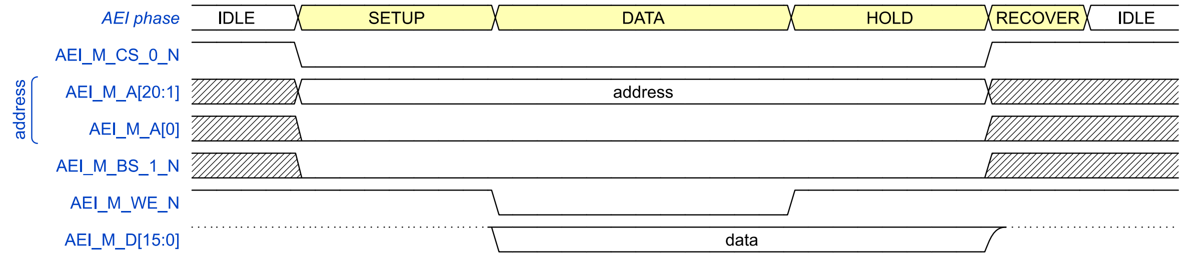


Figure 8-4 AEI Master – Write Transfer (Byte Access, Even Address)

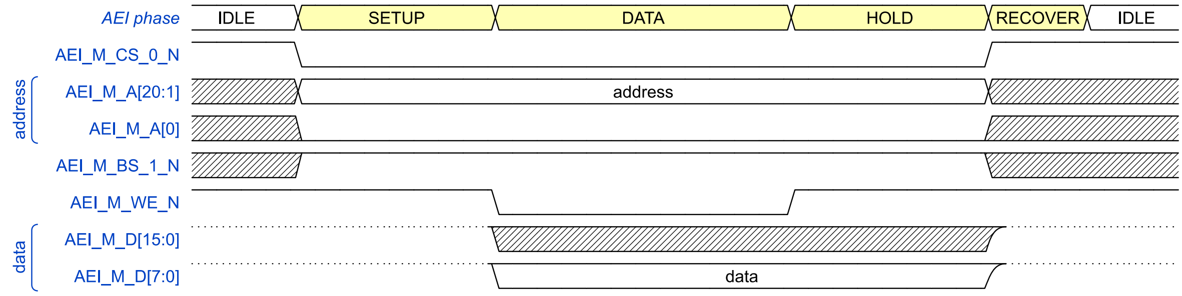
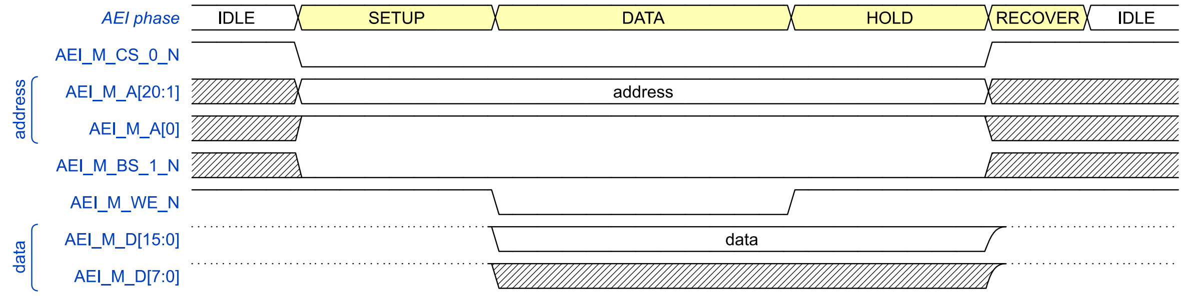


Figure 8-5 AEI Master – Write Transfer (Byte Access, Odd Address)



8.1.1.2 Read Transfer

Figure 8-6 AEI Master – Read Transfer (Wait Signal not Used)

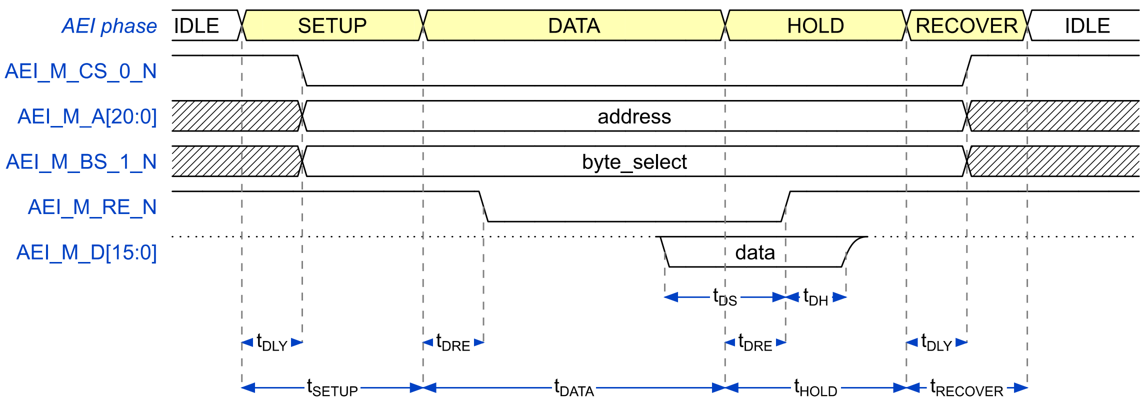


Figure 8-7 AEI Master – Read Transfer (Wait Signal Used)

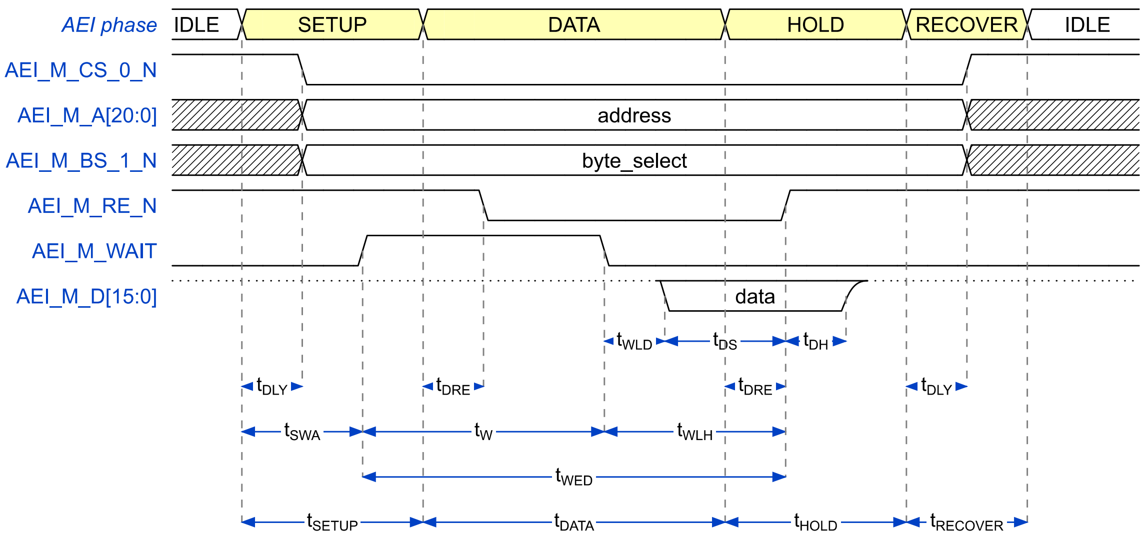


Table 8-2 AEI Master – Read Timing

Parameter	Symbol	Min	Max	Unit
Duration of AEI phases (with t_{CLK} = 1/200 MHz) (length of phase is adjustable via configuration register)	t_{SETUP} t_{DATA} t_{HOLD} $t_{RECOVER}$	1	127	t_{CLK}
Data setup time	t_{DS}	10	-	ns
Data hold time	t_{DH}	0	-	ns
change of AEI phase to change of select signals	t_{DLY}	-	6	ns
change of AEI phase to change of read enable	t_{DRE}	-	5	ns
AEI_M_WAIT enable to end of DATA phase; to allow detection of AEI_M_WAIT=1	t_{WED}	20	-	ns
AEI_M_WAIT active width	t_W	7	-	ns
AEI_M_WAIT low before data valid	t_{WLD}	-	10	ns
Start of access to AEI_WAIT active	t_{SWA}	0	t_{SETUP} + t_{DATA} – 20	ns
AEI_M_WAIT low to change to HOLD phase	t_{WLH}	10	20	ns

The calculation of the DATA phase for a READ transfer without using of wait signal is done as follows:

$$t_{DATA} = t_{DRE} + t_{RDAC} + t_{DS}$$

t_{RDAC} is the time from AEI_M_RE_N active to data valid (see chapter 8.1.2.2 AEI Slave Read Transfer).

The following figures show the usage of the signals.

Figure 8-8 AEI Master – Read Transfer (Word Access, Always Even Addresses)

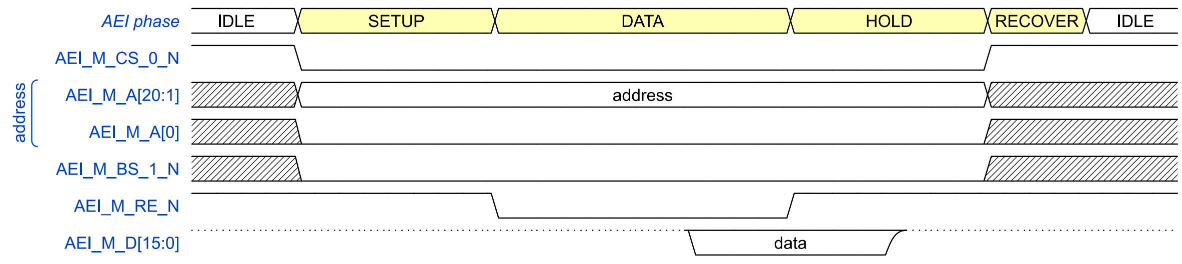


Figure 8-9 AEI Master – Read Transfer (Byte Access, Even Address)

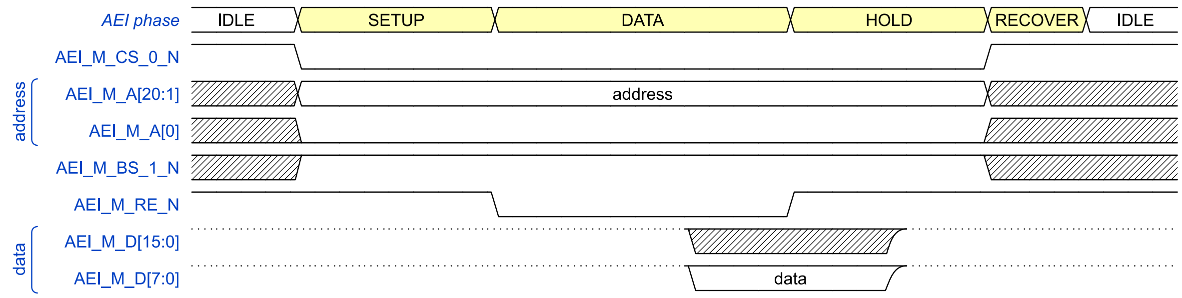
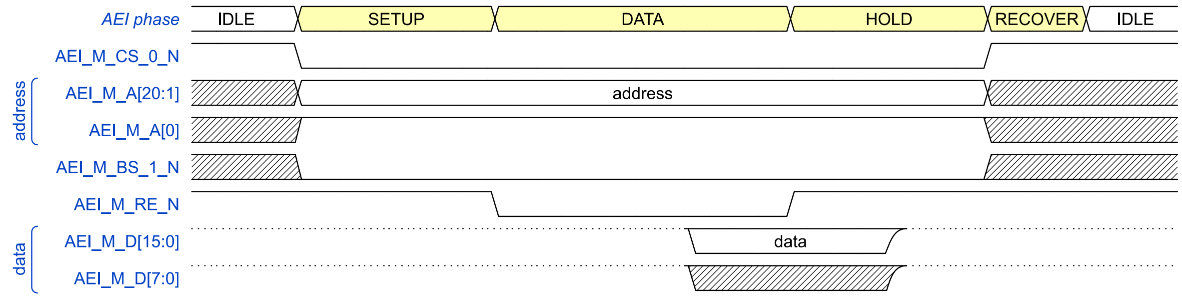


Figure 8-10 AEI Master – Read Transfer (Byte Access, Odd Address)



8.1.2 Slave

8.1.2.1 Write Transfer

Figure 8-11 AEI Slave – Write Transfer (Wait Signal not Used)

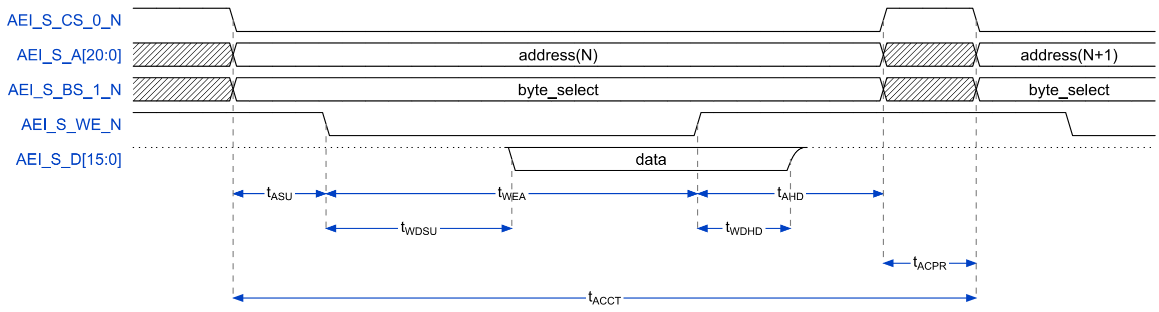


Figure 8-12 AEI Slave – Write Transfer (Wait Signal Used)

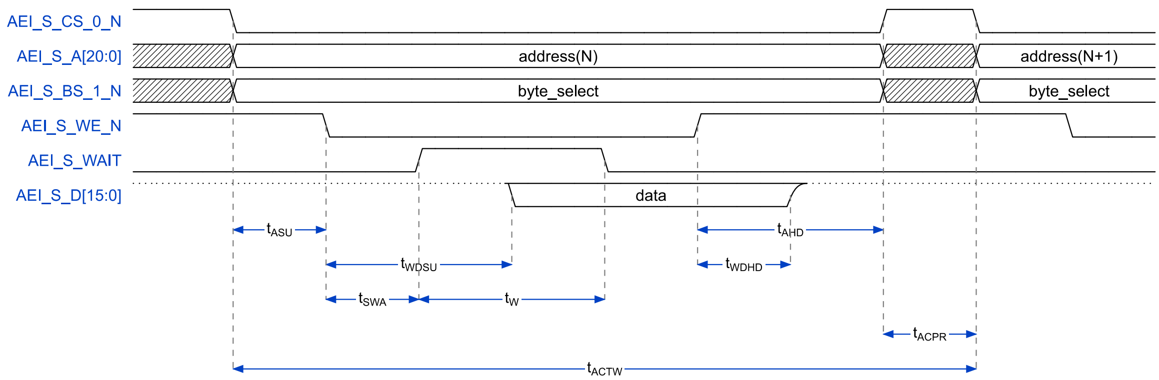


Table 8-3 AEI Slave – Write Timing

Parameter		Symbol	Min	Max	Unit
address setup time (address valid before write valid)		t_{ASU}	0	-	ns
address hold time (address valid after write invalid)		t_{AHD}	0	-	ns
write enable active (the value depends on the SRAM_WE_DELAY setting) [1]		t_{WEA}	25 + t_{WE_DELAY}	-	ns
data setup time (write data valid after write valid; the value depends on the SRAM_WE_DELAY setting) [1]		t_{WDSU}	8	8 + t_{WE_DELAY}	ns
write data hold time (write data hold after write invalid)		t_{WDHD}	0	-	ns
access cycle time (without wait)	no ARM access on CPU side of CI or FIFO interface [2]	t_{ACCT}	70	-	ns
	ARM access possible on CPU side of CI or FIFO interface [2]		120	-	ns
access time (with wait)	no ARM access on CPU side of CI or FIFO interface [2]	t_{ACTW}	-	70	ns
	ARM access possible on CPU side of CI or FIFO interface [2]		-	120	ns
access recovery time (independent on the access type)		t_{ACPR}	8	-	ns
start of access to AEI_S_WAIT active		t_{SWA}	-	36	ns
AEI_WAIT active width		t_{W}	5	90	ns

[1] t_{WE_DELAY}: value range of SRAM_WE_DELAY is 0..3 (= 0, 5, 10, 15 ns)

[2] The CPU side of Consistency Interface and FIFO interface is drawn at the bottom in the block diagram (see chapter 2).

