



ERTEC 200P-2

Enhanced Real-Time Ethernet Controller

Manual

DISCLAIMER

WE HAVE CHECKED THE CONTENTS OF THIS MANUAL FOR AGREEMENT WITH THE HARDWARE AND SOFTWARE DESCRIBED. SINCE DEVIATIONS CANNOT BE PRECLUDED ENTIRELY, WE CANNOT GUARANTEE FULL AGREEMENT. HOWEVER, THE DATA IN THIS MANUAL ARE REVIEWED REGULARLY. NECESSARY CORRECTIONS ARE INCLUDED IN SUBSEQUENT EDITIONS. SUGGESTIONS FOR IMPROVEMENT ARE WELCOMED.

THIS MANUAL CONTAINS HYPERLINKS TO THE WEB SITES AS WELL AS TO ELECTRONIC DOCUMENTS OF THIRD PARTY COMPANIES. SIEMENS CANNOT BE HELD RESPONSIBLE FOR THE CONTENTS OF THESE WEB SITES AND ELECTRONIC DOCUMENTS, NOR DOES SIEMENS ADOPT THESE WEB SITES AND ELECTRONIC DOCUMENTS AND THEIR CONTENTS AS THEIR OWN. YOU THEREFORE USE THESE LINKS AT YOUR OWN RISK. SINCE SIEMENS IS NOT RESPONSIBLE FOR LINKED CONTENTS AND INFORMATION ON THE WEB SITES AND ELECTRONIC DOCUMENTS OF THIRD PARTIES, THIS INFORMATION IS NOT CHECKED BY SIEMENS.

COPYRIGHT

© SIEMENS AG 2016. ALL RIGHTS RESERVED

THE REPRODUCTION, TRANSMISSION OR USE OF THIS DOCUMENT OR ITS CONTENTS IS NOT PERMITTED WITHOUT EXPRESS WRITTEN AUTHORITY. OFFENDERS WILL BE LIABLE FOR DAMAGES. ALL RIGHTS, INCLUDING RIGHTS CREATED BY PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN, ARE RESERVED.

ALL PRODUCT AND SYSTEM NAMES ARE REGISTERED TRADEMARKS OF THEIR RESPECTIVE OWNER AND MUST BE TREATED AS SUCH.

TECHNICAL DATA SUBJECT TO CHANGE.

Preface

Target Audience of this Manual

This manual is intended for hardware developers who want to use the ERTEC 200P for new products. Experience working with processors and designing embedded systems and knowledge of Ethernet are required for this. It describes all ERTEC function groups in detail and provides information that you must take into account when configuring your own PROFINET IO device hardware.

The manual serves as a reference for software developers. The address areas and register contents are described in detail for all function groups.

Structure of this Manual

- o Section 1 Introduction
- o Section 2 Description of functions
- o Section 3 IO Interface
- o Section 4 Layout and Design Hints
- o Section 5 Package
- o Section 6 Quality
- o Section 7 Miscellaneous

Scope of the Manual

This manual applies to the following product:

ERTEC 200P-2

This manual will be updated as required. You can find the current version of the manual on the Internet at <http://www.siemens.com/comdec>.

Guide

To help you quickly find the information you need, this manual contains the following aids:

- o A complete table of contents as well as a list of all figures and tables in the manual are provided at the beginning of the manual.
- o A glossary containing definitions of important terms used in the manual is located following the appendices.
- o References to other documents are indicated by the document reference number enclosed in slashes (/No./). The complete title of the document can be obtained from the list of references at the end of the manual.

Additional Support

If you have questions regarding use of the described block that are not addressed in the documentation, please contact your Siemens representative.

Please send your written questions, comments, and suggestions regarding the manual to the hotline via the e-mail address indicated above.

In addition, you can receive general information, current product information, FAQs, and downloads pertaining to your application on the Internet at:

<http://www.siemens.com/comdec>

Technical Contacts for Germany / Worldwide

Siemens AG
ComDeC

Street address:	Phone: 0911/750-2080
Breslauer Str. 5	Fax: 0911/750-2100
90766 Fürth	E-mail: ComDeC@siemens.com

Technical Contacts for USA

<u>PROFI Interface Center:</u>	Fax: +1(423)- 262- 2103
One Internet Plaza	Phone: +1(423)- 262- 2576
PO Box 4991	E-mail: pic.industry@siemens.com
Johnson City, TN 37602-4991	

Technical Contacts for China

PIC China: Mr. Xu Bin
7, Wangjing Zhonghuan Nanku E-mail: Profinet.cn@siemens.com
100102 BEIJING

Security Information:

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.

Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.

Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit (<http://www.siemens.com/industrialsecurity>).

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under (<http://www.siemens.com/industrialsecurity>).

Table of contents

1	INTRODUCTION	15
1.1	Overview of the ERTEC 200P	15
1.1.1	Mechanisms for PROFINET IO (NRT communication)	15
1.1.2	Mechanisms for PROFINET IO (RT communication) RTC1/2	15
1.1.3	Mechanisms for PROFINET IO (IRT communication) RTC3	15
1.1.4	Mechanisms for PROFINET IO (IRT High Performance communication) RTC3	16
1.2	ERTEC 200P data	17
1.2.1	System Reliability	18
1.2.2	Electromagnetic Compatibility (EMC)	18
1.3	ERTEC 200P Use Cases	19
1.3.1	Use Case 1 (UC1): Operation with external host	19
1.3.2	Use Case 2 (UC2): Operation without external Host	20
1.4	Application notes	21
1.4.1	EMC SDRAM Interface	21
1.4.2	EMC Burst Flash Interface	22
1.4.3	No free-running frequency at quartz break	22
1.4.4	16-bit Data Access over XHIF in Buffered Mode	23
2	DESCRIPTION OF FUNCTIONS	24
2.1	Block diagram	24
2.2	General Function Description (Motivation)	24
2.3	Description of Individual ERTEC 200P Blocks	25
2.3.1	Processor Subsystem (ARM926)	25
2.3.1.1	Features	26
2.3.1.2	Block diagram	26
2.3.1.3	ARM926EJ-S processor	27
2.3.1.3.1	Cache structure of ARM926EJ-S	27
2.3.1.3.2	ARM926 Tightly Coupled Memories (ARM926_TCM)	28
2.3.1.3.3	Memory Management Unit (MMU)	29
2.3.1.3.4	Bus Interface of the ARM926 Processor	30
2.3.1.3.5	ARM926 Embedded Trace Macrocell (ETM9), Trace Buffer (ETB11)	30
2.3.1.4	Debug Support	31
2.3.1.4.1	Debug Configuration	31
2.3.1.4.2	ARM926EJ-S Debug Interface	31
2.3.1.5	Boot Process for the ERTEC 200P	32
2.3.1.5.1	Primary Boot Loader	34
2.3.1.5.2	Boot mode 1 ... 2 (booting with NOR flash) in compile mode	37
2.3.1.5.3	Boot Modes 5 and 6 (Bootting with SPI Master)	38
2.3.1.5.4	Boot Mode 7 (Bootting with External Host)	40
2.3.1.6	Secondary Boot Loader	43
2.3.1.6.1	Memory Swapping by the Secondary Boot Loader	43
2.3.2	AMBA (Internal Bus System)	43
2.3.2.1	Characteristics of the Bus System in ERTEC 200P	43
2.3.2.2	Endianness	44
2.3.2.3	AHB Subsystem (Multi-Layer AHB)	44
2.3.2.3.1	AHB Arbiters	45
2.3.2.3.2	AHB Multi-layer Configuration	45
2.3.2.3.3	AHB Burst Breaker	46
2.3.2.4	APB Subsystem (Peripheral Bus)	47
2.3.2.4.1	APB Decoder	51
2.3.2.5	Address Range and Acknowledgement Delay Monitoring	51
2.3.2.5.1	Monitoring at the AHB Side	51

2.3.2.5.2	Monitoring at the APB Side	52
2.3.2.5.3	Monitoring at the EMC.....	52
2.3.2.5.4	Monitoring in the Individual Modules.....	52
2.3.2.6	Overview	54
2.3.2.7	Interface	54
2.3.2.8	Interrupt acquisition.....	55
2.3.2.9	Interrupt masking / prioritisation	57
2.3.2.10	Interrupt post-processing.....	58
2.3.2.11	Special functions.....	59
2.3.2.12	Debug functions	59
2.3.2.13	Miscellaneous	59
2.3.2.13.1	Synchronizing the inputs	59
2.3.2.13.2	Bus interface.....	60
2.3.2.13.3	Sequences.....	60
2.3.2.13.4	Operating rules	61
2.3.2.13.5	Startup/shutdown	62
2.3.2.14	Interrupt sources for ARM-IRQ	63
2.3.2.14.1	Interrupts for accesses to missing addresses	68
2.3.2.14.2	Confirmation delay in the Memory Controller (EMC) address area	68
2.3.2.15	Interrupt sources for ARM-FIQ	69
2.3.2.15.1	High-priority Interrupts for Debugging	69
2.3.2.16	Address Mapping	70
2.3.2.17	Register description	76
2.3.3	Host ICU (PerIF: Event Unit).....	103
2.3.4	GDMA (Direct Memory Access)	103
2.3.4.1	GDMA Function Description.....	103
2.3.4.1.1	AHB interfaces	104
2.3.4.1.2	Job priorities	104
2.3.4.1.3	Details.....	104
2.3.4.1.4	Usage	114
2.3.4.1.5	Result	115
2.3.4.1.6	Memory.....	116
2.3.4.1.7	Interrupts	117
2.3.4.2	ERTEC 200P GDMA Use Cases.....	117
2.3.4.2.1	Using the Job Start and HW_REQ Signal List.....	117
2.3.4.2.2	DMA with UART Interface	125
2.3.4.3	GDMA-IP Bugs	125
2.3.4.4	Address Mapping	126
2.3.4.5	Register Description.....	129
2.3.5	EMC (External Memory Controller).....	157
2.3.5.1	Block Diagram of the EMC	158
2.3.5.2	EMC Reset and Clock.....	159
2.3.5.3	AHB-Slave Interface.....	161
2.3.5.4	EMC Notes	162
2.3.5.4.1	Remapping	162
2.3.5.4.2	Maximum number of wait cycles.....	162
2.3.5.4.3	Shift Mode with the Asynchronous EMC Interface	162
2.3.5.4.4	Control of an External Driver	163
2.3.5.4.5	QVZ Acknowledgement Delay.....	163
2.3.5.5	Timing of the Asynchronous Memory Interfaces	164
2.3.5.5.1	Read Access.....	164
2.3.5.5.2	Write Access.....	165
2.3.5.6	Application examples	166
2.3.5.7	Interface signals.....	167
2.3.5.8	Address Mapping	169
2.3.5.9	Register description	170
2.3.6	Host Interface – Parallel (XHIF).....	182
2.3.6.1	Block Diagram of the Host Interface	182
2.3.6.2	Function Description (XHIF)	183
2.3.6.2.1	AHB Master Interface.....	186
2.3.6.2.2	APB Slave Interface.....	186
2.3.6.2.3	XHIF Configuration.....	186
2.3.6.2.4	Application Information.....	187

2.3.6.3	Address Mapping	189
2.3.6.4	Register Description.....	190
2.3.7	Peripheral Interface	197
2.3.7.1	APB Interface.....	199
2.3.7.1.1	Address Mapping	199
2.3.7.1.2	Register Description.....	203
2.3.8	PROFINET-IP	230
2.3.8.1	PN-IP Interfaces	230
2.3.8.1.1	AHB Interface.....	230
2.3.8.1.2	Interrupt Management.....	230
2.3.8.1.3	PN-PLL (incl. 3x Application Timer Blocks).....	232
2.3.8.1.4	EDC in PN-IP RAMs	234
2.3.8.1.5	Recommended Parameter Assignment for ERTEC 200P	234
2.3.8.2	Ethernet PHY (integrated)	235
2.3.8.2.1	Status of the integrated PHY	236
2.3.9	CRU (Clock and Reset Unit)	236
2.3.9.1	Clock System.....	236
2.3.9.2	Clock Generation and Distribution	236
2.3.9.2.1	PLL Clock Generation	237
2.3.9.2.2	Clock Source for the PHYs and Ethernet MACs.....	238
2.3.9.2.3	Clock Source for JTAG.....	238
2.3.9.3	Clock Monitoring	238
2.3.9.3.1	Lock Timer 1	238
2.3.9.3.2	Lock Timer 2	238
2.3.9.3.3	Lock Monitor	239
2.3.9.4	Reset System	241
2.3.9.4.1	Asynchronous PowerOn Reset.....	243
2.3.9.4.2	Asynchronous Hardware Reset	244
2.3.9.4.3	Asynchronous JTAG Reset	244
2.3.9.4.4	Asynchronous ARM926 Watchdog Reset	245
2.3.9.4.5	Asynchronous Software Reset for ERTEC 200P (Without PN-IP)	245
2.3.9.4.6	Asynchronous Software Reset for PN-IP	245
2.3.9.4.7	Asynchronous Software Reset for the ARM926EJ-S Core	246
2.3.9.4.8	Synchronous Software Reset (PN-IP, PER-IF, Host Interface).....	246
2.3.10	APB Peripherals	247
2.3.10.1	I-Filter	248
2.3.10.1.1	Operating Principle of the RC Filter	250
2.3.10.1.2	Address Mapping	251
2.3.10.1.3	Register Description.....	253
2.3.10.2	ARM926 Watchdog	260
2.3.10.2.1	Overview.....	260
2.3.10.2.2	Block diagram	260
2.3.10.2.3	Signal waveforms.....	263
2.3.10.2.4	Write protection of the watchdog register.....	264
2.3.10.2.5	Starting the Watchdog.....	264
2.3.10.2.6	Notes	265
2.3.10.2.7	Address Mapping	265
2.3.10.2.8	Register Description.....	266
2.3.10.3	*Timer 0 – 5	269
2.3.10.3.1	Overview.....	269
2.3.10.3.2	TIMER_TOP functionality.....	270
2.3.10.3.3	Timer module	273
2.3.10.3.4	Overview of the count modes	280
2.3.10.3.5	Timing requirements	281
2.3.10.3.6	Operating rules	282
2.3.10.3.7	Connections on ERTEC 200P Toplevel	283
2.3.10.3.8	Address Mapping	285
2.3.10.3.9	Register Description.....	286
2.3.10.4	F-Counter	302
2.3.10.4.1	Function Description	303
2.3.10.4.2	Application Information.....	303
2.3.10.4.3	Address Mapping	303
2.3.10.4.4	Register Description.....	304

2.3.10.5	UART1...4	305
2.3.10.5.1	UART Baud Rates.....	306
2.3.10.5.2	Address Mapping	308
2.3.10.5.3	Register Description.....	310
2.3.10.6	I ² C	325
2.3.10.6.1	Baudrate generator	325
2.3.10.6.2	IO Expansion Unit	325
2.3.10.6.3	Important Software rules	326
2.3.10.6.4	Address Mapping	328
2.3.10.6.5	Register Description.....	329
2.3.10.7	SPI1/2 (Serial Peripheral Interface)	335
2.3.10.7.1	Features of the SPI Interface.....	337
2.3.10.7.2	SSPMS IP Extension.....	338
2.3.10.7.3	SSPMS IP Integration	339
2.3.10.7.4	Send Support with Pause between Individual Characters	342
2.3.10.7.5	SPI Flash Boot	343
2.3.10.7.6	Address Mapping	345
2.3.10.7.7	Register Description.....	345
2.3.10.8	GPIO (General Purpose Input / Output).....	348
2.3.10.8.1	GPIO Module Function Description	349
2.3.10.8.2	Integration of the GPIO Module	352
2.3.10.8.3	GPIO Pad Multiplexing	353
2.3.10.8.4	GPIO Assignment	353
2.3.10.8.5	Address Mapping	356
2.3.10.8.6	Register Description.....	357
2.3.10.9	SCRB (System Control Register Block)	365
2.3.10.9.1	Hardware Identifier Register	365
2.3.10.9.2	Boot Register	366
2.3.10.9.3	Config Register	366
2.3.10.9.4	Reset Registers	367
2.3.10.9.5	ACCESS_ERROR and QVZ Register.....	368
2.3.10.9.6	PLL Status Register	368
2.3.10.9.7	Memory Swapping	368
2.3.10.9.8	ARM Control Register	369
2.3.10.9.9	PHY Register	369
2.3.10.9.10	AHB Burst Breaker Register	369
2.3.10.9.11	GPIO Control Register	369
2.3.10.9.12	I ² C Clock Divider	370
2.3.10.9.13	EDC Register	370
2.3.10.9.14	ARM926 Mapping	370
2.3.10.9.15	PAD Control Register.....	370
2.3.10.9.16	Clear DMA Request	373
2.3.10.9.17	SPI Parity Error	373
2.3.10.9.18	XHIF Mode.....	373
2.3.10.9.19	Ext. Driver Enable	373
2.3.10.9.20	SPI Mode	373
2.3.10.9.21	SD Signal Handling	373
2.3.10.9.22	Address Mapping	374
2.3.10.9.23	Register Description.....	376
2.4	Memory Mapping.....	415
2.4.1	Memory Mapping ARM926-I	415
2.4.2	ARM926-D Memory Mapping.....	416
2.4.3	PN-IP Memory Mapping.....	417
2.4.4	Host IF Memory Mapping.....	418
2.4.5	GDMA Memory Mapping.....	421
2.5	Detailed Address Mapping.....	422
2.6	Register Description	425
3	IO INTERFACE.....	426

3.1	Overview	426
3.2	Detailed Signal Description	427
3.2.1	Strapping Pins	450
3.3	IO Timing	451
3.3.1	EMC Timing	452
3.3.1.1	SRAM Timing	452
3.3.1.1.1	SRAM Timing for read access	452
3.3.1.1.2	SRAM Timing for write access	453
3.3.1.2	BurstMode Flash Timing	454
3.3.1.3	SDRAM Timing	455
3.3.1.3.1	SDRAM Timing for read access	457
3.3.1.3.2	SDRAM Timing for write access	458
3.3.2	Host-Interface Timing	459
3.3.2.1	XHIF Timing	459
3.3.2.1.1	Separate RD/WR	459
3.3.2.1.2	Common RD/WR	461
3.3.2.2	SPI Timing	463
3.3.3	PN-IP Timing	464
3.3.3.1	MII Timing	464
3.3.3.1.1	MII Timing at Integrated PHY	464
3.3.3.2	MDIO Timing	464
3.3.3.2.1	MDIO Timing at the Integrated PHY	465
3.3.3.3	Integrated PHY Timing	465
3.3.3.3.1	PHY FX Timing	465
3.3.3.3.2	PHY LED Timing	465
3.3.3.4	PNPLL Timing	465
3.3.3.4.1	Timing for time synchronization	467
3.3.4	SPI Timing	468
3.3.4.1	SPI1	468
3.3.4.2	SPI2	468
3.3.5	UART Timing	470
3.3.5.1	UART1	470
3.3.5.2	UART2	470
3.3.5.3	UART3	470
3.3.5.4	UART4	470
3.3.6	I ² C Timing	471
3.3.6.1	I ² C – APB	471
3.3.6.2	I ² C – PN-IP	471
3.3.7	GPIO Timing	471
3.3.8	JTAG Timing	471
3.3.9	ARM926 Trace Timing	472
4	LAYOUT AND DESIGN HINTS	474
4.1	EMC Measures	474
4.1.1	ESD Protection	474
4.1.2	Immunity to ESD	474
4.1.3	Package Power Distribution	474
4.1.4	Spike Filter	474
4.2	Oscillator Circuit	476
4.2.1	Circuit with Ext. Crystal	476
4.2.2	Circuit with External Crystal Oscillator	478
4.3	PLL Wiring (Power Supply)	479
4.4	Test Signal Configuration	480
4.5	JTAG Wiring	480

4.6	PHY Wiring	481
4.6.1	PHY-TX Wiring	481
4.6.1.1	PHY-TX Wiring – not used	482
4.6.2	PHY-FX Wiring	483
4.6.2.1	PHY-FX Wiring – not used	485
4.6.2.2	PHY-SD Wiring – Avago QFBR-5978AZ	485
4.6.2.2.1	PxSD circuit	485
4.6.2.2.2	GPIO circuit	485
4.7	Wiring of pins not used.....	486
4.8	Operating Conditions.....	488
4.8.1	Power Up.....	488
4.8.2	Wiring of CTRL-STBY	488
4.8.3	Power-Up Sequence (PLL)	489
4.8.4	PLL Behavior	489
4.8.4.1	following crystal break.....	489
4.8.4.2	upon temporary clock failure	489
4.8.5	Readiness of internal resources once a reset is cleared	490
4.9	Pull-up/Pull-down Resistor Values.....	490
4.10	Schmitt Trigger Characteristics.....	490
4.11	Module and ASIC Code (Chip ID).....	490
4.12	Power dissipation.....	491
5	PACKAGE	493
5.1	Package Drawing.....	493
5.2	Ball Layout.....	493
5.3	Marking (Printed).....	495
5.3.1	Order codes.....	495
5.4	SiP – FPBGA400 Thermal Characteristics	495
5.4.1	Max. junction temperature T_J	496
5.5	Solder Profile.....	497
5.6	Packing Information	498
5.6.1	Tray.....	498
5.6.2	Tape&Reel	498
6	QUALITY	499
6.1	Hard-Error FIT rates	499
6.2	Soft-Error FIT rates.....	499
7	MISCELLANEOUS	500
7.1	Abbreviations	500
7.2	Literature list	501

List of Figures

Figure 1: ERTEC 200P use cases	19
Figure 2: Application operation with external host	20
Figure 3: Application operation without external host	21
Figure 4: ERTEC 200P Step2 block diagram	24
Figure 5: ARM926 subsystem	26
Figure 6: JTAG chain	31
Figure 7: AHB/APB bridge, block diagram	49
Figure 8: APB bridge, timing	50
Figure 9: EMC Block Diagram	158
Figure 10: EMC interface with two SDRAM / two Burst Flash configuration	159
Figure 11: EMC interface with only one SDRAM / one Burst Flash configuration	160
Figure 12: EMC interface with asynchronous RAM	160
Figure 13: EMC, Address Space	161
Figure 14: EMC, Notation definition for a read access (ASYNC)	164
Figure 15: EMC, Notation definition for a write access (ASYNC)	165
Figure 16: EMC, Application Example SDRAM	166
Figure 17: EMC, Application example: Combination of SDRAM and asynchronous Interfaces	166
Figure 18: EMC, Interface Signals	167
Figure 19: Block diagram of the host interface	182
Figure 20: XHIF, Symbol and Signals	184
Figure 21: XHIF interface adjustment	186
Figure 22: Block diagram of the peripheral interface	198
Figure 23: Block diagram of PN-ICU for PN_IRQx(0/1:0) group interrupts	231
Figure 24: Block diagram of PN-MUX for PN_IRQx(2:1/15:2) single interrupts	232
Figure 25: PNPLL with 3 application time blocks (application connection)	233
Figure 26: Quartz wiring	237
Figure 27: Clock source for the Ethernet connection	238
Figure 28: ERTEC 200P reset matrix	242
Figure 29: PLL startup phase	243
Figure 30: Block diagram of the IO filter	248
Figure 31: Block diagram of the filter structure of a channel	248
Figure 32: Block diagram of the central clock divider	249
Figure 33: Signal in signal filtering	251
Figure 34: Block diagram ARM926 Watchdog	260
Figure 35: XWD_OUT0/ WD_INT signal sequence	263
Figure 36: XWD_OUT1 signal sequence	264
Figure 37: TIMER block diagram	273
Figure 38: F-timer block diagram	303
Figure 39: Baud rates UART when $F_{UARTCLK} = 125 \text{ MHz}$	307
Figure 40: SPI Block Diagram	340
Figure 41: SPI flash: serial output timing	343
Figure 42: SPI flash: serial input timing	343
Figure 43: Block diagram of a GPIO module	350
Figure 44: GPIO, IO Circuit	352

Figure 45: Definition of time reference	451
Figure 46: MII interface	464
Figure 47: MDIO interface	464
Figure 48: Debug interface	472
Figure 49: ARM926 trace interface	472
Figure 50: Spike Filter Implementation	475
Figure 51: Oscillator Circuitry Layout Example	477
Figure 52: Connection of an external oscillator	478
Figure 53: Recommended for PLL Power Supply Filter	479
Figure 54: UTP circuit	482
Figure 55: FX circuit	484
Figure 56: FX circuit unused pins	485
Figure 57: SD level translation circuit	486
Figure 58: Recommendation for handling special function signals	487
Figure 59: 400-ball SIP-FPBGA	493

List of Tables

Table 1: Types of ARM926EJ-S access to I/D-TCM / AHB	29
Table 2: Boot, ERTEC 200P boot modes	32
Table 3: Boot mode adjustment	36
Table 4: Startup Times	37
Table 5: Internal bus system, parameters of internal buses	44
Table 6: Fixed priority assignment (no default)	45
Table 7: AHB Master-slave coupling	46
Table 8: Access Error Register (from SCRB)	53
Table 9: Host INTA, Interrupt sources	68
Table 10: Host INTB, Interrupt sources	69
Table 11: DMA, Table of HW_JOB_START_Signals	118
Table 12: DMA, Table of HW_DMA_REQ-Signals	121
Table 13: The DMA RAM address space.	154
Table 14: EMC, Connection to the Memory devices	168
Table 15: EMC – Address map	169
Table 16: Overview of ERTEC 200P clocks	236
Table 17: Data width of peripherals	247
Table 18: Sample filter times for I-filter	250
Table 19: Timing parameters for SPI flash (75 MHz)	344
Table 20: Overview of alternate functions (A – C)	355
Table 21: ERTEC 200P, ID register	365
Table 22: Bootmodi adjustment	366
Table 23: Configuration adjustment	367
Table 24: IO – pin count overview	426
Table 25: Signal Description and Pinning ERTEC 200P	449
Table 26: Constraining PNPLL Interface	467

1 INTRODUCTION

1.1 Overview of the ERTEC 200P

In the following manual ERTEC 200P means ERTEC 200P Step1/Step2 and ERTEC 200P-2 means ERTEC 200P Step2.

The ERTEC 200P provides switch functions covering all necessary mechanisms for targeted frame forwarding. The ERTEC 200P also features a function for preparing frame data for a given user interface.

ERTEC 200P adds IRT communication to the current IRT, RT and NRT communication for Ethernet. Standard NRT mode is not suitable for real-time control technology applications for modern control and drive concepts, as data has to be processed by the TCP/IP stack and runtimes in the network are variable. Improvements can be achieved with RT, which operates using the same switch mechanisms as NRT. RT prioritizes the frames, increasing network determinism, and processes data for the end nodes to achieve greater throughput.

A further increase in determinism is achieved by switching to IRT, in which cyclic communication runs in reserved time phases. A distinction is drawn between address-based forwarding and topological forwarding (planned communication).

Even better performance can be achieved with IRT High Performance, which operates on the basis of fast-forwarding multicast addresses (with frame ID in Octet 5/6) and uses pack frames (which pack together multiple devices). This significantly reduces the cut through time, which in turn leads to much shorter cycle times.

1.1.1 Mechanisms for PROFINET IO (NRT communication)

The basis for NRT requirements is switch functions for standard Ethernet (NRT) with NRT fragmentation for shorter send clocks. This is what is known as the address-based forwarding of frames. The required function is specified in the IEEE 802.1D /1/ and IEEE 802.1Q /2/ standards (see chapter 7.2). In addition to the switch functions, there is another function for receiving or for transferring data frames over the user interface to the host. The requirements for standard Ethernet also include all those functions that are needed with external PHYs or, as in the case of ERTEC 200P, with the integrated PHYs.

1.1.2 Mechanisms for PROFINET IO (RT communication) RTC1/2

The requirements for soft real-time Ethernet (RT) are based on Functional Specification PROFINET-IP. Below are just the main RT requirements:

- Supports cyclic and acyclic RT traffic
- Greater frame traffic determinism with priority tag option for RT data
- Option of high-priority forwarding for RT frames in the PN-IP
- Improved user interface performance achieved as a result of the hardware processing the user data in process images
- Option of high-priority forwarding of acyclic RT frames to the user interface

1.1.3 Mechanisms for PROFINET IO (IRT communication) RTC3

With IRT, cyclic communication operates in reserved time phases. You can choose between address-based forwarding and topological forwarding (planned communication). The cut through time is ca. 2.1 µs (incl. PHY).

In address-based forwarding, the control information required for frame transfer only needs to be saved with the sender and with the recipient. The route is determined automatically with the MAC target address. This type of forwarding does potentially imply non-deterministic time behavior; for simple topologies (line and ring)¹, it is however possible to guarantee not the receipt of an individual frame at a specific time but the arrival of all frames of this type within the reserved phase.

In topological forwarding, the necessary control information is saved for each individual frame in **all** nodes, i.e. not just in sender and recipient, but also all relevant forwarding nodes. Key benefits:

- Determinism in the network with time-controlled frame forwarding (requirement for this control is a common time base for all nodes in a real-time Ethernet network)
- User interface performance supported by processing of user data by the hardware in process images

For simple network topologies, ERTEC 200P can also be used as relative forwarder. Specific forwarding times for the RT frames received and to be forwarded do not need to be configured in the IRT table in this case; only the target port information needs to be saved in the IRT table.

1.1.4 Mechanisms for PROFINET IO (IRT High Performance communication) RTC3

The forwarding time in each individual node is a defining factor for performance for systems with a large number of nodes and in particular for large line depths. IRT uses super fast forwarding (locally administrated MultiCast frames with frame ID in Octet 1/2). This allows each IRT High Performance device to forward after just the first two bytes of the DA address. The cut through time is again significantly reduced to ca. 0.8 μ s (incl. PHY), which results in a shorter forwarding time / bridge delay and thus makes shorter cycle times possible.

Typical field devices in automation technology require only a few process data bytes for cyclic user data transport. PROFINET IO always requires a complete Ethernet frame with the corresponding overhead (64 bytes) for the transmission of those few bytes. To achieve significantly better performance than current PROFINET IO and performance at least comparable to EtherCAT, better use must be made of the available bandwidth with IRT High Performance. This is achieved with Dynamic Frame Packing (DFP). The data for multiple devices are saved in one frame, the "pack frame". Each device is assigned a datagram for its data (IEC standard: Subframe). Multiple datagrams (IEC standard: Subframes) then form an Ethernet frame.

IRT High Performance supports NRT fragmentation so that smaller send clocks (62.5 / 125 μ s) are possible. The switch automatically fragments NRT frames that are too long and dynamically adjusts the fragment frames to the remaining transmission time.

¹ Note: Branches and merges are possible; however, the configuration required for deterministic behavior and the possible times are more difficult to define.

1.2 ERTEC 200P data

Manufacturer, technology and package:

- Package: FPBGA400, 17 mm x 17 mm, System in Package (SiP)

Operating conditions:

- Ambient temperature: –40 to +85°C
- IO supply voltage: EMC interface: 1.8V +/-10% (sep. voltage island)
Host interface: 1.8V/3.3 V +/-10% (sep. voltage island)
GPIO31..0, ...: 3.3 V +/-10%
- Core supply voltage: 1.2 V +/-0.1 V
- PHY supply voltage: 1.5 V +/-10%
- Power dissipation: max. 1.2 W (incl. 2x internal PHY)

Processor system:

- ARM926EJ-S integrated processor system (frequency 125/250 MHz)
 - 16-KByte data and 16-KByte instruction cache
 - 256-KByte instruction/data tightly-coupled memory incl. byte EDC, can be set in 64-KByte increments (I-TCM: 0 – 256 KByte, D-TCM: 256 – 0 KByte)
 - Debug compatibility with embedded ICE with JTAG interface, ETM with ETB (Embedded Trace Buffer)
 - Memory Management Unit (MMU)

Bus structure:

- Internal 32-bit structure
- Multi-layer architecture with parallel multimaster to multislaves access structure (125 MHz)
- 16/32-bit bus interface to an external SDRAM/SRAM/Flash and external I/O

PROFINET IP:

- 2 Ethernet ports with integrated PHYs (100 Mb, FD)
- IRT High Performance, IRT, RT and NRT data traffic

Interfaces:

- External memory controller (EMC), can only be operated at 1.8 V
 - SDRAM interface (125 MHz)
 - Asynchronous SRAM interface (4-chip select areas, ready control) incl. burst flash interface
- Host interface (instead of GPIO95-32), can be operated at 1.8 V and 3.3 V
 - 16/32-bit data width
- GPIO: GPIO31-0 with 16 I-filters, GPIO95-32 with 64 I-filters in parallel
- 1x I²C
- 1x I²C (in PN-IP for fiber-optic transceiver)
- 4x UART including one UART as default without alternate function
- 2x SPI1 (master / slave,)

General functions:

- Internal clock generation (quartz oscillator, PLL)
- Integrated boot ROM (8 KByte)
- 6 x timer
- Watchdog
- F-counter
- GDMA controller
- ARM926 interrupt controller

Test functions:**Boundary scan**

The complete ERTEC 200P can be switched to external high impedance with a boundary scan command.

1.2.1 System Reliability

A soft error rate of fewer than 1000 FIT at an altitude of 2000 m is required for the complete ASIC. As the required SER cannot be achieved with the current level of RAM complexity without additional error mechanisms, an EDC mechanism (Error Detection and Correction) is in place for all more complex RAMs. The EDC circuit detects and corrects 1-bit errors and detects but does not correct 2-bit errors.

Logically adjacent bits are located in the manufacturer's single port RAMs in physically separate (different) words; 2-bit errors can therefore be corrected by the error correction in logically adjacent bits. RAMs with EDC can therefore be ignored for the soft error rate. A more precise breakdown of FIT rates can be found in 6.

1.2.2 Electromagnetic Compatibility (EMC)

The ERTEC 200P ASIC and the systems/devices implemented with it are designed for EMC in accordance with EN61000-6-2 (Generic standards - Immunity) and EN61000-6-4 (Generic standards - Emission standard).

These standards are the device standards for use in industrial environments. For compliance with these device standards, however, the EMC of the individual integrated circuits used (in particular immunity to interference in operation) is key.

1.3 ERTEC 200P Use Cases

The ERTEC 200P supports 2 use cases (**UC1**, **UC2**) that are shown in Figure 1: ERTEC 200P use cases.

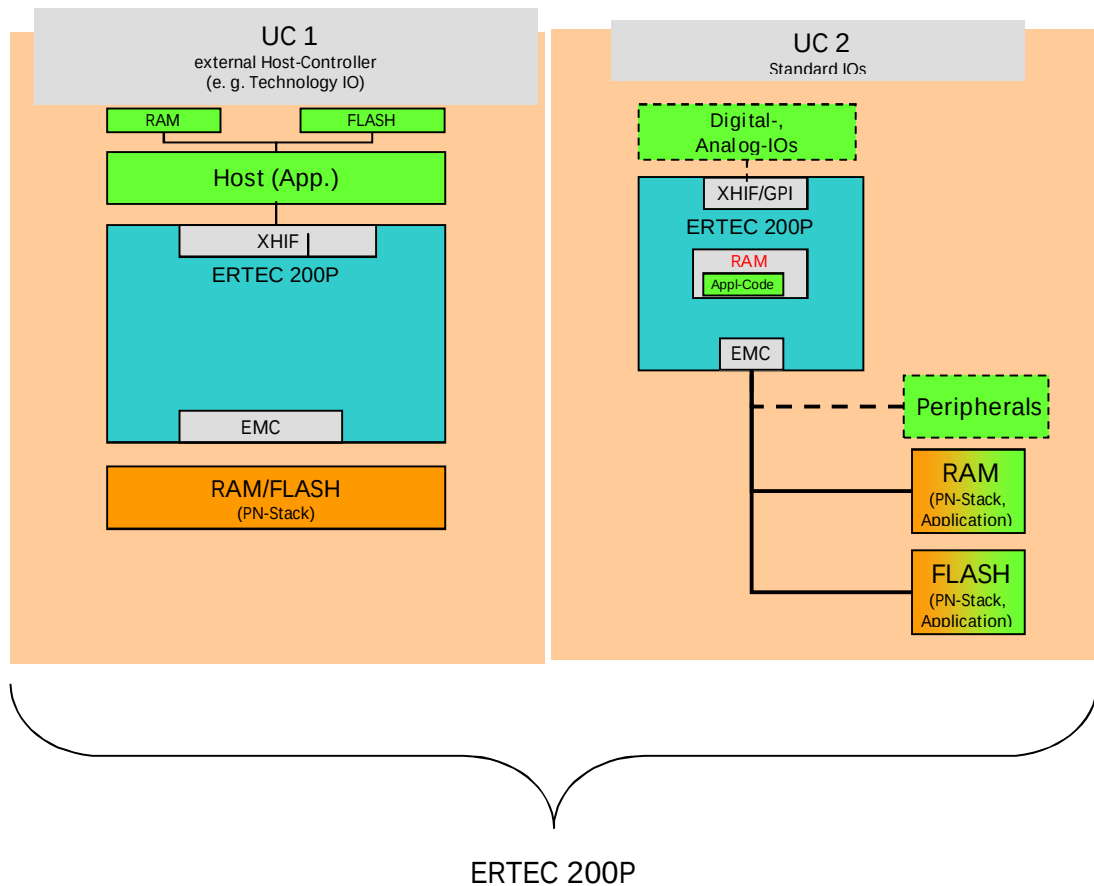


Figure 1: ERTEC 200P use cases

1.3.1 Use Case 1 (UC1): Operation with external host

At many devices the communication and the application (technology) are located on separate controllers. This is represented at the ERTEC 200P by the UC1. Only the PROFINET IO stack then runs on the ERTEC 200P. The parallel host interface (XHIF) with 16 or 32 bits can be connected with the external host.

Figure 2 shows the application with an external host processor. At the ERTEC 200P a SDRAM has to be connected to the memory interface (EMC) in which the PN stack is loaded when the host is boosted.

The acyclic communication data and the configuration data are stored, for example, in the integrated TCM (Tightly Coupled Memory) (256 kbytes) of the ARM926 and the cyclic data in the IO-RAM of the PER-IF. Transfer of the cyclic data (Arrow 1 + 2) is controlled by the PN-IP and by the host. The PN-IP transfers the data into the IO-RAM of the PER-IF. The host then transfers the data from there to its application.

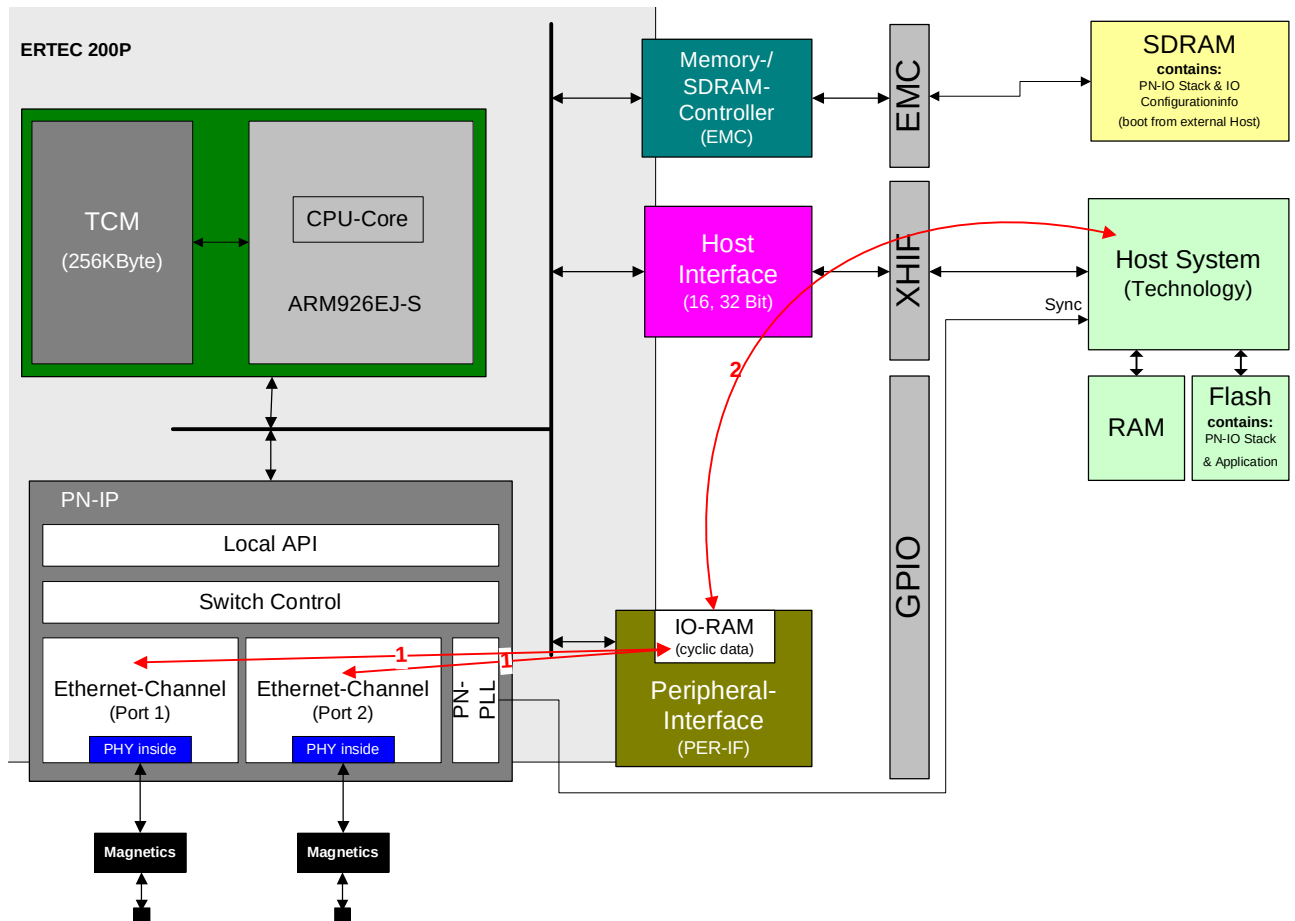


Figure 2: Application operation with external host

At GPIO31:0 outputs from the PLL are available in the PN-IP for synchronous applications on the host. The data transfer to / from the external host is then controlled through these phase signals and corresponding interrupts. If the host has an own PLL for cycle synchronization, these can be synchronized by the PLL of the ERTEC 200P.

1.3.2 Use Case 2 (UC2): Operation without external Host

The ERTEC 200P is available for the realization of complex standard IO and remote IO applications with notably more application code. The ERTEC 200P has an external memory interface (flash, SDRAM, SRAM, peripherals) and a small integrated memory (256 kbytes). The PN IO stack and the application lie in the external memory. High-performance acyclic communication data or application code that is to be executed rapidly are loaded in the integrated memory of the ERTEC 200P.

The digital and analog IOs are connected to the ERTEC 200P pins. Cyclic data can be controlled by the ARM926 or the GDMA controller via 2x SPI. Alternatively an external bus controller can also be connected via the EMC.

The following profile is supported:

- PN stack and application are combined as sources
- Time-critical application code (≤ 256 kbytes) can be loaded into the integrated RAM

Operation without external host

Figure 3 shows the application of the standard IO.

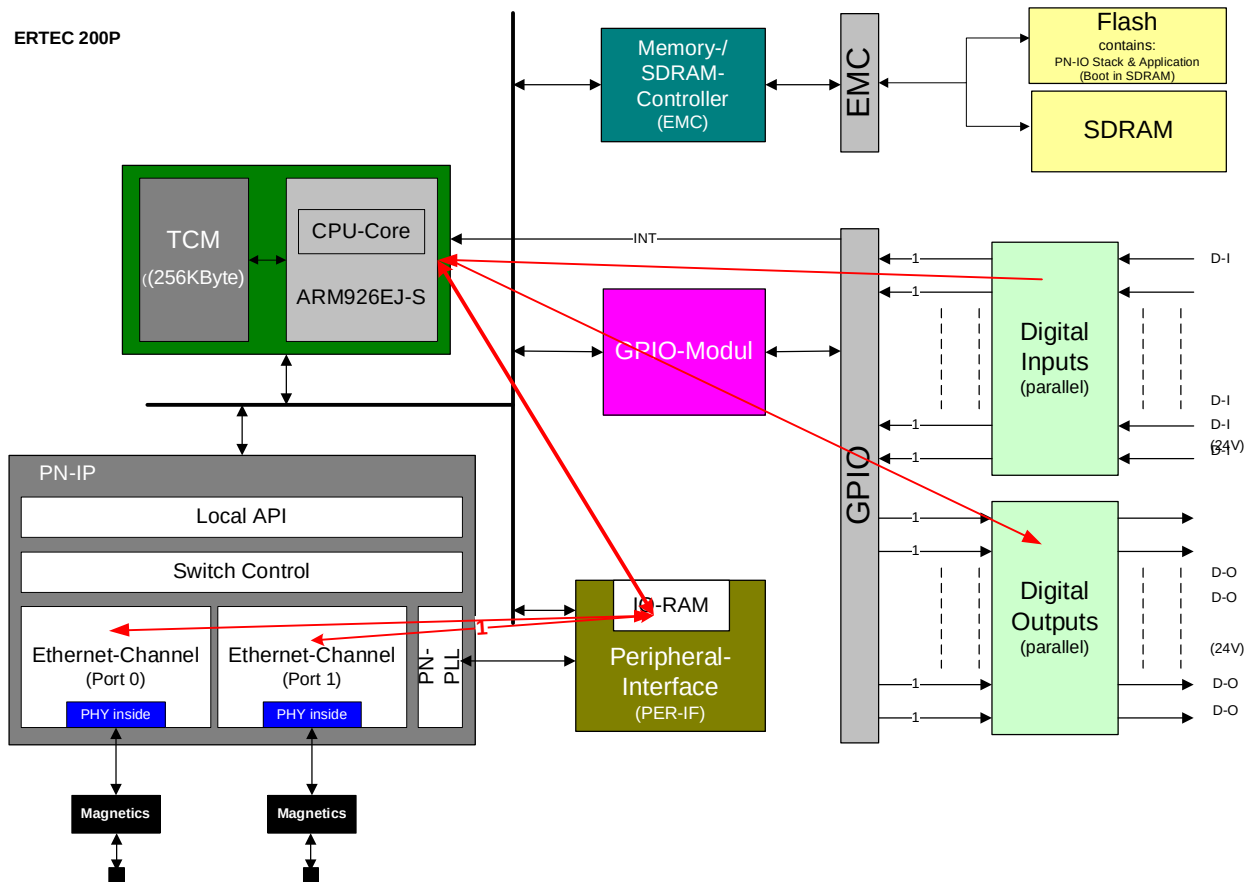


Figure 3: Application operation without external host

1.4 Application notes

The user should pay attention to the following application notes:

1.4.1 EMC SDRAM Interface

The EMC of the ERTEC 200P-2 has two possible settings for clocked signal output (address, data and control signals) to an external SDRAM. These can be configured over *EXTENDED_CONFIG.SODM* (see 2.3.5.8):

- *EXTENDED_CONFIG.SODM* = '0': SDRAM signal output with feedback clock at pin

CLK_I_SDRAM (register value after reset)
- *EXTENDED_CONFIG.SODM* = '1': SDRAM signal output with internal system clock

To achieve the timing below, configure *EXTENDED_CONFIG.SODM* = '0' (default value). *EXTENDED_CONFIG.SODM* = '1' is not permitted

Important:

In ERTEC 200P Step1, configure *EXTENDED_CONFIG.SODM* = '1'.

1.4.2 EMC Burst Flash Interface

The EMC of the ERTEC 200P has two possible settings for clocked signal output (XAV_BF) to external burst flash. These can be configured over *EXTENDED_CONFIG.BFODM* (see 2.3.5.8):

- *EXTENDED_CONFIG.BFODM* = '0': Burst flash signal output with feedback clock at pin CLK_I_BF (register value after reset)
- *EXTENDED_CONFIG.BFODM* = '1': Burst Flash signal output with internal system clock (value required for correct operation!)

To achieve the correct timing, configure *EXTENDED_CONFIG.BFODM* = '1'.
EXTENDED_CONFIG.BFODM = '0' (default value) is not permitted.

The EMC of the ERTEC 200P has two possible settings for selecting the operating frequency of externally connected burst flash. These can be configured over *BF_CONFIG.BF_CLK_F* (see chapter 2.3.5.8):

- *BF_CONFIG.BF_CLK_F* = '0': Half-rate BF, 62.5 MHz (register value after reset)
- *BF_CONFIG.BF_CLK_F* = '1': Full rate BF, 125 MHz (value is not supported in ERTEC 200P !)

To achieve the correct timing, configure / retain *BF_CONFIG.BF_CLK_F* = '0' (default value). *BF_CONFIG.BF_CLK_F* = '1' is not permitted.

Note:

The timing specifications for the ERTEC 200P board layout only apply for the above settings.

1.4.3 No free-running frequency at quartz break

To detect a quartz break, the LOSS signal is generated with the output frequency (in this case the free-running frequency) of the PLL (see chapter 4.4) to allow, for example, the F-timer to run and this event to be detected even following a quartz break, the PLL must continue to provide a free-running frequency despite the absence of a quartz clock (i.e. CLKP_A open or clamped at '1' or '0'). This free-running frequency must be:

- < 500 MHz AND
- > 4 x 32 kHz

(according to specifications: 350 MHz)

With the current PLL design, this is not the case. When there is no quartz clock, the PLL returns a free-running frequency of 0 Hz.

Background:

With the PLL used in ERTEC 200P, only the CLK output (CLKOA) generates a free-running frequency; the outputs CLKOB and CLKOC do not. As multiplication (factor of 20) of the quartz clock (25 MHz) to 500 MHz is only possible at the CLKOB output (a

max. multiplication factor of 16 is possible at CLKOA) and CLKOB is therefore used to generate internal ASIC clocks, no free-running frequency is visible.

1.4.4 16-bit Data Access over XHIF in Buffered Mode

If the parallel host interface (XHIF) is used with a 16-bit data bus (*8-bit data bus is not supported in ERTEC 200P*), buffered mode must be configured in *XHIF_0/1_Px_CFG* for consistent 32-bit access over ML-AHB (see 2.3.6.3).

When the lower 16 bits (*XHIF_A1 = '0'*) are read/written when buffered mode is active, the next read/write access must always be to the higher 16 bits of the same 4-byte address.

Background:

*Reading the lower 16 bits (*XHIF_A1 = '0'*) triggers 32-bit AHB read access with the XHIF and the higher 16 bits read are buffered in the XHIF. The next read access with *XHIF_A1 = '1'*, **irrespective of the higher address bits** (*XHIF_A2...19*), always reads the data buffered in the XHIF.*

*Writing to the lower 16 bits (*XHIF_A1 = '0'*) will buffer the data in the XHIF **without storing the higher address bits** (*XHIF_A2...19*). The next write access with *XHIF_A1 = '1'* always, irrespective of the higher address bits (*XHIF_A2...19*), triggers 32-bit AHB write access in the XHIF. This write access always uses the data buffered in the XHIF and the data currently pending, and applies to the address currently pending at the XHIF and implemented at AHB.*

This is allowing us to grow and consolidate our business: we are developing key technical product features more effectively than our competitors, and are facilitating the implementation of our technology in actual products. This is in turn acceptance on the part of end users and device manufacturers.

As ERTEC 200P is the first ASIC to integrate the new performance measures, the rollout date for PROFINET with performance improvements depends on the launch date for ERTEC 200P.

Addressing known errors, achieving performance improvements and optimization, and more efficient adjustment to I/O technological features are other equally important aspects to consider.

2.3 Description of Individual ERTEC 200P Blocks

The following sections go into more detail on the ERTEC 200P modules required:

- Processor subsystem (ARM926, TCM, debug support, booting)
 - ARM-ICU
- ML-AHB (internal bus system)
- Access violations
- PROFINET
 - PerIF (peripheral interface)
- EMC (External Memory Controller)
- HostIF
 - XHIF (External Host Interface)
- GDMA (Global DMA)

- APB peripherals
 - SPI 1 – 2
 - I2C
 - UART 1 - 4
 - Timer 0-5
 - Watchdog
 - F-timer
 - GPIOs 0...95
 - I-filter
 - SCRB (System Control Register Block)

- CRU (Clock Reset Unit)

2.3.1 Processor Subsystem (ARM926)

The subsystem used is an ARM926 subsystem based on an ARM9E-S processor. The ARM926 subsystem is primarily for the application and non-runtime-critical routines of the PN stack. Figure 12 shows the structure of the ARM926 subsystem. It consists of a core system, the JTAG interface and a TCM_Block_926. Alongside the ARM926EJ-S processor, the core system contains a data and an instruction cache, a memory management unit (MMU), separate connections to each AHB layer for instruction and data, an ETM9 (medium+) Trace Macrocell and an integrated trace buffer (ETB11). The tightly coupled memories for data and instructions are located in TCM_Block_926. AHB access to D-TCM is with a 'DMA-DTCM Access Controller'. I-TCM is only operated by ARM926EJ-S and cannot be accessed by the AHB. ARM926EJ-S is described in detailed under /7/.

Of the AHB masters, the host interface, PN-IP and GDMA can access D-TCM in the ARM926 subsystem. ARM926EJ-S has access to the AHB slaves, as detailed in 2.3.2.3.2.

2.3.1.1 Features

Macro Type:	CB-90, soft macro
Function summary:	ARM926EJ-S 16 KByte Daten- und 16 KByte Instruction-Cache 256 KByte Instruction/Data- Tightly Coupled Memory inkl. Byte EDC, einstellbar in 64 KByte Schritten (I-TCM: 0 – 256KByte, D-TCM: 256 – 0KByte) Debugfähigkeit durch Embedded ICE mit JTAG Interface, ETM-Zelle mit ETB (Embedded Trace Buffer) Memory Management Unit (MMU) Little endian support only
Operating frequency:	125/250MHz ARM926 125MHz AMBA bus system
Process:	CB-90

2.3.1.2 Block diagram

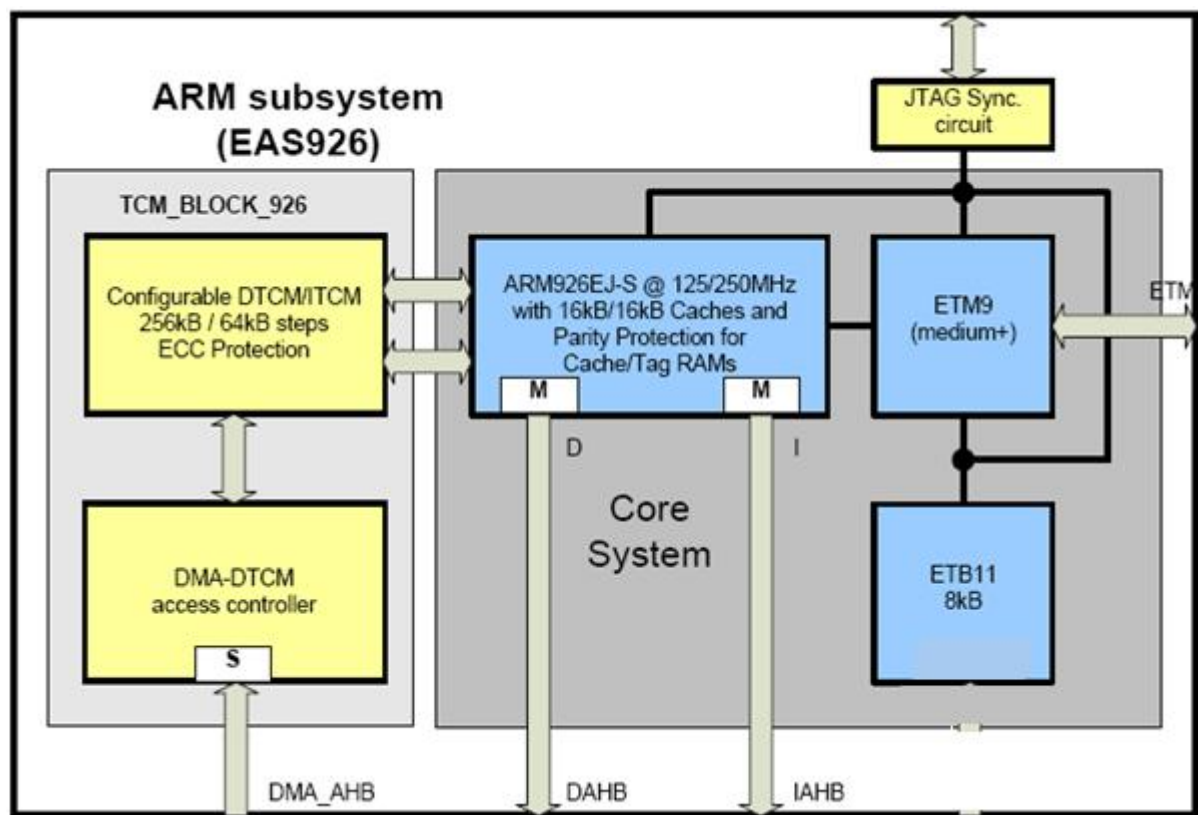


Figure 5: ARM926 subsystem

2.3.1.3 ARM926EJ-S processor

ARM926EJ-S revision r0p5 contains an ARM9E-S processor core with Harvard architecture. Unlike the ARM9 processor core of ARM9TDMI, this processor core has the improved **v5TE** architecture. The main improvements relate to ARM/thumb interworking (faster switching between ARM and thumb code segments) and an improved multiplier structure. Alongside the processor core, ARM9E-S also includes embedded ICE logic RT. This logic is controlled by the integrated JTAG interface. Document /9/ (see chapter 7.2) detail the connection of the embedded ICE interface connector to ERTEC 200P (reset, JTAG connector ...). A detailed description of ARM9E-S can be found in /6/.

The processor and TCMs (see 5.2.3) can be operated at 125 or 250 MHz. Setting is with the CONFIG(1) pin (see 2.3.10.9.3). The AHB interfaces always operate at 125 MHz.

Important: ARM926EJ-S can only be operated in little-endian mode.

The following bit must be set for the ARM926EJ-S core to detect unaligned access:

- CP15 register c1 (Control Register), set bit 1 (A-bit) to enable fault checking of address alignment.

2.3.1.3.1 Cache structure of ARM926EJ-S

The functional scope of the ARM926 cache has the following features:

- 16 KByte of instruction cache
- 16 KByte data cache
- Write buffer for data cache write back function
- Caches are 4-way set associative caches with 1-KByte segments
- One segment consists of 32 lines and each line contains 32 bytes (i.e. 8 words).

The content of the cache segments can be "locked". This lock function makes it possible permanently to retain the instruction set for fast routines in the I-cache. In ARM926EJ-S, this mechanism can only be implemented on a segment-specific basis. More information on caching can be found in /7/ (see chapter 7.2)

If a QVZ interrupt occurs during an I-cache refill following an AHB error (caused for example by a WRAP8 at the end of the memory), no Prefetch Abort Exception is triggered in ARM926EJ-S. The I-cache means that allocation to the erroneous word is no longer possible for the ARM. However, it is advisable to configure the MMU so that an ARM926EJ-S exception is generated in this case. An MMU exception is then triggered when you leave the memory area.

The I-cache and D-cache have one byte of parity. The I-tag and D-tag RAM is also parity-protected. If an error occurs when reading the I-cache or the relevant I-tag entry, the cause of the error (I_Cache_Parity, I_Tag_Parity) is saved in the SCRB register 'EDC Event' (see 2.3.10.9.22) and the interrupt 'EDC_Event' IRQ48 is triggered (see 2.3.2.14). An error while reading the D-cache or the relevant D-tag entry generates the entry (D_Cache_Parity, D_Tag_Parity) in the 'EDC Event Register' with the 'EDC_Event' interrupt. '0h' must be written to the EDC event register to clear the error.

The parity bits are undefined after reset. If parity monitoring of the caches is to be used, the caches must be initialized by the SW (each cache line must be initialized in a loop). Until this point, the parity logic is disabled. Control is with the SCRB register

'EDC_PARITY_EN' (see 2.3.10.9.22). 'I_CACHE_PAR_EN-Bit' enables the parity bits of the I-cache and I-tag, and 'D_CACHE_PAR_EN-Bit' the parity bits of the D-cache and D-tag. After reset, both bits are set to '0' and the parity bits are disabled. The SW can set these bits to '1' after cache initialization.

2.3.1.3.2 ARM926 Tightly Coupled Memories (ARM926_TCM)

The I-TCM and D-TCM are located in TCM_Block_926. The TCM configuration includes 256 KByte² for instruction and data TCM (I/D-TCM). The memory operates at the clock rate of the processor, 125/250 MHz. The memory consists of segments of 64 KByte, which can be assigned either to the instruction or to the data TCM. This results in a configurable memory size of 0 – 256 KByte for the I-TCM with 256 – 0 KByte for the D-TCM. After reset, the TCM926 configuration is 256 KByte D-TCM. The final setting is made during booting by the boot loader (SW), in the SCRB register 'TCM926_Map' (see 2.3.10.9.22). After this reconfiguration, the boot loader must still display the TCM memories in the address range of the coprocessor interface (CP15 c9) (see 2.4.1 and 2.4.2). Please note that I-TCM and D-TCM can only be displayed in the address range in increments of 2ⁿ. If, for example, a physical size of I-TCM = 192 KByte / D-TCM = 64 KByte has been selected in 'TCM926_Map Register', ARM926 can only assign I-TCM an address range of 256 KByte and D-TCM the required address range of 64 KByte. The values 256 KByte for I-TCM and 64 KByte for D-TCM are therefore to be read out in the coprocessor interface of ARM926 (CP15 c9).

The SW must then not access the unassigned area / hole (I-TCM: 192 – 256 KByte). If ARM926 accesses the unassigned area in the event of an error, either 'Invalid I-TCM926 Access Interrupt' (access to the I-TCM hole) or 'Invalid D-TCM926 Access Interrupt' (access to the D-TCM hole) is triggered (see 2.3.2.14).

Unpredictable access to the I/D-TCM hole cannot be prevented by ARM926; 'Invalid I/D-TCM926 Access Interrupt' is generated. The situation can be dealt with if the MMU decodes the hole and 'Invalid I/D-TCM926 Access Interrupt' is blocked. The MMU is not designed for unpredictable access.

I-TCM and D-TCM are each assigned an address range of 256 KByte in implementation. This covers all possible combinations of I-TCM and D-TCM. Once shown, the I-TCM is in the ARM926 address range from 0x0000_0000h and the D-TCM from 0x0800_0000h. More information on TCMs can be found in [7] (see chapter 7.2).

Only ARM926 D-TCM can be accessed from the AHB. In the event of access to the unassigned D-TCM address range from an AHB master, access is prevented and a QVZ interrupt is triggered (see 5.8.1).

AHB access to D-TCM is with the 'DMA-DTCM Access Controller'. The I-TCM is only used by ARM926EJ-S and cannot be accessed by the AHB.

The I-TCM and D-TCM have an EDC code, which is assigned byte by byte (5 bits per byte, 1-bit error, correctable; 2-bit error, detectable). If an error occurs during reading, the cause of the error (I/D-TCM926-1B: 1-bit error corrected or I/D-TCM926-2B: 2-bit error detected) is saved in the SCRB register 'EDC_EVENT' (see 2.3.10.9.22) and the interrupt 'EDC_Event' IRQ48 is triggered (see 2.3.2.14). '0h' must be written to the EDC event register to clear the error.

² 1 KByte is equal to 1024 bytes.

After ARM926-TCM reset, the I/D-TCM is automatically initialized by the HW. The initialization process (incl. EDC bits) for the complete I/D-TCM takes ca. 16.5 μ s. The end of the initialization process is signaled in the SCRB register 'EDC_INIT_DONE' (see 2.3.10.9.22) (I_TCM926_INIT_DONE, D_TCM926_INIT_DONE). The EDC logic can also be disabled using the SCRB register 'EDC_PARITY_EN' (see 2.3.10.9.22). The 'EDC_DISABLE_ARM926' bit deactivates the EDC logic. The EDC logic is enabled after reset.

The D-TCM can be written and read by the AHB masters (PN-IP, host interface and GDMA). Access to the D-TCM by one of these masters stalls the ARM (pipeline paused). Per word of transfer, this stall usually only lasts for one processor clock cycle to allow transfer from the 'DMA-DTCM Access Controller' (part of the ARM926 subsystem) to the D-TCM or vice versa. Arbitration is round robin, i.e. access to the D-TCM is executed time about by ARM and the 'DMA-DTCM Access Controller'. Arbitration is also carried again after each individual instance of access for burst transfer.

Table 1 lists the various types of ARM926EJ-S access to I/D-TCM and the AHB and the potential errors:

EAS 926 , 250 MHz	access types								Description:	
Memory area	AHB write access	AHB read access	AHB read getting an ECC detect	AHB read getting an ECC error	TCM write access	TCM read access	TCM read ECC detect	TCM read ECC error		
ITCM area (from AHB not possible)	na1	na1	na1	na1	ok	ok	I-TCM926-1B + I-TCM926-2B	I-TCM926-1B + I-TCM926-2B	ok	no error
ITCM area hole (from AHB not possible)	na1	na1	na1	na1	Invalid I-TCM926 Access	Invalid I-TCM926 Access	na2	na2	na1	not available / access is not possible
ITCM area mirror (not possible)	na1	na1	na1	na1	na1	na1	na1	na1	na2	not available / access never issues an ECC detect/error
									AHB error	AHB error response via HRESP
									D-TCM926-1B	AHB ECC detect signal
DTCM area	ok	ok	D-TCM926-1B + D-TCM926-2B	D-TCM926-1B + D-TCM926-2B	ok	ok	D-TCM926-1B + D-TCM926-2B	D-TCM926-1B + D-TCM926-2B	D-TCM926-2B	AHB ECC error signal 8ns width for the DTCM
DTCM area hole (from AHB not possible)	na1	na1	na1	na1	Invalid D-TCM926 Access	Invalid D-TCM926 Access	na2	na2	I-TCM926-1B	AHB ECC detect signal 8ns width for the ITCM
DTCM area mirror (not possible)	na1	na1	na1	na1	na1	na1	na1	na1	I-TCM926-2B	AHB ECC error signal 8ns width for the ITCM

Table 1: Types of ARM926EJ-S access to I/D-TCM / AHB

2.3.1.3.3 Memory Management Unit (MMU)

The MMU (see ARM926EJ-S Technical Reference Manual /7/) supports a demand page virtual memory system which may be required by operating systems such as Linux and ECOS depending on the application. The MMU contains the access protection mechanisms for all memory access. Address translation, access protection and region type are saved in one TLB (Translation Lookaside Buffer). Separate TLBs are available for instruction and data. These TLBs are automatically evaluated and updated by the MMU hardware.

- Page size: 1 MByte, 64 KByte, 4 KByte and 1 KByte
- Separate TLBs for instruction and data
- Access attributes can be changed without a TLB flush
- A fast context switch enables virtual address remapping in a 0 – 32 MByte area
- TLB entries can be locked

The MMU RAMs are **not assigned parity**.

2.3.1.3.4 Bus Interface of the ARM926 Processor

The ARM926 processor has separate AHB interfaces for opcode fetches (ARM926-I) and data transfers (ARM926-D). The interfaces operate at 125 MHz. The data bus width and address bus width are each 32 bits. The data bus interfaces operate over a write buffer (16-stage FIFO). When the write buffer is used, data write sequences are transferred to the write buffer and the data unit of the processor can continue processing immediately. More information on the bus interface and the write buffer can be found in [7] (see chapter 7.2.)

2.3.1.3.5 ARM926 Embedded Trace Macrocell (ETM9), Trace Buffer (ETB11)

To support debugging, an ETM9 revision r2p2 (medium+³) module with integrated trace buffer (ETB11) is connected to the ARM926EJ-S system in the ERTEC 200P. This module allows an instruction and data trace. The trace buffer is 2Kx32 bits and supports both ARM926 clocks (125 / 250 MHz).

Alongside the integrated trace buffer, an external ARM926 trace port at the XHIF port is also available as an alternative function for **ARM926 operation at 125 MHz**. The ARM926 trace port is enabled with the configuration pins (CONFIG(6..3) = 1110b, see 2.3.10.9.3).

The ETM module receives all signals required for the data and instruction trace from the processor. The ETM9 module is operated over the JTAG interface. The trace information is output at the trace port over a FIFO or entered in the internal trace buffer. In normal mode (not multiplexed or demultiplexed mode), the trace port is operated with a width of 4, 8 or 16 data bits. TRACEPKT outputs of the trace port at GPIO53-38 (when CONFIG6:3 = 1110, see 2.3.10.8.4.1) that are not used by the debugger remain high-impedance.

Only the half-rate mode is supported as clock mode. The trace clock runs at half the processor frequency (62.5 MHz with a processor clock of 125 MHz) and the trace data have a maximum change frequency of 125 MHz. The debugger accepts the data with both trace clock edges. Standard Lauterbach or Hitex debuggers support trace clock frequencies of up to 200 MHz.

The trace buffer (ETB11) is connected to the AHB and could be employed for normal rapid data storage if not used. However, as an external connection to the module is required for the XTRST pin (see note on the JTAG interface in 0), AHB access to the ETM trace buffer SRAM is not possible during operation.

Recording in the trace buffer can be controlled by the external debugger over the JTAG interface or by the SW on the ARM926EJ-S⁴. A register set that can also be accessed from the AHB is available in ETB11 for this purpose. ARM926EJ-S has two interrupts for processing the trace (ETB_ACQCOMP -> Required trace is saved in the ETB; ETB_FULL -> Trace overrun in the ETB) (see 2.3.2.14). The trace can, however, only be read from the trace buffer once tracing is complete (ETB_ACQCOMP interrupt).

Recording in the internal trace buffer can be suppressed when an external trace port is used.

³ The medium+ ETM9 cell has a larger FIFO (18 -> 45 bytes).

⁴ ETM trace condition configuration is only possible from the external debugger and not over the AHB interface of the ETB11.

More information is available in specifications /4/ and /5/ for the ETM9 cell (see chapter 7.2). In addition to these specifications, the following 8 MMD areas have been decoded in ERTEC 200P with the hardware.

2.3.1.4 Debug Support

2.3.1.4.1 Debug Configuration

Two TAP controllers are connected in the JTAG chain (see 2.3.1.4.2):

- TAP#0: ETB of ARM926EJ-S
- TAP#1: ARM926EJ-S

This produces the following queue:

- TDI → ETB → ARM926EJ-S → TDO

The number of instruction register bits is (see Figure 6: JTAG chain):

- ETB IR length: 4 bits
- ARM926EJ-S IR length: 4 bits

```

+++++
JTAG scan chain diagnosis:
IR_Width = 8
DR_Width = 2
ID_Width = 64
ID_TAP#1 = 0x7926021 NEC
ID_TAP#0 = 0x2b900f0f ARM9/ETB
ID_Code = 0x2B900F0F07926021
+++++

```

Figure 6: JTAG chain

2.3.1.4.2 ARM926EJ-S Debug Interface

Only external debuggers with JTAG interfaces are supported.

The JTAG interface comprises the standard signals (XTRST, TCK, TDI, TMS, TDO and XSRST). The RTCK clock output is also available. For debuggers that support RTCK, a JTAG clock rate of TCK = 32 MHz is possible. Otherwise, the maximum clock rate is TCK = 16 MHz.

For ARM926EJ-S, the two signals DBGREQ and DBGACK are also supported over the JTAG interface. With DBGREQ active, the external debugger forces ARM926EJ-S straight to the debug state. DBGACK active signals to the debugger that ARM926EJ-S is in the debug state. Both signals are available as alternate functions at the GPIO31-0 pins. Before these debug signals can be used, they must be enabled in the GPIO31-0 block. If DBGREQ is not enabled in its alternate function, DBGREQ is internally not active and ARM926EJ-S is not in the debug state.

2.3.1.5 Boot Process for the ERTEC 200P

A *primary boot loader* is integrated into ERTEC 200P (8 KByte boot ROM at the APB). The ERTEC 200P can be booted with a number of boot media and the boot modes required for the various different configurations are therefore supported. Booting is possible with one of four sources:

Boot mode	Bootling...	Use case
1 ... 2	...from the external NOR flash conneted to the EMC	Use case 2 (Standalone Application)
5, 6	... from firmware copied to the TCM over the SPI master interface of the ERTEC 200P	<i>Future use case, for example for low-cost applications</i>
7	... from an external host CPU that loads the firmware to the TCM over the XHIF	Use case 1 (External Host)

Table 2: Boot, ERTEC 200P boot modes

Bootling can be triggered by:

- HW reset (PowerOn reset, XRESET)
- Watchdog reset (WDT expires)
- Software reset

The complete list of reset triggers can be found in 2.3.10.9.4 and in the CRU (2.3.9.4).

From the boot ROM, ARM926 processes the primary boot loader, which loads a secondary boot loader from the corresponding boot medium to the D-TCM (Block3: 0x0803_0000) so that, after the switch to I-TCM (Block0: 0x0000_0000), the secondary boot loader is saved after the interrupt jump table (see 2.3.1.5.1). If an external host is responsible for booting, it transfers the secondary boot loader to the D-TCM (Block3: 0x0803_0000).

Once the secondary boot loader has been loaded, this 64 KByte segment is switched to the I-TCM and is shown at address 0x0000_0000. The secondary boot loader loaded is not lost when the system switches to the I-TCM. Following a PowerOn reset, the TCM926 configuration is 256 KByte D-TCM. As the assignment of the SRAM blocks in the I-TCM and D-TCM is completely reversed, the boot loader must copy the code image in 64 KByte blocks from the bottom D-TCM Block3 (0x0803_0000) → D-TCM Block 0 (0x0800_0000), i.e. in ascending order. The address order is not reversed in the TCM block. After download to the D-TCM, the final TCM926 configuration is set in *TCM926_MAP* (see 2.3.10.9.14). The boot loader must then display the TCM memories in the address range of the coprocessor interface (CP15 c9).

The primary boot loader then transfers to the secondary boot loader loaded in which the user then continues booting on an application-specific basis. One task of the secondary boot loader is to copy a code image to the I-TCM. There are two options available here:

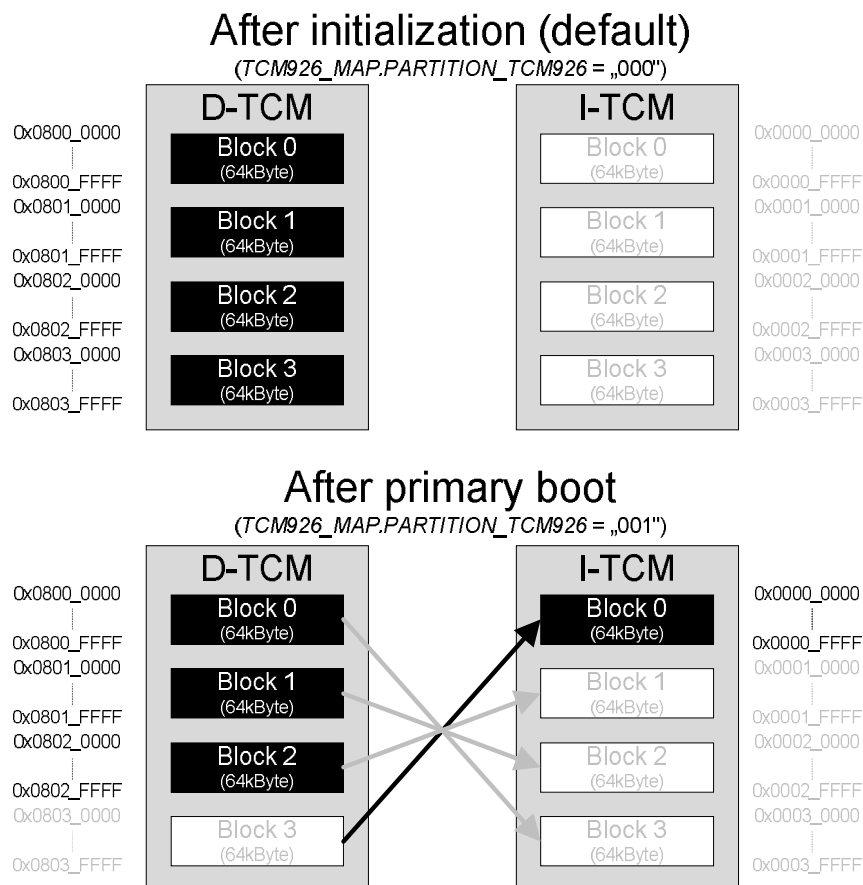
- The secondary boot loader sets the final TCM926 configuration required in *TCM926_MAP* (see 2.3.10.9.14). The code image can then be transferred by the boot medium to the I-TCM after the secondary boot loader.

- First, the 64 KByte segment that is already assigned to the I-TCM is completely filled. The code image is then loaded to the D-TCM, in the correct order. After this download, the final TCM926 configuration is set in *TCM926_MAP* (see 2.3.10.9.14). The secondary boot loader must also show the final configuration of the TCM in the coprocessor register (CP15 c9) address range.

GDMA can also be used to load the code image to the D-TCM. This code image is loaded to the D-TCM following booting by the host.

For booting from an external host, the primary boot loader creates a page for the D-TCM (boot RAM) in the 8-page XHIF registers. The remaining pages can then be set up specifically by the external host. It makes sense to set up additional pages for PN-IP, PER-IF, EMC-SDRAM / EMC-SRAM, GDMA and APB peripherals.

Important: Please note that TCM block assignment in the ARM926 subsystem is reversed when you switch from D-TCM to I-TCM.



The ERTEC 200P has a three-stage boot model. The primary boot loader, stored in the boot ROM, sets the hardware used to ensure that data can be read from the boot medium and copied to the D-TCM of the ARM926. Only the boot medium used is initially configured by the primary boot loader. The data copied comprises the secondary boot loader, which copies the actual user program to the destination memory (SDRAM, TCM or SRAM). This procedure keeps the primary boot loader very compact (max. 8 KByte are allowed), which makes it easy to test. Complete configuration of the ERTEC 200P is only implemented on a module-specific basis by the secondary boot loader or the user program, as the user knows which hardware components their system requires and how these are to be configured in the given environment. This also allows a wide range of different compilers, without having to consider special features such as INIT or ROM sections.

To provide the routines of the primary boot loader to the secondary boot loader, the primary boot code is located in the original address range of the boot ROM (address 0x4000_8000). The primary boot loader immediately branches from the mirrored area (address 0x0000_0000) to its original address range (address 0x4000_8000), where it processes the 1st-level boot code.

All boot modes with the exception of the boot mode from the external NOR flash in compile mode (see 2.3.1.5.2) process the secondary boot loader stored in the TCM. As no instructions can be processed from the D-TCM, the primary boot loader must map the data copied to D-TCM block 3 to I-TCM block 0.

2.3.1.5.1 Primary Boot Loader

The primary boot loader is a separate memory block (boot ROM, 8 KByte) which starts after a PowerOn reset from address 0x0. Depending on the boot mode settings (with boot pins or in the SCRB register *BOOT_REG*, see 2.3.10.9.2), the system recognizes the medium from which the software is to be loaded. The required system settings are only set for the download medium and the program code (secondary boot loader) is loaded to TCM. Finally the loaded code will be executed.