The following figures show the usage of the signals.

Figure 8-13 AEI Slave – Write Transfer (Word Access, Always Even Addresses)

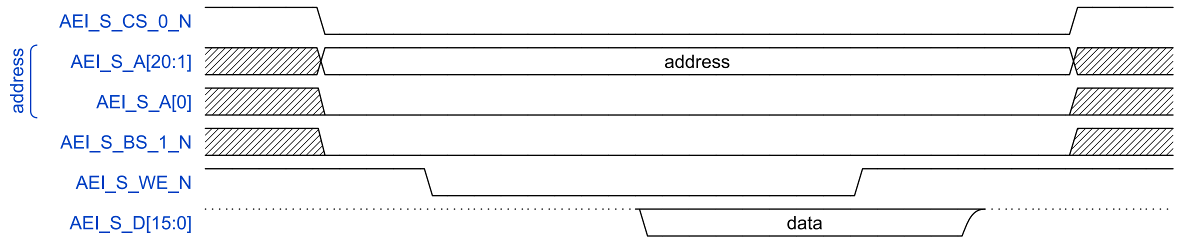


Figure 8-14 AEI Slave – Write Transfer (Byte Access, Even Address)

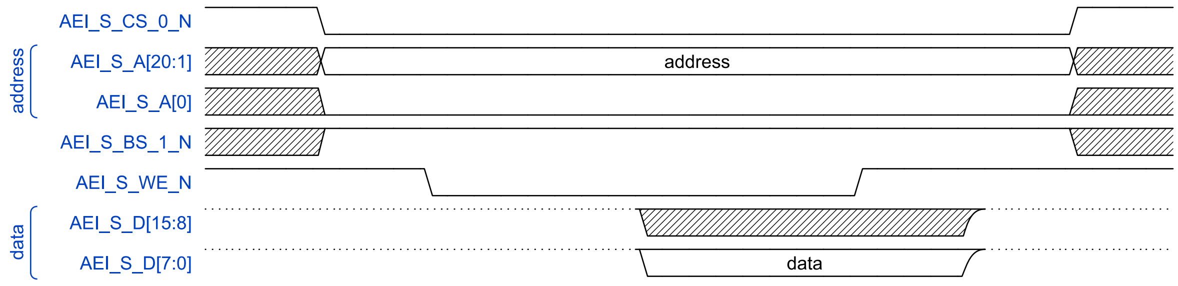
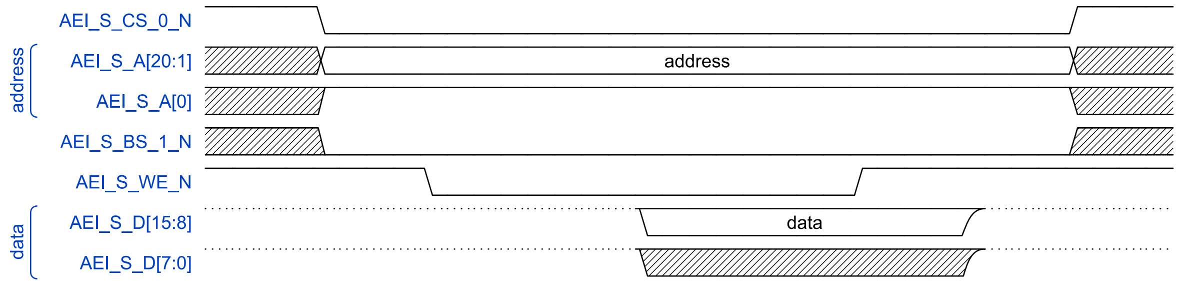


Figure 8-15 AEI Slave – Write Transfer (Byte Access, Odd Address)



8.1.2.2 Read Transfer

Figure 8-16 AEI Slave – Read Transfer (Wait Signal not Used)

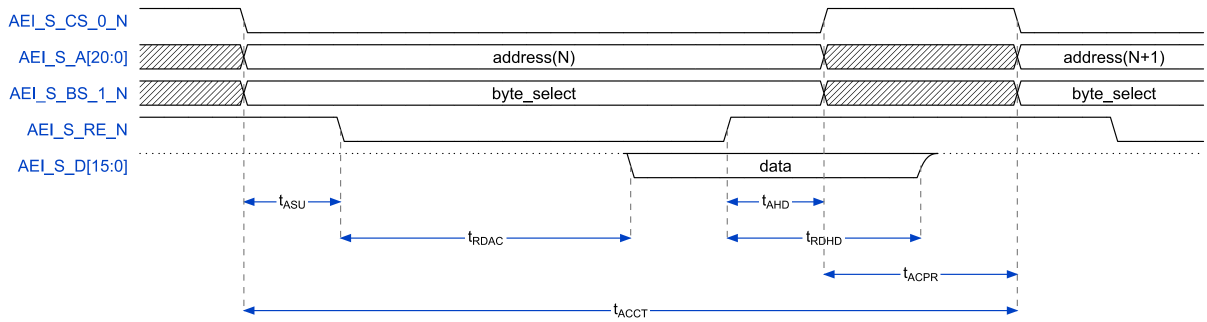


Figure 8-17 AEI Slave – Read Transfer (Wait Signal Used)

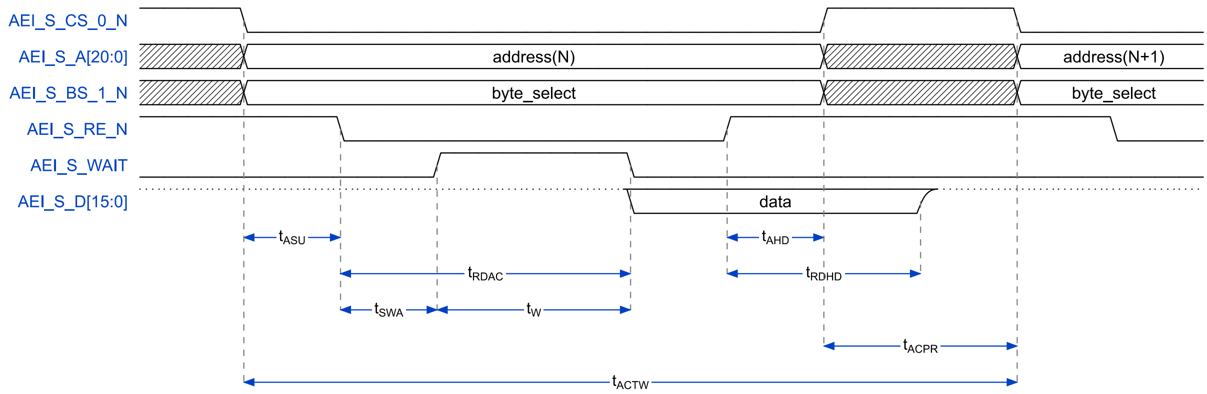


Table 8-4 AEI Slave – Read Timing

Parameter		Symbol	Min	Max	Unit
address setup time (address valid before read valid)		t_{ASU}	0	-	ns
address hold time (address valid after read invalid)		t_{AHD}	0	-	ns
data output delay time (data valid at the output)	no ARM access on CPU side of CI or FIFO interface [1]	t_{RDAC}	-	59	ns
	ARM access possible on CPU side of CI or FIFO interface [1]		-	116	ns
read data hold time (read data valid after read invalid)		t_{RDHD}	2	-	ns
access cycle time (without wait)	no ARM access on CPU side of CI or FIFO interface [1]	t_{ACCT}	70	-	ns
	ARM access possible on CPU side of CI or FIFO interface [1]		120	-	ns
access time (with wait)	no ARM access on CPU side of CI or FIFO interface [1]	t_{ACTW}	-	70	ns
	ARM access possible on CPU side of CI or FIFO interface [1]		-	120	ns
access recovery time (independent on the access type)		t_{ACPR}	8	-	ns
start of access to AEI_S_WAIT active		t_{SWA}	-	36	ns
AEI_WAIT active width		t_W	5	90	ns

[1] The CPU side of Consistency Interface and FIFO interface is drawn at the bottom in the block diagram (see chapter 2).

The following figures show the usage of the signals.

Figure 8-18 AEI Slave – Read Transfer (Word Access, Always Even Addresses)

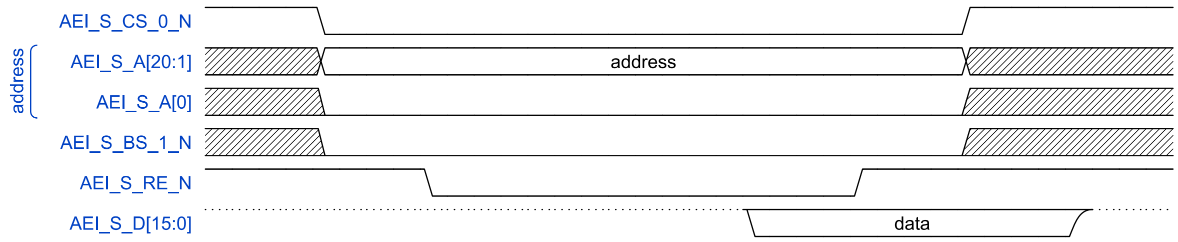


Figure 8-19 AEI Slave – Read Transfer (Byte Access, Even Address)

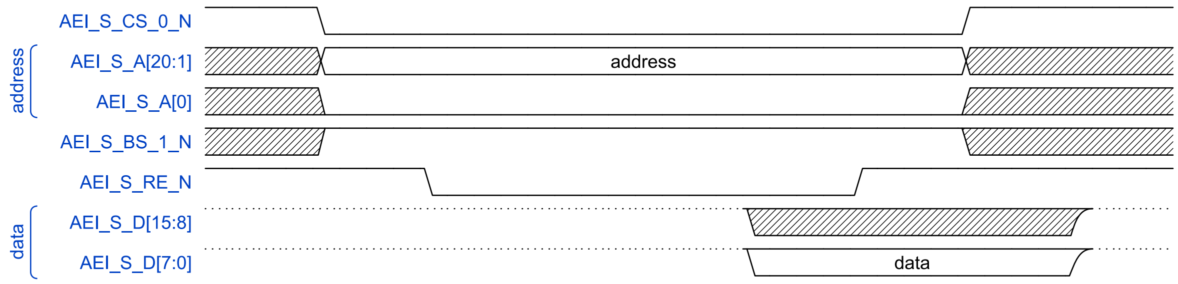
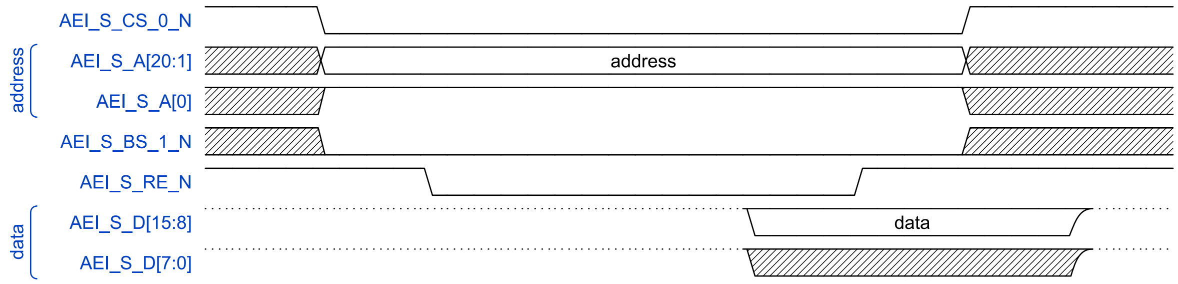


Figure 8-20 AEI Slave – Read Transfer (Byte Access, Odd Address)



8.2 NAND Flash Controller

The command sequence typically consists of a command state, address state and one or more data states (read or write).

Figure 8-21 NAND Flash – Command Latch Timing

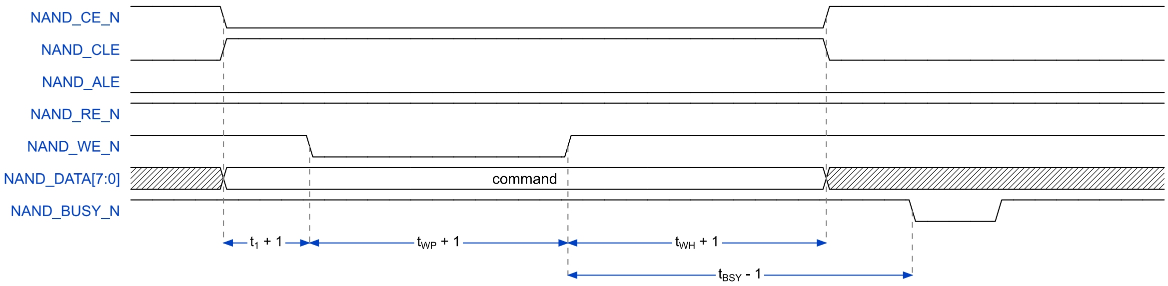


Figure 8-22 NAND Flash – Address Latch Timing

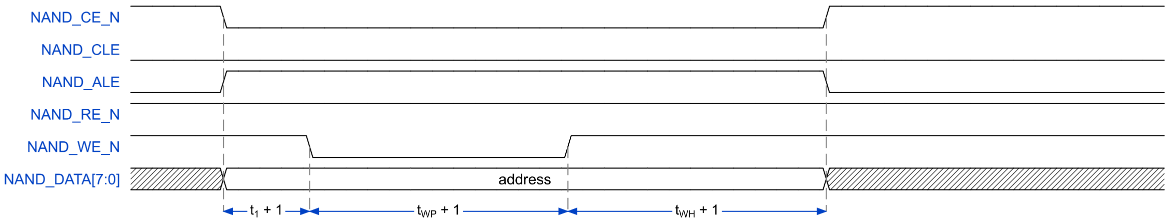


Figure 8-23 NAND Flash – Write Data Timing

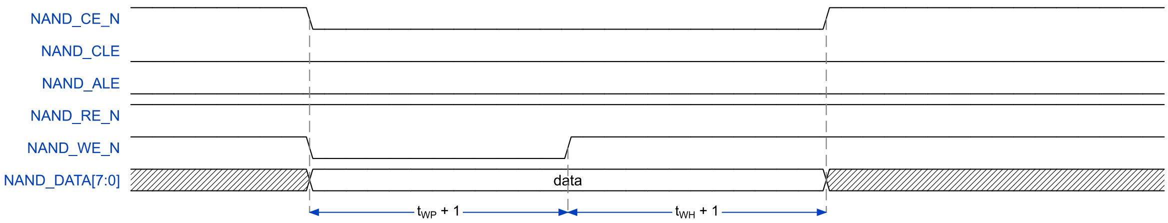


Figure 8-24 NAND Flash – Read Data Timing

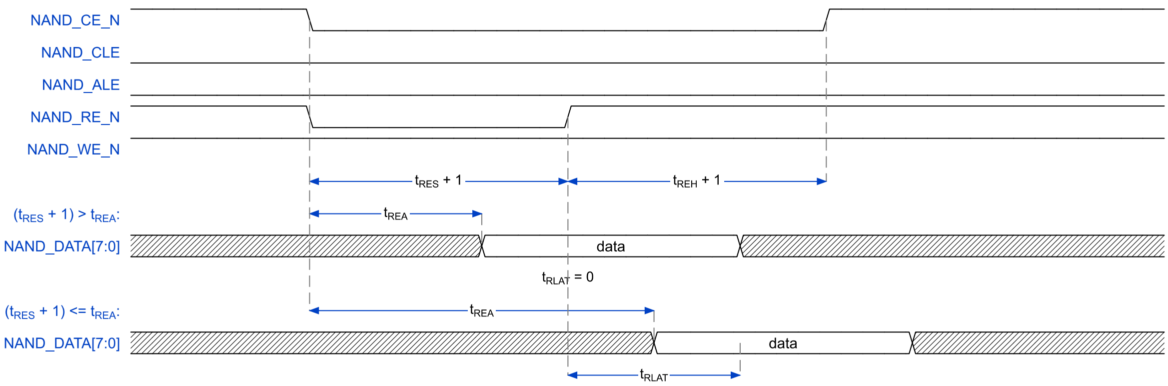


Table 8-5 NAND Flash – Timing (1)

Parameter	Symbol	Min	Max	Unit
NAND_WE_N high hold time (length of phase is adjustable via ACTR0N0 register) [2]	t_{WH} + 1	1	16	t_{MCLK} [1]
NAND_WE_N pulse width (length of phase is adjustable via ACTR0N0 register)	t_{WP} + 1	1	16	t_{MCLK} [1]
NAND_RE_N high hold time (length of phase is adjustable via ACTR0N0 register)	t_{REH} + 1	1	16	t_{MCLK} [1]
NAND_RE_N pulse width (length of phase is adjustable via ACTR0N0 register) [3]	t_{RES} + 1	1	16	t_{MCLK} [1]
NAND_CLE ↑ or NAND_ALE ↑ to NAND_WE_N ↓ (length of phase is adjustable via ACTR1N0 register) [4]	t_1 + 1	1	64	t_{MCLK} [1]
NAND_RE_N ↑ to sample data (length of phase is adjustable via ACTR1N0 register) [5]	t_{RLAT}	0	127	t_{MCLK} [1]
NAND_WE_N ↑ to NAND_BUSY_N ↓ (length of phase is adjustable via ACTR1N0 register) [6]	t_{BSY} - 1	0	126	t_{MCLK} [1]

All used timing parameters refer to the asynchronous/SDR data interface description of ONFi specification.

[1] t_{MCLK} = 1/192 MHz (period of NAND interface clock)

[2] t_{WH} + 1 =

Maximum value of the memory chip timing parameters t_{WH} (WE_N high hold time), t_{CH} (CE_N hold time), t_{CLH} (CLE hold time) and t_{ALH} (ALE hold time)

[3] **No EDO mode:**

t_{RES} + 1 =

Maximum value of the memory chip timing parameters t_{REA} (RE_N access time) and t_{RP} (RE_N pulse width)

EDO mode:

t_{RES} + 1 =

Memory chip timing parameter t_{RP} (RE_N pulse width)

[4] t_1 + 1 =

[Maximum value of the memory chip timing parameters t_{CS} (CE_N setup time), t_{CLS} (CLE setup time) and t_{ALS} (ALE setup time)] – [t_{WP} (WE_N pulse width) + 1]

[5] Timing parameter is used to shift the sampling if t_{RP} (RE_N pulse width) is smaller than t_{REA} (RE_N access time). t_{RP} of memory chip is equal to t_{RES} + 1.

[6] t_{BSY} -1 =

Memory chip timing parameters t_{WB} (WE_N high to busy)

Table 8-6 NAND Flash – Timing (2)

Parameter	Symbol	Min	Max	Unit
time between command/address phase to data phase (length of phase is adjustable via ACTR2N0 register) [2]	t_{BUF1} + 1	1	128	t_{MCLK} [1]
time before start reading output data (NAND_RE_N ↓) (length of phase is adjustable via ACTR2N0 register) [3]	t_{BUF2} + 1	1	128	t_{MCLK} [1]
time after a read access (length of phase is adjustable via ACTR2N0 register) [4]	t_{BUF3} + 1	1	128	t_{MCLK} [1]
time between writing command/address and start reading output data (length of phase is adjustable via ACTR2N0 register) [5]	t_{BUF4} + 1	1	128	t_{MCLK} [1]

All used timing parameters refer to the asynchronous/SDR data interface description of ONFi specification.

[1] t_{MCLK} = 1/192 MHz (period of NAND interface clock)

[2] t_{BUF1} + 1 =

Maximum value of the memory chip timing parameters t_{ADL} (ALE to data loading time) and t_{CCS} (change column setup time)

[3] t_{BUF2} + 1 =

Maximum value of the memory chip timing parameters t_{AR} (ALE to RE_N delay), t_{RR} (Ready to RE_N low (data only)) and t_{CLR} (CLE to RE_N delay).

[4] t_{BUF3} + 1 =

Maximum value of the memory chip timing parameters t_{RHW} (RE_N high to WE_N low) and t_{RHZ} (RE_N high to output hi-Z)

[5] t_{BUF4} + 1 =

Memory chip timing parameters t_{WHR} (WE_N high to RE_N low; used for Read ID, Read Status and Read Status Enhanced commands)

8.3 QuadSPI Controller

The QuadSPI Controller supports only the SPI transfer modes 0 and 3 (data sampling on the rising edge of SCK_OUT). This can be adjusted via SPI_CLK_MODE in the control register. For an overview about the SPI transfer modes please see following chapter of SPI controller.

Figure 8-25 QSPI - General Timing

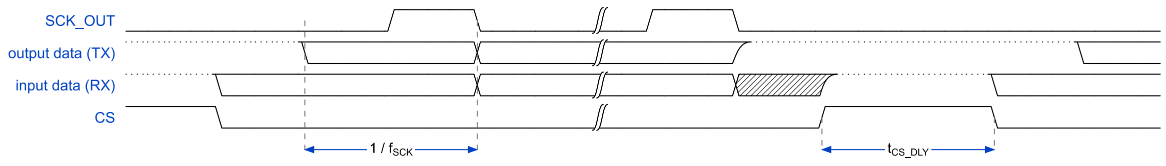


Figure 8-26 QSPI – Setup and Hold Timing

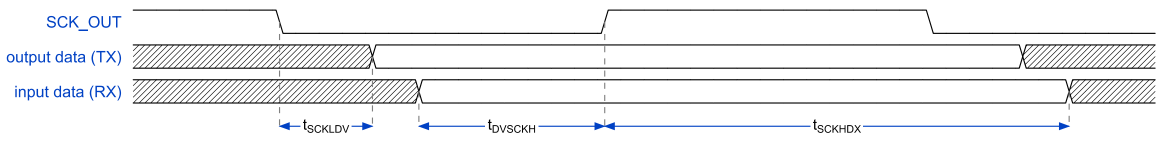


Table 8-7 QSPI – Timing

Parameter	Symbol	Min	Max	Unit
SCK_OUT clock frequency (frequency is adjustable via CR register)	f_{SCK}	24	96	MHz
CS deselect time after a command (length is adjustable via ACTR register)	t_{CS_DLY}	1	16	t_{SCK}
data in valid to SCK_OUT ↑ (input setup time)	t_{DVSCKH}	2	-	ns
SCK_OUT ↑ to data in invalid (input hold time)	t_{SCKHDX}	0	-	ns
SCK_OUT ↓ to data out valid	t_{SCKLDV}	-	0.8	ns

8.4 SD/MMC Card Controller

Figure 8-27 SD/MMC – Clock Timing

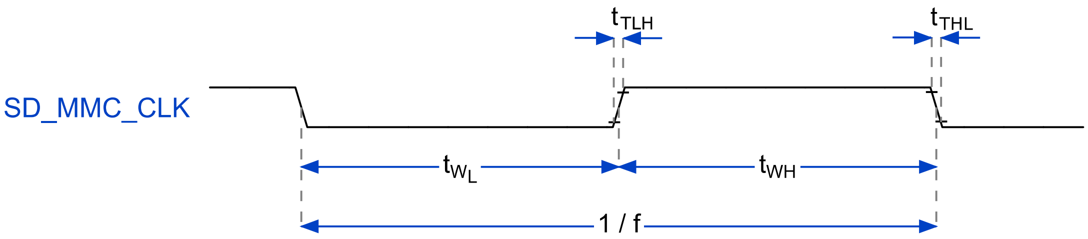


Table 8-8 SD/MMC – Clock Timing

Parameter			Symbol	Min	Max	Unit
SD_MMC_CLK clock frequency (frequency is adjustable via CCR register)[1]	identification mode (OD)		f_{OD}	0	400	kHz
	data transfer mode (PP)	default speed	f_{PP_DEF}	0	25	MHz
		high speed	f_{PP_HI}	0	50	MHz
default speed mode	clock low time		t_{WL}	tbd	-	ns
	clock high time		t_{WH}	tbd	-	ns
	clock rise time		t_{TLH}	-	tbd	ns
	clock fall time		t_{THL}	-	tbd	ns
high speed mode	clock low time		t_{WL}	tbd	-	ns
	clock high time		t_{WH}	tbd	-	ns
	clock rise time		t_{TLH}	-	tbd	ns
	clock fall time		t_{THL}	-	tbd	ns

[1] The base clock is specified by 100 MHz (period of module clock).

Figure 8-28 SD/MMC –Output Timing (ID Mode; Data Transfer – Default Speed)

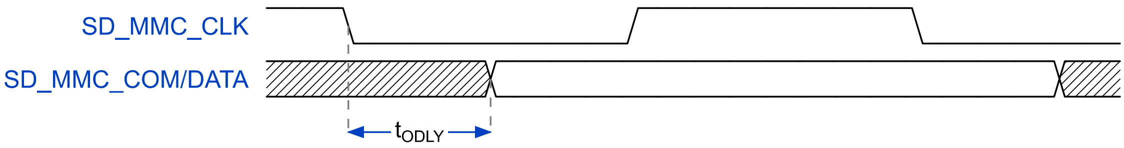


Figure 8-29 SD/MMC –Output Timing (Data Transfer– High Speed)

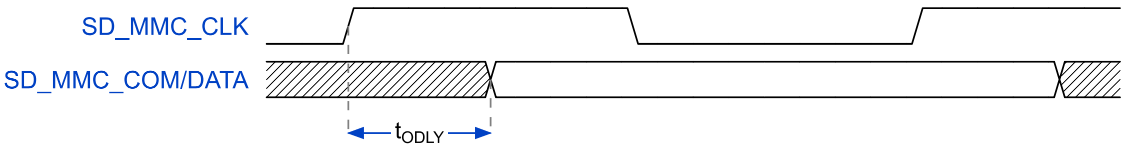


Figure 8-30 SD/MMC –Input Timing

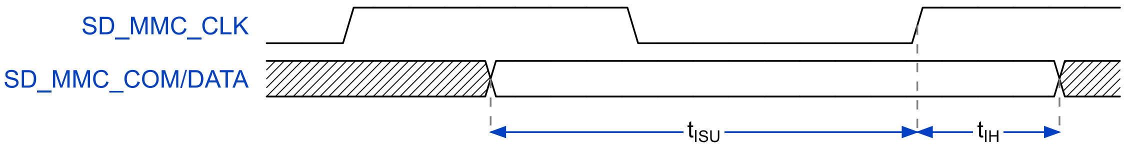


Table 8-9 SD/MMC – Input/Output Timing

Parameter			Symbol	Min	Max	Unit
Outputs CMD, DATA (referenced to CLK)						
output delay time	identification mode (OD)		t_{ODLY}	tbd	tbd	ns
	data transfer mode (PP)	default speed[1]		tbd	tbd	ns
		high speed[1]		tbd	tbd	ns
Inputs CMD, DATA (referenced to CLK)						
input setup time	identification mode (OD)		t_{ISU}	tbd	-	ns
	data transfer mode (PP)	default speed		tbd	-	ns
		high speed		tbd	-	ns
input hold time	identification mode (OD)		t_{IH}	tbd	-	ns
	data transfer mode (PP)	default speed		tbd	-	ns
		high speed		tbd	-	ns

[1] At which edge the outputs are updated is adjustable via INT_EDGE_SEL in the VR0 register.

8.5 SPI Controller

The SPI controller supports all SPI Transfer Formats. The mode can be configured by SCLKPO (SCLK polarity) and SCLKPH (SCLK phase) bits at SPICR0 register. The following pictures show the different modes, where the MSB is transferred first (LSB (bit sequence indicator) at SPICR0 register is set to zero).

Figure 8-31 SPI – Transfer Format Mode 0

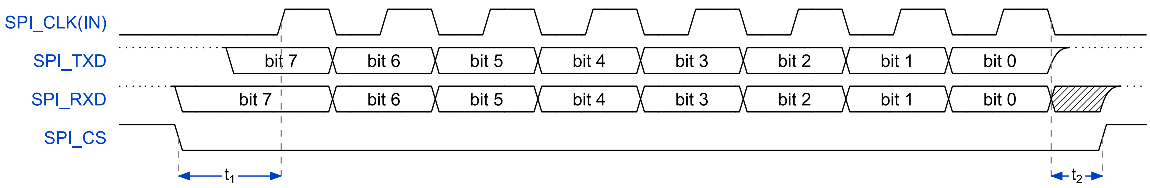


Figure 8-32 SPI – Transfer Format Mode 1

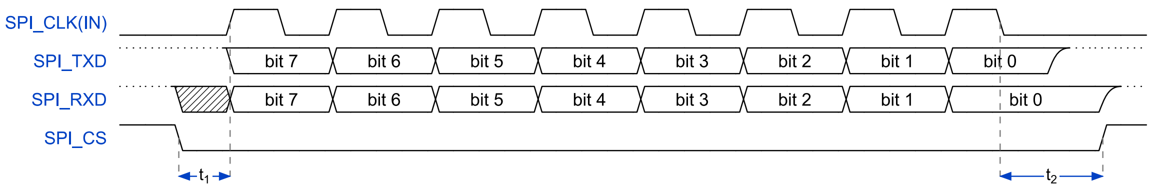


Figure 8-33 SPI – Transfer Format Mode 2

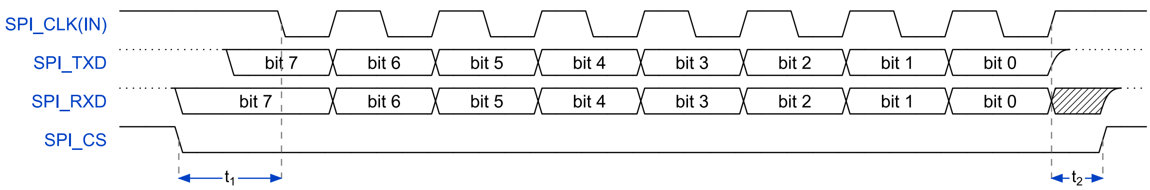


Figure 8-34 SPI – Transfer Format Mode 3

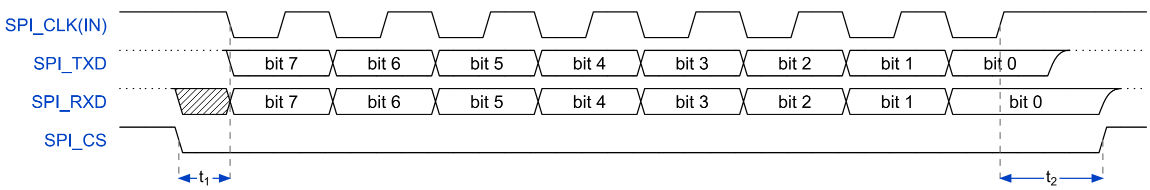
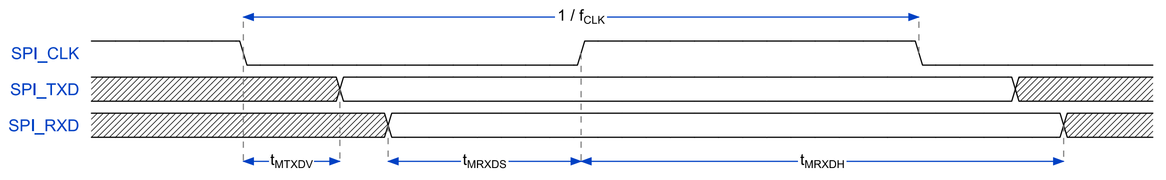


Table 8-10 SPI – Transfer Modes

Mode	SPICR0. SCLKPO	SPICR0. SCLKPH	Description				
			Idle State	Data Capturing	Data Output	t_1	t_2
0	0	0	low	CLK(IN) ↑	CLK(IN) ↓	CLK(IN) cycle	$\frac{CLK(IN)cycle}{2}$
1	0	1	low	CLK(IN) ↓	CLK(IN) ↑	$\frac{CLK(IN)cycle}{2}$	CLK(IN) cycle
2	1	0	high	CLK(IN) ↓	CLK(IN) ↑	CLK(IN) cycle	$\frac{CLK(IN)cycle}{2}$
3	1	1	high	CLK(IN) ↑	CLK(IN) ↓	$\frac{CLK(IN)cycle}{2}$	CLK(IN) cycle

Figure 8-35 SPI – Master Mode Timing

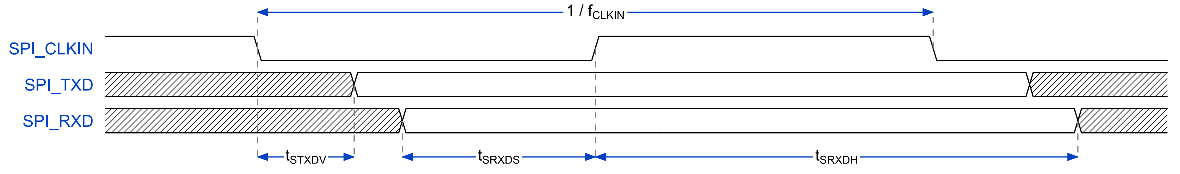


Above picture is drawn for mode 0 or 3, but the timing parameters have the same values in mode 1 and 2.