The following settings are **always** made by the primary boot loader irrespective of the boot mode:

- *ASYNC_ADDR_MODE*: (see /36/, page 105, register: *EXTENDED_CONFIG*)
 - When *ASYNC_ADDR_MODE* = '0':
no address shift (ERTEC 200 compatible) → MA(23:0) = HADDR(23:0).
 - When *ASYNC_ADDR_MODE* = '1':
address shift depending on configured memory data width
 - data width = 8 bit: MA(23:0) = HADDR(23:0)
 - data width = 16 bit: MA(23:0) = HADDR(24:1)
 - data width = 32 bit: MA(23:0) = HADDR(25:2)

The bit is set in the *EXTENDED_CONFIG* register of the EMC. An entry in the SCRB is therefore no longer required. After reset, the register bit is set to 0, i.e. there is no address shift. In ERTEC 200P, the bit *EXTENDED_CONFIG.ASYNC_ADDR_MODE* (EMC register) is set to 1 by the primary boot loader as the function *ASYNC_ADDR_MOD* = 0 is not likely to be required. This bit can, however, be changed by the software at any time.

Other EMC or SPI settings required are made in line with the boot mode by the primary boot loader and in line with XHIF by the host.

The following aspect is specific to ARM9 processors for data storage from address 0x00...00 (interrupt vector table).

Address	Data item	Comment	Priority
0x00	Reset vector	Jump command following power on/WD/SW reset → <i>Stack pointer from address: 0x0800_FD00</i>	1 (highest)
0x04	Undefined Instruction	Jump command after execution of an illegal opcode → <i>Stack pointer from address: 0x0800_FC00</i>	6 (lowest)
0x08	Software interrupt	Jump command after SWI triggered → <i>Stack pointer from address: 0x0800_FA00</i>	6 (lowest)
0x0c	Opcode from illegal ad-	Jump command after illegal opcode address is read → <i>Stack pointer from address: 0x0800_FB00</i>	5

	dress		
0x10	Data from illegal address	Jump command after illegal data address is read → Stack pointer from address: 0x0800_FB00	2
0x14	reserved		n.a.
0x18	IRQ	Jump command after standard interrupt (IRQ) → Stack pointer from address: 0x0800_F900	4
0x1C	FIQ	Jump command after fast interrupt (FIQ) → Stack pointer from address: 0x0800_F800	3
0x20	Opcode	First possible opcode of the secondary boot loader	
...			

Note: In the boot ROM, all jump commands of the addresses highlighted in **yellow** point to the reset vector.

The following procedure is used to boot as rapidly and reliably as possible. After power on or after a reset of ERTEC 200P, the primary boot loader waits until HW initialization of the TCM is complete (`EDC_INIT_DONE.D_TCM926_INIT_DONE = '1'`).

The settings that do not depend on the boot modes are then made (see above). The secondary boot loader is then copied to the TCM in line with the boot mode.

The current boot ROM version is available in the last 16 bytes of the boot ROM. An ASCII string is saved here as the version identifier.

Important:

The last 2k of the D-TCM in block 0 are reserved for internal variables and stack pointers and must not be used for a secondary boot loader. No data/code may be loaded here either before the secondary boot loader has set the system to its requirements and "enabled" this area (see figure above on TCM mapping).

2.3.1.5.1.1 Boot Mode Settings

The EMC pins A[16:15], XOE_DRIVER, DTSR are used to set the boot mode. The pins are latched to the boot register `BOOT_REG` (see 2.3.10.9.2) during an active reset and return to their normal function once the reset is cleared. The ERTEC 200P and the XHIF interface are also configured with the EMC pins A[23:17].

BootPin(3) = A[16]	BootPin(2) = A[15]	BootPin(1) = XOE_DRIVER	BootPin(0) = DTXR	Boot mode	Booting...	BootPin(4) = XAV_BF
1	0	0	1	1	External NOR flash (16-bit) ¹⁾ , ASYNC_ADDR_MODE = 1 → Compile mode → Copy mode	1 0
1	0	1	0	2	External NOR flash (32-bit) ¹⁾ , ASYNC_ADDR_MODE = 1 → Compile mode → Copy mode	1 0
1	1	0	1	5	SPI master (RD Cmd: 0xE8)	X
1	1	1	0	6	SPI master (RD Cmd: 0x03)	X
1	1	1	1	7	XHIF (ext. host)	X

¹⁾ The secondary boot loader is run straight from the NOR flash and not from the TCM. The default mode,

- i.e. without external resistors, is a NOR flash with an access width of 32 bits and is selected with the internal pull circuit (highlighted in blue).
- 2) Code mode is not supported

Table 3: Boot mode adjustment

2.3.1.5.1.2 Handling after ARM926EJ-S SW Reset

If a soft reset has been carried out for the ARM926EJ-S core (*RES_SOFT_ARM926_CORE*), *RES_STAT_REG.SW_RES_ARM926* is set in the SCRB register and the primary boot loader is **not** processed. Instead, the program is continued from the address in the *RES_SOFT_RETURN_ADDR* register.

Following a HW reset, i.e. power on, ASIC (XRESET), ARM926 watchdog or SW reset for ERTEC 200P, *RES_STAT_REG.SW_RES_ARM926* is cleared and the primary boot loader is processed.

2.3.1.5.1.3 Memory Swapping

The reset vector of the ARM926 processor points to address 0x0000_0000. That is why the mirrored area of the boot ROM starts at address 0x0000_0000 after reset (power on, HW, SW or ARM926 watchdog reset). The boot ROM can also be addressed in its original address range (see 2.4.1).

At the end of the boot operation, the EMC SDRAM (max. of 64 MByte) or the EMC memory (max. of 1x 64 MByte, chip select Bank0) can be mapped to address 0x0000_0000h to allow the exception vector table for ARM926EJ-S to be created in address range 0x0000_0000 – 0x0000_001F. The original address ranges for boot ROM (in segment 4), EMC SDRAM (segment 2) and EMC memory (segment 3) are not affected by memory swapping.

Swapping is done by programming the *MEM_SWAP* register in the system control status register block SCRB (see 2.3.10.9.22). With reserved coding, there is no swapping in the *MEM_SWAP* register (or to be precise, no memory space is shown in segment 0). If these addresses are then accessed at the AHB, a QVZ (see 2.3.2.5.1) is triggered.

2.3.1.5.1.4 Startup Times

Depending on the boot mode, the ERTEC 200P requires one of the following startup times:

Boot mode	Booting...	t ₁	t ₂ (startup time)
1	External NOR flash (16-bit), ASYNC_ ADDR_MODE = 1 Compile mode	1.5 ms	1.5 ms
2	External NOR flash (32-bit), ASYNC_ ADDR_MODE = 1 Compile mode	1.5 ms	1.5 ms
5	SPI master (RD Cmd:	1.07 ms	5.8 ms (256 data bytes copied) → Each additional byte requires

	0xE8)		14.7 us
6	SPI master (RD Cmd: 0x03)	1.07 ms	5.8 ms (256 data bytes copied) → Each additional byte requires 14.7 us
7	XHIF (external host)	1.06 ms	Depends on - access time for the external master - the configuration of the XHIF - the destination address (typ. ARM926 TCM)

t_1 : Time between XRESET end and the first data being loaded to the 2nd boot loader.

Note: This time includes the setup time of 1235.5µs (see 4.8.5) and, depending on the boot, if applicable also the required setup times for the memory.

t_2 : Time until the first command of the 2nd boot loader is executed (startup time). This includes the time for

copying data. If the alternative block is used or several data bytes are copied, the time may be longer.

The startup times in ERTEC 200P are therefore calculated on the basis of: t_2

Table 4: Startup Times

2.3.1.5.2 Boot mode 1 ... 2 (booting with NOR flash) in compile mode

When you boot from the external NOR flash in compile mode, you can choose between 16-bit and 32-bit ROM. Booting is with EMC Peripheral Bank 0 (XCS_PER0)

After reset, the ARM926 starts the instruction fetches from the primary boot loader in the boot ROM that is shown at address 0x0. The EMC interface (memory bank signal CS0) is set to slow timing by the primary boot loader. The bus width is selected using the boot pins, which are evaluated by the boot software and which configure the relevant peripheral bank.

The next boot code command (secondary boot loader) is processed from the NOR flash. This requires a jump to 0x3000_0000 (NOR flash memory address: 0x0000_0000).

- If the user program is to be processed straight from the NOR flash, the ARM926 interrupt vector table must be stored from NOR flash memory address 0x0000_0000 (see 2.3.1.5.1). The secondary boot loader must implement re-mapping (*MEM_SWAP.SWAP* = "10", see 2.3.10.9.7) to show the vector list in the NOR flash at address 0x0000_0000.
- If the user program is to be processed from the TCM (I-TCM) or SDRAM, the secondary boot loader must copy the data from NOR flash address 0x0000_0000 to the required destination memory (D-TCM, SDRAM), configure the relevant D-TCM as I-TCM (*TCM926_MAP*, see 2.3.10.9.14) or remap the SDRAM to address 0x0000_0000 (*MEM_SWAP.SWAP* = "01", see 2.3.10.9.7).

The complete interrupt vector table is available in the boot ROM to prevent system dead-lock. When an interrupt or an exception occurs, a restart is always carried out if the vector

address of the boot ROM is mapped at address 0x0000_0000 (*MEM_SWAP.SWAP* = "00", see 2.3.10.9.7) and no I-TCM has been configured (if the I-TCM is configured, this is always shown at address 0x0000_0000 irrespective of *MEM_SWAP.SWAP*).

Note: The reset pin of the NOR flash must be connected to ERTEC 200P reset. There is a wait of at least 70 µs in the boot operation before the NOR flash is first accessed. This procedure is necessary as the previous history is not known and a write or delete job previously activated could be canceled by the reset. The flash block needs this time to switch to "Read data" status.

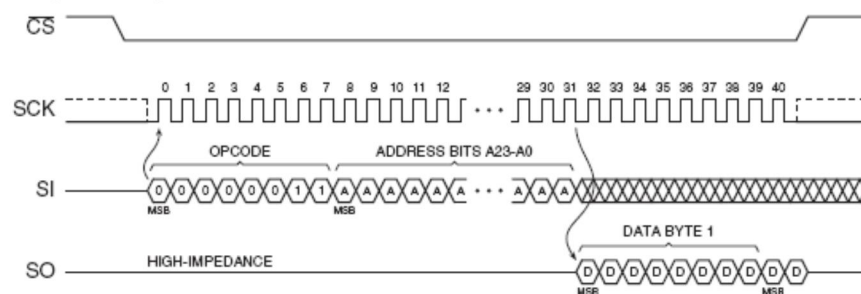
2.3.1.5.3 Boot Modes 5 and 6 (Booting with SPI Master)

The SPI interface of the ERTEC 200P is configured with the following parameters:

- SPI of the ERTEC 200P is the master (sets the SPI clock)
- 8 Data Bits
- Transmit/Receive with MSB first
- Idle Clock Line = 0
- Data is latched with a rising edge and output with a falling edge.
- The baud rate is programmed to a fixed 1 Mb/s
- The existing SPI module supports serial interfaces that are compatible with the Motorola, TI and NSC serial protocols.
- SPI memory chip select is addressed with GPIO31.

The SPI boot supports two types of SPI flash storage medium. Some of the SPI devices require the command 0xE8 for reading; others can be addressed with the read command 0x03. The subsequent SPI boot is, however, identical for the two read commands. A read command is structured as follows. In the standard sizes, 3 address bytes are used (A0...A23). The system always starts with the highest address byte.

Figure 7-2. Read Array – 03h Opcode



Following configuration of the SPI master, the boot code establishes which RD command can be used to address the SPI device. In device detection, the read command 0xE8 with the start address 0x0000 is first sent to the SPI device. If no identifier (*ID_OK* ≠ 0x5A) is found within 10 bytes, the read command 0xE8 is sent to the SPI device with start address 0x010000... (for alternative boot block in SPI device for data retention purposes, see 0). If no identifier (*ID_OK* ≠ 0x5A) is found within 10 bytes here either, the search for the device identifier is repeated with the command 0x03 (SPI-READ). If no identifier (*ID_OK* ≠ 0x5A) can be read back by the SPI device within 10 bytes for this command either (i.e. unprogrammed flash block), a watchdog is activated. When the watchdog expires (HW reset activation), device detection is run again.

If an identifier is read for one of the two read commands (*ID_OK* = 0x5A), the position of the identifier in the SPI data stream indicates how many address bytes (1, 2, 3 or 4 bytes)

are required for addressing the SPI device (distance between read command 1 and identifier = number of address bytes).

Once the identifier has been read, the information on the number of words to be read is read out (first the number of HighByte, then the number of LowByte). The following bytes are then interpreted as data and are stored in ascending order in the D-TCM from address 0x00...00 until the number of words to be copied is reached. The primary boot loader then configures the D-TCM as I-TCM (*TCM926_MAP*, see 2.3.10.9.14) and the I-TCM is shown at address 0x0000_0000. Processing then continues with a jump to address 0x0000_0044. The length of data to be copied is limited to the maximum size of the D-TCM.

Example:

Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	...
Send- ing	0x03 / 0xe8	0x0 0	0x0 0	0x0 0	0x0 0	0x0 0	0x0 0	0x0 0	0x0 0	0x0 0	0x0 0	0x0 0	0x0 0	...
	<i>RD</i>													
Re- ceipt	0xff	0xff	0xff	0xff	0x5 a	0x1 0	0x0 0	0x0 0	0x0 1	0xF F	0xF 0	0x0 0	0x0 0	...
					<i>ID</i>	<i>No. of high</i>	<i>No. of low</i>	<i>Da- ta 0</i>	<i>Da- ta 1</i>	<i>Da- ta 2</i>	<i>Da- ta 3</i>	<i>Da- ta 4</i>	<i>Da- ta 5</i>	...

In the example above, 0x1000 words are read from the medium. A maximum of 256 KByte (0xFFFF) can be read. A max. of 64 KByte (0x3FFF) is possible with the current D-TCM.

Memory image in the D-TCM

TCM+0	0x00	Data 0	Word 0
TCM+1	0x01	Data 1	
TCM+2	0xFF	Data 2	
TCM+3	0xF0	Data 3	
TCM+4	0x00	Data 4	Word 1
TCM+5	0x00	Data 5	
TCM+6	0x01	Data 6	
TCM+7	0x05	Data 7	
TCM+8	0x00	Data 8	
TCM+0x104			

A reload function cannot be provided in the primary boot loader for reading from an SPI flash memory as the number of addresses to be transferred is not known, and universal access therefore cannot be set. This must be done using the secondary boot loader.

2.3.1.5.4 Boot Mode 7 (Booting with External Host)

After reset, the ARM926 starts the instruction fetches from the primary boot loader. The basic XHIF configurations (16/32-bit bus, XHIF_ACC_MODE, XHIF_POL_RDY) are latched to the *CONFIG_REG* register after reset (see 2.3.10.9.3) and wired directly to the XHIF module. The parameter settings (for example page sizes, ...) at XHIF are made by the external host in the XHIF control registers. The ERTEC 200P signals readiness for firmware upload to the host with a software handshake (boot loader functionality, semaphore) (ready for firmware upload). At address D-TCM+0x40 (address 0x0803_0040 in block 3 of the D-TCM), the OK identifier = 0x6879_9786 (software handshake) and the first OP code of the secondary boot loader is processed from address D-TCM+0x44 (address 0x0803_0044 in block 3 of the D-TCM).

Once reset is inactive, the primary boot loader also sets the XHIF interface in line with the ConfigPins in the XHIF_Config register with the selected XHIF_CPU_Width, XHIF_POL_RDY and XHIF_ACC_Mode and enables XHIF_PAGE0 with a width of 1M for access to the D-TCM. Other XHIF pages are not changed. The primary boot loader sets the address D-TCM+0x40 (address 0x0803_0040 in block 3 of the D-TCM) to 0x0000_0000 (not OK) and waits until OK identifier = 0x6879_9786 is entered by the external host.

SW code download from the XHIF must be actively executed by the external host processor. The boot code can be transferred from the XHIF host to the D-TCM (block 3) of the ERTEC 200P. At the end of data transfer, the XHIF host sets identifier 0x6879_9786 at address D-TCM+0x40 (semaphore address). Once the semaphore has been set, the ERTEC 200P maps the D-TCM to I-TCM (*TCM926_MAP*, see 2.3.10.9.14) and the secondary boot loader is started at address 0x0000_0044 (I-TCM). Access to the I-TCM over XHIF is not possible; only the D-TCM can be addressed.

Detection of ERTEC 200P readiness (for XHIF boot) by the external host

For an external host processor to recognize whether the XHIF interface is ready, the following procedure must be observed:

The external host polls the XHIF range register of bank 0 until the value 0x0010_0000 (1 MByte range) appears. The reset value of the XHIF range register is 0x0000_0000. Access to this register is possible even during an active reset. If the external host reads the correct XHIF range register value (0x0010_0000), it then reads the offset for Page0, which must then contain the value 0x0800_0000. The host then reads the semaphore at D-TCM+0x40 (address 0x0803_0040 in block 3 of the D-TCM) until it returns the value 0x0000_0000.

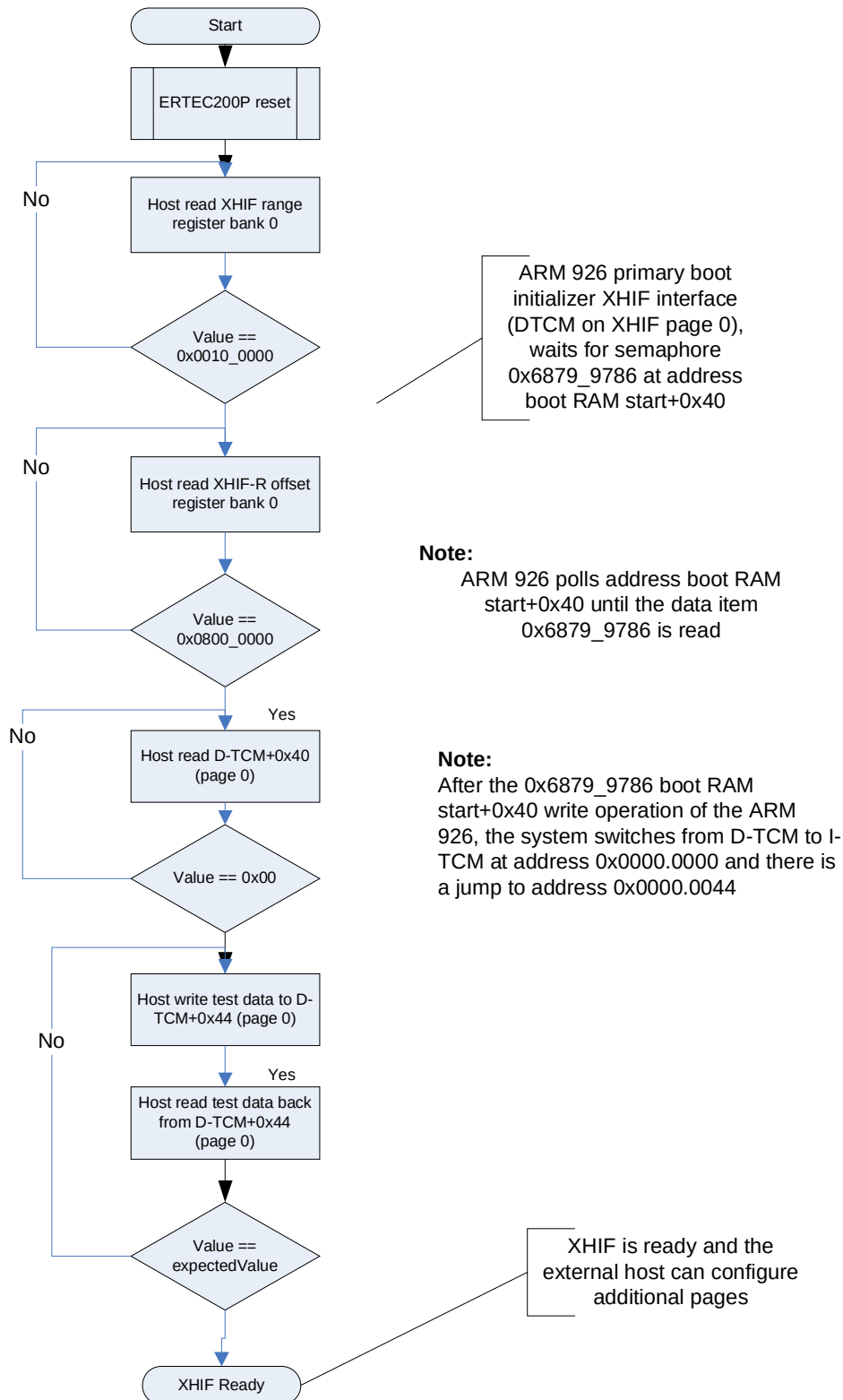
To ensure that access over XHIF to the D-TCM (block 3) is possible, the external host writes the address 0x0803_0044 of the D-TCM and reads back the value. If the value written has been read back correctly, transfer of the secondary boot loader can start. At the end of data transfer, the XHIF host sets identifier 0x6879_9786 at address D-TCM+0x40 (address 0x0803_0040 in block 3 of the D-TCM) (semaphore address).

Note for module development

Please see the note in Spec XHIF_V1.0c.pdf on wiring XHIF-RDY polarity with pullup/pulldown.

IMPORTANT:

The interrupt vector table (see 2.3.1.5.1) is to be stored at the addresses before the semaphore (i.e. from addresses 0x0803_0000)



Flow chart for the external host for establishing XHIF interface enable

2.3.1.6 Secondary Boot Loader

After leaving the boot ROM (processing of the primary boot loader), the ERTEC 200P is configured as follows:

1. ARM mode is SYS
2. MMU is not changed; reset value.
3. Watchdog is set to 251 ms and active.
4. SDRAM is not initialized.
5. Stack pointer (SP) points to 0x0800_fd00.
6. 64k I-TCM from address 0x0000_0000 and 192k D-TCM
(*TCM926_MAP.PARTITION_TCM926* = "001", **not** with NOR flash boot)
7. Shift mode active at EMC interface (*EXTENDED_CONFIG.ASYNC_ADDR_MODE* = 1)

2.3.1.6.1 Memory Swapping by the Secondary Boot Loader

As the size of the I/D-TCM can be set variably in the ARM926, it is left to the secondary boot loader to make the best settings for its needs.

The reset vector of the ARM926 processor points to address 0x0000_0000. That is why the boot ROM starts at address 0x0000_0000 after reset (power on, HW, SW or watchdog reset). The boot ROM can also be addressed in its mirrored area (0x4000_8000)

It is left to the secondary boot loader to change the TCM mapping. To allow the secondary boot loader to be run, the primary boot loader switches the first page (64 KByte) of the D-TCM to I-TCM (*TCM926_MAP*, see 2.3.10.9.14). If additional I-TCM is required, the secondary boot loader must switch again.

Swapping is done by programming the *MEM_SWAP.SWAP* register (see 2.3.10.9.7). This procedure ensures that the exception vectors are always assigned and prevents a deadlock.

2.3.2 AMBA (Internal Bus System)

The ERTEC 200P has 2 internal bus systems. One is a high-performance communication bus (multi-layer AHB) and one is a peripheral bus (APB).

All 6 masters (ARM926-I, ARM926-D, PN-IP-M1, PN-IP-M2, , GDMA controller and host interface) and the slaves (D-TCM926, interrupt controller (ARM-ICU), GDMA register/memory, external memory controller (EMC), APB bridge, , peripheral interface (2 connections), PN-IP) are connected at the multi-layer AHB, which has a high data transfer rate and high bus availability.

Over an AHB/APB bridge, the masters can access the rest of the peripherals which are connected at the low-performance peripheral bus (APB). The AHB/APB bridge is the only master at the peripheral bus.

2.3.2.1 Characteristics of the Bus System in ERTEC 200P

The table below sets out the characteristics of the internal bus system:

Requirements	
Processor frequency	125/250 MHz
AHB frequency	125 MHz
AHB width	32 bit
AHB bus access width	8/16/32 bit
APB bus frequency	125 MHz
APB bus width	32 bit
APB bus access width	8/16/32 bit

Table 5: Internal bus system, parameters of internal buses

2.3.2.2 Endianness

The ERTEC 200P operates with little endianness. **No** endian swappers are available at the AHB interfaces.

2.3.2.3 AHB Subsystem (Multi-Layer AHB)

The multi-layer AHB bus is a 32-bit bus with multi-master capability. It runs at 125 MHz and has the functionality of the ARM-AHB Lite bus (see document /11/, see chapter 7.2) connecting multiple AHB segments in a multi-layer AHB allows 7 AHB masters to access different slaves simultaneously.

An arbiter decides which master is to be assigned a given slave when multiple masters access the slave simultaneously.

A retry functionality is not implemented at the AHB in the light of the multi-layer architecture (AHB-Lite). Split functionality is not supported either (AHB-Lite).

Alongside Okay and Error, Split and Retry are also ways to acknowledge an AHB transfer from the slave. Retry response indicates that the transfer is not yet complete and that the master should repeat the transfer. Split response indicates that the transfer has not yet been successfully completed. The master must release the bus and cannot be re-assigned the bus until the slave indicates to the bus arbiter that it is able to complete an AHB transfer. The master must repeat the transfer for this purpose.

The lock function can be enabled for the ARM master (ARM926-D) in the SCRB register M_LOCK_CTRL. Default setting: lock functionality is disabled. The ARM926-I master cannot generate a lock.

Definition of AHB-Lite bus

AHB-Lite: A subset of the full AMBA AHB protocol specification. It provides all of the basic functions required by the majority of AMBA AHB slave and master designs, particularly when used with a multi-layer AMBA interconnect. In most cases, the extra facilities provided by a full AMBA AHB interface are implemented more efficiently by using an AMBA AXI protocol interface.

More detailed documentation is available at: /11/ AHB_Lite_Protocol.

The AHB slave modules allow the following types of access:

AHB module (slave)	Access permitted	Comment
PN-IP *)	8/16/32-bit	Data bus: 32-bit
Peripheral interface	8/16/32-bit	Data bus: 32-bit
EMC memory	8/16/32-bit	Data bus: 32-bit
EMC register	8/16/32-bit	Data bus: 32-bit
GDMA (RAM + Register)	Only 32-bit	Data bus: 32-bit
ARM-ICU	Only 32-bit	Data bus: 32-bit
ARM926 - TCM	8/16/32-bit	Data bus: 32-bit
ARM926 - ETB	8/16/32-bit	Data bus: 32-bit

*) Handling of access over AHB within the modules varies. Please see the module specifications/section on the relevant module.

2.3.2.3.1 AHB Arbiters

Each of the AHB arbiters uses the same arbitration procedure. The default setting is round robin. A fixed priority assignment for the AHB masters (see Table 6: Fixed priority assignment (no default)) could be used as an alternative arbitration algorithm by programming the ARB_MODE bit in the SCRB register M_LOCK_CTRL (see 2.3.10.9.22). This option should not, however, be implemented in the light of the dynamic processes at the multi-layer AHB.

Using round robin as the arbitration procedure prevents the AHB masters from blocking each other at the multi-layer AHB for a prolonged period of time (cf. 2.3.2.3.2)

PRIORITIES	MASTER	NOTES
7	Host interface	Highest priority
6	GDMA	
5	PN-IP-M1	5
4	----	
3	-----	
2	ARM926-D	
1	ARM926-I	Lowest priority

Table 6: Fixed priority assignment (no default)

2.3.2.3.2 AHB Multi-layer Configuration

The masters in the multi-layer AHB are not connected to all slaves. The table below shows which AHB master is connected to which AHB slave.

	S1	S2	S3	S5	S6	S7	S9	S10	S12
--	----	----	----	----	----	----	----	-----	-----

⁵ No fixed priority is assigned to the PN-IP-M2 master as the connection to the PER-PN-S10 is point-to-point. This prevents a conflict from ever occurring at PER-PN-S10.

	D-TCM926	ETB	ARM-ICU	GDM A	EMC	AHB/APB bridge	PerIF host	PerIF-PN	PN-IP
AHB master									
ARM926-D ^{*)}		x	x	x	x	x	x		x
ARM926-I ^{*)}					x	x			
GDMA	x	x		x	x	x	x		
Host interface	x			x	x	x	x		x
PN-IP M1	x				x				
PN-IP M2								x	

^{*)} Burst access by the AHB master can be interrupted with the AHB burst breaker.

Table 7: AHB Master-slave coupling

Important:

To avoid prolonged blocks during slave access, the AHB masters must not arbitrate a slave for too long. Arbitration could for example take too long if the ARM926-D master had too many transactions with the EMC slave in its write buffer. Another master (e.g. host interface or PN-IP-M1) would then be blocked from accessing the EMC slave during this time and would have to wait in "wait" until the ARM926-D master released the EMC slave.

The ARM926-D and ARM926-I masters have burst breakers fitted at the AHB access point. A burst breaker generates an idle phase for a clock after a configured number of consecutive address phases (see 2.3.2.3.3). A different AHB master can then be assigned the AHB slave during this idle phase. This is possible because re-arbitration at an AHB slave can only be carried out if a master releases that AHB slave (idle phase).

The PN-IP-M1/2 and GDMA AHB masters cannot occupy a slave for longer than one burst sequence of 8 data transfers. At the end of this sequence, the AHB master then always implements an idle phase. A different AHB master can therefore always be assigned the given slave at the end of an 8-beat burst sequence. This limit is fixed for the PN-IP-M1/2 AHB master, but can be set per DMA job for the GDMA AHB master. The host interface AHB master can only implement single transactions at the AHB.

2.3.2.3.3 AHB Burst Breaker

To improve the real-time capabilities of ERTEC 200P three Burst Breakers are used to prevent ARM926-D and ARM926-I to block resources by consecutive burst transfers without IDLE cycles in between, which are necessary for re-arbitration of AHB masters. The Burst Breaker counts the address phases of following ARM926-D/ARM926-I AHB transfers:

- SINGLE
- INCR unspecified length

- WRAP4
- INCR4
- WRAP8
- INCR8
- WRAP16
- INCR16
- Address phase of transfer type BUSY

If the number of consecutive address phases at ARM926-D AHB port / ARM926-I AHB port / AHB port without an IDLE cycle in between exceeds NR_ADDR_ARM926-D / NR_ADDR_ARM926-I / in the AHB_BURSTBREAKER Register (see chap. 2.3.10.9.22, configured by SW), the Burst Breaker will insert an IDLE cycle immediately if the current transfer is SINGLE or INCR unspecified length. In the later case the INCR will be split. If the current transfer is any kind of burst with specified length the IDLE insertion will be done at the end of the ongoing burst transfer (to not interrupt any cache transfers).

NR_ADDR_ARM926-D/NR_ADDR_ARM926-I set to '0' means Burst Breaker at ARM926-D AHB port/ARM926-I AHB port port is set to transparent mode. Any transfer is passed through as is; no IDLE insertion.

If $0 < \text{NR_ADDR_ARM926-D/NR_ADDR_ARM926-I} < 255$ IDLE insertion is activated for the threshold defined by NR_ADDR_ARM926-D/ NR_ADDR_ARM926-I. Changes of NR_ADDR_ARM926-D/ NR_ADDR_ARM926-I are updated at the next IDLE cycle.

Recommended is a value of 8 (max. 8er-Burst) for NR_ADDR_ARM926-D/ NR_ADDR_ARM926-I.

2.3.2.4 APB Subsystem (Peripheral Bus)

Only the AHB/APB bridge has a master function at the APB. All peripheral blocks of the APB are operated as slaves. All peripheral function blocks with low performance requirements (e.g. not burst-compatible) are connected to the APB bus. The AHB/APB bridge has a single-stage write buffer.

The following blocks are connected at the ERTEC 200P APB bus:

- SCRB
- Boot ROM
- Timer 0-5
- F-timer
- Watchdog
- 4x UART
- 2x SPI
- I²C
- GPIO
- HostIF (APB interface)
- PerIF (APB interface)
- I-filter

The APB bus is a 32-bit wide bus. The bus frequency is 125 MHz. The APB bus in the ERTEC 200P has a wider functionality than the ARM-APB bus (see document /11/, 5.5). Unlike the standard APB bus, it offers the option of variable timing (wait states) and byte

enables. Wait state and byte enable capability is achieved by introducing sideband signals. This allows byte-by-byte access to peripherals.
The APB modules allow the following types of access:

APB module	Access permitted	Comment
SCRB	Only 32-bit *)	Data bus: 32-bit
Boot ROM	8/16/32-bit	Data bus: 32-bit
Host interface (XHIF)	Only 32-bit	Data bus: 32-bit
UART	16/32-bit	Data bus: 16-bit, i.e. the top 16 bits are ignored
F-timer	Only 32-bit	Data bus: 32-bit
Timer 0-5	Only 32-bit	Data bus: 32-bit
Watchdog	Only 32-bit	Data bus: 32-bit
I2C	8/16/32-bit	Data bus: 8-bit, i.e. the top 24 bits are ignored
SPI	16/32-bit	Data bus: 16-bit, i.e. the top 16 bits are ignored
GPIO	Only 32-bit	Data bus: 32-bit
I-filter	8/16/32-bit	Data bus: 32-bit
Peripheral interface	8/16/32-bit	Data bus: 32-bit
SCRB	Only 32-bit	Data bus: 32-bit

*) Handling of access over APB within the modules varies. Please see the module specifications/section on the relevant module.

Note:

Burst access (INCR, INCR4, INCR8, INCR16, WRAP, WRAP4, WRAP8, WRAP16) is allowed; access is split into single instances of access at the APB by the AHB2APB bridge.

Important:

Write access to APB modules in the AHB2APB bridge is posted. Additional clocks are also required at the APB bus until write access is "effective" in the APB register/RAM. Subsequent read/write access via **another** data path (e.g. AHB) can therefore overtake the previous APB write access.

If this is to be prevented, the "effectiveness" of the APB write access must be ensured in the relevant APB register/RAM with a subsequent instance of APB read access to the same APB address. The following write/read access may only be implemented after that APB read access.

The AHB/APB bridge is implemented between the two buses. The AHB/APB bridge has a single-stage write buffer.

The function of the AHB bridge is detailed in document /40/ AHB_APB_BRIDGE_V1.0-pdf.

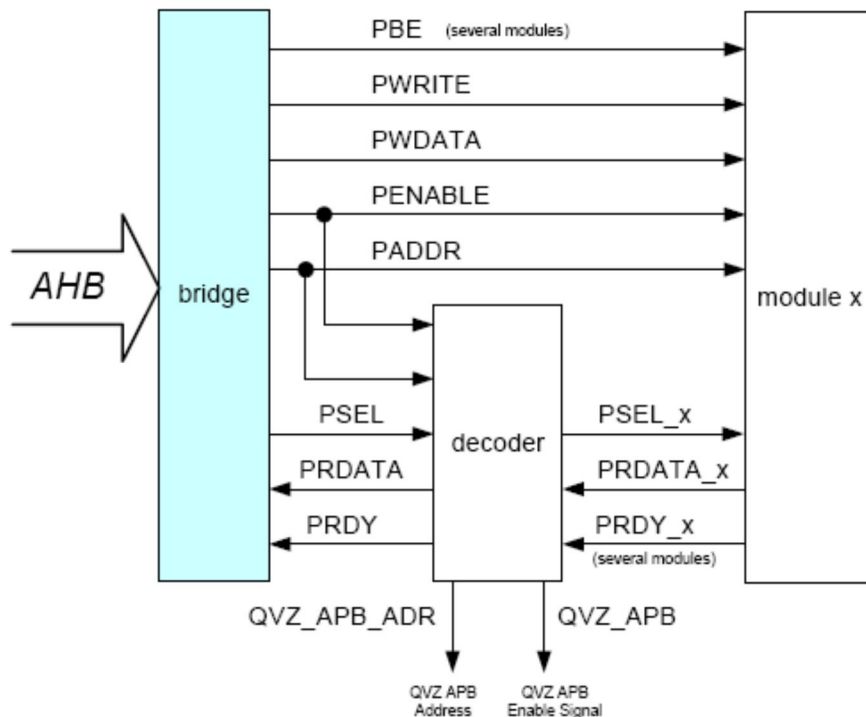


Figure 7: AHB/APB bridge, block diagram

The key features of the bridge are as follows:

- On the AHB side the AHB_APB_Bridge fully supports the AMBA 2.0 AHB protocol. It only uses the AHB-Lite subset of this protocol. The bridge is connected on the AHB-Layer as a slave.
- On the APB side the AHB_APB_Bridge supports the AMBA 2.0 APB protocol. Accesses on the AHB side are one by one translated into APB accesses, write accesses are handled as posted, i. e. AHB_APB_Bridge completes the access on the AHB side as soon as it is able to start it on the APB side.
- The AHB_APB_Bridge is not able to produce back-to-back accesses on the APB side. In difference to the APB specification it uses an IDLE state after every APB access even when there already is an AHB access waiting to be translated. AHB bursts are treated as sequences of single accesses.
- Expanding the AMBA 2.0 - APB specification, the AHB_APB_Bridge provides the option of using a variable timing of APB accesses. By introducing a ready signal, wait states can be implemented. The standard AMBA-APB bus is expanded by the PRDY sideband signal (see Figure 8: APB bridge, timing with sideband signals PRDY and PBE).
- Expanding the AMBA 2.0 - APB specification, the AHB_APB_Bridge provides the option of using "byte enables" on APB bus. Therefore the standard AMBA-APB bus is expanded by the sideband signals PBExx which exist in parallel to PADDR, i.e. PADDR and PBE timing are identical (see Figure 8: APB bridge, timing with sideband

signals PRDY and PBE).

- All signals are clocked. There are no snake paths from AHB side to APB side and vice versa.
- AHB frequency has to be the same as the APB frequency.
- The AHB data bus width is the same as the APB data bus width (32 bits).
- The AHB_APB_Bridge supports only one slave. Address decoding and multiplexing for several slaves are carried out outside the bridge, e.g. APB decoder.
- The AHB_APB_Bridge is implemented fully synchronously. The rising clock edge is the active one.
- The reset is designed asynchronously. The reset is not synchronized in the AHB_APB_Bridge therefore synchronous reset has to be applied to meet recovery/removal time requirements.

Bridge timing with the additional signals PBE and PRDY:

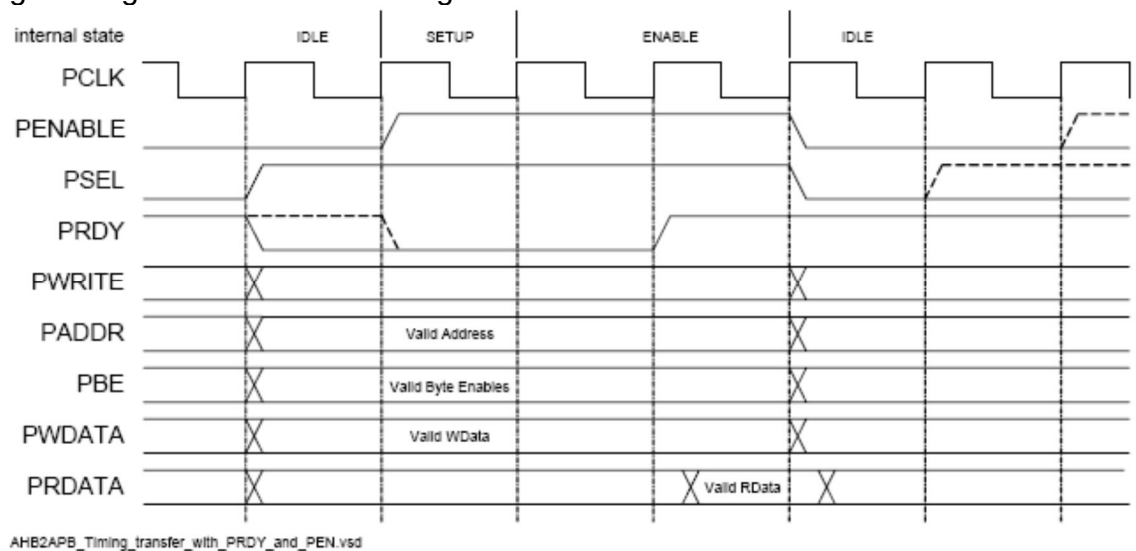


Figure 8: APB bridge, timing

2.3.2.4.1 APB Decoder

The decoder connected to the bridge for addresses and the read data bus enables the specific PSEL signals to be generated for the individual blocks of the APB bus. It is also possible to read the various RD data buses of the individual blocks over the decoder on the AHB bus.

If a PSEL is pending at the APB bus but the address is not within the slave addresses, an acknowledgement delay (QVZ enable) signal is generated and the address (QVZ address) is output. This error is saved externally in the ACCESS error registers of the SCRB (register SCRB_ACC_ERR_ALL).

The bridge is a slave at the AHB side and the master at the APB side. No error-response is generated to the AHB side in the case of access with errors at the APB side (posted behavior). As already indicated above, the instances of erroneous access (address of the access) are saved in the AHB_APB_ACC_ERR register in the SCRB and an interrupt is triggered. Erroneous write access is ignored. Erroneous read access returns the value 0x0000_0000.

2.3.2.5 Address Range and Acknowledgement Delay Monitoring

A range of monitoring mechanisms are implemented in the ERTEC 200P for detecting incorrect addressing and acknowledgement delays (ready delays).

2.3.2.5.1 Monitoring at the AHB Side

Each AHB master (ARM926-D, ARM926-I, PN-IP, GDMA, host IF) is assigned its own address range monitoring. If an AHB master accesses an address range that is not in use, the access is acknowledged with an error response and an interrupt IRQ51 (see 2.3.2.14) is triggered in the ARM interrupt controller. The address of the erroneous access is also saved in SCRB register QVZ_AHB_ADR and the corresponding access type (AHB control signals: Read/Write, HBURST, HSIZE) in SCRB register QVZ_AHB_CTRL (see 2.3.10.9.22).

The master that has triggered the access violation can be read in SCRB register QVZ_AHB_M.

If RD access at the host IF that has been triggered by parallel (XHIF) or serial (SPI) access is acknowledged with an AHB error response, an AHB error IRQ is also triggered at the event unit (peripheral interface). This entry activates the host interrupt XHIF_XIRQ.

The 3 diagnostic registers QVZ_AHB_ADR/CTRL/M are blocked for subsequent access violations until the register QVZ_AHB_CTRL has been read. Reading the QVZ_AHB_CTRL register only, however, unlocks it; the content of the QVZ_AHB_ADR/_CTRL/_M registers remains unchanged. Only a subsequent QVZ updates the QVZ_AHB_ADR/_CTRL/_M registers.

If multiple AHB masters cause an access violation at the same time (within one AHB clock cycle), only the violation by the highest priority AHB master (prioritization as detailed in 2.3.2.3.1) is displayed.

Important:

If an AHB error response is triggered by the ARM926, this results in a Data/Opcode illegal address interrupt in the ARM926. This interrupt is processed as higher priority by the ARM926 core.

*Address gaps in the APB address range are **not** visible at the AHB side; the monitoring mechanisms in 2.3.2.5.2 apply.*

2.3.2.5.2 Monitoring at the APB Side

The APB address range is monitored at the APB side. In the event of incorrect addressing in the APB address range, access to the APB and AHB side is completed with an OKAY response as the APB bus is not able to signal the response type.

An IRQ52 interrupt (see 2.3.2.14) is triggered in the ARM interrupt controller. The address of the erroneous access is saved in SCRB register QVZ_APB_ADR (see 2.3.10.9.22).

The diagnostic register QVZ_APB_ADR is blocked for subsequent access violations until it has been read. The master that has triggered the access violation can be read in SCRB register QVZ_AHB_M.

2.3.2.5.3 Monitoring at the EMC

The external ready signal XRDY_PER is monitored in the EMC. If one of the 4 external memory areas selected with the output pins XCS_PER(3:0) is addressed, the memory controller waits for the input signal XRDY_PER (provided ready control is activated in the corresponding configuration register ASYNC_BANK_x_CONFIG; see 2.3.5.8). Acknowledgement delay monitoring is activated in the EXTENDED_CONFIG register (see 2.3.5.8) and generates a ready signal internally for the memory controller and IRQ53 (see 2.3.2.14) in the ARM interrupt controller after a max. of $(1048575 + 1) \times 16$ AHB clock cycles. The duration of monitoring is set in the ASYNC_WAIT_CYCLE_CONFIG register (see 2.3.5.8) (for further information, see 2.3.5.4.5).

The address of the erroneous access is also saved in the EMC register AT_ADDR.

2.3.2.5.4 Monitoring in the Individual Modules

Implementation within the modules (IPs)

The following modules:

- PN-IP (Profinet-IP)
- PER-IF (peripheral interface)
- I-filter
- Host IF (host interface)
- SCRB

have monitoring for unassigned addresses. Access over the **AHB** and **APB** that addresses an undefined address range within a module (IP):

- returns read data 0x0000 if READ
- is ignored if WRITE.

Access is always completed with READY (AHB OK response) at the **APB**. At the **AHB**, RD access results in an AHB error response; write access is always acknowledged with an AHB OK response because of the posted behavior of the AHB slave interface.

The first instance of erroneous access is also saved in the module register and output as a separate output signal with the event `access_err_irq_o`. As long as the module register `Access_Error` has not been released for subsequent erroneous access by the SW (`ERR_LOCK = '0'`), all subsequent instances of access are processed without being saved in the register and without interrupts being output.

An entry in the module register `Access_Error` is saved when `ERR_LOCK = '0'` in the following cases:

- Write access to read-only registers
- Write and read access to address gaps in register and RAM address ranges

- Byte and halfword write access to addresses that can be written word by word

As an example, here is the SCRB access error register:

Register:		SCRB_ACC_ERR				Address:		38h			
Bits:		31dt0		Reset value:		8000000h		Attributes:		r(h)	(w)
Description:			Access Error Register								
Bit	Identifier			Reset	Attr.		Function / Description				
15dt0	ACCESS_ADDR			0000h	rh		Address that caused the last access error.				
27	ACCESS_BUS_TYPE			1h	r		Businterface that caused the last address error. 1 = APB-Interface.				
29dt28	ACCESS_SIZE			0h	rh		Shows the transfers size during the last access error. 00 = 8 bit transfer 01 = 16 bit transfer 10 = 32 bit transfer 11 = illegal combination of APB byte enables				
30	ACCESS_TYPE			0h	rh		Shows the transfer type during the last access error. 0 = read transfer 1 = write transfer				
31	ERR_LOCK			0h	rh		w	The hardware sets this bit during an access error. The bit remains set as long as the software does not write a logic 0 into this bit. As long as the bit is set, the content of the register remains unchanged even during further access errors. Read: 1 = Access Error			

Table 8: Access Error Register (from SCRB)

Implementation in ERTEC 200P top level:

The `access_err_irq_o` events of the individual modules are high-active for a **125 MHz system clock** in the top level and are all set out in the `MODUL_ACCESS_ERR` register (see 2.3.10.9.5).

An entry in `MODUL_ACCESS_ERR` triggers the ARM926 interrupt 'Modul_Access_Error' (see 2.3.2.14). Following a `MODUL_ACCESS_ERROR` interrupt, the SW must read the `MODUL_ACCESS_ERR` register to identify the module affected. The SW must then read the detailed data for the erroneous access from the access error register in the relevant

module and clear the register (ERR_LOCK = '0'). The SW must subsequently reset the MODUL_ACCESS_ERROR register in the SCRB by writing the value 0x0.

In the event of read access to unassigned addresses in AHB modules, an AHB error response is also generated on the AHB (PN-IP, PER-IF). This triggers an exception in ARM926EJ-S.ARM-ICU (ARM926 Interrupt Controller)

2.3.2.6 Overview

The Interrupt Controller Unit (ICU), see chapter 7.2, supports both IRQ (Interrupt Request) and FIQ (Fast Interrupt Request) interrupt levels which are named in the following IRQ subblock and FIQ subblock. This overview chapter describes the functionality using the IRQ subblock. The following sections discuss the differences between the IRQ subblock and the FIQ subblock.

The Interrupt Controller Unit both in the IRQ subblock and in the FIQ subblock consists of three functional units described in detail in the following sections.

The following overview provides an impression of the operation of the ICU:

1. The specific preprocessing for each interrupt is performed in the first functional unit of the IRQ subblock. This preprocessing includes:
 - Enable and disable of interrupts
 - Settings for trigger modes
 - Processing of software interrupts, etc.

When an interrupt event occurs at the ICU input, this preprocessing is performed initially before the interrupt is entered in the Interrupt Request Register (IRREG, can be read by the software).

2. The second functional unit is responsible for decoding the priorities (priority resolving) for each interrupt. It is determined in each clock:
 - Whether a pending interrupt will be forwarded to the third functional unit
 - If several interrupts are pending, which of these have the highest priority.
3. This interrupt (or its priority) is compared in the third functional unit (postprocessing, ICU-ISR) with any interrupt currently being processed by the software (using the priority).
 - If the pending interrupt is valid, namely has the appropriate priority, it will be reported to the CPU.
 - The CPU must confirm each interrupt with an access to the Acknowledge Register (IRQACK). This causes the appropriate bit for the interrupt number to be set in the In-Service-Register (ISREG) and cleared in the IRREG.
 - After completion of the Interrupt Service Routine by the CPU, the ICU requires an End-of-Interrupt command (EOI, a write access of the CPU to the register of the same name) to inform the ICU about the end of the interrupt processing and to initiate the clearing of the corresponding bits from the ISREG.

2.3.2.7 Interface

The Interrupt Controller Unit (ICU) runs fully synchronous with an operating clock of 125 MHz and is fully synchronous. The active edge is the rising clock edge. The ICU receives a reset that is not synchronized. The ICU can correctly process interrupts asynchronous to the operating clock provided each signal level lasts **at least two clock periods** or synchronous to the operating clock provided each signal level lasts **at least one clock**

periods. Incoming interrupts are synchronized over two clock cycles to the operating clock of the ICU.

It is a requirement that interrupt signals from all circuitry parts with higher clock frequency outside the ICU are lengthened to the appropriate minimum pulse time.

The ICU supplies a separate output signal to serve both interrupt levels of the CPU (INTB and INTA). The interrupt processing within the ICU is performed in separate subblocks for INTB and INTA. **The output signals for the ICU are level-triggered and "active high".** The output signals become inactive again when the interrupt from the CPU is confirmed or when the Interrupt Request is masked by the CPU before it is acknowledged.

The INTA subblock supports NUM_OF_INT (NUM_OF_INT = 96) interrupt inputs; the INTB subblock supports NUM_OF_FINT (NUM_OF_INT = 8) interrupt inputs.

Each interrupt source of the INTA subblock can be placed on each interrupt input of the INTB subblock, where the selected interrupt sources also continue to remain active in the INTA subblock. The "INTB0" is the exception, because no INTA can be placed on it. For the configuration of the assignment by the software, the numbers of the selected INTA inputs are entered in the INTB_SEL_1 – INTB_SEL_<NUM_OF_FINT-1> registers. Although the INTB_SEL_0 register can be read and written, it is not further processed by the hardware.

Note that both INTAs and INTBs are edge-triggered as initial setting. If level triggering is set, the interrupt at the source must be removed, otherwise it will issue a new interrupt (INTA or INTB) to the CPU after the acknowledge. **Note further that only high active level (in level triggering mode) is supported by ICU whereas in edge triggering mode both edges are supported.** INTA and INTB must always be set to the same triggering. Because this is not checked by the ICU, it must be ensured by the software. In addition, the software should normally mask the INTA input in the INTA subblock used as INTB, because otherwise both an INTA and an INTB will be signaled to the CPU when the interrupt occurs.

The interrupt with number "0" in the INTA subblock is not stimulated by any interrupt source. The Fast Interrupt with number "0" cannot be assigned by any arbitrary INTA input but is clamped statically in the ICU itself to "inactive".

The interrupt with number "0" may not be used either in the INTA subblock or in the INTB subblock. "INTA0" **outside** the ICU is clamped to the inactive level; the "INTB0" **inside** the ICU is clamped to the inactive level. This means that a differentiation between the default vector and the interrupt vector with number "0" is no longer necessary which simplifies the processing of the default vector by the software.

The ICU supports the AHB-Lite protocol. The ICU has a 32-bit wide data bus. Word accesses only are supported.

2.3.2.8 Interrupt acquisition

An interrupt to an input signal can be recognized both on an edge and on the level of the input signal. This trigger mode can be separated for each interrupt input by setting the appropriate bit in the TRIGREG register. **Level-triggered interrupts are acquired as "active high".** The initial setting for each interrupt is the detection of the positive edge.

In edge-triggered mode, each level on the input for synchronous signals must be stable for at least one clock, for asynchronous signals for at least two clocks.

- In edge-triggered mode, each level on the input for synchronous signals must be stable for at least one clock, for asynchronous signals for at least two clocks.
- In level-triggered mode, an active level must be present until the CPU confirms this interrupt with an acknowledge.
 - When in level-triggered mode an interrupt is still active at the end of the processing (acknowledge and EOI), a further interrupt will be output to the CPU as soon as the corresponding bit in the ISREG has been cleared by the EOI command and no higher-priority interrupt is present.
 - When in level-triggered mode an interrupt is removed by the source before the CPU can confirm it with an acknowledge, however after the removal of the interrupt the CPU nevertheless performs an acknowledge, the ICU returns the default vector (null vector) as interrupt vector.

The edge (rising/falling) for which the ICU detects an interrupt on an input set to edge triggering can be set separately for each interrupt input by setting the appropriate bit in the EDGEREG register.

An interrupt detected at the input will be entered in the Interrupt-Request register. Provided it is not masked, after being entered, it will be led to the priority decoder. The entry will be cleared when the corresponding interrupt is confirmed by the CPU with an acknowledge and so will be entered in the In-Service register.

Each bit in the Interrupt Request register can be cleared using an appropriate command issued in the IRCLVEC software register passing the number of the bit to be cleared. The bit is cleared when a write access to IRCLVEC is detected.

Each interrupt can also be triggered other than using the appropriate ICU input signal by setting the appropriate bit in the Software Interrupt registers (SWIRREG/INTB_SWIRREG). After setting the bit, no minimum time must be observed for the renewed clearing of the bit.

To allow several interrupts to be triggered simultaneously, the SWIRREG consists of several 32-bit registers. A software interrupt is forwarded directly to the IRREG and then handled like every other interrupt. The same procedure applies to the fast interrupts triggered in the register INTB_SWIRREG. The following section discusses the software interrupts depending on the trigger mode:

- In level-triggered mode, the Interrupt Request register (INTA_IRREG or INTB_IRREG) is affected directly by the INTA_SWIRREG or INTB_SWIRREG. Under the prerequisite that no interrupt is present at the ICU input, a bit in the Software Interrupt register represents the corresponding bit in the Interrupt Request register.
- In edge-triggered mode, a bit is set in the Interrupt Request register when the appropriate bit in the Software Interrupt register indicates a rising edge.

Warning:

There is a difference for the processing of software interrupts compared with normal interrupt sources: when the software triggers an interrupt by setting the appropriate bit in the Software Interrupt register, but clears it again with a clear command using the IRCLVEC software register, the corresponding bit in the Interrupt Request register will be inactive for only one clock and then becomes active again immediately, i.e. in interaction of software interrupt and clear command – and only in this regard – the software interrupt is always treated as level-triggered.

2.3.2.9 Interrupt masking / prioritisation

The ICU provides the capability to individually mask each interrupt. The setting is made using the appropriate bit in the MASKREG register. The masking does not have any effect on the Interrupt Request register. Masked interrupts are neither forwarded to the CPU nor do they prevent the forwarding of non-masked interrupts with lower priority.

Each interrupt can be enabled and disabled by using the appropriate bit in the MASKREG or INTB_MASKREG register. The interrupts are masked at the outputs of the Interrupt-Request register (IRREG or INTB_IRREG). This means that a pending interrupt is also entered in the Interrupt Request register when it is masked. After reset, all mask bits are set and thus all interrupts disabled.

A single command can be used to disable all interrupts in the INTA as well as in the INTB subblock as if all mask bits were set. However, to revoke the global masking of all interrupts, only those interrupts not individually masked are activated.

Note that the reset value of the global mask bit differs in each subblock: The reset value in INTA block is '1' (all ints are masked by default), the reset value of the INTB block is '0'.

Not only each INTA but also each INTB receives its own freely selectable priority. The software can set the priority using the INTA_PRIOREG0 – INTA_PRIOREG<NUM_OF_INT-1> or INTB_PRIOREG0 – INTB_PRIOREG<NUM_OF_FINT-1> register. The value range for the priorities is:

- "0" to NUM_OF_INT-1 for INTAs and
- "0" to NUM_OF_INT-1 for INTBs.

Where the "0" value means the highest priority. After reset, all registers have the lowest priority.

The ICU treats all interrupts in their priority order. If several interrupts are present, the ICU selects the interrupt with the highest priority. If in this case two priorities are identical, the interrupt with the smaller number will be processed first.

The same priorities will be caught by the ICU as follows: when interrupts have the same priorities, the interrupt number is used as second priority level, where the following rule also applies here: the lower interrupt number has a higher priority than the higher interrupt number. However, it remains true that when an interrupt is already being processed by the CPU, it will not be interrupted by another interrupt of the same priority. This interrupt will be processed only when the first interrupt has been processed and that with the same priority is still present as next interrupt.

The ICU suppresses all interrupts whose priorities are less than or equal to the value that the software has parameterized in the LOCKREG register. This function can be enabled or disabled using the LOCKREG_ENABLE register bit. All interrupts locked using the LOCKREG register will still be entered and stored in the Interrupt Request register, but no longer participate in the priority resolution.

If concurrently to the write access to LOCKREG, an interrupt with an affected priority occurs, the interrupt signal is initially set to the CPU, but will be cleared again after two clocks at the latest. If the CPU, despite removed interrupt signal, performs an acknowledge, the default vector will be returned as interrupt vector.

This also means: in the same clock in which the information is written from the AHB bus into the LOCKREG register. If the interrupt request has already been pending longer without it being acknowledged and the software now writes this LOCKREG with the same

or higher priority, the default vector will be entered as interrupt vector also two clocks after the LOCKREG was written.

If interrupts have been locked using LOCKREG, they will still be entered in the Interrupt Request register. As soon as the LOCKREG function is deactivated, these stored interrupts participate again in the priority resolution and possibly initiate an interrupt.

A pending and valid interrupt leads to an active INTA output. A pending and valid fast interrupt leads to an active INTB output.

2.3.2.10 Interrupt post-processing

The acknowledge of interrupts by the CPU is performed with a read access to the INTAACK or INTBACK register, where the ICU returns the number of the current pending interrupt with the highest priority. The bit with the corresponding number will be set in the In-Service register (ISREG) and the INTA or INTB signal to the CPU is removed.

The ICU outputs a default vector when an acknowledge is performed although no interrupt is present on the CPU. The null vector is used as default vector. The internal status of the ICU with regard to IRREG and ISREG does not change.

A default vector can occur:

1. When the ICU receives an acknowledge command without it previously outputting a pending interrupt.
2. When the source of a level-triggered interrupt removes it before being confirmed by the CPU.
3. When an interrupt input is masked or a clear command is performed for an interrupt input while concurrently an interrupt occurs at this input. Because the CPU commands arrive at the interrupt controller with a time delay, the interrupt is initially forwarded to the CPU, however, removed after the command takes affect. If, however, the CPU responds with acknowledge, the interrupt controller can only assign the default vector, because in the meantime a valid interrupt is no longer present.
4. If the time between acknowledgment and the End of Interrupt event is less than 3 ICU clock cycles, the Interrupt line to the CPU may rise again, although there is no interrupt present. In this case the CPU will read the default vector.

Each set bit in the ISREG causes all interrupts with the same or lower priority to be disabled. All occurring interrupts will be detected independent of their priority and entered in the IRREG. Bits set in the ISREG, however, cause only interrupts with a higher priority (higher than all interrupts represented by these bits) to output an the interrupt to the CPU. Only when a bit in the ISREG is cleared by the EOI command will the interrupts with the same or lower priority be reactivated.

If during the processing of an interrupt, namely between acknowledge and EOI, an interrupt with higher priority occurs and also confirmed by the CPU with acknowledge, the corresponding bit will be set in the ISREG in addition to the previously set bits.

The ICU detects the end of the interrupt processing by the EOI command that results from any write access to INTAEND or INTBEND. Each EOI clears the bit in the ISREG that belongs to the interrupt that currently has the highest priority.

The Interrupt Controller Unit does not require any further information from the CPU to identify the interrupt whose bit must be cleared in the ISREG (non-specific EOI command).

If more than one bit is set in the ISREG because during the currently running interrupt process interrupts with higher priority have occurred, the EOI command clears the bit that belongs to the currently processed interrupt with the highest priority.

To ensure that the entry in the ISREG with the highest priority matches the interrupt for which the last acknowledge was issued (i.e. that currently being processed by the CPU), the assignment of the priorities may no longer be changed after an acknowledge.

The ICU considers the interrupt processing to have finished completely when all bits in the ISREG have been cleared. When the last bit has been cleared in the ISREG, all interrupts with lower priority that have been entered in the Interrupt Request register in the meantime will also be further processed.

2.3.2.11 Special functions

The ICU permits the additional confirmation of an interrupt by any write access to the INTB_ACK/INTA_ACK register. This function is deactivated after reset and must be activated prior to use with the INTB_UNLOCK_READ_ONLY register bit (for each INTA/INTB block separately).

This acknowledge using write access is destructive. This means the vector number of the confirmed interrupt then cannot be fetched using the ACK or the IVEC register. Consequently, this function must be enabled prior to its use.

As an additional special function, there is an ID register that contains the implemented version number of the ICU. The software can read this version number.

The ICU contains an ID register that contains the currently implemented version number of the IP. The software can fetch this version number. Each subblock (INTB/INTA) has its own version register. The ICU ID can be obtained from the register descriptions.

2.3.2.12 Debug functions

The CPU can fetch the vector number of the currently pending interrupt on the CPU without confirming it. For this purpose, the vector number can be fetched using two different addresses, one of which performs the acknowledge with the read access and the other is provided only for debug purposes.

An interrupt that was reported to the CPU using the INTA or INTB signal is acknowledged with a read access to the interrupt vector register (INTA_ACK or INTB_ACK). Because this access is destructive, namely, the vector number can no longer be read after the acknowledge, the additional address allows the vector to be fetched for debug purposes without initiating any further processing.

The ICU has an input 'Debug acknowledge' (e.g. coming from the ARM9-Core in previous designs) that can act directly on the global mask bit. This procedure can be enabled and disabled using the MASK_ALL_INPUT_EN software register. The input is synchronized by the ICU with a double flop. **In ERTEC 200P this input is tied inactive.**

2.3.2.13 Miscellaneous

2.3.2.13.1 Synchronizing the inputs

All incoming interrupts are synchronized in two stages to the operating clock of the ICU. It must be guaranteed that interrupts at the asynchronous inputs must remain at least two clocks of the operating clock so they can be reliably detected by the ICU.

Note: *All level triggered interrupts has to be high active. **Low Active level triggered interrupts are not supported by ICU** and must be inverted on ERTEC 200P toplevel.*

The CPU can use the DBG_ACK signal (coming from ARM926-Core) to permit direct throughput to the global Enable Bit of all interrupts in the ICU. Because this signal comes from the CPU, and thus from another clock domain, the signal in the ICU is synchronized twice to the operating clock.

2.3.2.13.2 Bus interface

The ICU contains a standard AHB slave interface working in accordance with the AHB Lite protocol. The data width is 32 bits. Word, halfword and byte accesses are permitted. Bursts are commuted into single accesses.

The signals HPROT and HMASTLOCK are not evaluated. The bus interface of the ICU carries out one wait state in maximum for reads. For writes no wait state is performed. The ICU performs no error response at all.

2.3.2.13.3 Sequences

The following is a typical interrupt cycle:

1. A valid interrupt is pending at the input. Valid here means that the correct edge has been detected (assuming: edge-triggering activated) and the interrupt has not been entered in the interrupt request register yet.
2. This interrupt is then entered in the interrupt request register.
3. Before this interrupt now takes part in the priority logic, it is checked if it is masked. If it is not and if the interrupt locking feature is deactivated, the interrupt takes part in the priority check. At the end of the priority check, the interrupt - or its number - which is active and has the highest priority is displayed. If there is no interrupt pending, the priority check does not display anything.
4. It is then checked if this interrupt with the currently highest priority has a higher priority than the one which might currently be processed – i.e. which is "in service". If the pending interrupt has a higher priority, the ICU activates the interrupt output in the direction of the CPU.

This means that the CPU recognizes an exception at the interrupt input and executes its interrupt service routine:

5. The INTAACK register is read, which indicates the interrupt that currently has the highest priority and wants to interrupt the CPU.
6. The reading of the INTAACK register triggers an acknowledge process in the ICU in which the set bit is deleted from the interrupt register and entered (set) in the in-service register. The ICU memorizes internally that this interrupt is currently processed. The ICU even memorizes the sequence in which the interrupts have been acknowledged in the case of nested interrupts or several interrupts occurring simultaneously. This is mandatory since the CPU issues an end-of-interrupt command after the execution of the interrupt service routine, by carrying out a write access to

the EOI register. The ICU recognizes this and deletes the corresponding bit from the in-service register and also simultaneously cancels the above-mentioned flag.

2.3.2.13.4 Operating rules

Reconfiguration of priorities

Although the ICU can handle the same priorities for several interrupts, the following has to be taken into account for the reconfiguration: The reconfiguration of priorities is only permissible when there are no interrupts "in service" anymore. To ensure that no unforeseen event occurs during that time, the software has to mask both interrupts having the same priority.

Level-sensitive interrupts

It has to be taken into account that a level-sensitive interrupt is acknowledged at its source (i.e. the level reset there) before it is acknowledged on the software side. This is necessary because otherwise another entry would be made in the interrupt request register for this interrupt.

Acknowledgement of the INTBs with write access to acknowledge register

The software also has the option of carrying out the acknowledgement of the INTB interrupt via a write access to the INTBACK register. For that it has to be noted that this write access is destructive and the number of the fast interrupt which triggered the interrupt on the CPU is lost irrevocably. Therefore, in this operating mode, the CPU has to know exactly who triggered the fast interrupt.

Using INTAs as INTBs

INTA and INTB always have to be set to the same triggering. This is not checked by the ICU and has to be ensured by the software. Moreover, the SW should normally mask the INTA input in the INTA sub-block used as INTB since otherwise an INTA as well as an INTB is signaled to the CPU upon the occurrence of the interrupt.

Identical priorities

Identical priorities are intercepted by the ICU as follows: If interrupts have the same priority, the interrupt number is used as the second priority level, whereby the following applies: The lower interrupt number has a higher priority than the higher interrupt number. The following nevertheless applies additionally: If an interrupt is already processed by the CPU, it cannot be interrupted by another interrupt of the same priority. Only when the first one has been executed and the next pending interrupt is still the one of the same priority, is this interrupt processed.

Hardware latencies in certain back2back software accesses

Due to hardware latencies, an old and wrong data item might be read back in the following back2back accesses (described is a writing on the first and immediate reading of the second register):

```
IRCLVEC -> ACK
IRCLVEC -> IRREG1-5
IRCLVEC -> SWIRREG1-5
LOCKREG -> ACK
MASK -> ACK
ACK->IRREG1-5
```

Furthermore there is following problem: If the ICU assigns an interrupt and the software acknowledges it followed by a “fast” EOI (fast means, that the EOI is issued 1-3 operating clock cycles after acknowledge), the ICU outputs a new interrupt due to internal hardware latencies. This new interrupt is deleted again after a few clock cycles. When the CPU acknowledges this interrupt it will read the default vector. This scenario is only a performance issue but not a functional one. Besides it is currently not known that any CPU is as fast as to perform such a fast EOI after acknowledge.

Deletion of interrupts during the reset phase

If the ICU is in reset and there are already high levels pending at the interrupt inputs when the ICU comes from the reset phase, these high levels are evaluated as rising edge by the edge detection as regards the hardware and an interrupt is immediately entered in the interrupt request register (since edge-triggering to the positive edge is set by default during the reset). This is not a valid interrupt and has to be deleted again immediately by the software by writing on INTA_IRCLVEC or INTB_IRCLVEC.

Procedure for locking priorities via LOCKREG

Since it takes a certain latency from the issuing of the software write to LOCKREG until the register in the hardware is actually written on, it is reasonable to read back the LOCKREG register again immediately. Only when the read-back value returns to the CPU (it does not have to be checked if the register value is correct, it is just about the waiting), can one be sure that no more interrupts are triggered which have a lower priority than the priority which was entered in the LOCKREG register. Apart from that, there are no other specialties to be observed for the writing on the register, it can be written on at any time.

2.3.2.13.5 Startup/shutdown

After the reset, all interrupts are deactivated (dedicated mask bits set as well as the global mask bit, if any). The priority of each interrupt is the lowest priority which can be assigned. This means that for initializing the ICU, all interrupts have to be assigned an own priority first before the mask bit is released. Before releasing the interrupts, it should also be determined which interrupts are level/edge-triggered and stated which edge is the active one, if necessary. If a priority locking is to be carried out, this has to be configured correctly and activated before.

Furthermore, it has to be taken into account that INTAs as well as INTBs are edge-triggered by default. If level-triggering is set, the interrupt has to be canceled at the source. Otherwise, it triggers a new interrupt (INTA or INTB) at the CPU after the acknowledgement.

The INTA applied to the INTB always has to be set to the same triggering. This is not checked by the ICU and has to be ensured by the software. Moreover, the SW should normally mask the INTA input in the INTA sub-block used as INTB since otherwise an INTA as well as an INTB is signaled to the CPU upon the occurrence of the interrupt.

2.3.2.14 Interrupt sources for ARM-IRQ

The Interrupt controller will have interrupts from the following function blocks:

IRQ no.	Interrupt Name	Type	Signal	Cause	Interrupt source
0	Default interrupt	-	<i>Default Vector</i>	Interrupt which is never assigned (must be tied to '0' on toplevel)	
1	DMA_IRQ_O	LT	DMA_IRQ_O	GDMA Interrupt	GDMA
2	INT_I2C	LT	INT_I2C	Combined Interrupt	I²C
3	Reserved		-----		
4	Reserved		-----		
5	INT_UART1	LT	UART1_UARTIN TR (group interrupt)	Receive FIFO half-full or full	UART1... 4
				Transmit FIFO half-empty or empty	
				Receive Timeout	
6	INT_UART1_ER R	LT	UART1_UARTEI NTR (Error interrupt)	Error interrupt	
7	INT_UART2	LT	UART2_UARTIN TR (group interrupt)	Receive FIFO half-full or full	
				Transmit FIFO half-empty or empty	
				Receive Timeout	
8	INT_UART2_ER R	LT	UART2_UARTEI NTR (Error interrupt)	Error interrupt	
9	INT_UART3	LT	UART3_UARTIN TR (group interrupt)	Receive FIFO half-full or full	
				Transmit FIFO half-empty or empty	
				Receive Timeout	
10	INT_UART3_ER R	LT	UART3_UARTEI NTR (Error interrupt)	Error interrupt	
11	INT_UART4	LT	UART4_UARTIN TR (group interrupt)	Receive FIFO half-full or full	
				Transmit FIFO half-empty or empty	
				Receive Timeout	
12	INT_UART4_ER R	LT	UART4_UARTEI NTR (Error interrupt)	Error interrupt	

IRQ no.	Interrupt Name	Type	Signal	Cause	Interrupt source
13	INT_SPI1	LT	SPI1_SSPINTR (group interrupt)	Transmit FIFO empty Transmit FIFO half-empty or empty Transmit FIFO not full Receive FIFO not empty Receive FIFO half-full or full Receive FIFO overrun	SPI1, 2
14	INT_SPI1_Overrun	LT	SPI1_SSPRORINTR (Error interrupt)	Overrun Error Interrupt	
15	INT_SPI2	LT	SPI2_SSPINTR (group interrupt)	Transmit FIFO empty Transmit FIFO half-empty or empty Transmit FIFO not full Receive FIFO not empty Receive FIFO half-full or full Receive FIFO overrun	
16	INT_SPI2_Overrun	LT	SPI2_SSPRORINTR (Error interrupt)	Overrun Error Interrupt	
17	INT_SPI_ParityError	ET/R	SPI_PARITY_ERROR	Interrupt for SPI1/2_PARITY_ERR	
18	Reserved		-----		
19	Reserved		-----		
20	Reserved		-----		
21	INT_TIMER0	*)	TIM_OUT0	Timer expired (value 0 reached)	TIMER
22	INT_TIMER1	*)	TIM_OUT1	Timer expired (value 0 reached)	
23	INT_TIMER2	*)	TIM_OUT2	Timer expired (value 0 reached)	
24	INT_TIMER3	*)	TIM_OUT3	Timer expired (value 0 reached)	
25	INT_TIMER4	*)	TIM_OUT4	Timer expired (value 0 reached)	
26	INT_TIMER5	*)	TIM_OUT5	Timer expired (value 0 reached)	
27	WD_INT_ARM926	ET/R	Watchdog ARM926	ARM926 Watchdog Interrupt	WD
28	SPI1_TFE	LT	SPI1_TFE	Transmit FIFO empty	SPI1, 2

IRQ no.	Interrupt Name	Type	Signal	Cause	Interrupt source
29	SPI1_RNE	LT	SPI1_RNE	Receive FIFO not empty (corresponds to SPI1_SSPRXDMA)	
30	SPI2_TFE	LT	SPI2_TFE	Transmit FIFO empty	
31	SPI2_RNE	LT	SPI2_RNE	Receive FIFO not empty (corresponds to SPI2_SSPRXDMA)	
32	INT_GPIO0	**)	GPIO0	External interrupt over input pin GPIO 0	GPIO
33	INT_GPIO1	**)	GPIO1	External interrupt over GPIO 1 input pin	
34	INT_GPIO2	**)	GPIO2	External interrupt over GPIO 2 input pin	
35	INT_GPIO3	**)	GPIO3	External interrupt over GPIO 3 input pin	
36	INT_GPIO4	**)	GPIO4	External interrupt over GPIO 4 input pin	
37	INT_GPIO5	**)	GPIO5	External interrupt over GPIO 5 input pin	
38	INT_GPIO6	**)	GPIO6	External interrupt over GPIO 6 input pin	
39	INT_GPIO7	**)	GPIO7	External interrupt over GPIO 7 input pin	
40	INT_GPIO8	**)	GPIO8	External interrupt over GPIO 8 input pin	
41	INT_GPIO9	**)	GPIO9	External interrupt over GPIO 9 input pin	
42	INT_GPIO10	**)	GPIO10	External interrupt over GPIO 10 input pin	
43	INT_GPIO11	**)	GPIO11	External interrupt over GPIO 11 input pin	
44	INT_GPIO12	**)	GPIO12	External interrupt over GPIO 12 input pin	
45	INT_GPIO13	**)	GPIO13	External interrupt over GPIO 13 input pin	
46	INT_GPIO14	**)	GPIO14	External interrupt over GPIO 14 input pin	

IRQ no.	Interrupt Name	Type	Signal	Cause	Interrupt source
47	INT_GPIO15	**)	GPIO15	External interrupt over GPIO 15 input pin	
48	EDC_Event	LT		Combined Interrupt (EDC Event Register)	EDC
49	INT_PLL_UNLOCK	ET/R	<i>not</i> INT_PLL_300_LOCK	Interrupt if PLL is not locked.	PLL
50	INT_PLL_LOSS	ET/R	INT_PLL_LOSS	Interrupt if oscillator clock is not in the monitored range.	
51	AHB Address Error	ET/R	ACCESS_ERR (Group interrupt for the individual masters at the AHB)	Access to unavailable address at the AHB	QVZ
52	APB Address Error	ET/R	ACCESS_ERR (Group interrupt for the individual slaves at the AHB/APB)	Access to unavailable address at the APB	
53	EMC Address Error	ET/R	EMC_QVZ_INT	Access to unavailable address in the EMC address range	
54	PER_IF_ARM_IRQ	ET/R	PER_IF_ARM_IRQ	Event unit group interrupt	PerIF
55	PHY1/2_INT	ET/R	P1/2_INTERP	PHY1/2 interrupt	PHY1, 2
56	PN_IRQ2(0)	ET/R	PN_ICU2_IRQ2(0)	PN-ICU2 Combined Interrupt 1	PN-IP
57	PN_IRQ2(1)	ET/R	PN_ICU2_IRQ2(1)	PN-ICU2 Combined Interrupt 2	
58	PN_IRQ2(2)	ET/R	PN_ICU2_IRQ2(2)	Selectable PN-ICU Interrupt	
59	PN_IRQ2(3)	ET/R	PN_ICU2_IRQ2(3)	Selectable PN-ICU Interrupt	
60	PN_IRQ2(4)	ET/R	PN_ICU2_IRQ2(4)	Selectable PN-ICU Interrupt	
61	PN_IRQ2(5)	ET/R	PN_ICU2_IRQ2(5)	Selectable PN-ICU Interrupt	
62	PN_IRQ2(6)	ET/R	PN_ICU2_IRQ2(6)	Selectable PN-ICU Interrupt	
63	PN_IRQ2(7)	ET/R	PN_ICU2_IRQ2(7)	Selectable PN-ICU Interrupt	
64	PN_IRQ2(8)	ET/R	PN_ICU2_IRQ2(8)	Selectable PN-ICU Interrupt	

IRQ no.	Interrupt Name	Type	Signal	Cause	Interrupt source
		R	)		
65	PN_IRQ2(9)	ET/R	PN_ICU2_IRQ2(9)	Selectable PN-ICU Interrupt	
66	PN_IRQ2(10)	ET/R	PN_ICU2_IRQ2(10)	Selectable PN-ICU Interrupt	
67	PN_IRQ2(11)	ET/R	PN_ICU2_IRQ2(11)	Selectable PN-ICU Interrupt	
68	PN_IRQ2(12)	ET/R	PN_ICU2_IRQ2(12)	Selectable PN-ICU Interrupt	
69	PN_IRQ2(13)	ET/R	PN_ICU2_IRQ2(13)	Selectable PN-ICU Interrupt	
70	PN_IRQ2(14)	ET/R	PN_ICU2_IRQ2(14)	Selectable PN-ICU Interrupt	
71	PN_IRQ2(15)	ET/R	PN_ICU2_IRQ2(15)	Selectable PN-ICU Interrupt	
72	PNPLL_OUT9	ET/R	PNPLL_OUT(9)	Selectable PN-PLL Event	
73	PNPLL_OUT10	ET/R	PNPLL_OUT(10)	Selectable PN-PLL Event	
74	PNPLL_OUT11	ET/R	PNPLL_OUT(11)	Selectable PN-PLL Event	
75	PNPLL_OUT12	ET/R	PNPLL_OUT(12)	Selectable PN-PLL Event	
76	PNPLL_OUT13	ET/R	PNPLL_OUT(13)	Selectable PN-PLL Event	
77	PNPLL_OUT14	ET/R	PNPLL_OUT(14)	Selectable PN-PLL Event	
78	SW_INT_0	-	<i>not used in HW</i>		(SW)
79	SW_INT_1	-	<i>not used in HW</i>		
80	SW_INT_2	-	<i>not used in HW</i>		
81	SW_INT_3	-	<i>not used in HW</i>		
82	SW_INT_4	-	<i>not used in HW</i>		
83	SW_INT_5	-	<i>not used in HW</i>		
84	SW_INT_6	-	<i>not used in HW</i>		
85	SW_INT_7	-	<i>not used in HW</i>		
86	IP Access Error	ET/R	ACCESS_ERR (IPs)	Combined Interrupt for address mismatches in	QVZ

IRQ no.	Interrupt Name	Type	Signal	Cause	Interrupt source
				modules	
87	ETB_ACQCOMP	ET/R	ETB_ACQCOMP	Combined Interrupt for address mismatches in modules	ARM926
88	Invalid I-TCM926 Access	ET/R		ARM926 has accessed an I-TCM address gap	
89	Invalid D-TCM926 Access	ET/R		ARM926 has accessed a D-TCM address gap	
90	ETB_FULL	ET/R	ETB_FULL	Trace overrun in trace buffer	
91	Reserved		-----		
92	Reserved		-----		
93	Reserved		-----		
94	COMMRX926	LT	COMMRX926	Interrupt for Realtime Debugger (receive Buffer contains data)	ARM926
95	COMMTX926	LT	COMMTX926	Interrupt for Realtime Debugger (transmit buffer is empty)	

Type ET/R

Edge triggered interrupt, on the rising edge.

Type ET/F

Edge triggered interrupt, on the falling edge.

Type LT

Level triggered interrupt.

Type *)

This type is dependent on the selected mode of the module

Type **)

This type is dependent on the external switch (ASIC-Pins). It can be ET/F or ET/R or LT, depending on the external connection. Parameterization must be done depending on the request of the application about the Host-ICU (see chapter 2.3.2.17).

Table 9: Host INTA, Interrupt sources

Information:

The IRQ interrupt output of ARM-ICU (ICUIRQ_O) is inverted and connected to the interrupt input (nIRQ) of ARM926EJ-S.

2.3.2.14.1 Interrupts for accesses to missing addresses

The QVZ units detect any access to missing addresses and generate a pulse with duration $T_p = 2/125$ MHz. This pulse must be processed in the Interrupt Controller as an edge-triggered interrupt.

2.3.2.14.2 Confirmation delay in the Memory Controller (EMC) address area

In the address area of the EMC, the XCE_PER(3:0) outputs become active and the Memory Controller waits for the XRDY_PER input signal. The confirmation delay monitoring activated in the Async Wait Cycle Config Register (see chap. 2.3.5.8) after ((MAX_EXT_WAIT + 1) x 16) AHB clock pulses creates an internal ready signal for the Memory Controller and an IRQ. The IRQ will be removed when the QVZ monitoring is disabled.

2.3.2.15 Interrupt sources for ARM-FIQ

The Interrupt controller will have interrupts from the following function blocks:

IRQ no.	Interrupt Name	Type	Signal	Cause	Interrupt source
0	Default interrupt	-	<i>Default Vector</i>	Interrupt which is never assigned (must be tied to '0' on toplevel)	
1	FIQ_SEL_1	***)		Selectable interrupt from the INTA-Sources about <i>FIQ_SEL_1</i> Register	
2	FIQ_SEL_2	***)		Selectable interrupt from the INTA-Sources about <i>FIQ_SEL_2</i> Register	
3	FIQ_SEL_3	***)		Selectable interrupt from the INTA-Sources about <i>FIQ_SEL_3</i> Register	
4	FIQ_SEL_4	***)		Selectable interrupt from the INTA-Sources about <i>FIQ_SEL_4</i> Register	
5	FIQ_SEL_5	***)		Selectable interrupt from the INTA-Sources about <i>FIQ_SEL_5</i> Register	
6	FIQ_SEL_6	***)		Selectable interrupt from the INTA-Sources about <i>FIQ_SEL_6</i> Register	
7	FIQ_SEL_7	***)		Selectable interrupt from the INTA-Sources about <i>FIQ_SEL_7</i> Register	

Type ***) This type is dependent on the Type of ARM-IRQ. It can be ET/F or ET/R or LT.

Table 10: Host INTB, Interrupt sources

Information:

The FIQ interrupt output of ARM-ICU (CUFIQ_O) is inverted and connected to the interrupt input (nFIQ) of ARM926EJ-S.

2.3.2.15.1 High-priority Interrupts for Debugging

As effective RT debugging, the interrupts of the embedded ICE RT communication channel (COMMRX926, COMMTX926) or the interrupt of UART2 can be mapped over FIQ1 to FIQ7. This allows interrupt routine debugging.

2.3.2.16 Address Mapping

Start_Address	End_Address	Modul/Memory_Name	Interface
4000h	4AFFFh	icu_ertec_addr_dec_top	AHB
4000h	7FFFh	icu96_inst	AHB
8000h	BFFFh	icu8_inst	AHB
48000h	4AFFFh	icu_general_registers_inst	AHB

Module	Register/Memory	Read	Write	Address
<u>/icu_ertec_addr_dec_top/icu96_inst</u>				
	ID_REGISTER	r		4000h
	IRVEC	rh		4004h
	ACK	rht		4008h
	IRCLVEC		wt	400Ch
	MASKALL	r	w	4010h
	EOI		t	4014h
	UNLOCK_RD_ONLY_ACK	r	w	4018h
	MASK_ALL_INPUT_EN	r	w	401Ch
	LOCKREG	r	w	4020h
	MASKREG0	r	w	5000h
	MASKREG1	r	w	5004h
	MASKREG2	r	w	5008h
	IRR0	rh		5100h
	IRR1	rh		5104h
	IRR2	rh		5108h
	ISR0	rh		5200h
	ISR1	rh		5204h
	ISR2	rh		5208h
	TRIGREG0	r	w	5300h
	TRIGREG1	r	w	5304h
	TRIGREG2	r	w	5308h
	EDGEREG0	r	w	5400h
	EDGEREG1	r	w	5404h

	EDGEREG2	r	w	5408h
	SWIRREG0	rh	w	5500h
	SWIRREG1	rh	w	5504h
	SWIRREG2	rh	w	5508h
	PRIOREG0	r	w	6000h
	PRIOREG1	r	w	6004h
	PRIOREG2	r	w	6008h
	PRIOREG3	r	w	600Ch
	PRIOREG4	r	w	6010h
	PRIOREG5	r	w	6014h
	PRIOREG6	r	w	6018h
	PRIOREG7	r	w	601Ch
	PRIOREG8	r	w	6020h
	PRIOREG9	r	w	6024h
	PRIOREG10	r	w	6028h
	PRIOREG11	r	w	602Ch
	PRIOREG12	r	w	6030h
	PRIOREG13	r	w	6034h
	PRIOREG14	r	w	6038h
	PRIOREG15	r	w	603Ch
	PRIOREG16	r	w	6040h
	PRIOREG17	r	w	6044h
	PRIOREG18	r	w	6048h
	PRIOREG19	r	w	604Ch
	PRIOREG20	r	w	6050h
	PRIOREG21	r	w	6054h
	PRIOREG22	r	w	6058h
	PRIOREG23	r	w	605Ch
	PRIOREG24	r	w	6060h
	PRIOREG25	r	w	6064h
	PRIOREG26	r	w	6068h
	PRIOREG27	r	w	606Ch
	PRIOREG28	r	w	6070h

	PRIOREG29	r	w	6074h
	PRIOREG30	r	w	6078h
	PRIOREG31	r	w	607Ch
	PRIOREG32	r	w	6080h
	PRIOREG33	r	w	6084h
	PRIOREG34	r	w	6088h
	PRIOREG35	r	w	608Ch
	PRIOREG36	r	w	6090h
	PRIOREG37	r	w	6094h
	PRIOREG38	r	w	6098h
	PRIOREG39	r	w	609Ch
	PRIOREG40	r	w	60A0h
	PRIOREG41	r	w	60A4h
	PRIOREG42	r	w	60A8h
	PRIOREG43	r	w	60ACh
	PRIOREG44	r	w	60B0h
	PRIOREG45	r	w	60B4h
	PRIOREG46	r	w	60B8h
	PRIOREG47	r	w	60BCh
	PRIOREG48	r	w	60C0h
	PRIOREG49	r	w	60C4h
	PRIOREG50	r	w	60C8h
	PRIOREG51	r	w	60CCh
	PRIOREG52	r	w	60D0h
	PRIOREG53	r	w	60D4h
	PRIOREG54	r	w	60D8h
	PRIOREG55	r	w	60DCh
	PRIOREG56	r	w	60E0h
	PRIOREG57	r	w	60E4h
	PRIOREG58	r	w	60E8h
	PRIOREG59	r	w	60ECh
	PRIOREG60	r	w	60F0h
	PRIOREG61	r	w	60F4h

	PRIOREG62	r	w	60F8h
	PRIOREG63	r	w	60FCh
	PRIOREG64	r	w	6100h
	PRIOREG65	r	w	6104h
	PRIOREG66	r	w	6108h
	PRIOREG67	r	w	610Ch
	PRIOREG68	r	w	6110h
	PRIOREG69	r	w	6114h
	PRIOREG70	r	w	6118h
	PRIOREG71	r	w	611Ch
	PRIOREG72	r	w	6120h
	PRIOREG73	r	w	6124h
	PRIOREG74	r	w	6128h
	PRIOREG75	r	w	612Ch
	PRIOREG76	r	w	6130h
	PRIOREG77	r	w	6134h
	PRIOREG78	r	w	6138h
	PRIOREG79	r	w	613Ch
	PRIOREG80	r	w	6140h
	PRIOREG81	r	w	6144h
	PRIOREG82	r	w	6148h
	PRIOREG83	r	w	614Ch
	PRIOREG84	r	w	6150h
	PRIOREG85	r	w	6154h
	PRIOREG86	r	w	6158h
	PRIOREG87	r	w	615Ch
	PRIOREG88	r	w	6160h
	PRIOREG89	r	w	6164h
	PRIOREG90	r	w	6168h
	PRIOREG91	r	w	616Ch
	PRIOREG92	r	w	6170h
	PRIOREG93	r	w	6174h
	PRIOREG94	r	w	6178h

	PRIOREG95	r	w	617Ch
/icu_ertec_addr_dec_top/icu8_inst				
	ID_REGISTER	r		8000h
	IRVEC	rh		8004h
	ACK	rht		8008h
	IRCLVEC		wt	800Ch
	MASKALL	r	w	8010h
	EOI		t	8014h
	UNLOCK_RD_ONLY_ACK	r	w	8018h
	MASK_ALL_INPUT_EN	r	w	801Ch
	LOCKREG	r	w	8020h
	MASKREG0	r	w	9000h
	IRR0	rh		9100h
	ISR0	rh		9200h
	TRIGREG0	r	w	9300h
	EDGEREG0	r	w	9400h
	SWIRREG0	rh	w	9500h
	PRIOREG0	r	w	A000h
	PRIOREG1	r	w	A004h
	PRIOREG2	r	w	A008h
	PRIOREG3	r	w	A00Ch
	PRIOREG4	r	w	A010h
	PRIOREG5	r	w	A014h
	PRIOREG6	r	w	A018h
	PRIOREG7	r	w	A01Ch
/icu_ertec_addr_dec_top/icu_general_registers_inst				
<i>not usable</i>	FIQ_SEL_0	r	w	48000h
	FIQ_SEL_1	r	w	48004h
	FIQ_SEL_2	r	w	48008h
	FIQ_SEL_3	r	w	4800Ch
	FIQ_SEL_4	r	w	48010h

	FIQ_SEL_5	r	w	48014h
	FIQ_SEL_6	r	w	48018h
	FIQ_SEL_7	r	w	4801Ch

2.3.2.17 Register description

Module: /icu_ertec_addr_dec_top/icu96_inst

Register:	ID_REGISTER			Address:	4000h	
Bits:	15dt0	Reset value:	0006h	Attributes:	r	
Description:	Version number of the Interrupt Controller Unit Version number 1: 128 IR 2: 16 IR, no ZSV error, default vector FFFFFFFF 3: 32 IR 4: 32 IR, DBG-ACK mask 5: generic ICU 6: ICU core IP					

Register:	IRVEC			Address:	4004h	
Bits:	6dt0	Reset value:	00h	Attributes:	rh	
Description:	Interrupt Vector Register Number of the highest priority pending Interrupt Request For pending valid interrupt: binary code of the Interrupt number. Default vector: 0h Important: If SW acknowledges the current pending Interrupt Request with a write access on ACK, the content on IRVEC is also lost.					

Register:	ACK			Address:	4008h	
Bits:	6dt0	Reset value:	00h	Attributes:	rht	
Description:	Interrupt Vector Register with IRQ Acknowledge Acknowledge the highest priority pending interrupt request by reading the associated interrupt vector For valid request: binary code of the input number Otherwise: Default vector 0h					

Register:	IRCLVEC			Address:	400Ch	
Bits:	6dt0	Reset value:	00h	Attributes:		wt
Description:	Interrupt Request Clear Vector Direct clearing of an interrupt request in the Interrupt Request Register Binary code of the input number of the request to be cleared					

Register:	MASKALL			Address:	4010h	
Bits:	0	Reset value:	1h	Attributes:	r	w
Description:	Mask all interrupts					

	Global lock of all IRQ interrupt inputs '0' = Enable all unmasked IRQ interrupt inputs (use the set mask bits) '1' = Global lock of all IRQ interrupt inputs (independent of the interrupt mask)				
--	--	--	--	--	--

Register:	EOI			Address:	4014h
Bits:	0	Reset value:	0h	Attributes:	t
Description:	End of interrupt (IRQ) Informs the IRQ interrupt controller about the completion of the interrupt service routine associated with the current request				

Register:	UNLOCK_RD_ONLY_ACK			Address:	4018h
Bits:	0	Reset value:	0h	Attributes:	r w
Description:	Unlocks the read only acknowledge mechanism. By setting this bit to 1 one can reach an acknowledge via a write access to ACK 0: Read only mechanism is active to acknowledge an Interrupt Request 1: Also a write access to ACK acknowledges the Interrupt Request. Be aware that this write access is destructive – the data is never be accessible any more.				

Register:	MASK_ALL_INPUT_EN			Address:	401Ch
Bits:	0	Reset value:	0h	Attributes:	r w
Description:	Enable the masking of all interrupt inputs using e.g. a DBGACK signal from the CPU. There is an input to the generic ICU to mask all interrupts. This input is called "mask_all". 0: Function disabled, "mask_all" is not used 1: Function enabled, all IRQ inputs are masked when "mask_all" =1				

Register:	LOCKREG			Address:	4020h
Bits:	31dt0	Reset value:	00000000h	Attributes:	r w
Description:	Priority Lock Register Specification of a priority to lock interrupt requests with lower or equal priority				
Bit	Identifier	Reset	Attr.	Function / Description	
6dt0	LOCK_PRIO	00h	r w	Binary code of the locking priority	
31	LOCKREG_ENABLE	0h	r w	0 = lock inactive / 1 = lock active	

Register:	MASKREG0			Address:	5000h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:	Interrupt Mask Register Enable/disable the interrupt inputs 0 – 31 inputs of the interrupt controller '0' = Interrupt input enabled '1' = Interrupt input disabled					

Register:	MASKREG1			Address:	5004h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:	Interrupt Mask Register Enable/disable the interrupt inputs 32 – 63 inputs of the interrupt controller '0' = Interrupt input enabled '1' = Interrupt input disabled					

Register:	MASKREG2			Address:	5008h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:	Interrupt Mask Register Enable/disable the interrupt inputs 64 – 95 inputs of the interrupt controller '0' = Interrupt input enabled '1' = Interrupt input disabled					

Register:	IRR0			Address:	5100h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Request Register Flag for the Interrupt Request detected as result of a positive edge 0 – 31 inputs of the interrupt controller '0' = No request '1' = Request has occurred					

Register:	IRR1			Address:	5104h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Request Register Flag for the Interrupt Request detected as result of a positive edge 32 – 63 inputs of the interrupt controller '0' = No request '1' = Request has occurred					

Register:	IRR2			Address:	5108h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Request Register Flag for the Interrupt Request detected as result of a positive edge 64 – 95 inputs of the interrupt controller '0' = No request '1' = Request has occurred					

Register:	ISR0			Address:	5200h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	In Service Register Flag for Interrupt Request acknowledged by CPU 0 – 31 inputs of the interrupt controller '0' = Interrupt Request not acknowledged '1' = Interrupt Request has been acknowledged					

Register:	ISR1			Address:	5204h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	In Service Register Flag for Interrupt Request acknowledged by CPU 32 – 63 inputs of the interrupt controller '0' = Interrupt Request not acknowledged '1' = Interrupt Request has been acknowledged					

Register:	ISR2			Address:	5208h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	In Service Register Flag for Interrupt Request acknowledged by CPU 64 – 95 inputs of the interrupt controller '0' = Interrupt Request not acknowledged '1' = Interrupt Request has been acknowledged					

Register:	TRIGREG0			Address:	5300h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Trigger Select Register Select the interrupt detection Interrupt inputs 0 – 31 0 = Interrupt detection using edge 1 = Interrupt detection using level					

Register:	TRIGREG1			Address:	5304h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Trigger Select Register Select the interrupt detection Interrupt inputs 32 – 63 0 = Interrupt detection using edge 1 = Interrupt detection using level					

Register:	TRIGREG2			Address:	5308h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Trigger Select Register Select the interrupt detection Interrupt inputs 64 – 95 0 = Interrupt detection using edge 1 = Interrupt detection using level					

Register:	EDGEREG0			Address:	5400h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Edge Select Register Select the edge for the interrupt detection (only when edge detection has been set for the associated input) Interrupt inputs 0 – 31 0 = Interrupt detection for positive edge 1 = Interrupt detection for negative edge					

Register:	EDGEREG1			Address:	5404h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Edge Select Register Select the edge for the interrupt detection (only when edge detection has been set for the associated input) Interrupt inputs 32 – 63 0 = Interrupt detection for positive edge 1 = Interrupt detection for negative edge					

Register:	EDGEREG2			Address:	5408h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Edge Select Register Select the edge for the interrupt detection (only when edge detection has been set for the associated input) Interrupt inputs 64 – 95					

	0 = Interrupt detection for positive edge 1 = Interrupt detection for negative edge
--	--

Register:	SWIRREG0			Address:	5500h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	w
Description:	Software Interrupt Register Specification of interrupt requests Interrupt inputs 0 – 31 0 = No interrupt request 1 = Set interrupt request					

Register:	SWIRREG1			Address:	5504h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	w
Description:	Software Interrupt Register Specification of interrupt requests Interrupt inputs 32 – 63 0 = No interrupt request 1 = Set interrupt request					

Register:	SWIRREG2			Address:	5508h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	w
Description:	Software Interrupt Register Specification of interrupt requests Interrupt inputs 64 – 95 0 = No interrupt request 1 = Set interrupt request					

Register:	PRIOREG0			Address:	6000h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG1			Address:	6004h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG2			Address:	6008h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG3			Address:	600Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG4			Address:	6010h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG5			Address:	6014h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG6			Address:	6018h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG7			Address:	601Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG8			Address:	6020h	
-----------	----------	--	--	----------	-------	--

Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG9			Address:	6024h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG10			Address:	6028h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG11			Address:	602Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG12			Address:	6030h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG13			Address:	6034h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG14			Address:	6038h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w

Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input		
--------------	---	--	--

Register:	PRIOREG15			Address:	603Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG16			Address:	6040h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG17			Address:	6044h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG18			Address:	6048h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG19			Address:	604Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG20			Address:	6050h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register					

	Specification of the priority of an interrupt request at the associated Binary code of the priority input		
--	--	--	--

Register:	PRIOREG21			Address:	6054h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG22			Address:	6058h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG23			Address:	605Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG24			Address:	6060h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG25			Address:	6064h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG26			Address:	6068h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG27			Address:	606Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG28			Address:	6070h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG29			Address:	6074h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG30			Address:	6078h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG31			Address:	607Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG32			Address:	6080h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG33			Address:	6084h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG34			Address:	6088h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG35			Address:	608Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG36			Address:	6090h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG37			Address:	6094h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG38			Address:	6098h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG39			Address:	609Ch	
-----------	-----------	--	--	----------	-------	--

Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG40			Address:	60A0h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG41			Address:	60A4h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG42			Address:	60A8h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG43			Address:	60ACh	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG44			Address:	60B0h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG45			Address:	60B4h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w

Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input		
--------------	---	--	--

Register:	PRIOREG46			Address:	60B8h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG47			Address:	60BCh	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG48			Address:	60C0h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG49			Address:	60C4h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG50			Address:	60C8h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG51			Address:	60CCh	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register					

	Specification of the priority of an interrupt request at the associated Binary code of the priority input		
--	--	--	--

Register:	PRIOREG52			Address:	60D0h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG53			Address:	60D4h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG54			Address:	60D8h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG55			Address:	60DCh	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG56			Address:	60E0h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG57			Address:	60E4h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG58			Address:	60E8h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG59			Address:	60ECh	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG60			Address:	60F0h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG61			Address:	60F4h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG62			Address:	60F8h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG63			Address:	60FCh	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG64			Address:	6100h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG65			Address:	6104h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG66			Address:	6108h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG67			Address:	610Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG68			Address:	6110h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG69			Address:	6114h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG70			Address:	6118h	
-----------	-----------	--	--	----------	-------	--

Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG71			Address:	611Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG72			Address:	6120h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG73			Address:	6124h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG74			Address:	6128h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG75			Address:	612Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG76			Address:	6130h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w

Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input		
--------------	---	--	--

Register:	PRIOREG77			Address:	6134h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG78			Address:	6138h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG79			Address:	613Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG80			Address:	6140h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG81			Address:	6144h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG82			Address:	6148h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register					

	Specification of the priority of an interrupt request at the associated Binary code of the priority input
--	--

Register:	PRIOREG83			Address:	614Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG84			Address:	6150h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG85			Address:	6154h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG86			Address:	6158h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG87			Address:	615Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG88			Address:	6160h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG89			Address:	6164h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG90			Address:	6168h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG91			Address:	616Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG92			Address:	6170h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG93			Address:	6174h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG94			Address:	6178h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG95			Address:	617Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Module: /icu_ertec_addr_dec_top/icu8_inst

Register:	ID_REGISTER			Address:	8000h	
Bits:	15dt0	Reset value:	0006h	Attributes:	r	
Description:	Version number of the Interrupt Controller Unit Version number 1: 128 IR 2: 16 IR, no ZSV error, default vector FFFFFFFF 3: 32 IR 4: 32 IR, DBG-ACK mask 5: generic ICU 6:ICU core IP					

Register:	IRVEC			Address:	8004h	
Bits:	2dt0	Reset value:	0h	Attributes:	rh	
Description:	Interrupt Vector Register Number of the highest priority pending Interrupt Request For pending valid interrupt: binary code of the Interrupt number. Default vector: 0h Important: If SW acknowledges the current pending Interrupt Request with a write access on ACK, the content on IRVEC is also lost.					

Register:	ACK			Address:	8008h	
Bits:	2dt0	Reset value:	0h	Attributes:	rht	
Description:	Interrupt Vector Register with IRQ Acknowledge Acknowledge the highest priority pending interrupt request by reading the associated interrupt vector For valid request: binary code of the input number Otherwise: Default vector 0h					

Register:	IRCLVEC			Address:	800Ch	
Bits:	2dt0	Reset value:	0h	Attributes:		wt
Description:	Interrupt Request Clear Vector Direct clearing of an interrupt request in the Interrupt Request Register Binary code of the input number of the request to be cleared					

Register:	MASKALL			Address:	8010h	
Bits:	0	Reset value:	0h	Attributes:	r	w
Description:	Mask all interrupts Global lock of all IRQ interrupt inputs '0' = Enable all unmasked IRQ interrupt inputs (use the set mask bits) '1' = Global lock of all IRQ interrupt inputs (independent of the interrupt mask)					

Register:	EOI			Address:	8014h	
Bits:	0	Reset value:	0h	Attributes:		t
Description:	End of interrupt (IRQ) Informs the IRQ interrupt controller about the completion of the interrupt service routine associated with the current request					

Register:	UNLOCK_RD_ONLY_ACK			Address:	8018h	
Bits:	0	Reset value:	0h	Attributes:	r	w
Description:	Unlocks the read only acknowledge mechanism. By setting this bit to 1 one can reach an acknowledge via a write access to ACK 0: Read only mechanism is active to acknowledge an Interrupt Request 1: Also a write access to ACK acknowledges the Interrupt Request. Be aware that this write access is destructive – the data is never be accessible any more.					

Register:	MASK_ALL_INPUT_EN			Address:	801Ch	
Bits:	0	Reset value:	0h	Attributes:	r	w
Description:	Enable the masking of all interrupt inputs using e.g. a DBGACK signal from the CPU. There is an input to the generic ICU to mask all interrupts. This input is called "mask_all". 0: Function disabled, "mask_all" is not used 1: Function enabled, all IRQ inputs are masked when "mask_all" =1					

Register:	LOCKREG			Address:	8020h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Priority Lock Register Specification of a priority to lock interrupt requests with lower or equal priority					

Bit	Identifier	Reset	Attr.		Function / Description
2dt0	LOCK_PRIO	0h	r	w	Binary code of the locking priority
31	LOCKREG_ENABLE	0h	r	w	0 = lock inactive / 1 = lock active

Register:	MASKREG0				Address:	9000h	
Bits:	7dt0	Reset value:	FFh	Attributes:	r	w	
Description:	Interrupt Mask Register Enable/disable the interrupt inputs interrupt inputs 0 – 7 0' = Interrupt input enabled '1' = Interrupt input disabled						

Register:	IRR0				Address:	9100h	
Bits:	7dt0	Reset value:	00h		Attributes:	rh	
Description:	Request Register Flag for the Interrupt Request detected as result of a positive edge 0 – 7 inputs of the interrupt controller '0' = No request '1' = Request has occurred						

Register:	ISR0				Address:	9200h	
Bits:	7dt0	Reset value:	00h		Attributes:	rh	
Description:	In Service Register Flag for Interrupt Request acknowledged by CPU 0 – 7 inputs of the interrupt controller '0' = Interrupt Request not acknowledged '1' = Interrupt Request has been acknowledged						

Register:	TRIGREG0				Address:	9300h	
Bits:	7dt0	Reset value:	00h		Attributes:	r	w
Description:	Trigger Select Register Select the interrupt detection Interrupt inputs 0 – 7 0 = Interrupt detection using edge 1 = Interrupt detection using level						

Register:	EDGEREG0				Address:	9400h	
Bits:	7dt0	Reset value:	00h		Attributes:	r	w

Description:	Edge Select Register Select the edge for the interrupt detection (only when edge detection has been set for the associated input) Interrupt inputs 0 – 7 0 = Interrupt detection for positive edge 1 = Interrupt detection for negative edge		
--------------	---	--	--

Register:	SWIRREG0		Address:	9500h	
Bits:	7dt0	Reset value:	00h	Attributes:	rh w
Description:	Software Interrupt Register Specification of interrupt requests Interrupt inputs 0 – 7 0 = No interrupt request 1 = Set interrupt request				

Register:	PRIOREG0		Address:	A000h	
Bits:	2dt0	Reset value:	7h	Attributes:	r w
Description:	Priority Register for IRQ0, the higher the number, the lower the priority				

Register:	PRIOREG1		Address:	A004h	
Bits:	2dt0	Reset value:	7h	Attributes:	r w
Description:	Priority Register for IRQ1, the higher the number, the lower the priority				

Register:	PRIOREG2		Address:	A008h	
Bits:	2dt0	Reset value:	7h	Attributes:	r w
Description:	Priority Register for IRQ2, the higher the number, the lower the priority				

Register:	PRIOREG3		Address:	A00Ch	
Bits:	2dt0	Reset value:	7h	Attributes:	r w
Description:	Priority Register for IRQ3, the higher the number, the lower the priority				

Register:	PRIOREG4		Address:	A010h	
Bits:	2dt0	Reset value:	7h	Attributes:	r w
Description:	Priority Register for IRQ4, the higher the number, the lower the priority				

Register:	PRIOREG5		Address:	A014h	
-----------	----------	--	----------	-------	--

Bits:	2dt0	Reset value:	7h	Attributes:	r	w
Description:	Priority Register for IRQ5, the higher the number, the lower the priority					

Register:	PRIOREG6			Address:	A018h	
Bits:	2dt0	Reset value:	7h	Attributes:	r	w
Description:	Priority Register for IRQ6, the higher the number, the lower the priority					

Register:	PRIOREG7			Address:	A01Ch	
Bits:	2dt0	Reset value:	7h	Attributes:	r	w
Description:	Priority Register for IRQ7, the higher the number, the lower the priority					

Module: /icu_ertec_addr_dec_top/icu_general_registers_inst

Register:	FIQ_SEL_0			Address:	48000h	
Bits:	6dt0	Reset value:	00h	Attributes:	r	w
Description:	Important: This INTB_SEL0 register can be written and read by Software - nevertheless this register does not affect the Hardware at all					

Register:	FIQ_SEL_1			Address:	48004h	
Bits:	6dt0	Reset value:	00h	Attributes:	r	w
Description:	Selects one of the IRQs as input to FIQ processing. Important: The FIQ_SEL register have to be configured with different values (except 0) for each select register. The ICU does not take care at all about this issue.					

Register:	FIQ_SEL_2			Address:	48008h	
Bits:	6dt0	Reset value:	00h	Attributes:	r	w
Description:	Selects one of the IRQs as input to FIQ processing. Important: The FIQ_SEL register have to be configured with different values (except 0) for each select register. The ICU does not take care at all about this issue.					

Register:	FIQ_SEL_3			Address:	4800Ch	
Bits:	6dt0	Reset value:	00h	Attributes:	r	w
Description:	Selects one of the IRQs as input to FIQ processing. Important: The FIQ_SEL register have to be configured with different values (except 0) for each select register. The ICU does not take care at all					

	about this issue.
--	-------------------

Register:	FIQ_SEL_4			Address:	48010h	
Bits:	6dt0	Reset value:	00h	Attributes:	r	w
Description:	Selects one of the IRQs as input to FIQ processing. Important: The FIQ_SEL register have to be configured with different values (except 0) for each select register. The ICU does not take care at all about this issue.					

Register:	FIQ_SEL_5			Address:	48014h	
Bits:	6dt0	Reset value:	00h	Attributes:	r	w
Description:	Selects one of the IRQs as input to FIQ processing. Important: The FIQ_SEL register have to be configured with different values (except 0) for each select register. The ICU does not take care at all about this issue.					

Register:	FIQ_SEL_6			Address:	48018h	
Bits:	6dt0	Reset value:	00h	Attributes:	r	w
Description:	Selects one of the IRQs as input to FIQ processing. Important: The FIQ_SEL register have to be configured with different values (except 0) for each select register. The ICU does not take care at all about this issue.					

Register:	FIQ_SEL_7			Address:	4801Ch	
Bits:	6dt0	Reset value:	00h	Attributes:	r	w
Description:	Selects one of the IRQs as input to FIQ processing. Important: The FIQ_SEL register have to be configured with different values (except 0) for each select register. The ICU does not take care at all about this issue.					

2.3.3 Host ICU (PerIF: Event Unit)

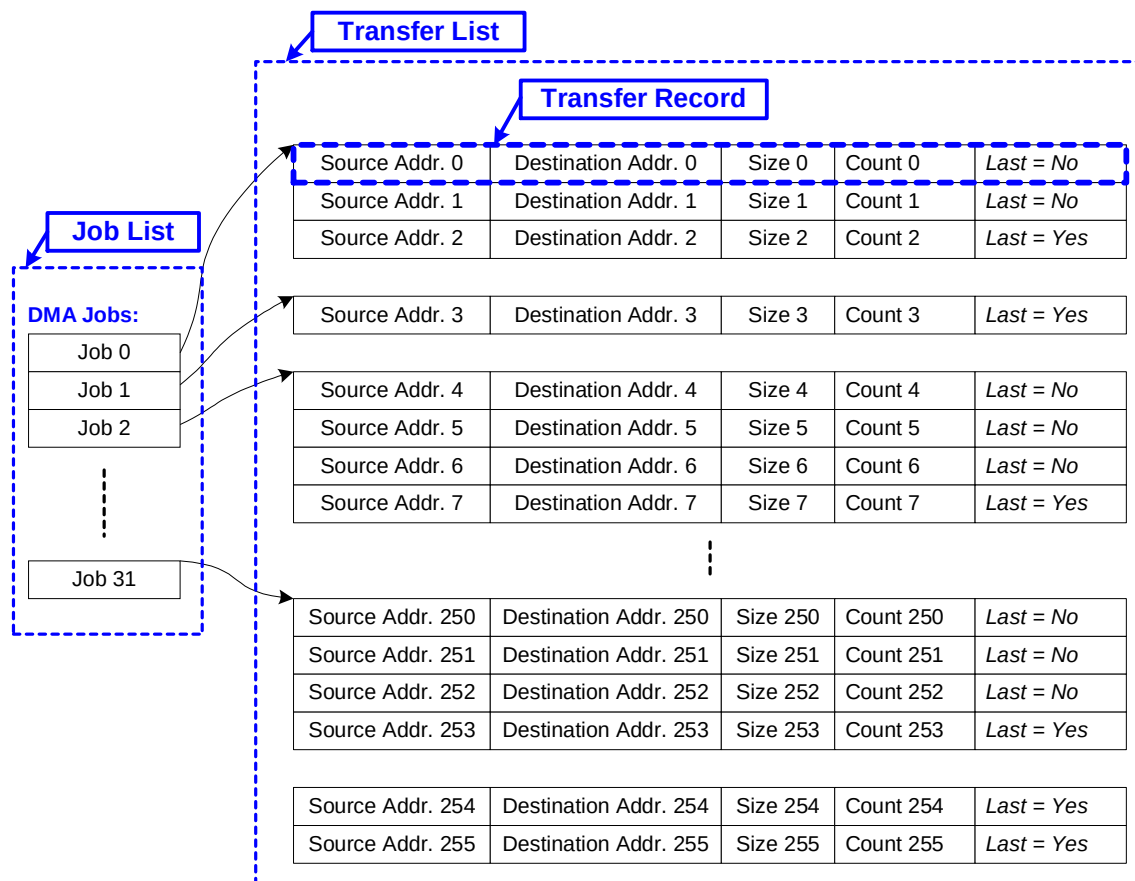
The Host Interrupt Controller is included in the Peripherie-IF (see chapter 7.2).

2.3.4 GDMA (Direct Memory Access)

2.3.4.1 GDMA Function Description

The GDMA controller is programmable to perform DMA jobs. A DMA job is a collection of single transfers which are organized in a transfer list. The transfer list contains a parameterizable maximum number of single transfers. In the GDMA, the parameter is set to allow maximally 256 single transfers. A single DMA transfer represents copying of a programmable count of data elements, with programmable element size, from source address to destination address. The number of data elements can be programmed within the range of 0 to 65535, thus the count of data elements can be set within the range of 1 to 65536 elements. A transfer record of the transfer list defines these parameters of the single DMA transfer. Each transfer record holds information, whether it is the last one in the job. The principle of the DMA jobs organization is shown in Figure 1.

The transfer list is saved in the DMA RAM, which is in an internal RAM. The Job List base address is saved in a GDMA register. The number of transfers per job is programmable; the maximum number of jobs, supported by the GDMA within ERTEC 200P is 32.



Transfer_lists.vsd

Figure 1: The principle of the DMA Job organization

2.3.4.1.1 AHB interfaces

The GDMA controller contains two AHB interfaces: The AHB Master Interface of the GDMA is used for data transfer from the DMA source address to the DMA destination address. This interface also serves for access to the DMA RAM.

The AHB Slave Interface of the GDMA is used as access to the GDMA registers. When the GDMA controller is configured for an internal DMA RAM, this interface also serves for access from the CPU or from the GDMA AHB Master to this RAM. As mentioned above, the DMA RAM is used for programming of the transfer list. The GDMA registers serve for programming and status monitoring of the GDMA controller.

2.3.4.1.2 Job priorities

Each job has its own programmable priority. The DMA job can be started either by HW or by SW. The started job with the highest priority begins to run (if other conditions, described later, are fulfilled). When a job is running and a job with a higher priority is started, the running job is interrupted and the job with higher priority begins to run. After the job with the higher priority is finished, the interrupted job will continue. A higher value of JOB_PRIO means a higher priority. If several jobs have the same value of JOB_PRIO, then the job priority is defined by the job number. In that case the smaller job number means a higher priority.

2.3.4.1.3 Details

2.3.4.1.3.1 Blocks

The GDMA controller consists of the following blocks:

- GDMA Controller Core – Controls DMA transfers via the AHB Master IF
- AHB Master Handler – HW block of AHB Master IF
- AHB Slave Handler – HW block of AHB Slave IF
- GDMA registers – the DMA Control registers serve for global configuration of the GDMA, the Job Control registers serve for job programming and the Status registers serve for monitoring of the DMA status. The address space of the GDMA registers is parameterizable, defined by the maximum number of DMA jobs. The word size of the registers is 32 bits.
- DMA RAM serves for transfer list programming and can be configured by a constant in VHDL code; either as an internal RAM or as an external RAM. The address space of the DMA RAM is programmable. It is defined by the DMA Control register, see below. The word size of the DMA RAM is 32 bits.
- HW Peripheral MUX – selects one of „n” job-starting input signals from the PROFINET-IP (3x Application Timer Block Modul), the Timer Unit and GPIO input signals. The input selection is programmable through the Job Control registers. The maximum number of job-starting input signals is parameterizable up to 64.

2.3.4.1.3.2 Interfaces

The GDMA controller contains the following interfaces:

- AHB Slave Interface – Serves for access to the GDMA registers as well as to the DMA RAM, if the GDMA controller is configured for an internal RAM.
- AHB Master Interface – Serves for transferring data via a DMA channel (read and write accesses). The AHB Master also is used to access the DMA RAM from the GDMA controller.
- Job-starting input signals – Start jobs by HW, they are connected to PROFINET-IP (3x Application Timer), Timer and GPIO input signals. Note that start of a job by HW is enabled by means of bit HW_JOB_START_EN of the Job Control register.
- DMA request by HW signals – For a job dedicated to data transfer from a peripheral, the HW DMA request signal must inform, if data is available on the peripheral. For a job dedicated to data transfer to the peripheral, the HW DMA request signal must inform, if the peripheral is ready to receive the data.

According to these needs, the HW DMA request signals of the GDMA controller are connected to the corresponding outputs of the peripheral devices. The number of DMA request signals is parameterizable and equals to the maximum number of jobs. Note that the ready-checking of the peripheral is enabled by means of bit HW_FLOW_EN of the Job Control register. For every HW DMA request signal input an HW DMA acknowledge output exists, which is asserted at the end of a DMA transfer, if this is enabled in the DMA Transfer Record and the bit HW_FLOW_EN in the job control register is set.

- DMA Interrupt Request (DMA_IRQ) output – If a job is finished or a DMA error occurs (and if the other conditions, described later, are fulfilled), then the GDMA controller generates a DMA interrupt request.
- Error Detection and Correction (EDC) interrupt request – these signals (GDMA-1B: 1Bit Error corrected, GDMA-2B: 2Bit Error recognized) are wired to the EDC_EVENT Register in the SCRB Register Block (see chap. 2.3.10.9.22). Additionally the interrupt 'EDC_Event' IRQ48 is set (see chap. 7.4.1). For the purpose of cleanup the EDC Event Register must be written with '0h'. By reading the EDC Event Register all Bits are cleared. After Reset an initialisation of the EDC-Bits is made. The conclusion of the initialisation can be read in the SCRB Register 'EDC_INIT_DONE' (see chap. 2.3.10.9.22).

These GDMA outputs are internally connected to the outputs of the MEM Wrapper of the internal DMA RAM. Note that this interrupt is used only when the GDMA controller is configured for an internal DMA RAM and the Memory Wrapper is applied.

2.3.4.1.3.3 Block diagram

The block diagram of the GDMA controller is shown in Figure 2.

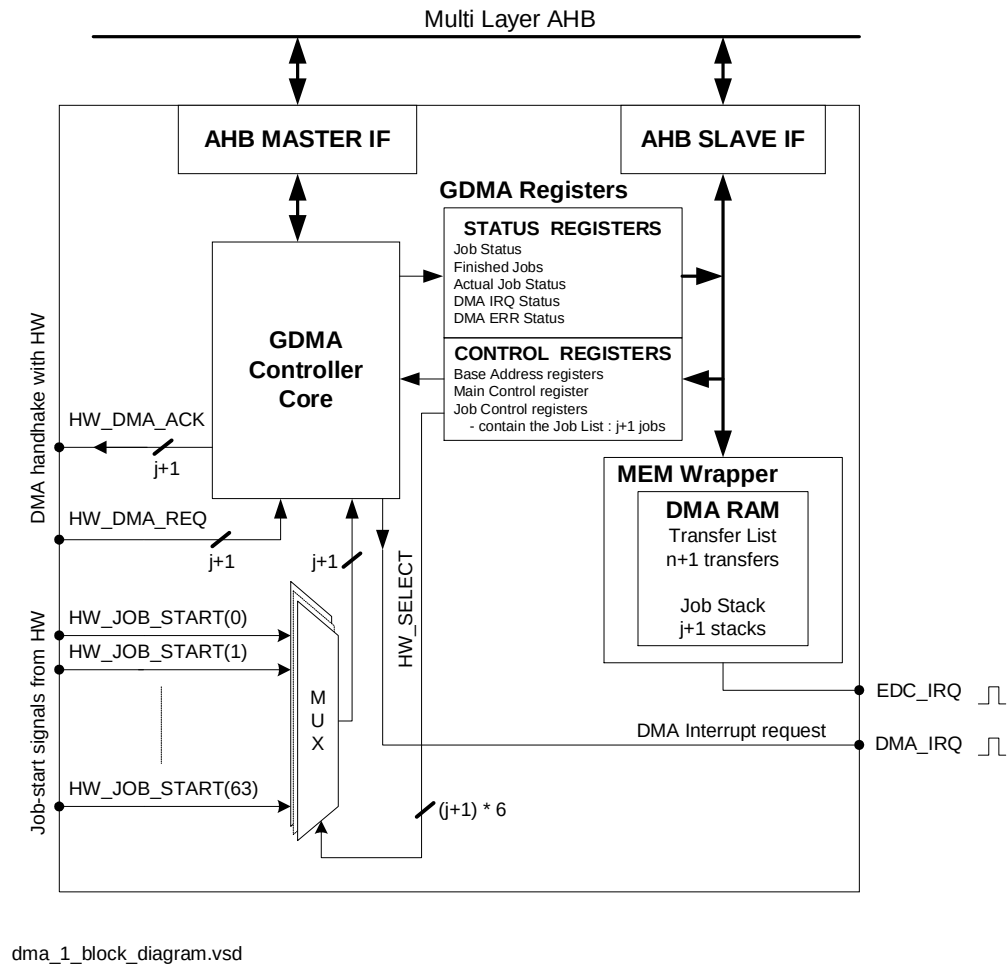


Figure 2: Block diagram of the GDMA controller

DMA jobs are programmed by means of GDMA Job Control registers and the transfer records of the DMA RAM, as illustrated in Figure 3.

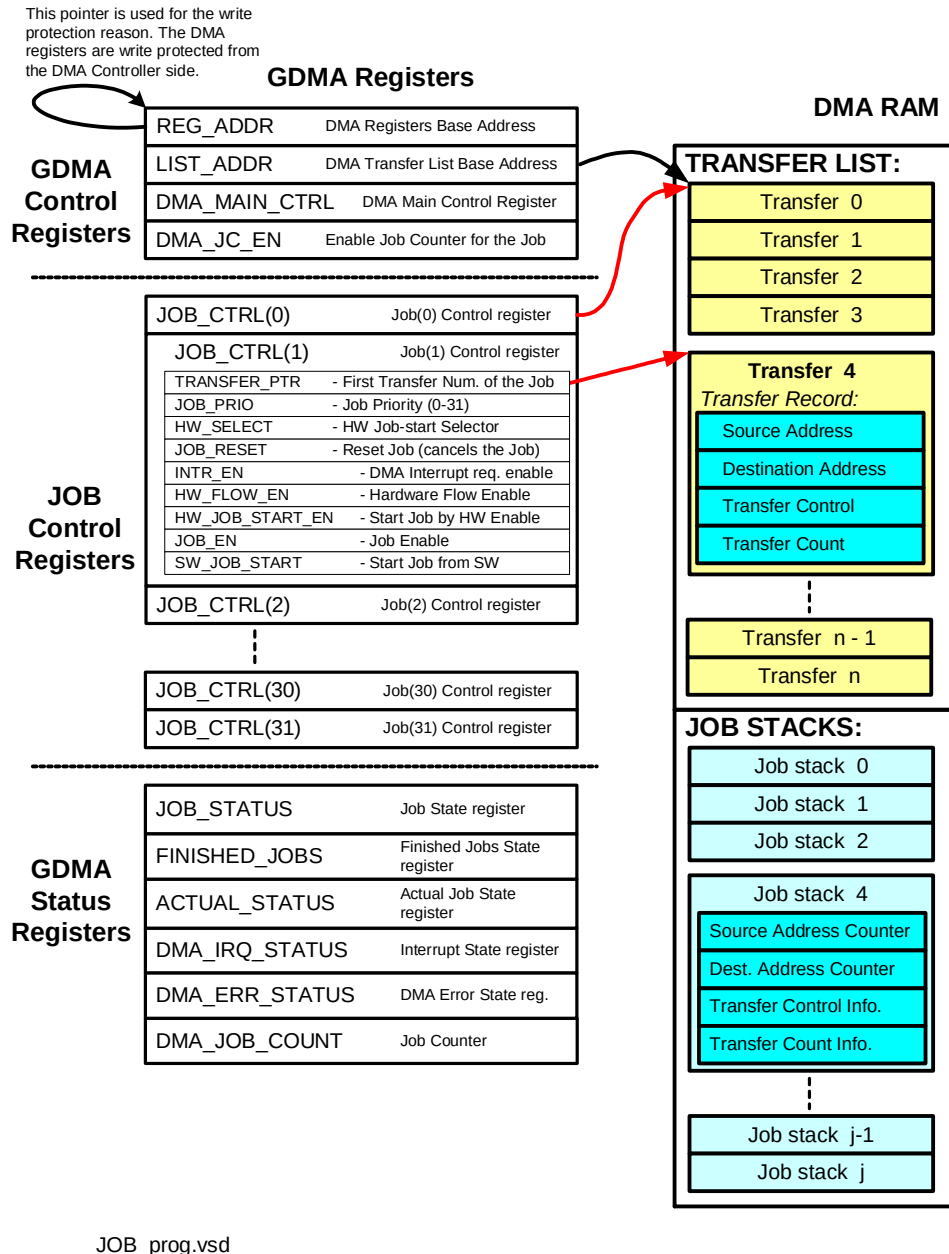


Figure 3: DMA jobs programming

Note: The Job stack is not a stack as software programmers are used to work with. It is a 32 byte status field for every job, where the information to continue an interrupted job, is stored. It can be read and written by software for debug purposes.

2.3.4.1.3.4 DMA Job Control registers

A Job Control register consists of the following fields:

- First Transfer Number of the Job (TRANSFER_PTR) is a pointer to the transfer number of the first transfer of the job. (The pointer is shown in Figure 3)
- Job Priority (JOB_PRIO) defines the priority of the job. A higher value of JOB_PRIO means higher priority. If two or more jobs have the same value of JOB_PRIO, then the job priority is defined by the job number. In such a case, the smaller job number means the higher priority. When e.g. the JOB_PRIO field has the same value for each job, the highest priority has the started job with the smallest job number.
- HW Job Start Selector (HW_SELECT) defines, which of „n“ HW job-starting signals is chosen to start the job by HW. Note that the maximum number of job-starting input signals is parameterizable up to 64. The HW parameter has no influence on the register width.
- Job reset (JOB_RESET) cancels the running job.
- Interrupt request generation enable (INTR_EN) enables GDMA controller to generate interrupt request when the job has finished.
- Hardware Triggered Flow Enable (HW_FLOW_EN) enables/disables the check whether the HW peripheral is ready.
- Start Job by HW enable (HW_JOB_START_EN).
- Enable Job (JOB_EN) enables the run of the Job or interrupts the running job.
- Start Job by SW (SW_JOB_START).

2.3.4.1.3.5 Transfer list

The DMA RAM's transfer record defines all parameters of a single DMA transfer and consists of four 32-bit words:

- “Source Address” defines the address of source data for the DMA transfer. This can be either a memory or a peripheral address.
- “Destination Address” defines the address of the destination for the DMA transfer. This can be either a memory or a peripheral address. The addresses of the GDMA registers and DMA RAM are protected against write accesses by the DMA transfers. The reason for this is as follows: When the GDMA controller is programmed incorrectly, the data in the GDMA registers and DMA RAM could be corrupted and subsequently the system might fail. When a wrong destination address is programmed, then the DMA destination address error is assumed. When at the same time bit “Error Interrupt Enable” (ERR_INT_EN) in the GDMA Main Control register is “1”, the relevant error

bit in the GDMA Error Interrupt State register is set to “1” and the GDMA interrupt request (DMA_IRQ) is generated.

- “Transfer Control” consists of three 2-bit fields: Source Address Mode, Destination Address Mode and Burst Mode.
 - The Source Address Mode and Destination Address Mode can be either “increment” or “hold”. Note that the second bit of this field serves as a reserve bit.
 - The Burst Mode holds the information about the AHB burst type which will be used, and can be Single, INCR4, INCR8 and INCR16 (see necessary selection under HW_DMA_REQ signals in chapter 0).
- “Transfer Count” consists of 4 fields: 1-bit flag “Last Transfer of the Job”, 1-bit flag “Enable DMA Acknowledge”, 2-bit field “Element Size” and 16-bit field “Transfer Count”.
 - Last Transfer of the Job - Indicates the last transfer record in a job (thus the transfers in a job are defined by pointer TRANSFER_PTR in the Job Control register and the bit “Last Transfer of the Job”). When no transfer in the Transfer List is marked as the last one, the transfers will be executed in an infinite loop.
 - Enable DMA Acknowledge – The output HW_DMA_ACK of a job is set ‘1’, if the input HW_DMA_REQ of this job is 1, and the bit HW_FLOW_EN of the job register is 1, and the current transfer is finished, and the bit DMA_ACK_EN in the Transfer Record is 1. The output HW_DMA_ACK of a job is reset ‘0’, if the input HW_DMA_REQ of this job is 0.
 - Element Size - Sets the size of elements to be copied in the transfer and can be 8, 16 or 32 bits. The number of bytes to transfer is (element size) x (transfer count)
 - Transfer count - Holds the information on how many elements have to be copied in this transfer. It can be programmed from 0 to 65535 elements, representing the range of 1 to 65536 elements.

2.3.4.1.3.6 DMA Control registers

The global SW configuration of the GDMA controller is located in the GDMA Control registers. They are organized in a set of four 32-bit registers: “GDMA Registers Base Address”, “DMA Transfer List Base Address”, “GDMA Main Control” and “Enable Job Counter for the Job”.

The DMA Control registers configure the address space of the GDMA registers and the transfer list as well as control the functions of the GDMA controller, as described below:

- GDMA Registers Base Address (GDMA_REG_ADDR) - Points to the first address of the GDMA registers. It is used only for comparison purposes. The DMA destination address is compared with the GDMA registers address space to protect the registers against undesirable write accesses by the GDMA controller, as will be described later.

- DMA Transfer List Base Address (GDMA_LIST_ADDR) - Points to the first address of the transfer list of the DMA RAM.
- GDMA Main Control register (GDMA_MAIN_CTRL) consists of five fields:
 1. Total Number of Transfers in Transfer List (LIST_SIZE) - Defines the address space of the transfer list. Address space of transfer list in DMA RAM is defined by its base address and by the total number of transfers in the transfer list.
 2. Software reset (SW_RESET) - The started jobs can be reset and the running or interrupted jobs cancelled when SW_RESET bit in the GDMA Main Control register is set.
 3. DMA Global Enable (DMA_EN) - Enables transfer activity of the GDMA controller. When DMA_EN bit is reset while a DMA job is in progress, this job is interrupted and no other started job can begin to run. When subsequently DMA_EN bit is set again, the interrupted job will continue.
 4. Reset Job Counter (JC_RESET) resets the GDMA_JOB_COUNT status register. See the description of the GDMA Status registers below.
 5. Error Interrupt Enable (ERR_INT_EN) - This bit enables generation of GDMA interrupt request (DMA_IRQ), when an error occurs in the GDMA controller. Four types of errors are defined (see below).
- Enable Job Counter for the Job (GDMA_JC_EN) is a 32-bit register, which selects the jobs whose time of activity shall be measured via the GDMA_JOB_COUNT status register. See the description of the GDMA status registers below.

2.3.4.1.3.7 Job control by SW or by HW

SW control of the job:

- Start of job by SW:
The job is started by SW when bit SW_JOB_START is set in the Job Control register. Start of DMA job is disabled, when bit SW_RESET in the GDMA Main Control register is set to “1” or when bit HW_JOB_START_EN in the Job Control register is set to “1”.
- Make a job running:
When one or more jobs are started, the started job with the highest priority begins to run (becomes active). The running job can be interrupted in three ways:
 1. When a job is running and a job with higher priority is started, the running job is interrupted and the job with higher priority begins to run. After the job with the higher priority is finished, the interrupted job will continue.

2. When bit JOB_EN in the Job Control register is reset from '1' to "0", a running job is interrupted and a started job with the next, smaller priority begins to run. After this bit is changed back to "1", the interrupted job will be reactivated. JOB_EN = 0 does not cancel a job. It makes the job only sleeping.
3. When bit DMA_EN is reset while a DMA job is running, this job is interrupted and no new job begins to run. When subsequently bit DMA_EN is set again, the interrupted job will continue. DMA_EN has the same function as JOB_EN, but for all jobs.

The running job and all interrupted jobs can be cancelled by setting the JOB_RESET bit in the Job Control register to "1".

- Job finished:
When a DMA job is completed, a "job finished" bit in the Finished Jobs State register (GDMA_FINISHED_JOBS) is set and the GDMA controller generates an interrupt request. This interrupt request generation can be enabled/disabled for each job using bit INTR_EN of the Job Control registers. When the GDMA interrupt is generated, a "job finished" bit in the Interrupt State register is set. This Interrupt State register is described in the following section "GDMA Status registers".

HW control of the job:

- Start of job by HW:
HW starts the job by rising edge of signal HW_JOB_START. This signal is selectable as one of „n" (max. 64) input signals using field HW_SELECT in the Job Control register and can be enabled/disabled by means of bit HW_JOB_START_EN in the GDMA Job Control register.
Start of a DMA job is disabled also when bit SW_RESET in the GDMA Main Control register is set to "1". When HW starts a job again, before the running job is finished and if at the same time bit "Error Interrupt Enable" (ERR_INT_EN) in the GDMA Main Control register is "1", the relevant error bit in the GDMA Error Interrupt State register is set to "1" and the interrupt request at the DMA_IRQ output of the GDMA controller is generated.
- Make a job running:
For the started job with the highest priority, dedicated to data transfer from a peripheral, the HW DMA request signal is inquired, whether the peripheral device is ready to send data. For the job dedicated to data transfer to a peripheral, the HW DMA request signal is inquired, whether the peripheral device is ready to receive data. If so, the job starts to run, otherwise the job is interrupted and the next job can start to run. When a job is running and the HW DMA request signal indicates that the peripheral is no longer ready to send/receive data, the job is also interrupted. When the peripheral is ready to send/receive data again, the job will continue. The check, whether the HW peripheral is ready, can be enabled/disabled by means of field HW_FLOW_EN of the Job Control

register. Furthermore, the job can be interrupted or cancelled just as in the case of the SW control of the job described above.

- Job finished:
The job is finished in the same way as in the case of the SW control of the job described above.

The job control is illustrated in Figure 4.

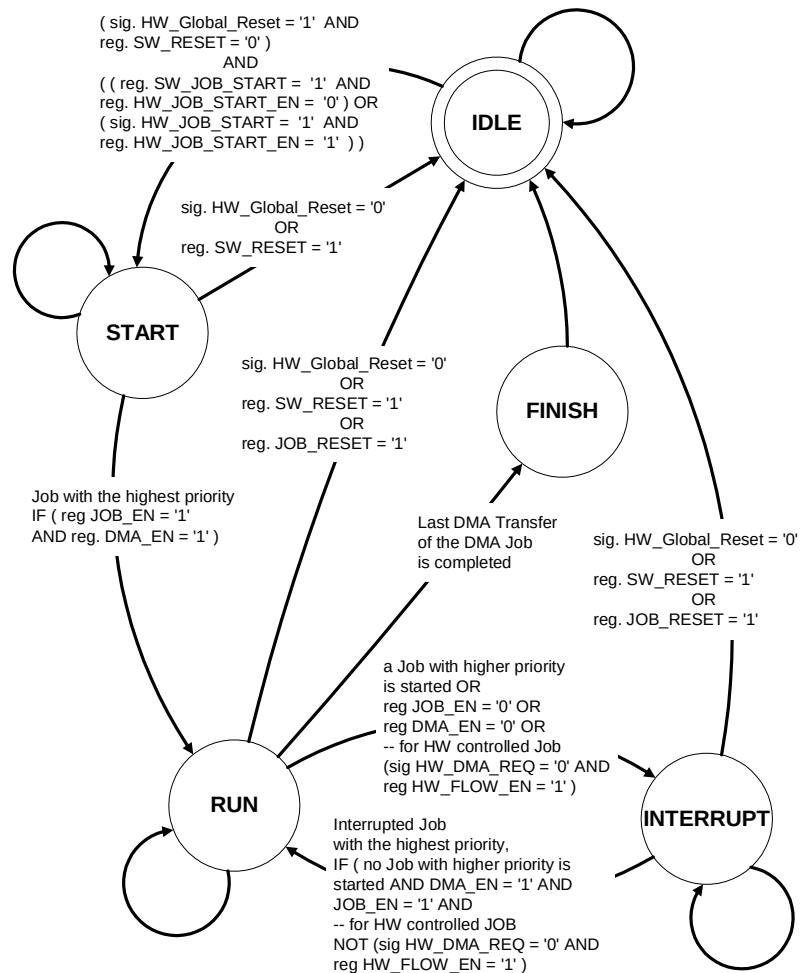


Figure 4: Finite state machine diagram of the job control algorithm

2.3.4.1.3.8 GDMA Status registers

For the monitoring of the GDMA controller via SW, important feedback information about the status of the GDMA is the status parameters, contained in the GDMA Status registers. Six status registers are available:

- Job State register (GDMA_JOB_STATUS) indicates via its 32 bits, which job is active (i.e. started, running or interrupted).
- Finished Jobs State register (GDMA_FINISHED_JOBS) contains one “job finished” bit for each job. This bit is set to “1” each time after the job is completed. Writing a “1” clears the bit.
- Actual Job State register (GDMA_ACTUAL_STATUS) contains 1-bit flag “Active Job Number Valid” (ACT_JOB_VAL) and 5-bit field “Active Job Number” (ACT_JOB). Active Job Number indicates the number of the running job. Bit “Active Job Number Valid” indicates whether the job number is a valid number, which represents active DMA transfers.
- GDMA Interrupt State register (GDMA_IRQ_STATUS) contains a 1-bit “job finished” bit for each job. This bit is set when a job is finished and an interrupt request (DMA_IRQ) is generated. Writing a “1” clears the bit.
- Error State register (GDMA_ERR_STATUS) contains 32 error bits, one bit for each type of error. The error bit is set to “1” when the corresponding type of error occurs and when the ERR_INT_EN bit in the GDMA Main Control register is set to “1”. Writing a “1” clears the error bit. If an error bit is set to “1”, then the interrupt request (DMA_IRQ) is generated.
- Job counter register (GDMA_JOB_COUNT) counts clock cycles for a selected running job. Selection of the jobs for which the clocks shall be counted is done with 32-bit register Enable Job Counter for the Job (GDMA_JC_EN), one of the GDMA Control registers. Reset of the job counter is done by bit “Reset Job Counter” (JC_RESET) of the GDMA Main Control register (GDMA_MAIN_CTRL).

2.3.4.1.4 Usage

Usage of the GDMA controller is described below:

After reset, the DMA channels must be initialized. This is carried out in such a way that jobs are defined in the Job Control registers and the transfer list is written to DMA RAM.

2.3.4.1.4.1 First step

First, "GDMA Registers Base Address" (GDMA_REG_ADR), "DMA Transfer List Base Address" (GDMA_LIST_ADDR) and "Total Number of Transfers in Transfer List" (LIST_SIZE) must be defined and the bit "Global Enable" (DMA_EN) must be reset in the GDMA Control registers.

In the Job Control register, following fields must be defined:

- First Transfer Number of the job (TRANSFER_PTR)
- Job Priority (JOB_PRIO)
- HW Job Start Selector (HW_SELECT)
- Job Reset (JOB_RESET) must remain disabled (set to "0" after reset) if running of the job is required.
- Interrupt Request Generation Enable (INTR_EN)
- Hardware Triggered Flow Enable (HW_FLOW_EN)
- HW Job start enable (HW_JOB_START_EN) In the first step this bit must remain disabled (set to "0" after reset). The job can be started by HW after the second step is finished (transfer list must be defined).
- Enable Job (JOB_EN) must be set to '1' (set to "0" after reset) if running of the job is required.
- Start Job by SW (SW_JOB_START) In the first step this bit must remain disabled (set to "0" after reset). The job can be started by SW after the second step is finished (transfer list must be defined).

2.3.4.1.4.2 Second step

Second step is to set up the transfer records in the transfer list. The transfer record defines all parameters of a single DMA transfer and consists of four 32-bit words:

- Source Address
- Destination Address
- Transfer Control
 - consists of three 2-bit fields: Source Address Mode, Destination Address Mode and Burst Mode, as described above
- Transfer Count

- consists of 4 fields: 1-bit flag “Last Transfer of the Job”, 1-bit flag “Enable DMA Acknowledge”, 2-bit field “Element Size” and 16-bit field “Transfer Count”, as described above

2.3.4.1.4.3 Third step

The bit “Global Enable” (DMA_EN) must be set in the GDMA Control register.

2.3.4.1.5 Result

The GDMA controller is now prepared to receive job-starting requests.

2.3.4.1.5.1 DMA job started by SW or by HW

The DMA job can be started either by SW or HW:

Start of the job by SW

- Start of a DMA job by SW is realized through Job Control Register. Logical “1”, written to the SW_JOB_START bit of the Job Control Register, starts the corresponding job. Start of the DMA job is disabled, when bit SW_RESET in the GDMA Main Control register is set to “1” or when bit HW_JOB_START_EN in the Job Control register is set to “1”. When one or more jobs are started, the job with the highest priority begins to run.

Start of the job by HW

- Start of a DMA job by HW is realized by a rising edge of signal HW_JOB_START, when bit HW_JOB_START_EN in the Job Control register is set. The HW_JOB_START signal is selectable as one of „n” (max. 64) input signals, using the HW_SELECT field in the Job Control register. SW can disable the start of the DMA job as described in the above section “Start of the job by SW”.

2.3.4.1.5.2 DMA job controlled by SW or by HW

Control of the job by SW

A started job begins to run when the currently running job and other started jobs have lower priority than this job, or when no other job is running or started. When a job is running and a DMA request for a job with higher priority occurs, the current job is interrupted and the new job begins to run. After finishing of this job, the interrupted job will continue.

Additionally, jobs can be reset, interrupted or cancelled as follows:

- When bit DMA_EN in the GDMA Main Control register is reset while a DMA job is running, the job is interrupted and no new job begins to run. When subsequently bit DMA_EN is set again, the interrupted job will continue.
- The started jobs can be reset and the running job as well as all interrupted ones can be cancelled when bit SW_RESET in GDMA Main Control register is set.

- The running job can be interrupted when the job-enabling bit (JOB_EN) of the Job Control register is reset from '1' to '0'.
- The running and all interrupted jobs can be cancelled, when the job-resetting bit (JOB_RESET) in the Job Control register is set.

Control of the job by HW

The running of a DMA job dedicated to transfers from HW peripheral to memory or from memory to HW peripheral, is controlled by HW through the use of the input signals "HW Job Start" and "HW DMA request". See the description above.

Furthermore, also the job controlled by HW can be interrupted and cancelled via SW, just as in the case of the SW control of the job, described above.

2.3.4.1.5.3 Job finished

When a DMA job is finished, the GDMA controller generates a DMA interrupt request. This interrupt request generation can be enabled/disabled for each job, using bit INTR_EN of the Job Control registers. When the DMA interrupt is generated, a "job finished" bit in the Interrupt State register is set.

2.3.4.1.5.4 Monitoring of GDMA Controller Status

For monitoring of the GDMA controller via SW, important feedback information about the status of the GDMA is indicated by the DMA_IRQ interrupt request signal and by the GDMA Status Register, see above.

2.3.4.1.6 Memory

The GDMA RAM is used to store the DMA transfer list and the job stack. It can be used either as an internal GDMA RAM or as an external RAM. This feature is HW configurable by means of a constant in the VHDL code.

The external RAM is accessible from the GDMA through its AHB Master interface. The address space of the GDMA RAM is configurable by means of GDMA Control registers (LIST_ADDR and LIST_SIZE). Thus the GDMA controller can use any RAM that is accessible via the AHB bus.

The internal GDMA RAM is accessible from the CPU through the AHB Slave interface of the GDMA only with word accesses (byte and halfword write accesses not possible). From the GDMA controller is accessible through its AHB Master interface. Note that the datapath from the GDMA controller core to the GDMA RAM is: GDMA controller core -> AHB Master of the GDMA -> Multi-layer AHB Bus -> AHB Slave of the GDMA -> RAM Wrapper -> RAM. Thus the access algorithm is the same for both GDMA RAM configurations. The address space of the internal GDMA RAM is also configurable by means of GDMA Control registers (LIST_ADDR and LIST_SIZE).

The implementation of the GDMA for the ERTEC 200P the configuration with an internal GDMA RAM is the same as in Triton2. The GDMA RAM has size of $((n+1) \times 16 \text{ Byte}) + ((j+1) \times 16 \text{ Byte})$, where "n" is the index of the last transfer from the transfer list and "j" is the index of the last job stack. The configuration for ERTEC 200P is $n = 255$ and $J = 31$.

For this configuration the internal GDMA SRAM is organized in 1152 x 32 Bit.

The GDMA RAM has additionally EDC bits (7 Bit for a 32Bit word, 1Bit error correctable, 2Bit error recognizable). If an EDC error is detected, in the SCRB Register 'EDC_EVENT' (see chap. 2.3.10.9.22) the appropriate reason is stored (GDMA-1B: 1Bit-Error corrected or GDMA-2B: 2Bit-Error recognized) and the Interrupt 'EDC_Event' IRQ48 is generated (see chap. 2.3.2.14). The EDC Event Register is reseted by writing the register with '0h'. After reset the initialisation of the EDC-Bits isn't done by hardware. This must be done by software. After this initialisation the SW must finally set 'GDMA_INIT_DONE = 1b' in the SCRB-Register 'EDC_INIT_DONE' (see chap. 2.3.10.9.22).

2.3.4.1.7 Interrupts

The DMA controller has two interrupt request outputs. These signals are high active pulse. The length of the interrupt request pulse is at least 2 and at most 5 AHB clock cycles.

The GDMA controller incorporates no local interrupt controller of its own.

The two interrupt requests are:

DMA Interrupt Request (DMA_IRQ) - generated in the following cases:

- Job is finished and bit "Interrupt Request Generation Enable" (INTR_EN) in the Job Control register is set to "1". When Interrupt request is generated, a "job finished" bit is set in the Interrupt State register (GDMA_IRQ_STATUS). This status register must be read by the interrupt controller to figure out which job has caused the interrupt. The register is cleared by writing a "1" to the related bit position.
- A monitored error occurs at the DMA controller and bit "Error Interrupt Enable" (ERR_INT_EN) in the GDMA Main Control register is set to "1". In this case the relevant error bit in the GDMA Error Interrupt State register is set to "1". This status register must be read by the interrupt controller to figure out which type of error has caused the interrupt. The register is cleared by writing a "1" to the related bit position.

The monitored types of error are:

- DMA Destination Address error - Assumed when a wrong destination address, pointing to the DMA registers or the DMA RAM, is programmed. The relevant error bit in the GDMA_ERR_IRQ_STATUS register is ERR_DST_ADDR.
- AHB Master Interface error - Assumed when an error response occurs at the AHB Master Interface. The relevant error bit in the GDMA_ERR_IRQ_STATUS register is ERR_AHB.
- HW Job Start error - Assumed when HW starts a job again, before it is finished. The relevant error bit in the GDMA_ERR_IRQ_STATUS register is ERR_JOB_START.
- Not allowed write access to the AHB Slave - Assumed when a different transfer size other than Word is used during a write access. The relevant error bit in the GDMA_ERR_IRQ_STATUS register is ERR_AHBSLV_WRITE.

2.3.4.2 ERTEC 200P GDMA Use Cases

2.3.4.2.1 Using the Job Start and HW_REQ Signal List

There are 2 ways to implement a HW-triggered DMA transfer.

2.3.4.2.1.1 HW Trigger with HW_JOB_START Input Signals (Edge-Sensitive)

One of the 64 possible HW_JOB_START trigger inputs can be selected in each GDMA_JOB_CTL register with the HW_SELECT bit field. This means that there is **no** fixed assignment of JOB trigger signals (HW_JOBx_START) to JOB-CTL register number (GDMA_JOBx_CTL register).

The typical applications are free-running time triggers (cyclic pulse upon zero run) that trigger cyclic DMA transfers. The **HW_FLOW_EN** bit in the GDMA_JOBx_CTL register **must be set to 0**.

Port	Source	Active level	Description
HW_JOB_START (starts the job in HW by rising edge):			
HW_JOB_START_0	PNPLL_OUT11 (PN-IP)	high	coming from PNPLL in PN-IP
HW_JOB_START_1	PNPLL_OUT12 (PN-IP)	high	coming from PNPLL in PN-IP
HW_JOB_START_2	PNPLL_OUT13 (PN-IP)	high	coming from PNPLL in PN-IP
HW_JOB_START_3	PNPLL_OUT14 (PN-IP)	high	coming from PNPLL in PN-IP
HW_JOB_START_4	PNPLL_OUT15 (PN-IP)	high	coming from PNPLL in PN-IP
HW_JOB_START_5	PNPLL_OUT16 (PN-IP)	high	coming from PNPLL in PN-IP
HW_JOB_START_6	PNPLL_OUT17 (PN-IP)	high	coming from PNPLL in PN-IP
HW_JOB_START_7	PNPLL_OUT18 (PN-IP)	high	coming from PNPLL in PN-IP
HW_JOB_START_8	PNPLL_OUT19 (PN-IP)	high	coming from PNPLL in PN-IP
HW_JOB_START_9	PNPLL_OUT20 (PN-IP)	high	coming from PNPLL in PN-IP
HW_JOB_START_10	TIM_OUT0 (Timer Unit)	high	coming from Timer 0 in the Timer Unit
HW_JOB_START_11	TIM_OUT1 (Timer Unit)	high	coming from Timer 1 in the Timer Unit
HW_JOB_START_12	TIM_OUT2 (Timer Unit)	high	coming from Timer2 in the Timer Unit
HW_JOB_START_13	TIM_OUT3 (Timer Unit)	high	coming from Timer 3 in the Timer Unit
HW_JOB_START_14	TIM_OUT4 (Timer Unit)	high	coming from Timer 4 in the Timer Unit
HW_JOB_START_15	TIM_OUT5 (Timer Unit)	high	coming from Timer 5 in the Timer Unit
HW_JOB_START_16	Reserved		
HW_JOB_START_17	Reserved		
HW_JOB_START_18	Reserved		
HW_JOB_START_19	Reserved		
HW_JOB_START_20...63	Not used for ERTEC 200P; tied to inactive values		

Table 11: DMA, Table of HW_JOB_START_Signals

Not all HW_JOB_START signals are used in the ERTEC 200P. The input signals for the HW_JOB_START inputs that are not used are set to inactive level. The GPIO inputs for GDMA (GPIO3...GPIO0) must be synchronized in the I-filter with the system clock and can be inverted/not inverted with GPIO_POLSEL (see chapter 2.3.10.9.11).

HW Flow Control with REQ and DMA_ACK Handshake (Level-Sensitive)

The assignment of GDMA job number to HW request input/HW acknowledge output is fixed, i.e. job number 4 is assigned to the ports HW_DMA_REQ4 / HW_DMA_ACK4.

In these DMA transfers, the transfer is started (and controlled) by request signals and controlled by the DMA DMA_ACK signal. If the request signal is active at 1, the DMA transfers data until the transfer counter switches to 0 or the request signal becomes inactive.

As an acknowledgement signal, the GDMA controller returns a DMA_ACK pulse at the end of the transfer (transfer counter = 0) if bit 30 (EN_DMA_ACK) = 1 in the transfer record.

The **request** signal must also have the **state 1** (active level) for the **ACK** signal to be **output**. The **HW_FLOW_EN** bit in the *GDMA_JOBx_CTL* register **must be set to 1**.

The typical use cases are GDMA transfers between serial devices (SPI, UARTs with FIFO control)

In this mode, the programming rules for the GDMA and the connected IP (SPI, UART) must be followed. The GDMA IP connection only works if these programming rules are followed (see page 70, MC_GDMA_V1.0.pdf) /55/.

The GDMA job always terminates when its transfer counter has counted to 0 (irrespective of the current state of the request signal).

The job control register must then be written again to repeat the transfers.

Use of request signals in the ERTEC 200P:

HW_DMA_REQ (controls DMA transfers to / from serial peripherals):			
HW_DMA_REQ_0	UART1_TX-FIFO half-full_of_less	high	UART1 Tx-FIFO not half full (set GDMA to INCR8 mode)
HW_DMA_REQ_1	UART1_RX-FIFO not empty	high	UART1 Rx-FIFO not empty (set GDMA to single mode)
HW_DMA_REQ_2	UART2_TX-FIFO half-full_of_less	high	UART2 Tx-FIFO not half full (set GDMA to INCR8 mode)
HW_DMA_REQ_3	UART2_RX-FIFO not empty	high	UART2 Rx-FIFO not empty (set GDMA to single mode)
HW_DMA_REQ_4	UART3_TX-FIFO half-full_of_less	high	UART3 Tx-FIFO not half full (set GDMA to INCR8 mode)
HW_DMA_REQ_5	UART3_RX-FIFO not empty	high	UART3 Rx-FIFO not empty (set GDMA to single mode)
HW_DMA_REQ_6	UART4_TX-FIFO half-full_of_less	high	UART4 Tx-FIFO not half full (set GDMA to INCR8 mode)
HW_DMA_REQ_7	UART4_RX-FIFO not empty	high	UART4 Rx-FIFO not empty (set GDMA to single mode)
HW_DMA_REQ_8	SPI1_SSPRXDMA	high	SPI1 Rx-FIFO not empty - DMA request (set GDMA to single mode)
HW_DMA_REQ_9	SPI1_SSPTXINTR	high	SPI1 Tx-FIFO not half full - DMA request (the SSPTXINTR interrupt (Transmit FIFO is half full or less) must be enabled; set GDMA to INCR4 mode)
HW_DMA_REQ_10	SPI1_SSPTX_Delayed_Request	high	SPI1 transmit delayed – DMA request (mapped in the GDMA Shell) Expiry of timer 4 when 'SPI1 Tx-FIFO not half full' sets this bit; set GDMA to single byte transfer. DMA_ACK is used resets this bit.
HW_DMA_REQ_11	SPI2_SSPRXDMA	high	SPI2 Rx-FIFO not empty - DMA request (set GDMA to single mode)
HW_DMA_REQ_12	SPI2_SSPTXINTR	high	SPI2 Tx-FIFO not half full - DMA request (the SSPTXINTR interrupt (Transmit FIFO is half full or less) must be enabled; set GDMA to INCR4 mode)

HW_DMA_REQ_13	SPI2_SSPTX_Delayed_Request	high	SPI2 transmit delayed – DMA request (mapped in the GDMA Shell) Expiry of timer 5 when 'SPI2 Tx-FIFO not half full' sets this bit; set GDMA to single byte transfer. DMA_ACK is used resets this bit.
HW_DMA_REQ_15...31	Not used for ERTEC 200P; tied to inactive values		

Table 12: DMA, Table of HW_DMA_REQ-Signals
DMA with SPI Interface

Description of DMA functionality with SPI-IP (the ARM IP *SSPMS PL021* is used).

2.3.4.2.1.2 Receive Data

The SPI is switched to slave mode by the SW parameter assignment. The SW then starts the DMA job.

The SPI interface receives data over the serial interface. A DMA request signal becomes active as soon as at least once character is available (**SPI_RNE**: Receive FIFO not empty). The SPI FIFO has a depth of 8 words, each 16 bits wide. The DMA request signal activates the DMA job. If the DMA job has the highest priority, it starts to run (job is running). The data are then retrieved by the DMA and buffered in PN RAM, for example. The number of items of data to be processed is specified in the transfer list. After all SPI data have been received and forwarded, the DMA outputs a DMA-ACK signal and, if enabled, a DMA interrupt for the processor is generated. The DMA-ACK signal cannot evaluate the SPI-IP. A FIFO overflow of the SPI receive FIFO is not, however, to be expected as the DMA retrieve the data quickly enough. This does, of course, depend on the baud rate of the SPI-IP. In this case, the receive baud rate is a maximum of 10.42 Mb/s in slave mode ($1/12 \cdot$ APB clock frequency of 125 MHz). The DMA operates at the APB bus at 125 MHz with a maximum bus width of 32 bits. In this case, the DMA will read the data much faster than it can be provided by the SPI-IP.

Configuring the JOB_CTL Register and Transfer_Record

Receiving data of different lengths from **2 to 256 bytes**. Controlling DMA traffic with SPI DMA-REQ. The data is read out by the receive FIFO of the SPI and written to the PB-RAM, for example. The data bus width of the SPI is 16 bits. JOB_CTL register0 is linked to DMA_ACK0, which is connected to SPI_RFNE (Receive FIFO not empty).

The job is started by the SW (job started) and starts to run when the DMA request signals are queried (job is running); that is why the SW_JOB_START bit is set to 1. The job can also be started by the HW (HW_JOB_START_EN + HW_JOB_START signal). However, start with the SW is the more useful option as the SW needs to send queries and make settings on the SPI, and processing is not cyclic.

Entries required in the Job Control Register 0 (SPI receive data):

TRANSFER_PTR =	0-255	First Transfer Number of Job in Transfer List (0-255)
JOB_PRIO =	0-31	Job Priority(0-31)
HW_SELECT =	0	HW Job Start Selector (selects one from 64 inputs)

JOB_RESET =	0	Reset Job (cancel running job) 0 = don't care, 1 =
Cancel INTR_EN =	0	Interrupt Request Generation Enable 0 = disabled
HW_FLOW_EN =	1	HW triggered Flow enabled
HW_JOB_START_EN =	0	Start Job by Hardware hier disabled
JOB_EN =	1	Job Enable
SW_JOB_START	1	Start Job from Software enabled

Parameter assignment of DMA Transfer Record for SPI operation (receive data):

Source Address	=	32-bit source address
Destination Address	=	32-bit destination address
Transfer Control	=	SOURCE_AMODE = hold address, DEST_AMODE = incr address, BURST_MODE = Single
Transfer Count	=	Last Transfer, Enable DMA_ACK, ESIZE = 16, TC = 007Fh (128 transfers, 16 bit)

The address remains the same in source address mode as the data from the SPI receive FIFO are always read at the same address. For writing DMA data to PN-RAM, however, incremental address mode is set. The transfer counter (TC) is set to the number of transfers to be carried out, for example 128. If the number changes, Transfer_Record must be reconfigured. If the TC reaches zero, all data should have been transferred and the job is complete.

Only one transfer record is likely to be needed.

2.3.4.2.1.3 Transmit Data

The SPI has been switched to master mode by the SW parameter assignment. The SW then starts the DMA job.

The job is started by the SW (Job started) and starts to run when the DMA request signals are queried (job is running); that is why the SW_JOB_START bit is set to 1.