Table 8-11 SPI – Master Mode Timing

Parameter	Symbol	Min	Max	Unit
CLK clock frequency (frequency is adjustable via SPICR1 register)	f_{CLK}	1.2207	80 000	kHz
data in valid to CLK receive edge (input setup time)	t_{MRXDS}	tbd	-	ns
CLK receive edge to data in invalid (input hold time)	t_{MRXDH}	tbd	-	ns
CLK transmit edge to data out valid	t_{MTXDV}	-	tbd	ns

Figure 8-36 SPI – Slave Mode Timing



Above picture is drawn for mode 0 or 3, but the timing parameters have the same values in mode 1 and 2.

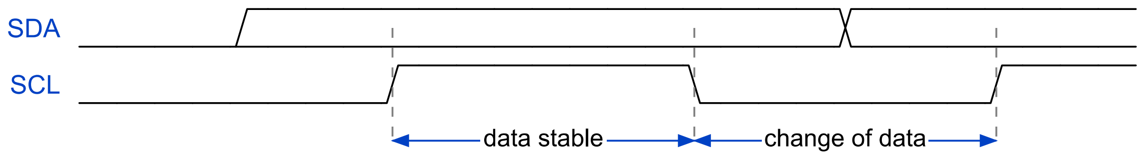
Table 8-12 SPI – Slave Mode Timing

Parameter	Symbol	Min	Max	Unit
CLK clock frequency (frequency is adjustable via SPICR1 register)	f_{CLKIN}	1.2207	24 000	kHz
data in valid to CLK receive edge (input setup time)	t_{SRXDS}	tbd	-	ns
CLK receive edge to data in invalid (input hold time)	t_{SRXDH}	tbd	-	ns
CLK transmit edge to data out valid	t_{STXDV}	-	tbd	ns

8.6 I²C Controller

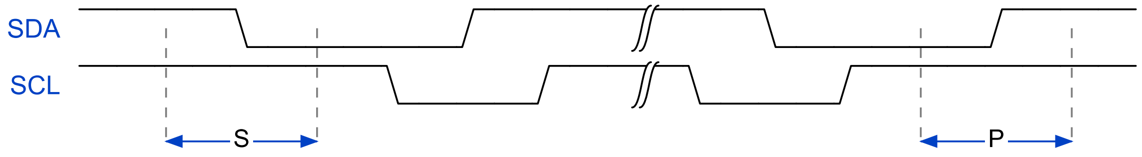
The data on the SDA line must be stable during the high period of the clock. The high or low state of the data line can only change when the clock signal on the SCL line is low. One clock pulse is generated for each data bit transferred.

Figure 8-37 I²C – Bit Transfer



All transactions begin with a START (S) and can be terminated by as STOP (P) condition. A high to low transition on the SDA line while SCL is high defines a START condition. A low to high transition on the SDA line while SCL is high defines a STOP condition.

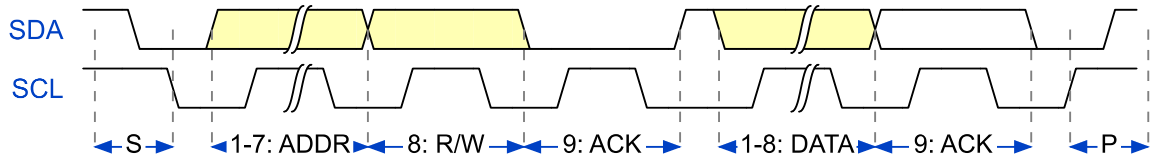
Figure 8-38 I²C – START and STOP conditions



START and STOP conditions are always generated by the master. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition. The bus stays busy if a repeated START (Sr) is generated instead of a STOP condition. In this respect, the START (S) and repeated START (Sr) conditions are functionally identical.

Every byte sent on the SDA line must be 8 bits long. The number of bytes that can be transmitted per transfer in unrestricted. Each byte has to be followed by an acknowledge bit. Data is transferred with the Most Significant Bit (MSB) first.

Figure 8-39 I²C – Data Transfer



Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit. The device that acknowledges, has to pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable low during the high period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. During reads, a master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave will leave the data line high to enable the master to generate the STOP condition.

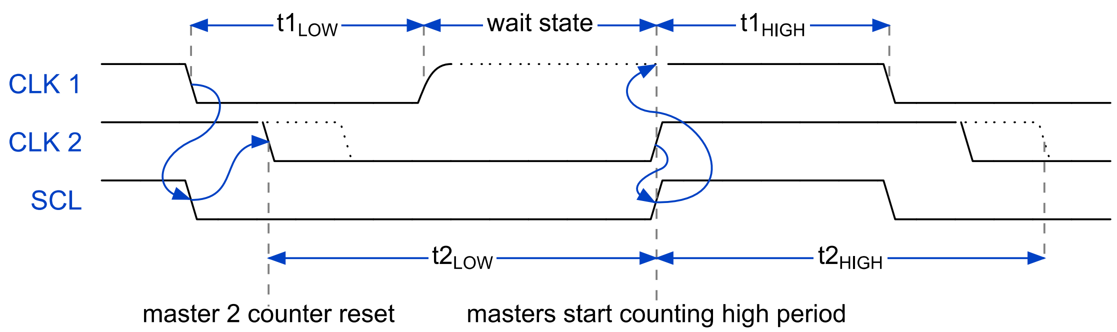
A control byte is the first byte received following the START condition from the master device. The control byte consists of a seven-bit slave address to select which device is

accessed. The last bit of the control byte defines the operation to be performed. When set to one, a read operation is selected. When set to zero, a write operation is selected.
The I²C protocol allows bus systems with several masters. Two problems arise in multi-master systems:

Synchronization

All masters generate their own clock on the SCL line to transfer messages on the bus. Data is only valid during high period of the clock. A defined clock is therefore needed for the bit-by-bit arbitration procedures to take place.

Figure 8-40 I²C – clock synchronization



Clock synchronization is performed using the wired-AND connection of I²C interfaces to the SCL line. This means that a high to low transition on the SCL line causes the masters concerned to start counting off their low period and, once a master clock has gone low, it will hold the SCL line in that state until the clock high state is reached. However, if another clock is still within its low period, the low to high transition of this clock may not change the state of the SCL line. The SCL line will therefore be held low by the device with the longest low period. Devices with shorter low periods enter a high wait-state during this time.
When all masters concerned have counted off their low period, the clock line will be released and goes high. There is then no difference between the master clocks and the state of the SCL line, and all the masters start counting their high periods. The first master to complete its high period pulls the SCL line low again.
In this way, a synchronized SCL clock is generated with its low period determined by the device with the longest clock low period, and its high period determined by the one with the shortest clock high period.

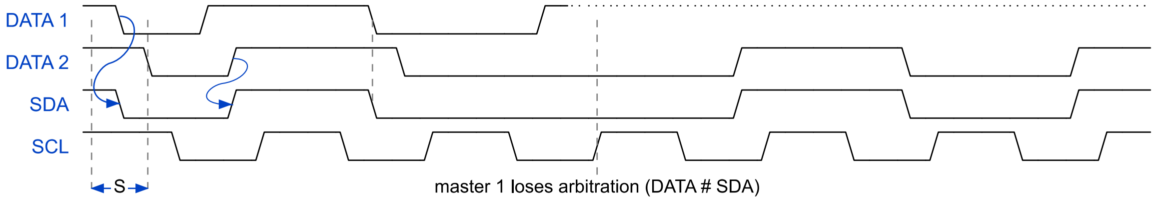
Arbitration

A master may start a transfer only if the bus is free. Two masters may generate a START condition within the minimum hold time ($t_{HD;STA}$) of the START condition which results in a valid START condition on the bus. Arbitration is then required to determine which master will complete its transmission.
Arbitration proceeds bit by bit. During every bit, while SCL is high, each master checks to see if the SDA level matches what it has sent. This process may take many bits. Two masters can actually complete an entire transaction without error, as long as the transmissions are identical. The first time a master tries to send a high, but detects that the SDA level is low,

the master knows that it has lost the arbitration and turns off its SDA output driver. The other master goes on to complete its transaction.

If several masters are trying to address the same slave, arbitration will continue into the data packet.

Figure 8-41 I²C – Arbitration



No information is lost during the arbitration process. A master that loses the arbitration can generate clock pulses until the end of the byte in which it loses the arbitration and must restart its transaction when the bus is free.

If a master also incorporates a slave function and it loses arbitration during the addressing stage, it is possible that the winning master is trying to address it. The losing master must therefore switch over immediately to its slave mode.

Since control of the I²C-bus is decided solely on the address and data sent by competing masters, there is no central master, nor any order of priority on the bus.

There is an undefined condition if the arbitration procedure is still in progress at the moment when one master sends a repeated START or a STOP condition while the other master is still sending data. In other words, the following combinations result in an undefined condition:

- master 1 sends a repeated START condition and master 2 sends a data bit
- master 1 sends a STOP condition and master 2 sends a data bit
- master 1 sends a repeated START condition and master 2 sends a STOP condition

Figure 8-42 I²C – Input/Output Timing (1)

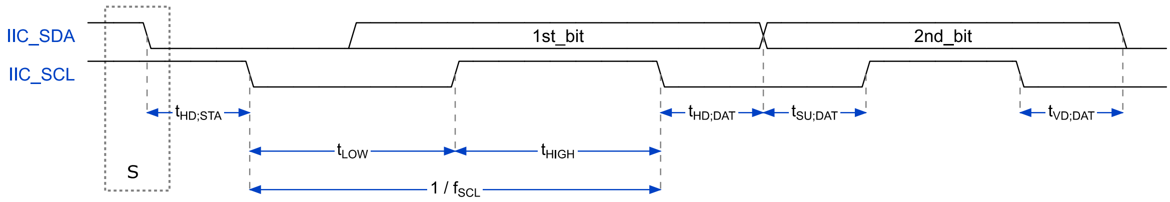


Figure 8-43 I²C – Input/Output Timing (2)

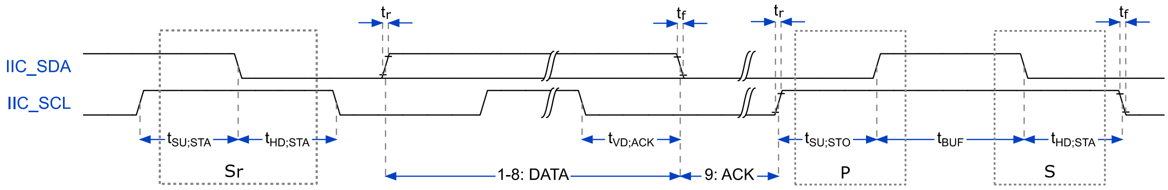


Table 8-13 I²C – Input/Output Timing

Parameter	Symbol	Min	Max	Unit
SCL clock frequency (frequency is adjustable via CDR and TGSR registers)[2]	f_{SCL}	91.55	400 000	Hz
tolerable spike width on bus (width is adjustable via TGSR register) [3]	t_{SW}	0	15	t_{APB_CLK} [1]
SCL and SDA rise time	t_r	-	tbd	ns
SCL and SDA fall time	t_f	-	tbd	ns
low period of the SCL clock	t_{LOW}	tbd	-	ns
high period of the SCL clock	t_{HIGH}	tbd	-	ns
hold time (repeated) START condition (After this period, the first clock pulse is generated)	$t_{HD;STA}$	tbd	-	ns
setup time for a repeated START condition	$t_{SU;STA}$	tbd	-	ns
data hold time	$t_{HD;DAT}$	tbd	-	ns
data setup time	$t_{SU;DAT}$	tbd	-	ns
setup time for STOP condition	$t_{SU;STO}$	tbd	-	ns
bus free time between a STOP and START condition	t_{BUF}	tbd	-	ns
data valid time (delay value is adjustable via TGSR register)[4]	$t_{VD;DAT}$	5	1042	t_{APB_CLK} [1]
data valid acknowledge time (delay value is adjustable via TGSR register) [4]	$t_{VD;ACK}$	5	1042	t_{APB_CLK} [1]

[1] t_{APB_CLK} = 1/48 MHz (period of module clock)

[2] $f_{SCL} = \frac{f_{APB\_CLK}}{2(COUNT+2)+GSR}$

[3] It must be observed: CDR > 3 + GSR + TSR

[4] The actual delay value is GSR + TSR + 4.

9 PCB Layout

9.1 Guidelines

9.1.1 DDR2 SDRAM

Termination:

- 50-60 Ω target impedance (Z_0) is recommended for all traces of single ended signals (address, command and control). FR-4 is commonly used for the dielectric material. The board thickness and trace width and thickness should be adjusted to match the impedance. Trace lengths are also influential, and they should be determined by simulation for each signal group and verified in test.
- DDR2 SDRAM offers on-die termination (ODT) features for bidirectional signals (DM, DQS and DQ). ODT improves signal quality while eliminating most of the external termination resistors.
- The clocks (CLK and CLK_N) are a differential signal and should be terminated at the end of the line between CLK and CLK_N with a 100-120 Ω resistor. If two DRAM chips are used, then the clock pair may be split into several segments. In the case of multiple segments, the termination resistor should be placed at the first split, or the resistor value should be increased and a resistor placed at the end of each segment.

For further information, please refer to the data sheets and application notes of the SDRAM manufacturer.

The rules below are based on the assumption of a signal slew rate of 1 V/1 ns. In slower applications. cross-talk generally is not a factor, and closer spacing may be allowed.

- Signals from the same net group should be routed on the same layer
- Signals form Byte group, such as DQS, DM and bits of DQ, must be routed in the same layer.
- The deviation of signal propagation delay is dependent on the timing budget on the application. The following values in the table are good examples at the start of a design.

Table 9-1 Deviation of Signal Propagation Delay

Signals on Net	maximum Deviation of Signal Propagation Difference	maximum Deviation of Trace Length
All data, address and command signals must be followed within this variation.	±50 ps	±6.635 mm (261 mil)
between CLK and CLK_N between DQS and DQS_N	±2 ps	±0.254 mm (10 mil)
between one clock pair and another clock pair, e.g. CLK/CLK_N and DQS/DQS_N	±5 ps	±0.635 mm (25 mil)
between signals within byte group (DQS, DM, 8 bits of DQ)	±10 ps	±1.270 mm (50 mil)

- Recommended trace width is 0.13 mm (5 mil).
- Intranet spacing, the distance between two adjacent traces within a net, is 0.2 mm (7 mil).
- Internet spacing, the distance between the two outermost signals of different signal group is 15 mil. The same rule applies between one clock pair and another clock pair.
- Differential clocks should be routed in parallel and keep the trace length short.
- Differential clocks must be routed on the same layer and placed on an internal layer minimize the noise.
- Keep the internet spacing rule between CKE and CLK/CLK_N.

9.1.2 Ethernet PHYs

In order to achieve the optimal performance, attention should be payed to the layout of the circuit board, shielding and placements of the components:

- The differential signals on both TX/RX should have equal length.
- The differential signals on both TX/RX should be close to each other.
- The PHY_TX_P/N and PHY_RX_P/N pairs for PCB trace must take care of impedance matching.
- There should not be any noisy signal close to the differential signals on both TX/RX.
- There should not be any noisy signal close to the bias resistor and crystal.
- The decoupling capacitors should be placed close to ANTAIOS nearby the PHY signals (close to Ethernet PHY component).
- In the PCB power plane, the analog power should be separated from the digital power and the analog ground should be separated from the digital ground.
- PHY_BIAS are the most important analog pins and users should keep the noisy source away from this pin.
- All PHY power pins (VCC12A_PHY, VCC12D_PHY and VCC33A_PHY), except VCC12CORE, belong to the analog power. Please feed these pins with good quality power supply.
- All PCB trace lengths of the Ethernet PHY should be as short as possible for better performance.
- To avoid the 100M overshoot, please place the 49.9 Ω resistors (R1, R2, R3, R4) and 0.1 µF capacitor (C1, C2) near the PHY_TX_P/N and PHY_RX_P/N pins of PHY. The distance between the resistor and chip output, PHY_TX_P/N, should be less than 3.0 cm. C3 and C4 should be placed close to the magnetics and R6 and R7 should be placed next to RJ-45 connector. Please refer to the application circuit as shown below.
- Make sure that the path connecting PHY_BIAS and the external reference resistor is less than 3.0 cm.

9.1.3 USB

Characteristic Impedance

The differential impedance means the ratio between the differential voltage and the effective differential current on a pair of traces in the differential mode. If the lines are far apart to have negligible direct coupling, the differential impedance equals to twice of the single-ended impedance. On the other hand, if the lines are coupled when the space narrows between a pair traces, the differential impedance will decrease. It is required to maintain the desired differential impedance.
The trace impedance to ground for USB_DP/DM signals should be 45 Ω ± 10 %. The impedance is 90 Ω ± 10 % between the differential signal pair, USB_DP and USB_DM, to match the 90 Ω ± 10 % of the USB twisted pair cable impedance. The trace impedance can be controlled by carefully selecting the line width, trace distance from power or ground planes, and physical proximity of nearby traces.