The data are written to the transmit FIFO of the SPI-IP by the DMA as soon as the DMA transmit request signal becomes active (TFE=Transmit-FIFO empty). To avoid a FIFO overflow, a maximum of 4 halfwords are written to the 8-entry deep FIFO (INCR4). Transfer is in an 8-byte INCR4 burst. As soon as the first data item is written to the FIFO, the TFE signal becomes inactive and transfer is interrupted after the 8 bytes. As soon as the data have been sent, the signal is activated again and the next 8 bytes can be transferred.

Configuring the JOB_CTL Register and Transfer_Record

Sending data of different lengths from **2 to 256 bytes**. Controlling DMA traffic with SPI DMA-REQ. Data are read by the DMA from PN-RAM, for example, and written to the transmit FIFO of the SPI. The data bus width of the SPI is 16 bits. JOB_CTL register1 is linked to DMA_ACK1, which is connected to SPI_TFE (Transmit Buffer empty).

Entries required in Job Control Register 1 (SPI transmit data):

TRANSFER_PTR =	0-255	First Transfer Number of Job in Transfer List (0-255)
JOB_PRIO =	0-31	Job Priority(0-31)
HW_SELECT =	0	HW Job Start Selector (selects one from 64 inputs)

JOB_RESET =	0	Reset Job (cancel running job) 0 = don't care, 1 = Cancel
INTR_EN =	0	Interrupt Request Generation Enable 0 = disabled
HW_FLOW_EN =	1	HW triggered Flow enabled
HW_JOB_START_EN =	0	Start Job by Hardware disabled
JOB_EN =	1	Job Enable
SW_JOB_START	1	Start Job from Software enabled

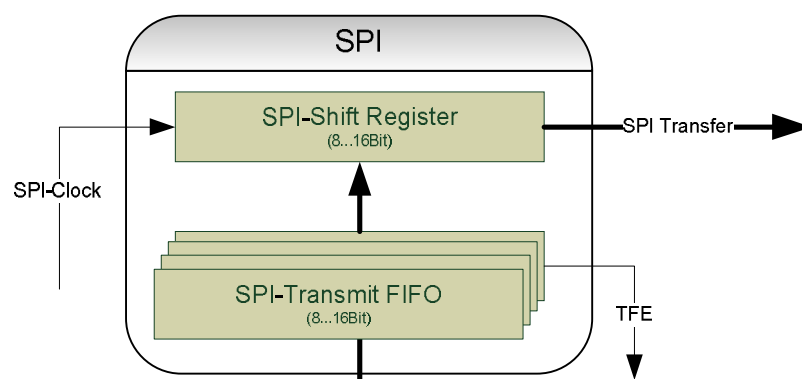
Parameter assignment of DMA Transfer Record for SPI operation (send data):

Source Address	=	32-bit source address
Destination Address	=	32-bit destination address
Transfer Control	=	SOURCE_AMODE = incr. address, DEST_AMODE = hold address, BURST_MODE = 8-byte burst
Transfer Count	=	Last Transfer, Enable DMA_ACK, ESIZE = 16, TC = 0FFh (e.g. 256 transfers, 16-bit)

In source address mode, the address is incremented, as reading is from PN-RAM. Send data is always written to the same address of the SPI transmit FIFO (hold address). The transfer counter (TC) is set to the number of bytes to be transferred. If the number changes, Transfer_Record must be reconfigured. If the TC reaches zero, all data should have been transferred and the job is complete.

Generating the GDMA interrupt in line with the user data to be transferred

The user data to be transferred (8...16 bits) are entered in the SPI-IP (the ARM IP *SSPMS PL021* is used). The APB enters them in the SPI transmit FIFO, from where they are transferred to the SPI shift register and then over the SPI interface.



The SPI-IP returns TFE (Transmit Fifo Empty) to the GDMA if there is no user data in the SPI transmit FIFO.

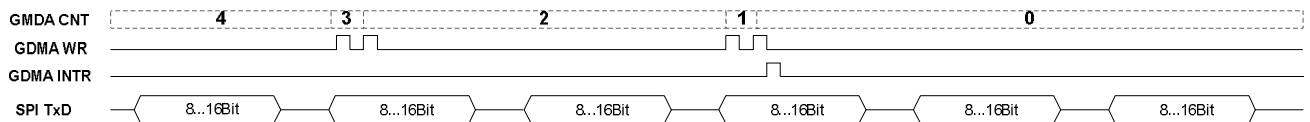
Writing GDMA user data to the SPI transmit FIFO is controlled with TFE (Transmit FIFO Empty). The GDMA writes data to the SPI transmit FIFO until TFE is canceled by the SPI-IP. This is the case if there is at least one user data entry (8...16 bits) in the SPI transmit FIFO.

As a result of the posted response at the AHB interface of the GDMA, the GDMA interrupt generation behavior for SPI data transfer varies depending on the transfer mode at the AHB (single / burst mode).

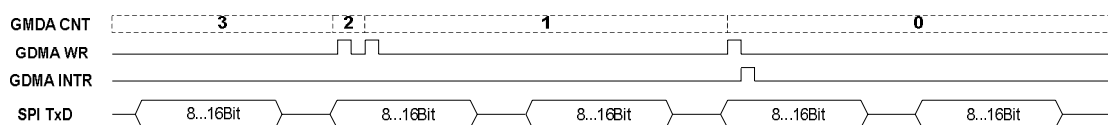
- In **single transfers**, the first instance of write access to the SPI-IP is acknowledged at the GDMA immediately by the ML-AHB (posted behavior) even before it is entered in the SPI transmit FIFO. As TFE (Transmit FIFO Empty) is not yet reset, another instance of write access to the SPI-IP is triggered and implemented by the GDMA despite single transfer.

Consequence: Despite single mode, up to 2 user data transfers to the SPI-IP can be carried out by the GDMA (behavior in this case corresponds to an INCR2 burst). This means that:

- o when the number of data transfers is even, the GDMA interrupt always becomes active 3 SPI transfers (8...16 bits) before the end of the SPI transfer
→ 1 SPI transfer in the SPI shift register + 2 SPI transfers in the SPI transmit FIFO



- o when the number of data transfers is odd, the GDMA interrupt always becomes active 2 SPI transfers (8...16 bits) before the end of the SPI transfer
→ 1 SPI transfer in the SPI shift register + 1 SPI transfer in the SPI transmit FIFO



- In **burst transfers**, complete burst access (INCR, INCR4, INCR8, ...) is always to the SPI-IP. The user must therefore ensure that the number of items of user data entered in the SPI transmit FIFO by the GDMA does not result in a FIFO overflow. As the first user data transfer in burst access resets TFE (Transmit FIFO Empty), this information is available at the GDMA at the end of burst access and ensures that there is no further GDMA write access

Consequence: In burst mode, the GDMA can carry out user data transfers to the SPI-IP of up to the AHB burst length. This makes the GDMA interrupt active up to "AHB burst length + 1" before the end of the SPI transfer.

The GDMA interrupt can be generated independently of the user data to be transferred if the final user data transfer to be carried out with the SPI is processed with its own transfer element in the GDMA job. As in this case the GDMA must first read the next transfer element over the AHB, TFE (Transmit Fifo Empty) is reset by the SPI-IP until the time of the actual single transfer so that the GDMA does not transmit this last instance of write access to the SPI-IP until TFE (Transmit Fifo Empty) is set again, which is always the case when there are no more data in the SPI transmit FIFO. The GDMA interrupt therefore always becomes active 2 SPI transfers (8...16 bits) before the end of the SPI transfer.

2.3.4.2.2 DMA with UART Interface

As the DMA request signals available at UART1...4 are not suitable for GDMA operation, internal status signals are used from the FIFO controller and sent to the GDMA as DMA request signals. The signals for controlling the transmitter and receiver separately over one GDMA channel each are as follows:

- **UART_RX-FIFO not empty:**
The RX FIFO contains at least one character and is not empty. As the UART cannot know when the last character (byte) was read in, the GDMA must operate in SINGLE-byte access mode (AHB). 1 to 65536 characters can therefore be transferred per DMA request; the job consists of a transfer entry.
- **UART_TX-FIFO half-full or less:**
The TX FIFO is at least half empty and therefore has space for 8 characters (bytes). The GDMA must not operate in more than INCR8-byte access mode (AHB) to ensure that there is no FIFO overrun. When GDMA write access is set to HOLD Address, the bytes are **always** transferred to the TX FIFO with SINGLE-byte access irrespective of the access mode. INCR8-byte access mode is therefore only effective for reading data (e.g. from RAM). If the transfer length is not modulo 8, the remaining characters are read by the GDMA controller with the INCR burst byte (undefined length) and transferred to TX FIFO with SINGLE-byte. 1 to 65536 characters can therefore be transferred per DMA request; the job consists of a transfer entry.

Note: Unlike with frame-oriented transfer over SPI, the GDMA cannot wait for TX FIFO Empty (TFE) to continue writing the TX FIFO: because of GDMA latency, this could lead to a TX FIFO underrun and the UART serial data stream could not be maintained. It is however possible with a GDMA-SPI interconnect as there is a longer pause between the SPI frames to be transmitted.

2.3.4.3 GDMA-IP Bugs

- **JOB_Reset:**
JOB_RESET for the last job (ERTEC 200P: JOB31) does not work reliably. With JOB31, JOB_RESET does not delete all information for the last job. This means that the information that the JOB has been interrupted and needs to be continued in the JOB STACK has not been deleted either. A new JOB configuration is therefore not effective.
With ERTEC 200P, JOB31 should not be used as HW-JOB; otherwise, the following error description and workaround apply. JOB31 could be used as SW-JOB if the SW does not interrupt it with JOB_Reset.

Error description: JOB31 is to be canceled and restarted

1. JOB31 is disabled with
 - a) JOB_EN = 0 (bit 1 in GDMA_JOB31_CTRL) or
 - b) HW_DMA_REQ_i = 0 (GDMA input signal)
2. JOB31 is reset by JOB_RESET = 1 (bit 5 in GDMA_JOB31_CTRL)
3. JOB31 is enabled, but not started. -> GDMA continues JOB31 from where it was interrupted, i.e. deleting does not work for JOB31 in the GDMA

Workaround:

1. Disable the JOB (JOB_EN = 0).
2. Wait until JOB stops copying. Check whether JOB has been completed.
3. If the JOB has not been completed, the first transfer of the new partial transfer list is copied from the transfer list to the interrupt memory (JOB stack) of the JOB.
4. The JOB is enabled (JOB_EN = 1), but not started.

- **Errors in JOB Stop with GDMA_MAIN_CTRL.DMA_EN = 0:**

Conditions:

- 1 job (GDMA_JOB0_CTRL)
- 1 transfer record run by that job
- SW triggering with GDMA_JOB0_CTRL.SW_JOB_START = 1

If you stop processing globally with GDMA_MAIN_CTRL.DMA_EN = 0 (GDMA_JOB0_CTRL.JOB_EN = 1), GDMA_JOB_COUNT continues to be incremented cyclically and

GDMA_ACTUAL_STATUS = 0x00000000 (no job running, actual job number 0) set.

This behavior is not correct as DMA_EN is supposed to pause job processing (and therefore also the GDMA_JOB_COUNT).

(Data sheet: When bit DMA_EN is reset while a DMA job is running, this job is interrupted and no new job begins to run. When subsequently bit DMA_EN is set again, the interrupted job will continue. DMA_EN has the same function as JOB_EN, but for all jobs).

Workaround:

- The difference in GDMA_ACTUAL_STATUS poses no problem as the content of this register is only valid when bit 5 = 1. When bit 5 = 0, the content of the register must not be evaluated!
- It is not OK that the job counter keeps counting when GDMA_MAIN_CTRL.DMA_EN = 0. However, this is only a minor flaw as the job counter is designed to measure the runtime of a job and the job cannot be disabled because that distorts the measurement. This is therefore not an appropriate use case.

2.3.4.4 Address Mapping

Address range allocation for the GDMA registers and the transfer list RAM:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	12AFh	gdma	
B0h	10AFh	LIST_RAM	
10B0h	12AFh	JOB_STACK_RAM	

Module	Register/Memory	Read	Write	Address
/gdma				
	GDMA_REG_ADDR	r	(w)	00h
	GDMA_LIST_ADDR	r	(w)	04h
	GDMA_MAIN_CTRL	r	(w)	08h
	GDMA_JC_EN	r	w	0Ch
	GDMA_JOB0_CTRL	(r)	(w)(t)	10h
	GDMA_JOB1_CTRL	(r)	(w)(t)	14h
	GDMA_JOB2_CTRL	(r)	(w)(t)	18h
	GDMA_JOB3_CTRL	(r)	(w)(t)	1Ch
	GDMA_JOB4_CTRL	(r)	(w)(t)	20h
	GDMA_JOB5_CTRL	(r)	(w)(t)	24h
	GDMA_JOB6_CTRL	(r)	(w)(t)	28h
	GDMA_JOB7_CTRL	(r)	(w)(t)	2Ch
	GDMA_JOB8_CTRL	(r)	(w)(t)	30h
	GDMA_JOB9_CTRL	(r)	(w)(t)	34h
	GDMA_JOB10_CTRL	(r)	(w)(t)	38h
	GDMA_JOB11_CTRL	(r)	(w)(t)	3Ch
	GDMA_JOB12_CTRL	(r)	(w)(t)	40h
	GDMA_JOB13_CTRL	(r)	(w)(t)	44h
	GDMA_JOB14_CTRL	(r)	(w)(t)	48h
	GDMA_JOB15_CTRL	(r)	(w)(t)	4Ch
	GDMA_JOB16_CTRL	(r)	(w)(t)	50h
	GDMA_JOB17_CTRL	(r)	(w)(t)	54h
	GDMA_JOB18_CTRL	(r)	(w)(t)	58h
	GDMA_JOB19_CTRL	(r)	(w)(t)	5Ch
	GDMA_JOB20_CTRL	(r)	(w)(t)	60h
	GDMA_JOB21_CTRL	(r)	(w)(t)	64h
	GDMA_JOB22_CTRL	(r)	(w)(t)	68h
	GDMA_JOB23_CTRL	(r)	(w)(t)	6Ch
	GDMA_JOB24_CTRL	(r)	(w)(t)	70h
	GDMA_JOB25_CTRL	(r)	(w)(t)	74h
	GDMA_JOB26_CTRL	(r)	(w)(t)	78h

	GDMA_JOB27_CTRL	(r)	(w)(t)	7Ch
	GDMA_JOB28_CTRL	(r)	(w)(t)	80h
	GDMA_JOB29_CTRL	(r)	(w)(t)	84h
	GDMA_JOB30_CTRL	(r)	(w)(t)	88h
	GDMA_JOB31_CTRL	(r)	(w)(t)	8Ch
	GDMA_JOB_STATUS	rh		90h
	GDMA_FINISHED_JOBS	rh	w	94h
	GDMA_ACTUAL_STATUS	r(h)		98h
	GDMA_IRQ_STATUS	rh	w	9Ch
	GDMA_ERR_IRQ_STATUS	r(h)	(w)	A0h
	GDMA_JOB_COUNT	rh		A4h
	REVISION_CODE	r		A8h
	LIST_RAM	R	W	00B0h - 10AFh
	JOB_STACK_RAM	R	W	10B0h - 12AFh

2.3.4.5 Register Description

A description of the registers can be found in the specifications: /55/.

Note: The "notused" register bits can also be written, unlike in the IP specifications.

The data-bus-width of all specified registers in this module is: 32.

A '0' is read from Software for each not specified Bit in the registers.

The addressing is organized: WORD - wise

Module: /gdma

Register:	GDMA_REG_ADDR			Address:	0h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	(w)
Description:	GDMA Register Base Address					
Bit	Identifier	Reset	Attr.	Function / Description		
3dt0	<notused>	0h	r	all bits = '0'		
31dt4	REG_ADDR	0000000h	r w	DMA Register Base Address		

Register:	GDMA_LIST_ADDR			Address:	4h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	(w)
Description:	DMA Transfer List Base Address					
Bit	Identifier	Reset	Attr.	Function / Description		
3dt0	<notused>	0h	r	all bits = '0'		
31dt4	LIST_ADDR	0000000h	r w	DMA Transfer List Base Address		

Register:	GDMA_MAIN_CTRL			Address:	8h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	(w)
Description:	Main Control register					
Bit	Identifier	Reset	Attr.	Function / Description		
0dt0	DMA_EN	0h	r w	Global Enable		
1dt1	SW_RESET	0h	r w	Software Reset '0' = don't care, '1' = Reset		
14dt2	<notused>	00h	r	all bits = '0'		
15dt15	JC_RESET	0h	r w	Reset Job Counter '0' = don't care, '1' = Reset		
16dt16	ERR_INT_EN	0h	r w	Error Interrupt Enable		

23dt17	<notused>	00h	r		all bits = '0'
31dt24	LIST_SIZE	00h	r	w	Total Number of Transfers in Transfer List (0-255) 0 means 1, 255-> 256

Register:	GDMA_JC_EN				Address:	Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w	
Description:		Enable Job Counter for Job					

Register:	GDMA_JOB0_CTRL				Address:	10h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r) (w)(t)
Description:		Job Control Registers 0					
Bit	Identifier		Reset	Attr.		Function / Description	
0dt0	SW_JOB_START		0h		t	Start Job from SW 0 = don't care, 1 = Start	
1dt1	JOB_EN		0h	r	w	Job Enable 0 = disable, 1 = enable	
2dt2	HW_JOB_START_EN		0h	r	w	Start Job by Hardware 0 = disable, 1 = enable	
3dt3	HW_FLOW_EN		0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled	
4dt4	INTR_EN		0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled	
5dt5	JOB_RESET		0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel	
7dt6	<notused>		0h	r		all bits = '0'	
13dt8	HW_SELECT		00h	r	w	HW Job Start Selector (selects one from 64 inputs)	
15dt14	<notused>		0h	r		all bits = '0'	
20dt16	JOB_PRIO		00h	r	w	Job Priority (0-31)	
23dt21	<notused>		0h	r		all bits = '0'	
31dt24	TRANSFER_PTR		00h	r	w	First Transfer Number of Job in Transfer List (0-255)	

Register:	GDMA_JOB1_CTRL				Address:	14h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)	
Description:	Job Control Registers 1						

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel
7dt6	<notused>	0h	r		all bits = '0'
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:	GDMA_JOB2_CTRL				Address:	18h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 2					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		

7dt6	<notused>	0h	r		all bits = '0'
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:	GDMA_JOB3_CTRL				Address:	1Ch	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 3					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	<notused>	0h	r		all bits = '0'		
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB4_CTRL				Address:	20h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)

Description:		Job Control Registers 4			
Bit	Identifier	Reset	Attr.		Function / Description
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel
7dt6	<notused>	0h	r		all bits = '0'
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:	GDMA_JOB5_CTRL				Address:	24h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 5					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		

5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel
7dt6	<notused>	0h	r		all bits = '0'
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:	GDMA_JOB6_CTRL				Address:	28h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r) (w)(t)
Description:		Job Control Registers 6					
Bit	Identifier		Reset	Attr.		Function / Description	
0dt0	SW_JOB_START		0h		t	Start Job from SW 0 = don't care, 1 = Start	
1dt1	JOB_EN		0h	r	w	Job Enable 0 = disable, 1 = enable	
2dt2	HW_JOB_START_EN		0h	r	w	Start Job by Hardware 0 = disable, 1 = enable	
3dt3	HW_FLOW_EN		0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled	
4dt4	INTR_EN		0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled	
5dt5	JOB_RESET		0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel	
7dt6	<notused>		0h	r		all bits = '0'	
13dt8	HW_SELECT		00h	r	w	HW Job Start Selector (selects one from 64 inputs)	
15dt14	<notused>		0h	r		all bits = '0'	
20dt16	JOB_PRIO		00h	r	w	Job Priority (0-31)	
23dt21	<notused>		0h	r		all bits = '0'	
31dt24	TRANSFER_PTR		00h	r	w	First Transfer Number of Job in Transfer List (0-255)	

Register:	GDMA_JOB7_CTRL	Address:	2Ch
-----------	----------------	----------	-----

Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 7					
Bit	Identifier	Reset	Attr.	Function / Description		
0dt0	SW_JOB_START	0h	t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r	all bits = '0'		
13dt8	HW_SELECT	00h	r w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r	all bits = '0'		
20dt16	JOB_PRIO	00h	r w	Job Priority (0-31)		
23dt21	<notused>	0h	r	all bits = '0'		
31dt24	TRANSFER_PTR	00h	r w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB8_CTRL			Address:	30h
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r) (w)(t)
Description:	Job Control Registers 8				
Bit	Identifier	Reset	Attr.	Function / Description	
0dt0	SW_JOB_START	0h	t	Start Job from SW 0 = don't care, 1 = Start	
1dt1	JOB_EN	0h	r w	Job Enable 0 = disable, 1 = enable	
2dt2	HW_JOB_START_EN	0h	r w	Start Job by Hardware 0 = disable, 1 = enable	
3dt3	HW_FLOW_EN	0h	r w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled	
4dt4	INTR_EN	0h	r w	Interrupt Request Generation Enable	

					0 = disabled, 1 = enabled
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel
7dt6	<notused>	0h	r		all bits = '0'
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:		GDMA_JOB9_CTRL				Address:		34h			
Bits:		31dt0		Reset value:		00000000h		Attributes:		(r)	(w)(t)
Description:			Job Control Registers 9								
Bit	Identifier			Reset		Attr.		Function / Description			
0dt0	SW_JOB_START			0h		t		Start Job from SW 0 = don't care, 1 = Start			
1dt1	JOB_EN			0h		r w		Job Enable 0 = disable, 1 = enable			
2dt2	HW_JOB_START_EN			0h		r w		Start Job by Hardware 0 = disable, 1 = enable			
3dt3	HW_FLOW_EN			0h		r w		Hardware Triggered Flow Enable 0 = disabled, 1 = enabled			
4dt4	INTR_EN			0h		r w		Interrupt Request Generation Enable 0 = disabled, 1 = enabled			
5dt5	JOB_RESET			0h		r w		Reset Job (cancel running job) 0 = don't care, 1 = Cancel			
7dt6	<notused>			0h		r		all bits = '0'			
13dt8	HW_SELECT			00h		r w		HW Job Start Selector (selects one from 64 inputs)			
15dt14	<notused>			0h		r		all bits = '0'			
20dt16	JOB_PRIO			00h		r w		Job Priority (0-31)			
23dt21	<notused>			0h		r		all bits = '0'			
31dt24	TRANSFER_PTR			00h		r w		First Transfer Number of Job in Transfer List (0-255)			

Register:	GDMA_JOB10_CTRL				Address:	38h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 10					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	<notused>	0h	r		all bits = '0'		
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB11_CTRL				Address:	3Ch	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 11					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled. 1 = enabled		

4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel
7dt6	<notused>	0h	r		all bits = '0'
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:	GDMA_JOB12_CTRL				Address:	40h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 12					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	<notused>	0h	r		all bits = '0'		
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB13_CTRL				Address:	44h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 13					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	<notused>	0h	r		all bits = '0'		
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB14_CTRL				Address:	48h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r) (w)(t)
Description:		Job Control Registers 14					
Bit	Identifier		Reset	Attr.		Function / Description	
0dt0	SW_JOB_START		0h		t	Start Job from SW 0 = don't care, 1 = Start	
1dt1	JOB_EN		0h	r	w	Job Enable 0 = disable, 1 = enable	
2dt2	HW_JOB_START_EN		0h	r	w	Start Job by Hardware 0 = disable, 1 = enable	
3dt3	HW_FLOW_EN		0h	r	w	Hardware Triggered Flow Enable	

					0 = disabled, 1 = enabled
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel
7dt6	<notused>	0h	r		all bits = '0'
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:		GDMA_JOB15_CTRL				Address:		4Ch			
Bits:		31dt0		Reset value:		00000000h		Attributes:		(r)	(w)(t)
Description:			Job Control Registers 15								
Bit	Identifier			Reset		Attr.		Function / Description			
0dt0	SW_JOB_START			0h			t	Start Job from SW 0 = don't care, 1 = Start			
1dt1	JOB_EN			0h		r	w	Job Enable 0 = disable, 1 = enable			
2dt2	HW_JOB_START_EN			0h		r	w	Start Job by Hardware 0 = disable, 1 = enable			
3dt3	HW_FLOW_EN			0h		r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled			
4dt4	INTR_EN			0h		r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled			
5dt5	JOB_RESET			0h		r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel			
7dt6	<notused>			0h		r		all bits = '0'			
13dt8	HW_SELECT			00h		r	w	HW Job Start Selector (selects one from 64 inputs)			
15dt14	<notused>			0h		r		all bits = '0'			
20dt16	JOB_PRIO			00h		r	w	Job Priority (0-31)			
23dt21	<notused>			0h		r		all bits = '0'			
31dt24	TRANSFER_PTR			00h		r	w	First Transfer Number of Job in Trans-			

				fer List (0-255)
--	--	--	--	------------------

Register:	GDMA_JOB16_CTRL				Address:	50h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 16					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	<notused>	0h	r		all bits = '0'		
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB17_CTRL				Address:	54h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r) (w)(t)
Description:		Job Control Registers 17					
Bit	Identifier		Reset	Attr.		Function / Description	
0dt0	SW_JOB_START		0h		t	Start Job from SW 0 = don't care, 1 = Start	
1dt1	JOB_EN		0h	r	w	Job Enable 0 = disable, 1 = enable	
2dt2	HW_JOB_START_EN		0h	r	w	Start Job by Hardware 0 = disable, 1 = enable	

3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel
7dt6	<notused>	0h	r		all bits = '0'
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:	GDMA_JOB18_CTRL				Address:	58h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 18					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	<notused>	0h	r		all bits = '0'		

31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)
--------	--------------	-----	---	---	---

Register:	GDMA_JOB19_CTRL				Address:	5Ch	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 19					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	<notused>	0h	r		all bits = '0'		
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB20_CTRL					Address:	60h		
Bits:	31dt0		Reset value:		00000000h		Attributes:	(r)	(w)(t)
Description:			Job Control Registers 20						
Bit	Identifier		Reset	Attr.		Function / Description			
0dt0	SW_JOB_START		0h		t	Start Job from SW 0 = don't care, 1 = Start			
1dt1	JOB_EN		0h	r	w	Job Enable 0 = disable, 1 = enable			
2dt2	HW JOB START EN		0h	r	w	Start Job by Hardware			

					0 = disable, 1 = enable
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel
7dt6	<notused>	0h	r		all bits = '0'
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:	GDMA_JOB21_CTRL				Address:	64h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 21					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		

23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:		GDMA_JOB22_CTRL				Address:		68h			
Bits:		31dt0		Reset value:		00000000h		Attributes:		(r)	(w)(t)
Description:			Job Control Registers 22								
Bit	Identifier			Reset		Attr.		Function / Description			
0dt0	SW_JOB_START			0h			t	Start Job from SW 0 = don't care, 1 = Start			
1dt1	JOB_EN			0h		r	w	Job Enable 0 = disable, 1 = enable			
2dt2	HW_JOB_START_EN			0h		r	w	Start Job by Hardware 0 = disable, 1 = enable			
3dt3	HW_FLOW_EN			0h		r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled			
4dt4	INTR_EN			0h		r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled			
5dt5	JOB_RESET			0h		r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel			
7dt6	<notused>			0h		r		all bits = '0'			
13dt8	HW_SELECT			00h		r	w	HW Job Start Selector (selects one from 64 inputs)			
15dt14	<notused>			0h		r		all bits = '0'			
20dt16	JOB_PRIO			00h		r	w	Job Priority (0-31)			
23dt21	<notused>			0h		r		all bits = '0'			
31dt24	TRANSFER_PTR			00h		r	w	First Transfer Number of Job in Transfer List (0-255)			

Register:	GDMA_JOB23_CTRL				Address:	6Ch	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r) (w)(t)
Description:		Job Control Registers 23					
Bit	Identifier		Reset	Attr.		Function / Description	
0dt0	SW_JOB_START		0h		t	Start Job from SW 0 = don't care, 1 = Start	
1dt1	JOB_EN		0h	r	w	Job Enable 0 = disable, 1 = enable	

2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel
7dt6	<notused>	0h	r		all bits = '0'
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:	GDMA_JOB24_CTRL				Address:	70h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 24					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		

20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:	GDMA_JOB25_CTRL				Address:	74h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 25					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	<notused>	0h	r		all bits = '0'		
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB26_CTRL				Address:	78h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r) (w)(t)
Description:		Job Control Registers 26					
Bit	Identifier		Reset	Attr.	Function / Description		
0dt0	SW_JOB_START		0h	t	Start Job from SW 0 = don't care, 1 = Start		

1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel
7dt6	<notused>	0h	r		all bits = '0'
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:	GDMA_JOB27_CTRL				Address:	7Ch	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 27					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		

15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:		GDMA_JOB28_CTRL				Address:		80h			
Bits:		31dt0		Reset value:		00000000h		Attributes:		(r)	(w)(t)
Description:			Job Control Registers 28								
Bit	Identifier			Reset		Attr.		Function / Description			
0dt0	SW_JOB_START			0h			t	Start Job from SW 0 = don't care, 1 = Start			
1dt1	JOB_EN			0h		r	w	Job Enable 0 = disable, 1 = enable			
2dt2	HW_JOB_START_EN			0h		r	w	Start Job by Hardware 0 = disable, 1 = enable			
3dt3	HW_FLOW_EN			0h		r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled			
4dt4	INTR_EN			0h		r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled			
5dt5	JOB_RESET			0h		r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel			
7dt6	<notused>			0h		r		all bits = '0'			
13dt8	HW_SELECT			00h		r	w	HW Job Start Selector (selects one from 64 inputs)			
15dt14	<notused>			0h		r		all bits = '0'			
20dt16	JOB_PRIO			00h		r	w	Job Priority (0-31)			
23dt21	<notused>			0h		r		all bits = '0'			
31dt24	TRANSFER_PTR			00h		r	w	First Transfer Number of Job in Transfer List (0-255)			

Register:	GDMA_JOB29_CTRL				Address:	84h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 29					
Bit	Identifier	Reset	Attr.	Function / Description			
0dt0	SW JOB START	0h	t	Start Job from SW			

					0 = don't care, 1 = Start
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel
7dt6	<notused>	0h	r		all bits = '0'
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:	GDMA_JOB30_CTRL				Address:	88h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 30					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		

13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)
15dt14	<notused>	0h	r		all bits = '0'
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)
23dt21	<notused>	0h	r		all bits = '0'
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)

Register:	GDMA_JOB31_CTRL				Address:	8Ch	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)(t)
Description:		Job Control Registers 31					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	<notused>	0h	r		all bits = '0'		
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB_STATUS				Address:	90h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	rh	
Description:		Job (31:0) State 0 = not started, 1 = started					

Register:	GDMA_FINISHED_JOBS			Address:	94h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	w
Description:	Finished Job(31:0) State (write 1 to clear) (INTR_EN independent) 0 = not finished, 1 = finished					

Register:	GDMA_ACTUAL_STATUS			Address:	98h	
Bits:	31dt0	Reset value:	0000001Fh	Attributes:	r(h)	
Description:	Actual Job State					
Bit	Identifier	Reset	Attr.	Function / Description		
4dt0	ACT_JOB	1Fh	rh	Actual Job Number – the number of the running Job		
5dt5	ACT_JOB_VAL	0h	rh	Actual Job Number Valid 0 = no job running, 1 = Job Number valid Note: this flag is needed for ACT_JOB = "00000"		
31dt6	<notused>	0000000h	hr	all bits = '0'		

Register:	GDMA_IRQ_STATUS			Address:	9Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	w
Description:	Job finished, Interrupt generated (INTR_EN dependent) (write 1 to clear) 0 = Job not finished, 1 = Job finished					

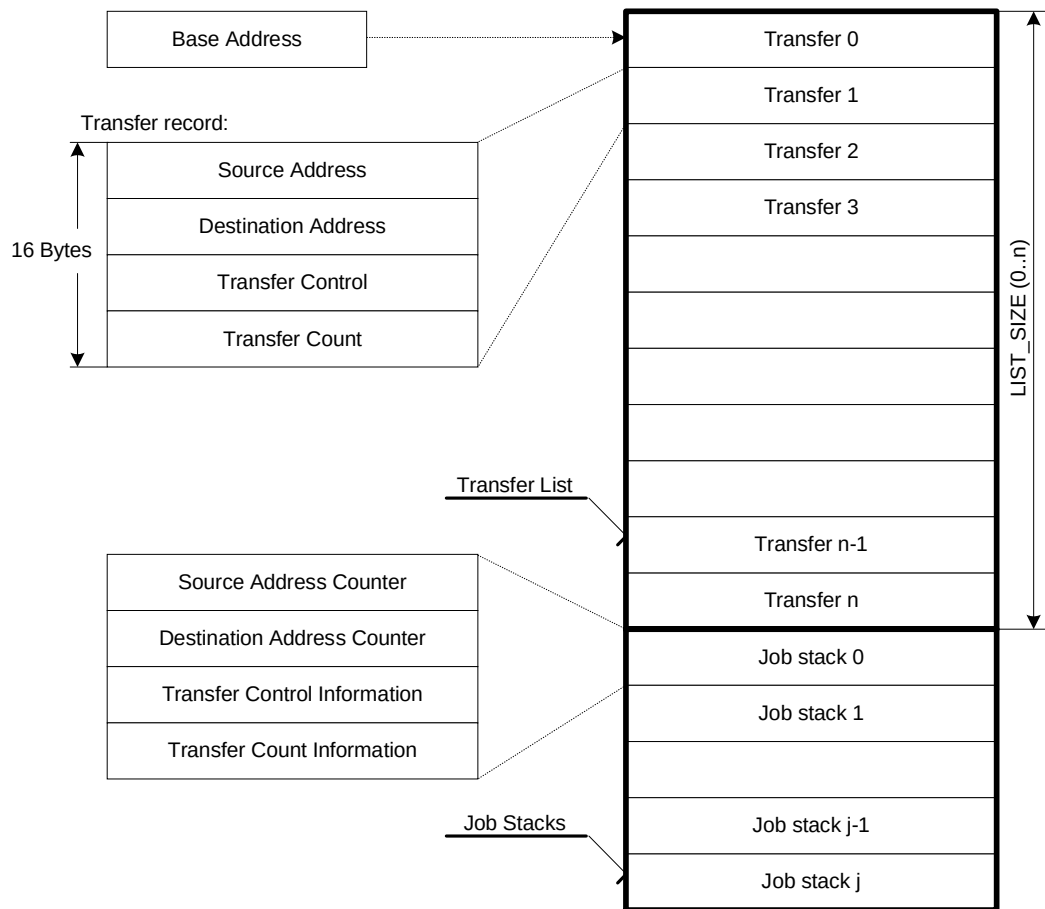
Register:	GDMA_ERR_IRQ_STATUS			Address:	A0h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r(h)	(w)
Description:	Error interrupt State register					
Bit	Identifier	Reset	Attr.	Function / Description		
0dt0	ERR_DST_ADDR	0h	rh w	DMA Destination Address Error (write 1 to clear) 0 = no Error, 1 = Error occurred		
5dt1	ERR_DST_ADDR_JOB_NR	00h	rh	DMA Destination Address Error JOB number		
6dt6	ERR_AHB	0h	rh w	AHB Master Interface Error (write 1 to clear)		

					0 = no Error, 1 = AHB Error Response occurred
11dt7	ERR_AHB_JOB_NR	00h	rh		AHB Master Interface Error JOB number
12dt12	ERR_JOB_START	0h	rh	w	Hardware start of Job occurred before Job is finished (write 1 to clear) 0 = no Error, 1 = Error
17dt13	ERR_JOB_START_JOB_NR	00h	rh		Hardware start of Job occurred before Job is finished JOB number
18dt18	ERR_AHB_SLV_WRITE	0h	rh	w	8 or 16 bit write access on AHB slave interface (write 1 to clear) 0 = no Error, 1 = not allowed write
19dt19	ERR_AM_HOLD	0h	rh	w	Addressmode of Source or Destination address is HOLD and ESIZE and address are not aligned (write 1 to clear) 0 = no Error, 1 = Error
24dt20	ERR_AM_HOLD_JOB_NR	00h	rh		ERR_AM_HOLD JOB number
31dt25	<notused>	00h	r		all bits = '0'

Register:	GDMA_JOB_COUNT			Address:	A4h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Counts clock cycles, when selected Jobs are active. Selection of the counted Jobs is done through the GDMA_JC_EN register. Each Job has own enable bit here. The reset of Job Counter is done through "Reset Job Counter" flag of the GDMA_MAIN_CTRL register.					

Register:	REVISION_CODE			Address:	A8h	
Bits:	15dt0	Reset value:	0101h	Attributes:	r	
Description:	Revision Code register					
Bit	Identifier	Reset	Attr.	Function / Description		
7dt0	MINOR_REVISION	01h	r	minor revision = 0x1		
15dt8	MAJOR_REVISION	01h	r	major revision = 0x1		

DMA RAM:



$$\text{RAM Size} = (n+1)*16+(j+1)*16 \text{ Byte}$$

-- stack begins after last Transfer List entry:

```
stack_addr_s <= std_logic_vector(conv_unsigned (16*(LIST_SIZE+1),13));
```

Table 13: The DMA RAM address space.

The configuration for ERTEC 200P is $n = 255$ and $J = 31$. **For this configuration the internal DMA SRAM is organized in 1152 x 32 Bit.**

DMA Transfer Record – Source Address SRC_ADDR

Bit no.	Name	Description
31:0	SRC	Source Address

DMA Transfer Record – Destination Address DEST_ADDR

Bit no.	Name	Description
31:0	DEST	Destination Address

DMA Transfer Record – Transfer Control

Bit no.	Name	Description
---------	------	-------------

31:24		reserved
23:22	SRC_AMODE	Source Address Mode 00 = increment address 10 = hold address x1 = reserved for future implementation = hold address
21:20	DEST_AMODE	Destination Address Mode (reserved = hold address) 00 = increment address 10 = hold address x1 = reserved for future implementation = hold address
19:18	BURST_MODE	00 = Single 01 = INCR4 10 = INCR8 11 = INCR16
17:16		reserved

DMA Transfer Record – Transfer Count

Bit no.	Name	Description
31	LAST_TR	Last Transfer of the Job
30	EN_DMA_ACK	1 = set output HW_DMA_ACK if transfer is finished
29:24		reserved
23:22	ESIZE	Element size 00 = 8bit 01 = 16 bit 10 = 32 bit 11 = reserved
21:16		reserved
15:0	TR_COUNT	Transfer Count 0x0 = 1 element, 0xFFFF = 65536 elements

DMA Stack – Source Address Stack

Bit no.	Name	Description
31:0	CSRC	Current Source Address

DMA Stack – Destination Address Stack

Bit no.	Name	Description
31:0	CDEST	Current Destination Address

DMA Stack – Transfer Control Stack

Bit no.	Name	Description
---------	------	-------------

31:24		reserved
23:22	SRC_AMODE	Source Address Mode 00 = increment address 10 = hold address x1 = reserved for future implementation = hold address
21:20	DEST_AMODE	Destination Address Mode 00 = increment address 10 = hold address x1 = reserved for future implementation = hold address
19:18	BURST_MODE	00 = Single 01 = INCR4 10 = INCR8 11 = INCR16
15:8		reserved
7:0	CTransfer	Current Transfer List Number to continue from after interruption

DMA Stack – Transfer Count Stack

Bit no.	Name	Description
31	LAST_TR	Last Transfer of the Job 0 = not last, 1=last
30	EN_DMA_ACK	1 = set output HW_DMA_ACK if transfer is finished
29:24		reserved
23:22	ESIZE	Element Size 00 = 8 bit 01 = 16 bit 10 = 32 bit 11 = reserved
21:16		reserved
15:0	CTR_COUNT	Current Transfer Count 0x0 = 1 element, 0xFFFF = 63536 elements

2.3.5 EMC (External Memory Controller)

Modern SoC designs are often based on an “onchip” bus system, where various modules are connected to. Within I DT / I IA the AMBA bus system of ARM is the most used one. To allow masters on this bus system to access external memory devices, a functionality is needed, which bridge from the internal AHB bus to external memory devices. This functionality is provided with the External Memory Controller (EMC) module. The complete description of the EMC please see following document 'EMC_....pdf' /36/ (see chapter):

The EMC interface contains 2 separate memory controllers – SDRAM controller (SDRAMC) and an asynchronous controller (ASYNCC) – for different devices like

- SDRAM (including Mobile SDRAM) Memory
- SRAM (including Flash EPROMs) Memory
- External devices, running a SRAM timing with additional READY signal
- Burst Mode Flash ROM

It is connected to the system bus using an AHB slave interface with the following features

- Supports AHB 2.0 lite protocol (No SPLIT, no RETRY)
- Burst transactions are accepted at the AHB input side.
 - o if the SDRAM controller is active, bursts of undefined length are split into bursts of length 16.
 - o if the asynchronous controller is active, any burst is split into single transfers (exception of this rule: read access to Burst Flash ROM allows 16 beat burst, read access to Page Mode ROM allows 16 beat burst).
- The EMC itself can be configured using the AHB slave interface

EMC comprises of 2 different controllers, one supporting the SDRAM Memory devices (including Mobile SDRAM), the other supporting asynchronous SRAM timing in different flavors, including Burst Flash ROM memory devices.

- SDRAM-Controller features:
 - o 16/32 Bit databus width
 - o PC133 SDRAM-compatible (125 MHz synchron is used in ERTEC 200P)
 - o 1 Bank with max. 256 MByte SDRAM (32 Bit databus)
 - o SDRAM support for following parts:
 - CAS-Latency: 2 or 3 clocks
 - Bank-address bits (1/2/4 internal banks), realized via the lowest two bits of the address bus MA(1:0)
 - 8/9/10/11 bits column-address MA(13), MA(11:2)
 - max. 14 bits row-address MA(15:2)

With 27 address lines (2 BANK, 14 ROW, 11 COL) it is possible to create 128M different addresses. This 128M different addresses build up an address space of 512 MByte using a 32 bit databus width, or 256 MByte using a 16 bit databus width. But the usable size is limited by the EMC internal address decoder to a size of 256 Mbytes in total. SDRAMs have a maximum of 4 internal banks. The SDRAM controller can open all 4 banks in parallel. Those 4 banks are a quarter of the SDRAM address space at the AHB bus.

- Asynchronous Controller features:
 - o Can be set to 8/16/32-bit data bus width (for each chip select programmable)
 - o 4 chip selects
 - o The timing for each chip select can be set individually
 - o The response to ready signal can be set individually for each chip select

- o The default setting is slow timing for booting purposes
- o A maximum of 64 MByte address area for each chip select
- o Acknowledgement delay monitoring for external components can be set by software

2.3.5.1 Block Diagram of the EMC

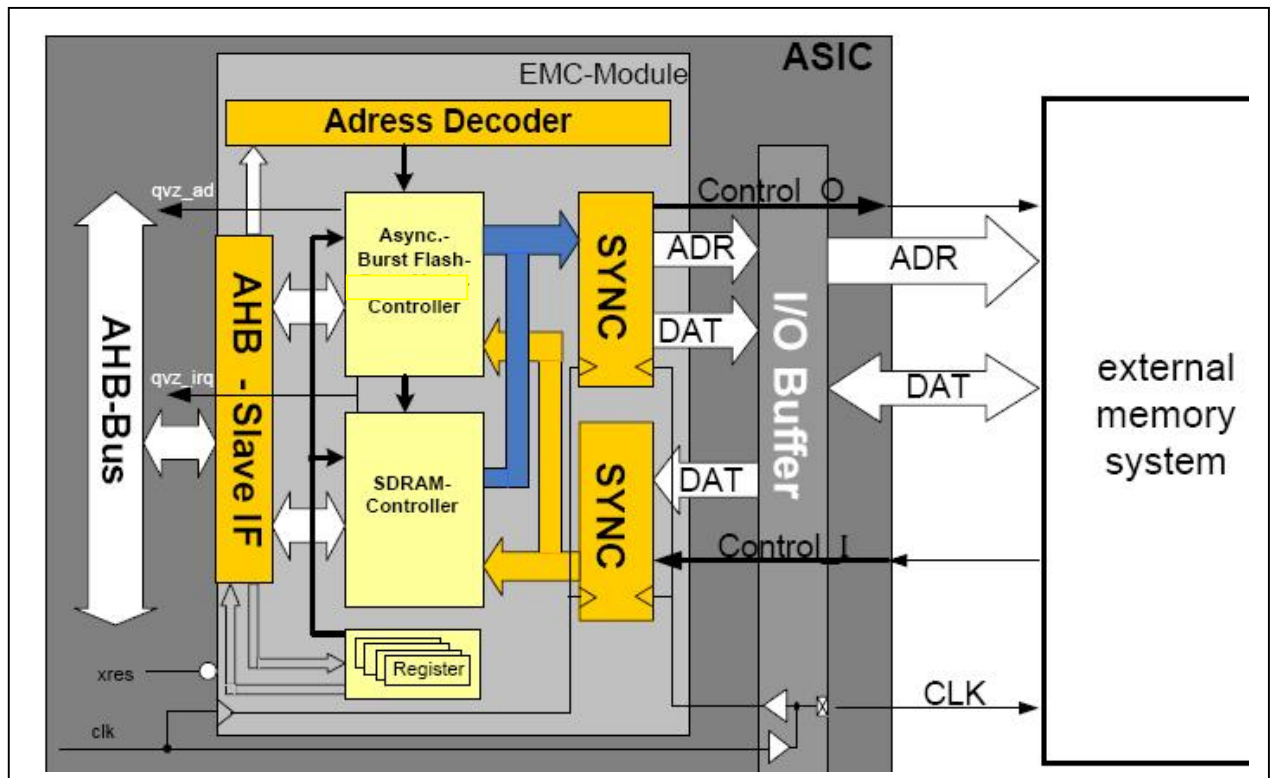


Figure 9: EMC Block Diagram

Note: External memory system is a representation for all types of memories, which can be connected to the EMC. Forwarding the AHB clock to the external memory system via a BIDI buffer, which feeds back the clock to the receive part of the EMC module is only an example application. For high clock frequencies (> 100 MHz) it is recommended to use separated input/output-buffers to feedback the external memory clock.

The EMC interface can only be driven by 1,8V (necessary for mobile SDRAM). The EMC IO pads are separated from the other IOs and have their own power supply (VDD1,8V and GND). This supply ring contains all 77 IO Pads of the EMC interface. So the EMC can work with 1,8V and the other IOs (e.g. GPIOs) with 3,3V.

The driver strength of the 1.8V EMC pads is 12mA after PowerOn Reset by default.

2.3.5.2 EMC Reset and Clock

In main ximum memory configuration at the EMC interface is shown (2x SDRAM devices, 2x Burst Flash devices, 1x Peripheral / Level Shifter for 3,3V). Each memory device gets its own clock. The ERTEC 200P has 3 Clocks for the SDRAM memory (CLK_O_SDRAM2/1/0) and 3 Clocks for the Burst Flash memory (CLK_O_BF2/1/0).

The CLK_O_SDRAM0 / CLK_O_BF0 must be used for feedback the external memory clock to the ERTEC 200P inputs CLK_I_SDRAM / CLK_I_BF.

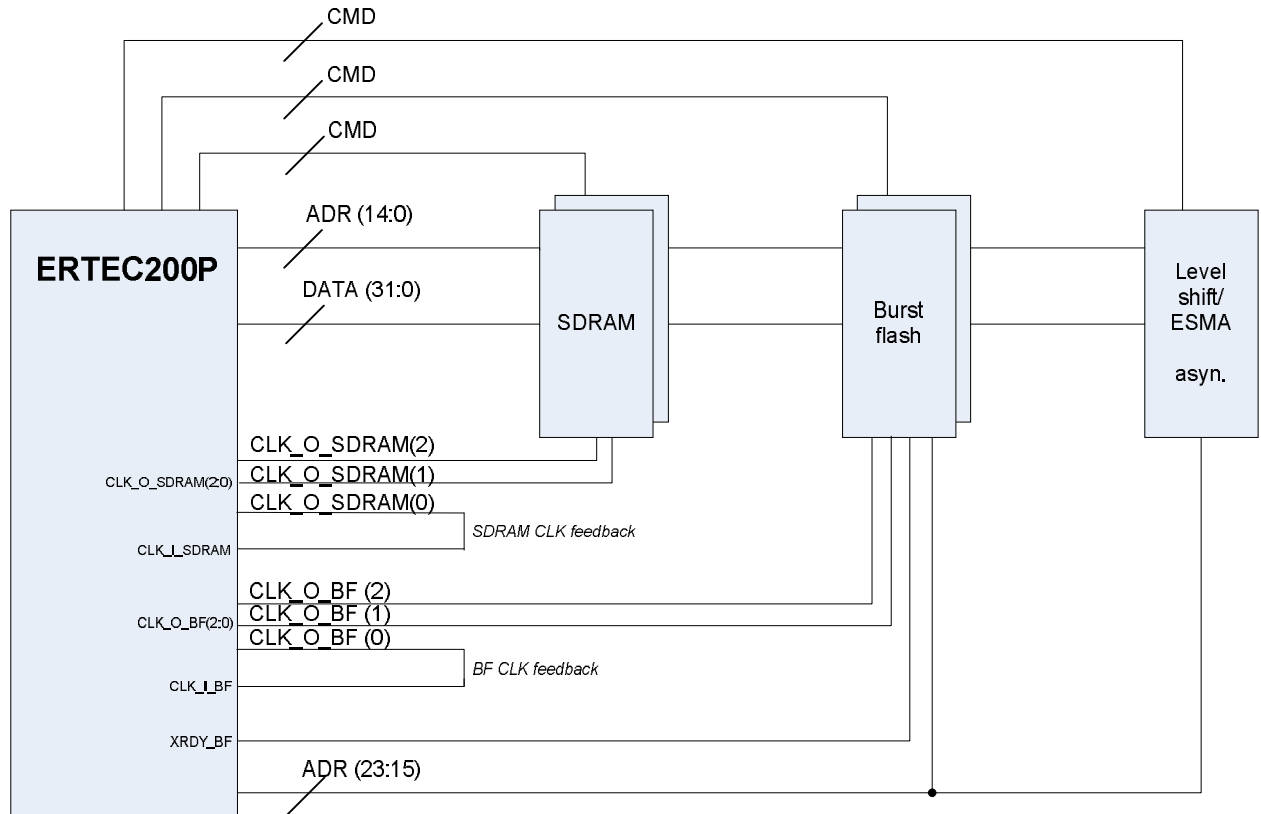


Figure 10: EMC interface with two SDRAM / two Burst Flash configuration

CLK_O_SDRAM1 / CLK_O_BF1 are used for the respective memory device no.1 and CLK_O_SDRAM2 / CLK_O_BF2 for the respective memory device no.2.

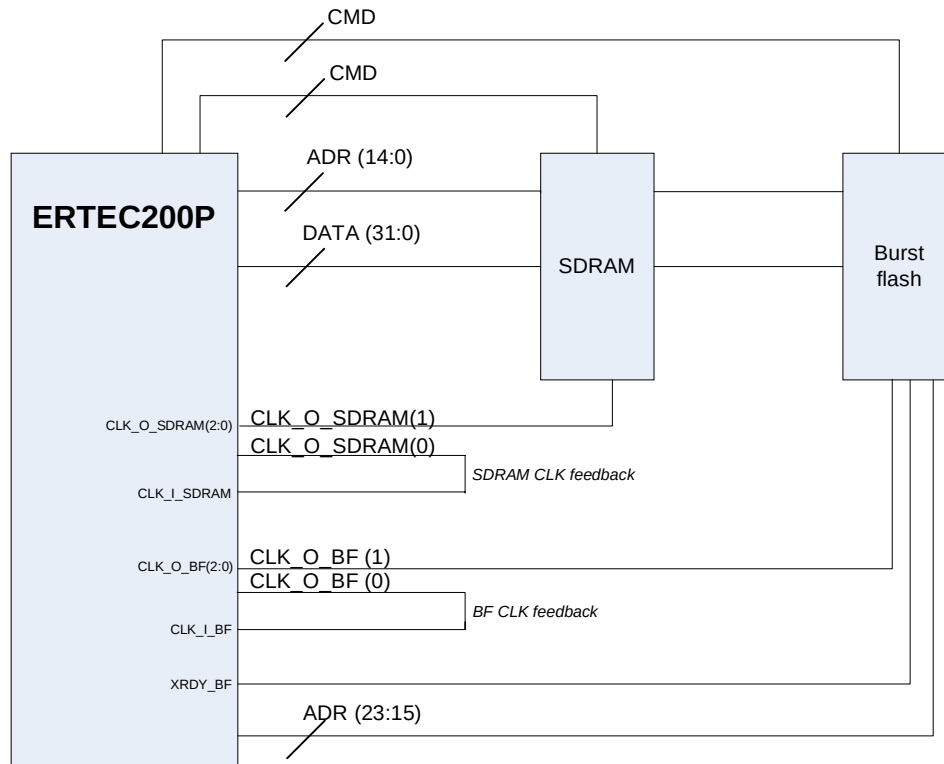


Figure 11: EMC interface with only one SDRAM / one Burst Flash configuration

CLK_O_SDRAM0 must always be returned back to CLK_I_SDRAM, even if only one external SRAM or an EMC / XHIF coupling to a second ERTEC 200P is used (see Figure 11). The feedback clock stores the incoming read data inside ERTEC 200P. The storing time can be influenced by boardlayout through the length of feedbackclock line.

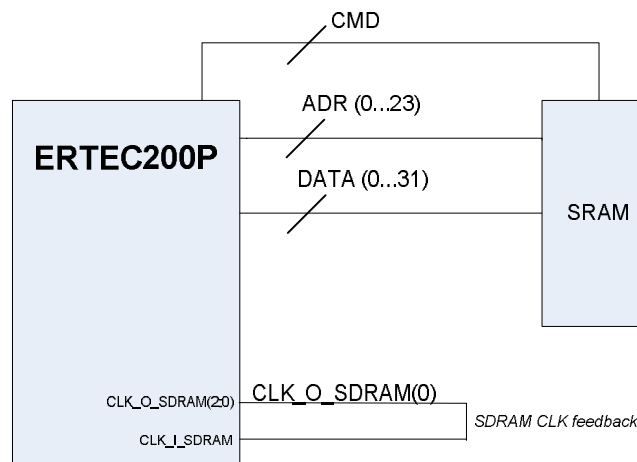


Figure 12: EMC interface with asynchronous RAM

All Clocks can be switched off, if they don't be used. In the DRIVE_EMC register (see chap. 2.3.10.9.22) are the appropriate bits for this function. CLK_O_SDRAM0/1, CLK_O_SDRAM2, CLK_O_BF0/1 and CLK_O_BF2 could be switched separately.

Attention: After PowerOn Reset all CLK_O_SDRAM0...2 are switched on.

2.3.5.3 AHB-Slave Interface

The AHB-slave interface supports the following features:

- Supports AHB 2.0 lite protocol (No SPLIT, no RETRY)
- Burst transactions are accepted at the AHB input side.
 - o if the SDRAM controller is active, bursts of undefined length are split into bursts of length 16.
 - o if the asynchronous controller is active, any burst is split into single transfers (exception of this rule: read access to Burst Flash ROM allows 16 beat burst, read access to Page Mode ROM allows 16 beat burst).
- The EMC itself can be configured using the AHB slave interface

and contains an address code which divides the EMC address range of the AHB into 6 sub-ranges. The first sub-range is designed for 256 MByte SDRAM. The second range is subdivided into 4 areas of 64 MByte each and is designed for the SRAM and other asynchronous memory modules. The last range contains the internal EMC registers which have an address range of 1 MByte.

The interface can be adapted to the RAM blocks used by the configuration registers.

The diagram below shows address range allocation:

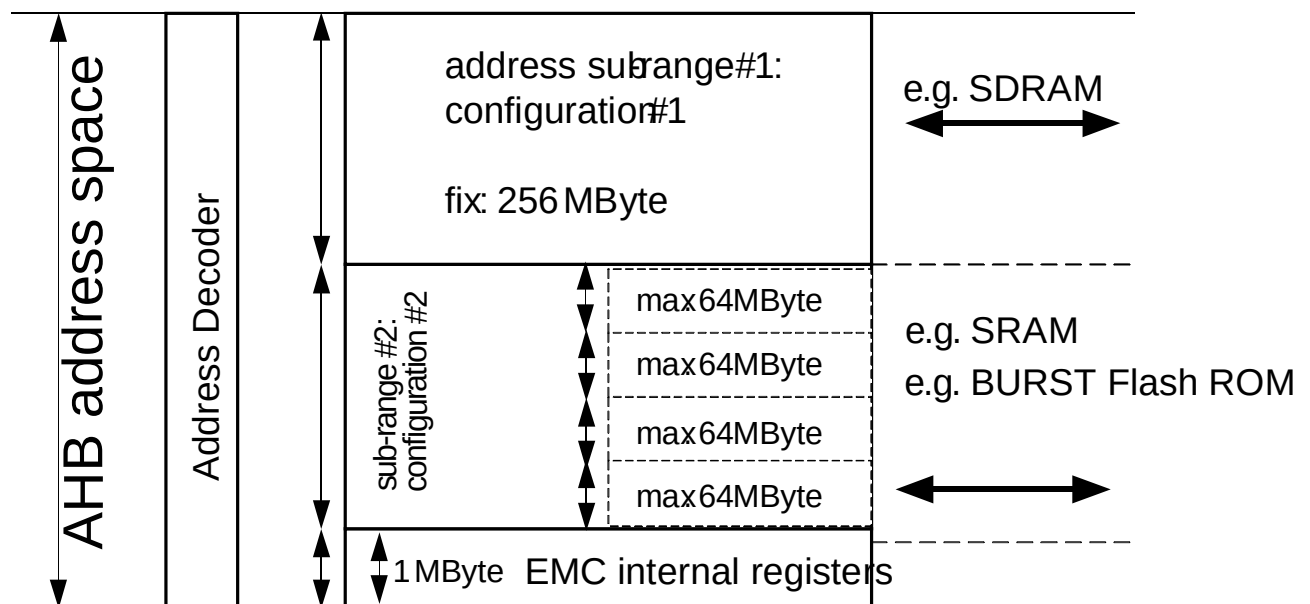


Figure 13: EMC, Address Space

2.3.5.4 EMC Notes

2.3.5.4.1 Remapping

Remapping the AHB address range from 0x0000_0000 to 0x0FFF_FFFF to the range of the SDRAM or the flash/SRAM block. This is required for some masters (ARM CPUs) because their exception table is expected when you boot at address 0x0000_0000.

The address may be moved from 0 to another address range depending on the *MEM_SWAP* signal.

- "00" → no action (*Default*: Boot ROM)
- "01" → start address of SDRAM device (sub-range #1)
- "10" → start address of Asynchronous device (sub-range #2)
- "11" → no action (QVZ error upon access to 0x0000_0000, see 2.3.2.5.1)

There is no address remap with 00 or 11. With 01, address 0x0000_0000 is forwarded to the start address of the SDRAM. With 10, address 0x0000_0000 is forwarded to the start address of the asynchronous RAM, ROM or flash. In the latter two cases, address conversion is carried out so that the required memory can be reached.

If, for example, the exception table of a CPU is located in the flash memory (sub-range#2), an ARM-CPU requires this table at 0x0000_0000 during booting. The address is therefore remapped (in line with *MEM_SWAP*).

In order to avoid unnecessary wait states at the AHB bus, the EMC can buffer a maximum of 4 instances of AHB write access and acknowledge them immediately (provided there are no more than 4). Read access is delayed until the write buffer is empty (all data have been written).

2.3.5.4.2 Maximum number of wait cycles

The configured value for the maximum number of wait cycles should be configured at least 5 times bigger than the expected worst case delay time of the external wait signal. False QVZ interrupts are possible when the number of wait cycles and the delay of the external wait signal have the same size. There is no way to detect a false QVZ interrupt by software but to compare a read value with an expected value. If a false QVZ interrupt is generated then the stored QVZ address can be false, too.

2.3.5.4.3 Shift Mode with the Asynchronous EMC Interface

Irrespective of the configuration of the boot pins, i.e. in every boot mode, the asynchronous EMC interface (CS0 – CS3) is configured in shift mode so that a max. address range of 64 MByte can be addressed per chip select. The *EXTENDED_CONFIG.ASYNC_ADDR_MODE* bit is set in the EMC interface by the primary boot loader (see 2.3.1.5.1).

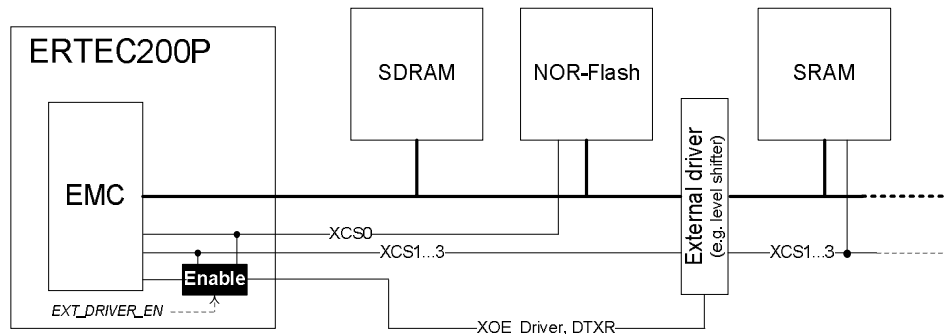
This parameter assignment allows you to address:

- 16 MByte with a data bus width of 8 bits (byte)
- 32 MByte with a data bus width of 8 bits (byte)
- 64 MByte with a data bus width of 32 bits (word)

2.3.5.4.4 Control of an External Driver

The signals DTXR and XOE_DRIVER at the EMC are used to control external drivers (timing, see 2.3.5.5.1 and 2.3.5.5.2). These signals can only be used for asynchronous memory. The use of these signals is optional.

As the signals for controlling the drivers are always activated irrespective of the address range of the asynchronous EMC interface (CS0 – CS3) and not on a chip select-specific basis, the driver becomes active even when memory without/before the driver is accessed; in this case two devices would drive the data bus for the EMC.



To prevent this from happening, a chip select-specific enable for driver control can be configured with the SCRB register *EXT_DRIVER_EN* (see 2.3.10.9.16).

Driver control is disabled after a reset. To allow the flash block in the NOR flash boot (see 0) to be operated both before and after the driver, the driver control information (*EXT_DRIVER_DISABLE_CS0*) for address range: 0x3000_0000 – 0x33FF_FFFF (chip select 0) is also latched to *EXT_DRIVER_EN.CS0_ENABLE* over EMC pin XRDY_BF when the reset is cleared. The integrated pull-up means that control for an external driver is deactivated by default and needs to be activated with a pull-down for the module.

2.3.5.4.5 QVZ Acknowledgement Delay

If the ready signal does not respond in time with asynchronous RAMs (...ROMs, all devices with ready), an interrupt is generated and the address that has generated the error is saved. This is known as QVZ (*Quittungsverzug*, acknowledgement delay) monitoring. The signals ASYNC_WAIT and BF_RDY are monitored. The QVZ interrupt (*EMC_QVZ_INT*, see 2.3.2.14) is triggered synchronously to the AHB clock.

All read and write access to external asynchronous RAMs interrupts the SDRAM refresh cycles until access is ended. The value *MAX_EXT_WAIT* is the limit for one complete instance of burst access on the RAM interface, not at the AHB bus. For example, 32-bit read access to an 8-bit block is converted to a burst with 4 reads. All 4 reads must be carried out within the set time window (*MAX_EXT_WAIT*).

The memory is addressed at the address of EMC Peripheral Bank 0. With NAND flash with a **page size of 512 bytes**, GPIO31 is used as chip select (CE), i.e. there must be no XCS_PER0 signal connected for selection at the blocks. This is important as NAND flashes with a page size of 512 bytes do not permit any change in the CS signal during address byte transfer and would otherwise not respond correctly.

2.3.5.5 Timing of the Asynchronous Memory Interfaces

The asynchronous interface can be activated by writing a 1 to the *ADB* bit of the *EXTENDED_CONFIG* register. The data bus is then kept at 1 for one cycle after access. With pull-up resistors, this speeds up capacity reloading (with wiring). The internal pull-ups drive the data bus after the active phase. No external pull-ups are therefore required. For further explanation, SETUP, STROBE and HOLD phases are defined for asynchronous access.

The figures below detail the individual phases in read and write access.

2.3.5.5.1 Read Access

The SETUP phase starts with an active CS signal and ends when the output enable signal becomes active. The STROBE phase then starts. The STROBE phase lasts until the output enable signal is deactivated again. This then starts the hold phase. The hold phase ends when the CS signal is deactivated.

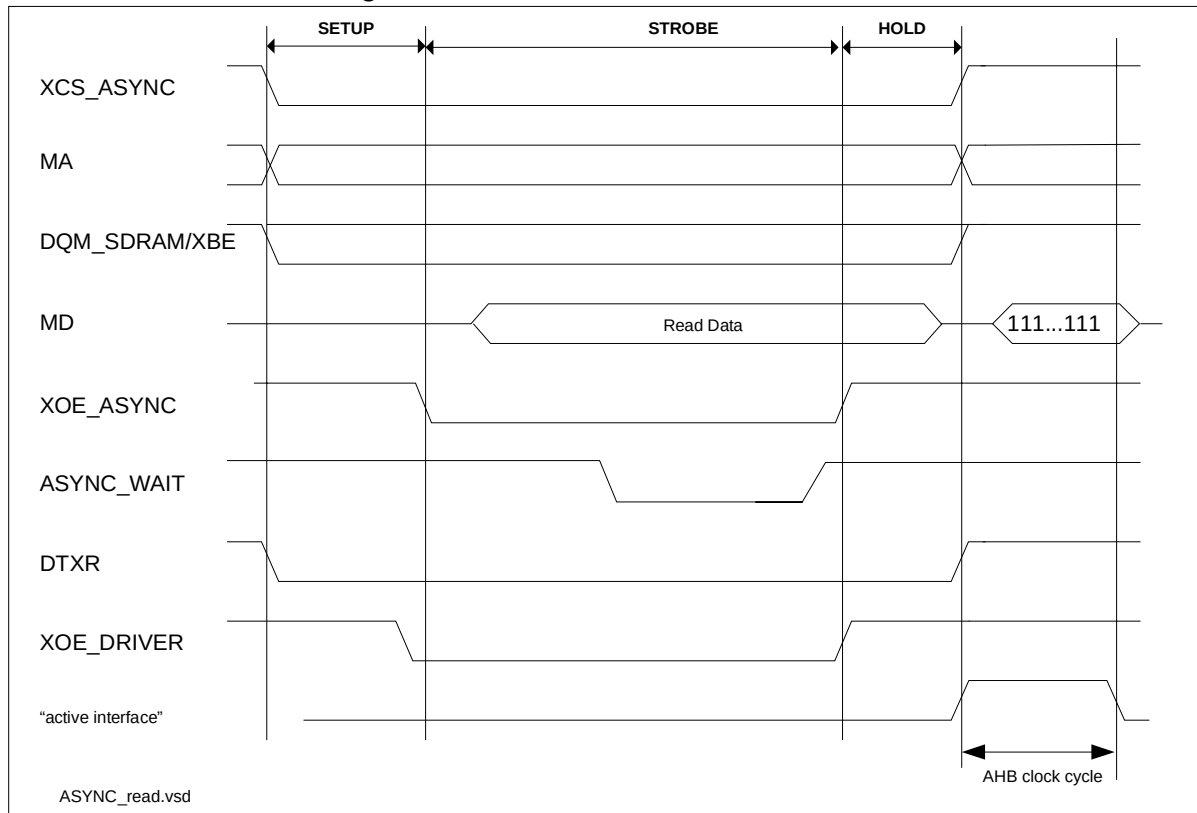


Figure 14: EMC, Notation definition for a read access (ASYNC)

2.3.5.5.2 Write Access

CS and WE signals are key to the individual phases in write access. See the write timing below.

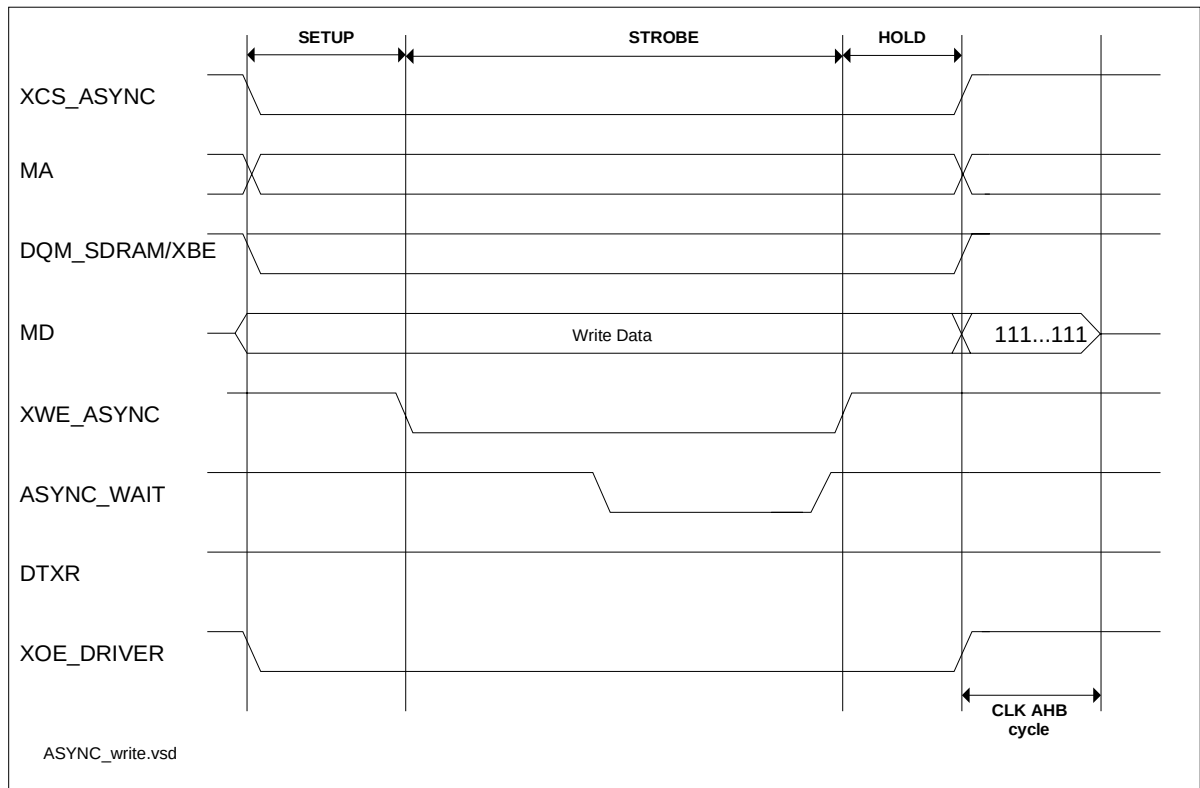


Figure 15: EMC, Notation definition for a write access (ASYNC)

For the asynchronous interface (SRAM), the data are valid until the end of the HOLD phase.

2.3.5.6 Application examples

Following figure shows an example of connections between EMC and external SDRAM devices:

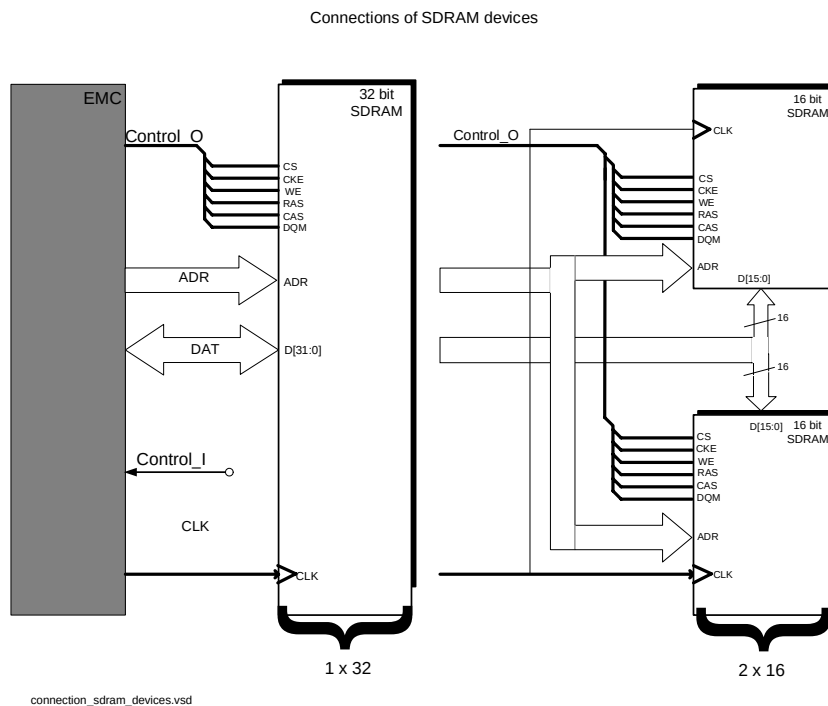


Figure 16: EMC, Application Example SDRAM

Following figure shows an example of connections between EMC and external SDRAM device in combination with an external driver to separate SDRAM- and SRAM-like devices.

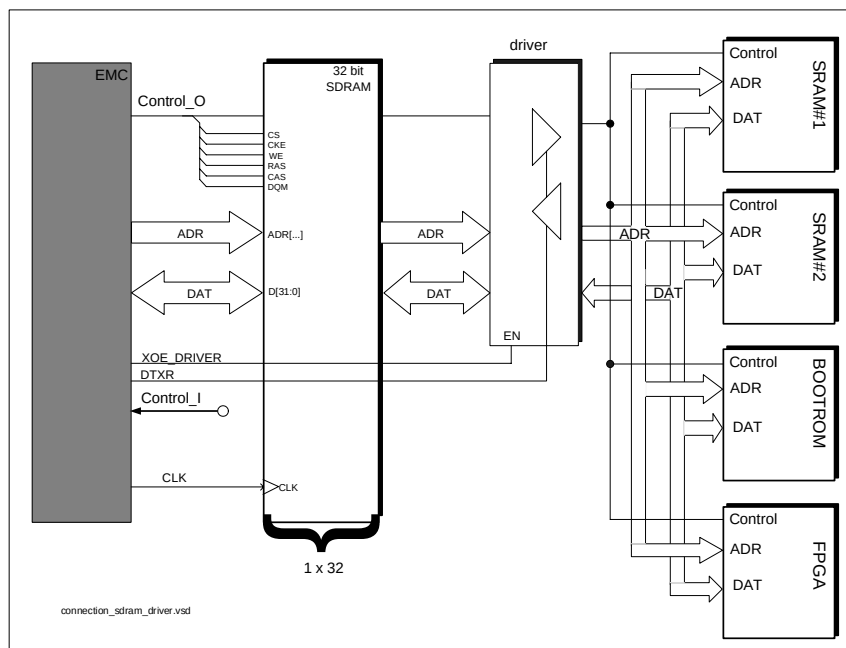


Figure 17: EMC, Application example: Combination of SDRAM and asynchronous Interfaces

2.3.5.7 Interface signals

The following figure lists the interface signals. Details to all interface signals can be found in the addendum.

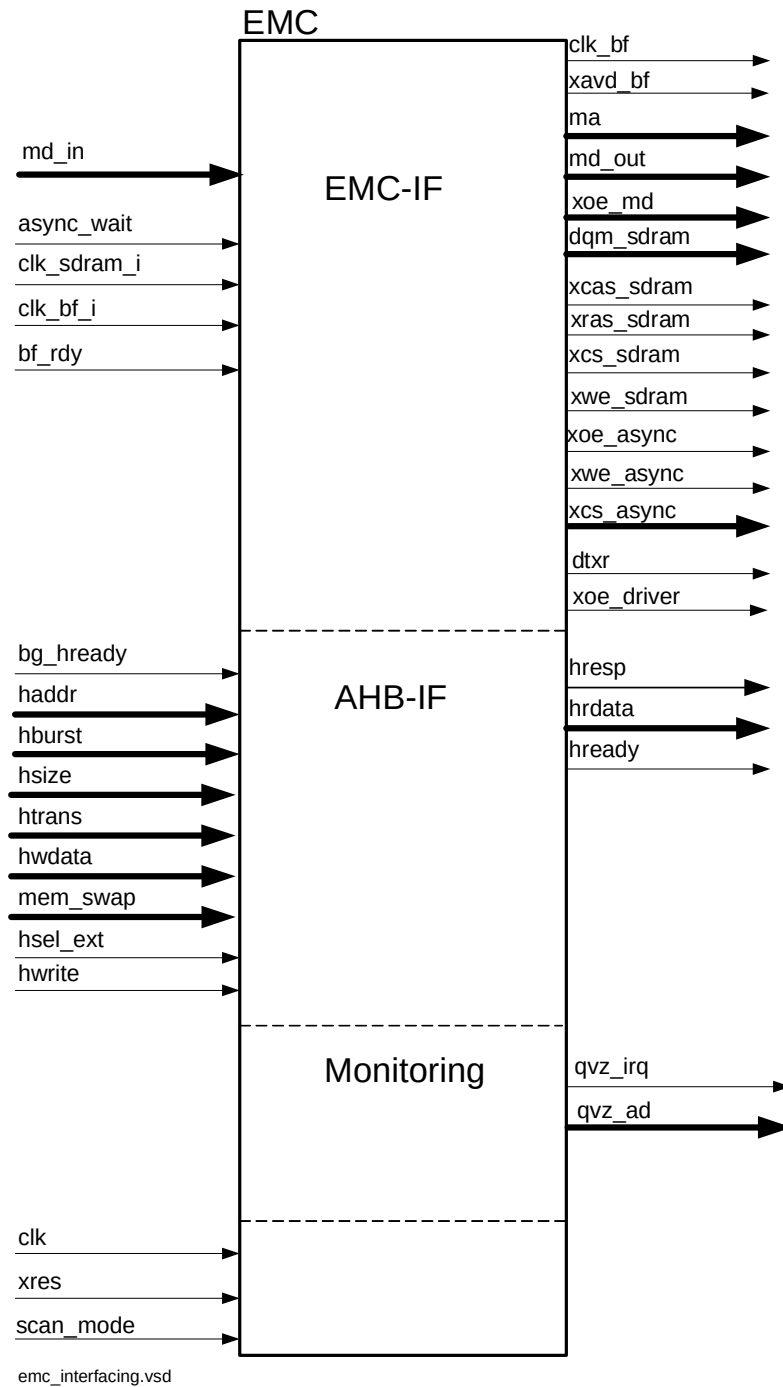


Figure 18: EMC, Interface Signals

The following table shows the relationship of the interface signals and the connected memory device:

Interface Signal	SDRAM device (incl. Mobile SDRAM)	SRAM device	Burst Flash ROM
	supported in ERTEC 200P	supported in ERTEC 200P	supported in ERTEC 200P
MD_IN – read data	x	x	x
CLK_SDRAM_I	x	use 125MHz	
CLK_BF_I			x
ASYNC_WAIT		x	
BF_RDY			x
CLK_BF_O			x
MA – memory address	(15:0)	x	x
MD_OUT – write data	x	x	x
XOE_MD	x	x	x
DQM_SDRAM	x	XBE	XBE
XCS_SDRAM	x		
XRAS_SDRAM	x		
XCAS_SDRAM	x		
XWE_SDRAM	x		
XCS_ASYNC		x	x
XWE_ASYNC		x	x
XOE_ASYNC		x	x
DTXR		[x]	
XOE_DRIVER		[x]	
XBF_AVDD			x

Table 14: EMC, Connection to the Memory devices

2.3.5.8 Address Mapping

Start_Address	End_Address	Modul/Memory_Name	Interface	Fill_Mode
10D0.0000h	10DF.FFFFh	EMC REGISTER		1 MB
2000.0000h	2FFF.FFFFh	SDRAM		256 MB
3000.0000h	33FF.FFFFh	ASYNC_CHIP_SELECT0		64 MB
3400.0000h	37FF.FFFFh	ASYNC_CHIP_SELECT1		64 MB
3800.0000h	3BFF.FFFFh	ASYNC_CHIP_SELECT2		64 MB
3C00.0000h	3FFF.FFFFh	ASYNC_CHIP_SELECT3		64 MB

Module	Register/Memory	Read	Write	Address
/emc_reg				
	REVISION_CODE	r		0h
	ASYNC_WAIT_CYCLE_CONFIG	r	(w)	4h
	SDRAM_CONFIG	r	(w)(t)(p)	8h
	SDRAM_REFRESH	r(h)	(w)	Ch
	ASYNC_BANK0	r	w	10h
	ASYNC_BANK1	r	w	14h
	ASYNC_BANK2	r	w	18h
	ASYNC_BANK3	r	w	1Ch
	EXTENDED_CONFIG	r	(w)	20h
	AT_ADDR	rh		24h
	LPEMR	r	(w)	28h
	BF_CONFIG	r	(w)	2Ch
	PM_CONFIG	r	(w)	30h
	RECOV_CONFIG	r	(w)	34h

Table 15: EMC – Address map

Note: EMC register address base can be any address 0xXYZ_00000, but X must not be 0, 2, 3.

2.3.5.9 Register description

Note: The parameter "ASYNC_ADDR_MODE" in the EMC-IP is set to '1' by the ERTEC 200P bootcode.

Module: /emc_reg

Register:	REVISION_CODE			Address:	0h	
Bits:	31dt0	Reset value:	04200806h	Attributes:	r	
Description:	Revision Code					
Bit	Identifier	Reset	Attr.	Function / Description		
10dt0	LABEL	006h	r	Release Label		
15dt11	INCREMENT	01h	r	Increment		
18dt16	PATCH	0h	r	Patch Version		
20dt19	PLATFORM	0h	r	Platform code		
31dt21	ID	021h	r	Identification code		

Register:	ASYNC_WAIT_CYCLE_CONFIG			Address:	4h	
Bits:	31dt0	Reset value:	40000080h	Attributes:	r	(w)
Description:	Async. Wait Cycle Configuration register					
Bit	Identifier	Reset	Attr.	Function / Description		
19dt0	MAX_EXT_WAIT	00080h	r	w	Maximum number of wait cycles. If an access is delayed by WAIT input for more than [(MAX_EXT_WAIT+1) x 16] clocks, the access is completed.	
23dt20	reserved_3	0h	r		reserved	
24	BE_CTRL1	0h	r	w	Byte Enable Control Async. Bank 1: 0 = all Byte Enables low during read 1 = Byte Enable reflect AHB HSIZE during read	
25	BE_CTRL2	0h	r	w	Byte Enable Control Async. Bank 2: 0 = all Byte Enables low during read 1 = Byte Enable reflect AHB HSIZE during read	
26	BE_CTRL3	0h	r	w	Byte Enable Control Async. Bank 3: 0 = all Byte Enables low during read 1 = Byte Enable reflect AHB HSIZE during read	
27	BE_CTRL4	0h	r	w	Byte Enable Control Async. Bank 4: 0 = all Byte Enables low during read	

					1 = Byte Enable reflect AHB HSIZE during read
29dt28	reserved_2	0h	r		reserved
30	WAIT_POLARITY	1h	r	w	WAIT input polarity 0 = low active 1 = high active
31	reserved_1	0h	r		reserved

Register:		SDRAM_CONFIG				Address:		8h	
Bits:		31dt0		Reset value:		00002020h		Attributes: r (w)(t)(p)	
Description:			SDRAM Configuration register A write on this register starts a Load Mode Register sequence. This register can only be written after the SDRAM Initialization sequence is done (Bit 29 of register SDRAM_REFRESH).						
Bit	Identifier			Reset	Attr.		Function / Description		
2dt0	PAGE_SIZE			0h	r	w	Page Size 000 = 8 Column address lines 001 = 9 Column address lines 010 = 10 Column address lines 011 = 11 Column address lines 100-111 = reserved		
3	reserved_1			0h	r	w	reserved		
6dt4	SDRAM_BANKS			2h	r	w	Number of banks inside SDRAM 000 = 1 Bank 001 = 2 Bank 010 = 4 Bank 011-111 = reserved		
7	reserved_2			0h	r	w	reserved		
10dt8	ROW_SIZE			0h	r	wpt	Row Size 000 = 8 Row address lines 001 = 9 Row address lines 010 = 10 Row address lines 011 = 11 Row address lines 100 = 12 Row address lines 101 = 13 Row address lines 110 = 14 Row address lines 111 = 15 Row address lines Write protected as long as bit29 in SDRAM_REFRESH = 0. A write starts a Mode Register Set to external SDRAM.		
12dt11	reserved_3			0h	r		reserved		

13	CL	1h	r	wpt	CAS Latency 0 = 2 clocks 1 = 3 clocks Write protected as long as bit29 in SDRAM_REFRESH = 0. A write starts a Mode Register Set to external SDRAM.
31dt14	reserved_4	00000h	r		reserved

Register:	SDRAM_REFRESH				Address:	Ch	
Bits:	31dt0	Reset value:	00000190h		Attributes:	r(h)	(w)
Description:		SDRAM Refresh Control register					
Bit	Identifier	Reset	Attr.		Function / Description		
12dt0	REFRESH_RATE	0190h	r	w	Refresh Rate Number of clocks between 2 SDRAM Refresh cycles = REFRESH_RATE + 1		
28dt13	reserved_1	0000h	r		reserved		
29	INIT_DONE	0h	rh		SDRAM Initialization done 0 = default 1 = SDRAM initialization after Hardware Reset is done (needs 28670 clocks)		
30	AT	0h	rh		Async. Timeout 0 = default 1 = Async. access was finished because Wait Cycle Counter expired.		
31	reserved_2	0h	r		reserved		

Register:	ASYNC_BANK0					Address:	10h	
Bits:	31dt0		Reset value:	3FFFFFF2h		Attributes:	r	w
Description:		Async. Bank 0 Configuration register						
Bit	Identifier		Reset	Attr.		Function / Description		
1dt0	SIZE		2h	r	w	Bank Size 00 = 8 bit device 01 = 16 bit device 10 = 32 bit device 11 = reserved		
3dt2	MODE		0h	r	w	Mode 00 = SRAM 01 = Pagemode ROM		

					10 = reserved 11 = Burstflash ROM
6dt4	R_HOLD	7h	r	w	Read Hold Cycles Hold Phase of read access lasts (R_HOLD + 1) clocks.
12dt7	R_STROBE	3Fh	r	w	Read Strobe Duration Cycles Strobe Phase of read access (Read Enable = 0) lasts (R_STROBE + 1) clocks. Value 0 is not supported.
16dt13	R_SU	Fh	r	w	Read Setup Cycles Setup Phase of read access lasts R_SU clocks.
19dt17	W_HOLD	7h	r	w	Write Hold Cycles Hold Phase of write access lasts (W_HOLD + 1) clocks.
25dt20	W_STROBE	3Fh	r	w	Write Strobe Duration Cycles Strobe Phase of write access (Write Enable = 0) lasts (W_STROBE + 1) clocks.
29dt26	W_SU	Fh	r	w	Write Setup Cycles Setup Phase of write access lasts W_SU clocks.
30	EW	0h	r	w	Extend Wait mode 0 = WAIT input = don't care 1 = Wait input extends access (valid in SRAM mode only)
31	WSM	0h	r	w	Wait input synchronisation mode 0 = 2 Flipflops 1 = 1 Flipflop

Register:		ASYNC_BANK1				Address:		14h		
Bits:		31dt0	Reset value:		3FFFFFF2h		Attributes:		r	w
Description:			Async. Bank 1 Configuration register							
Bit	Identifier		Reset	Attr.		Function / Description				
1dt0	SIZE		2h	r	w	Bank Size 00 = 8 bit device 01 = 16 bit device 10 = 32 bit device 11 = reserved				
3dt2	MODE		0h	r	w	Mode 00 = SRAM 01 = Pagemode ROM 10 = reserved				

					11 = Burstflash ROM
6dt4	R_HOLD	7h	r	w	Read Hold Cycles Hold Phase of read access lasts (R_HOLD + 1) clocks.
12dt7	R_STROBE	3Fh	r	w	Read Strobe Duration Cycles Strobe Phase of read access (Read Enable = 0) lasts (R_STROBE + 1) clocks. Value 0 is not supported.
16dt13	R_SU	Fh	r	w	Read Setup Cycles Setup Phase of read access lasts R_SU clocks.
19dt17	W_HOLD	7h	r	w	Write Hold Cycles Hold Phase of write access lasts (W_HOLD + 1) clocks.
25dt20	W_STROBE	3Fh	r	w	Write Strobe Duration Cycles Strobe Phase of write access (Write Enable = 0) lasts (W_STROBE + 1) clocks.
29dt26	W_SU	Fh	r	w	Write Setup Cycles Setup Phase of write access lasts W_SU clocks.
30	EW	0h	r	w	Extend Wait mode 0 = WAIT input = don't care 1 = Wait input extends access (valid in SRAM mode only)
31	WSM	0h	r	w	Wait input synchronisation mode 0 = 2 Flipflops 1 = 1 Flipflop

Register:	ASYNC_BANK2					Address:	18h		
Bits:	31dt0		Reset value:		3FFFFFF2h		Attributes:	r	w
Description:		Async. Bank 2 Configuration register							
Bit	Identifier		Reset	Attr.		Function / Description			
1dt0	SIZE		2h	r	w	Bank Size 00 = 8 bit device 01 = 16 bit device 10 = 32 bit device 11 = reserved			
3dt2	MODE		0h	r	w	Mode 00 = SRAM 01 = Pagemode ROM 10 = reserved 11 = Burstflash ROM			

6dt4	R_HOLD	7h	r	w	Read Hold Cycles Hold Phase of read access lasts (R_HOLD + 1) clocks.
12dt7	R_STROBE	3Fh	r	w	Read Strobe Duration Cycles Strobe Phase of read access (Read Enable = 0) lasts (R_STROBE + 1) clocks. Value 0 is not supported.
16dt13	R_SU	Fh	r	w	Read Setup Cycles Setup Phase of read access lasts R_SU clocks.
19dt17	W_HOLD	7h	r	w	Write Hold Cycles Hold Phase of write access lasts (W_HOLD + 1) clocks.
25dt20	W_STROBE	3Fh	r	w	Write Strobe Duration Cycles Strobe Phase of write access (Write Enable = 0) lasts (W_STROBE + 1) clocks.
29dt26	W_SU	Fh	r	w	Write Setup Cycles Setup Phase of write access lasts W_SU clocks.
30	EW	0h	r	w	Extend Wait mode 0 = WAIT input = don't care 1 = Wait input extends access (valid in SRAM mode only)
31	WSM	0h	r	w	Wait input synchronisation mode 0 = 2 Flipflops 1 = 1 Flipflop

Register:	ASYNC_BANK3					Address:	1Ch		
Bits:	31dt0		Reset value:		3FFFFFF2h		Attributes:	r	w
Description:		Async. Bank 3 Configuration register							
Bit	Identifier		Reset	Attr.		Function / Description			
1dt0	SIZE		2h	r	w	Bank Size 00 = 8 bit device 01 = 16 bit device 10 = 32 bit device 11 = reserved			
3dt2	MODE		0h	r	w	Mode 00 = SRAM 01 = Pagemode ROM 10 = reserved 11 = Burstflash ROM			
6dt4	R_HOLD		7h	r	w	Read Hold Cycles			

					Hold Phase of read access lasts (R_HOLD + 1) clocks.
12dt7	R_STROBE	3Fh	r	w	Read Strobe Duration Cycles Strobe Phase of read access (Read Enable = 0) lasts (R_STROBE + 1) clocks. Value 0 is not supported.
16dt13	R_SU	Fh	r	w	Read Setup Cycles Setup Phase of read access lasts R_SU clocks.
19dt17	W_HOLD	7h	r	w	Write Hold Cycles Hold Phase of write access lasts (W_HOLD + 1) clocks.
25dt20	W_STROBE	3Fh	r	w	Write Strobe Duration Cycles Strobe Phase of write access (Write Enable = 0) lasts (W_STROBE + 1) clocks.
29dt26	W_SU	Fh	r	w	Write Setup Cycles Setup Phase of write access lasts W_SU clocks.
30	EW	0h	r	w	Extend Wait mode 0 = WAIT input = don't care 1 = Wait input extends access (valid in SRAM mode only)
31	WSM	0h	r	w	Wait input synchronisation mode 0 = 2 Flipflops 1 = 1 Flipflop

Register:	EXTENDED_CONFIG				Address:	20h	
Bits:	31dt0	Reset value:	01F70000h		Attributes:	r	(w)
Description:		Extended Configuration register					
Bit	Identifier	Reset	Attr.		Function / Description		
6dt0	reserved_1	00h	r	w	reserved		
7	AT_EN	0h	r	w	Async. Timeout Enable 0 = disabled 1 = enabled		
8	SDRAM_DW	0h	r	w	SDRAM interface data width 0 = 32 bit 1 = 16 bit		
9	MOBILE_SDRAM	0h	r	w	Mobile SDRAM Mode 0 = disabled (1 SDRAM mode register) 1 = enabled (2 SDRAM mode register)		
10	ASYNC_ADDR_MODE	0h	r	w	Async. Address Output Mode		

					0 = no address shift (ERTEC 200 compatible) MA(23:0) = HADDR(23:0) 1 = address shift depending on configured memory datawidth datawidth = 8 bit : MA(23:0) = HADDR(23:0) datawidth = 16 bit : MA(23:0) = HADDR(24:1) datawidth = 32 bit : MA(23:0) = HADDR(25:2)
13dt11	reserved_2	0h	r	w	reserved
15dt14	TRCD	0h	r	w	TRCD (RAS to CAS delay) 00 = 2 clocks 01 = 3 clocks 10 = 4 clocks 11 = reserved
18dt16	SDRAM_BL	7h	r	w	SDRAM Burst Length This value is written into SDRAM Mode Register. 000 = 1 (32 bit SDRAM interface only) 001 = 2 (16 bit SDRAM interface only) 010 = 4 (not supported) 011 = 8 100-110 = reserved 111 = continuous/full page (recommended)
19	reserved_3	0h	r		reserved
23dt20	TRFC	Fh	r	w	AUTO REFRESH period (cycle) time Minimum time between to AUTO REFRESH commands in clocks 0x0 = reserved 0x1 = 4 clocks : 0xF = 18 clocks
24	ADB	1h	r	w	active data bus 0 = disabled 1 = enabled
25	reserved_4	0h	r	w	reserved
26	BFODM	0h	r	w	BurstflashROM Output Delay Mode 0 = XBF_AVD is delayed with register clocked by input BurstflashROM clock 1 = XBF_AVD is delayed with delay buffer
27	SODM	0h	r	w	Output Delay Mode 0 = outputs are delayed with register clocked by input SDRAM clock 1 = outputs are delayed with delay buffers

28	TM4	0h	r	w	Test Mode bit 4 0 = normal operation 1 = async write data outputs toggle 1 clock after address outputs
29	TM2	0h	r	w	Test Mode bit 2 0 = normal operation 1 = all SDRAM reads/writes are of type miss (32 bit SDRAM only)
30	TM1	0h	r	w	Test Mode bit 1 0 = normal operation 1 = SDRAM initialization delay after reset is cleared
31	reserved_5	0h	r	w	reserved

Register:	AT_ADDR			Address:	24h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Async. Timeout Address register Contains after asyc. timeout address of the access, which was finished with timeout.					

Register:	LPEMR				Address:	28h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r (w)
Description:		Low Power Extended Mode Register This value is written into Extended Mode Register of Mobile SDRAM					
Bit	Identifier		Reset	Attr.		Function / Description	
12dt0	LPEMR		0000h	r	w	This value is written into Extended Mode Register of Mobile SDRAM. See Mobile SDRAM datasheet.	
31dt13	reserved 1		00000h	r		reserved	

Register:	BF_CONFIG				Address:	2Ch	
Bits:	31dt0	Reset value:	00000000h		Attributes:	r	(w)
Description:		Burstflash ROM Configuration register					
Bit	Identifier	Reset	Attr.		Function / Description		
0	BF_CLK_F	0h	r		Clock Frequency 0 = Burstflash clock = HCLK/2 1 = Burstflash clock = HCLK (HCLK = AHB clock)		
1	BF_CLK_EN	0h	r		Clock Enable		

				0 = Burstflash Clock always at fixed level defined in bit 3 1 = Burstflash clock toggles
2	AVD_MODE	0h	r	Address Valid output mode 0 = AVD always low 1 = AVD toggles
3	BF_CDV	0h	r	Clock Disable Value 0 = Burstflash Clock fixed level = 0 1 = Burstflash Clock fixed level = 1
7dt4	reserved_1	0h	r	reserved
10dt8	RM	0h	r	Read Mode 000 = Continuous 001 = 8-word linear without wrap around 010 = 16-word linear without wrap around 011 = 32-word linear without wrap around 100 = reserved 101 = 8-word linear with wrap around 110 = 16-word linear with wrap around 111 = 32-word linear with wrap around
13dt11	reserved_2	0h	r	reserved
14	RDY_DELAY	0h	r	Ready Delay Mode 0 = RDY active with data 1 = RDY active one clock cycle before data
15	SYNC_READ	0h	r	Set Device Read Mode 0 = Asynchronous Read Mode 1 = Synchronous Read (Burst Mode) enabled
19dt16	AVD_DELAY	0h	r	AVD delay 0000 = 1 AHB clock .. 1111 = 16 AHB clocks Delay after falling edge of chip select to falling edge of AVD
22dt20	AVD_PW	0h	r	AVD pulse width 000 = 1 AHB clock .. 111 = 8 AHB clocks
23	reserved_3	0h	r	reserved
31dt24	reserved_4	00h	r	reserved

Register:	PM_CONFIG			Address:	30h	
Bits:	31dt0	Reset value:	0000003Fh	Attributes:	r	(w)
Description:	Pagemode ROM Configuration register					

Bit	Identifier	Reset	Attr.		Function / Description
5dt0	PACC	3Fh	r	w	Page Access time Strobe Phase of page read access lasts (PACC + 1) clocks.
6	PAGE_MODE	0h	r	w	0 = 4 word page 1 = 8 word page
7	reserved_1	0h	r	w	reserved
31dt8	reserved_2	000000h	r		reserved

Register:		RECOV_CONFIG				Address:		34h	
Bits:		31dt0		Reset value:		00000000h		Attributes: r (w)	
Description:			Recovery Phase Configuration register						
Bit	Identifier			Reset	Attr.		Function / Description		
3dt0	RECOV0			0h	r	w	Async. Bank 0 Recovery Phase 0000 = 1 clock (WR access) / 2 clocks (RD access) 0001 = 1 clock (WR access) / 2 clocks (RD access) 0010 = 2 clocks 0011 = 3 clocks 0100 = 4 clocks ... 1111 = 15 clocks		
7dt4	RECOV1			0h	r	w	Async. Bank 1 Recovery Phase 0000 = 1 clock (WR access) / 2 clocks (RD access) 0001 = 1 clock (WR access) / 2 clocks (RD access) 0010 = 2 clocks 0011 = 3 clocks 0100 = 4 clocks ... 1111 = 15 clocks		
11dt8	RECOV2			0h	r	w	Async. Bank 2 Recovery Phase 0000 = 1 clock (WR access) / 2 clocks (RD access) 0001 = 1 clock (WR access) / 2 clocks (RD access) 0010 = 2 clocks 0011 = 3 clocks 0100 = 4 clocks ... 1111 = 15 clocks		

15dt12	RECOV3	0h	r	w	Async. Bank 3 Recovery Phase 0000 = 1 clock (WR access) / 2 clocks (RD access) 0001 = 1 clock (WR access) / 2 clocks (RD access) 0010 = 2 clocks 0011 = 3 clocks 0100 = 4 clocks ... 1111 = 15 clocks
31dt16	reserved_1	0000h	r		reserved

2.3.6 Host Interface – Parallel (XHIF)

Parallel external host access is processed over the host interface.

A local bus unit (2x XHIF-IP) is implemented for access to a host system connected in parallel. The local bus unit (XHIF) has two internal XHIF-IPs and supports access from an external 16-bit and 32-bit host CPU. The complete ERTEC 200P address range can be accessed by the external host system over the host interface (over a max. of 8 configurable address windows (pages) of 256 bytes each – 1 / 2 MByte (see below)).

The host interface is configured in the "Parameter" module with APB access.. The basic configuration (data width, ready polarity, read/write line) for the XHIF is set (jointly for the two IP XHIF_0/XHIF_1) with a PowerOn reset over HW_ConfigPins 5...3. Subsequent changes can be made to this configuration by the ARM926EJ-S in the XHIF_CONTROL register over the APB interface. The external host should not access this register, as this could cause undefined states at the host interface. The basic configuration therefore cannot subsequently be changed by the external host.

2.3.6.1 Block Diagram of the Host Interface

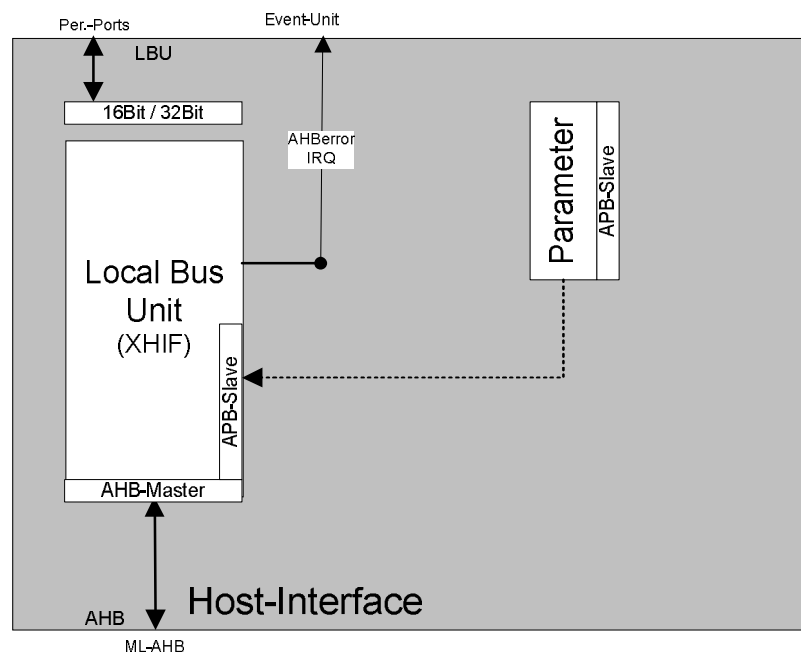


Figure 19: Block diagram of the host interface

The figure below shows the structure of the XHIF, which consists of two individual XHIF modules (XHIF_0/1) (2x 4 pages of 2 MByte each).

The XHIF module (XHIF_0 or XHIF_1) is selected with the XHIF_SEG_2 pin. The XHIF interface is switched to the selected XHIF module. Access to the AHB interface is with arbitration. The pins XHIF_SEG_0 and XHIF_SEG_1 address the corresponding pages of an XHIF module.

The XHIF configuration (XHIF_ACC_Mode, XHIF_POL_RDY, XHIF_CPU_Width) is the same for both XHIF modules and is connected straight to both modules (source: Config-Pins or XHIF_Control register). The XHIF interrupt (XHIFerr_IRQ) is connected from each module with an OR logic operation and connected externally (to the event unit in PER-IF).

2.3.6.2 Function Description (XHIF)

XHIF0, 1 is an interface module that enables an external CPU to access the AHB bus as master over a memory interface. The basic features of the interface module are as follows:

32/16 and 8-bit CPUs can be connected with an address range (A0 to A22) of 10 to 23 address bits (1k to 8M) (**however, only 16/32-bit CPUs are supported in the ERTEC 200P**) Addresses A21 (XHIF_SEG0) and A22 (XHIF_SEG1) are used to select the 4 possible pages with an address range of 256 bytes to 2 MByte.

The interface module can be configured by the external CPU over a 16-bit or 32-bit data bus.

The interface module can also always be configured with the internal ARM926 APB bus. XHIF interface timing is asynchronous to the AHB bus clock.

Areas in the AHB address range (pages) can be configured for access by the external CPU (XHIF internal paging register). ERTEC 200P address mapping (see 2.4.4) is designed to ensure that all relevant AHB address ranges can be configured over the 8 pages (8 x 2MByte) during ERTEC 200P operation.

Below is an example of possible page configuration:

	XHIF address range	AHB module	AHB address range
XHIF0	Page 0	APB peripherals	1 MByte
	Page 1	Profinet IP	2 MByte
	Page 2	Peripheral interface	1 MByte
	Page 3	GDMA	1 MByte
XHIF1	Page 4	Reserved	2 MByte
	Page 5	SDRAM (EMC)	2 MByte
	Page 6	D-TCM (ARM926)	2 MByte
	Page 7	SRAM (EMC)	2 MByte

The AHB master interface is based on the AHB standard 2.0 AHB-Lite protocol.

The XHIF module has only one clock source, which must correspond to the frequency of the AHB/APB system connected.

Single read and write commands are supported, but burst mode is not.

The XHIF module only supports little-endian mode (least significant byte at the lower memory address)

The address windows in the page registers (offset, range and buffer mode) can be configured by the ARM926 over the APB provided the XHIF is not yet active (enabled). The external host uses the register chip select XHIF_XCS_R for configuring the page registers.

The external host can also configure the page registers over the APB interface of the XHIF. In this case, however, the page for the APB address range should not be changed, as further changes to the page registers over APB will otherwise not be possible.

Important: This type of page configuration has not been released in the XHIF spec., but was successfully used in previous ASIC (ERTEC 200P).

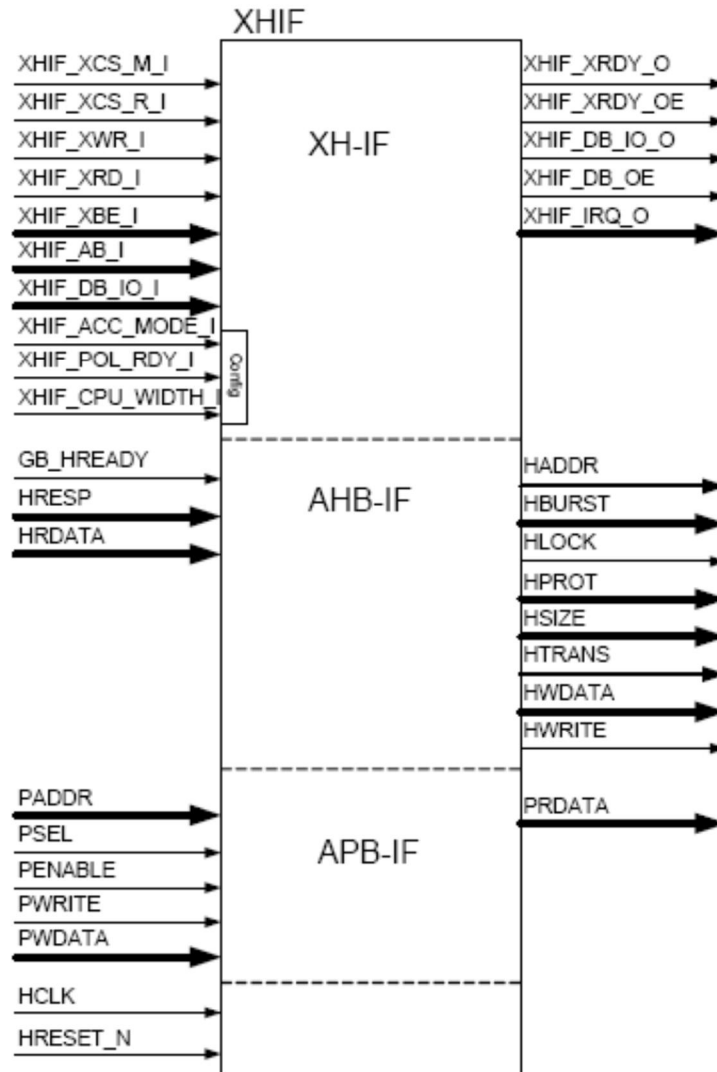


Figure 20: XHIF, Symbol and Signals

Two functions have been implemented with the 'XHIF_XCS_R_A20' input pin in the light of restrictions on the ballout and package design of the ERTEC 200P. They allow the signal at this input to be used:

as chip select for page configuration over the external host OR

as address line A20 (→page address range increased to 2 MByte)

Just how this input pin is used in the XHIF modules is set in the SCR_B register 'XHIF_Mode' with the 'XHIF_Mode' bit:

- XHIF_Mode = 0b: Page register chip select XHIF_XCS_R → Page = max. 1 MByte

- XHIF_Mode = 1b: Address line XHIF_A20 → Page = max. 2 MByte

There are therefore 2 use cases with different max. page sizes.

8 pages with a 1 MByte address range each:

This is the default setting (XHIF_Mode = '0'). The external host and its address lines are wired to the XHIF interface as follows:

XHIF interface (ERTEC 200P)	Ext. host (ext. configuration)
XHIF_A1	A1 with 16-bit interface fixed at '0' with 32-bit interface
XHIF_A2 ... XHIF_A19	A2...A19
XHIF_XCS_R_A20	XCS_R
XHIF_SEG_0	A20
XHIF_SEG_1	A21
XHIF_SEG_2	A22

The 8 address windows (1 MByte each) are generally assigned as follows: 2x PN-IP, 1x PER-IF, 2x EMC-SDRAM, 1x GDMA, 1x dyn. (to ARM926 D-TCM or EMC-SRAM or APB peripherals or EMC register).

8 pages with a 2 MByte address range each:

(XHIF_Mode = '1') must be set for this setting. The external host and its address lines are wired to the XHIF interface as follows:

XHIF interface (ERTEC 200P)	Ext. host (ext. configuration)
XHIF_A1	A1 with 16-bit interface, fixed at '0' with 32-bit interface
XHIF_A2 ... XHIF_A19	A2...A19
XHIF_XCS_R_A20	A20
XHIF_SEG_0	A21
XHIF_SEG_1	A22
XHIF_SEG_2	A23

The procedure for XHIF configuration is as follows:

The XHIF interface with 8 pages of 1 MByte each and page register configuration over the external host (XHIF_XCS_R) is available after a ERTEC 200P reset.

Page register configuration from the host is with A20 = '0' (XHIF_XCS_R = '0').

The page from which the APB area can be accessed is accessed from the host with A20 = '1' (XHIF_XCS_R = '1') and XHIF_XCS_M = '0'. The register XHIF_Mode = '1' is set in the SCRb block.

From now on, A20 acts from the host perspective as an address line (XHIF_A20) and 2 MByte of address range are therefore available for each page.

The XHIF interface can be operated at either 3.3 V or 1.8 V. The IO pads are connected separately with VDD_XHIF (either 1.8 V or 3.3 VM; no other setting possible) and supplied with GND. This supply system includes all 64 IO pads of the host interface. One use case for 1.8 V is the connection of two ERTEC 200P (see Figure 10). The ERTEC 200P EMC interface is always operated at VDD=1.8 V. The connection must therefore also be 1.8 V. The driver capacity of the pad cell is 25pF with 1.8 V and 35pF with 3.3 V. The max. board capacities are still to be defined.

The driver power of the 1.8 V XHIF pads in the ERTEC 200P is set to 4 mA after power on. An external host processor should increase this to 9mA for the XHIF pads (configuration in DRIVE95_32GPIO register, see 2.3.10.9.22) before its first read access to the XHIF interface. 4 mA is enough for the XHIF_XRDY pin as it is wired straight from the host to the input pin and only needs to recharge this one load.

2.3.6.2.1 AHB Master Interface

The 32-bit data bus supports the AHB-Lite protocol. Erroneous access is indicated by an interrupt (AHBError_INT). The lock functionality is not used. Burst access is not supported; only single access over the host interface is permitted.

(The following AHB signals are not used, but set static to HLOCK = "0", HPROT = "0001" (i.e. DATA_ACCESS), HBURST = "000").

2.3.6.2.2 APB Slave Interface

The HostIF module contains an APB slave interface with a 32-bit data bus and APB protocol. The 8 pages and the serial interface can be configured over this slave interface. XHIF configuration parameter assignment is also possible in the HostIF with *XHIF_CONTROL*.

2.3.6.2.3 XHIF Configuration

The XHIF configurations (data width, ready polarity and read/write line) for the XHIF-IP are set with the ConfigPins(5:3) and latched to *SCRB_CONFIG_REG* when the PowerOn reset is cleared. ARM926 can also assign parameters for XHIF configuration by changing the XHIF configurations (data width, ready polarity or read/write line) in *SCRB_CONFIG_REG* over the APB interface. The external host should not access *SCRB_CONFIG_REG*, as this could cause undefined states at the host interface.

ConfigPin(5) = A[22] = XHIF_CPU_WIDTH(0)	ConfigPin(4) = A[21] = not XHIF_POL_RDY	ConfigPin(3) = A[20] = not XHIF_ACC_MODE	XHIF interface
0	0	0	16-bit, XRDY high-active, XWR only
0	0	1	16-bit, XRDY high-active, RD-WR separate
0	1	0	16-bit, XRDY low-active, XWR only
0	1	1	16-bit, XRDY low-active, RD-WR separate
1	0	0	32-bit, XRDY high-active, XWR only
1	0	1	32-bit, XRDY high-active, RD-WR separate
1	1	0	32-bit, XRDY low-active, XWR only
1	1	1	32-bit, XRDY low-active, RD-WR separate

Figure 21: XHIF interface adjustment

Note:

XHIF_CPU_WIDTH(1) is fixed at "0" as an external host with an 8-bit data interface is not supported.

Description of configuration pins

XHIF_POL_RDY = 0: The XHIF module signals to the host that access with XHIF_XRDY = 0 is complete (ready).

1: The XHIF module signals to the host that access with XHIF_XRDY = 1 is complete (ready).

XHIF_ACC_MODE = 0: Separate read/write lines are provided
1: A combined read/write signal is used.

XHIF_CPU_WIDTH0: 16-bit data bus of the external host

01: 32-bit data bus

10: 8-bit data bus

11: Not permitted

The configuration signals are synchronized with the XHIF clock.

2.3.6.2.4 Application Information

The XHIF configurations (16/32-bit data width, ready polarity and read/write line) for the XHIF-IP are set with a PowerOn reset over ConfigPins 5...3. Subsequent changes can be made to these configurations by the ARM926EJ-S in the *XHIF_CONTROL* HostIF register over the APB interface.

XHIF_CONTROL should not be changed by the external host. According to the XHIF-IP specifications, the external host must not address the configuration registers over the APB interface (AHB2APB bridge). ConfigPins 6..0 may **only** be set with a PowerOn reset.

The external host can only address the internal XHIF-IP page registers (offset, range and buffer mode) and XHIF_VERSION over the register chip select XHIF_XCS_R. Only the lowest-value address bits are considered for register selection (address $\geq 0x40h$ are invalid according to the XHIF-IP spec).

Only halfword write/read access to the registers is permitted with a configured data width of 16 bits, and only word write/read access with a configured data width of 32 bits. According to the XHIF-IP spec, byte-specific access is not permitted.

When the memory chip select XHIF_XCS_M is active, write/read access is forwarded straight to the HOSTIF AHB master interface and all aligned access types, byte, halfword and word, are permitted.

If you configure a 32-bit data width, the XHIF_ADR(1) pin must be clamped to '0' for the ERTEC 200P (XHIF_ADR(0) internal static setting '0') so that there are always word addresses internally in the XHIF-IP.

If a data width of 16 bits is configured, the XHIF_ADR(1) pin is required for halfword addressing and is driven by the external host.

Initialization procedure for XHIF-IP after reset according to IP spec:

- Power on reset cleared
- XHIF configuration with ConfigPins 5...3, thereafter only the ARM926EJ-S can change configuration
(access by the external host not yet permitted)
- Page setting by ARM926EJ-S: access over the APB to the relevant registers
or page setting by host: access over the XHIF_XCS_R chip select to the relevant registers

- Write access (HOSTIF-AHB master) by the external host
- The ARM926EJ-S must not change the XHIF configuration (data width, ready polarity or command mode)

once the external host has started write access.

The XHIF_XRDY signal is implemented as a bi-directional buffer. However, the output driver only becomes active with an XHIF CS, i.e. other devices can actively pull the signal when there is no access with XHIF. This basically provides open drain functionality.

XHIF_XRDY must be pulled to "normally ready" with an external pull resistor, i.e. pull-up if XHIF_XRDY = high-active and pull-down if XHIF_XRDY = low-active, so that at the end of access the XHIF_XRDY signal can also be recognized by an external host operating with a lower clock rate than the ERTEC 200P (*background: XHIF_XRDY is only actively driven by the XHIF module for **one clock cycle***).

The XHIF_XRDY signal sends a ready message to the external XHIF host. XHIF_XRDY is implemented with a tristate driver whose Output_Enable does not become active until the start of access (as soon as CS and RD or WR are active) and becomes inactive again at the end of access (one clock cycle after active XHIF_XRDY).

For module design, please note that XHIF_XRDY is a controlled **push-pull** output which requires an external pull resistor in accordance with its polarity (i.e. pull-down if XHIF_XRDY is low-active and vice versa). The pull resistor ensures that XHIF_XRDY is pending at the external host for long enough.

When configuring the external host, please note that XHIF_XRDY briefly appears to the host as active at the start of access because of the pull resistor (it takes up to 11 ns for the pin to be driven by ERTEC 200P). To prevent the external host from interpreting access as acknowledged at this stage, there must be a delay before it evaluates

XHIF_XRDY. When the external host is an ERTEC 200P, this delay would for example be implemented by configuring the appropriate number of "Read/Write Strobe Cycles" in the EMC interface (R_Strobe or W_Strobe parameter in the ASYNC_BANK0-3 EMC registers).

How does the host detect that the ERTEC 200P is ready for host access after the reset phase?

The host recognizes that the ERTEC 200P is ready for host access with the CS_R Page 0 range and offset registers. Page 0 range and offset registers have the default value 0x0000_0000; after configuration by ERTEC 200P, the registers of page 0 must contain values of $\neq$ 0x0000_0000.

This means that the host must poll these registers in the CS-R area after an ERTEC 200P reset until the values initialized by the ERTEC 200P are read; the host is then able to recognize that the ERTEC 200P is ready for host access.

Can deadlock occur during host access?

Problems can only occur during the active reset phase of the ERTEC 200P. They may for example be triggered by a:

- Software reset
- Watchdog reset

If the host accesses the CS_M area during the reset phase, access by ERTEC 200P will not be completed with READY. Access by the host to CS_R is completed during the reset phase, in other words there are no problems there.

Please note the following points to detect / avoid the problem:

- The host must detect an ERTEC 200P reset (at the host, use a GPIO that signals the reset state) and block access to CS_M

- The XHIF master must force READY if READY is not output (i.e. cancel access) and initiate an error signal (interrupt).

2.3.6.3 Address Mapping

Start Address	End Address	Modul/Memory Name
0h	FCh	HOSTIF

Module	Register/Memory	Read	Write	AHB-Master Address	Ext. Host XHIF_XCS_R Address ¹⁾
/HOSTIF					
	HOST_CONTROL	r	w	0h	<i>not accessible</i>
	IP_VERSION	r		40h	
	IP_DEVELOPMENT	r		44h	
	ACCESS_ERROR	r	(w)	48h	
	XHIF_CONTROL	rh	w	70h	
	XHIF_0_P0_RG	r	(w)	80h	0h
	XHIF_0_P0_OF	r	(w)	84h	4h
	XHIF_0_P0_CFG	r	w	88h	8h
	XHIF_0_P1_RG	r	(w)	90h	10h
	XHIF_0_P1_OF	r	(w)	94h	14h
	XHIF_0_P1_CFG	r	w	98h	18h
	XHIF_0_P2_RG	r	(w)	A0h	20h
	XHIF_0_P2_OF	r	(w)	A4h	24h
	XHIF_0_P2_CFG	r	w	A8h	28h
	XHIF_0_P3_RG	r	(w)	B0h	30h
	XHIF_0_P3_OF	r	(w)	B4h	34h
	XHIF_0_P3_CFG	r	w	B8h	38h
	XHIF_0_VERSION	r		BCh	3Ch
	XHIF_1_P0_RG	r	(w)	C0h	40h
	XHIF_1_P0_OF	r	(w)	C4h	44h
	XHIF_1_P0_CFG	r	w	C8h	48h
	XHIF_1_P1_RG	r	(w)	D0h	50h
	XHIF_1_P1_OF	r	(w)	D4h	54h
	XHIF_1_P1_CFG	r	w	D8h	58h
	XHIF_1_P2_RG	r	(w)	E0h	60h
	XHIF_1_P2_OF	r	(w)	E4h	64h
	XHIF_1_P2_CFG	r	w	E8h	68h
	XHIF_1_P3_RG	r	(w)	F0h	70h

	XHIF_1_P3_OF	r	(w)	F4h	74h
	XHIF_1_P3_CFG	r	w	F8h	78h
	XHIF_1_VERSION	r		FCh	7Ch

¹⁾ The addresses also apply to access from an external host with a 32-bit data bus at the XHIF interface with *XHIF_XCS_R* (XHIF chip select register: offset address = 0x0000). An external host with a 16-bit data bus addresses on a 16-bit basis (address: 0h, 2h, 4h, 6h, ...).

2.3.6.4 Register Description

Module: /HOSTIF

Register:		HOST_CONTROL				Address:		0h			
Bits:		0dt0		Reset value:		0h		Attributes:		r	w
Description:			Selection of the serial or parallel path								
Bit	Identifier			Reset		Attr.		Function / Description			
0	CONNECT_MODE			0h		r	w	0 =parallel connection (XHIF) 1 =serial connection (SPI)			

Register:		IP_VERSION				Address:		40h			
Bits:		31dt0		Reset value:		100h		Attributes:		r	
Description:			Metal-fix register for IP Version								
Bit	Identifier			Reset		Attr.		Function / Description			
7dt0	DEBUG_VERSION			00h		r				IP Debug Version	
15dt8	VERSION			01h		r				IP Version	
31dt16	IP_CONFIGURATION			0000h		r				IP Configuration	

Register:		IP_DEVELOPMENT				Address:		44h			
Bits:		31dt0		Reset value:		0h		Attributes:		r	
Description:			Metal-fix register for IP Development								
Bit	Identifier			Reset		Attr.		Function / Description			
31dt0	LABEL_STRUCTURE			00000000h		r		Design Label			

Register:	ACCESS_ERROR					Address:	48h		
Bits:	31dt0		Reset value:		20000000h		Attributes:	r	(w)
Description:			Access error register						
Bit	Identifier		Reset		Attr.		Function / Description		
7dt0	APB_ADDRESS		00h		r		Erroneous APB address in the HostIF		

27dt8	<reserved>	00000h	r		The higher-value APB address bits(31:8) are mirrored in the HostIF
29dt28	APB_SIZE	2h	r		"10": Word access (32-bit) ?? fixed value as only word access possible
30	APB_RD_WR	0h	r		'0': RD access '1': WR access
31	ERROR_LOCK	0h	r	w	Set to '1' by the HW if erroneous APB access is detected. Further HW entries are then blocked. The SW must reset the bit to '0' to enable new entries.

Register:	XHIF_CONTROL				Address:	70h	
Bits:	3dt0	Reset value:	0h		Attributes:	rh	w
Description:	XHIF Interface Settings After reset and when the SCRB CONFIG_REG register is written, these settings are taken from the register						
Bit	Identifier	Reset	Attr.		Function / Description		
0	XHIF_ACC_MODE	0h	rh	w	XHIF Handshake protocol: 0 =Intel Mode, 1 =Motorola Mode		
1	XHIF_POL_RDY	0h	rh	w	'0': XHIF_XRDY is low_aktiv '1': XHIF_XRDY is high_aktiv		
3dt2	XHIF_CPU_WIDTH	0h	rh	w	Data bus width: 00 =8b *, 01 =16b, 10 =32b, 11 =32b * 00 =8b is however not supported by the XHIF interface.		

Register:	XHIF_0_P0_RG					Address:	80h		
Bits:	31dt0		Reset value:		0h		Attributes:	r	(w)
Description:			Range value of the page #0						
Bit	Identifier		Reset	Attr.		Function / Description			
7dt0	XHIF_0_P0_RG_ROSL2		00h	r		Read only value = 0			
21dt8	XHIF_0_P0_RG_RW		0000h	r	w	Read/Write part of the Range register			

31dt22	XHIF_0_P0_RG_ROSL1	000h	r		Read only value = 0
--------	--------------------	------	---	--	---------------------

Register:	XHIF_0_P0_OF				Address:	84h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:		Offset value of the page #0					
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_0_P0_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_0_P0_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:		XHIF_0_P0_CFG				Address:		88h			
Bits:		0dt0		Reset value:		0h		Attributes:		r	w
Description:			Configuration of the buffering mode for page #0								
Bit	Identifier			Reset		Attr.		Function / Description			
0	XHIF_0_P0_CFG_BUFMOD			0h		r	w	16bit access mode of page #0			

Register:	XHIF_0_P1_RG					Address:	90h		
Bits:	31dt0		Reset value:		0h		Attributes:	r	(w)
Description:			Range value of the page #1						
Bit	Identifier		Reset	Attr.		Function / Description			
7dt0	XHIF_0_P1_RG_ROSL2		00h	r		Read only value = 0			
21dt8	XHIF_0_P1_RG_RW		0000h	r	w	Read/Write part of the Range register			
31dt22	XHIF_0_P1_RG_ROSL1		000h	r		Read only value = 0			

Register:	XHIF_0_P1_OF				Address:	94h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:		Offset value of the page #1					
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_0_P1_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_0_P1_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:	XHIF_0_P1_CFG				Address:	98h	
Bits:	0dt0	Reset value:	0h		Attributes:	r	w

Description:		Configuration of the buffering mode for page #1			
Bit	Identifier	Reset	Attr.		Function / Description
0	XHIF_0_P1_CFG_BUFMOD	0h	r	w	16bit access mode of page #1

Register:		XHIF_0_P2_RG				Address:		A0h			
Bits:		31dt0		Reset value:		0h		Attributes:		r	(w)
Description:			Range value of the page #2								
Bit	Identifier			Reset		Attr.		Function / Description			
7dt0	XHIF_0_P2_RG_ROSL2			00h		r				Read only value = 0	
21dt8	XHIF_0_P2_RG_RW			0000h		r		w		Read/Write part of the Range register	
31dt22	XHIF_0_P2_RG_ROSL1			000h		r				Read only value = 0	

Register:		XHIF_0_P2_OF				Address:		A4h			
Bits:		31dt0		Reset value:		0h		Attributes:		r	(w)
Description:			Offset value of the page #2								
Bit	Identifier			Reset		Attr.		Function / Description			
7dt0	XHIF_0_P2_OF_RO			00h		r				Read only value = 0	
31dt8	XHIF_0_P2_OF_RW			000000h		r		w		Read/Write value of the offset register	

Register:		XHIF_0_P2_CFG				Address:		A8h			
Bits:		0dt0		Reset value:		0h		Attributes:		r	w
Description:			Configuration of the buffering mode for page #2								
Bit	Identifier			Reset		Attr.		Function / Description			
0	XHIF_0_P2_CFG_BUFMOD			0h		r	w	16bit access mode of page #2			

Register:		XHIF_0_P3_RG				Address:		B0h			
Bits:		31dt0		Reset value:		0h		Attributes:		r	(w)
Description:			Range value of the page #3								
Bit	Identifier			Reset		Attr.		Function / Description			
7dt0	XHIF_0_P3_RG_ROSL2			00h		r				Read only value = 0	
21dt8	XHIF_0_P3_RG_RW			0000h		r		w		Read/Write part of the Range register	
31dt22	XHIF_0_P3_RG_ROSL1			000h		r				Read only value = 0	

Register:	XHIF_0_P3_OF				Address:	B4h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:		Offset value of the page #3					
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_0_P3_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_0_P3_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:	XHIF_0_P3_CFG					Address:	B8h		
Bits:	0dt0		Reset value:		0h		Attributes:	r	w
Description:			Configuration of the buffering mode for page #3						
Bit	Identifier		Reset		Attr.		Function / Description		
0	XHIF_0_P3_CFG_BUFMOD		0h		r		w		16bit access mode of page #3

Register:	XHIF_0_VERSION				Address:	BCh	
Bits:	31dt0	Reset value:	04000804h	Attributes:	r		
Description:	Metal-fix register for XHIF Version						

Register:	XHIF_1_P0_RG					Address:	C0h		
Bits:	31dt0		Reset value:		0h		Attributes:	r	(w)
Description:			Range value of the page #0						
Bit	Identifier		Reset	Attr.		Function / Description			
7dt0	XHIF_1_P0_RG_ROSL2		00h	r		Read only value = 0			
21dt8	XHIF_1_P0_RG_RW		0000h	r	w	Read/Write part of the Range register			
31dt22	XHIF_1_P0_RG_ROSL1		000h	r		Read only value = 0			

Register:	XHIF_1_P0_OF				Address:	C4h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:		Offset value of the page #0					
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_1_P0_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_1_P0_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:	XHIF_1_P0_CFG					Address:	C8h		
Bits:	0dt0		Reset value:		0h		Attributes:	r	w
Description:			Configuration of the buffering mode for page #0						
Bit	Identifier		Reset	Attr.		Function / Description			
0	XHIF_1_P0_CFG_BUFMOD		0h	r	w	16bit access mode of page #0			

Register:		XHIF_1_P1_RG				Address:		D0h			
Bits:		31dt0		Reset value:		0h		Attributes:		r	(w)
Description:			Range value of the page #1								
Bit	Identifier			Reset		Attr.		Function / Description			
7dt0	XHIF_1_P1_RG_ROSL2			00h		r				Read only value = 0	
21dt8	XHIF_1_P1_RG_RW			0000h		r		w		Read/Write part of the Range register	
31dt22	XHIF_1_P1_RG_ROSL1			000h		r				Read only value = 0	

Register:		XHIF_1_P1_OF				Address:		D4h			
Bits:		31dt0		Reset value:		0h		Attributes:		r	(w)
Description:			Offset value of the page #1								
Bit	Identifier			Reset		Attr.		Function / Description			
7dt0	XHIF_1_P1_OF_RO			00h		r				Read only value = 0	
31dt8	XHIF_1_P1_OF_RW			000000h		r		w		Read/Write value of the offset register	

Register:		XHIF_1_P1_CFG				Address:		D8h			
Bits:		0dt0		Reset value:		0h		Attributes:		r	w
Description:			Configuration of the buffering mode for page #1								
Bit	Identifier			Reset		Attr.		Function / Description			
0	XHIF_1_P1_CFG_BUFMOD			0h		r	w	16bit access mode of page #1			

Register:	XHIF_1_P2_RG					Address:	E0h		
Bits:	31dt0		Reset value:		0h		Attributes:	r	(w)
Description:			Range value of the page #2						
Bit	Identifier		Reset		Attr.		Function / Description		
7dt0	XHIF_1_P2_RG_ROSL2		00h		r		Read only value = 0		

21dt8	XHIF_1_P2_RG_RW	0000h	r	w	Read/Write part of the Range register
31dt22	XHIF_1_P2_RG_ROSL1	000h	r		Read only value = 0

Register:	XHIF_1_P2_OF				Address:	E4h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:		Offset value of the page #2					
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_1_P2_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_1_P2_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:		XHIF_1_P2_CFG				Address:		E8h			
Bits:		0dt0		Reset value:		0h		Attributes:		r	w
Description:			Configuration of the buffering mode for page #2								
Bit	Identifier			Reset		Attr.		Function / Description			
0	XHIF_1_P2_CFG_BUFMOD			0h		r	w	16bit access mode of page #2			

Register:	XHIF_1_P3_RG					Address:	F0h		
Bits:	31dt0		Reset value:		0h		Attributes:	r	(w)
Description:			Range value of the page #3						
Bit	Identifier		Reset	Attr.		Function / Description			
7dt0	XHIF_1_P3_RG_ROSL2		00h	r		Read only value = 0			
21dt8	XHIF_1_P3_RG_RW		0000h	r	w	Read/Write part of the Range register			
31dt22	XHIF_1_P3_RG_ROSL1		000h	r		Read only value = 0			

Register:	XHIF_1_P3_OF				Address:	F4h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:		Offset value of the page #3					
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_1_P3_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_1_P3_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:	XHIF_1_P3_CFG				Address:	F8h	
Bits:	0dt0	Reset value:	0h		Attributes:	r	w
Description:	Configuration of the buffering mode for page #3						
Bit	Identifier	Reset	Attr.	Function / Description			
0	XHIF_1_P3_CFG_BUFMOD	0h	r w	16bit access mode of page #3			

Register:	XHIF_1_VERSION				Address:	FCh	
Bits:	31dt0	Reset value:	04000804h		Attributes:	r	
Description:	Metal-fix register for XHIF Version						

2.3.7 Peripheral Interface

The peripheral interface is operated as passive (i.e. as slave) at the multi-layer AHB and, in one version, acts as the communication interface between an external application (host CPU) and the PN stack running internally (ARM-CPU) or the Profinet IP (PN-IP). In another version (application running on an external host), the peripheral interface is used merely as the interface for consistent data management of the cyclic IO data for the PN-IP.. The cyclic IO data are also managed consistently by the Consistence Control module in the peripheral interface. ARM and host interface access to Consistence Control are multiplexed (arbitrated), as only the internal ARM926 or an external host system requires access.. As direct access to the IO data is possible from the PN-IP, there are no relevant lockout times for data transport. The IO data required can therefore be sent straight from or received directly by the peripheral interface without buffering at the local API (cut-through).

Consistence-Control ensures the consistency of the IO data saved in the IO-RAM. Consistent storage between the PN-IP (communicative instance) and the ARM, host interface and data IO interfaces (applicative instances) is realized in the process.

The APB slave IF of the parameter module is used for application-specific configuration of the peripheral interface by ARM926. Access to the entire memory area is possible over the APB for initializing the IO RAM. This requires both ready control and byte-by-byte access with byte enable at the APB. Access to the local IO APB and therefore to the modules connected to it is possible from the APB slave IF over the data IO interface for initial configuration.

The mechanisms required for two-way communication between the ARM and host CPU are implemented in the event unit. The event unit stores event bits which can be set both by the ARM for the external host system and vice versa. These events cause the module to trigger an interrupt for the ARM926 interrupt controller (IRQ54: ARM-IRQ, see 2.3.2.14) or the external host system (Host_IRQ = ASIC pin: XHIF_XIRQ). The registers can be

written and read with common AHB access. The event unit is configured with the parameter module and therefore the APB.

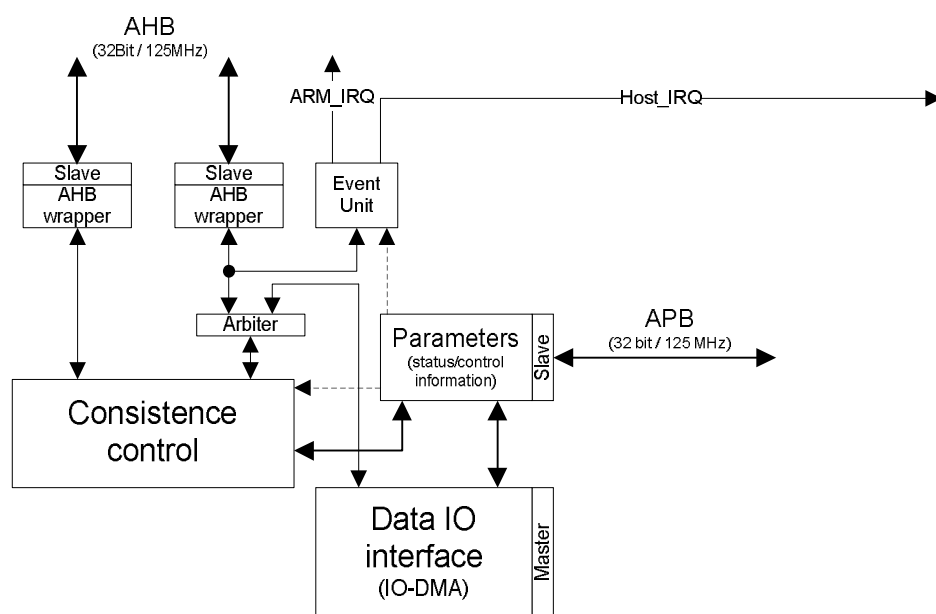


Figure 22: Block diagram of the peripheral interface

2.3.7.1 APB Interface

2.3.7.1.1 Address Mapping

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	5FFFh	perif_apb	
2000h	5FFFh	PERIF_IO_RAM	

Module	Register/Memory	Read	Write	Address
/perif_apb				
	CR_Address_1	r(h)	(w)	0h
	CR_Address_2	r(h)	(w)	4h
	CR_Address_3	r(h)	(w)	8h
	CR_Address_4	r(h)	(w)	Ch
	CR_Address_5	r(h)	(w)	10h
	CR_Address_6	r(h)	(w)	14h
	CR_Address_7	r(h)	(w)	18h
	CR_Address_8	r(h)	(w)	1Ch
	CR_Address_9	r(h)	(w)	20h
	CR_Address_10	r(h)	(w)	24h
	CR_Address_11	r(h)	(w)	28h
	CR_Address_12	r(h)	(w)	2Ch
	Reserved	r(h)	(w)	30h
	Reserved	r(h)	(w)	34h
	Reserved	r(h)	(w)	38h
	Reserved	r(h)	(w)	3Ch
	Reserved	r(h)	(w)	40h
	Reserved	r(h)	(w)	44h
	Reserved	r(h)	(w)	48h
	Reserved	r(h)	(w)	4Ch
	Reserved	r(h)	(w)	50h
	Reserved	r(h)	(w)	54h
	Reserved	r(h)	(w)	58h
	Reserved	r(h)	(w)	5Ch
	Reserved	r(h)	(w)	60h

	Reserved	r(h)	(w)	64h
	Reserved	r(h)	(w)	68h
	CR_State_1	r(h)	(w)	100h
	CR_State_2	r(h)	(w)	104h
	CR_State_3	r(h)	(w)	108h
	CR_State_4	r(h)	(w)	10Ch
	CR_State_5	r(h)	(w)	110h
	CR_State_6	r(h)	(w)	114h
	CR_State_7	r(h)	(w)	118h
	CR_State_8	r(h)	(w)	11Ch
	CR_State_9	r(h)	(w)	120h
	CR_State_10	r(h)	(w)	124h
	CR_State_11	r(h)	(w)	128h
	CR_State_12	r(h)	(w)	12Ch
	Reserved	r(h)	(w)	130h
	Reserved	r(h)	(w)	134h
	Reserved	r(h)	(w)	138h
	Reserved	r(h)	(w)	13Ch
	Reserved	r(h)	(w)	140h
	Reserved	r(h)	(w)	144h
	Reserved	r(h)	(w)	148h
	Reserved	r(h)	(w)	14Ch
	Reserved	r(h)	(w)	150h
	Reserved	r(h)	(w)	154h
	Reserved	r(h)	(w)	158h
	Reserved	r(h)	(w)	15Ch
	Reserved	r(h)	(w)	160h
	Reserved	r(h)	(w)	164h
	Reserved	r(h)	(w)	168h
	Guard_Control_1	r	(w)	200h
	Guard_Control_2	r	(w)	204h
	Guard_Control_3	r	(w)	208h
	Guard_Control_4	r	(w)	20Ch

	Guard_Control_5	r	(w)	210h
	Guard_Control_6	r	(w)	214h
	Guard_Control_7	r	(w)	218h
	Guard_Control_8	r	(w)	21Ch
	Guard_Control_9	r	(w)	220h
	Guard_Control_10	r	(w)	224h
	Guard_Control_11	r	(w)	228h
	Guard_Control_12	r	(w)	22Ch
	Reserved	r	(w)	230h
	Reserved	r	(w)	234h
	Reserved	r	(w)	238h
	Reserved	r	(w)	23Ch
	Reserved	r	(w)	240h
	Reserved	r	(w)	244h
	Reserved	r	(w)	248h
	Reserved	r	(w)	24Ch
	Reserved	r	(w)	250h
	Reserved	r	(w)	254h
	Reserved	r	(w)	258h
	Reserved	r	(w)	25Ch
	Reserved	r	(w)	260h
	Reserved	r	(w)	264h
	Reserved	r	(w)	268h
	IO_Mode	r	(w)	300h
	OutData_SrcAddress	r	(w)	304h
	OutData_DstAddress	r	(w)	308h
	OutData_Offset	r	(w)	30Ch
	Sub_Value	r(h)	(w)	310h
	Default_Value	r	(w)	314h
	Last_Value	r	(w)	318h
	RecordValue_O	r	(w)	31Ch
	RecordValue_I	r	(w)	320h
	InData_SrcAddress	r	(w)	324h

	InData_DstAddress	r	(w)	328h
	InData_Offset	r	(w)	32Ch
	SPI_OutHeaderProtocol_0	r	w	380h
	SPI_OutHeaderProtocol_1	r	w	384h
	SPI_OutHeaderLength	r	(w)	388h
	SPI_OutMode	r	(w)	38Ch
	SPI_InHeaderProtocol_0	r	w	390h
	SPI_InHeaderProtocol_1	r	w	394h
	SPI_InHeaderLength	r	(w)	398h
	SPI_InMode	r	(w)	39Ch
	SPI_Handshake	r	(w)	3A0h
	GPIO_IOCTL_0	r	w	400h
	GPIO_OUT_0	rh	w	404h
	GPIO_OUT_SET_0	rh	w	408h
	GPIO_OUT_CLEAR_0	rh	w	40Ch
	GPIO_RES_DIS_0	r	w	410h
	GPIO_IN_0	rh		414h
	GPIO_PORT_MODE_0_L	r	w	418h
	GPIO_PORT_MODE_0_H	r	w	41Ch
	GPIO_IOCTL_1	r	w	420h
	GPIO_OUT_1	rh	w	424h
	GPIO_OUT_SET_1	rh	w	428h
	GPIO_OUT_CLEAR_1	rh	w	42Ch
	GPIO_RES_DIS_1	r	w	430h
	GPIO_IN_1	rh		434h
	GPIO_PORT_MODE_1_L	r	w	438h
	GPIO_PORT_MODE_1_H	r	w	43Ch
	SSPCR0	r	w	500h
	SSPCR1	r	w	504h
	SSPDR	rh	w	508h
	SSPSR	rh		50Ch
	SSPCPSR	r(h)	(w)	510h
	SSPIIR_SSPICR	r(h)		514h

	SSPCR0_1	r	w	520h
	SSPCR1_1	r	w	524h
	SSPDR_1	rh	w	528h
	SSPSR_1	rh		52Ch
	SSPCPSR_1	r(h)	(w)	530h
	SSPIIR_SSPICR_1	rh		534h
	Address_Mode	r	w	600h
	IP_Version	r		800h
	IP_Development	r		804h
	Access_Err	r(h)	(w)	808h
	Burst_Config	r	(w)	80Ch
	PERIF_IO_RAM	r	w	2000h - 4FFFh

2.3.7.1.2 Register Description

Module: perif_apb

Register:	CR_Address_1			Address:	0h	
Bits:	31dt0	Reset value:	FFCh	Attributes:	(r)	(w)
Description:	CR_Address					
Bit	Identifier	Reset	Attr.		Function / Description	
1dt0	reserved	0h				
11dt2	CR_StartAddress	3FFh	r	w	CR_StartAddress	
15dt12	reserved	0h				
25dt16	CR_EndAddress	0h	r	w	CR_EndAddress	
26dt26	reserved	0h				
27dt27	Zero_Data	0h	r	w	Read returns zero data	
31dt28	New_Data_INT	0h	r	w	New_Data_INT	

Register:	CR_Address_2			Address:	4h	
Bits:	31dt0	Reset value:	FFCh	Attributes:	(r)	(w)
Description:	CR_Address					
Bit	Identifier	Reset	Attr.		Function / Description	
1dt0	reserved	0h				

11dt2	CR_StartAddress	3FFh	r	w	CR_StartAddress
15dt12	reserved	0h			
25dt16	CR_EndAddress	0h	r	w	CR_EndAddress
26dt26	reserved	0h			
27dt27	Zero_Data	0h	r	w	Read returns zero data
31dt28	New_Data_INT	0h	r	w	New_Data_INT

Register:	CR_Address_3				Address:	8h	
Bits:	31dt0	Reset value:	FFCh		Attributes:	(r)	(w)
Description:		CR_Address					
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	reserved	0h					
11dt2	CR_StartAddress	3FFh	r	w	CR_StartAddress		
15dt12	reserved	0h					
25dt16	CR_EndAddress	0h	r	w	CR_EndAddress		
26dt26	reserved	0h					
27dt27	Zero_Data	0h	r	w	Read returns zero data		
31dt28	New_Data_INT	0h	r	w	New_Data_INT		

Register:	CR_Address_4				Address:	Ch	
Bits:	31dt0	Reset value:	FFCh		Attributes:	(r)	(w)
Description:		CR_Address					
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	reserved	0h					
11dt2	CR_StartAddress	3FFh	r	w	CR_StartAddress		
15dt12	reserved	0h					
25dt16	CR_EndAddress	0h	r	w	CR_EndAddress		
26dt26	reserved	0h					
27dt27	Zero_Data	0h	r	w	Read returns zero data		
31dt28	New_Data_INT	0h	r	w	New_Data_INT		

Register:	CR_Address_5				Address:	10h	
Bits:	31dt0	Reset value:	FFCh		Attributes:	(r)	(w)

Description:		CR_Address			
Bit	Identifier	Reset	Attr.		Function / Description
1dt0	reserved	0h			
11dt2	CR_StartAddress	3FFh	r	w	CR_StartAddress
15dt12	reserved	0h			
25dt16	CR_EndAddress	0h	r	w	CR_EndAddress
26dt26	reserved	0h			
27dt27	Zero_Data	0h	r	w	Read returns zero data
31dt28	New_Data_INT	0h	r	w	New_Data_INT

Register:		CR_Address_6				Address:		14h			
Bits:		31dt0		Reset value:		FFCh		Attributes:		(r)	(w)
Description:		CR_Address									
Bit	Identifier			Reset		Attr.		Function / Description			
1dt0	reserved			0h							
11dt2	CR_StartAddress			3FFh		r	w	CR_StartAddress			
15dt12	reserved			0h							
25dt16	CR_EndAddress			0h		r	w	CR_EndAddress			
26dt26	reserved			0h							
27dt27	Zero_Data			0h		r	w	Read returns zero data			
31dt28	New_Data_INT			0h		r	w	New_Data_INT			

Register:		CR_Address_7				Address:		18h			
Bits:		31dt0		Reset value:		FFCh		Attributes:		(r)	(w)
Description:		CR_Address									
Bit	Identifier			Reset		Attr.		Function / Description			
1dt0	reserved			0h							
11dt2	CR_StartAddress			3FFh		r	w	CR_StartAddress			
15dt12	reserved			0h							
25dt16	CR_EndAddress			0h		r	w	CR_EndAddress			
26dt26	reserved			0h							
27dt27	Zero_Data			0h		r	w	Read returns zero data			
31dt28	New_Data_INT			0h		r	w	New_Data_INT			

Register:	CR_Address_8			Address:	1Ch	
Bits:	31dt0	Reset value:	FFCh	Attributes:	(r)	(w)
Description:	CR_Address					
Bit	Identifier	Reset	Attr.	Function / Description		
1dt0	reserved	0h				
11dt2	CR_StartAddress	3FFh	r w	CR_StartAddress		
15dt12	reserved	0h				
25dt16	CR_EndAddress	0h	r w	CR_EndAddress		
26dt26	reserved	0h				
27dt27	Zero_Data	0h	r w	Read returns zero data		
31dt28	New_Data_INT	0h	r w	New_Data_INT		

Register:	CR_Address_9			Address:	20h	
Bits:	31dt0	Reset value:	FFCh	Attributes:	(r)	(w)
Description:	CR_Address					
Bit	Identifier	Reset	Attr.	Function / Description		
1dt0	reserved	0h				
11dt2	CR_StartAddress	3FFh	r w	CR_StartAddress		
15dt12	reserved	0h				
25dt16	CR_EndAddress	0h	r w	CR_EndAddress		
26dt26	reserved	0h				
27dt27	Zero_Data	0h	r w	Read returns zero data		
31dt28	New_Data_INT	0h	r w	New_Data_INT		

Register:	CR_Address_10			Address:	24h	
Bits:	31dt0	Reset value:	FFCh	Attributes:	(r)	(w)
Description:	CR_Address					
Bit	Identifier	Reset	Attr.	Function / Description		
1dt0	reserved	0h				
11dt2	CR_StartAddress	3FFh	r w	CR_StartAddress		
15dt12	reserved	0h				
25dt16	CR_EndAddress	0h	r w	CR_EndAddress		

26dt26	reserved	0h			
27dt27	Zero_Data	0h	r	w	Read returns zero data
31dt28	New_Data_INT	0h	r	w	New_Data_INT

Register:	CR_Address_11				Address:	28h	
Bits:	31dt0	Reset value:	FFCh		Attributes:	(r)	(w)
Description:		CR_Address					
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	reserved	0h					
11dt2	CR_StartAddress	3FFh	r	w	CR_StartAddress		
15dt12	reserved	0h					
25dt16	CR_EndAddress	0h	r	w	CR_EndAddress		
26dt26	reserved	0h					
27dt27	Zero_Data	0h	r	w	Read returns zero data		
31dt28	New_Data_INT	0h	r	w	New_Data_INT		

Register:	CR_Address_12				Address:	2Ch	
Bits:	31dt0	Reset value:	FFCh		Attributes:	(r)	(w)
Description:		CR_Address					
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	reserved	0h					
11dt2	CR_StartAddress	3FFh	r	w	CR_StartAddress		
15dt12	reserved	0h					
25dt16	CR_EndAddress	0h	r	w	CR_EndAddress		
26dt26	reserved	0h					
27dt27	Zero_Data	0h	r	w	Read returns zero data		
31dt28	New_Data_INT	0h	r	w	New_Data_INT		

Register:		CR_State_1				Address:		100h	
Bits:		31dt0		Reset value:		FD2h		Attributes: r(h) w	
Description:		CR_State							
Bit	Identifier			Reset	Attr.	Function / Description			

1dt0	buffer_number_data_buffer	2h	rh	w	buffer_number_data_buffer
3dt2	buffer_number_r2_buffer	0h	rh	w	buffer_number_r2_buffer
5dt4	buffer_number_free_buffer	1h	rh	w	buffer_number_free_buffer
7dt6	buffer_number_f2_buffer	3h	rh	w	buffer_number_f2_buffer
9dt8	buffer_number_next_buffer	3h	rh	w	buffer_number_next_buffer
11dt10	buffer_number_n2_buffer	3h	rh	w	buffer_number_n2_buffer
13dt12	buffer_number_user_buffer	0h	rh	w	buffer_number_user_buffer
14dt14	Redundance_Mode	0h	r	w	Redundance_Mode
15dt15	EXT_BUF_Mode	0h	r	w	EXT_BUF_Mode
31dt16	Mapping	0h	r	w	Mapping

Register:	CR_State_2				Address:	104h	
Bits:	31dt0	Reset value:	FD2h		Attributes:	r(h)	w
Description:		CR_State					
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	buffer_number_data_buffer		
3dt2	buffer_number_r2_buffer	0h	rh	w	buffer_number_r2_buffer		
5dt4	buffer_number_free_buffer	1h	rh	w	buffer_number_free_buffer		
7dt6	buffer_number_f2_buffer	3h	rh	w	buffer_number_f2_buffer		
9dt8	buffer_number_next_buffer	3h	rh	w	buffer_number_next_buffer		
11dt10	buffer_number_n2_buffer	3h	rh	w	buffer_number_n2_buffer		
13dt12	buffer_number_user_buffer	0h	rh	w	buffer_number_user_buffer		
14dt14	Redundance_Mode	0h	r	w	Redundance_Mode		
15dt15	EXT_BUF_Mode	0h	r	w	EXT_BUF_Mode		
31dt16	Mapping	0h	r	w	Mapping		

Register:	CR_State_3					Address:	108h	
Bits:	31dt0		Reset value:		FD2h	Attributes:	r(h)	w
Description:		CR_State						
Bit	Identifier		Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer		2h	rh	w	buffer_number_data_buffer		
3dt2	buffer_number_r2_buffer		0h	rh	w	buffer_number_r2_buffer		
5dt4	buffer number free buffer		1h	rh	w	buffer number free buffer		

7dt6	buffer_number_f2_buffer	3h	rh	w	buffer_number_f2_buffer
9dt8	buffer_number_next_buffer	3h	rh	w	buffer_number_next_buffer
11dt10	buffer_number_n2_buffer	3h	rh	w	buffer_number_n2_buffer
13dt12	buffer_number_user_buffer	0h	rh	w	buffer_number_user_buffer
14dt14	Redundance_Mode	0h	r	w	Redundance_Mode
15dt15	EXT_BUF_Mode	0h	r	w	EXT_BUF_Mode
31dt16	Mapping	0h	r	w	Mapping

Register:	CR_State_4				Address:	10Ch	
Bits:	31dt0	Reset value:	FD2h		Attributes:	r(h)	w
Description:		CR_State					
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	buffer_number_data_buffer		
3dt2	buffer_number_r2_buffer	0h	rh	w	buffer_number_r2_buffer		
5dt4	buffer_number_free_buffer	1h	rh	w	buffer_number_free_buffer		
7dt6	buffer_number_f2_buffer	3h	rh	w	buffer_number_f2_buffer		
9dt8	buffer_number_next_buffer	3h	rh	w	buffer_number_next_buffer		
11dt10	buffer_number_n2_buffer	3h	rh	w	buffer_number_n2_buffer		
13dt12	buffer_number_user_buffer	0h	rh	w	buffer_number_user_buffer		
14dt14	Redundance_Mode	0h	r	w	Redundance_Mode		
15dt15	EXT_BUF_Mode	0h	r	w	EXT_BUF_Mode		
31dt16	Mapping	0h	r	w	Mapping		

Register:	CR_State_5				Address:	110h	
Bits:	31dt0	Reset value:	FD2h		Attributes:	r(h)	w
Description:		CR_State					
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	buffer_number_data_buffer		
3dt2	buffer_number_r2_buffer	0h	rh	w	buffer_number_r2_buffer		
5dt4	buffer_number_free_buffer	1h	rh	w	buffer_number_free_buffer		
7dt6	buffer_number_f2_buffer	3h	rh	w	buffer_number_f2_buffer		
9dt8	buffer_number_next_buffer	3h	rh	w	buffer_number_next_buffer		
11dt10	buffer number n2 buffer	3h	rh	w	buffer number n2 buffer		

13dt12	buffer_number_user_buffer	0h	rh	w	buffer_number_user_buffer
14dt14	Redundance_Mode	0h	r	w	Redundance_Mode
15dt15	EXT_BUF_Mode	0h	r	w	EXT_BUF_Mode
31dt16	Mapping	0h	r	w	Mapping

Register:	CR_State_6				Address:	114h	
Bits:	31dt0	Reset value:	FD2h		Attributes:	r(h)	w
Description:		CR_State					
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	buffer_number_data_buffer		
3dt2	buffer_number_r2_buffer	0h	rh	w	buffer_number_r2_buffer		
5dt4	buffer_number_free_buffer	1h	rh	w	buffer_number_free_buffer		
7dt6	buffer_number_f2_buffer	3h	rh	w	buffer_number_f2_buffer		
9dt8	buffer_number_next_buffer	3h	rh	w	buffer_number_next_buffer		
11dt10	buffer_number_n2_buffer	3h	rh	w	buffer_number_n2_buffer		
13dt12	buffer_number_user_buffer	0h	rh	w	buffer_number_user_buffer		
14dt14	Redundance_Mode	0h	r	w	Redundance_Mode		
15dt15	EXT_BUF_Mode	0h	r	w	EXT_BUF_Mode		
31dt16	Mapping	0h	r	w	Mapping		

Register:	CR_State_7				Address:	118h	
Bits:	31dt0	Reset value:	FD2h		Attributes:	r(h)	w
Description:		CR_State					
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	buffer_number_data_buffer		
3dt2	buffer_number_r2_buffer	0h	rh	w	buffer_number_r2_buffer		
5dt4	buffer_number_free_buffer	1h	rh	w	buffer_number_free_buffer		
7dt6	buffer_number_f2_buffer	3h	rh	w	buffer_number_f2_buffer		
9dt8	buffer_number_next_buffer	3h	rh	w	buffer_number_next_buffer		
11dt10	buffer_number_n2_buffer	3h	rh	w	buffer_number_n2_buffer		
13dt12	buffer_number_user_buffer	0h	rh	w	buffer_number_user_buffer		
14dt14	Redundance_Mode	0h	r	w	Redundance_Mode		
15dt15	EXT BUF Mode	0h	r	w	EXT BUF Mode		

31dt16	Mapping	0h	r	w	Mapping
--------	---------	----	---	---	---------

Register:	CR_State_8				Address:	11Ch	
Bits:	31dt0	Reset value:	FD2h		Attributes:	r(h)	w
Description:		CR_State					
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	buffer_number_data_buffer		
3dt2	buffer_number_r2_buffer	0h	rh	w	buffer_number_r2_buffer		
5dt4	buffer_number_free_buffer	1h	rh	w	buffer_number_free_buffer		
7dt6	buffer_number_f2_buffer	3h	rh	w	buffer_number_f2_buffer		
9dt8	buffer_number_next_buffer	3h	rh	w	buffer_number_next_buffer		
11dt10	buffer_number_n2_buffer	3h	rh	w	buffer_number_n2_buffer		
13dt12	buffer_number_user_buffer	0h	rh	w	buffer_number_user_buffer		
14dt14	Redundance_Mode	0h	r	w	Redundance_Mode		
15dt15	EXT_BUF_Mode	0h	r	w	EXT_BUF_Mode		
31dt16	Mapping	0h	r	w	Mapping		

Register:		CR_State_9				Address:		120h	
Bits:		31dt0		Reset value:		FD2h		Attributes: r(h) w	
Description:		CR_State							
Bit	Identifier			Reset		Attr.		Function / Description	
1dt0	buffer_number_data_buffer			2h		rh	w	buffer_number_data_buffer	
3dt2	buffer_number_r2_buffer			0h		rh	w	buffer_number_r2_buffer	
5dt4	buffer_number_free_buffer			1h		rh	w	buffer_number_free_buffer	
7dt6	buffer_number_f2_buffer			3h		rh	w	buffer_number_f2_buffer	
9dt8	buffer_number_next_buffer			3h		rh	w	buffer_number_next_buffer	
11dt10	buffer_number_n2_buffer			3h		rh	w	buffer_number_n2_buffer	
13dt12	buffer_number_user_buffer			0h		rh	w	buffer_number_user_buffer	
14dt14	Redundance_Mode			0h		r	w	Redundance_Mode	
15dt15	EXT_BUF_Mode			0h		r	w	EXT_BUF_Mode	
31dt16	Mapping			0h		r	w	Mapping	