Place and Route Rules

The general routing and placement guidelines listed below should be followed to lay out a new design. These guidelines can help users minimize the EMI problems and improve the signal quality.

- Route USB_DP/DM signal traces with a minimum trace length and maintain the maximum distance between high-speed clocks and periodic signals to the USB differential pair and connectors leaving PCB.
- Route USB_DP/DM using a minimum of vias and corners to reduce the reflections and impedance changes.
- When it is necessary to turn 90°, users should use two 45° turns or an arc instead of making a single 90° turn. This reduces the signal reflections by minimizing the impedance discontinuities.
- Do not route USB traces under crystals, oscillators, clock synthesizers, magnetic devices, or ICs that use and/or duplicate clocks.
- Avoid stubs on the USB signals, as stubs will cause the signal reflections and affect the signal quality. When a stub is unavoidable in a design, the total stubs on a particular line should not be greater than 200 mils.
- Routing over an unbroken ground plane is preferred. If the unbroken ground plane is not available, route over an unbroken voltage plane. Avoid crossing over anti-etch when possible. Crossing over anti-etch (plane splits) increases inductance and radiation levels by forcing a greater loop area. Likewise, avoid changing layers with USB traces as much as practical. It is preferable to change layers to avoid crossing a plane split.
- The USB_DP/DM signal pair must be routed together, parallel to each other, on the same layer, and cannot be parallel with other non-USB signal traces.
- The USB_DP/DM signal traces must have equal length. It minimizes the EMI effect of the common-mode current.

9.2 Examples

9.2.1 TFBGA-380

Following figures show the different layers of a PCB, from top to bottom, for an industrial Ethernet application with the ANTAIOS TFBGA-380. This PCB layout uses microvias (blind or buried vias with a diameter ≤ 150 µm), which are recommended. The diameter of the BGA ball pads is 300 µm. The same size is used for the solder mask opening.

Figure 9-1 Top Layer

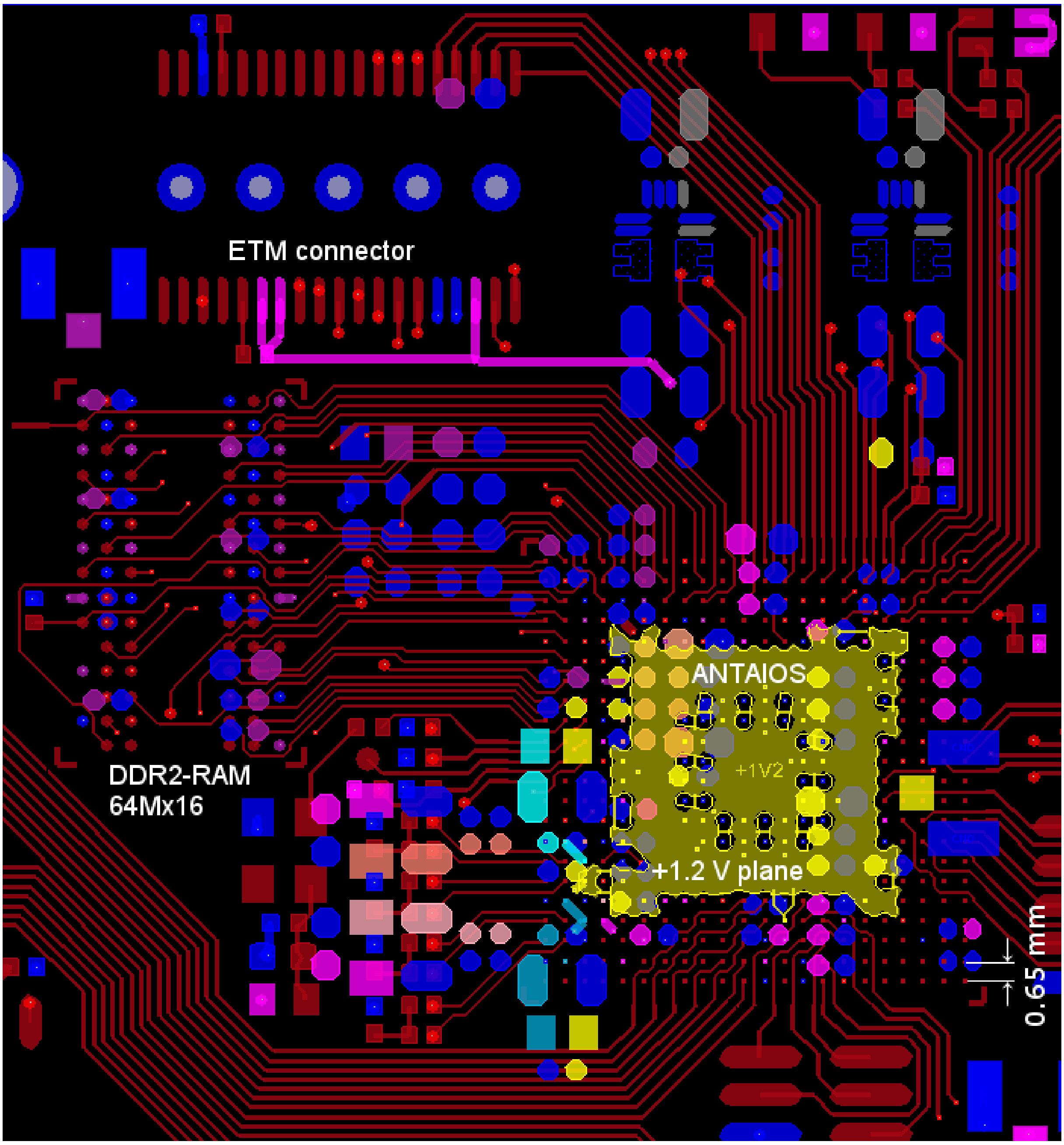


Figure 9-2 Internal Layer 1

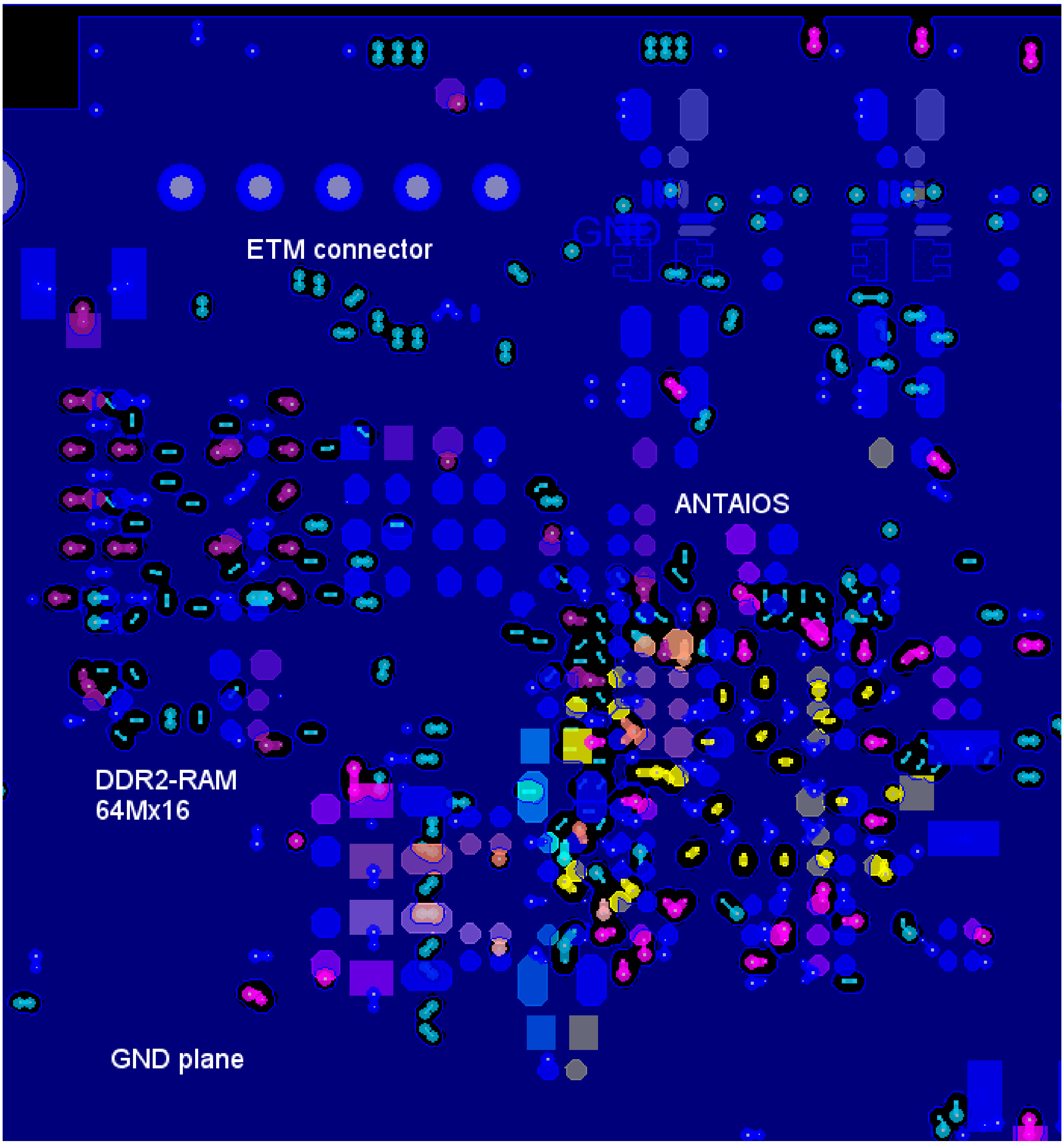


Figure 9-3 Internal Layer 2

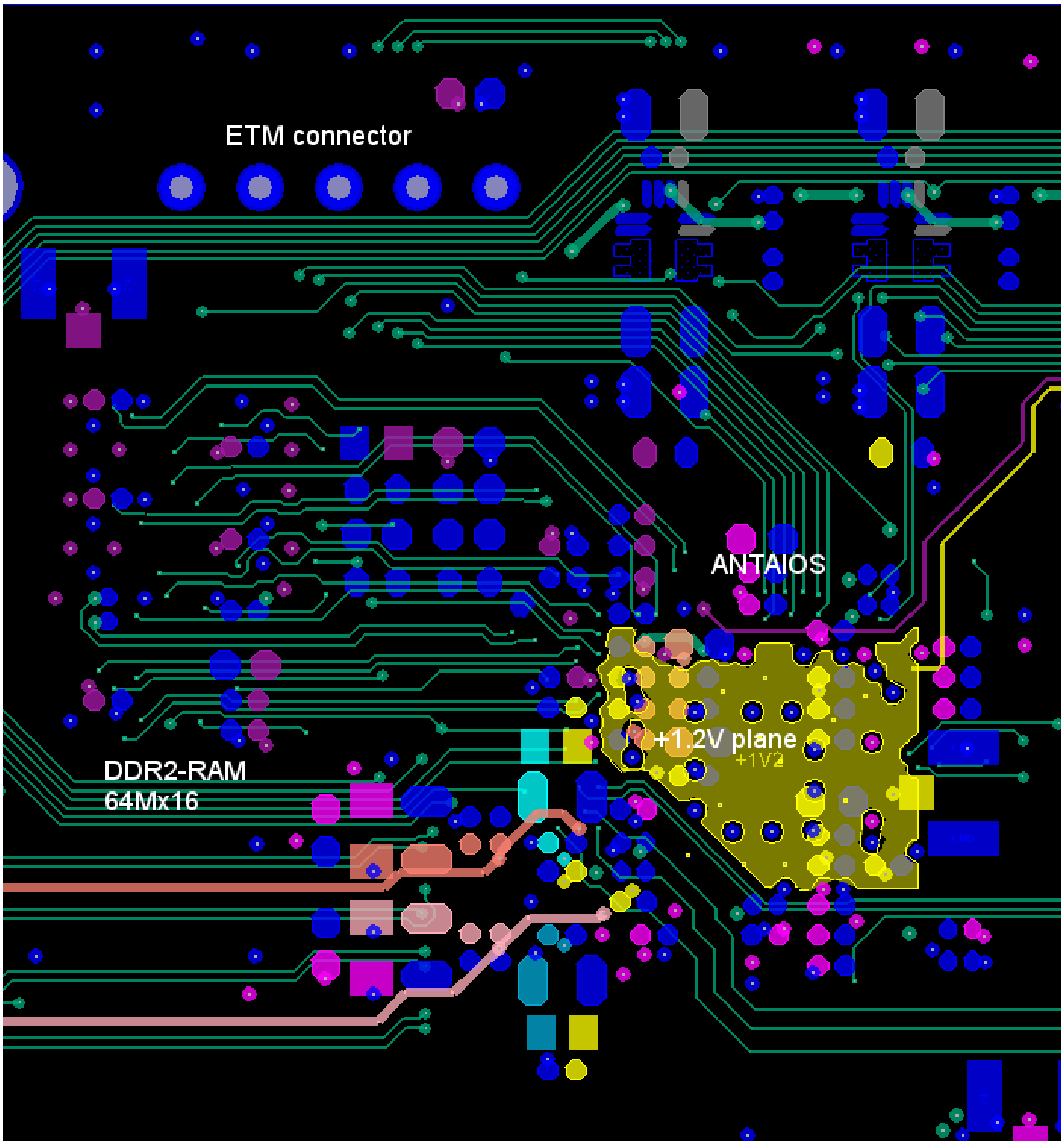


Figure 9-4 **Internal Layer 3**

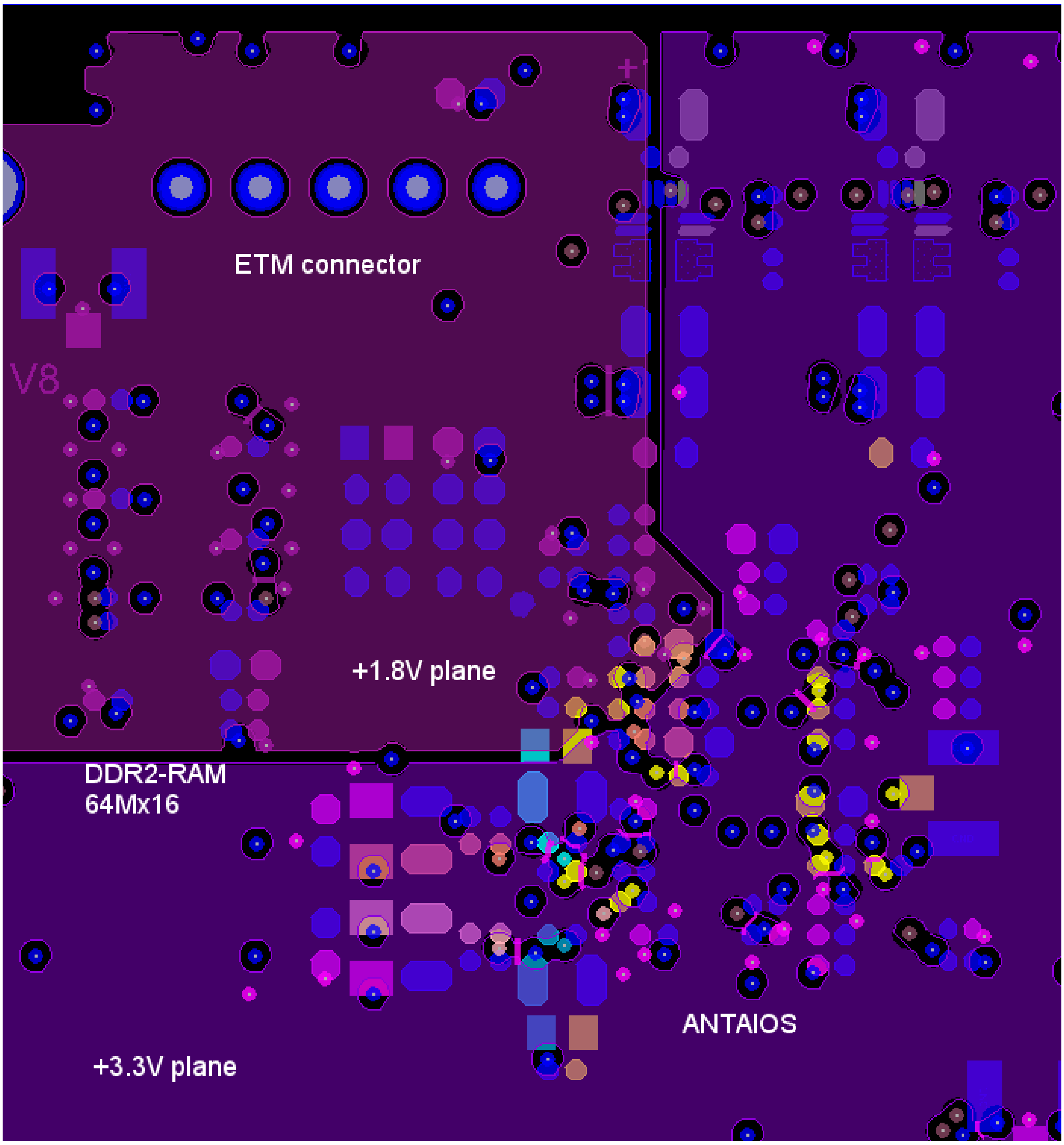


Figure 9-5 **Internal Layer 4**

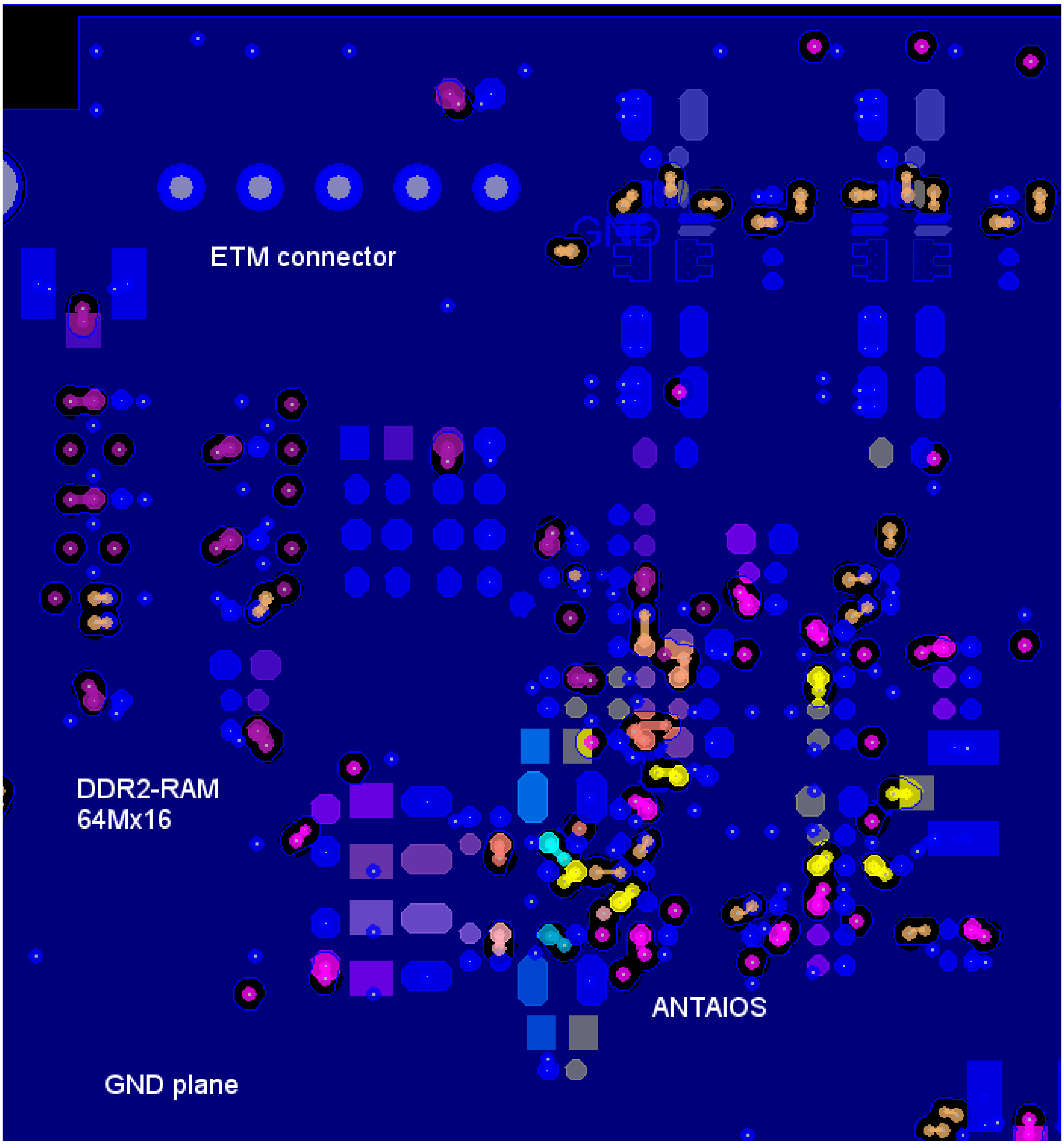
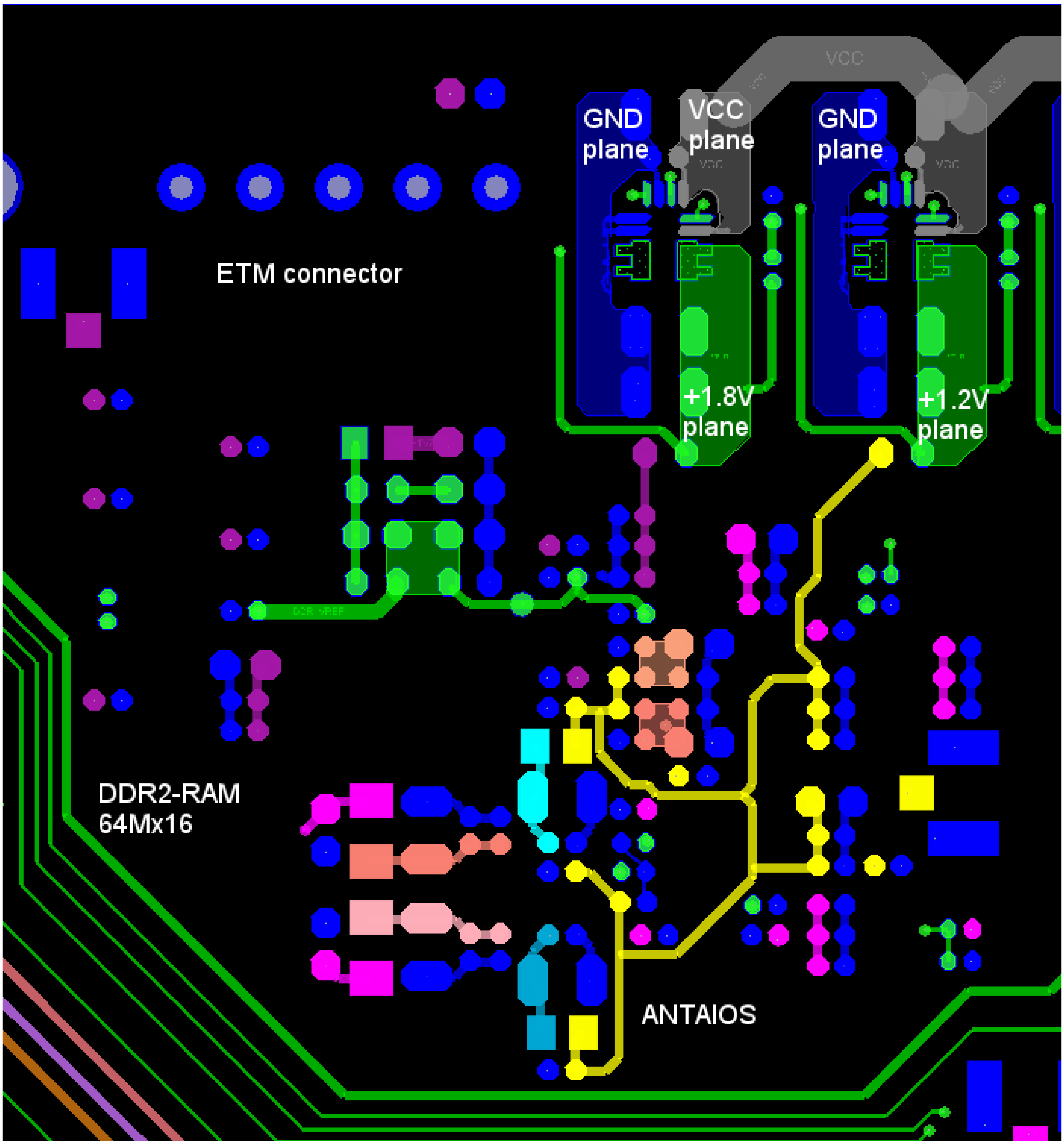


Figure 9-6 Bottom Layer



9.2.2 TFBGA-385

Following figures show the different layers of a PCB example, from top to bottom, with the ANTAIOS TFBGA-385. In this example the complete signals are routed. No microvias used in this layout. The only aim of this example layout is to show the complete routing of signals, therefore no additional components like capacitors or resistors are placed.
The diameter of the BGA ball pads is 400 µm. The diameter of the solder mask opening is 500 µm.