Register:	CR_State_10				Address:	124h	
-----------	-------------	--	--	--	----------	------	--

Bits:	31dt0	Reset value:	FD2h	Attributes:	r(h)	w
Description:	CR_State					
Bit	Identifier	Reset	Attr.		Function / Description	
1dt0	buffer_number_data_buffer	2h	rh	w	buffer_number_data_buffer	
3dt2	buffer_number_r2_buffer	0h	rh	w	buffer_number_r2_buffer	
5dt4	buffer_number_free_buffer	1h	rh	w	buffer_number_free_buffer	
7dt6	buffer_number_f2_buffer	3h	rh	w	buffer_number_f2_buffer	
9dt8	buffer_number_next_buffer	3h	rh	w	buffer_number_next_buffer	
11dt10	buffer_number_n2_buffer	3h	rh	w	buffer_number_n2_buffer	
13dt12	buffer_number_user_buffer	0h	rh	w	buffer_number_user_buffer	
14dt14	Redundance_Mode	0h	r	w	Redundance_Mode	
15dt15	EXT_BUF_Mode	0h	r	w	EXT_BUF_Mode	
31dt16	Mapping	0h	r	w	Mapping	

Register:	CR_State_11				Address:	128h
Bits:	31dt0	Reset value:	FD2h	Attributes:	r(h)	w
Description:	CR_State					
Bit	Identifier	Reset	Attr.		Function / Description	
1dt0	buffer_number_data_buffer	2h	rh	w	buffer_number_data_buffer	
3dt2	buffer_number_r2_buffer	0h	rh	w	buffer_number_r2_buffer	
5dt4	buffer_number_free_buffer	1h	rh	w	buffer_number_free_buffer	
7dt6	buffer_number_f2_buffer	3h	rh	w	buffer_number_f2_buffer	
9dt8	buffer_number_next_buffer	3h	rh	w	buffer_number_next_buffer	
11dt10	buffer_number_n2_buffer	3h	rh	w	buffer_number_n2_buffer	
13dt12	buffer_number_user_buffer	0h	rh	w	buffer_number_user_buffer	
14dt14	Redundance_Mode	0h	r	w	Redundance_Mode	
15dt15	EXT_BUF_Mode	0h	r	w	EXT_BUF_Mode	
31dt16	Mapping	0h	r	w	Mapping	

Register:	CR_State_12				Address:	12Ch
Bits:	31dt0	Reset value:	FD2h	Attributes:	r(h)	w
Description:	CR_State					
Bit	Identifier	Reset	Attr.		Function / Description	

1dt0	buffer_number_data_buffer	2h	rh	w	buffer_number_data_buffer
3dt2	buffer_number_r2_buffer	0h	rh	w	buffer_number_r2_buffer
5dt4	buffer_number_free_buffer	1h	rh	w	buffer_number_free_buffer
7dt6	buffer_number_f2_buffer	3h	rh	w	buffer_number_f2_buffer
9dt8	buffer_number_next_buffer	3h	rh	w	buffer_number_next_buffer
11dt10	buffer_number_n2_buffer	3h	rh	w	buffer_number_n2_buffer
13dt12	buffer_number_user_buffer	0h	rh	w	buffer_number_user_buffer
14dt14	Redundance_Mode	0h	r	w	Redundance_Mode
15dt15	EXT_BUF_Mode	0h	r	w	EXT_BUF_Mode
31dt16	Mapping	0h	r	w	Mapping

Register:		Guard_Control_1				Address:		200h			
Bits:		31dt0		Reset value:		0h		Attributes:		(r)	(w)
Description:		Guard_Control_1									
Bit	Identifier			Reset		Attr.		Function / Description			
11dt0	guard_address			0h		r	w				
29dt12	reserved			0h							
30dt30	Guard_Type			0h		r	w	Guard_Type			
31dt31	Guard_Valid			0h		r	w	Guard_Valid			

Register:		Guard_Control_2				Address:		204h	
Bits:		31dt0	Reset value:		0h		Attributes:	(r)	(w)
Description:		Guard_Control_1							
Bit	Identifier		Reset	Attr.		Function / Description			
11dt0	guard_address		0h	r	w				
29dt12	reserved		0h						
30dt30	Guard_Type		0h	r	w	Guard_Type			
31dt31	Guard_Valid		0h	r	w	Guard_Valid			

Register:	Guard_Control_3				Address:	208h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)	
Description:	Guard_Control_1						

Bit	Identifier	Reset	Attr.		Function / Description
11dt0	guard_address	0h	r	w	
29dt12	reserved	0h			
30dt30	Guard_Type	0h	r	w	Guard_Type
31dt31	Guard_Valid	0h	r	w	Guard_Valid

Register:	Guard_Control_4				Address:	20Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	(r)	(w)
Description:		Guard_Control_1					
Bit	Identifier	Reset	Attr.		Function / Description		
11dt0	guard_address	0h	r	w			
29dt12	reserved	0h					
30dt30	Guard_Type	0h	r	w	Guard_Type		
31dt31	Guard_Valid	0h	r	w	Guard_Valid		

Register:	Guard_Control_5				Address:	210h	
Bits:	31dt0	Reset value:	0h		Attributes:	(r)	(w)
Description:		Guard_Control_1					
Bit	Identifier	Reset	Attr.		Function / Description		
11dt0	guard_address	0h	r	w			
29dt12	reserved	0h					
30dt30	Guard_Type	0h	r	w	Guard_Type		
31dt31	Guard_Valid	0h	r	w	Guard_Valid		

Register:		Guard_Control_6				Address:		214h			
Bits:		31dt0		Reset value:		0h		Attributes:		(r)	(w)
Description:		Guard_Control_1									
Bit	Identifier			Reset		Attr.		Function / Description			
11dt0	guard_address			0h		r	w				
29dt12	reserved			0h							
30dt30	Guard_Type			0h		r	w	Guard_Type			
31dt31	Guard_Valid			0h		r	w	Guard_Valid			

Register:		Guard_Control_7				Address:		218h			
Bits:		31dt0		Reset value:		0h		Attributes:		(r)	(w)
Description:		Guard_Control_1									
Bit	Identifier			Reset		Attr.		Function / Description			
11dt0	guard_address			0h		r	w				
29dt12	reserved			0h							
30dt30	Guard_Type			0h		r	w	Guard_Type			
31dt31	Guard_Valid			0h		r	w	Guard_Valid			

Register:		Guard_Control_8				Address:		21Ch			
Bits:		31dt0		Reset value:		0h		Attributes:		(r)	(w)
Description:		Guard_Control_1									
Bit	Identifier			Reset		Attr.		Function / Description			
11dt0	guard_address			0h		r	w				
29dt12	reserved			0h							
30dt30	Guard_Type			0h		r	w	Guard_Type			
31dt31	Guard_Valid			0h		r	w	Guard_Valid			

Register:		Guard_Control_9				Address:		220h			
Bits:		31dt0		Reset value:		0h		Attributes:		(r)	(w)
Description:		Guard_Control_1									
Bit	Identifier			Reset		Attr.		Function / Description			
11dt0	guard_address			0h		r	w				
29dt12	reserved			0h							
30dt30	Guard_Type			0h		r	w	Guard_Type			
31dt31	Guard Valid			0h		r	w	Guard Valid			

Register:	Guard_Control_10				Address:	224h	
Bits:	31dt0	Reset value:	0h		Attributes:	(r)	(w)
Description:	Guard_Control_1						
Bit	Identifier	Reset	Attr.		Function / Description		
11dt0	guard_address	0h	r	w			
29dt12	reserved	0h					

30dt30	Guard_Type	0h	r	w	Guard_Type
31dt31	Guard_Valid	0h	r	w	Guard_Valid

Register:	Guard_Control_11					Address:	228h	
Bits:	31dt0	Reset value:	0h		Attributes:	(r)	(w)	
Description:		Guard_Control_1						
Bit	Identifier	Reset	Attr.		Function / Description			
11dt0	guard_address	0h	r	w				
29dt12	reserved	0h						
30dt30	Guard_Type	0h	r	w	Guard_Type			
31dt31	Guard_Valid	0h	r	w	Guard_Valid			

Register:		Guard_Control_12				Address:		22Ch			
Bits:		31dt0		Reset value:		0h		Attributes:		(r)	(w)
Description:		Guard_Control_1									
Bit	Identifier			Reset		Attr.		Function / Description			
11dt0	guard_address			0h		r	w				
29dt12	reserved			0h							
30dt30	Guard_Type			0h		r	w	Guard_Type			
31dt31	Guard_Valid			0h		r	w	Guard_Valid			

Register:	IO_Mode				Address:	300h	
Bits:	31dt0	Reset value:	0h		Attributes:	(r)	(w)
Description:		IO_Mode					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	O_Interface	0h	r	w	O_Interface		
1dt1	I_Interface	0h	r	w	I_Interface		
2dt2	IO_Mapping	0h	r	w	IO_Mapping		
6dt3	reserved	0h					
7dt7	Access_Mode	0h	r	w	Access_Mode		
31dt8	reserved	0h					

Register:	OutData_SrcAddress				Address:	304h	
-----------	--------------------	--	--	--	----------	------	--

Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	OutData_SrcAddress					
Bit	Identifier	Reset	Attr.		Function / Description	
11dt0	Src_Address	0h	r	w	Src_Address	
15dt12	reserved	0h				
26dt16	Length	0h	r	w	Length	
31dt27	CR_Number	0h	r	w	CR_Number	

Register:	OutData_DstAddress				Address:	308h
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	OutData_DstAddress					
Bit	Identifier	Reset	Attr.		Function / Description	
11dt0	GPIO_OUTAddress	0h	r	w	GPIO_OUTAddress	
31dt12	reserved	0h				

Register:	OutData_Offset				Address:	30Ch
Bits:	31dt0	Reset value:	20h	Attributes:	(r)	(w)
Description:	OutData_Offset					
Bit	Identifier	Reset	Attr.		Function / Description	
11dt0	GPIO_OUTOffset	20h	r	w	GPIO_OUTOffset	
31dt12	reserved	0h				

Register:	Sub_Value				Address:	310h
Bits:	31dt0	Reset value:	0h	Attributes:	(r)(h)	(w)
Description:	Sub_Value					
Bit	Identifier	Reset	Attr.		Function / Description	
0dt0	Sub_Mode	0h	r	w	Sub_Mode	
7dt1	reserved	0h				
8dt8	SubActiv_direct	0h	r	w	SubActiv_direct	
9dt9	SubActiv_state	0h	rh		SubActiv_state	
31dt10	reserved	0h				

Register:	Default_Value				Address:	314h
-----------	---------------	--	--	--	----------	------

Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	Default_Value					
Bit	Identifier	Reset	Attr.	Function / Description		
11dt0	DefaultValue	0h	r w	DefaultValue		
26dt12	reserved	0h				
31dt27	DefaultValue_CR_Nr	0h	r w	DefaultValue_CR_Nr		

Register:	Last_Value			Address:	318h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	Last_Value					
Bit	Identifier	Reset	Attr.	Function / Description		
11dt0	LastValue	0h	r w	LastValue		
26dt12	reserved	0h				
31dt27	LastValue_CR_Nr	0h	r w	LastValue_CR_Nr		

Register:	RecordValue_O			Address:	31Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	RecordValue_O					
Bit	Identifier	Reset	Attr.	Function / Description		
11dt0	RecordValue_O	0h	r w	RecordValue_O		
26dt12	reserved	0h				
31dt27	RecordValue_CR_Nr_O	0h	r w	RecordValue_CR_Nr_O		

Register:	RecordValue_I			Address:	320h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	RecordValue_I					
Bit	Identifier	Reset	Attr.	Function / Description		
11dt0	RecordValue_I	0h	r w	RecordValue_I		
26dt12	reserved	0h				
31dt27	RecordValue_CR_Nr_I	0h	r w	RecordValue_CR_Nr_I		

Register:	InData_SrcAddress			Address:	324h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)

Description:		InData_SrcAddress			
Bit	Identifier	Reset	Attr.		Function / Description
11dt0	GPIO_InAddress	0h	r	w	GPIO_InAddress
31dt12	reserved	0h			

Register:		InData_DstAddress				Address:		328h			
Bits:		31dt0		Reset value:		0h		Attributes:		(r)	(w)
Description:			InData_DstAddress								
Bit	Identifier			Reset		Attr.		Function / Description			
11dt0	Dst_Address			0h		r	w	Dst_Address			
15dt12	reserved			0h							
26dt16	Length			0h		r	w	Length			
31dt27	CR_Number			0h		r	w	CR_Number			

Register:		InData_Offset				Address:		32Ch			
Bits:		31dt0		Reset value:		20h		Attributes:		(r)	(w)
Description:			InData_Offset								
Bit	Identifier			Reset		Attr.		Function / Description			
11dt0	GPIO_InOffset			20h		r	w	GPIO_InOffset			
31dt12	reserved			0h							

Register:		SPI_OutHeaderProtocol_0				Address:		380h			
Bits:		31dt0		Reset value:		0h		Attributes:		r	w
Description:		SPI_OutHeaderProtocol_0									
Bit	Identifier			Reset		Attr.		Function / Description			
31dt0	SPI_OutHeaderProtocol_0			00h		r w		SPI_OutHeaderProtocol_0			

Register:		SPI_OutHeaderProtocol_1				Address:		384h			
Bits:		31dt0		Reset value:		0h		Attributes:		r	w
Description:		SPI_OutHeaderProtocol_1									
Bit	Identifier			Reset		Attr.		Function / Description			
31dt0	SPI_OutHeaderProtocol_1			00h		r w		SPI_OutHeaderProtocol_1			

Register:	SPI_OutHeaderLength			Address:	388h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	SPI_OutHeaderLength					
Bit	Identifier	Reset	Attr.	Function / Description		
3dt0	SPI_Length	0h	r w	SPI_Length		
31dt4	reserved	0h				

Register:	SPI_OutMode			Address:	38Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	SPI_OutMode					
Bit	Identifier	Reset	Attr.	Function / Description		
0dt0	OutTransfer_Mode	0h	r w	OutTransfer_Mode		
1dt1	OutTransfer_DIR	0h	r w	OutTransfer_DIR		
7dt2	reserved	0h				
15dt8	OutStatus	0h	r w	OutStatus		
31dt16	reserved	0h				

Register:	SPI_InHeaderProtocol_0			Address:	390h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	SPI_InHeaderProtocol_0					
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	SPI_InHeaderProtocol_0	00h	r w	SPI_InHeaderProtocol_0		

Register:	SPI_InHeaderProtocol_1			Address:	394h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	SPI_InHeaderProtocol_1					
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	SPI_InHeaderProtocol_1	00h	r w	SPI_InHeaderProtocol_1		

Register:	SPI_InHeaderLength			Address:	398h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	SPI_InHeaderLength					
Bit	Identifier	Reset	Attr.	Function / Description		

3dt0	SPI_Length	0h	r	w	SPI_Length
31dt4	reserved	0h			

Register:	SPI_InMode					Address:	39Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w	
Description:		SPI_InMode						
Bit	Identifier	Reset	Attr.		Function / Description			
0dt0	InTransfer_Mode	0h	r	w	InTransfer_Mode			
1dt1	InTransfer_DIR	0h	r	w	InTransfer_DIR			

Register:	SPI_Handshake				Address:	3A0h	
Bits:	31dt0	Reset value:	0h		Attributes:	(r)	(w)
Description:		SPI_Handshake					
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	Handshake_Mode	0h	r	w	Handshake_Mode		
1dt1	Busy_POL	0h	r	w	Busy_POL		
7dt2	reserved	0h					
23dt8	Wait_Time	0h	r	w	Wait_Time		
31dt24	reserved	0h					

Register:	GPIO_IOCTL_0				Address:	400h	
Bits:	31dt0	Reset value:	FFFFFFFF Fh		Attributes:	r	w
Description:		GPIO_IOCTL_0					
Bit	Identifier		Reset	Attr.	Function / Description		
31dt0	GPIO_IOCTL_0		FFFFFFFFFh	r w	GPIO_IOCTL_0		

Register:	GPIO_OUT_0					Address:	404h		
Bits:	31dt0		Reset value:		0h		Attributes:	rh	w
Description:		GPIO_OUT_0							
Bit	Identifier		Reset		Attr.		Function / Description		
31dt0	GPIO_OUT_0		00h		rh		w	GPIO_OUT_0	

Register:	GPIO_OUT_SET_0				Address:	408h	
-----------	----------------	--	--	--	----------	------	--

Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	GPIO_OUT_SET_0					
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	GPIO_OUT_SET_0	00h	rh w	GPIO_OUT_SET_0		

Register:	GPIO_OUT_CLEAR_0			Address:	40Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	GPIO_OUT_CLEAR_0					
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	GPIO_OUT_CLEAR_0	00h	rh w	GPIO_OUT_CLEAR_0		

Register:	GPIO_RES_DIS_0			Address:	410h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	GPIO_RES_DIS_0					
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	GPIO_RES_DIS_0	00h	r w	GPIO_RES_DIS_0		

Register:	GPIO_IN_0			Address:	414h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:	GPIO_IN_0					
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	GPIO_IN_0	00h	rh	GPIO_IN_0		

Register:	GPIO_PORT_MODE_0_L			Address:	418h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	GPIO_PORT_MODE_0_L					
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	GPIO_PORT_MODE_0_L	00h	r w	GPIO_PORT_MODE_0_L		

Register:	GPIO_PORT_MODE_0_H			Address:	41Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	GPIO_PORT_MODE_0_H					
Bit	Identifier	Reset	Attr.	Function / Description		

31dt0	GPIO_PORT_MODE_0_H	00h	r	w	GPIO_PORT_MODE_0_H
-------	--------------------	-----	---	---	--------------------

Register:	GPIO_IOCTL_1				Address:	420h	
Bits:	31dt0	Reset value:	FFFFFFFh		Attributes:	r	w
Description:		GPIO_IOCTL_1					
Bit	Identifier	Reset	Attr.	Function / Description			
31dt0	GPIO_IOCTL_1	FFFFFFFFh	r	w	GPIO_IOCTL_1		

Register:		GPIO_OUT_1				Address:		424h			
Bits:		31dt0		Reset value:		0h		Attributes:		rh	w
Description:		GPIO_OUT_1									
Bit	Identifier			Reset		Attr.		Function / Description			
31dt0	GPIO_OUT_1			00h		rh w		GPIO_OUT_1			

Register:		GPIO_OUT_SET_1				Address:		428h			
Bits:		31dt0		Reset value:		0h		Attributes:		rh	w
Description:		GPIO_OUT_SET_1									
Bit	Identifier			Reset		Attr.		Function / Description			
31dt0	GPIO_OUT_SET_1			00h		rh w		GPIO_OUT_SET_1			

Register:	GPIO_OUT_CLEAR_1					Address:	42Ch	
Bits:	31dt0		Reset value:		0h	Attributes:	rh	w
Description:		GPIO_OUT_CLEAR_1						
Bit	Identifier		Reset		Attr.	Function / Description		
31dt0	GPIO_OUT_CLEAR_1		00h		rh	w	GPIO_OUT_CLEAR_1	

Register:	GPIO_RES_DIS_1				Address:	430h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:		GPIO_RES_DIS_1					
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	GPIO_RES_DIS_1	00h	r	w	GPIO_RES_DIS_1		

Register:	GPIO_IN_1				Address:	434h	
-----------	-----------	--	--	--	----------	------	--

Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:	GPIO_IN_1					
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	GPIO_IN_1	00h	rh	GPIO_IN_1		

Register:	GPIO_PORT_MODE_1_L			Address:	438h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	GPIO_PORT_MODE_1_L					
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	GPIO_PORT_MODE_1_L	00h	r w	GPIO_PORT_MODE_1_L		

Register:	GPIO_PORT_MODE_1_H			Address:	43Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	GPIO_PORT_MODE_1_H					
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	GPIO_PORT_MODE_1_H	00h	r w	GPIO_PORT_MODE_1_H		

Register:	SSPCR0			Address:	500h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	SSPCR0					
Bit	Identifier	Reset	Attr.	Function / Description		
3dt0	DSS	0h	r w	DSS=Data Size Select		
5dt4	FRF	0h	r w	FRF=Frame Format		
6dt6	SPO	0h	r w	SPO=SCLKOUT Polarity		
7dt7	SPH	0h	r w	SPH=SCLKOUT Phase		
15dt8	SCR	0h	r w	SCR=Serial Clock Rate		

Register:	SSPCR1			Address:	504h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	SSPCR1					
Bit	Identifier	Reset	Attr.	Function / Description		
0dt0	RIE	0h	r w	RIE=Receive FIFO Interrupt Enable		
1dt1	TIE	0h	r w	TIE=Transmit FIFO Interrupt		

					Enable
2dt2	RORIE	0h	r	w	RORIE=Receive FIFO Overrun Interrupt Enable
3dt3	LBM	0h	r	w	LBM=Loop Back Mode
4dt4	SSE	0h	r	w	SSE=Synchronous Serial Port Enable
5dt5	MS	0h	r	w	MS=master/slave mode select
6dt6	SOD	0h	r	w	SOD=slave mode output disable

Register:		SSPDR				Address:		508h			
Bits:		31dt0		Reset value:		0h		Attributes:		rh	w
Description:			SSPDR								
Bit	Identifier			Reset		Attr.		Function / Description			
15dt0	DATA			0h		rh	w	Data			

Register:	SSPSR					Address:	50Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	rh		
Description:		SSPSR						
Bit	Identifier	Reset	Attr.	Function / Description				
0dt0	TFE	0h	rh		TFE=Transmit FIFO empty			
1dt1	TNF	0h	rh		TNF=Transmit FIFO not full			
2dt2	RNE	0h	rh		RNE=Receive FIFO not empty			
3dt3	RFF	0h	rh		RFF=Receive FIFO Full			
4dt4	BSY	0h	rh		BSY=SSP Busy Flag			

Register:	SSPCPSR				Address:	510h	
Bits:	31dt0	Reset value:	0h		Attributes:	(r)(h)	(w)
Description:		SSPCPSR					
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	CPSDVSR	0h	rh	w	CPSDVSR=Clock Prescale divisor		
31dt8	reserved	0h					

Register:		SSPIIR_SSPICR				Address:		514h			
Bits:		31dt0		Reset value:		0h		Attributes:		(r)(h)	
Description:		SSPIIR_SSPICR									
Bit	Identifier			Reset		Attr.		Function / Description			
0dt0	RIS			0h		rh		RIS=SSP Transmit FIFO service request interrupt status			
1dt1	TIS			0h		rh		TIS=SSP Receive FIFO service request interrupt status			
2dt2	RORIS			0h		rh		RORIS=SSP Receive FIFO overrun interrupt status			
31dt3	reserved			0h							

Register:		SSPCR0_1				Address:		520h	
Bits:		31dt0	Reset value:		0h		Attributes:	r	w
Description:		SSPCR0							
Bit	Identifier		Reset	Attr.		Function / Description			
3dt0	DSS		0h	r	w	DSS=Data Size Select			
5dt4	FRF		0h	r	w	FRF=Frame Format			
6dt6	SPO		0h	r	w	SPO=SCLKOUT Polarity			
7dt7	SPH		0h	r	w	SPH=SCLKOUT Phase			
15dt8	SCR		0h	r	w	SCR=Serial Clock Rate			

Register:	SSPCR1_1				Address:	524h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:		SSPCR1					
Bit	Identifier		Reset	Attr.		Function / Description	
0dt0	RIE		0h	r	w	RIE=Receive FIFO Interrupt Enable	
1dt1	TIE		0h	r	w	TIE=Transmit FIFO Interrupt Enable	
2dt2	RORIE		0h	r	w	RORIE=Receive FIFO Overrun Interrupt Enable	
3dt3	LBM		0h	r	w	LBM=Loop Back Mode	
4dt4	SSE		0h	r	w	SSE=Synchronous Serial Port Enable	

5dt5	MS	0h	r	w	MS=master/slave mode select
6dt6	SOD	0h	r	w	SOD=slave mode output disable

Register:		SSPDR_1				Address:		528h			
Bits:		31dt0		Reset value:		0h		Attributes:		rh	w
Description:		SSPDR									
Bit	Identifier			Reset		Attr.		Function / Description			
15dt0	DATA			0h		rh w		Data			

Register:	SSPSR_1				Address:	52Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	
Description:		SSPSR					
Bit	Identifier		Reset	Attr.		Function / Description	
0dt0	TFE		0h	rh		TFE=Transmit FIFO empty	
1dt1	TNF		0h	rh		TNF=Transmit FIFO not full	
2dt2	RNE		0h	rh		RNE=Receive FIFO not empty	
3dt3	RFF		0h	rh		RFF=Receive FIFO Full	
4dt4	BSY		0h	rh		BSY=SSP Busy Flag	

Register:	SSPCPSR_1				Address:	530h	
Bits:	31dt0	Reset value:	0h		Attributes:	(r)(h)	(w)
Description:		SSPCPSR					
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	CPSDVSR	0h	rh	w	CPSDVSR=Clock Prescale divisor		
31dt8	reserved	0h					

Register:	SSPIIR_SSPICR_1					Address:	534h	
Bits:	31dt0		Reset value:		0h		Attributes:	(r)(h)
Description:		SSPIIR_SSPICR						
Bit	Identifier		Reset		Attr.		Function / Description	
0dt0	RIS		0h		rh		RIS=SSP Transmit FIFO ser-	

					vice request interrupt status
1dt1	TIS	0h	rh		TIS=SSP Receive FIFO service request interrupt status
2dt2	RORIS	0h	rh		RORIS=SSP Receive FIFO overrun interrupt status
31dt3	reserved	0h			

Register:	Address_Mode				Address:	600h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w	
Description:	Address_Mode						
Bit	Identifier	Reset	Attr.	Function / Description			
0dt0	Address_Mode	0h	r w	Addres_Mode			

Register:	IP_Version				Address:	800h	
Bits:	31dt0	Reset value:	30200h	Attributes:	r		
Description:	IP_Version						
Bit	Identifier	Reset	Attr.	Function / Description			
7dt0	debug_version	0h	r	debug_version			
15dt8	version	02h	r	version			
31dt16	configuration	0003h	r	configuration			

Register:	IP_Development				Address:	804h	
Bits:	31dt0	Reset value:	460100Ah	Attributes:	r		
Description:	IP_Development						
Bit	Identifier	Reset	Attr.	Function / Description			
10dt0	r_label	00Ah	r	CC RR label			
15dt11	Inkr	02h	r	Inkr			
18dt16	label_structure	0h	r	patch			
20dt19	type	0h	r	platform			
31dt21	identification	023h	r	identification			

Register:	Access_Error				Address:	808h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)(h)	(w)	

Description:		Access_Err			
Bit	Identifier	Reset	Attr.		Function / Description
16dt0	ahb_apb_address	0h	rh		ahb_apb_address
26dt17	reserved	0h			
27dt27	ahb_apb_select	0h	rh		ahb_apb_select
29dt28	ahb_apb_size	0h	rh		ahb_apb_size
30dt30	ahb_apb_rd_wr	0h	rh		ahb_apb_rd_wr
31dt31	err_lock	0h	rh	w	err_lock

Register:		Burst_Config			Address:	80Ch	
Bits:	31dt0	Reset value:		0003h	Attributes:	(r)	(w)
Description:		Burst_Config					
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	BurstMode_comAHB	3h	r	w	At the communicative AHB slave interface, an AHB RD/WR burst is "x0": not supported (→ single access executed) "01": supported for all burst types except for INCR (→ single access executed) "11": supported for all burst types (INCR, INCR4, ... , WRAP4, ... WRAP16) at the SC bus.		
7dt2	reserved	0h					
9dt8	BurstMode_applAHB	0h	r	w	At the applicative AHB slave interface, an AHB RD/WR burst is "x0": not supported (→ single access executed) "01": supported for all burst types except for INCR (→ single access executed) "11": supported for all burst types (INCR, INCR4, ... , WRAP4, ... WRAP16) at the SC bus.		
31dt10	reserved	0h					

Configuration rule:

If the data to be sent cyclically, which the PN-IP requests at the FSO (FrameSendOffset) time by accessing the PerIF.IO-RAM, are to be provided in time (max. time: 1280ns, equal to the header feed of 16 bytes), register **Burst_Config = 0x0003** (default value) **must be configured**.

This ensures that burst access by the PN-IP (communicative instance) is treated as such at the IO-RAM while burst access by the applicative instances (host interface, ARM and

GDMA) is sent to the IO-RAM as instances of single access; this minimizes the lockout time for the communicative instance for the IO_RAM.

2.3.8 PROFINET-IP

2.3.8.1 PN-IP Interfaces

2.3.8.1.1 AHB Interface

The PN-IP is connected to the multi-layer AHB over an AHB master and an AHB slave interface.

Over the AHB master interface, the PN-IP can transfer data over the multi-layer AHB as the active bus master. At the same time, other AHB masters can access PN-IP resources over the AHB slave interface.

The AHB interfaces of the PN-IP do not support split or retry functions. All other transfer types, transfer sizes and burst operations as per the AMBA specifications are supported (see chapter 7.2.)

2.3.8.1.2 Interrupt Management

All interrupt events generated by mechanisms of the PN-IP can be provided over CPUs connected with Interrupt Management. Three CPU systems (CPU subsystems) are supported. These are as follows for the ERTEC 200P:

- ARM926 subsystem
- Event unit in PER-IF (external host CPU)

PN stack and application SW allocation in the CPU systems is not fixed. Individual interrupt event assignment to interrupt inputs / interrupt controllers of the CPUs is therefore universally applicable. For the ARM926 and host subsystem, the specific number of all interrupt events occurring in the PN-IP is provided. This is done as follows:

- with a CPU-specific interrupt controller (PN-ICU) that generates two group interrupts each for all interrupt events for the ARM926 (PN-ICU 2) / external host (PN-ICU 3) → PN_IRQ2/3(1:0)
- with a CPU-specific multiplexer structure (PN-MUX) that provides 14 selectable PN interrupt events for the ARM926 (PN-ICU 2) / external host (PN-ICU 3) as single interrupts → PN_IRQ2/3(15:2).

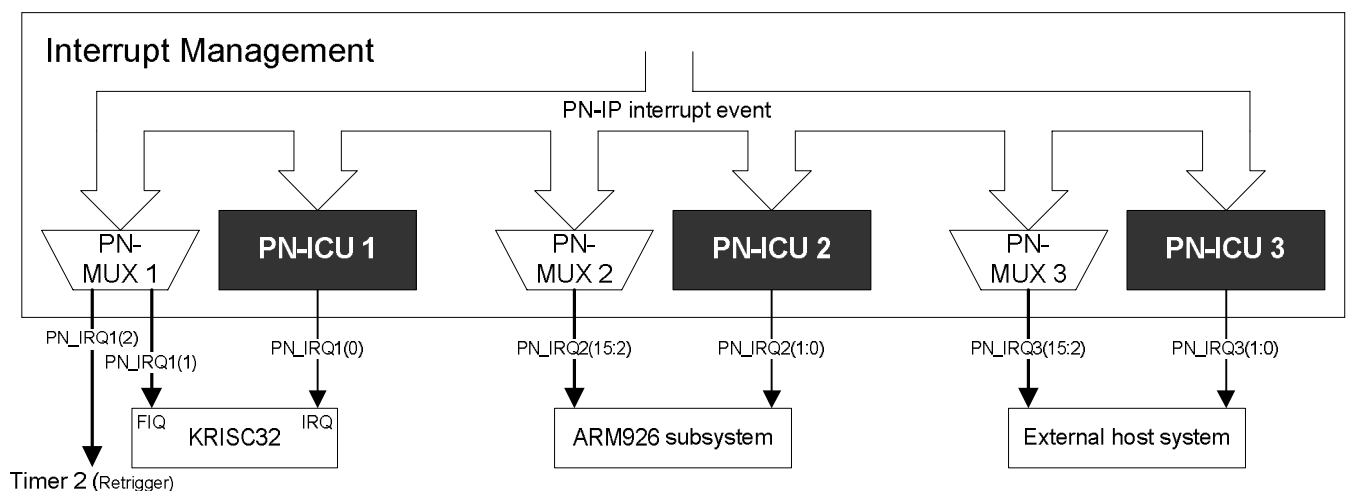


Figure 5: Diagram of Interrupt Management in the PN-IP

The group interrupts PN_IRQ2(1:0) (IRQ56/57) and 14 selectable single interrupts PN_IRQ2(15:2) (IRQ58-71) are connected to the ARM926EJ-S interrupt controller (ARM-ICU) by the PN-IP (see 2.3.2.14).

Interrupts for the host are generated by the event unit in the peripheral interface. The PN-IP forwards the relevant interrupts (group interrupts: PN_IRQ3(1:0), single interrupts: PN_IRQ3(15:2)) to the event unit, Where a group interrupt signal to the external host (ASIC pin: XHIF_XIRQ) is generated.

The group interrupt PN_IRQx(0/1:0) is generated as shown below:

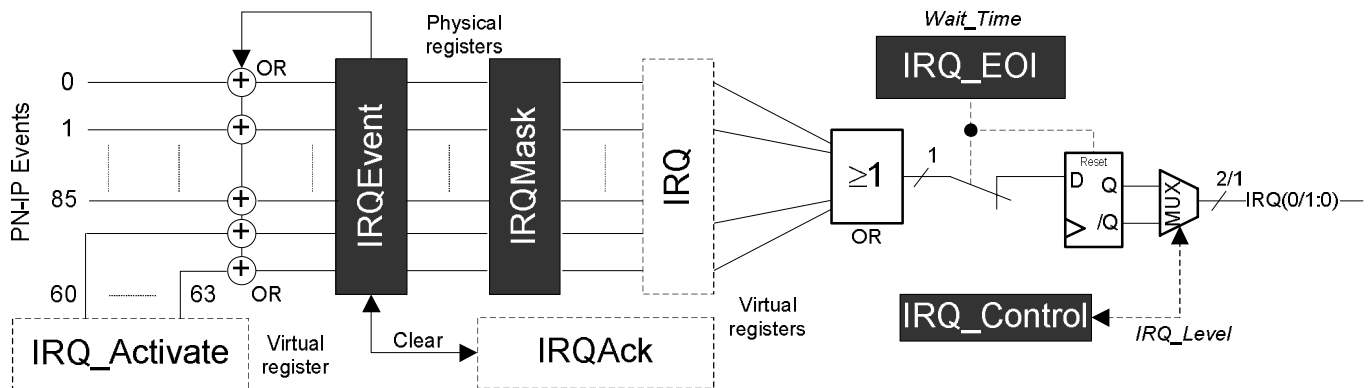


Figure 23: Block diagram of PN-ICU for PN_IRQx(0/1:0) group interrupts

For each group interrupt (2x for ARM926 / external host: PN_IRQ2/3(1:0): PN_IRQ1(0), there is one interrupt controller. IRQEvent registers are implemented for signaling and saving internal PN-IP events. The same events are connected to all interrupt controllers. One or more event bits set can each trigger a group interrupt for the external CPU sub-system → PN_IRQx(0/1:0). For all event bits, the interrupt is triggered by:

- internal PN-IP events (PN-IP events) OR
- dedicated AHB write access to IRQEvent OR
- dedicated AHB write access (SW events over IRQ_Activate).

Event bits already written/set are retained and not reset by subsequent access to the IRQEvent registers. The assignment and significance of the individual event bits is identical for all PN-ICUs (1..3).

The decision whether an interrupt event (set event bit) triggers the corresponding PN_IRQx(0/1:0) group interrupt is configured with event bit masking with the IRQMask registers. A set mask bit blocks the corresponding entry and prevents PN_IRQx(0/1:0) from being triggered. Over the IRQ registers, the event bits that trigger a PN_IRQx(0/1:0) group interrupt (IRQ bits) can then be read. Writing to these registers is ignored. The two group interrupts are masked differently by the SW to create interrupt groups (e.g. acyclic API and cyclic API by group).

Bits in the IRQEvent registers are reset by write access to the IRQAck registers. The PN-ICU may only be operated in "**Acknowledge**" mode, i.e. the event bits set are established with read access to the IRQ registers. Subsequent write access to the IRQAck registers resets the register bits written in the IRQEvent registers and therefore also the IRQ bits in the IRQ registers.

PN_IRQx(0/1:0) group interrupts are deactivated over the IRQ_EOI registers. Write access to the registers resets the PN_IRQx(0/1:0) group interrupts. The PN_IRQx(0/1:0) group interrupt cannot be reactivated by set event bits in the IRQEvent registers until the wait time (Wait_Time) written is up.

The PN-PLL contains a clock instance (Clock_A), a time instance and 3 application timer blocks. Each application timer block controls its own application cycle and is connected to the Clock_A instance. The application clock (APPL1/2/3_CLK_INT) and 6 programmable phase signals (APPL1/2/3_CMP_INT5...0) are available as outputs for each application timer block. These 21 outputs are connected to a 21-input multiplexer (PNPLL_OUT20...0). Any output of the 3 application timer blocks can be selected at each MUX output. The controlled clock of the clock instance (PNCLKA_OUT) can also be connected to the MUX outputs.

Figure 24 shows how these outputs (PNPLL_OUT20...0) are connected to the application instances at the ERTEC 200P top level.

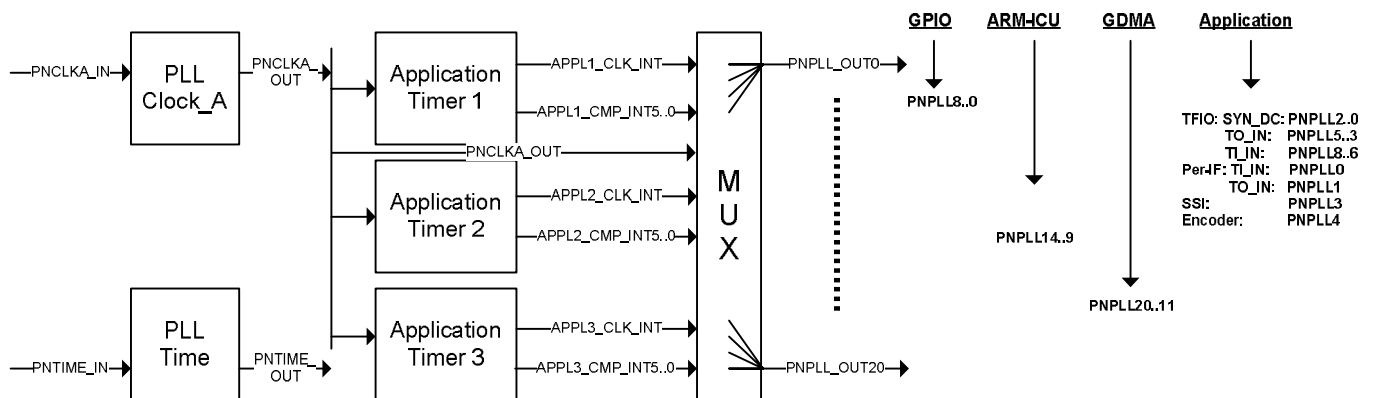


Figure 25: PNPLL with 3 application time blocks (application connection)

The connection matrix is as follows:

	GPIO	ARM-ICU	GDMA		
PNPLL0	GPIO0_INT				
PNPLL1	GPIO1_INT				
PNPLL2	GPIO2_INT				
PNPLL3	GPIO3_INT				
PNPLL4	GPIO4_INT				
PNPLL5	GPIO5_INT				
PNPLL6	GPIO6_INT				
PNPLL7	GPIO7_INT				
PNPLL8	GPIO8_INT				
PNPLL9		IRQ72			
PNPLL10		IRQ73			
PNPLL11		IRQ74	HWJob(0)		
PNPLL12		IRQ75	HWJob(1)		
PNPLL13		IRQ76	HWJob(2)		
PNPLL14		IRQ77	HWJob(3)		
PNPLL15			HWJob(4)		
PNPLL16			HWJob(5)		
PNPLL17			HWJob(6)		
PNPLL18			HWJob(7)		
PNPLL19			HWJob(8)		
PNPLL20			HWJob(9)		

The overlap of the PNPLL signals (e.g. PNPLL8..0) with different IPs is necessary to cover the different use cases. However, there is no overlapping within any one use case, i.e. multiple IPs are **never** triggered with the same PNPLL signals. The PNPLL signals that overlap always go to inactive IPs. Behavior is as follows:

- With GPIO, the PNPLL8..0 signals must in this case not be connected to the pins.
- With the ARM-ICU, the PNPLL14..9 signals must set the relevant interrupt masks.
- With GDMA, the PNPLL20..11 signals must not be selected for job control.

2.3.8.1.4 EDC in PN-IP RAMs

The memories in the PN-IP have an EDC code (1-bit error, correctable; 2-bit error, detectable). The PN-IP has the group outputs PN-1B (1 bit error correct) and PN-2B (2 bit error detected). If an error occurs during reading, the cause of the error (PN-1B: 1 bit error or PN-2B: 2 bit error) is saved in the SCRB register 'EDC Event' (see) and the interrupt 'EDC_Event' IRQ48 is triggered (see 2.3.2.14). 0x00...00 must be written to the EDC event register to clear the error. The PN-IP RAM is initialized after this reset. The end of initialization (incl. EDC bits) is signaled in the SCRB register 'EDC_INIT_DONE' (see 2.3.10.9.22).

2.3.8.1.5 Recommended Parameter Assignment for ERTEC 200P

The PN-IP has no internal RAM for managing the data to be sent and received. Instead, it uses either the IO-RAM in the PerIF module (see 2.3.7) or external memory connected to the EMC (see 2.3.5). For high-performance data transfer from/to the PN-IP, we recommend configuring the ERTEC 200P modules listed below as detailed.

Recommended parameter assignment for time-triggered injection with PPM and the receipt of cyclic RT frames by the CPM:

Module	Register	Comment
PerIF	Burst_Config(7:0) = 0x03	Communicative AHB access is implemented as burst access at the IO-RAM. Please note that the PerIF interrupt CR_State_comErr_INT can be triggered in error in the settings (see page 71 of PerIF Spec V2.0). Remedy: one of the two options below: - Mask interrupt in the PerIF registers Host_IRQmask_low and PN_IRQmask_low (this interrupt is merely a debug feature). - Make the CR one 32-bit word greater than the user data read by the PPM / written by the CPM.
	Burst_Config(15:8) = 0x00	Applicative AHB access is implemented as single access at the IO-RAM. This is necessary to ensure in-line PPM/CPM data transfer (communicative AHB access).

Recommended parameter assignment for CPM cut-through receipt at an external memory connected to the EMC:

Module	Register	Comment
PN IP	CPM_AHB_Lockout_Starttime CPM_AHB_Lockout_Stoptime CPM_AHB_Lockout_Control	Use of the registers listed to block acyclic Snd-/Rcv-API access to the AHB bus for a configurable time in the communication cycle.
SCRB	AHB_BurstBreaker(7: 0) = 0x08	For the ARM926I and ARM926D AHB master inter-

	AHB_BurstBreaker(15:8) = 0x08	face, the AHB burst length is to be limited to 8 in the AHB_BURSTBREAKER register (default is no limit)
GDMA	Burst_Mode = Single - INCR8	GDMA jobs whose source or destination address points to an address range of the EMC must be limited to single, INCR4 or INCR8 in Burst_Mode (i.e. do not use INCR16)
EMC	Asyn Memory Banks 0-3 Waits	Access by other masters (ARM926D, ARM926I, GDMA or HostIF) to EMC asyn memory banks 0-3 delays CPM data transfer to the backplane bus ASIC, which is also connected to one of the 4 banks. The wait states of the other 3 banks (either set in the configuration or created by an external "ready") must therefore be included in the calculation of the CPM transfer delay if they are greater than the time required for an 8-transfer burst to the SDRAM and other masters access the CPM during CPM data transfer.

2.3.8.2 Ethernet PHY (integrated)

A 2-port multiport PHY (Physical Layer Transceiver) that supports 10BASE-T, 100BASE-TX and 100BASE-FX is integrated into the ERTEC 200P. The interfaces for 10BASE-T/100BASE-TX and 100BASE-FX are separately for each port and can be set differently. The PHYs are fully compatible with IEEE802.3 and other standards (ANSI X3.263-1995 and ISO/IEC9314 etc.).

The PHYs are controlled by the PNIP software.

The PHY supports the following functions:

- 10Base-T/100Base-TX PHY (10 MBd is not supported by ERTEC 200P; the SW must take account of this)
- MII/RMII MAC interface (ERTEC 200P only uses MII)
- Support Fiber mode
- Auto Cross Over
- Auto MDIX
- Auto RX polarity in 10BT
- Fast link up time
- Next Page support
- Single Central PLL and BIAS gen
- Best in class latency (190ns for Transmit and Receive)
- Jitter free Latency
- Fast line break detection
- TDR based enhanced cable diagnosis

The data interface to the Ethernet MACs is over MII and the management interface is over the MDIO interface (SMI interface). The clock source is 25 MHz and is provided either with a 25 MHz quartz at the CLKP_A/B external pins or a 25 MHz external clock supply at the CLKP_A pin.

2.3.8.2.1 Status of the integrated PHY

2.3.8.2.1.1 PHY Status LED

The following PHY status LEDs are connected to the pinning (see 3.2):

Dedicated PHY LEDs:

- P1/2_XLINK_STATUS (Link / No Link)
- P1/2_XACTIVITY (Receive or Transmit Activity / No Activity)

These LED outputs can be controlled directly from the multiport PHY. P1/2_XActivity combines the receive and transmit activity outputs of the multiport PHY. Alternatively, these LED outputs can also be controlled by the SW. SW control is with the PHY_LED_CONTROL register in the SCRB (see 2.3.10.9.22). The status of the PHY outputs 'P1/2_XLINK_STATUS / P1/2_XACTIVITY' can also be queried with the PHY_LED_CONTROL register if the SW is controlling the LEDs.

2.3.9 CRU (Clock and Reset Unit)

2.3.9.1 Clock System

The ERTEC 200P clock system consists primarily of four clock groups which are isolated with asynchronous crossings. These systems are as follows:

- ARM926EJ-S processor system
- JTAG interfaces
- AHB system, APB system, PN-IP and PER_IF
- Host interface (parallel: XHIF, serial: host SPI)
- PHYs + Ethernet MACs

2.3.9.2 Clock Generation and Distribution

The ERTEC 200P clocks required are provided over internal PLL and/or with direct supply. Apart from the JTAG clock and the PHY clock, all clocks are generated by the integrated PLL.

MODULE	CLOCK SOURCE	FREQUENCY
ARM926ES	PLL	250/125 MHz
AHB/EMC/ICU/HOST-IF/GDMA	PLL	125 MHz
PN-IP (except MAC MII/GMII)	PLL	125/250 MHz
PER_IF	PLL	125 MHz
APB	PLL	125 MHz
JTAG	JTAG clock	16/32 MHz
MAC MII/GMII / PHY	CLKP_A / external	25/125 MHz

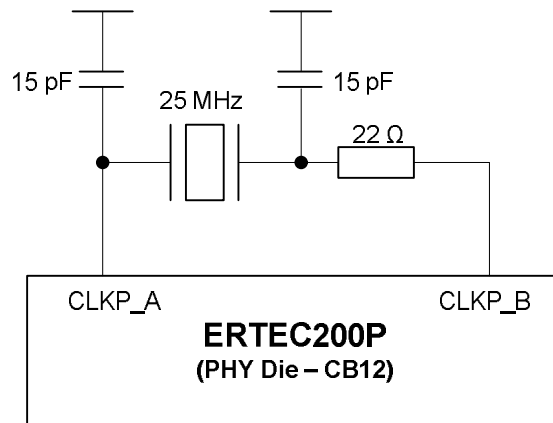
Table 16: Overview of ERTEC 200P clocks

In ERTEC 200P, the clock is distributed to the various modules. The PHYs start in the disable state and must be enabled by the SW.

2.3.9.2.1 PLL Clock Generation

As shown in Table 16, ERTEC 200P largely runs with a synchronous 125 MHz or 250 MHz clock. These clocks are generated with an integrated PLL. The PLL is supplied with a 25 MHz clock signal. The PLL clock source is an integrated quartz oscillator, to which an external 25 MHz quartz must be connected (CLKP_A/B). Alternatively, an external 25 MHz generator can be connected to the CLKP_A pin.

The external quartz wiring is shown in Figure 26 (the capacity and resistor values given here are valid for the TSX-3225 quartz types from Epson):



Note:
The input capacity of the integrated oscillator TDOSAC33N32M in the ERTEC200+ is **4.2pF** (this includes the effect of the package)

Figure 26: Quartz wiring

The PLL provides 500 MHz at its output. In clock generation downstream from the PLL, the PLL output frequency is converted into:

- 125 MHz → System clock (CLK_SYS)
- 250 MHz → Synchronization in the PN-IP (CLK_TIME)
- 125/250 MHz → ARM926 (CLK_ARM_SLOW/FAST)

It is also possible to bypass the PLL with the configuration pin CONFIG(2) (see 2.3.10.9.3). In this case, clock supply is over the BYP_CLK pin. With a recommended input frequency of 125 MHz, the output clocks are 125 MHz for ARM926EJ-S, system clock and synchronization. This PLL bypass is only useful for debug operation.

2.3.9.2.2 Clock Source for the PHYs and Ethernet MACs

Both Ethernet MACs are connected directly to the integrated PHYs over MII. The clock source for the PHYs (25 MHz) is from the quartz oscillator on the PHY die. This clock also functions as the input clock for the ERTEC die (see 2.3.9.2, Figure 74). The PHYs generate the clock signals RX_CLK and TX_CLK for the relevant Ethernet MACs. The 25 MHz can be output over the REF_CLK pin when external PHYs are operated. This clock output is enabled and disabled with the configuration pin CONFIG(0), see 2.3.10.9.3.

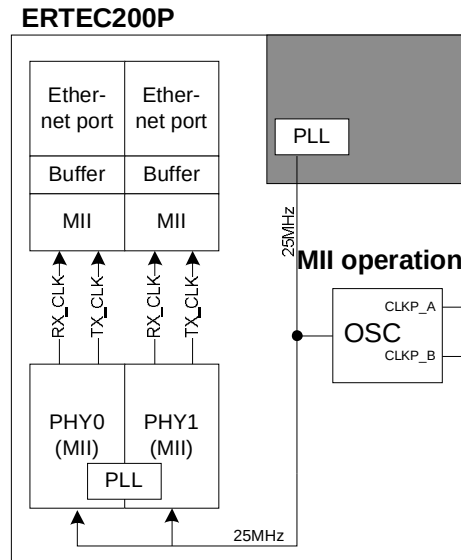


Figure 27: Clock source for the Ethernet connection

2.3.9.2.3 Clock Source for JTAG

The ARM926EJ-S debug interface and the ARM926EJ-S ETM / ETB macrocell are operated over the JTAG interface. The clock source is from a separate JTAG-TCK. The JTAG-TCK frequency is 16/32 MHz. The max. for debuggers that do not support the RTCK clock at the JTAG interface is 16 MHz; otherwise, 32 MHz is available.

2.3.9.3 Clock Monitoring

2.3.9.3.1 Lock Timer 1

One-Shot Timer generates a logic zero signal at the STBY output after a minimum time of **2,5µs**.

Features:

- Operation frequency of 25 MHz
- Starts counting after XRESET is changing to high
- Synchronized reset signal needed (has to operate as an asynchronous reset).
- STBY-Output goes a logic zero after approx. **2,5µs** and remains set until the next reset.

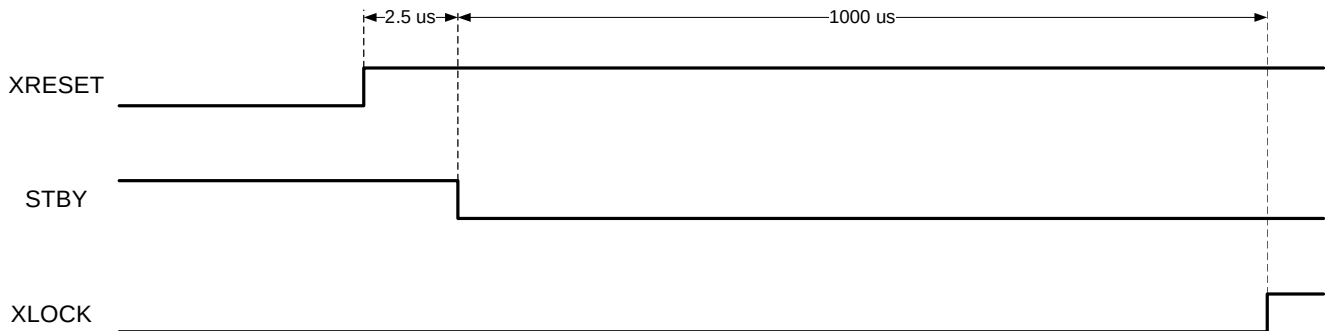
2.3.9.3.2 Lock Timer 2

One-Shot Timer generates a logic one signal at the XLOCK output after a minimum time of **1000µs** after STBY was activated.

Features:

- Operation frequency of 25 MHz

- Starts counting after XRESET is changing to high
- Synchronized reset signal needed (has to operate as an asynchronous reset).
- XLOCK-Output goes a logic one after approx. **1000µs** after STBY was activated and remains set until the next reset.



2.3.9.3.3 Lock Monitor

Features:

- Monitors the output of the oscillator cell and sets the LOSS signal if the oscillator frequency is not working.
- Monitors the output of the PLLs and sets the LOCK signals if the PLL frequencies are within the defined limits.
- All signals are synchronized twice with 125MHz clock signal
- Each of the signals triggers its own interrupt (PLL-Lock, PLL-Loss, see chapter 2.3.2.14).

Limitations:

- The free running frequency of the PLL is max. 300MHz (100MHz ... 300MHz) when the input of PLLs is missing (open, clamped low, clamped high).
- As long as the frequency ratio between the PLL input and the PLL output remains unchanged, the Lock-Monitor does not detect a drift of the frequencies.

Function:

The LOSS signal:

- The LOSS signal is generated with 500MHz. The signal is synchronized to 125MHz clock.
- The generation of the LOSS signal is disabled as long as the Lock-Monitor is disabled (i.e. XRESET = 0).
- The LOSS signal is the output of a count down counter. If the counter reaches zero, then the LOSS signal is set.
- Each rising edge of the oscillator output retriggers the count down counter.
- The count down counter expires, if the oscillator is broken (open) or clamped (tied to GND or VDD). The count down counter **not** expires, if the period of the oscillator frequency rises above e.g. 2.5 times (< 10MHz) of the nominal oscillator frequency (25MHz) because the frequency ratio (20) between rises oscillator frequency (PLL input, e.g. 10MHz) and resulting PLL output (e.g. 200MHz) will be reached within the distance of measure.
- The counter operates with the output frequency of the 500 MHz PLL (i.e. in case of an error the PLL will change its output frequency towards the free running frequency of the PLL).

Note:

Evaluation of the LOSS mechanism by the FW is guaranteed as PLL free-running frequency (100MHz – 300MHz) < PLL operating frequency (500 MHz).

The LOCK signals:

- The generation of the LOCK signals is disabled as long as the Lock-Monitor is disabled (i.e. XRESET = 0).
- The core of the LOCK logic are a count up counter that increment with the output frequencies (500MHz) of the PLL. With each rising edge of the oscillator output:
 - o the value of the counter are checked against the limits (+/- 15%) compared to the nominal value (< 425MHz or > 575MHz) for LOCK signal is inactiv
 - o the value of the counter are checked against the limits (+/- 10%) compared to the nominal value (> 450MHz and < 550MHz) for LOCK signal is activ
 - o The counter are reset to zero.
- The LOCK signal is active if the ratio between the oscillator frequency and the output frequencies of the PLLs are within the limits (+/- 10%) for more than three oscillator periods.
- The LOCK signal is inactive, if the ratio between the oscillator frequency and the output frequencies of the PLLs are outside the limits (+/- 15%) for more than three oscillator periods.
- Note that “three oscillator periods” means an integral action.

Note:

*If the deviation from the PLL output frequency is caused by the wrong oscillator input frequency, i.e. an oscillator input frequency that does not correspond to the rated value (25 MHz), this **cannot** be detected by the LOCK monitor because the PLL has already adjusted to the new output frequency during easurement (3 oscillator clocks) in line with the specified frequency (20).*

2.3.9.4 Reset System

There are a number of options for asynchronous ERTEC 200P resets alongside an external PowerOn reset (XRESET): an additional external hardware reset by the debugger (XSRST), an internal ARM926 watchdog reset (XRES_ARM926_WD) and a number of resets with the software. The reset matrix shows all reset sources (rows) and their implications for each circuit component (columns).

A PowerOn reset (XRESET) is applied to ERTEC 200P with an external pin. This PowerOn reset resets the entire circuit (incl. clock system). The XSRST pin is used for a hardware reset by the debugger. The clock system is not reset and communication over the JTAF interface is possible during this reset phase. The ERTEC 200P can be monitored with an ARM926 watchdog. An ARM926 watchdog event (XRES_ARM926_WD) then resets the ERTEC 200P. The SCRB register 'ASYN_RES_CTRL_REG' (see 2.3.10.9.22) allows you to exclude the PN-IP from a watchdog reset (EN_WD_RES_PN).

'ASYN_RES_CTRL_REG' can also be used by the software to trigger an internal asynchronous reset without the PN-IP (RES_SOFT), an internal asynchronous reset for the PN-IP (RES_SOFT_PN) only, an internal asynchronous reset for the ARM926 core system (RES_SOFT_ARM926_CORE). The trigger event for the last reset for the ARM926EJ-S core can be read from the SCRB register RES_STAT_REG (see 2.3.10.9.22).

642
643
644
645

Legend

 Reset on the modul

Copyright © Siemens AG 2016. All rights reserved
Technical data subject to change

2.3.9.4.1 Asynchronous PowerOn Reset

The asynchronous⁶ PowerOn reset is connected to the ERTEC 200P with the XRESET pin. In response to this reset, the complete circuit (incl. clock system) of the ERTEC 200P is reset and the configuration pins are latched (see 2.3.10.9.3). The PowerOn reset must be applied steadily for at least 30µs after a steady voltage is reached upon ERTEC 200P startup. The PLL then starts up and after a further 1000 µs, the PLL is locked. This time until the PLL locks is t_{LOCK} . Internally, the PowerOn reset phase is extended by this time (fixed setting; the PLL lock is not evaluated) and the clock system is not connected until the end of the startup phase. The internal reset remains active for a further 16 clocks after clock system startup to execute the reset internally. Debugger communication over the JTAG interface is not possible during this time.

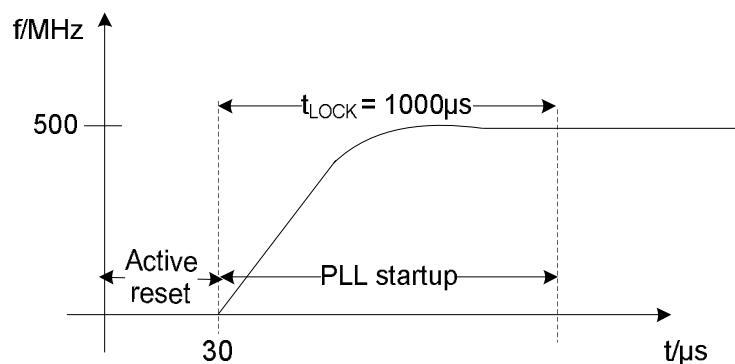


Figure 29: PLL startup phase

Hardware monitors the locked state of the PLL. The IRQ49 interrupt signals whether the PLL has lost its input clock (quartz break) or the PLL is not locked (PLL monitor, monitors input and output frequency). The two error states can also be queried directly from the SCRB register 'PLL_STAT_REG' (see 2.3.10.9.22).

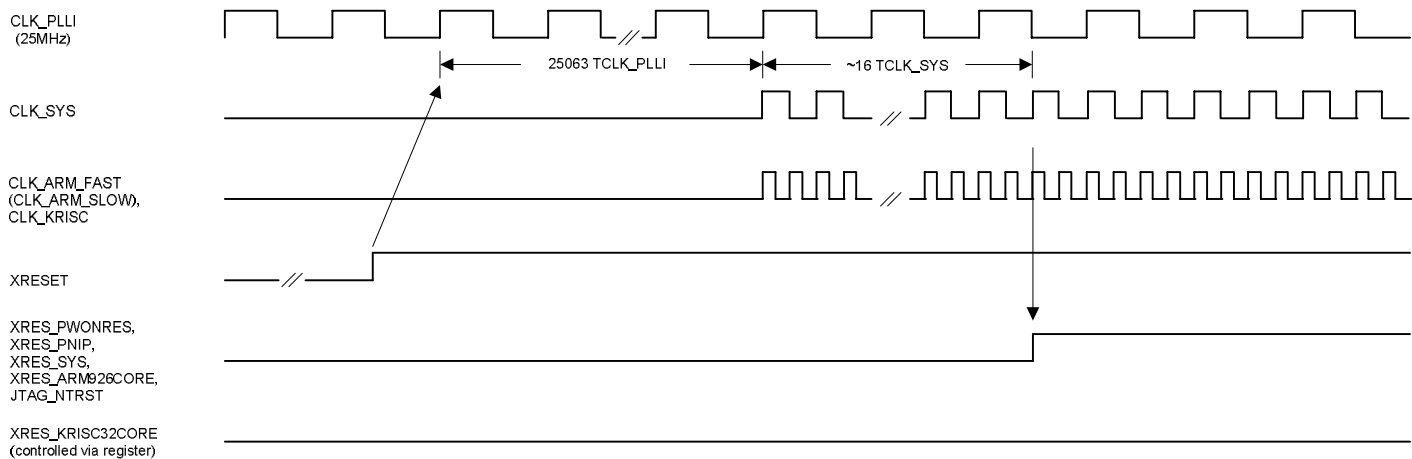
A filter integrated in ERTEC 200P ensures that spikes ≤ 40 ns (best case) at the XRESET input are suppressed.

While the XRESET pin is active, the bidirectional pin XSRST is switched to output and activated. The debugger can then detect the PowerOn reset phase.

The PWRON_HW_RES bit in RES_STAT_REG is set during a PowerOn reset to allow an analysis of the reset event after a system restart. This bit is not affected by the reset function triggered. Upon restart, the software can read RES_STAT_REG (see 2.3.10.9.22).

⁶ An asynchronous reset affects the reset input of a flip-flop. This reset is applied asynchronously but cleared synchronously.

2.3.9.4.1.1 Reset Timing (PowerOn Reset)



2.3.9.4.2 Asynchronous Hardware Reset

The hardware reset is triggered with the XSRST pin by the external debugger. XSRST is a bidirectional IO cell with an open drain output. The complete internal logic is reset in the active XSRST phase, but not the clock system. The configurations pins are also **not** latched. During this hardware reset phase, the debugger can communicate with the embedded ICE logic over the JTAG interface, for example to load a breakpoint. Single stepping is therefore possible from the reset address.

A filter integrated in ERTEC 200P ensures that spikes $\leq 40 \text{ ns}$ (best case) at the XRESET input are suppressed.

The PWRON_HW_RES bit in RES_STAT_REG is set during a hardware reset to allow an analysis of the reset event after a system restart. This bit is not affected by the reset function triggered. Upon restart, the software can read RES_STAT_REG (see 2.3.10.9.22).

When booting after a hardware reset, the system uses the boot mode latched internally during the PowerOn reset.

XSRST is activated for the debugger if a PowerOn reset is active. XSRST must never be activated with 'RES_SOFT_ARM926_CORE', as this would prevent the debugger from running while one of the two cores was in reset.

2.3.9.4.3 Asynchronous JTAG Reset

The JTAG reset is triggered with the XTRST pin by the external debugger. Only the Embedded ICE logic of the ARM926EJ-S is reset. To ensure that this Embedded ICE logic enters a defined state in operation without a debugger, it is also reset by a PowerOn reset (XRESET). An internal logic operation is implemented for this purpose. A filter integrated in ERTEC 200P ensures that spikes $\leq 40 \text{ ns}$ (best case) at XTRST are suppressed. A spike at XTRST is not usually forwarded to the JTAG controller as this would require a sequence over TDI/TMS and TCK.

Note for HW-Development:

For rules for the external pull circuit for the XTRST pin, see 0.

2.3.9.4.4 Asynchronous ARM926 Watchdog Reset

The ARM926 watchdog reset is hardware monitoring of the software on the ARM926EJ-S. The basis for monitoring is a time set in the watchdog timer. This time starts to run when the watchdog is activated. If the timer is not retriggered to its initial value during this time, a watchdog reset (XRES_ARM926_WD) is triggered (output ARM926 watchdog: WD_XWDOUT1). If the watchdog function is enabled (WD_RES_FREI_ARM926) (see ASYN_RES_CTRL_REG in 2.3.10.9.22), ERTEC 200P is reset. The actual reset signal is connected with configurable pulse generation. The ARM926_WDOG_RES bit in RES_STAT_REG is set during an ARM926 watchdog reset to allow an analysis of the reset event after a system restart. This bit is not affected by the reset function triggered. Upon restart, the software can read RES_STAT_REG (see 2.3.10.9.22).

If, in a given application, you do not want expiry of the watchdog time to affect the operation of the PN-IP, you can exclude the PN-IP from the watchdog reset (EN_WD_RES_PN = 0 in ASYN_RES_CTRL_REG).

Important: The EN_WD_RES_PN bit is always set (→ PNIP subject to ARM926 watchdog reset) if the PNIP is reset. If an asynchronous software reset is to be generated by the SW for the PN-IP (see 2.3.9.4.6), the SW must if necessary then set 'EN_WD_RES_PN = 0' again if you do not want a PN-IP reset at the end of the ARM926 watchdog time.

Before the watchdog expires, an interrupt is generated for the ARM926 'WD_INT_ARM926' (see 2.3.2.14) and the preliminary event 'WD_XWDOUT0' is signaled to the external host over a GPIO pin.

The watchdog also runs when the clock source fails (e.g. quartz break). In this case, the PLL then switches to its free-running frequency (100 – 300 MHz).

When booting after a watchdog reset, the system uses the boot mode latched internally during PowerOn reset.

2.3.9.4.5 Asynchronous Software Reset for ERTEC 200P (Without PN-IP)

In the ERTEC 200P, an asynchronous software reset can be triggered by setting the 'RES_SOFT' bit in RES_CTRL_REG (in SCRB, see 2.3.10.9.22); the PN-IP and PHYs are not reset. The SW_RES bit in RES_STAT_REG is set during an asynchronous software reset to allow an analysis of the reset event after a system restart. This bit is not affected by the reset function triggered. Upon restart, the software can read RES_STAT_REG (see 2.3.10.9.22).

When booting after a software reset, the system uses the boot mode latched internally during PowerOn reset.

2.3.9.4.6 Asynchronous Software Reset for PN-IP

The PN-IP and PHYs can only be reset asynchronously by the software with the 'RES_SOFT_PN' bit in the SCRB register 'ASYN_RES_CTRL_REG' (see 2.3.10.9.22).

The HW reset for the integrated PHYs is triggered by the 'phy_reset_o' output of the PN-IP. Whenever the SMI module in the PN-IP is not activated (configuration mode), 'phy_reset_o – output' is active and keeps the PHYs in the reset state. A simultaneous asynchronous SW reset for the ERTEC 200P and the PN-IP resets the complete ERTEC 200P.

2.3.9.4.7 Asynchronous Software Reset for the ARM926EJ-S Core

The ARM926EJ-S core (without TCM926) has its own reset, which can be executed asynchronously by the software with the 'RES_SOFT_ARM926_CORE' bit in the SCRB register 'ASYN_RES_CTRL_REG' (see 2.3.10.9.22). 'RES_SOFT_ARM926_CORE' only affects the ARM926EJ-S core system and not TCM_Block_926 (see Figure 12). TCM_Block_926 is reset with XRESET, XSRST, XRES_ARM926_WD or RES_SOFT.

The SW_RES_ARM926 bit in RES_STAT_REG is set during an asynchronous software reset for the ARM926EJ-S core system to allow an analysis of the reset event after a system restart. This bit is not affected by the reset function triggered. Upon restart, the software can read RES_STAT_REG (see 2.3.10.9.22).

The asynchronous software reset for the ARM926EJ-S core system is needed once the boot loader has set the final TCM926 configuration. The TCM926 configuration (DRSIZE for D-TCM and IRSIZE for I-TCM → from TCM926_MAP register, see 2.3.10.9.22) is only applied to the ARM926EJ-S after a reset.

2.3.9.4.8 Synchronous Software Reset (PN-IP, PER-IF, Host Interface)

The PN-IP, peripheral interface and host interface can be reset synchronously by the software in the SCRB register 'SYN_RES_CTRL_REG' (see 2.3.10.9.22). These synchronous resets affect the SYN reset inputs of the corresponding IPs and reset only the state machines and the local registers and not the parameter registers or the AHB interface. The synchronous reset does not affect the reset input of a flip-flop. The SW must set and then reset the relevant bits in 'SYN_RES_CTRL_REG'. This allows the SW to set the reset state itself (→ short pulse: dyn. reset, continuously '1': disable state).

2.3.10 APB Peripherals

The function blocks connected to the APB have interfaces of differing widths. The table below shows the access mechanisms supported and the data widths of these blocks.

Access modes				Wait states at the AHB		
Bit 31:24	Bit 23:16	Bit 15:8	Bit 7:0	Read	Write	Function block
8-bit	8-bit	8-bit	8-bit	2	0	IO filter, Boot_ROM
16-bit		16-bit				
32-bit						
8-bit	8-bit	8-bit	8-bit	Ready	0	PER_IF (except PER_IF-GPIO)
16-bit		16-bit				
32-bit						
32-bit				2	0	GPIO, PER_IF-GPIO, timer0-5, F-counter, watchdog, SCRB, SPI1/2, flash controller, UART1-4, I ² C, Host interface

Table 17: Data width of peripherals

Access modes that are not permitted (e.g. writing byte by byte / halfword by halfword to the timer) are not prevented by the hardware, nor are they indicated at the AHB in the form of an error response or QVZ interrupt.

Access to memory ranges that are not decoded (marked as "not used" in memory mapping) triggers IRQ52 (see 2.3.2.14) and ends with a ready signal generated by the APB address decoder. Write access does not have any effect on the system. Read access returns undefined data.

2.3.10.1 I-Filter

Up to 80 input signals can be sent by the GPIO interface (GPIO15-0, GPIO95-32) through an I-filter. Please see 3.2 for the corresponding I-filter input for each GPIO. The pending input signals D_IN can be saved unfiltered and directly (mode 1: no synchronization), unfiltered and synchronized (mode 2) or filtered (mode 3) in the modules below. In all cases, the input signals are sent through the I-filter which operates on a channel-specific basis and can be configured over the APB.

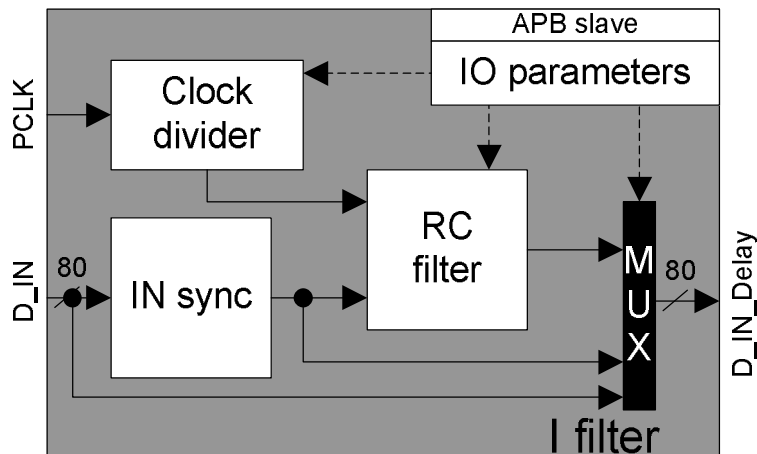


Figure 30: Block diagram of the IO filter

The APB module "I-Filter" allows you to filter out brief signal spikes and interference. Filtering is done using a synchronization unit (IN_Sync) and filter stage (RC_Filter) in the module, which are available for each input (channel). If you do not want to filter an input signal, either the synchronization stage of the I-filter is used (mode 2) or the input signal itself is forwarded without processing by the I-filter (mode 1). The filter stage is supplied by a central clock divider, whose configurable clock sources can also be selected on a channel-specific basis. The block diagram below shows exactly how the I-filter operates, using the example of one channel:

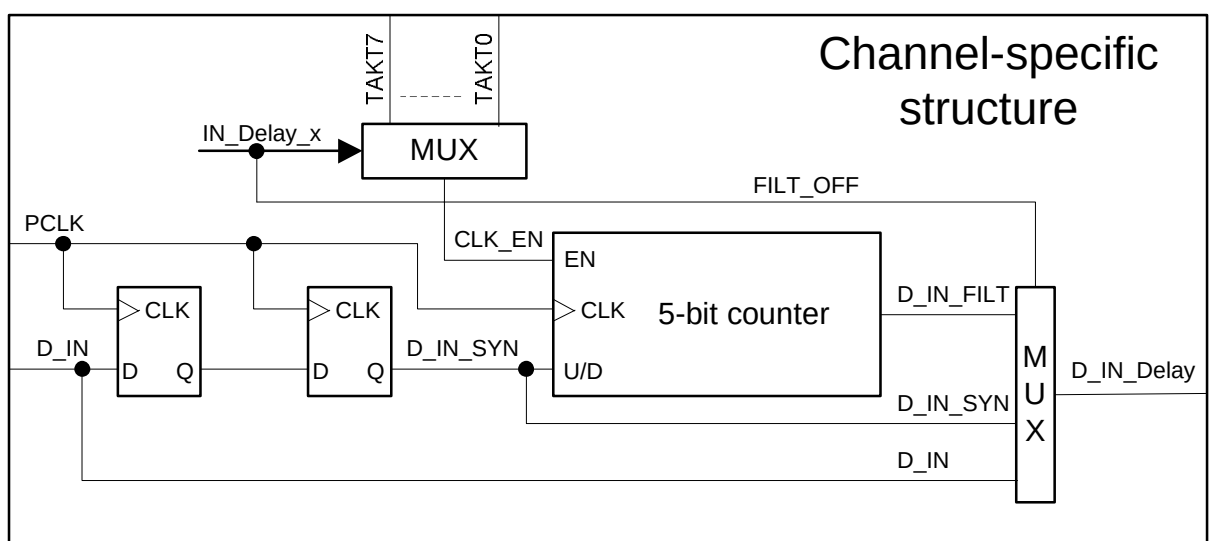


Figure 31: Block diagram of the filter structure of a channel

The input signal is filtered following two-stage synchronization (this is required to avoid metastable states). The channel-specific filter is implemented as a 5-bit up/down counter that counts up when the input signal is "high" and down when the input signal is "low". The counter value after RESET is '00h' (load value).

As the input channels for the digital input modules are operated separately, each input signal requires its own counter. These counters must enable different, channel-specific filter time settings. The filter time is selected in the 'FILT_Delay_x registers', which in line with the settings select different clock sources on a channel-specific basis (mode 3) or select input signal synchronization only without RC filtering (mode 2). In mode 2, the doubly synchronized input signal D_IN_SYN is connected straight to the output signal D_IN_Delay and not through the counter register. In mode 1, the input signal D_IN is connected straight to the output signal D_IN_Delay, bypassing the synchronization stage and counter register. Dynamic FILT_Delay_x toggling is not permitted during operation. Parameter assignment is in the initialization phase only.

The clock sources (CLOCK0 to CLOCK7) are generated for all channels by a central clock divider. The base clock for all divider stages is the APB clock PCLK (125 MHz). The block diagram below shows how the system operates:

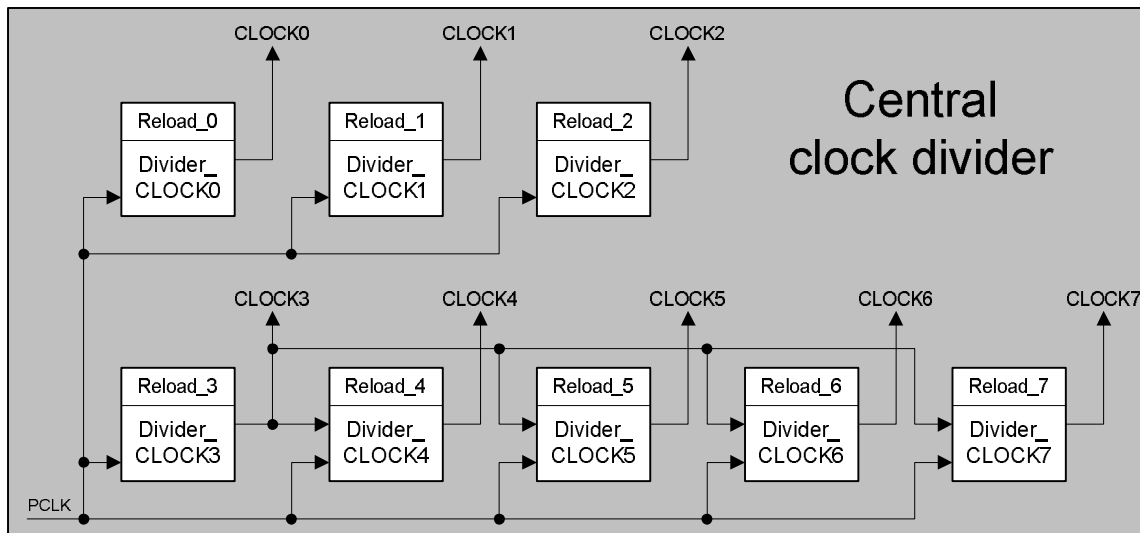


Figure 32: Block diagram of the central clock divider

The central clock divider consists of several divider stages, some of which are cascaded. The division factors are set for each divider stage with a dedicated 'FILT_Reload_x' register. Dynamic FILT_Reload_x toggling is not permitted during runtime.

The division factors of the individual stages of cascaded dividers must be multiplied to calculate the "total division factor". The table below shows the filter times achieved with sample division factors at 125 MHz:

Parameter register <i>FILT_Reload_x</i>	Sample division factor	Filter time (without 2-stage synchr.) t_{delay}	Minimum filter time (filter time – CLOCK jitter)
Reload_0	8	1.024 μs (CLOCK0)	0.960 μs
Reload_1	40	5.12 μs (CLOCK1)	4.80 μs

Reload_2	78	9.984 μ s (CLOCK2)	9.360 μ s
Reload_3	390	42.42 μ s (CLOCK3)	39.30 μ s
Reload_4	2	99.84 μ s (CLOCK4)	93.60 μ s
Reload_5	6	299.5 μ s (CLOCK5)	280.78 μ s
Reload_6	30	1.5 ms (CLOCK6)	1.406 ms
Reload_7	40	2.0 ms (CLOCK7)	1.875 ms

Table 18: Sample filter times for I-filter

The filter time is calculated by multiplying the division factor (cascaded from divider 3) by 16xTPCLK, as the filter system means that the output value can only change for every 16th clock. The filter times are as follows:

- min. **128 ns** e.g. when: *IN_Delay_x=0* & *FILT_Reload_0=0*
- max. **134 ms** e.g. when: *IN_Delay_x=7* & *FILT_Reload_3=0x3FF* & *FILT_Reload_7=0x3FF*

The maximum (total) throughput time through the filter module is if configured equal to the filter time plus 16 ns for two-stage synchronization. The minimum (total) throughput time (see Table 18) in this configuration depends on the CLOCK0 to CLOCK7 clock signal jitter.

2.3.10.1.1 Operating Principle of the RC Filter

The most significant bit in the 5-bit counter is the filtered output signal (*D_IN_Delay*).

If the input signal (*D_IN_SYN*) switches to "High", the counter starts to count up. If the counter status changes from '15' to '16', the counter value is automatically set to '31' ('0x1Fh'). When the counter reaches '31', the output signal (*D_IN_Delay*) is also set to "High". The counter value remains at '31' until the input signal (*D_IN_SYN*) changes from "High" to "Low".

If the input signal (*D_IN_SYN*) switches to "Low", the counter starts to count down. If the counter status changes from '15' to '16', the counter value is automatically set to '0' ('0x00h'). When the counter reaches '0', the output signal (*D_IN_Delay*) is also set to "Low". The counter value remains at '0' until the input signal (*D_IN_SYN*) changes from "Low" to "High".

Brief signal changes during the counting phase cause the counter to count either up or down in line with its signal state, i.e. the counter does not start from '0' following a signal change (that would be a signal delay, not filtering).

The filter structure therefore acts as an RC circuit or digital I-controller.

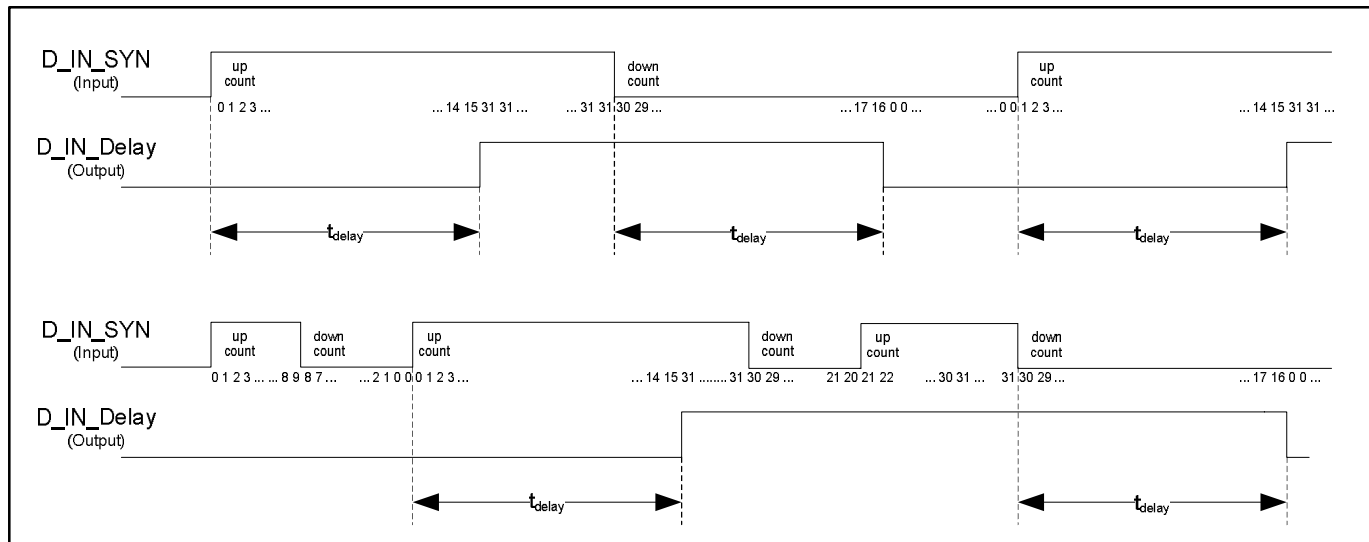


Figure 33: Signal in signal filtering

2.3.10.1.2 Address Mapping

Start_Address	End_Address	Modul/Memory_Name	Interface	Fill_Mode
0h	50h	i_filter	APB	none

Module	Register/Memory	Read	Write	Address	Revision
/i_filter					
	FILT_IP_VERSION	r		0h	
	FILT_IP_DEVELOPMENT	r		4h	
	FILT_ACCESS_ERR	(r)(h)	(w)	8h	
	FILT_RELOAD_0	r	w	Ch	
	FILT_RELOAD_1	r	w	10h	
	FILT_RELOAD_2	r	w	14h	
	FILT_RELOAD_3	r	w	18h	
	FILT_RELOAD_4	r	w	1Ch	
	FILT_RELOAD_5	r	w	20h	
	FILT_RELOAD_6	r	w	24h	
	FILT_RELOAD_7	r	w	28h	
	FILT_DELAY_0	r	w	2Ch	
	FILT_DELAY_1	r	w	30h	

	FILT_DELAY_2	r	w	34h	
	FILT_DELAY_3	r	w	38h	
	FILT_DELAY_4	r	w	3Ch	
	FILT_DELAY_5	r	w	40h	
	FILT_DELAY_6	r	w	44h	
	FILT_DELAY_7	r	w	48h	
	FILT_DELAY_8	r	w	4Ch	
	FILT_DELAY_9	r	w	50h	

2.3.10.1.3 Register Description

Module: /i_filter

Register:	FILT_IP_VERSION					Address:	0h	
Bits:	31dt0	Reset value:		10100h		Attributes:	r	
Description:								
Bit	Identifier	Reset	Attr.		Function / Description			
7dt0	DEBUG_VERSION	00h	r		Incremented for error correction (e.g. metal fix)			
15dt8	VERSION	01h	r		IP version: 0x01: Initial version			
31dt16	CONFIGURATION	0001h	r		IP configuration: 0x0000: 8 filter times, 64 input filters 0x0001: 8 filter times, 80 input filters			

Register:	FILT_IP_DEVELOPMENT				Address:	4h	
Bits:	31dt0	Reset value:	6800808h		Attributes:	r	
Description:		Project-specific design label					
Bit	Identifier	Reset	Attr.		Function / Description		
10dt0	BASELINE	008h	r		Number of RR label		
15dt11	INKREMENT	01h	r		HDL increment of the ClearCase label		
18dt16	PATCH	0h	r		For labeling metal fixes in the ASIC flow; only valid if platform = ASIC. Inkrement/Baseline is then invalid		
20dt19	PLATFORM	0h	r		00 = ASIC 01 = FPGA 10 = reserved 11 = user defined		
31dt21	IDENTIFIKATION	034h	r		Unique number per module		

Register:	FILT_ACCESS_ERR					Address:	8h	
Bits:	31dt0	Reset value:		none		Attributes:	(r)(h)	(w)
Description:								
Bit	Identifier	Reset	Attr.		Function / Description			
6dt0	APB_ADDRESS	00h	rh	w	Erroneous APB address in the I-filter			
27dt7	<reserved>							
29dt28	APB_SIZE	0h	rh	w	00: Byte access			

					01: Halfword access 10: Word access 11: reserved
30	APB_WRITE	0h	rh	w	0: Read access 1: Write access
31	ERR_LOCK	0h	rh	w	Set to 1 by the HW if erroneous APB access is detected. This blocks further HW entries. The SW (PNB stack) must reset the bit to 0 to enable new entries.

Register:	FILT_RELOAD_0			Address:	Ch
Bits:	9dt0	Reset value:	000h	Attributes:	r w
Description:	Load value of clock divider 0 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.				

Register:	FILT_RELOAD_1			Address:	10h
Bits:	9dt0	Reset value:	000h	Attributes:	r w
Description:	Load value of clock divider 1 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.				

Register:	FILT_RELOAD_2			Address:	14h
Bits:	9dt0	Reset value:	000h	Attributes:	r w
Description:	Load value of clock divider 2 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.				

Register:	FILT_RELOAD_3			Address:	18h
Bits:	9dt0	Reset value:	000h	Attributes:	r w
Description:	Load value of clock divider 3 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.				

Register:	FILT_RELOAD_4			Address:	1Ch
Bits:	9dt0	Reset value:	000h	Attributes:	r w
Description:	Load value of clock divider 4				

	The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.
--	---

Register:	FILT_RELOAD_5			Address:	20h	
Bits:	9dt0	Reset value:	000h	Attributes:	r	w
Description:	Load value of clock divider 5 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.					

Register:	FILT_RELOAD_6			Address:	24h	
Bits:	9dt0	Reset value:	000h	Attributes:	r	w
Description:	Load value of clock divider 6 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.					

Register:	FILT_RELOAD_7			Address:	28h	
Bits:	9dt0	Reset value:	000h	Attributes:	r	w
Description:	Load value of clock divider 7 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.					

Register:	FILT_DELAY_0			Address:	2Ch	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:						
Bit	Identifier	Reset	Attr.		Function / Description	
3dt0	IN_DELAY_0	Fh	r	w	Selection of clock source for input signal 0: 0x0: CLOCK0 0x1: CLOCK1 0x2: CLOCK2 0x3: CLOCK3 0x4: CLOCK4 0x5: CLOCK5 0x6: CLOCK6 0x7: CLOCK7 0x8 - 0xE: no filtering, only synchronization 0xF: no filtering and no synchronization, only forwarding	
7dt4	IN_DELAY_1	Fh	r	w	see IN_DELAY_0	
11dt8	IN_DELAY_2	Fh	r	w	see IN_DELAY_0	

15dt12	IN_DELAY_3	Fh	r	w	see IN_DELAY_0
19dt16	IN_DELAY_4	Fh	r	w	see IN_DELAY_0
23dt20	IN_DELAY_5	Fh	r	w	see IN_DELAY_0
27dt24	IN_DELAY_6	Fh	r	w	see IN_DELAY_0
31dt28	IN_DELAY_7	Fh	r	w	see IN_DELAY_0

Register:	FILT_DELAY_1					Address:	30h	
Bits:	31dt0	Reset value:		FFFFFFFFh		Attributes:	r	w
Description:								
Bit	Identifier	Reset	Attr.		Function / Description			
3dt0	IN_DELAY_8	Fh	r	w	see IN_DELAY_0			
7dt4	IN_DELAY_9	Fh	r	w	see IN_DELAY_0			
11dt8	IN_DELAY_10	Fh	r	w	see IN_DELAY_0			
15dt12	IN_DELAY_11	Fh	r	w	see IN_DELAY_0			
19dt16	IN_DELAY_12	Fh	r	w	see IN_DELAY_0			
23dt20	IN_DELAY_13	Fh	r	w	see IN_DELAY_0			
27dt24	IN_DELAY_14	Fh	r	w	see IN_DELAY_0			
31dt28	IN_DELAY_15	Fh	r	w	see IN_DELAY_0			

Register:		FILT_DELAY_2				Address:		34h	
Bits:		31dt0		Reset value:		FFFFFFFFh		Attributes: r w	
Description:									
Bit	Identifier		Reset	Attr.		Function / Description			
3dt0	IN_DELAY_16		Fh	r	w	see IN_DELAY_0			
7dt4	IN_DELAY_17		Fh	r	w	see IN_DELAY_0			
11dt8	IN_DELAY_18		Fh	r	w	see IN_DELAY_0			
15dt12	IN_DELAY_19		Fh	r	w	see IN_DELAY_0			
19dt16	IN_DELAY_20		Fh	r	w	see IN_DELAY_0			
23dt20	IN_DELAY_21		Fh	r	w	see IN_DELAY_0			
27dt24	IN_DELAY_22		Fh	r	w	see IN_DELAY_0			
31dt28	IN_DELAY_23		Fh	r	w	see IN_DELAY_0			

Register:	FILT_DELAY_3				Address:	38h	
-----------	--------------	--	--	--	----------	-----	--

Bits:	31dt0	Reset value:		FFFFFFFFh	Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	IN_DELAY_24	Fh	r	w	see IN_DELAY_0		
7dt4	IN_DELAY_25	Fh	r	w	see IN_DELAY_0		
11dt8	IN_DELAY_26	Fh	r	w	see IN_DELAY_0		
15dt12	IN_DELAY_27	Fh	r	w	see IN_DELAY_0		
19dt16	IN_DELAY_28	Fh	r	w	see IN_DELAY_0		
23dt20	IN_DELAY_29	Fh	r	w	see IN_DELAY_0		
27dt24	IN_DELAY_30	Fh	r	w	see IN_DELAY_0		
31dt28	IN_DELAY_31	Fh	r	w	see IN_DELAY_0		

Register:		FILT_DELAY_4				Address:		3Ch	
Bits:		31dt0		Reset value:		FFFFFFFFh		Attributes: r w	
Description:									
Bit	Identifier		Reset	Attr.		Function / Description			
3dt0	IN_DELAY_32		Fh	r	w	see IN_DELAY_0			
7dt4	IN_DELAY_33		Fh	r	w	see IN_DELAY_0			
11dt8	IN_DELAY_34		Fh	r	w	see IN_DELAY_0			
15dt12	IN_DELAY_35		Fh	r	w	see IN_DELAY_0			
19dt16	IN_DELAY_36		Fh	r	w	see IN_DELAY_0			
23dt20	IN_DELAY_37		Fh	r	w	see IN_DELAY_0			
27dt24	IN_DELAY_38		Fh	r	w	see IN_DELAY_0			
31dt28	IN_DELAY_39		Fh	r	w	see IN_DELAY_0			

Register:	FILT_DELAY_5					Address:	40h		
Bits:	31dt0		Reset value:		FFFFFFFFh		Attributes:	r	w
Description:									
Bit	Identifier		Reset	Attr.		Function / Description			
3dt0	IN_DELAY_40		Fh	r	w	see IN_DELAY_0			
7dt4	IN_DELAY_41		Fh	r	w	see IN_DELAY_0			
11dt8	IN_DELAY_42		Fh	r	w	see IN_DELAY_0			
15dt12	IN_DELAY_43		Fh	r	w	see IN_DELAY_0			

19dt16	IN_DELAY_44	Fh	r	w	see IN_DELAY_0
23dt20	IN_DELAY_45	Fh	r	w	see IN_DELAY_0
27dt24	IN_DELAY_46	Fh	r	w	see IN_DELAY_0
31dt28	IN_DELAY_47	Fh	r	w	see IN_DELAY_0

Register:		FILT_DELAY_6				Address:		44h	
Bits:		31dt0	Reset value:		FFFFFFFFh		Attributes:	r	w
Description:									
Bit	Identifier		Reset	Attr.		Function / Description			
3dt0	IN_DELAY_48		Fh	r	w	see IN_DELAY_0			
7dt4	IN_DELAY_49		Fh	r	w	see IN_DELAY_0			
11dt8	IN_DELAY_50		Fh	r	w	see IN_DELAY_0			
15dt12	IN_DELAY_51		Fh	r	w	see IN_DELAY_0			
19dt16	IN_DELAY_52		Fh	r	w	see IN_DELAY_0			
23dt20	IN_DELAY_53		Fh	r	w	see IN_DELAY_0			
27dt24	IN_DELAY_54		Fh	r	w	see IN_DELAY_0			
31dt28	IN_DELAY_55		Fh	r	w	see IN_DELAY_0			

Register:	FILT_DELAY_7					Address:	48h	
Bits:	31dt0		Reset value:		FFFFFFFFh	Attributes:	r	w
Description:								
Bit	Identifier		Reset	Attr.		Function / Description		
3dt0	IN_DELAY_56		Fh	r	w	see IN_DELAY_0		
7dt4	IN_DELAY_57		Fh	r	w	see IN_DELAY_0		
11dt8	IN_DELAY_58		Fh	r	w	see IN_DELAY_0		
15dt12	IN_DELAY_59		Fh	r	w	see IN_DELAY_0		
19dt16	IN_DELAY_60		Fh	r	w	see IN_DELAY_0		
23dt20	IN_DELAY_61		Fh	r	w	see IN_DELAY_0		
27dt24	IN_DELAY_62		Fh	r	w	see IN_DELAY_0		
31dt28	IN_DELAY_63		Fh	r	w	see IN_DELAY_0		

Register:	FILT_DELAY_8				Address:	4Ch	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	r	w

Description:					
Bit	Identifier	Reset	Attr.		Function / Description
3dt0	IN_DELAY_64	Fh	r	w	see IN_DELAY_0
7dt4	IN_DELAY_65	Fh	r	w	see IN_DELAY_0
11dt8	IN_DELAY_66	Fh	r	w	see IN_DELAY_0
15dt12	IN_DELAY_67	Fh	r	w	see IN_DELAY_0
19dt16	IN_DELAY_68	Fh	r	w	see IN_DELAY_0
23dt20	IN_DELAY_69	Fh	r	w	see IN_DELAY_0
27dt24	IN_DELAY_70	Fh	r	w	see IN_DELAY_0
31dt28	IN_DELAY_71	Fh	r	w	see IN_DELAY_0

Register:	FILT_DELAY_9				Address:	50h	
Bits:	31dt0	Reset value:		FFFFFFFFh	Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	IN_DELAY_72	Fh	r	w	see IN_DELAY_0		
7dt4	IN_DELAY_73	Fh	r	w	see IN_DELAY_0		
11dt8	IN_DELAY_74	Fh	r	w	see IN_DELAY_0		
15dt12	IN_DELAY_75	Fh	r	w	see IN_DELAY_0		
19dt16	IN_DELAY_76	Fh	r	w	see IN_DELAY_0		
23dt20	IN_DELAY_77	Fh	r	w	see IN_DELAY_0		
27dt24	IN_DELAY_78	Fh	r	w	see IN_DELAY_0		
31dt28	IN_DELAY_79	Fh	r	w	see IN_DELAY_0		

2.3.10.2

ARM926 Watchdog

2.3.10.2.1 Overview

The ARM926 watchdog primarily consists of two counters with different widths (32 and 36 bit) that count down from a parameterized initial value.

The watchdog has the following principal characteristics:

- The clock pulse supply is 125 MHz (APB clock)
- One output signal of the Counter0 watchdog (XWD_OUT0) is output at an alternate Function of a GPIO pin.
- After the watchdog Counter0 expires an interrupt (WD_INT: WD_INT_ARM926) is generated (IRQ27, see chapter 2.3.2.14).
- After the watchdog Counter1 expires (XWD_OUT1) the 'XRES_ARM926_WD' is generated (see chapter 2.3.9.4.4).

2.3.10.2.2 Block diagram

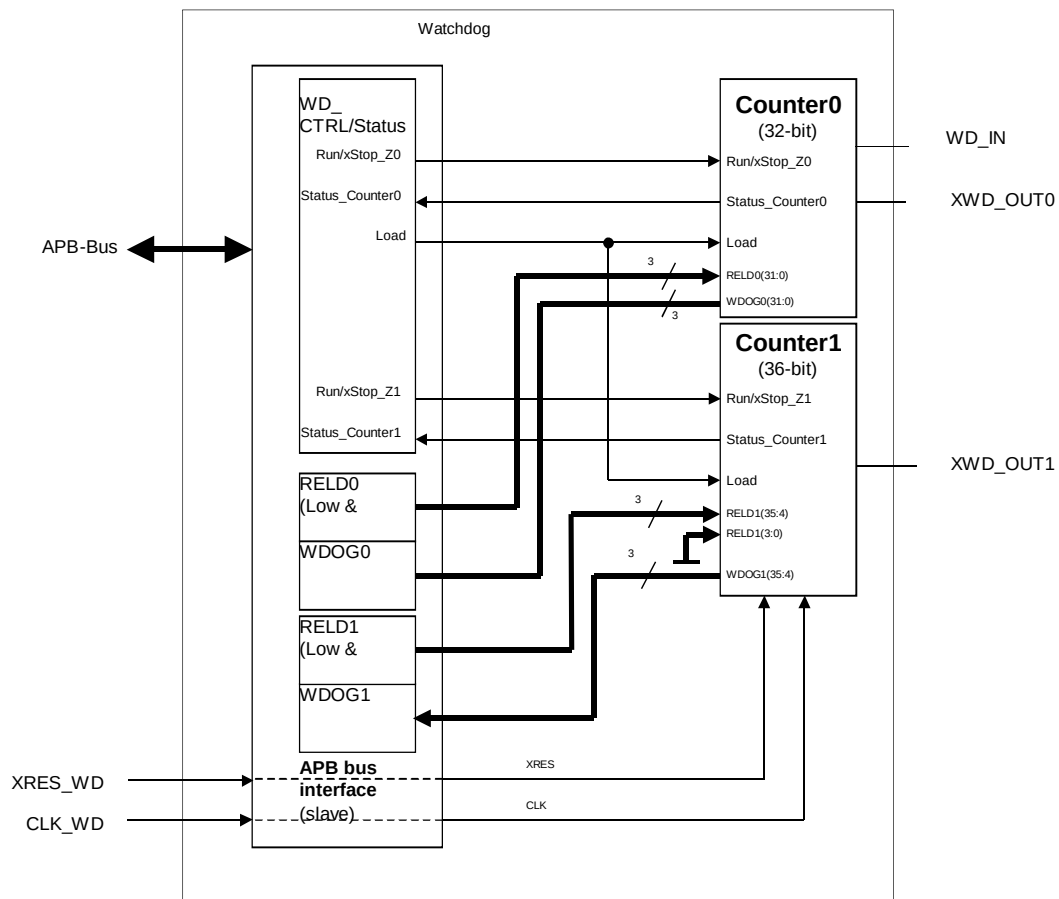


Figure 34: Block diagram ARM926 Watchdog

Counter0:

Counter0 is a 32-bit wide counter output that counts to 0 starting with the value passed in the RELD0(_LOW/_HIGH) register with the clock pulse from the CLK_WD pin. The watchdog is (re)started with Run/xStop_Z0=1 and, when required, stopped with Run/xStop_Z0=0.

The output is `XWD_OUT0=0` when the watchdog is stopped. If the watchdog has been started and `Counter0  $\neq$  0`, then the output is `XWD_OUT0=1`, otherwise `XWD_OUT0=0`. Denote that in the case that `Counter0` is preloaded with 0 and is started, the output `XWD_OUT0` is also 1 for one clock cycle. Probably this is not a use case, this is stated only to get a whole picture of the circuit.

If the input is `Load=1`, the watchdog will be triggered, i.e. the `Counter0` will be loaded with the value contained in the `RELD0(_LOW/_HIGH)` register and will continue counting from this value.

The `WDOG0` register can be used to fetch the current value of the counter.

The `Status_Counter0` output is active only when `Run/xStop_Z0=1` and `Counter0` has expired.

An expired `Counter0`, i.e. `Counter0` decremented to 0, initiates an interrupt (`WD_INT`). For a Timing Diagram see also figure “`XWD_OUT0/ WD_INT` signal sequence”

Counter1:

`Counter1` is a 36-bit wide counter that counts to 0 starting with the value passed in the `RELD1(_LOW/_HIGH)` register with the clock pulse from `CLK_WD` pin. The `RELD1(_LOW/_HIGH)` register contains only the high order 32 bits of the counter. The four low order bits of the counter are always loaded with the value 0x0.

The watchdog is (re)started with `Run/xStop_Z1=1` and, when required, stopped with `Run/xStop_Z1=0`. If `Counter1` is stopped or it is started but has not yet attained the value 0, then the output is `XWD_OUT1=1`. The output is `XWD_OUT1=0` when `Counter1` has expired.

If the input is `Load=1`, the watchdog will be triggered, i.e. the upper 32 bits of `Counter1` will be loaded with the value contained in the `RELD1(_LOW/_HIGH)` register and the watchdog will continue counting from this value to zero.

The `WDOG1` register can be used to fetch the current value of the counter (only the upper 32 bits).

The `Status_Counter1` output is active only when `Run/xStop_Z1=1` and `Counter1` has expired.

CTRL/Status register (X=0,1):

The register is 32-bit wide and contains data only in the lower 16 bits. The upper 16 bits contain a special signature (see also 'write protection').

- `Run/xStop_ZX`-Bits: Starts and stops the `Counter0`.
- `Load`: The `Load` signal acts simultaneously on both counters (provided they have been enabled). The `Load` signal from the register is synchronized for the counter side. An increasing edge at the load input of the counter loads the `CounterX` with the value from `RELD0(_LOW/_HIGH)` or `RELD1(_LOW/_HIGH)`.
The software does not need to reset the load bit.
- `Status_counterX`: This bit is a single status bit present separately for each counter. If the bit is set, then the associated counter has attained the value 0. If `Run/xStop_ZX=0`, then the associated status bit always reads logical 0.

RELD0_LOW & RELD0_HIGH:

These two 32-bit registers contain user data only in the lower 16 bits. Each of the upper 16 bits is reserved for a special signature (see also 'write protection'). Thus, the 32-bit value for the counter consists of the lower halves of both registers. The register values must be changed only when the counter is stopped (`Run/xStop_ZX = 0`).

RELD1_LOW & RELD1_HIGH :

These two 32-bit registers contain user data only in the lower 16 bits. Each of the upper 16 bits is reserved for a special signature (see also 'write protection'). Thus, the 32-bit value for the counter consists of the lower halves of both registers. Only the upper 32 bits of the reload value can be specified for these counters – the lower 4 bits are always logical 0. The register values must be changed only when the counter is stopped (Run/xStop_ZX = 0).

WDOG0 & WDOG1:

These two registers can be used to read the current values of the two counters. Only the upper 32 bits of the current counter value can be read for Counter1. The content of the two registers will be updated after each increasing edge of the counter cycle clock (CLK_WD). Thus, without requiring the wait state, a read access to these registers always returns the contents of the two counters after the last recognized counter cycle

2.3.10.2.3 Signal waveforms

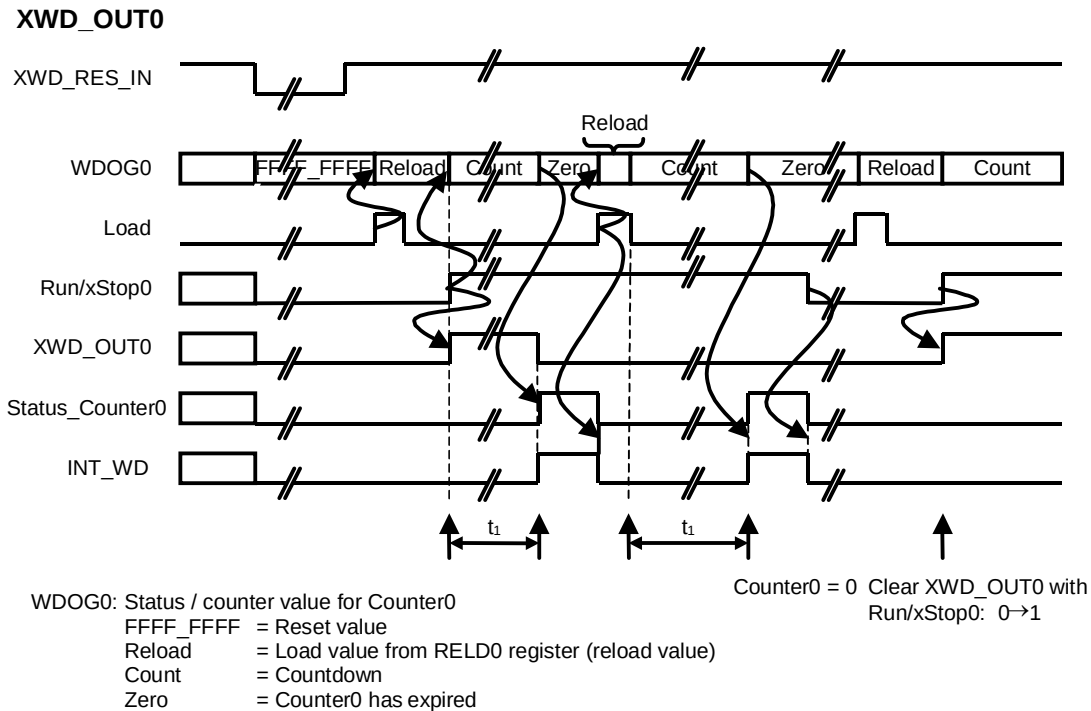


Figure 35: XWD_OUT0/ WD_INT signal sequence

After a reset, the XWD_OUT0 output is initially logical 0. The XWD_OUT0 output is inactive (=1) only after Counter0 has been started (Run/xStop_Z0=1) and provided the counter value $\neq 0$.

If after the start or the last retrigger pulse Counter0 expires after time t_1 , then

- the XWD_OUT0 output becomes active (=0),
- the "Status_Counter0" status bit is set and
- the INT_WD interrupt signal becomes active (the increasing edge for INT_WD initiates the interrupt).

A retrigger pulse (Load=1 & Reload-value $\neq 0$) or stopping the counter (Run/xStop_Z0=0) causes the status bit and the interrupt signal to be reset.

The XWD_OUT0 output assumes the value logical 1 again only when Counter0 is subsequently restarted with a value $\neq 0$.

$$t_1 = (RELD0 + 1) \times T_{REF_CLK_IN}$$

t_1 : Time until the counter expires
 RELD0: Decimal value of the reload value for Counter0
 $T_{REF_CLK_IN}$: Period duration of the system cycle clock (8 ns).
 $T_{REF_CLK_IN} = 125 \text{ MHz}$:
 $t_{1MIN} = 0 \text{ sec}$, $t_{1MAX} = 34,36 \text{ sec}$, Interval = 8 ns

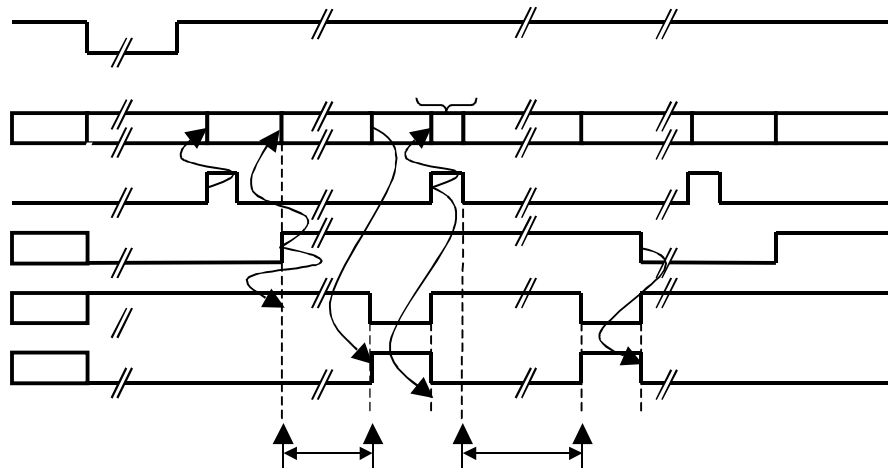


Figure 36: XWD_OUT1 signal sequence

After a reset, the XWD_OUT1 output is initially logical 1. The XWD_OUT1 output becomes logical 0 and the "Status_Counter1" status bit is set only after Counter1 has been started (Run/xStop_Z1=1) and the time t_2 has expired after the start or the last retrigger pulse (Load=1).

A retrigger pulse (Load=1 & Reload-value $\neq 0$) or stopping the counter (Run/xStop_Z1=0) causes the status bit and the XWD_OUT1 output to be reset again.

$$t_2 = (RELD1 \times 16 + 1) \times T_{REF_CLK_IN}$$

t_2 : Time until the counter expires
 $RELD1$: Decimal value of the reload value for Counter1
 $T_{REF_CLK_IN}$: Period duration of the system cycle clock (8 ns)
 $T_{REF_CLK_IN} = 125 \text{ MHz}$:
 $t_{2MIN} = 0 \text{ sec}$, $t_{2MAX} = 549.76 \text{ sec}$, Interval = 128 ns

2.3.10.2.4 Write protection of the watchdog register

If the Watchdog Control/Status register or one of the Watchdog Reload registers is to be written, a defined bit combination in the upper 16 bits (key bits) must be written simultaneously. The key bits are 9876h (arbitrary). The read access returns the value 0000h in the upper 16 bits.

2.3.10.2.5 Starting the Watchdog

The following sequence describes the initialization phase which at first has to be performed by software:

- Write a proper value for RELD0/1 (_LOW/_HIGH)
- Load the values by writing to the Load register
- Activate the watchdog by writing 1 to the Run/xStop_Z0/1 registers

2.3.10.2.6 Notes

- The ERTEC 200P watchdog can't be used by architectures with an external PCI host (PCI bridge outside the ERTEC 200P), because XWD_OUT0=0 (counter0 expired) can't be signaled to the external host over the PCI bridge.
In this use case the watchdog function must be realized by the external host. The ERTEC 200P must increment a counter in the external PCI host memory and the host must check this counter to see whether the ERTEC 200P is still running.
- By coupling an external host over the XHIF interface directly, the watchdog in the ERTEC 200P can be used. The XWD_OUT0=0 (counter0 expired) can be signaled to the external host over a GPIO port.
- For embedded single processor systems with ERTEC 200P the watchdog function is mandatory.

2.3.10.2.7 Address Mapping

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	18h	wdog	

Module	Register/Memory	Read	Write	Address
wdog	WD_CTRL_STATUS	(r)(h)	(w)(k)	0h
	RELD0_LOW	(r)	w(k)	4h
	RELD0_HIGH	(r)	w(k)	8h
	RELD1_LOW	(r)	w(k)	Ch
	RELD1_HIGH	(r)	w(k)	10h
	WDOG0	rh		14h
	WDOG1	rh		18h

2.3.10.2.8 Register Description

The data-bus-width of all specified registers in this module is: 32 .

A '0h ' is read from Software for each not specified Bit in the registers.

The addressing is organized: WORD - wise

Register:	WD_CTRL_STATUS				Address:	0h	
Bits:	31dt0	Reset-Value:	00000000h		Attribute:	(r)(h)	(w)(k)
Description:		Control/Status register. Configuration and control bits for the watchdog.					
Bit	Identifier	Reset	Attr.		Function / Description		
31dt16	Key_bits	0000h		wk	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-4 of this register will be written, otherwise the operation has no effect.		
4	Status_Counter1	0h		rh	Watchdog status counter 1 (write is ignored): 0: Watchdog counter 1 has not expired 1: Watchdog counter 1 has expired Note: This bit then can only be read as '1' when RUN/xStop_Z1 is active (1).		
3	Status_Counter0	0h		rh	Watchdog status counter 0 (write is ignored): 0: Watchdog counter 0 has not expired 1: Watchdog counter 0 has expired Note: This bit then can only be read as '1' when RUN/xStop_Z0 is active (1).		
2	Load_Trigger	0h	r	w	Watchdog trigger (load watchdog counters 0 and 1 with the value of the Reload registers): 0: Do not trigger watchdog 1: Trigger watchdog Although this bit can be read back, it acts only during a write. To trigger the watchdog counters, it suffices to write a 1 to this bit; no 0/1 edge is required. The trigger signal acts on both watchdog counters.		
1	Run_xStop_Z1	0h	r	w	Enable/disable watchdog counter 1: 0: Watchdog counter 1 disabled 1: Watchdog counter 1 enabled Note: If this bit = 0, the XWDOUT1 output of the ERTEC 200P is passive (1) and the status bit of counter 1 (bit 4) is 0.		
0	Run_xStop_Z0	0h	r	w	Enable/disable watchdog counter 0: 0: Watchdog counter 0 disabled		

					1: Watchdog counter 0 enabled Note: If this bit = 0, the XWDOUT0 output of the ERTEC 200P is active (0), the interrupt of the watchdog (WDINT) is 0 and the status bit of counter 0 (bit 3) is 0.
--	--	--	--	--	--

Register:	RELD0_LOW				Address:	4h	
Bits:	31dt0	Reset-Value:	0000FFFFh		Attribute:	(r)	w(k)
Description:		Reload Register 0_low. Reload value for bits 15:0 of watchdog counter 0.					
Bit	Bezeichner	Reset	Attr.		Function / Description		
31dt16	Key_bits	0000h		wk	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-15 of this register will be written, otherwise the operation has no effect.		
15dt0	Reload0	FFFFh	r	w	Reload value for bits 15:0 of watchdog counter 0		

Register:	RELD0_HIGH				Address:	8h	
Bits:	31dt0	Reset-Value:	0000FFFFh		Attribute:	(r)	w(k)
Description:		Reload Register 0_high. Reload value for bits 31:16 of watchdog counter 0.					
Bit	Bezeichner	Reset	Attr.		Function / Description		
31dt16	Key_bits	0000h		wk	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-15 of this register will be written, otherwise the operation has no effect.		
15dt0	Reload0	FFFFh	r	w	Reload value for bits 31-16 of watchdog counter 0		

Register:		RELD1_LOW				Address:		Ch		
Bits:		31dt0	Reset-Value:		0000FFFFh		Attribute:		(r)	w(k)
Description:			Reload Register 1_low. Reload value for bits 19:4 of watchdog counter 1.							
Bit	Bezeichner		Reset	Attr.		Function / Description				
31dt16	Key_bits		0000h		wk	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-15 of this register will be written, otherwise the operation has no effect.				

15dt0	Reload1	FFFFh	r	w	Reload value for bits 19:4 of watchdog counter 1.
-------	---------	-------	---	---	---

Register:	RELD1_HIGH				Address:	10h	
Bits:	31dt0	Reset-Value:	0000FFFFh		Attribute:	(r)	w(k)
Description:		Reload Register 1_high. Reload value for bits 35:20 of watchdog counter 1.					
Bit	Bezeichner	Reset	Attr.		Function / Description		
31dt16	Key_bits	0000h		wk	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-15 of this register will be written, otherwise the operation has no effect.		
15dt0	Reload1	FFFFh	r	w	Reload value for bits 35:20 of watchdog counter 1		

Register:	WDOG0				Address:	14h	
Bits:	31dt0	Reset-Value:	FFFFFFFFh		Attribute:	rh	
Description:	Watchdog value 0. Value of watchdog counter 0.						

Register:	WDOG1				Address:	18h	
Bits:	31dt0	Reset-Value:	FFFFFFFFh	Attribute:	rh		
Description:	Watchdog value 1. Value of watchdog counter 1.						

2.3.10.3

*Timer 0 – 5

2.3.10.3.1 Overview

In ERTEC 200P there are six independent timers integrated which serve for monitoring various software routines. Each of these software timers is assigned an own interrupt. Access to the timer registers is carried out at word limits (32 bits).

Summary of the timer functions

- Bit down-counter
- Programmable bit prescaler connectable (separately for each timer)
- Loadable/reloadable
- Start, stop and continue function
- Interrupt when counter value '0' has been reached
- Read register for reading out the counter value
- Three event registers for event-driven storing of the counter value:
 - Two external HW events (selectable from the EXTERNAL_INPUTS signals of the TIMER_TOP module) for storing in two HW event registers
 - One internal SW event for storing in one SW event register
- Timers are cascadable (prerequisite: timer output of the low-order counter(s) is fed back to the TIMER_TOP module via the EXTERNAL_INPUTS signals and is selected as GATE_TRIG signal at the high-order timer)
- Clocking of the counter can be done with the input clock CLK_TIMT = APB_Takt and via the SW (internal gate/trigger signal) as well as via an input signal (external gate/trigger signal)
- Triggering of the counter (loading) can be done via the SW (internal gate/trigger signal) as well as via an input signal (external gate/trigger signal)
- Clock output for supplying further modules (bit divider for CLK_TIMT = APB_Clock).

In the zero crossing, an output pulse (TIMER_OUT) is generated which can be evaluated as interrupt. The TIMER_TOP module contains six TIMER and multiplexer submodules each for selecting the trigger signals for the individual timers, and functions expanding over all timers (e.g. synchronous releasing of all timers, address coding). The TIMER submodule contains the basic timer function (e.g. counter, prescaler, load register for counter). The distribution of the timer functions to the TIMER and TIMER_TOP modules is supposed to simplify the reusability in different ASICs.

The TIMER and TIMER_TOP modules are described in detail in the following sections. Connections of the TIMER_TOP module to other modules or to ASIC pins are to be found in the functional description of the chip/core level.

2.3.10.3.2 TIMER_TOP functionality

Block diagram:

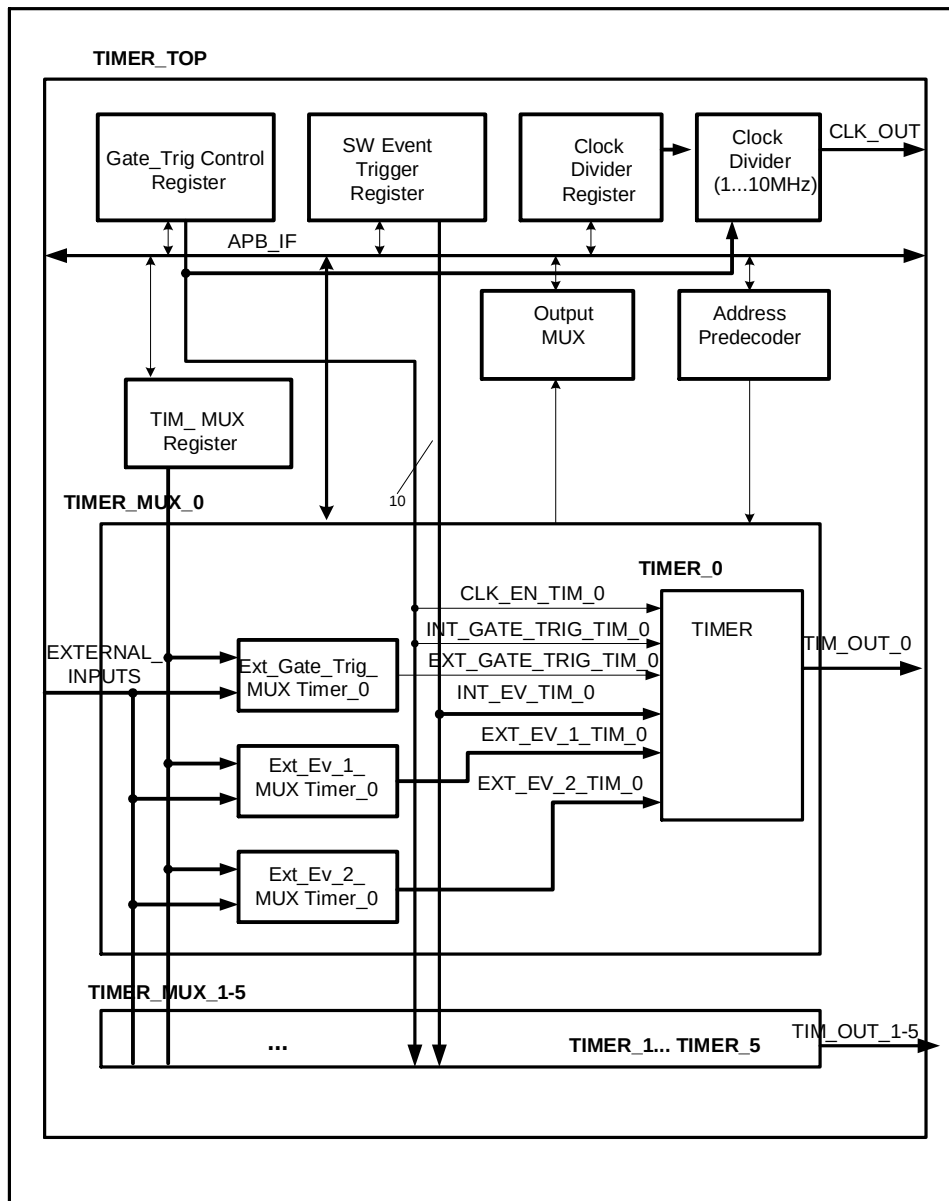


Figure 6: TIMER_TOP block diagram

The TIMER_TOP submodule contains the wiring of the individual TIMER modules, the Gate_Trig_Control registers, the multiplexer and timer MUX registers for the input signals INT_GATE_TRIG_TIM, EXT_GATE_TRIG_TIM, EVENT1, EVENT2, and the clock divider.

The TIMER_TOP module consists of the following functional units:

- Gate_Trig_Control register for triggering SW gate/trigger signals and for releasing the count pulses for the timers
- SW event trigger register for storing the current counter values in the Int_Event registers

- Six timers
- Multiplexers and the corresponding registers for selecting the sources for the inputs of the TIMER submodules
- Clock divider and the corresponding register for activating the clock divider.

The TIMER_TOP module has an APB interface (AMBA 2.0). The access width has to be 32 bits. Different accesses lead to a faulty writing of the register of the TIMER_TOP module.

Gate Count Register:

The TIMER_TOP module has a Gate_Trig_Control register in which a SW gate signal can be set/reset for each TIMER module (= INT_GATE_TRIG_TIM inputs of the TIMERS). The effect depends on the selected mode.

Moreover, the count clock can be released/blocked for each TIMER module (= CLK_EN inputs of the TIMERS). By writing on this register, all TIMER modules can be started / stopped synchronously.

Attention: If the SW gate signal is changed while simultaneously the count clock operating mode is released/blocked, the following behavior applies to all operating modes in which the SW gate signal is relevant:

- When blocking the counting pulse:
SW gate signal change has no effect anymore.
- When releasing the counting pulse:
SW gate signal change is effective in the next count/load clock, depending on the selected operating mode. This also applies to operating modes in which the edge of the SW gate signal is relevant.

Software event trigger register:

Writing a '1' into the bit "n" of the SW event trigger register triggers a positive edge at the INT_EV input of the TIMER module "n" and leads to a storing of the counter value of the TIMER module "n" in the Int_Event register of the TIMER module "n".

The '1' is not stored (read=0).

By writing on this register, the current counter values of all timers can be synchronously stored in the SW event registers of the TIMER modules.

Multiplexer/TIM_MUX register:

The TIMER_TOP submodule has own multiplexers and registers for their control for each timer, which serve for selecting the sources for the inputs of the TIMER modules EXT_GATE_TRIG_TIM, EVENT1, EVENT2. The EXTERNAL_INPUTS (15:0) inputs of the TIMER_TOP module are available as sources.

The registers for the multiplexers are designed in such a way that the numbers of the select signals entered there select the corresponding number of the input signal.

Each TIMER module can be assigned own sources for the inputs EXT_GATE_TRIG, EVENT1, EVENT2.

Thus, versatile measuring tasks can be configured (for examples, see Section "Sequences").

For the assignment of the input signals of the individual multiplexers, see "Chip/core level", Subsection "Timer".

clock divider / clock divider register:

The TIMER_TOP module has a clock divider for providing a clock for external modules, which is symmetrical with the input clock CLK_TIMT. The clock divider is 8 bits wide (NUM_OF_CLKDIV_BITS). The clock divider value is set by writing on the bits in the CLOCK_DIVIDER_REG register. A writing of the register CLOCK_DIVIDER_REG loads the clock divider with the clock divider value. The current value of the clock divider is nonreadable.

The clock divider output CLK_OUT is '0' as long as the clock dividing has not been activated.

The clock dividing is activated by setting the CLK_DIV_EN bit to '1' and by writing the clock divider value CLOCK_DIVIDER_VALUE with a value unequal to 0 (in the register CLOCK_DIVIDER_REG).

When clock dividing is active, the following applies:

- The clock divider output CLK_OUT is '0' as long as the clock divider has a value smaller than $\text{CLKDIV_VALUE}/2$, otherwise '1'.
- The divided clock CLK_OUT is symmetrical if the clock divider value is odd-numbered.
- If the current value of the clock divider is '0', the clock divider is loaded with the value of the bits CLOCK_DIVIDER_VALUE.

When clock dividing is enabled, the clock divider value CLOCK_DIVIDER_VALUE must not be changed. Before changing the clock divider value CLOCK_DIVIDER_VALUE the clock dividing must be disabled by setting Clock Divider Enable CLK_DIV_EN to 0.

Attention: Writing a new clock divider value CLOCK_DIVIDER_VALUE may have the effect that CLK_OUT runs with a wrong clock period once.

The clock divider is implemented as a down-counter which counts down cyclically from the clock divider value CLKDIV_VALUE.

The clock divider divides the frequency of input clock CLK_TIMT by the factor $1 / (\text{CLKDIV_VALUE} + 1)$. Exception: A clock divider value of 0 supplies CLK_OUT=0.

Possible values for the clock divider value CLKDIV_VALUE: '0' to $2^8 - 1$. At 125 MHz there are therefore output frequencies of $125 \text{ MHz} / 2^8$ - $125 \text{ MHz} / 2$ possible.

Timer cascading:

The TIMER modules are cascadable under certain conditions. The items stated under "Cascading of TIMER modules" have to be taken into account for that.

2.3.10.3.3 Timer module

Timer block diagram:

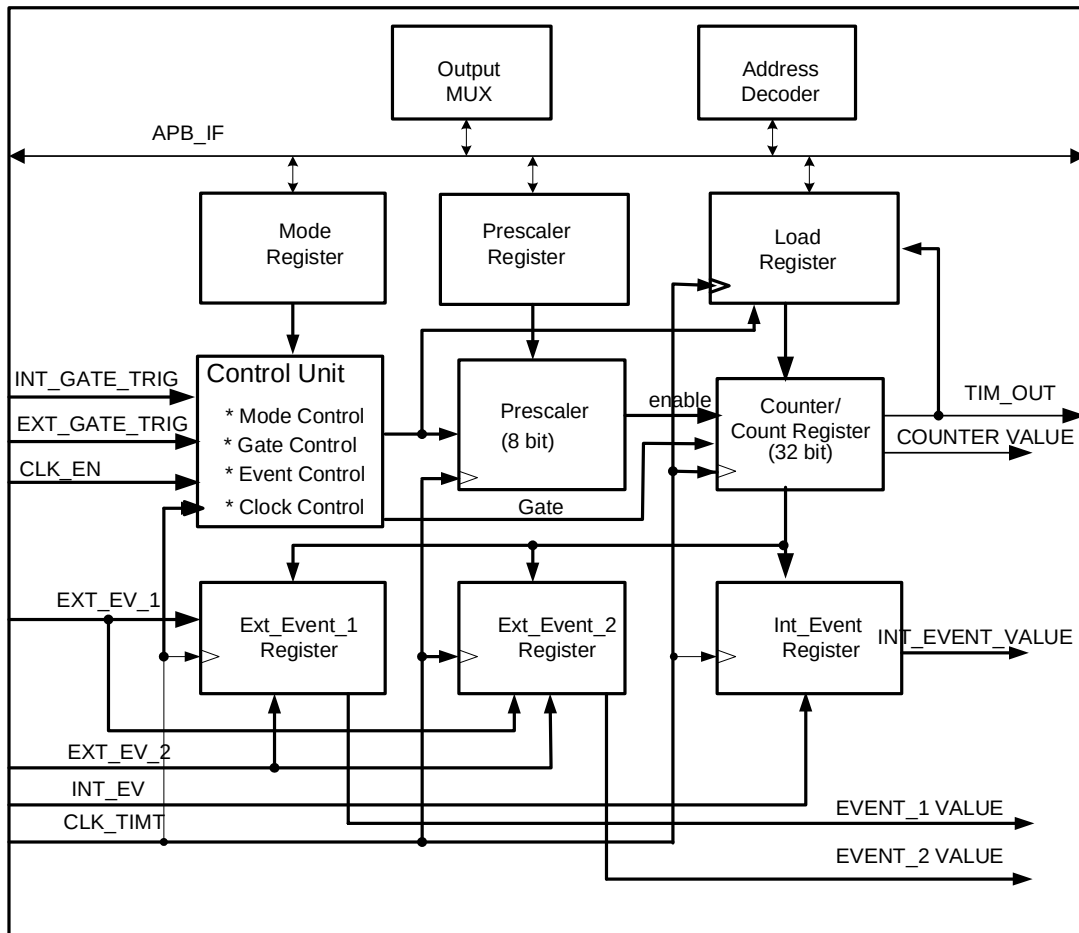


Figure 37: TIMER block diagram

The submodules `TIMER_TOP` and `TIMER` operate with the operating clock `CLK_TIMT` or `CLK`. `CLK_TIMT` is wired to `CLK`. The operating clock is 125MHz and the circuit is designed synchronously with that clock. The reset input of the submodules `TIMER_TOP` and `TIMER` is used as asynchronous reset; a synchronization of the reset input has not been implemented.

Each `TIMER` submodule consists of the following functional units:

- Counter/count register (counter and register for reading the current counter value)
- Prescaler/prescaler register (prescaler and register for prescaler load value)
- Load register (load value of the counter)
- Mode register (setting the operating mode)
- Event register for storing the counter values dependent on events (register `Ext_Event_1`, `Ext_Event_2`, `Int_Event`)

Counter / counter register:

Each `TIMER` module contains a 32-bit counter which counts down from a start value. The counter stops when the value '0' has been reached or is automatically reloaded if the reload function has been activated in the mode register.

The following sources can be selected as count/load clock of the counter:

- Input clock CLK_TIMT (125MHz = APB clock = setting after reset)
- Input clock CLK_TIMT divided by an 8-bit prescaler
- External gate/trigger signal (input signal INT_GATE_TRIG_TIM or EXT_GATE_TRIG_TIM, synchronized with input clock CLK_TIMT (two flip flops)).

The software can read the current value of the counter in the count register.

The counter range is up to 26 days (Input Clk = 125MHz, 8bit Clock Divider, 8bit Prescaler, 32bit Timer).

Prescaler / prescaler register:

In accordance with the presetting, the counter is operated with CLK_TIMT (125MHz = APB clock). Nevertheless, each counter can be connected with an 8-bit prescaler (independent parameterization for each timer) so that the runtime of the individual TIMER modules can be increased accordingly.

Each TIMER module contains an 8-bit prescaler which counts down cyclically from a start value, and a prescale register which contains the start value for the prescaler (= PRESCALER_VAL).

The prescaler counts / loads with the internal clock CLK_TIMT.

The prescaler is deactivated if

- timer input CLK_EN=0
- Gate_effect=0 and external gate/trigger signal has passive level

The current counter value of the prescaler is nonreadable. There is no signal for the prescaler indicating the counter value '0'. The prescaler register is 8 bits wide (NUM_OF_PRESCALER_BITS). The prescaler divides the counter frequency by the factor $1 / (\text{PRESCALER_VALUE} + 1)$.

Possible values for the prescaler register: '0' to $2^8 - 1$

Load Register:

The load register contains the load value (start value) of the counter. Writing on the load register with a new value causes the takeover of the load value into the counter if the following conditions are met:

- Timer input CLK_EN = '1' during writing
- Bit DIS_RLD_WHEN_WR_LDREG in the mode register = '0' during writing

The time of takeover of the load value into the counter also depends on the operating mode of the timer and is carried out with the count/load clock of the counter.

Mode Register:

The operating mode of the TIMER module is set via the following bits in the mode register:

- Init_bit
- Clk_input select
- Reload_disable
- DIS_RLD_WHEN_WR_LDREG

- Ext_gate_trig_enable
- Gate_polarity
- Gate_effect
- Timer_out_polarity
- Event1/2_control
- Event1/2_Inversion

The **Init_bit** is an initialization bit which resets the TIMER module, loads (restarts) the counter with the value from the load register and loads the prescaler with the value from the prescaler register. It deletes all values in the event registers and resets the edge evaluation of the event inputs and the EXT_GATE_TRIG signal. After having been set by the software, this bit is automatically deleted again, a reading of the bit by the software always results in the value '0'. The definition is carried out in accordance with the following table:

Init_bit	Function
0	Bit Init_bit not active (no initialization)
1	Bit Init_bit active (initialization)

The counter **counts/loads** with the count/load clock. The selection is done by the mode register bit Clk_input_select and the value of the prescaler register in accordance with the following table:

Clk_input_select	Function
0	Count/load clock = rising edge of CLK_TIMT; The counting/loading of the counter with the CLK_TIMT edge only takes place if the prescaler value is '0'
1	Count/load clock = external gate/trigger signal (edge evaluation with CLK_TIMT is valued as count/load enable signal)

The **Reload_disable bit** determines if the counter stops the counting process when it has reached the value '0' (single mode) or if the counter is automatically loaded = newly started with the reload value from the load register when the value '0' has been reached (reload mode). The definition is carried out in accordance with the following table:

Reload_disable	Function
0	Reload mode active
1	Single mode active

The **DIS_RLD_WHEN_WR_LDREG** bit in the mode register determines the effect of the writing of a new value on the load register:

DIS_RLD_WHEN_WR_LDREG	Function
0	Writing a load value on the load register has the effect that the load value is taken over into the counter with the next counter edge or the next gate/trigger signal (depending on the setting in the mode register).
1	Writing a load value on the load register has the effect that the load value is taken over into the counter under the following conditions • The counter has reached the value '0' • Bit Reload_disable in the mode register is '1' If these conditions are met, the takeover is carried out with the next counter edge or the next gate/trigger signal (depending on the setting in the mode register).

By setting the DIS_RLD_WHEN_WR_LDREG bit to 1, the takeover of the load register value is only possible during the zero crossing. This function is required to use the timers as phase-shifted cyclic timers.

One of the two input signals **EXT_GATE_TRIG** or **INT_GATE_TRIG** can be used:

- as gate signal for releasing/blocking the counter, or
- as trigger signal for triggering the counter, or
- as clock signal for clocking the counter.

The selection of the external gate/trigger signal is done by the mode register bit Ext_gate_trig_enable in accordance with the following table:

Ext_gate_trig_enable	Function
0	INT_GATE_TRIG = internal gate/trigger signal
1	EXT_GATE_TRIG = external gate/trigger signal

Differentiation between INT_GATE_TRIG and EXT_GATE_TRIG (internal and external gate/trigger signal):

- INT_GATE_TRIG is triggered by the software
- EXT_GATE_TRIG is triggered by an external hardware signal.

The **Gate_polarity bit** defines the active level or the active edge of the external gate/trigger signal in accordance with the following table:

Gate_polarity	Function
0	Level of the external gate/trigger signal is "high active" or rising edge is active edge
1	Level of the external gate/trigger signal is "low active" or falling edge is active edge

Together with the Clk_input_select bit, the **Gate_effect bit** defines the effect of the external gate/trigger signal in accordance with the following table:

Clk_input_select=0 (counter counts with CLK_TIMT as count/load clock dependent on the preselector value): Gate/Trigger mode switching:

Gate_effect	Function
0	Gate mode: External gate/trigger signal has the effect of a gate (=gate mode) for the count/load clock and has to be active for counting
1	Trigger mode: Each active signal edge of the external gate/trigger signal causes the triggering of the counter with the reload value if the current counter value is unequal to the reload value or has no effect if the current counter value is equal to the reload value (=Trigger mode)

Note on the trigger mode:

If the active signal edge of the external gate/trigger signal occurs in the Trigger mode during CLK_EN=1 and then CLK_EN becomes 0 without the preselector value having gone to 0, the previous active signal edge only becomes effective when CLK_EN is 1 again and the preselector value is 0, i.e. the requirement does not get lost because of CLK_EN=0.

Clk_input_select=1 (counter counts with external gate/trigger signal as count/load clock (edge evaluation)): Count/Toggle mode switching:

Gate_effect	Function
0	Count mode: External gate/trigger signal is the count/load clock (Count mode)
1	Toggle mode: External gate/trigger signal is the count/load clock and causes the switching between - loading of the counter with the reload value (if the current counter value is unequal to the reload value) - down-counting (if the current counter value is equal to the reload value)

Thus, delays as well as time monitorings can be implemented.

Note: The loading of the counter can simultaneously mean the starting of the counter, namely if Reload_disable=1 and Counter_value=0 and load value #0.

Note: The bit combination Gate_effect=1 and Clk_input_select=1 can be used to generate a symmetrical output signal out of an unsymmetrical input signal (reload value=1).

The TIM_OUT output of the TIMER module is active when the counter has the value '0' (non-saving), otherwise passive. The active level of the TIM_OUT output is defined by the **Timer_out_polarity bit** in accordance with the following table:

Timer_out_polarity	Function
0	TIM_OUT = high active

1	TIM_OUT = low active
---	----------------------

The storing of the current counter values in the event registers **Ext_Event_1** and **Ext_Event_2** is done dependent on the bits Event1_control, Event2_control, Event1_Inversion, Event2_inversion in the mode register and dependent on the inputs EXT_EV_1, EXT_EV_2 in accordance with the following tables:

Bit Event1_control

Event1_control	Function
00	EXT_EV_1 input does not affect register Ext_Event_1 and register Ext_Event_2
01	Rising edge of the EXT_EV_1 input leads to the storing of the counter value in the register Ext_Event_1 *)
10	Falling edge of the EXT_EV_1 input leads to the storing of the counter value in the register Ext_Event_2 *)
11	Rising edge of the EXT_EV_1 input leads to the storing of the counter value in the register Ext_Event_1, falling edge of the EXT_EV_1 input leads to the storing of the counter value in the register Ext_Event_2 *)

*) The above table is valid if the Event1_Inversion bit is set to 0; if the bit is set to 1, the "rising edge" and "falling edge" are to be exchanged.

Bit Event2_control

Event2_control	Function
00	EXT_EV_2 input does not affect register Ext_Event_1 and register Ext_Event_2
01	Rising edge of the EXT_EV_2 input leads to the storing of the counter value in the register Ext_Event_1
10	Falling edge of the EXT_EV_2 input leads to the storing of the counter value in the register Ext_Event_2
11	Rising edge of the EXT_EV_2 input leads to the storing of the counter value in the register Ext_Event_1, the falling edge of the EXT_EV_2 input leads to the storing of the counter value in the register Ext_Event_2

*) The above table is valid if the Event2_Inversion bit is set to 0; if the bit is set to 1, the "rising edge" and "falling edge" are to be exchanged

All combinations of the bits Event2_control, Event1_control in which the EXT_EV_1 input as well as the EXT_EV_2 input would cause a storing of the counter value in the same register are forbidden. If these combinations are set nevertheless, the counter value is not stored.

The registers Ext_Event_1 and Ext_Event_2 are readable and writable by the software; the setting of the registers after reset is '0000h'.

If the condition for storing the current counter values in the event registers Ext_Event_1 and Ext_Event_2 occurs simultaneously with a writing of one of these registers, the writing is treated preferentially.

The storing of the current counter values in the event register Int_Event is triggered by a positive edge of the INT_EV input. The register is readable by the software and writable; the setting of the registers after reset is '0000h'.

By storing the current counter values with the edges of the inputs EXT_EV_1, EXT_EV_2 into the event registers Ext_Event_1 and Ext_Event_2 it is possible to measure time intervals and periods between the inputs EXT_EV_1, EXT_EV_2. By storing the current counter value with the rising edge of the INT_EV input into the event register Int_Event it is possible to measure time intervals and periods between write accesses to the SW event trigger register.

The **loading of the counter** can be triggered in four different ways:

- Loading of the counter by writing the Init_bit in the mode register
- Loading of the counter by writing the load register
- Automatic loading of the counter after having reached the value '0' in the reload mode
- Loading of the counter by an external load signal (EXT_GATE_TRIG signal)

The counter counts/loads only if the following count/load conditions are met:

- Input CLK_EN=1
- Count/load clock of the TIMER module active.

Exception: The loading of the counter with the value from the load register always takes place when writing the Init_bit in the mode register.

Each TIMER activity which is carried out dependent on the edge of an input signal (INT_GATE_TRIG or EXT_GATE_TRIG, EXT_EV1, 2, INT_EV) becomes effective with the third CLK_TIMT signal at the latest.

The active edges of the input signals at which the edge evaluation is carried out have to have a minimum distance, also see "Timing requirements".

Event register:

Each TIMER module has:

- Two HW event registers (Ext_Event_1 and Ext_Event_2) in which the current counter values are stored with an edge of the inputs Ext_Ev_1, Ext_Ev_2 (edge programmable), and
- One SW event register (Int_Event) for storing the current counter values with a positive edge of the input INT_EV.

2.3.10.3.4 Overview of the count modes

Gate_polarity	Ext_gate_trig_enable	Gate_effect	Clk_input_select	Function
0	0	0	0	Gate mode: Counter counts/loads with internal clock CLK_TIMT dependent on the preslector value as long as INT_GATE_TRIG=1.
0	0	0	1	Count mode: Counter counts/loads with each rising edge of INT_GATE_TRIG, independently of the preslector value.
0	0	1	0	Trigger mode: Counter counts/loads with internal clock dependent on the preslector value and each rising edge of INT_GATE_TRIG triggers the counter (sets the counter to the load/reload value if the current counter value is unequal to the load/reload value or has no effect if the current counter value is equal to the reload value).
0	0	1	1	Toggle mode: Counter changes with each rising edge of INT_GATE_TRIG independently of the preslector value between - loading with load/reload value (if the current counter value is unequal to the load/reload value) and - down-counting (if the current counter value is equal to the load/reload value)
0	1	0	0	Gate mode: Counter counts/loads with internal clock CLK_TIMT dependent on the preslector value as long as EXT_GATE_TRIG=1.
0	1	0	1	Count mode: Counter counts/loads with each rising edge of EXT_GATE_TRIG, independently of the preslector value.
0	1	1	0	Trigger mode: Counter counts/loads with internal clock dependent on the preslector value and each rising edge of EXT_GATE_TRIG triggers the counter (sets the counter to the load/reload value if the current counter value is unequal to the load/reload value or has no effect if the current counter value is equal to the reload value).
0	1	1	1	Toggle mode: Counter changes with each rising edge of EXT_GATE_TRIG independently of the preslector value between - loading with load/reload value (if the current counter value is unequal to the load/reload value) and - down-counting (if the current counter value is equal to the load/reload value)
1	0	0	0	Gate mode: Counter counts/loads with internal clock CLK_TIMT dependent on the preslector value as long as INT_GATE_TRIG=0.
1	0	0	1	Count mode: Counter counts/loads with each falling edge of INT_GATE_TRIG, independently of the preslector value.
1	0	1	0	Trigger mode: Counter counts/loads with internal clock dependent on the preslector value and each falling edge of INT_GATE_TRIG triggers the counter (sets the counter to the load/reload value if the current counter value is unequal to the load/reload value or has no effect if the current counter value is equal to the reload value).
1	0	1	1	Toggle mode: Counter changes with each falling edge of INT_GATE_TRIG independently of the preslector value between - loading with load/reload value (if the current counter value is unequal to the load/reload value) and - down-counting (if the current counter value is equal to the load/reload value)
1	1	0	0	Gate mode: Counter counts/loads with internal clock CLK_TIMT dependent on the preslector

				value as long as EXT_GATE_TRIG=0.
1	1	0	1	Count mode: Counter counts/loads with each falling edge of EXT_GATE_TRIG independently of the preslector value.
1	1	1	0	Trigger mode: Counter counts/loads with internal clock dependent on the preslector value and each falling edge of EXT_GATE_TRIG triggers the counter (sets the counter to the load/reload value if the current counter value is unequal to the load/reload value or has no effect if the current counter value is equal to the reload value).
1	1	1	1	Toggle mode: Counter changes with each falling edge of EXT_GATE_TRIG independently of the preslector value between - loading with load/reload value (if the current counter value is unequal to the load/reload value) and - down-counting (if the current counter value is equal to the load/reload value)

2.3.10.3.5 Timing requirements

The following applies to the TIMER submodule:

The minimum low pulse width and the minimum high pulse width of the signals INT_GATE_TRIG, EXT_GATE_TRIG, EXT_EV1, EXT_EV2, INT_EV must be at least one CLK_TIMT clock.

Furthermore it is required, that the active edges of the input signals at which the edge evaluation is carried out must have a minimum distance according to the following table:

Signal	Minimum distance
INT_GATE_TRIG, EXT_GATE_TRIG	At least three CLK_TIMT clocks; The following applies additionally: If one of the signals INT_GATE_TRIG or EXT_GATE_TRIG is used as a gate or trigger, it has to take at least three CLK_TIMT clocks plus the number of CLK_TIMT clocks set in the preslector in order to become effective.
EXT_EV1, EXT_EV2, INT_EV	At least two CLK_TIMT clocks

2.3.10.3.6 Operating rules

Initialization sequence, modifications of the mode register

Initialization sequence:

1. Stop counter (CLK_EN=0 = setting after reset)
2. Write load register (counter is not loaded if CLK_EN=0)
3. Write mode register with Init_bit = '1':
counter is loaded with load register
4. Start counter (CLK_EN=1).

The initialization sequence also has to be carried out with each modification of the mode register to prevent any side effects when changing the operating mode.

Cascading of TIMER modules:

The TIMER modules can be cascaded with each other, provided that the timer output of the lower-value counter(s) is fed back to the TIMER_TOP module via the EXTERNAL_INPUTS signals and is selected as GATE_TRIG signal at the high-order timer).

Proceed as follows for that:

1. Program the multiplexer in such a way that each high-order counter has the timer output of the low-order counter as GATE_TRIG signal.
2. Define the interrupt evaluation in such a way that the interrupt of the high-order counter is evaluated. The interrupt of the low-order counter(s) must not be evaluated.
3. Set the operating modes (mode register, prescaler register) of the cascaded TIMER modules the same way (not absolutely necessary but reasonable if the cascaded TIMER modules are to count with the same clock).
4. When reading the counter value, make sure that the data is consistent, e.g. write SW event trigger register bits for all cascaded TIMER modules in order to take over the values of these TIMER modules synchronously into the Int_Event registers; then, read the Int_Event registers of these TIMER modules.
5. The setting of the preselectors for the cascaded TIMER modules has to be adapted to the application (same preselectors are reasonable).

Note on the Toggle mode

For the first count/load clock after having set the Toggle mode or after reset or after active Init_bit, it cannot be predicted if the count/load clock causes a loading of the counter with the reload value or a down-counting (depends on the level of the count/load clock during the setting or during the active Init_bits).

If this information is important for the software, it has to poll the level and value of the counter during the first count/load clock.

Be careful when storing into event registers!

The following combination of the bits Event2_control, Event1_control is forbidden:

Event2_control	Event1_control
01	01
01	11

10	10
10	11
11	01
11	10
11	11

2.3.10.3.7 Connections on ERTEC 200P Toplevel

Timer gate/trigger/event inputs

The timer module has an own multiplexer for each timer for the selection of the sources for the external timer gate/trigger inputs and the external timer event inputs. The assignment of the sources to the timer multiplexers is stated in the table below.

ATTENTION: Cascading of the timers is only possible when the outputs of the lower-level timers are connected to the EXT_GATE_TRIG mux of the respective higher-level timer(s)!

ATTENTION: It is expected that all timer inputs/outputs are synchronous with the timer clock. Asynchronous inputs (e.g. of ASIC pins) have to be synchronized accordingly. The synchronization of the ASIC pins is done on the core level via two flip-flops each with 125 MHz.

Timer	Ext_Gate_Trig_MUX	Event_1_MUX	Event_2_MUX
Timer 0-5	0: TIM_TRIG0 (GPIO6/26)	0: TIM_TRIG0 (GPIO6/26)	0: TIM_TRIG0 (GPIO6/26)
	1: TIM_TRIG1 (GPIO7/27)	1: TIM_TRIG1 (GPIO7/27)	1: TIM_TRIG1 (GPIO7/27)
	2: TIM_TRIG2 (GPIO8)	2: TIM_TRIG2 (GPIO8)	2: TIM_TRIG2 (GPIO8)
	3: TIM_TRIG3 (GPIO9)	3: TIM_TRIG3 (GPIO9)	3: TIM_TRIG3 (GPIO9)
	4: TIM_TRIG4 (GPIO10)	4: TIM_TRIG4 (GPIO10)	4: TIM_TRIG4 (GPIO10)
	5: TIM_TRIG5 (GPIO11)	5: TIM_TRIG5 (GPIO11)	5: TIM_TRIG5 (GPIO11)
	6: TIM_OUT0 (Timer 0)	6: TIM_OUT0 (Timer 0)	6: TIM_OUT0 (Timer 0)
	7: TIM_OUT2 (Timer 2)	7: TIM_OUT2 (Timer 2)	7: TIM_OUT2 (Timer 2)
	8: TIM_OUT4 (Timer 4)	8: TIM_OUT4 (Timer 4)	8: TIM_OUT4 (Timer 4)
	9: CLK_OUT (clock divider output timer top)	9: CLK_OUT (clock divider output timer top)	9: CLK_OUT (clock divider output timer top)
	10: '0'	10: '0'	10: '0'
	11: '0'	11: '0'	11: '0'
	12: '0'	12: '0'	12: '0'
	13: '0'	13: '0'	13: '0'
	14: '0'	14: '0'	14: '0'
	15: '0'	15: '0'	15: '0'

Tabelle 1 : Table Assignment of the timer gate/trigger/event inputs to the timer multiplexers

With the external inputs TIM_TRIG0-5 it is possible to control the Timers from external HW. With the internal inputs TIM_OUT0, TIM_OUT2, TIM_OUT4, it is possible to link timers to each other. The output of the clock divider CLK_OUT in the timer top is connected with bit9 of the input multiplexer.

All six timer outputs (TIM_OUT0-5) are connected to ASIC outputs via GPIOs and the Interrupt Controller (IRQ21-26) (Kap. 2.3.2.14).

The minimum distance of all signals (Bit0 – Bit15) must be at least 3 x 125MHz clocks (see chapter 1.3 in Errata-Sheet of Timer_TOP modul /31/). That means e.g. clock divider value > 3 for CLK_OUT.

Note:

It is advisable to share the timers between the application and the PN stack. Timer 0-2 should use for the PN stack and Timer 3-5 for the application.