Figure 9-7 Top Layer

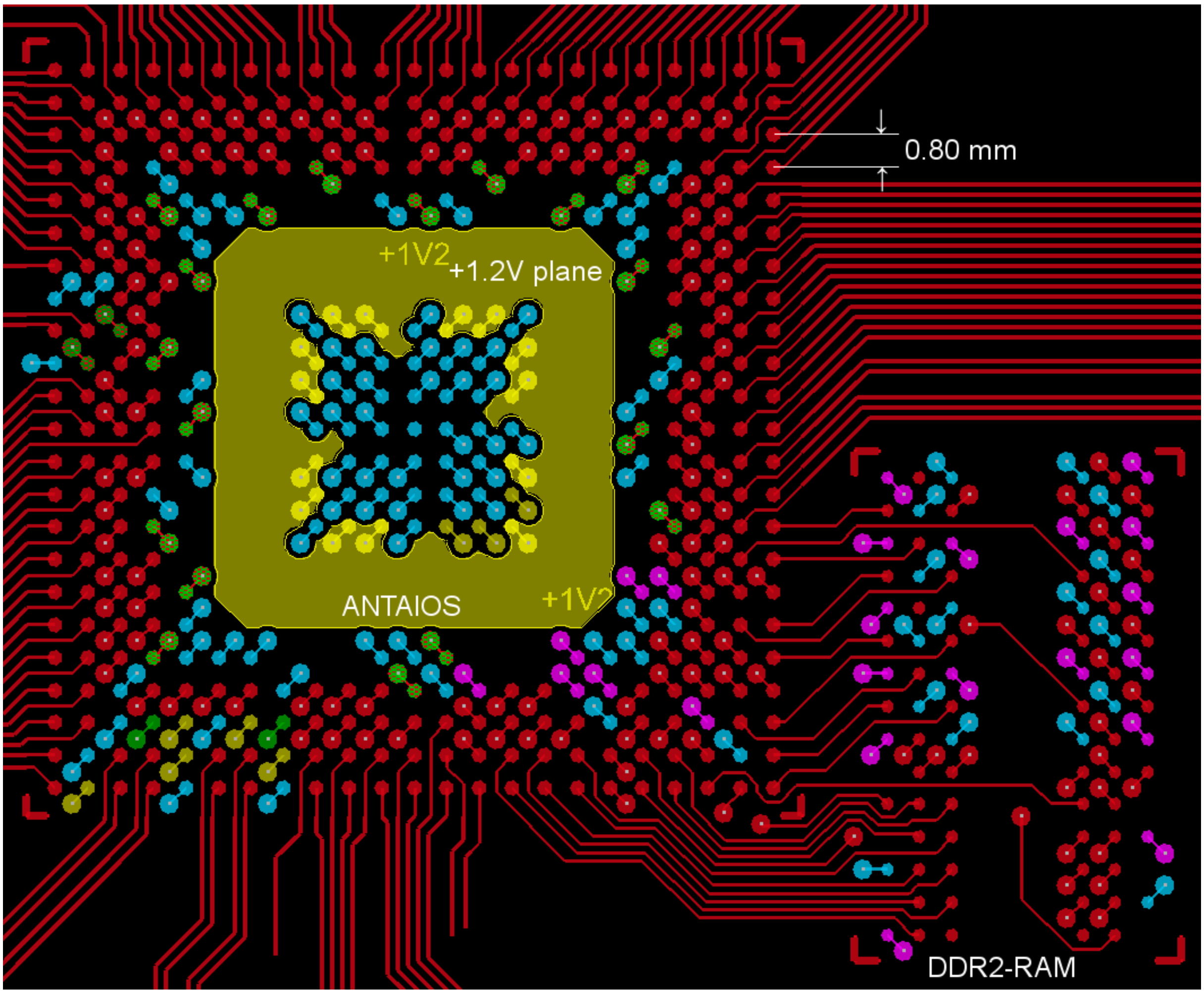


Figure 9-8 Internal Layer 1

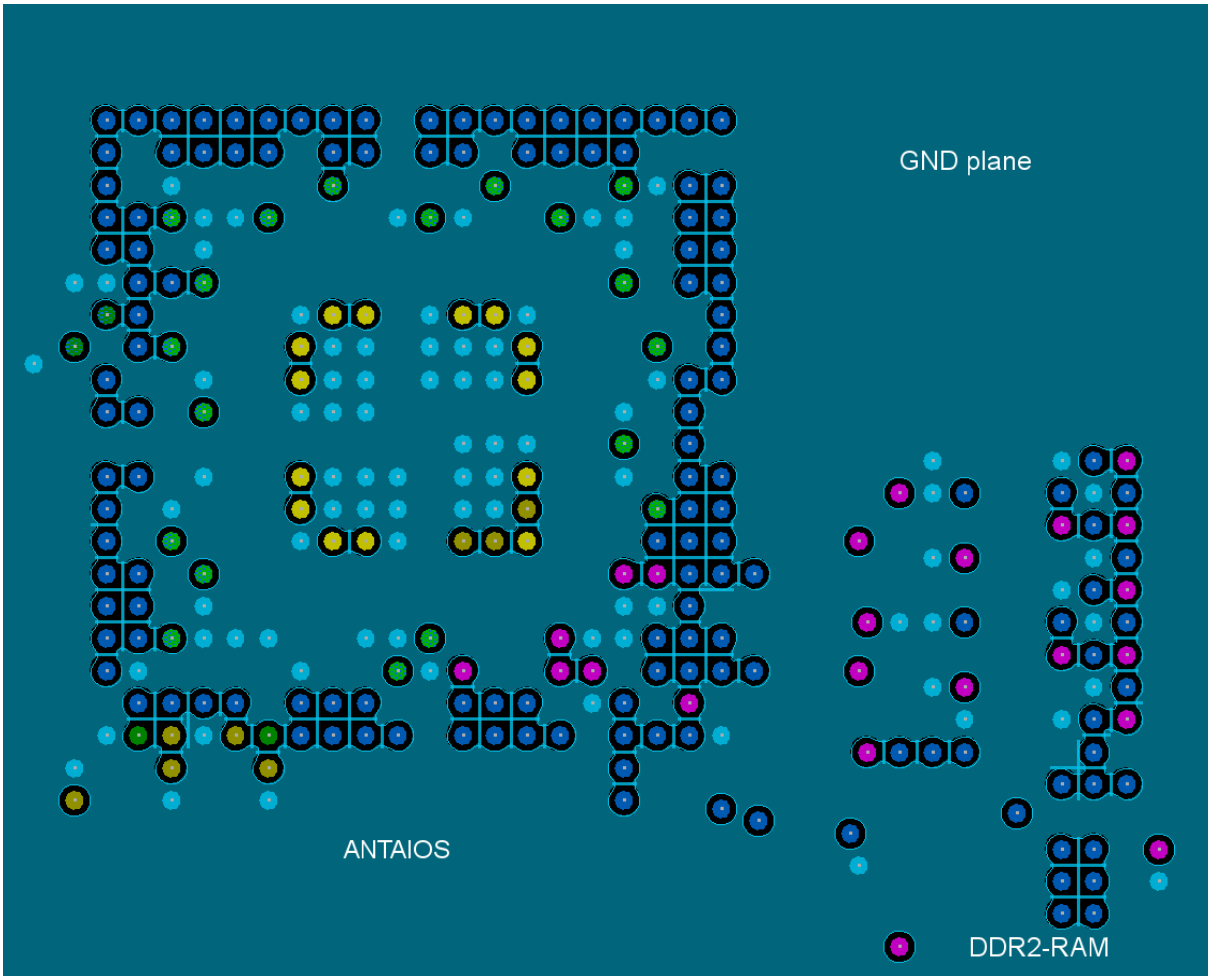


Figure 9-9 **Internal Layer 2**

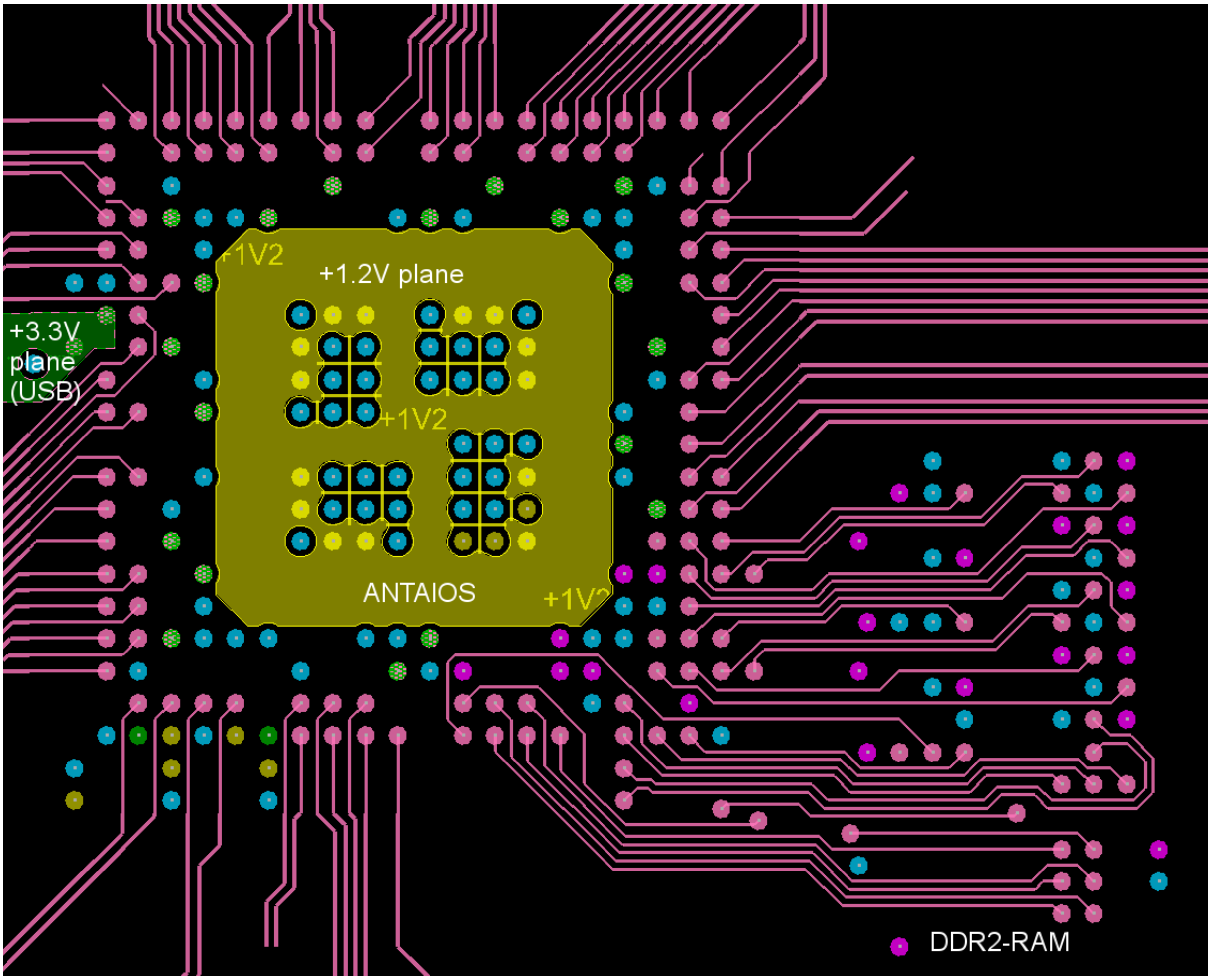


Figure 9-10 Internal Layer 3

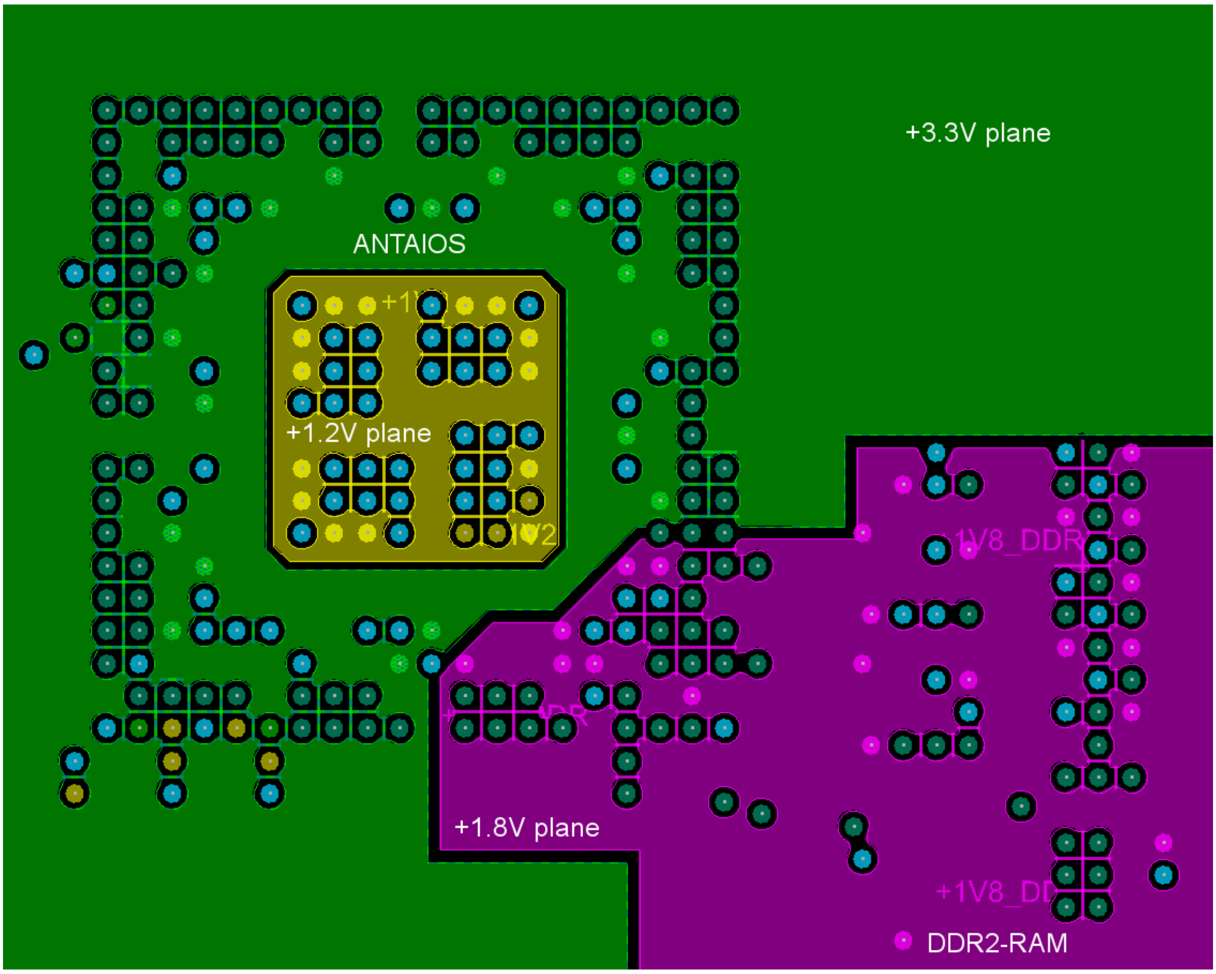


Figure 9-11 Internal Layer 4

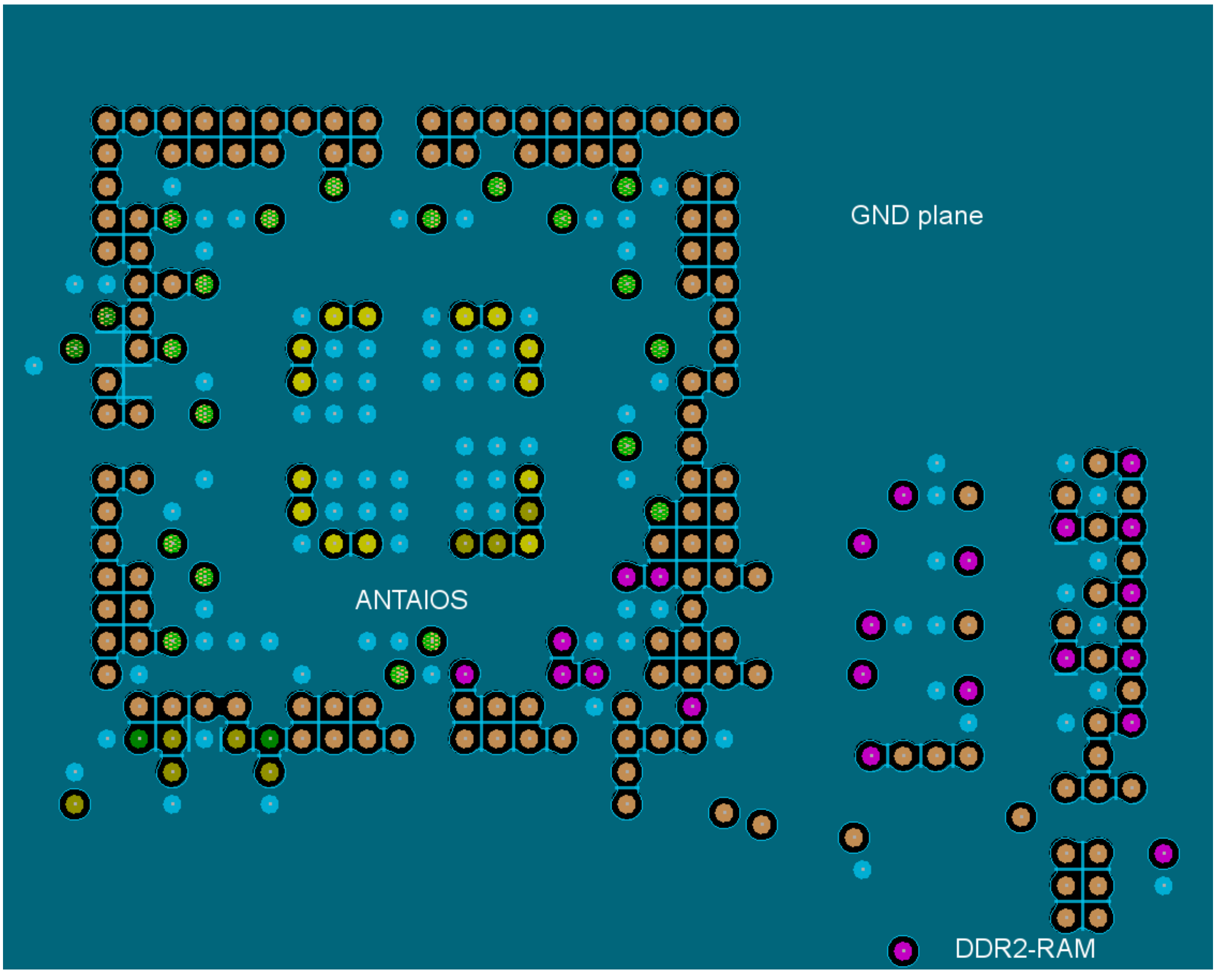
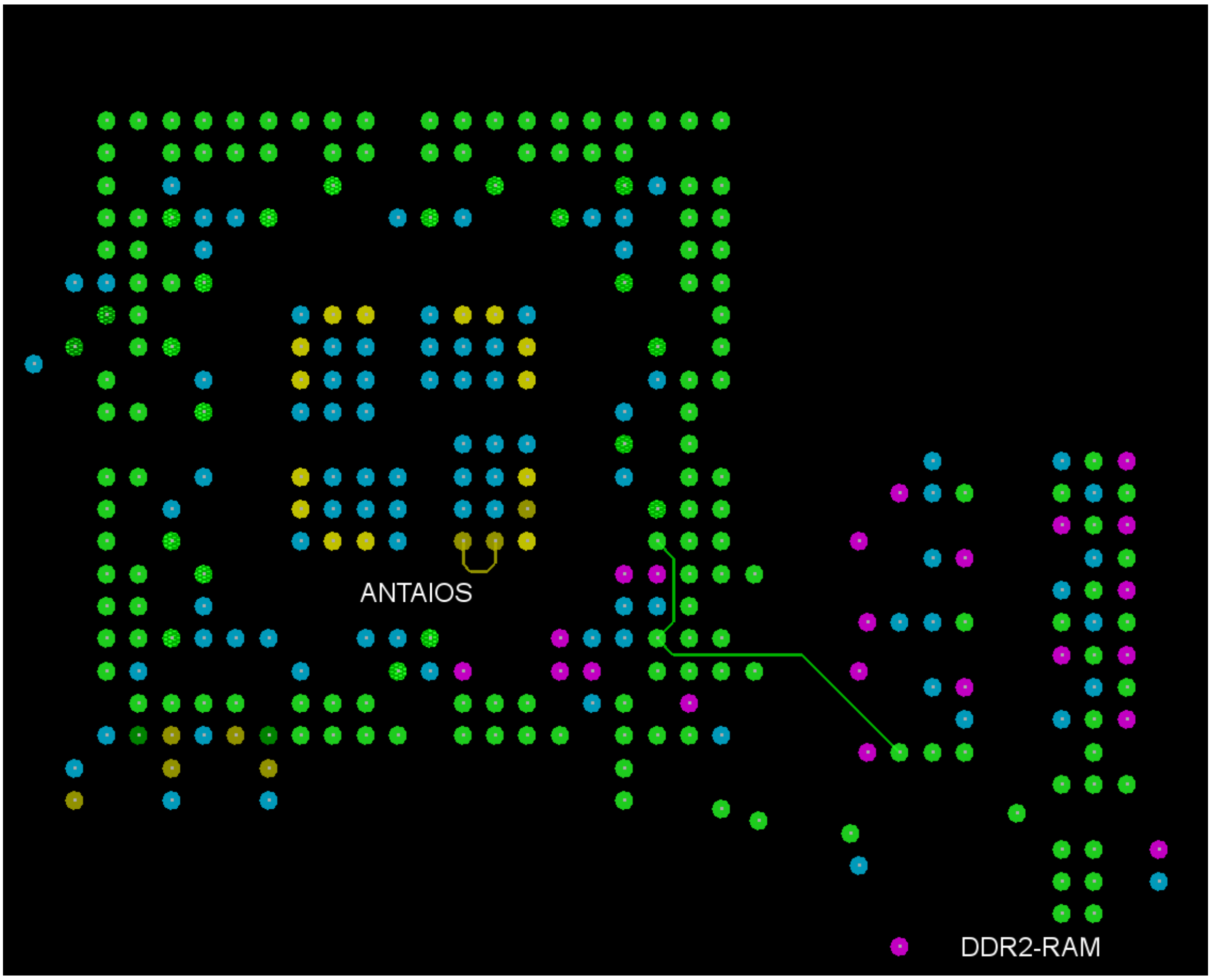


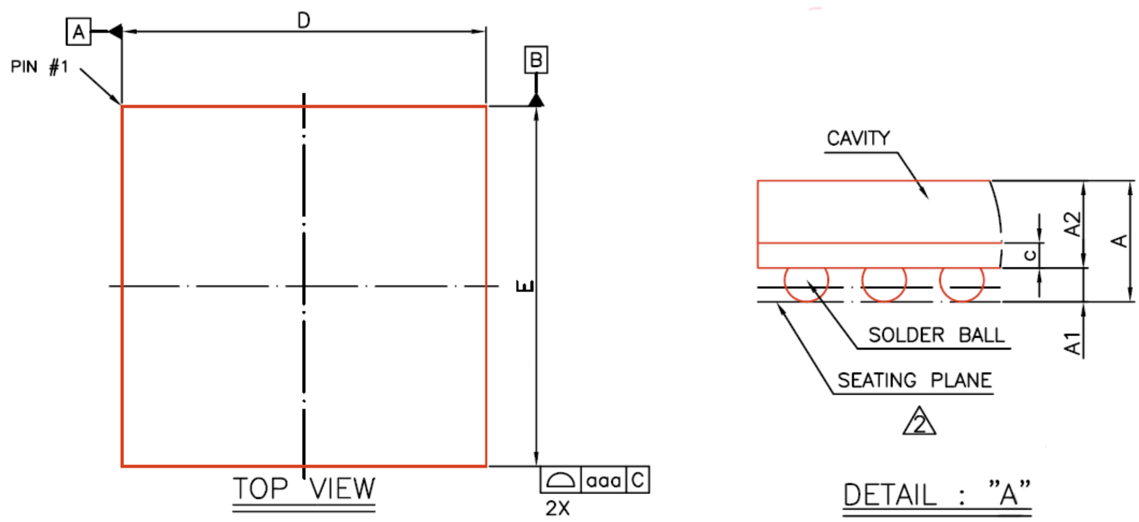
Figure 9-12 Bottom Layer

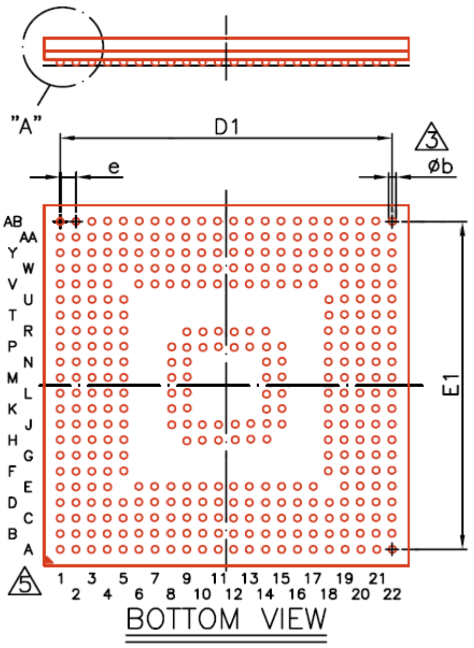


10 Package Specifications

10.1 TFBGA-380 (15x15 mm)

Figure 10-1 TFBGA-380 Package Dimensions and Tolerances





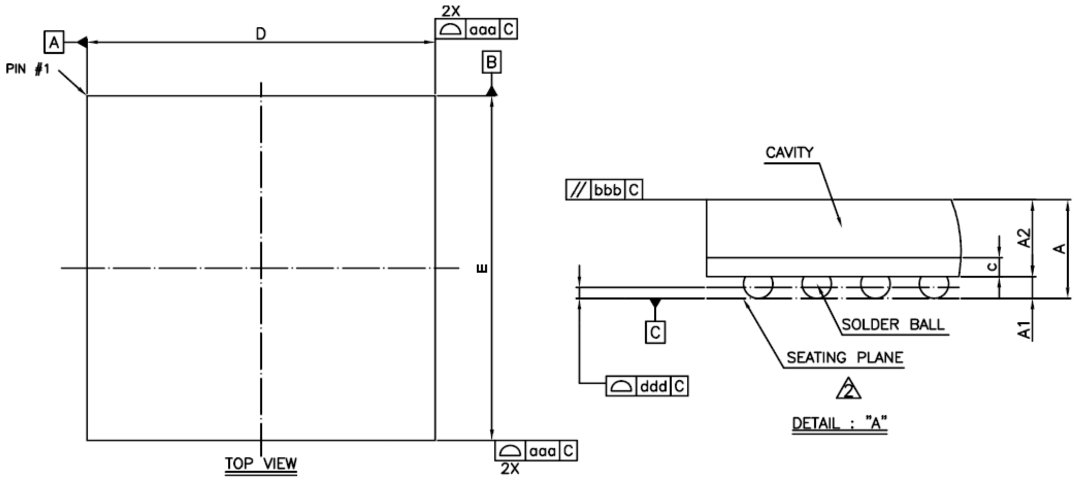
Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.03	1.10	1.17	0.041	0.043	0.046
A1	0.16	0.21	0.26	0.006	0.008	0.010
A2	0.84	0.89	0.94	0.033	0.035	0.037
c	0.32	0.36	0.40	0.013	0.014	0.016
D	14.90	15.00	15.10	0.587	0.591	0.594
E	14.90	15.00	15.10	0.587	0.591	0.594
D1	---	13.65	---	---	0.537	---
E1	---	13.65	---	---	0.537	---
e	---	0.65	---	---	0.026	---
b	0.25	0.30	0.35	0.010	0.012	0.014
MD/ME	22/22					

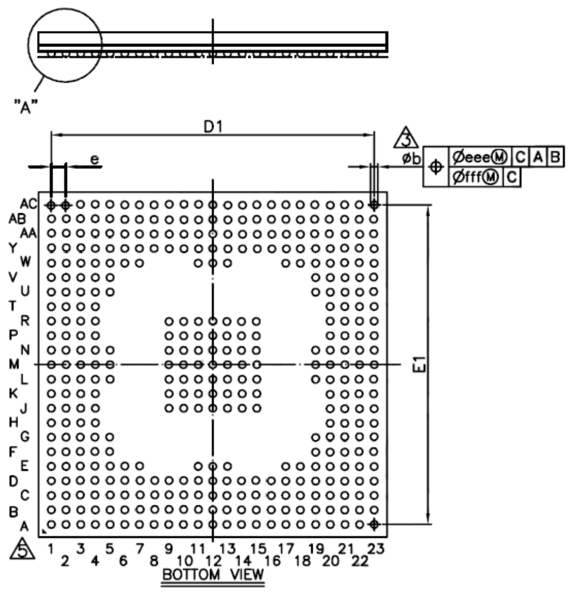
NOTE :

1. CONTROLLING DIMENSION : MILLIMETER.
2. PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
3. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM C.
4. SPECIAL CHARACTERISTICS C CLASS: bbb, ddd
5. THE PATTERN OF PIN 1 FIDUCIAL IS FOR REFERENCE ONLY .

10.2 TFBGA-385 (19x19 mm)

Figure 10-2 TFBGA-385 Package Dimensions and Tolerances





Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	---	---	1.45	---	---	0.057
A1	0.25	0.30	0.35	0.010	0.012	0.014
A2	1.01	1.06	1.11	0.040	0.042	0.044
c	0.32	0.36	0.40	0.013	0.014	0.016
D	18.90	19.00	19.10	0.744	0.748	0.752
E	18.90	19.00	19.10	0.744	0.748	0.752
D1	---	17.60	---	---	0.693	---
E1	---	17.60	---	---	0.693	---
e	---	0.80	---	---	0.031	---
b	0.35	0.40	0.45	0.014	0.016	0.018
aaa		0.10			0.004	
bbb		0.20			0.008	
ddd		0.15			0.006	
eee		0.15			0.006	
fff		0.08			0.003	
MD/ME			23/23			

NOTE :

1. CONTROLLING DIMENSION : MILLIMETER.
2. PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
3. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM C.
4. SPECIAL CHARACTERISTICS C CLASS: bbb, ddd(SPIL STANDARD)
5. THE PATTERN OF PIN 1 FIDUCIAL IS FOR REFERENCE ONLY .
6. REFERANCE DOCUMENT : JEDEC PUBLICATION 95 DESIGN GUIDE 4.5

10.3 Processing Instructions

Generally, ESD protective measures must be maintained for all electronic components. The ANTAIOS is a cracking-endangered component that must be handled properly.

Profichip products are tested and classified for moisture sensitivity according to the procedures outlined by JEDEC. The ANTAIOS is classified as moisture sensitivity level (MSL) 3.

In order to minimize any potential risk caused by moisture trapped inside non-hermetic packages it is a general recommendation to perform a drying process before soldering

11 Soldering Profile

Figure 11-1 Green Package Reflow Profile

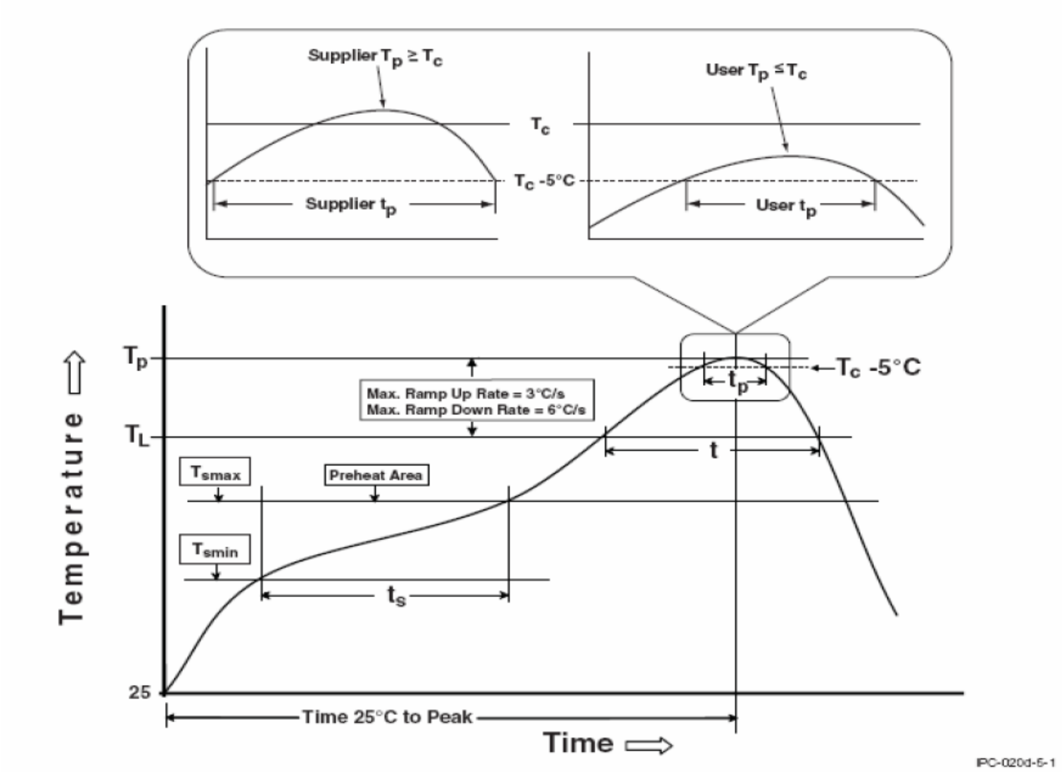


Table 11-1 Reflow Parameters (based on IPC/JEDEC J-STD-020D)

Profile Feature	Pb-Free Assembly (260 °C)
Preheat and Soak Temperature min (T_{smin}) Temperature max (T_{smax}) Time (T_{smin} to T_{smax}) (t_s)	150 °C 200 °C 60-120 seconds
Average ramp-up rate Time (T_{smax} to T_p)	3 °C/second max.
Liquid temperature (T_L) Time at liquid (t_L)	217 °C 60-150 seconds
Peak package body temperature (T_p)*	260 °C
Time (t_p) ** within 5 °C of the specified classification temperature (T_c)	30** seconds.
Average ramp-down rate (T_p to T_{smax})	6 °C/second max.
Time 25 °C to peak temperature	8 minutes max.

* Tolerance for peak profile temperature (T_p) is defined as a supplier minimum and a user maximum.

** Tolerance for time at peak profile temperature (t_p) is defined as a supplier minimum and a user maximum.

12 Reliability (FIT)

The Failures in Time (FIT) rate of a device is the number of failures that can be expected in one billion (10^9) hours of operation. Failure Rate for the complete chip is not available. However, as a recommended value, the known FIT value of the ANTAIOS process technology can be used.

Table 12-1: Failure Rate (FIT)

Parameter	Value	Unit
FIT (ANTAIOS process technology)	117.6	$1/(10^9$ h)

13 Ordering Information

Table 13-1 Order Codes

Version	Order Code	Package	Temperature Range	Notes
ANTAIOS-BGA380	ANT1000	TFBGA-380	Industrial	
ANTAIOS-BGA385	ANT1001	TFBGA-385	Industrial	

14 Trademarks

ARM and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.
CoreSight, ETM, Embedded Trace Macrocell and Embedded Trace Buffer are trademarks of ARM Limited.

EtherCAT is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

EtherNet/IP and CIP Sync are trademarks of ODVA, Inc.

Ethernet Powerlink is a trademark and brand of Bernecker + Rainer Industrie-Elektronik Ges.m.b.H.

MECHATROLINK-III is a registered trademark of the MECHATROLINK Members Association (MMA).

Modbus is a registered trademark of Schneider Electric and is licensed by the Modbus Organization, Inc.

PROFIBUS and PROFINET are registered trademarks of PROFIBUS & PROFINET International (PI).

All trademarks are the property of their respective owners.

15 Revision History

Table 15-1 Revision history

Version	Date	Remarks
V0.10	07.12.2012	First release
V0.11	09.12.2012	Overview added
V0.12	06.02.2014	Complete review and update
V0.13	25.09.2014	Added 64kByte Caches and new size of FIFO and CI
V0.14	29.09.2014	Update to Overview Image
V0.15	30.09.2014	Added Package Drawing, corrected TechIOs
V0.16	09.10.2014	Corrected Number of UARTs to 2, added comments to CI, FIFO and USB
V0.17	01.04.2015	FA616 replaced by ARM Cortex-A5
V0.18	24.09.2015	Added debug options to CA5, Changed frequencies of SPI, Added second Package Option
V0.19	25.01.2016	Changed small Package
V0.20	15.06.2016	Updated package 2
V0.21	16.06.2016	Minor corrections, block diagram updated, order information added
V0.22	22.08.2016	Removed 10BaseT Option
V0.23	30.09.2016	Changed notation of bit to low letters Added QuadSPI information Added Pin Description Added Ordering Information Added Package Specification and Process Instructions Updated Operational Specifications
V0.24	26.10.2016	Updated Operational Specifications (LVTTL, SSTL18, USB)
V0.25	27.10.2016	Updated Power Consumption Change boundary of recommended operating conditions to ±5 % Add power consumption for PROFINET device application
V0.26	24.11.2016	Modified application chapter
V0.27	25.11.2016	Insert color scheme for pinout Move information about power supply into subchapter of power dissipation
V0.28	1.12.2016	Added 100ppm for clock, added short PROFINET description Added Psi, added measured delta T for PCB and Case
V0.29	01.12.2016	Fix current consumption Update reset state of some signals
V0.30	07.12.2016	Add remarks to thermal characteristics Add reset information
V0.31	12.01.2017	Append _N to chip select names of AEI Bootloader description updated Add explanation to DBG_CLK/STB Add timing of AEI interface
V0.32	17.01.2017	Add delay for AEI slave interface
V.0.33	25.01.2017	Pin assignment: more detailed description for tristate value Added schematics to 4.7 Ethernet PHYs Correct description of DDR interface Correct value for signal detect of PHY in fiber mode Add PCB layout Edit system clock specification Add trademark remarks Add remarks to qualified chips
V1.00	24.02.2017	Add application details about debug interfaces and USB endpoint connection Fix list of figures Modified 3.7 (AEI) and 4.10 (AEI)

V1.01	27.02.2017	Add description for Port B Version 3 Modified t_{ACCT} of AEI Slave
V1.02	2.03.2017	Modified package specification TFBGA-385 Add timing of NAND flash
V1.03	17.03.2017	Add Power Consumption or EtherCAT Slave Add PCB Layout Guidelines (DDR, PHYs, USB); remove USB under Application Details) Add timing of QSPI, SD/MMC, SPI, I²C
V1.04	19.05.2017	Changed document title
V1.05	28.06.2017	Correct description of port E Correct bandwidth value of DDR2-RAM Remove modules with incomplete timing description
V1.06	14.07.2017	Reinsert timing of SD/MMC, SPI, I²C
V1.07	24.07.2017	Add hints for magnetics and oscillator
V1.08	08.08.2017	move QuadSPI pin description from 5.2.9.6 Serial Interfaces to 5.2.9.1 Flash Memory Interfaces
V 1.09	04.09.2017	Change description of optical interface of internal Ethernet PHYs
V 1.10	13.10.2017	add PCB layout information about pads
V 1.11	19.10.2017	add FIT rate
V 1.12	09.11.2017	add recommendation for SliceBus line driver
V 1.12	30.11.2017	update list of recommended QuadSPI NOR Flash devices
V 1.14	10.01.2018	extend AEI description
V 1.15	07.02.2018	Add typical power consumption for profinet application (chapter 7.4.1 at Table 7-7 and Table 7-8)
V 1.16	16.02.2018	Add QSPI-Flash: Macronix MX25L6433F
V1.17	12.04.2018	Added note for USB controller if not used
V1.18	02.05.2018	update timing of AEI Master read access
V1.19	17.09.2018	correct connection of AEI Slave address bus
V1.20	19.09.2018	Add QSPI-Flashes: Integrated Silicon Solution IS25LP064A and IS25LP128-JBLE
V1.21	03.06.2019	Soldering profile updated
V1.22	01.10.2019	New document design
V1.23	04.02.2020	New DDR2: Winbond W9751G6NB-25I

YASKAWA Europe GmbH
Ohmstr. 4, 91074 Herzogenaurach
Germany
Phone: +49 (0)9132 744-200
E-Mail: support.profichip@yaskawa.eu.com
www.profichip.com

YASKAWA