2.3.10.3.8 Address Mapping

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	D4h	timer	APB

Module	Register/Memory	Read	Write	Address
/timer				
	TIM_0_MODE_REG	(r)	(w)	0h
	TIM_0_PRESCALER_REG	(r)	(w)	4h
	TIM_0_LOAD_REG	r	w	8h
	TIM_0_COUNT_REG	r		Ch
	TIM_0_INT_EV_REG	rh	w	10h
	TIM_0_EXT_EV_1_REG	rh	w	14h
	TIM_0_EXT_EV_2_REG	rh	w	18h
	TIM_1_MODE_REG	(r)	(w)	20h
	TIM_1_PRESCALER_REG	(r)	(w)	24h
	TIM_1_LOAD_REG	r	w	28h
	TIM_1_COUNT_REG	r		2Ch
	TIM_1_INT_EV_REG	rh	w	30h
	TIM_1_EXT_EV_1_REG	rh	w	34h
	TIM_1_EXT_EV_2_REG	rh	w	38h
	TIM_2_MODE_REG	(r)	(w)	40h
	TIM_2_PRESCALER_REG	(r)	(w)	44h
	TIM_2_LOAD_REG	r	w	48h
	TIM_2_COUNT_REG	r		4Ch
	TIM_2_INT_EV_REG	rh	w	50h
	TIM_2_EXT_EV_1_REG	rh	w	54h
	TIM_2_EXT_EV_2_REG	rh	w	58h
	TIM_3_MODE_REG	(r)	(w)	60h
	TIM_3_PRESCALER_REG	(r)	(w)	64h
	TIM_3_LOAD_REG	r	w	68h
	TIM_3_COUNT_REG	r		6Ch
	TIM_3_INT_EV_REG	rh	w	70h

	TIM_3_EXT_EV_1_REG	rh	w	74h
	TIM_3_EXT_EV_2_REG	rh	w	78h
	TIM_4_MODE_REG	(r)	(w)	80h
	TIM_4_PRESCALER_REG	(r)	(w)	84h
	TIM_4_LOAD_REG	r	w	88h
	TIM_4_COUNT_REG	r		8Ch
	TIM_4_INT_EV_REG	rh	w	90h
	TIM_4_EXT_EV_1_REG	rh	w	94h
	TIM_4_EXT_EV_2_REG	rh	w	98h
	TIM_5_MODE_REG	(r)	(w)	A0h
	TIM_5_PRESCALER_REG	(r)	(w)	A4h
	TIM_5_LOAD_REG	r	w	A8h
	TIM_5_COUNT_REG	r		ACh
	TIM_5_INT_EV_REG	rh	w	B0h
	TIM_5_EXT_EV_1_REG	rh	w	B4h
	TIM_5_EXT_EV_2_REG	rh	w	B8h
	GATE_TRIG_CONTROL_REG	(r)	(w)	C0h
	CLOCK_DIVIDER_REG	(r)	(w)	C4h
	EXT_GATE_TRIG_MUX_REG	(r)	(w)	C8h
	EXT_EV_1_MUX_REG	(r)	(w)	CCh
	EXT_EV_2_MUX_REG	(r)	(w)	D0h
	SW_EVENT_TRIGGER_REG		(w)	D4h

2.3.10.3.9 Register Description

Module: /timer

Register:	TIM_0_MODE_REG			Address:	0h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)
Description:	Timer Mode Register for Timer 0					
Bit	Identifier	Reset	Attr.	Function / Description		
0	INIT_BIT	0h	w	Initialization (0 =INIT_BIT not active, 1 =INIT_BIT active)		

3dt1	<reserved>	0h			not used
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0 =CLK_TIMT, 1 =ext. Gate-/Triggersignal, see also bit GATE_EFFECT)
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0 =Reload Mode is active, 1 =Single Mode is active
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when Writing Load-Register 0 =Reload Mode is active, 1 =Single Mode is active
7	EXT_GATE_TRIG_ENABLE	0h	r	w	Selection of Gate-/Trigger signal 0 =INT_GATE_TRIG, 1 =EXT_GATE_TRIG is Gate-/Trigger signal
8	GATE_POLARITY	0h	r	w	polarity of ext. Gate/Trigger signal 0 =high active/rising edge, 1 =low active/falling edge
9	GATE_EFFECT	0h	r	w	Effect of ext. Gate/Trigger signal 0 =Gate-/Count-Mode for CLK_INPUT_SELECT =0/1,1 =Trigger-/Toggle-Mode for CLK_INPUT_SELECT =0/1
10	TIMER_OUT_POLARITY	0h	r	w	polarity of TIM_OUT 0 =high active, 1 =low active
12dt11	EVENT1_CONTROL	0h	r	w	Effect of EXT_EV1 on External Event register
13	EVENT1_INVERSION	0h	r	w	Inversion of Event1 0 =not inverted, 1 =inverted
15dt14	EVENT2_CONTROL	0h	r	w	Effect of EXT_EV2 on External Event register
16	EVENT2_INVERSION	0h	r	w	Inversion of Event2 0 =not inverted, 1 =inverted
31dt17	<reserved>	0h			not used

Register:	TIM_0_PRESCALER_REG			Address:	4h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)
Description:	Timer PRESCALER Register for Timer 0					
Bit	Identifier	Reset	Attr.	Function / Description		

7dt0	PRESCALER_VALUE	00h	r	w	Prescaler Value
31dt8	<reserved>	000000h			not used

Register:	TIM_0_LOAD_REG					Address:	8h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer LOAD/RELOAD Register for Timer 0					
Bit	Identifier		Reset	Attr.	Function / Description			
31dt0	LOAD_RELOAD_VALUE		00000000h	r	w	Load/Reload value		

Register:	TIM_0_COUNT_REG					Address:	Ch
Bits:	31dt0		Reset value:	00000000h		Attributes:	r
Description:		Timer COUNT Register for Timer 0					
Bit	Identifier		Reset	Attr.	Function / Description		
31dt0	COUNTER_VALUE		00000000hr		Counter Value		

Register:	TIM_0_INT_EV_REG					Address:	10h
Bits:	31dt0		Reset value:	00000000h		Attributes:	r w
Description:			Timer INT_EVENT Register for Timer 0				
Bit	Identifier		Reset	Attr.	Function / Description		
31dt0	INT_EVENT_VALUE		00000000h	r w	Internal Event Value		

Register:	TIM_0_EXT_EV_1_REG					Address:	14h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer EXT_EVENT_2 Register for Timer 0					
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_1_VALUE		00000000h	r	w	External Event 1 Value		

Register:	TIM_0_EXT_EV_2_REG					Address:	18h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer EXT_EVENT_1 Register for Timer 0					
Bit	Identifier		Reset	Attr.	Function / Description			
31dt0	EXT_EVENT_2_VALUE		00000000h	r	w	External Event 2 Value		

Register:	TIM_1_MODE_REG				Address:	20h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)
Description:		Timer Mode Register for Timer 1					
Bit	Identifier	Reset	Attr.		Function / Description		
0	INIT_BIT	0h		w	Initialization (0 =INIT_BIT not active, 1 =INIT_BIT active)		
3dt1	<reserved>	0h			not used		
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0 =CLK_TIMT, 1 =ext. Gate-/Triggersignal, see also bit GATE_EFFECT)		
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0 =Reload Mode is active, 1 =Single Mode is active		
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when Writing Load-Register 0 =Reload Mode is active, 1 =Single Mode is active		
7	EXT_GATE_TRIG_ENABLE	0h	r	w	Selection of Gate-/Trigger signal 0 =INT_GATE_TRIG, 1 =EXT_GATE_TRIG is Gate-/Trigger signal		
8	GATE_POLARITY	0h	r	w	polarity of ext. Gate/Trigger signal 0 =high active/rising edge, 1 =low active/falling edge		
9	GATE_EFFECT	0h	r	w	Effect of ext. Gate/Trigger signal 0 =Gate-/Count-Mode for CLK_INPUT_SELECT =0/1,1 =Trigger-/Toggle-Mode for CLK_INPUT_SELECT =0/1		
10	TIMER_OUT_POLARITY	0h	r	w	polarity of TIM_OUT 0 =high active, 1 =low active		
12dt11	EVENT1_CONTROL	0h	r	w	Effect of EXT_EV1 on External Event register		
13	EVENT1_INVERSION	0h	r	w	Inversion of Event1 0 =not inverted, 1 =inverted		
15dt14	EVENT2_CONTROL	0h	r	w	Effect of EXT_EV2 on External Event register		
16	EVENT2_INVERSION	0h	r	w	Inversion of Event2 0 =not inverted, 1 =inverted		
31dt17	<reserved>	0h			not used		

Register:	TIM_1_PRESCALER_REG					Address:	24h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r)	(w)
Description:		Timer PRESCALER Register for Timer 1						
Bit	Identifier		Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE		00h	r	w	Prescaler Value		
31dt8	<reserved>		000000h			not used		

Register:	TIM_1_LOAD_REG					Address:	28h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer LOAD/RELOAD Register for Timer 1					
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	LOAD RELOAD VALUE		00000000h	r	w	Load/Reload value		

Register:	TIM_1_COUNT_REG					Address:	2Ch	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	
Description:			Timer COUNT Register for Timer 1					
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	COUNTER_VALUE		00000000hr			Counter Value		

Register:	TIM_1_INT_EV_REG					Address:	30h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer INT_EVENT Register for Timer 1					
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	INT_EVENT_VALUE		00000000h	r	w	Internal Event Value		

Register:	TIM_1_EXT_EV_1_REG					Address:	34h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:		Timer EXT_EVENT_1 Register for Timer 1						
Bit	Identifier		Reset	Attr.	Function / Description			
31dt0	EXT_EVENT_1_VALUE		00000000h	r	w	External Event 1 Value		

Register:	TIM_1_EXT_EV_2_REG				Address:	38h	
-----------	--------------------	--	--	--	----------	-----	--

Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Timer EXT_EVENT_2 Register for Timer 1					
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	EXT_EVENT_2_VALUE	00000000h	rh w	External Event 2 Value		

Register:	TIM_2_MODE_REG				Address:	40h
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)
Description:	Timer Mode Register for Timer 2					
Bit	Identifier	Reset	Attr.	Function / Description		
0	INIT_BIT	0h	w	Initialization (0 =INIT_BIT not active, 1 =INIT_BIT active)		
3dt1	<reserved>	0h		not used		
4	CLK_INPUT_SELECT	0h	r w	Selection of Count-/Load-Clock 0 =CLK_TIMT, 1 =ext. Gate-/Triggersignal, see also bit GATE_EFFECT)		
5	RELOAD_DISABLE	0h	r w	Reload Mode disable 0 =Reload Mode is active, 1 =Single Mode is active		
6	DIS_RLD_WHEN_WR_LDREG	0h	r w	Disable Reload when Writing Load-Register 0 =Reload Mode is active, 1 =Single Mode is active		
7	EXT_GATE_TRIG_ENABLE	0h	r w	Selection of Gate-/Trigger signal 0 =INT_GATE_TRIG, 1 =EXT_GATE_TRIG is Gate-/Trigger signal		
8	GATE_POLARITY	0h	r w	polarity of ext. Gate/Trigger signal 0 =high active/rising edge, 1 =low active/falling edge		
9	GATE_EFFECT	0h	r w	Effect of ext. Gate/Trigger signal 0 =Gate-/Count-Mode for CLK_INPUT_SELECT =0/1, 1 =Trigger-/Toggle-Mode for CLK_INPUT_SELECT =0/1		
10	TIMER_OUT_POLARITY	0h	r w	polarity of TIM_OUT 0 =high active, 1 =low active		
12dt11	EVENT1_CONTROL	0h	r w	Effect of EXT_EV1 on External Event register		
13	EVENT1_INVERSION	0h	r w	Inversion of Event1		

					0 =not inverted, 1 =inverted
15dt14	EVENT2_CONTROL	0h	r	w	Effect of EXT_EV2 on External Event register
16	EVENT2_INVERSION	0h	r	w	Inversion of Event2 0 =not inverted, 1 =inverted
31dt17	<reserved>	0h			not used

Register:	TIM_2_PRESCALER_REG					Address:	44h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r)	(w)
Description:		Timer PRESCALER Register for Timer 2						
Bit	Identifier		Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE		00h	r	w	Prescaler Value		
31dt8	<reserved>		000000h			not used		

Register:	TIM_2_LOAD_REG					Address:	48h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer LOAD/RELOAD Register for Timer 2					
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	LOAD RELOAD VALUE		00000000h	r	w	Load/Reload value		

Register:	TIM_2_COUNT_REG				Address:	4Ch	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r
Description:			Timer COUNT Register for Timer 2				
Bit	Identifier		Reset	Attr.	Function / Description		
31dt0	COUNTER_VALUE		00000000hr		Counter Value		

Register:	TIM_2_INT_EV_REG					Address:	50h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer INT_EVENT Register for Timer 2					
Bit	Identifier		Reset	Attr.	Function / Description			
31dt0	INT_EVENT_VALUE		00000000h	r	w	Internal Event Value		

Register:	TIM_2_EXT_EV_1_REG				Address:	54h
Bits:	31dt0	Reset value:	00000000h		Attributes:	r w

Description:		Timer EXT_EVENT_1 Register for Timer 2			
Bit	Identifier	Reset	Attr.		Function / Description
31dt0	EXT_EVENT_1_VALUE	00000000h	r	w	External Event 1 Value

Register:		TIM_2_EXT_EV_2_REG				Address:		58h			
Bits:		31dt0		Reset value:		00000000h		Attributes:		r	w
Description:			Timer EXT_EVENT_2 Register for Timer 2								
Bit	Identifier			Reset		Attr.		Function / Description			
31dt0	EXT_EVENT_2_VALUE			00000000h		r	w	External Event 2 Value			

Register:		TIM_3_MODE_REG				Address:		60h			
Bits:		31dt0		Reset value:		00000000h		Attributes:		(r)	(w)
Description:			Timer Mode Register for Timer 3								
Bit	Identifier			Reset		Attr.		Function / Description			
0	INIT_BIT			0h			w	Initialization (0 =INIT_BIT not active, 1 =INIT_BIT active)			
3dt1	<reserved>			0h				not used			
4	CLK_INPUT_SELECT			0h		r	w	Selection of Count-/Load-Clock 0 =CLK_TIMT, 1 =ext. Gate-/Triggersignal, see also bit GATE_EFFECT)			
5	RELOAD_DISABLE			0h		r	w	Reload Mode disable 0 =Reload Mode is active, 1 =Single Mode is active			
6	DIS_RLD_WHEN_WR_LDREG			0h		r	w	Disable Reload when Writing Load-Register 0 =Reload Mode is active, 1 =Single Mode is active			
7	EXT_GATE_TRIG_ENABLE			0h		r	w	Selection of Gate-/Trigger signal 0 =INT_GATE_TRIG, 1 =EXT_GATE_TRIG is Gate-/Trigger signal			
8	GATE_POLARITY			0h		r	w	polarity of ext. Gate/Trigger signal 0 =high active/rising edge, 1 =low active/falling edge			
9	GATE_EFFECT			0h		r	w	Effect of ext. Gate/Trigger signal 0 =Gate-/Count-Mode for CLK_INPUT_SELECT =0/1,1 =Trigger-/Toggle-Mode for			

					CLK_INPUT_SELECT =0/1
10	TIMER_OUT_POLARITY	0h	r	w	polarity of TIM_OUT 0 =high active, 1 =low active
12dt11	EVENT1_CONTROL	0h	r	w	Effect of EXT_EV1 on External Event register
13	EVENT1_INVERSION	0h	r	w	Inversion of Event1 0 =not inverted, 1 =inverted
15dt14	EVENT2_CONTROL	0h	r	w	Effect of EXT_EV2 on External Event register
16	EVENT2_INVERSION	0h	r	w	Inversion of Event2 0 =not inverted, 1 =inverted
31dt17	<reserved>	0h			not used

Register:	TIM_3_PRESCALER_REG				Address:	64h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)
Description:		Timer PRESCALER Register for Timer 3					
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE	00h	r	w	Prescaler Value		
31dt8	<reserved>	000000h			not used		

Register:	TIM_3_LOAD_REG					Address:	68h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer LOAD/RELOAD Register for Timer 3					
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	LOAD RELOAD VALUE		00000000h	r	w	Load/Reload value		

Register:	TIM_3_COUNT_REG					Address:	6Ch	
Bits:	31dt0		Reset value:		00000000h		Attributes:	r
Description:			Timer COUNT Register for Timer 3					
Bit	Identifier		Reset	Attr.	Function / Description			
31dt0	COUNTER_VALUE		00000000hr		Counter Value			

Register:	TIM_3_INT_EV_REG				Address:	70h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r w
Description:		Timer INT_EVENT Register for Timer 3					

Bit	Identifier	Reset	Attr.	Function / Description
31dt0	INT_EVENT_VALUE	00000000h	rh w	Internal Event Value

Register:	TIM_3_EXT_EV_1_REG			Address:	74h
Bits:	31dt0	Reset value:	00000000h	Attributes:	r w
Description:	Timer EXT_EVENT_1 Register for Timer 3				
Bit	Identifier	Reset	Attr.	Function / Description	
31dt0	EXT_EVENT_1_VALUE	00000000h	rh w		

Register:	TIM_3_EXT_EV_2_REG			Address:	78h
Bits:	31dt0	Reset value:	00000000h	Attributes:	r w
Description:	Timer EXT_EVENT_2 Register for Timer 3				
Bit	Identifier	Reset	Attr.	Function / Description	
31dt0	EXT_EVENT_2_VALUE	00000000h	rh w		

Register:	TIM_4_MODE_REG			Address:	80h
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r) (w)
Description:	Timer Mode Register for Timer 4				
Bit	Identifier	Reset	Attr.	Function / Description	
0	INIT_BIT	0h	w	Initialization (0 =INIT_BIT not active, 1 =INIT_BIT active)	
3dt1	<reserved>	0h		not used	
4	CLK_INPUT_SELECT	0h	r w	Selection of Count-/Load-Clock 0 =CLK_TIMT, 1 =ext. Gate- /Triggersignal, see also bit GATE_EFFECT)	
5	RELOAD_DISABLE	0h	r w	Reload Mode disable 0 =Reload Mode is active, 1 =Single Mode is active	
6	DIS_RLD_WHEN_WR_LDREG	0h	r w	Disable Reload when Writing Load- Register 0 =Reload Mode is active, 1 =Single Mode is active	
7	EXT_GATE_TRIG_ENABLE	0h	r w	Selection of Gate-/Trigger signal 0 =INT_GATE_TRIG, 1 =EXT_GATE_TRIG is Gate-/Trigger signal	

8	GATE_POLARITY	0h	r	w	polarity of ext. Gate/Trigger signal 0 =high active/rising edge, 1 =low active/falling edge
9	GATE_EFFECT	0h	r	w	Effect of ext. Gate/Trigger signal 0 =Gate-/Count-Mode for CLK_INPUT_SELECT =0/1,1 =Trigger-/Toggle-Mode for CLK_INPUT_SELECT =0/1
10	TIMER_OUT_POLARITY	0h	r	w	polarity of TIM_OUT 0 =high active, 1 =low active
12dt11	EVENT1_CONTROL	0h	r	w	Effect of EXT_EV1 on External Event register
13	EVENT1_INVERSION	0h	r	w	Inversion of Event1 0 =not inverted, 1 =inverted
15dt14	EVENT2_CONTROL	0h	r	w	Effect of EXT_EV2 on External Event register
16	EVENT2_INVERSION	0h	r	w	Inversion of Event2 0 =not inverted, 1 =inverted
31dt17	<reserved>	0h			not used

Register:	TIM_4_PRESCALER_REG					Address:	84h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r)	(w)
Description:		Timer PRESCALER Register for Timer 4						
Bit	Identifier		Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE		00h	r	w	Prescaler Value		
31dt8	<reserved>		000000h			not used		

Register:	TIM_4_LOAD_REG					Address:	88h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer LOAD/RELOAD Register for Timer 4					
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	LOAD_RELOAD_VALUE		00000000h	r	w	Load/Reload value		

Register:	TIM_4_COUNT_REG				Address:	8Ch	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r
Description:		Timer COUNT Register for Timer 4					

Bit	Identifier	Reset	Attr.	Function / Description
31dt0	COUNTER_VALUE	00000000h	r	Counter Value

Register:	TIM_4_INT_EV_REG					Address:	90h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer INT_EVENT Register for Timer 4					
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	INT_EVENT_VALUE		00000000h	r	h	w	Internal Event Value	

Register:	TIM_4_EXT_EV_1_REG					Address:	94h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer EXT_EVENT_1 Register for Timer 4					
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_1_VALUE		00000000h	r	h	w	External Event 1 Value	

Register:	TIM_4_EXT_EV_2_REG					Address:	98h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer EXT_EVENT_2 Register for Timer 4					
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_2_VALUE		00000000h	r	w	External Event 2 Value		

Register:	TIM_5_MODE_REG				Address:	A0h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r) (w)
Description:		Timer Mode Register for Timer 5					
Bit	Identifier		Reset	Attr.		Function / Description	
0	INIT_BIT		0h		w	Initialization (0 =INIT_BIT not active, 1 =INIT_BIT active)	
3dt1	<reserved>		0h			not used	
4	CLK_INPUT_SELECT		0h	r	w	Selection of Count-/Load-Clock 0 =CLK_TIMT, 1 =ext. Gate- /Triggersignal, see also bit GATE_EFFECT)	
5	RELOAD_DISABLE		0h	r	w	Reload Mode disable 0 =Reload Mode is active, 1 =Single Mode is active	

6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when Writing Load-Register 0 =Reload Mode is active, 1 =Single Mode is active
7	EXT_GATE_TRIG_ENABLE	0h	r	w	Selection of Gate-/Trigger signal 0 =INT_GATE_TRIG, 1 =EXT_GATE_TRIG is Gate-/Trigger signal
8	GATE_POLARITY	0h	r	w	polarity of ext. Gate/Trigger signal 0 =high active/rising edge, 1 =low active/falling edge
9	GATE_EFFECT	0h	r	w	Effect of ext. Gate/Trigger signal 0 =Gate-/Count-Mode for CLK_INPUT_SELECT =0/1, 1 =Trigger-/Toggle-Mode for CLK_INPUT_SELECT =0/1
10	TIMER_OUT_POLARITY	0h	r	w	polarity of TIM_OUT 0 =high active, 1 =low active
12dt11	EVENT1_CONTROL	0h	r	w	Effect of EXT_EV1 on External Event register
13	EVENT1_INVERSION	0h	r	w	Inversion of Event1 0 =not inverted, 1 =inverted
15dt14	EVENT2_CONTROL	0h	r	w	Effect of EXT_EV2 on External Event register
16	EVENT2_INVERSION	0h	r	w	Inversion of Event2 0 =not inverted, 1 =inverted
31dt17	<reserved>	0h			not used

Register:	TIM_5_PRESCALER_REG					Address:	A4h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r)	(w)
Description:			Timer PRESCALER Register for Timer 5					
Bit	Identifier		Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE		00h	r	w	Prescaler Value		
31dt8	<reserved>		000000h			not used		

Register:	TIM_5_LOAD_REG				Address:	A8h	
Bits:	31dt0		Reset value:	00000000h	Attributes:	r	w
Description:		Timer LOAD/RELOAD Register for Timer 5					
Bit	Identifier		Reset	Attr.	Function / Description		

31dt0	LOAD_RELOAD_VALUE	00000000h	r	w	Load/Reload value
-------	-------------------	-----------	---	---	-------------------

Register:	TIM_5_COUNT_REG					Address:	ACh	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	
Description:			Timer COUNT Register for Timer 5					
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	COUNTER_VALUE		00000000hr			Counter Value		

Register:	TIM_5_INT_EV_REG					Address:	B0h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:			Timer INT_EVENT Register for Timer 5					
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	INT_EVENT_VALUE		00000000h	r	w	Internal Event Value		

Register:	TIM_5_EXT_EV_1_REG					Address:	B4h		
Bits:	31dt0		Reset value:		00000000h		Attributes:	r	w
Description:			Timer EXT_EVENT_1 Register for Timer 5						
Bit	Identifier		Reset		Attr.		Function / Description		
31dt0	EXT_EVENT_1_VALUE		00000000h		r	w	External Event 1 Value		

Register:	TIM_5_EXT_EV_2_REG					Address:	B8h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r	w
Description:		Timer EXT_EVENT_2 Register for Timer 5						
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_2_VALUE		00000000h	r	w	External Event 2 Value		

Register:	GATE_TRIG_CONTROL_REG					Address:	C0h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r)	(w)
Description:		Timer Gate Control Register						
Bit	Identifier		Reset	Attr.		Function / Description		
0	TIM_0_INT_GATE_TRIG		0h	r	w	Software Gate-/Trigger signal Timer 0		
1	TIM_1_INT_GATE_TRIG		0h	r	w	Software Gate-/Trigger signal Timer		

					1
2	TIM_2_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 2
3	TIM_3_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 3
4	TIM_4_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 4
5	TIM_5_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 5
6	TIM_0_CLK_EN	0h	r	w	Clock Enable Timer 0
7	TIM_1_CLK_EN	0h	r	w	Clock Enable Timer 1
8	TIM_2_CLK_EN	0h	r	w	Clock Enable Timer 2
9	TIM_3_CLK_EN	0h	r	w	Clock Enable Timer 3
10	TIM_4_CLK_EN	0h	r	w	Clock Enable Timer 4
11	TIM_5_CLK_EN	0h	r	w	Clock Enable Timer 5
31dt12	<reserved>	00000h			not used

Register:	CLOCK_DIVIDER_REG					Address:	C4h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(r)	(w)
Description:			Timer Clock Divider Register					
Bit	Identifier		Reset	Attr.		Function / Description		
7dt0	CLOCK_DIVIDER_VALUE		00h	r	w	Clock Divider Value		
8	CLK_DIV_EN		0h	r	w	Clock Divider Enable (0 =disabled		
31dt9	<reserved>		000000h			not used		

Register:	EXT_GATE_TRIG_MUX_REG				Address:	C8h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)
Description:		Timer MUX Register External Gate					
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	TIM0_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_GATE_TRIG of Timer 0		
7dt4	TIM1_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_GATE_TRIG of Timer 1		
11dt8	TIM2_EXT_G_T_SEL_3_0	0h	r	w	Selection of		

					EXTERNAL_INPUTS(15:0) for EXT_GATE_TRIG of Timer 2
15dt12	TIM3_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_GATE_TRIG of Timer 3
19dt16	TIM4_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_GATE_TRIG of Timer 4
23dt20	TIM5_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_GATE_TRIG of Timer 5
31dt24	<reserved>	00h			not used

Register:	EXT_EV_1_MUX_REG				Address:	CCh	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)
Description:		Timer MUX Register Event1 Select					
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	TIM0_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 1 of Timer 0		
7dt4	TIM1_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 1 of Timer 1		
11dt8	TIM2_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 1 of Timer 2		
15dt12	TIM3_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 1 of Timer 3		
19dt16	TIM4_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 1 of Timer 4		
23dt20	TIM5_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 1 of Timer 5		
31dt24	<reserved>	00h			not used		

Register:	EXT_EV_2_MUX_REG				Address:	D0h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)

Description:		Timer MUX Register Event2 Select			
Bit	Identifier	Reset	Attr.		Function / Description
3dt0	TIM0_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 2 of Timer 0
7dt4	TIM1_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 2 of Timer 1
11dt8	TIM2_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 2 of Timer 2
15dt12	TIM3_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 2 of Timer 3
19dt16	TIM4_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 2 of Timer 4
23dt20	TIM5_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 2 of Timer 5
31dt24	<reserved>	00h			not used

Register:	SW_EVENT_TRIGGER_REG				Address:	D4h	
Bits:	31dt0		Reset value:	00000000h		Attributes:	(w)
Description:		Timer SW Event Register					
Bit	Identifier		Reset	Attr.		Function / Description	
5dt0	SW_EVENT_TRIG5_0		00h		w	Software Event for Saving Counter value in Int_Event register of Timer 5...0 1: Software event active, read =always 0	
31dt6	<reserved>		0000000h			not used	

2.3.10.4 F-Counter

For F-applications, another timer not dependant on the system clock is required in addition to a system timer. The functionality corresponds to the F-counter function of the IBC blocks.

The F-counter is triggered with a separate `BYP_CLK` input. The F-counter cannot be triggered in clock bypass mode (`CONFIG(2)='1'`) (see 2.3.10.9.3).

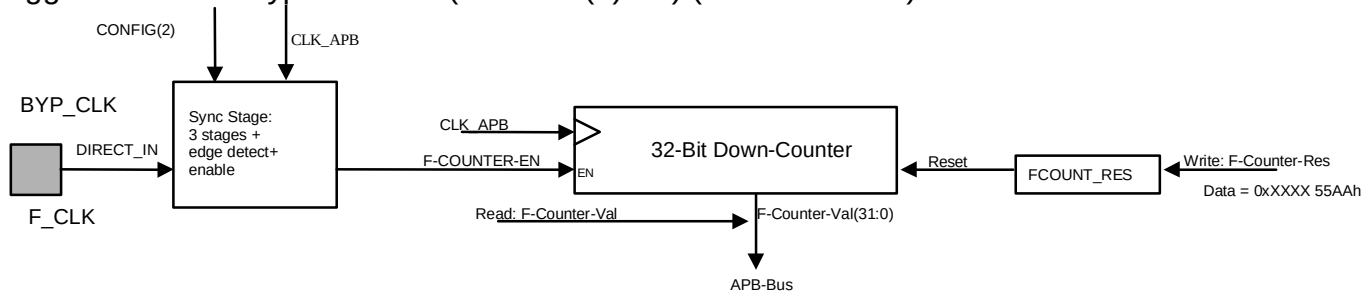


Figure 38: F-timer block diagram

2.3.10.4.1 Function Description

The asynchronous input signal `F_CLK` of the external independent time base is connected to a synchronization stage with the input pin `BYP_CLK`. The SYNC stage is implemented with 3 FF stages to prevent metastable states arising at the counter input. In a downstream stage for edge detection, the count pulses (F-Counter-EN) are generated and connected to the 32-bit down counter. All flip-flops in the circuit operate with the APB clock (125 MHz).

The 32-bit counter is reset to 0x0000 0000h with an asynchronous block reset (`XRESET`, `XSRST` not shown in Figure 38: F-timer block diagram). The next counter pulse sets the counter to 0xFFFF FFFFh. Each subsequent pulse decrements the counter value. When the counter reaches 0x0000 0000h, the next counter pulse triggers a wrap-around to 0xFFFF FFFFh. Writing a value of 0xFFFF 55AAh (`X:= don't care`) to the address of the F-counter reset register (word or halfword) sets the `FCOUNTER_RES` register (see 2.3.10.4.3), which resets the 32-bit counter to 0x0000 0000h. The `FCOUNTER_RES` register is automatically reset to 0 one clock cycle later.

With read access (word) to the F-Counter, the complete 32-bit counter is switched to the APB bus. Word access is required to ensure that the counter value is read consistently. Halfword and byte access is technically possible, but will return an inconsistent counter value.

2.3.10.4.2 Application Information

The maximum input frequency permitted for `F_CLK` is $\frac{1}{4}$ of the APB clock upon quartz failure with a 50% duty cycle.

In standard operation, the PLL outputs 500 MHz with a quartz supply of 25 MHz.

A minimum output frequency is established at the PLL following quartz failure at the ERTEC 200P. This free-running PLL frequency is 100 - 300 MHz. The min. `APB_CLK` frequency following quartz failure is therefore $\text{PLLOUT}_{\min}/4$ (~ 25 MHz).

For safety reasons, the external independent F-CLK should be a maximum of $\frac{1}{4}$ of the minimum possible APB clock (following quartz failure) in order to prevent any errors in edge detection.

The current setting is for an independent F-CLK of 32768 Hz. This ensures a safe difference from the internal system clock.

2.3.10.4.3 Address Mapping

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	4h	FCOUNT	<interface>

Module	Register/Memory	Read	Write	Address
/FCOUNT				
	FCOUNTER_VAL	r(h)		0h
	FCOUNTER_RES	rh	w	4h

2.3.10.4.4 Register Description

Module: /FCOUNT

Register:	FCOUNTER_VAL			Address:	0h
Bits:	31dt0	Reset value:	00000000h	Attributes:	r(h)
Description:	Timerwert des F-Timers				

Register:	FCOUNTER_RES			Address:	4h
Bits:	31dt0	Reset value:	0h	Attributes:	rh w
Description:	Reset register for F-counters. The F-counter is only reset if a data item 0xFFFF 55AAh is entered in this register. Resets can therefore be executed with word and double word access.				
Bit	Identifier	Reset	Attr.	Function / Description	
15dt0	FCOUNT_RES_LWORD	0000h	r(h) w	Lower halfword of F-counter reset	
31dt16	FCOUNT_RES_HWORD	0000h	r(h) w	Upper halfword of F-counter reset (don't care)	

2.3.10.5 UART1...4

ERTEC 200P contains 4 UARTs (UART1...4).

UART1...4 is implemented with the ARM soft macro (PL011). This is *similar* to the standard UART 16C550. A detailed description of the registers and individual functions can be found in 'DDI0183F_uart_pl011_r1p4_trm.pdf' /15/ (see chapter 7.2.)

The PL011 soft macro differs from the standard UART 16C550 as follows:

- Receive FIFO trigger level can be set: 1/8, 1/4, 1/2, 3/4 or 7/8
- The deltas from the modem status signals are not available
- Internal register address mapping and the register bit functions differ

The following 16C550 features are not supported by PL011:

- 1.5 stop bits (only 1 or 2 stop bits are supported)
- Independent 'Receive Clock'

The UARTs can be controlled by the GDMA (see 2.3.4.2.2) and by ARM926.

As the DMA request signals available at UART1...4 are not suitable for GDMA operation, internal status signals are used from the FIFO controller and sent to the GDMA as DMA request signals. The signals for controlling the transmitter and receiver separately over one GDMA channel each (see 0) are as follows:

- UART_RX-FIFO not empty:
The RX FIFO contains at least one character and is not empty. As the UART cannot know when the last character was read in, the GDMA must operate in SINGLE-byte access mode (AHB). 1 to 65536 characters can therefore be transferred per DMA request; the job consists of a transfer entry.
- UART_TX-FIFO half-full or less:
The TX FIFO is at least half empty and therefore has space for 8 characters. The GDMA must operate in INCR8-byte access mode (AHB) to ensure that there is no FIFO overrun. If the transfer length is not modulo 8, the remaining characters are transferred by the GDMA controller with the INCR burst byte (undefined length). 1 to 65536 characters can therefore be transferred per DMA request; the job consists of a transfer entry.

The following UART interrupts are connected to the ARM interrupt controller (IRQ5-12) (2.3.2.14):

- UART1-4_UARTINTR: Group interrupt (modem status, receive FIFO, transmit FIFO, receive timeout, error);
individual interrupts can be masked
- UART1-4_UARTEINTR: Error interrupt

UART supports the following external signals:

- Transmit Data (TXD, Output)
- Receive Data (RXD, Input)
- Clear To Send (CTS, Input)
- Data Carrier Detect (DCD, Input)
- Data Set Ready (DSR, Input)

- Ring Indicator (RI, Input)
- Request to Send (RTS, Output)
- Data Terminal Ready (DTR, Output)

UART1-4 interfaces are available at the GPIO as alternate functions. The alternate function can be set with the SW in the GPIO registers (see 2.3.10.8.4.2).

UART1 incl. modem signals is fully available on GPIO95-32. UART2 can only enable transmit, receive data, CTS and RTS. Only transmit and receive data are available from UART3-4. UART3 is designed for debugging and is located on GPIO31-0. UART2 is also located on GPIO31-0 and UART4 on GPIO95-32.

The IO-Link baud rates are supported; the residual error is < 0.01 %. A total of up to 4 IO-Link interfaces can be used. GPIOs are also required for an IO-Link channel. The function is still to be defined.

2.3.10.5.1 UART Baud Rates

The baud rates are based on the APB clock (125 MHz). The baud rate is calculated using the following formula (ideal baud rate):

$$\mathbf{BRI = F_{UARTCLK} / (BAUDDIV * 16) \quad or \quad BAUDDIV = (F_{UARTCLK} / (16 * BRI))}$$

with $BAUDDIV = (BRD_I), (BRD_F)$ (e.g. when $BAUDIV = 1085 \rightarrow BRD_I = 1, BRD_F = 0.085$)

BAUDDIV comprises an integer part (BRD_I) and a fractional part (BRD_F). The value (m) for setting the fractional divider is calculating using the following formula:

$$\mathbf{m = integer ((BRD_F * 64) + 0.5)}$$

This produces the baud rate divisor used in UART:

$$\mathbf{BAUDDIV_{real} = BRD_I + (m / 64)}$$

This is used to calculate the actual baud rate of the UART using the following formula:

$$\mathbf{BR = F_{UARTCLK} / (BAUDDIV_{real} * 16)}$$

The baud rate error is calculated using the following formula:

$$\mathbf{E_P = ((BR - BRI)/BRI) * 100 [\%]}$$

with:

$F_{UARTCLK}$	= UART basic frequency = APB clock frequency (125 MHz)
BR	= baud rate
BRI	= ideal baud rate
BRD_I	= integer part of the baud rate divisor to be programmed
BRD_F	= fractional part of the baud rate divisor to be programmed
m	= fractional value to be set for the fractional divider (1/64)
E_P	= percentage baud rate deviation from the ideal baud rate

The table below shows the divisor values to be set for the baud rates and deviations from the standard baud rates. The UART (incl. APB interface) with a clock of 125 MHz.

BRI	BRD _i (Integer)	m (Fractional)	BR	E _P %	UseCase
460800	16	61	460829,49	+ 0,01	Debugger
230400	33	58	230414,75	+ 0,01	IO-Link
187500	41	43	187476,57	- 0,01	KBUS-300
115200	67	52	115207,37	+ 0,01	PC
76800	101	46	76804,92	+ 0,01	PC
57600	135	41	57597,05	- 0,01	PC
38400	203	29	38399,51	~ 0	IO-Link
19200	406	58	19199,75	~ 0	PC
14400	542	34	14400,09	~ 0	PC
9600	813	51	9600,06	~ 0	PC
4800	1627	39	4799,98	~ 0	IO-Link
2400	3255	13	2400,00	~ 0	PC
1200	6510	27	1199,99	~ 0	PC
110	<i>Not supported</i>	<i>Not supported</i>	<i>Not supported</i>		

Figure 39: Baud rates UART when F_{UARTCLK} = 125 MHz

2.3.10.5.2 Address Mapping

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	FFCh	UART_PL011	APB

Module	Register/Memory	Read	Write	Address
/UART_PL011				
	UARTDR	r(h)	(w)	0h
	UARTRSR_UARTECR	r(h)		4h
	UARTFR	r(h)		18h
	USRTILPR	r	(w)	20h
	UARTIBRD	r	(w)	24h
	UARTFBRD	r	(w)	28h
	UARTLCR_H	r	(w)	2Ch
	UARTCR	r	(w)	30h
	UARTIFLS	r	(w)	34h
	UARTIMSC	r	(w)	38h
	UARTRIS	r(h)		3Ch
	UARTMIS	r(h)		40h
	UARTICR	(r)	(w)	44h
	UARTDMACR	r	(w)	48h
	UARTTCR	r	(w)	80h
	UARTITIP	r(h)	(w)	84h
	UARTITOP	r(h)	(w)	88h
	UARTTDR	r(h)	(w)	8Ch
	UARTPeriphID0	r		FE0h
	UARTPeriphID1	r		FE4h
	UARTPeriphID2	r		FE8h
	UARTPeriphID3	r		FECh
	UARTPCellID0	r		FF0h
	UARTPCellID1	r		FF4h
	UARTPCellID2	r		FF8h
	UARTPCellID3	r		FFCh

2.3.10.5.3 Register Description

In the description of the UART IP, the read value of the unassigned register bits is in some cases specified as "unpredictable. A read value of 0 is, however, implemented (see UartApbif.vhd, assignment to the NextPRDATA signal).

Module: /UART_PL011

Register:		UARTDR				Address:		0h			
Bits:		31dt0		Reset value:		0h		Attributes:		r(h)	(w)
Description:		UARTDR is the data register Receive (read) data character Transmit (write) data character									
Bit	Identifier			Reset		Attr.		Function / Description			
7dt0	Data			00h		rh w		Receive (read) data character. Transmit (write) data character.			
8	Framing_Error			0h		rh		Framing error. When this bit is set to 1, it indicates that the received character did not have a valid stop bit (a valid stop bit is 1). In FIFO mode, this error is associated with the character at the top of the FIFO.			
9	Parity_Error			0h		rh		Parity error. When this bit is set to 1, it indicates that the parity of the received data character does not match the parity selected as defined by bits 2 and 7 of the UARTLCR_H register. In FIFO mode, this error is associated with the character at the top of the FIFO.			
10	Break_Error			0h		rh		Break error. This bit is set to 1 if a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time (defined as start, data, parity and stop bits). In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state), and the next valid start bit is received.			

11	Overrun_Error	0h	rh	Overrun error. This bit is set to 1 if data is received and the receive FIFO is already full. This is cleared to 0 once there is an empty space in the FIFO and a new character can be written to it.
31dt12	<unused>	00000h	r	

Register:		UARTSR_UARTECR				Address:		4h			
Bits:		31dt0		Reset value:		0h		Attributes:		r(h)	
Description:			receive status register/ error clear register A write to this Register clears the framing, parity, break and over-run error. The data value is not important.								
Bit	Identifier			Reset		Attr.		Function / Description			
0	Framing_Error			0h		rh		When this bit is set to 1, it indicates that the received character did not have a valid stop bit (a valid stop bit is 1).			
1	Parity_Error			0h		rh		When this bit is set to 1, it indicates that the parity of the received data character does not match the parity selected in UARTLCR:H (bit 2).			
2	Break_Error			0h		rh		This bit is set to 1 if a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time (defined as start, data, parity and stop bits).			
3	Overrun_Error			0h		rh		This bit is set to 1 if data is received and the FIFO is already full. This bit is cleared to 0 by a write to UARTECR			
31dt4	<unused>			0000000h		r		Reserved, unpredictable when read.			

Register:		UARTFR				Address:		18h			
Bits:		31dt0		Reset value:		90h		Attributes:		r(h)	
Description:				flag register; after reset TXFF, RXFF and BUSY are 0, TXFE and RXFE are 1.							
Bit	Identifier			Reset		Attr.		Function / Description			
0	Clear_To_Send			0h		rh		(CTS) This bit is the complement of the PrimeCell UART clear to send			

				(nUARTCTS) modem status input. That is, the bit is 1 when the modem status input is 0.
1	Data_Set_Ready	0h	rh	(DSR) This bit is the complement of the PrimeCell UART data set ready (nUARTDSR) modem status input. That is, the bit is 1 when the modem status input is 0.
2	Data_Carrier_Detect	0h	rh	(DCD) This bit is the complement of the PrimeCell UART data carrier detect (nUARTTDCD) modem status input. That is, the bit is 1 when the modem status input is 0.
3	UART_Busy	0h	rh	(BUSY) If this bit is set to 1, the PrimeCell UART is busy transmitting data. This bit remain set until the complete byte, including all the stop bits, has been sent from the shift register. This bit is set as soon as the transmit FIFO becomes non-empty.
4	Receive_FIFO_Empty	1h	rh	(RXFE) The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the receive holding register is empty. If the FIFO is enabled, the RXFE bit is set when the receive FIFO is empty.
5	Transmit_FIFO_Full	0h	rh	(TXFF) The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the transmit holding register is full. If the FIFO is enabled, the TXFF bit is set when the transmit FIFO is full.
6	Receive_FIFO_Full	0h	rh	(RXFF) The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the receive holding register is full. If the FIFO is enabled, the RXFF bit is set when the receive FIFO is full.
7	Transmit_FIFO_Empty	1h	rh	(TXFE) The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the transmit holding register is empty. If

					the FIFO is enabled, the TXFE bit is set when the transmit FIFO is empty.
8	Ring_Indicator	0h	rh		Ring indicator. This bit is the complement of the UART ring indicator (nUARTRI) modem status input. That is, the bit is 1 when the modem status input is 0.
31dt9	<unused>	000000h	r		Reserved, read as zero.

Register:	USRTILPR				Address:	20h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:		8-bit-low-power divisor value. These bits are cleared to 0 at reset.					
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	USRTILPR	00h	r	w	8-bit low-power divisor value.		
31dt8	<unused>	000000h	r		Reserved, read as zero.		

Register:	UARTIBRD					Address:	24h	
Bits:	31dt0		Reset value:	0h		Attributes:	r	(w)
Description:		Integer Part of the Baud Rate Divisor Value Register						
Bit	Identifier		Reset	Attr.		Function / Description		
15dt0	UARTIBRD		0000h	r	w	The integer baud rate divisor.		
31dt16	<unused>		0000h	r		Reserved, read as zero.		

Register:	UARTFBRD					Address:	28h		
Bits:	31dt0		Reset value:		0h		Attributes:	r	(w)
Description:		Fractional Part of the Baud Rate Divisor Value Register							
Bit	Identifier		Reset	Attr.		Function / Description			
5dt0	UARTFBRD		00h	r	w	The fractional baud rate divisor.			
31dt6	<unused>		0000000h	r		Reserved, do not modify, read as zero.			

Register:	UARTLCR_H				Address:	2Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:		line control register, high byte					
Bit	Identifier	Reset	Attr.		Function / Description		
0	Send Break	0h	r	w	If this bit is set to 1, a low level is con-		

					tinually output on the UARTTXD output, after completing transmission of the current character. (...) For normal use, this bit must be cleared to 0.
1	Parity_Enable	0h	r	w	If this bit is set to 1, parity checking and generation is enabled, else parity is disabled and no parity bit added to the data frame.
2	Even_Parity_Select	0h	r	w	If this bit is set to 1, even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits. When cleared to 0 then odd parity is performed which checks for an odd number of 1s.
3	Two_Stop_Bits_Select	0h	r	w	If this bit is set to 1, two stop bits are transmitted at the end of the frame. The receive logic does not check for two stop bits being received.
4	Enable_FIFOs	0h	r	w	If this bit is set to 1, transmit and receive FIFO buffers are enabled (FIFO mode). When cleared to 0 the FIFOs are disabled (character mode), that is, the FIFOs become 1-byte-deep holding registers.
6dt5	Word_lenght	0h	r	w	Word length. The select bits indicate the number of data bits transmitted or received in a frame as follows: 11 = 8 bits 10 = 7 bits 01 = 6 bits 00 = 5 bits.
7	Stick_Parity_Select	0h	r	w	Stick parity select. When bits 1, 2, and 7 of the UARTLCR_H register are set, the parity bit is transmitted and checked as a 0. When bits 1 and 7 are set, and bit 2 is 0, the parity bit is transmitted and checked as a 1. When this bit is cleared stick parity is disabled.
31dt8	<unused>	000000h	r		Reserved, do not modify, read as zero.

Register:	UARTCR			Address:	30h	
Bits:	31dt0	Reset value:	300h	Attributes:	r	(w)

Description:		Control register; all the bits are cleared to 0 on reset.			
Bit	Identifier	Reset	Attr.		Function / Description
0	UART_Enable	0h	r	w	If this bit is set to 1, the PrimeCell UART is enabled.
1	SIR_Enable	0h	r	w	If this bit is set to 1, the IrDA SIR Encoder is enabled. This bit has no effect if the UART is not enabled by bit 0 being set to 1.
2	IrDA_SIR_Low_Power_Mode	0h	r	w	This bit selects the IrDA encoding mode. If this bit is cleared to 0, low level bits are transmitted as an active high pulse with an width of 3/16th of the bit period. If this bit is set to 1, low level bits are transmitted with a pulse width which is 3 times the period of the IrLPBaud16 input signal, regardless of the selected bit rate. Setting this bit uses less power, but may reduce transmission distances.
6dt3	<unused>	0h	r	w	Reserved, do not modify, read as zero.
7	Loop_Back_Enable	0h	r	w	If this bit is set to 1, the SIR Enable is set to 1, and the test register UARTTMR bit 1 (SIRTEST) is set to 1, the nSIROUT path is inverted, and fed through to the SIRIN path. This bit is cleared to 0 on reset, which disables the loopback mode.
8	Transmit_Enable	1h	r	w	If this bit is set to 1, the transmit section is enabled.
9	Receive_Enable	1h	r	w	If this bit is set to 1, the receive section is enabled.
10	DTR	0h	r	w	If this bit is set to 1, the output DTR in 0.
11	RTS	0h	r	w	If this bit is set to 1, the output RTS in 0.
12	Out1	0h	r	w	If this bit is set to 1, the output OUT1 in 0.
13	Out2	0h	r	w	If this bit is set to 1, the output OUT2 in 0.
14	RTS_Enable	0h	r	w	If 1, RTS hardware flow is enabled.
15	CTS_Enable	0h	r	w	If 1, CTR hardware flow is enabled.
31dt16	<unused>	0h	r	w	Reserved, do not modify, read as zero.

Register:	UARTIFLS				Address:	34h	
Bits:	31dt0	Reset value:	12h	Attributes:	r	(w)	
Description:		Interrupt FIFO Level Select Register					
Bit	Identifier	Reset	Attr.		Function / Description		
2dt0	Tx_Int_FIFO_Level	2h	r	w	The trigger points for the transmit interrupt: 000 = RX FIFO becomes <= 1/8 full 001 = RX FIFO becomes <= 1/4 full 010 = RX FIFO becomes <= 1/2 full 011 = RX FIFO becomes <= 3/4 full 100 = RX FIFO becomes <= 7/8 full 101:111 = reserved		
5dt3	Rx_Int_FIFO_Level	2h	r	w	The trigger points for the receive interrupt: 000 = RX FIFO becomes >= 1/8 full 001 = RX FIFO becomes >= 1/4 full 010 = RX FIFO becomes >= 1/2 full 011 = RX FIFO becomes >= 3/4 full 100 = RX FIFO becomes >= 7/8 full 101:111 = reserved		
31dt6	<unused>	0000000h	r		Reserved, do not modify, read as zero.		

Register:	UARTIMSC				Address:	38h	
Bits:	31dt0		Reset value:	0h		Attributes:	r (w)
Description:		Interrupt Mask Set/Clear Register On a write of 1, the mask of the interrupt is set. A write of 0 clears the mask.					
Bit	Identifier		Reset	Attr.		Function / Description	
0	RI_Modem_Int_Mask		0h	r	w		
1	CTS_Modem_Int_Mask		0h	r	w		
2	DCD_Modem_Int_Mask		0h	r	w		
3	DSR_Modem_Int_Mask		0h	r	w		
4	Rx_Int_Mask		0h	r	w		
5	Tx_Int_Mask		0h	r	w		
6	Receive_Timeout_Mask		0h	r	w		
7	Framing_Error_Mask		0h	r	w		
8	Parity_Error_Mask		0h	r	w		

9	Break_Error_Mask	0h	r	w	
10	Overrun_Error_Mask	0h	r	w	
31dt11	<unused>	000000h	r		Reserved, do not modify, read as zero.

Register:		UARTRIS				Address:		3Ch			
Bits:		31dt0		Reset value:		0h		Attributes:		r(h)	
Description:			Raw Interrupt Status Register Gives the raw interrupt state (prior to masking) of the interrupt								
Bit	Identifier			Reset	Attr.		Function / Description				
0	RI_Modem_Int_Status			0h	rh						
1	CTS_Modem_Int_Status			0h	rh						
2	DCD_Modem_Int_Status			0h	rh						
3	DSR_Modem_Int_Status			0h	rh						
4	Rx_Int_Status			0h	rh						
5	Tx_Int_Status			0h	rh						
6	Rx_Timout_Int_Status			0h	rh						
7	Framing_Error_Int_Status			0h	rh						
8	Parity_Error_Int_Status			0h	rh						
9	Break_Err_Int_Status			0h	rh						
10	Overrun_Err_Int_Status			0h	rh						
31dt11	<unused>			000000h	r		Reserved, do not modify, read as zero.				

Register:	UARTMIS				Address:	40h	
Bits:	31dt0		Reset value:	0h		Attributes:	r(h)
Description:		Mask Interrupt Status Register Gives the masked interrupt state (after masking) of the interrupt					
Bit	Identifier		Reset	Attr.		Function / Description	
0	RIMMIS		0h	rh		...RI Intr.	
1	CTSMMIS		0h	rh		...CTS Intr.	
2	DCDMMIS		0h	rh		...DCD Intr.	
3	DSRMMIS		0h	rh		...DSR Intr.	
4	RXMIS		0h	rh		...Receive Interrupt	
5	TXMIS		0h	rh		...Transmit Interrupt	

6	RTMIS	0h	rh		...Receive Timeout
7	FEMIS	0h	rh		...Framing Error
8	PEMIS	0h	rh		...Parity Error
9	BEMIS	0h	rh		...Break Error
10	OEMIS	0h	rh		...Overrun Error
31dt11	<unused>	000000h	r		Reserved, do not modify, read as zero.

Register:		UARTICR				Address:		44h			
Bits:		31dt0		Reset value:		0h		Attributes:		(r)	(w)
Description:			Interrupt Clear Register Clears the Interrupt ...								
Bit	Identifier			Reset		Attr.		Function / Description			
0	RIMIC			0h			w	...RI Intr.			
1	CTSMIC			0h			w	...CTS Intr.			
2	DCDMIC			0h			w	...DCD Intr.			
3	DSRMIC			0h			w	...DSR Intr.			
4	RXIC			0h			w	...Receive Interrupt			
5	TXIC			0h			w	...Transmit Interrupt			
6	RTIC			0h			w	...Receive Timeout			
7	FEIC			0h			w	...Framing Error			
8	PEIC			0h			w	...Parity Error			
9	BEIC			0h			w	...Break Error			
10	OEIC			0h			w	...Overrun Error			
31dt11	<unused>			000000h		r		Reserved, do not modify, read as zero.			

Register:		UARTDMACR				Address:		48h			
Bits:		31dt0		Reset value:		0h		Attributes:		r	(w)
Description:			DMA Control Register								
Bit	Identifier			Reset		Attr.		Function / Description			
0	RXDMAE			0h		r	w	If 1, DMA for the receive FIFO is enabled.			
1	TXDMAE			0h		r	w	If 1, DMA for the transmit FIFO is enabled.			
2	DMAONERR			0h		r	w	If 1, DMA request outputs are disabled			

					when UART error interrupt is asserted.
31dt3	<unused>	00000000h	r		Reserved, do not modify, read as zero.

Register:		UARTTCR				Address:		80h			
Bits:		31dt0		Reset value:		0h		Attributes:		r	(w)
Description:			Test control register (for integration test)								
Bit	Identifier			Reset		Attr.		Function / Description			
0	ITEN			0h		r	w	Integration test enable. When this bit is 1, the UART is placed in integration test mode, otherwise it is in normal mode.			
1	TESTFIFO			0h		r	w	Test FIFO enable. When this bit it 1, a write to the UARTTDR writes data into the receive FIFO, and reads from the UARTTDR reads data out of the transmit FIFO. When this bit is 0, data cannot be read directly from the transmit FIFO or written directly to the receive FIFO (normal operation).			
2	SIRTEST			0h		r	w	SIR test enable. Setting this bit to 1 enables the receive data path during IrDa transmission (testing requires the SIR to be configured in full-duplex mode). This bit must be set to 1 to enable SIR system loop back testing, when the normal mode control register UARTCR bit 7, Loop Back Enable (LBE) has been set to 1. Clearing this bit to 0 disables the receive logic when the SIR is transmitting (normal operation). This bit defaults to 0 for normal operation (half-duplex peration).			
31dt3	<unused>			00000000h		r		Reserved, unpredictable when read.			

Register:		UARTITIP				Address:		84h			
Bits:		31dt0		Reset value:		0h		Attributes:		r(h)	(w)
Description:			Integration test input read/set register (for integration test)								
Bit	Identifier			Reset	Attr.	Function / Description					

0	UARTRXD	0h	rh	w	value of the UARTRXD primary input.
1	SIRIN	0h	rh	w	value of the SIRIN primary input.
2	nUARTDSR	0h	rh	w	value of the nUARTDSR primary input.
3	nUARTCTS	0h	rh	w	value of the nUARTCTS primary input.
4	nUARTDCD	0h	rh	w	value of the nUARTDCD primary input.
5	nUARTRI	0h	rh	w	value of the nUARTRI primary input.
6	UARTRXDMACLR	0h	rh	w	Writes to this bit specify the value to be driven on the intra-chip input, UARTRXDMACLR, in the integration test mode. Reads return the value of UARTRXDMACLR at the output of the test multiplexor.
7	UARTTXDMACLR	0h	rh	w	Writes to this bit specify the value to be driven on the intra-chip input, UARTTXDMACLR, in the integration test mode. Reads return the value of UARTTXDMACLR at the output of the test multiplexor.
31dt8	<unused>	000000h	r		Reserved, unpredictable when read.

Register:	UARTITOP				Address:	88h	
Bits:	31dt0		Reset value:	0h		Attributes:	r(h) (w)
Description:		Integration test output read/set register (for integration test)					
Bit	Identifier		Reset	Attr.		Function / Description	
0	UARTTXD		0h	rh	w	Writes specify the value to be driven on the UARTTXD line in the integration test mode.	
1	nSIROUT		0h	rh	w	Writes specify the value to be driven on the nSIROUT line in the integration test mode.	
2	nUARTDTR		0h	rh	w	Writes specify the value to be driven on the nUARTDTR line in the integration test mode.	
3	nUARTRTS		0h	rh	w	Writes specify the value to be driven on the nUARTRTS line in the integration test mode.	
4	nUARTOut1		0h	rh	w	Writes specify the value to be driven	

					on the nUARTOut1 line in the integration test mode.
5	nUARTOut2	0h	rh	w	Writes specify the value to be driven on the nUARTOut2 line in the integration test mode.
6	UARTINTR	0h	rh	w	Writes specify the value to be driven on the UARTINTR line in the integration test mode. Reads return the value of UARTINTR at the output of the test multiplexor.
7	UARTEINTR	0h	rh	w	Writes specify the value to be driven on the UARTEINTR line in the integration test mode. Reads return the value of UARTEINTR at the output of the test multiplexor.
8	UARTRTINTR	0h	rh	w	Writes specify the value to be driven on the ARTRTINTR line in the integration test mode. Reads return the value of UARTRTINTR at the output of the test multiplexor.
9	UARTTXINTR	0h	rh	w	Writes specify the value to be driven on the UARCTXINTR line in the integration test mode. Reads return the value of UARCTXINTR at the output of the test multiplexor.
10	UARTRXINTR	0h	rh	w	Writes specify the value to be driven on the UARTRXINTR line in the integration test mode. Reads return the value of UARTRXINTR at the output of the test multiplexor.
11	UARTMSINTR	0h	rh	w	Writes specify the value to be driven on the UARTMSINTR line in the integration test mode. Reads return the value of UARTMSINTR at the output of the test multiplexor.
12	UARTRXDMABREQ	0h	rh	w	Writes specify the value to be driven on the UARTRXDMABREQ line in the integration test mode. Reads return the value of UARTRXDMABREQ at the output of the test multiplexor.

13	UARTRXDMASREQ	0h	rh	w	Writes specify the value to be driven on the ARTRXDMASREQ line in the integration test mode. Reads return the value of UARTRXDMASREQ at the output of the test multiplexor.
14	UARTTXDMABREQ	0h	rh	w	Writes specify the value to be driven on the ARTTXDMABREQ line in the integration test mode. Reads return the value of UARTTXDMABREQ at the output of the test multiplexor.
15	UARTTXDMASREQ	0h	rh	w	Writes specify the value to be driven on the ARTTXDMASREQ line in the integration test mode. Reads return the value of UARTTXDMASREQ at the output of the test multiplexor.
31dt16	<unused>	0000h	r		Reserved, read as zero.

Register:		UARTTDR				Address:		8Ch			
Bits:		31dt0		Reset value:		0h		Attributes:		r(h)	(w)
Description:			Test data register (for integration test)								
Bit	Identifier			Reset		Attr.		Function / Description			
10dt0	DATA			000h		rh	w	When the TESTFIFO signal is asserted, data is written into the receive FIFO and read out of the transmit FIFO			
31dt11	<unused>			000000h		r		Reserved, unpredictable when read.			

Register:	UARTPeriphID0					Address:	FE0h		
Bits:	31dt0		Reset value:		11h		Attributes:	r	
Description:		Peripheral ID0 Register, hard coded							
Bit	Identifier		Reset		Attr.		Function / Description		
7dt0	UARTCellID0		11h		r				
31dt8	<unused>		000000h		r		Reserved, read undefined must read as zeros		

Register:	UARTPeriphID1				Address:	FE4h	
-----------	---------------	--	--	--	----------	------	--

Bits:	31dt0	Reset value:	10h	Attributes:	r	
Description:	Peripheral ID1 Register, hard coded					
Bit	Identifier	Reset	Attr.	Function / Description		
7dt0	UARTCellID1	10h	r			
31dt8	<unused>	000000h	r	Reserved, read undefined must read as zeros		

Register:	UARTPeriphID2				Address:	FE8h
Bits:	31dt0	Reset value:	24h	Attributes:	r	
Description:	Peripheral ID2 Register, hard coded					
Bit	Identifier	Reset	Attr.	Function / Description		
7dt0	UARTCellID2	24h	r			
31dt8	<unused>	000000h	r	Reserved, read undefined must read as zeros		

Register:	UARTPeriphID3				Address:	FECh
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	Peripheral ID3 Register, hard coded					
Bit	Identifier	Reset	Attr.	Function / Description		
7dt0	UARTCellID3	00h	r	The fractional baud rate divisor.		
31dt8	<unused>	000000h	r	Reserved, read undefined must read as zeros		

Register:	UARTPCellID0				Address:	FF0h
Bits:	31dt0	Reset value:	Dh	Attributes:	r	
Description:	PrimeCell ID0 Register, hard coded					
Bit	Identifier	Reset	Attr.	Function / Description		
7dt0	UARTPCellID0	0Dh	r			
31dt8	<unused>	000000h	r	Reserved, read undefined must read as zeros		

Register:	UARTPCellID1				Address:	FF4h
Bits:	31dt0	Reset value:	F0h	Attributes:	r	
Description:	PrimeCell ID1 Register, hard coded					

Bit	Identifier	Reset	Attr.	Function / Description
7dt0	UARTPCellID1	F0h	r	
31dt8	<unused>	000000h	r	Reserved, read undefined must read as zeros

Register:	UARTPCellID2					Address:	FF8h	
Bits:	31dt0		Reset value:		5h	Attributes:	r	
Description:		PrimeCell ID2 Register, hard coded						
Bit	Identifier		Reset	Attr.		Function / Description		
7dt0	UARTPCellID2		05h	r				
31dt8	<unused>		000000h	r		Reserved, read undefined must read as zeros		

Register:	UARTPCellID3					Address:	FFCh	
Bits:	31dt0		Reset value:		B1h		Attributes:	r
Description:		PrimeCell ID3 Register, hard coded						
Bit	Identifier		Reset	Attr.		Function / Description		
7dt0	UARTPCellID3		B1h	r				
31dt8	<unused>		000000h	r		Reserved, read undefined must read as zeros		

2.3.10.6 I²C

The ERTEC 200P contains one I²C module for general purpose applications (I²C_3, master / slave interface). This module is located at the Toplevel and controlled by the ARM926EJ-S. Another I²C module (master / slave interface) is located in the PN-IP. The two I²C interfaces for the LWL Transceiver (I²C_1/2) and the general purpose interface (I²C_3) are multiplexed with other peripheral functions on the GPIO interface.

In the document 'I2C_Philips.pdf' /29/ (see chapter 7.2) you will find the I²C Bus Specification and in the document 'mi2c_ps.pdf' the discription of the used Inventra IP MI2C / 30/:

To use the direct access to the I²C bus, the ARM must access appropriately the I²C interface macro registers. These registers are contained in the I²C interface address area. The complete functional description of the I²C interface macro and the included registers is contained in the MI2C document.

The I²C interface contains 2 registers (SW_I2C_EN and SW_I2C_CTRL) which allow to control the I²C-Bus signals by Software. This software interface is enabled by setting bit 0 in register SW_I²C_EN. If this software interface is enabled, the rest of the I²C interface cannot be used.

The 8Bit I²C modules are connected to the APB / SC-Bus (PN-IP). Only word addresses are used for addressing the internal registers. In the case of a write to the module the bit position 31-8 are ignored and in the case of a read from the modul these bit positions are driven with '0'.

2.3.10.6.1 Baudrate generator

In contrast to the MI2C IP specification, the **Clock Control Register** (MI2C_CCR) in the I²C interface cannot be written because the register is not implemented. The Clock Control Register for the general purpose I²C interface (I²C_3) is contained in the SCRB (CCR_I2C). This allows the SCL clock pulse frequency on the I²C bus to be set (see chapter 2.3.10.9.22, System-Control-Register-Block (SCRB)).

For a baudrate of 100 kBaud on the I²C bus the module requires an internal enable signal of 1 MHz. Therefore the input clock has to be divided. This will be done by configuring the register CCR_I2C in the SCRB module. The register CCR_I2C is not configured after reset.

Formula for calculating the divider value:

$$((PCLK\ MHz) / (CCR_I2C + 1)) = 1\ MHz$$

Example:

If PCLK = 125 MHz

then CCR_I2C = 124d = ("01111100b")

Restrictions

Only the baudrate of 100 kBaud (normal mode) is supported. The baudrate of 400 kBaud (fast mode) cannot be supported because a divider value to generate 4MHz is not configurable in CCR_I2C by 125MHz (PCLK).

2.3.10.6.2 IO Expansion Unit

The "IO Expansion Unit" is used to operate a maximum of 4 commonly usable 8-bit IO expanders connected to the I²C interface of the ERTEC 200P. The corresponding registers are located in the I²C address area.

Once configured, the unit automatically reads data from the expander into the register DATA_IN_n (when CTRL_n (IN) = 1) and writes data from the register DATA_OUT_n into the expander (when CTRL_n (OUT) = 1), where n = expander 1 to 4.

The register ADDR_n is used to store the associated slave address of the various expanders. Although EX_ADDR_n is implemented as an 8-bit register, only bits 7....1 are used as slave address. Bit 0 of the register is used by the "IO Expansion Unit" to distinguish between a read access and a write access.

In automatic operation, the registers EX_ADDR_n and EX_DATA_OUT_n must be configured before access can be started by writing to EX_CTRL_n. Depending on the setting of the register CTRL_n (MODE), this function can be used either periodically or as required.

If this register is configured for "as required" (= 0), then CTRL_n (IN) and CTRL_n (OUT) will be reset after the I²C access. The service has to be disabled during the direct access to the I²C bus (CTRL_n (OUT) = CTRL_n (IN) = 0).

The CTRL_n (status) bit indicates the success of the last transfer. If an error has occurred, this bit is set to '1'; whereas it remains '0' after a successful transfer. The internal status register for the I²C module is monitored here.

The activated "IO Expansion Unit" takes control of the I²C macro. This prevents the ARM926 from accessing the I²C until one of the four expanders has been processed. The ARM926 must deactivate all IO Expander services in order to access the I²C. Then the ARM926 must wait until the register CTRL_n (BUSY) indicates the end of all automatic transfers. The microprocessor now has full access to the I²C and operates in immediate access mode. Because the default value set by the software is no longer valid, the register I2C_CNTR must be reconfigured to switch from immediate access to automatic operation.

2.3.10.6.3 Important Software rules

1. Ensure for consecutive sequences of start and stop conditions that a non-existent slave address is loaded into the data register before issuing the stop condition. After issuing the stop condition, the contents of the data register are output as slave address, followed by the stop condition. In automatic operation, the contents of the Error Slave Address register are used as slave address. This routine can be activated using bit '0' in the Error Slave Address register.
2. If a deadlock occurs on the I²C bus, write accesses to bit 7 in the register EX_Control_1 can be used to influence the SCL clock pin. Setting this bit causes SCL to become "low". Conversely, resetting the bit switches SCL to "high".
3. The register I2C_DATA does not have the same source or the same target when it is addressed with a write/read access. When this register is read, only the data transferred on the I²C bus is accessible. This means that the read value can differ from the written value for a write with a subsequent read.
4. To perform the handshake with the I²C, it is expected that the ARM926 either polls the IFLG interrupt flag of the CNTR register, or activates the interrupt output. Once an interrupt has been recognized, a wait time of at least 2 µs must be observed. Only then the next action on the I²C bus can be initiated. Otherwise errors can occur during the next address/data transfer.
5. The I²C controller has a bug which occurs in this way:

To initiate actions on I²C bus commands are written into the CNTR register of the controller in the following way:

Example: Send STOP condition

a) write: bit STB is set

b) write: bit IFLG is reset

Randomly the STOP condition is not sent.

The same bug can occur when sending a Repeated START condition or a STOP condition followed by a START condition. If the bug occurs the STOP condition is not sent and the following START condition is not sent, too. This malfunction is caused by setting the bit STP in one write and resetting bit IFLG in the next write. The bug can be observed by polling bit IFLG (stays 1 for ever), which should be reset by the write or by polling the Status register (keeps value before the write if bug occurs).

To avoid this bug write all commands in 1 write to the CNTR register.

6. When the I²C macro is in slave mode software must ensure that all interrupts of the I²C macro are recorded, otherwise data could get lost or the I²C macro may not respond to its slave address.

2.3.10.6.4 Address Mapping

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	68h	i2c	APB

Module	Register/Memory	Read	Write	Address
/i2c				
	MI2C_ADDR	r	w	0h
	MI2C_DATA	rh	w	4h
	MI2C_CNTR	r(h)	(w)	8h
	MI2C_STAT	rh		Ch
	MI2C_CCR		w	
	MI2C_XADDR	r	w	10h
	MI2C_SOFTWARE_RESET		w	1Ch
	EX_CTRL_1	r(h)	(w)	20h
	EX_ADDR_1	r	w	24h
	EX_DATA_OUT_1	r	w	28h
	EX_DATA_IN_1	rh		2Ch
	EX_CTRL_2	r(h)	(w)	30h
	EX_ADDR_2	r	w	34h
	EX_DATA_OUT_2	r	w	38h
	EX_DATA_IN_2	rh		3Ch
	EX_CTRL_3	r(h)	(w)	40h
	EX_ADDR_3	r	w	44h
	EX_DATA_OUT_3	r	w	48h
	EX_DATA_IN_3	rh		4Ch
	EX_CTRL_4	r(h)	(w)	50h
	EX_ADDR_4	r	w	54h
	EX_DATA_OUT_4	r	w	58h
	EX_DATA_IN_4	rh		5Ch
	ERROR_SLAVE_ADDRESS	r	w	60h
	SW_I2C_EN	r	w	64h
	SW_I2C_CTRL	r(h)	(w)	68h

2.3.10.6.5 Register Description

The addressing is organized: WORD - wise

Module: /i2c

Register:	MI2C_ADDR				Address:	0h	
Bits:	7dt0		Reset value:	00h		Attributes:	r w
Description:		Slave Address					
Bit	Identifier		Reset	Attr.		Function / Description	
0	GCE		0h	r	w	General Call Address enable	
1	SLA0		0h	r	w	Slave Address (1-0), Extended Address (9-8)	
2	SLA1		0h	r	w	Slave Address (1-0), Extended Address (9-8)	
3	SLA2		0h	r	w	Slave Address (6-2)	
4	SLA3		0h	r	w	Slave Address (6-2)	
5	SLA4		0h	r	w	Slave Address (6-2)	
6	SLA5		0h	r	w	Slave Address (6-2)	
7	SLA6		0h	r	w	Slave Address (6-2)	

Register:	MI2C_DATA				Address:	4h	
Bits:	7dt0	Reset value:	00h		Attributes:	rh	w
Description:		Transfer Data Register; Receive-/Transmit-Data					

Register:	MI2C_CNTR				Address:	8h	
Bits:	7dt2		Reset value:	00h		Attributes:	r(h) (w)
Description:							
Bit	Identifier		Reset	Attr.		Function / Description	
2	AAK		0h	r	w	Assert acknowledge	
3	IFLG		0h	rh		Interrupt flag	
4	STP		0h	rh	w	Master mode stop	
5	STA		0h	rh	w	Master mode start	
6	ENAB		0h	r	w	Bus enable	
7	IEN		0h	r	w	Interrupt enable	

Register:	MI2C_STAT			Address:	Ch	
Bits:	7dt0	Reset value:	F8h	Attributes:	rh	
Description:	Status register with content the following 31 status codes:					
	00h: Bus error 08h: START condition transmitted 10h: Repeated START condition transmitted 18h: Address + write bit transmitted, ACK received 20h: Address + write bit transmitted, ACK not received 28h: Data byte transmitted in master mode, ACK received 30h: Data byte transmitted in master mode, ACK not received 38h: Arbitration lost in address or data byte 40h: Address + read bit transmitted, ACK received 48h: Address + read bit transmitted, ACK not received 50h: Data byte received in master mode, ACK transmitted 58h: Data byte received in master mode, not ACK transmitted 60h: Slave address + write bit received, ACK transmitted 68h: Arbitration lost in address as master, slave address + write bit received, ACK transmitted 70h: General Call address received, ACK transmitted 78h: Arbitration lost in address as master, General Call address received, ACK transmitted 80h: Data byte received after slave address received, ACK transmitted 88h: Data byte received after slave address received, not ACK transmitted 90h: Data byte received after General Call received, ACK transmitted 98h: Data byte received after General Call received, not ACK transmitted A0h: STOP or repeated START condition received in slave mode A8h: Slave address + read bit received, ACK transmitted B0h: Arbitration lost in address as master, slave address + read bit received, ACK transmitted B8h: Data byte transmitted in slave mode, ACK received C0h: Data byte transmitted in slave mode, ACK not received C8h: Last byte transmitted in slave mode, ACK received D0h: Second Address byte + write bit transmitted, ACK received D8h: Second Address byte + write bit transmitted, ACK not received E0h: Second Address byte + read bit transmitted, ACK received E8h: Second Address byte + read bit transmitted, ACK not received F8h: No relevant status information, IFLG=0					

Register:	MI2C_XADDR			Address:	10h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:		Extended Slave Address				
0	SLAX0	0h	r	w	Extended Slave Address	
1	SLAX1	0h	r	w	Extended Slave Address	
2	SLAX2	0h	r	w	Extended Slave Address	
3	SLAX3	0h	r	w	Extended Slave Address	
4	SLAX4	0h	r	w	Extended Slave Address	
5	SLAX5	0h	r	w	Extended Slave Address	
6	SLAX6	0h	r	w	Extended Slave Address	
7	SLAX7	0h	r	w	Extended Slave Address	

Register:	MI2C_SOFTWARE_RESET			Address:	1Ch	
Bits:	7dt0	Reset value:	00h	Attributes:		w
Description:	Software Reset					

Register:	EX_CTRL_1			Address:	20h	
Bits:	7dt0	Reset value:	0h	Attributes:	r(h)	(w)
Description:	Control register for IO expander 1.					
Bit	Identifier	Reset	Attr.		Function / Description	
0	MODE	0h	r	w	0 = Service on demand, default 1 = Periodic service	
1	OUT	0h	rh	w	0 = Write service off, default 1 = Write service on Send output data as specified in MODE	
2	IN	0h	rh	w	0 = Read service off, default 1 = Read service on Read input data as specified in MODE	
3	BUSY	0h	rh		Read-only. 0 = Currently inactive, default 1 = Currently active	
4	ERROR	0h	rh		Read-only. 0 = No Error, default 1 = Error occurred	
7	SCL_Toggle	0h	r	w	SCL toggle bit: '0' keeps SCL high, '1' pulls SCL to low.	

Register:	EX_ADDR_1			Address:	24h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:	Slave address for expander 1.					

Register:	EX_DATA_OUT_1			Address:	28h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:	Send data for expander 1.					

Register:	EX_DATA_IN_1			Address:	2Ch	
Bits:	7dt0	Reset value:	00h	Attributes:	rh	

Description:	Read data for expander 1.
--------------	---------------------------

Register:		EX_CTRL_2				Address:		30h			
Bits:		4dt0		Reset value:		0h		Attributes:		r(h)	(w)
Description:			Control register for IO expander 2.								
Bit	Identifier			Reset	Attr.		Function / Description				
0	MODE			0h	r	w	0 = Service on demand, default 1 = Periodic service				
1	OUT			0h	rh	w	0 = Write service off, default 1 = Write service on Send output data as specified in MODE				
2	IN			0h	rh	w	0 = Read service off, default 1 = Read service on Read input data as specified in MODE				
3	BUSY			0h	rh		Read-only. 0 = Currently inactive, default 1 = Currently active				
4	ERROR			0h	rh		Read-only. 0 = No Error, default 1 = Error occurred				

Register:	EX_ADDR_2				Address:	34h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w	
Description:	Slave address for expander 2.						

Register:	EX_DATA_OUT_2				Address:	38h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w	
Description:	Send data for expander 2.						

Register:	EX_DATA_IN_2				Address:	3Ch	
Bits:	7dt0	Reset value:	00h	Attributes:	rh		
Description:		Read data for expander 2.					

Register:	EX_CTRL_3				Address:	40h	
Bits:	4dt0	Reset value:	0h	Attributes:	r(h)	(w)	
Description:	Control register for IO expander 3.						

Bit	Identifier	Reset	Attr.		Function / Description
0	MODE	0h	r	w	0 = Service on demand, default 1 = Periodic service
1	OUT	0h	rh	w	0 = Write service off, default 1 = Write service on Send output data as specified in MODE
2	IN	0h	rh	w	0 = Read service off, default 1 = Read service on Read input data as specified in MODE
3	BUSY	0h	rh		Read-only. 0 = Currently inactive, default 1 = Currently active
4	ERROR	0h	rh		Read-only. 0 = No Error, default 1 = Error occurred

Register:	EX_ADDR_3			Address:	44h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:	Slave address for expander 3.					

Register:	EX_DATA_OUT_3			Address:	48h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:	Send data for expander 3.					

Register:	EX_DATA_IN_3			Address:	4Ch	
Bits:	7dt0	Reset value:	00h	Attributes:	rh	
Description:	Read data for expander 3.					

Register:	EX_CTRL_4			Address:	50h	
Bits:	4dt0	Reset value:	0h	Attributes:	r(h)	(w)
Description:	Control register for IO expander 4.					
Bit	Identifier	Reset	Attr.		Function / Description	
0	MODE	0h	r	w	0 = Service on demand, default 1 = Periodic service	
1	OUT	0h	rh	w	0 = Write service off, default 1 = Write service on Send output data as specified in MODE	

2	IN	0h	rh	w	0 = Read service off, default 1 = Read service on Read input data as specified in MODE
3	BUSY	0h	rh		Read-only. 0 = Currently inactive, default 1 = Currently active
4	ERROR	0h	rh		Read-only. 0 = No Error, default 1 = Error occurred

Register:	EX_ADDR_4				Address:	54h	
Bits:	7dt0	Reset value:	00h		Attributes:	r	w
Description:		Slave address for expander 4.					

Register:	EX_DATA_OUT_4				Address:	58h	
Bits:	7dt0	Reset value:	00h		Attributes:	r	w
Description:		Send data for expander 4.					

Register:	EX_DATA_IN_4				Address:	5Ch	
Bits:	7dt0	Reset value:	00h		Attributes:	rh	
Description:		Read data for expander 4.					

Register:	ERROR_SLAVE_ADDRESS				Address:	60h	
Bits:	7dt0		Reset value:	FAh		Attributes:	r w
Description:			Fictive target address used to automatically create a stop condition following a failed start condition. Bit 0 activates this automatic action.				
Bit	Identifier		Reset	Attr.		Function / Description	
0	ERR_ON_OFF		0h	r	w	Err_on_off = '0' : Error routine off (default). Err_on_off = '1' : Error routine on.	
7dt1	ERROR_SLAVE_ADDRESS		7Dh	r	w	Fictive target address	

Register:	SW_I2C_EN				Address:	64h	
Bits:	0dt0	Reset value:	0h	Attributes:	r	w	
Description:	Software I2C enable: 0 = disabled 1 = enabled						

Register:		SW_I2C_CTRL				Address:		68h			
Bits:		3dt0		Reset value:		Fh		Attributes:		r(h)	(w)
Description:			Software I2C control register.								
Bit	Identifier			Reset	Attr.		Function / Description				
0	SW_SCL_O			1h	r	w	Software I2C clock output: SW_SCL_O = '1' : I2C clock output is tri-state (default). SW_SCL_O = '0' : I2C clock output is '0'.				
1	SW_SDA_O			1h	r	w	Software I2C data output: SW_SDA_O = '1' : I2C data output is tri-state (default). SW_SDA_O = '0' : I2C data output is '0'.				
2	SCL_I			1h	rh		I2C clock output: SCL_I = '1' : I2C clock line = '1'. SCL_I = '0' : I2C clock line is '0'.				
3	SDA_I			1h	rh		I2C data output: SDA_I = '1' : I2C data line = '1'. SDA_I = '0' : I2C data line is '0'.				

2.3.10.7 SPI1/2 (Serial Peripheral Interface)

The SPI1/2 interfaces are "Master-Slave" SPI function blocks. They are connected to the APB bus over a 16-bit interface.

The SPI interface of ERTEC 200P is implemented with an ARM IP, the *ARM PrimeCell™ Synchronous Serial Port Master and Slave (SSPMS) PL021*. Documentation for the SSPMS IP:

The basic frequency for bit rate generation is 125 MHz.

The macro supports the following modes:

- Motorola SPI-compatible mode
- Texas Instruments Synchronous Serial Interface
- National Semiconductor Microwire Interface

The SPI macro has the following features:

- Separate 16-bit wide and 8-entry deep receive and transmit FIFOs
- Programmable bit rate:
Master: 1922.27 Bd – 62.5 MBd (theoretical), constraining is limited to 25 MBd
Slave: max. bit rate 10.42 MBd
- Programmable data frame size of between 4 and 16 bits

The SPI module can be controlled by ARM926 or by the GDMA controller. Access from the external host over the host interface is also possible, but interrupts are not supported in this case. The module FIFO interrupt is not activated or deactivated until 4 entries have been made (in line with the transfer direction).

As the FIFO submodule for GDMA operation cannot be disabled, a module modification is required for independent GDMA operation (no ARM926 activity). For the receiver, the internal SPI module signal 'SPI1/2_SSPRXDMA, corresponding to RNE: Receive FIFO not empty' is sent to the module border to display the receive FIFO status for the GDMA. For the transmitter, the interrupt signal 'SPI1/2_SSPTXINTR, corresponding to the state: Transmit FIFO is half full or less' is used to display the transmit FIFO status for the GDMA.

- **SPI1/2_SSPRXDMA:**
The RX FIFO contains at least one character and is not empty. As the SPI cannot know when the last character was read in, the GDMA must operate in SINGLE-byte access mode (AHB). 1 to 65536 characters can therefore be transferred per DMA request; the job consists of a transfer entry.
- **SPI1/2_SSPTXINTR:**
The SSPTXINTR interrupt (Transmit FIFO is half full or less) must be enabled for operation. The TX FIFO is at least half empty and therefore has space for 4 characters. The GDMA must operate in INCR4-byte access mode (AHB) to ensure that there is no FIFO overrun. If the transfer length is not modulo 4, the remaining characters are transferred by the GDMA controller with the INCR burst byte (undefined length). 1 to 65536 characters can therefore be transferred per DMA request; the job consists of a transfer entry.

The following SPI interrupts are connected to the ARM interrupt controller (IRQ13-16) (2.3.2.14):

- **SPI1/2_SSPIINTR** Group interrupt
- **SPI1/2_SSPRORINTR** Overrun error interrupt

For byte-specific operation of the SPI1/2 interfaces by ARM926, the following internal status from the SPI modules is sent to the ARM interrupt controller (ARM-ICU) (IRQ28-31, see 2.3.2.14):

- **SPI1/2_RNE** Receive FIFO not empty (corresponds to SPI1/2_SSPRXDMA)
- **SPI1/2_TFE** Transmit FIFO empty

The status information are used from the internal SPI-IP status register SSPSR. This ensures the timing required for steady redundant communication between IM modules.

The SPI1/2 interfaces share the external pins with GPIOs / XHIF. The SPI1/2 interfaces are only available if the GPIO control registers are programmed accordingly or the relevant SPI interface is set as the boot medium with the boot pins. In this case, the boot loader must enable the SPI interface in the internal ROM with the required GPIO control register configuration.

Each SPI1/2 interface supports the following external signals:

- **SFRMOUT (Output):** Serial Frame Output (Master)
- **SCLKOUT (Output):** Serial Clock Output (Master)

- SSPCTLOE (Output): Output enable signal for SCLKOUT and SFRMOUT
- SSPTXD (OUTPUT): Serial data (Output:
- SSPOE (Output): Output enable signal when SSPTXD is valid
- SFRMIN (Input): Serial Frame Input (Slave)
- SCLKIN (Input): Serial Clock Input (Slave)
- SSPRXD (Input): Serial Data Input

If the SPI1/2 output signals are enabled at the GPIO, they drive the outputs directly and are not subject to output enable control.

The following baud rates apply for the clock output of the synchronous serial interfaces in line with the SPI1/2 parameters.

$$F_{\text{CLKOUT}} = 125\text{MHz} / (\text{CPSDVR} * (1 + \text{SCR}))$$

The following applies for the parameters:

SPI master:	CPSDVR	:= (2..254), multiples of 2 only
	SCR	:= (0..255)
SPI slave:	CPSDVR	:= (4..254), multiples of 2 only
	SCR	:= (2..255)

This produces baud rates ranging from 62.5 MBd (master) / 10.42 MBd (slave) down to 1922.27 Bd. The timing of the master for this baud rate cannot, however, be accessed from the pad cell or the board; it is limited 25 MBaud by constraining.

2.3.10.7.1 Features of the SPI Interface

- Can be used as SPI master and SPI slave.
- Maximum data rate (with 125 MHz APB clock):
 - 62.5 Mbps in master mode (1/2 * APB clock frequency) → limited to 25 Mdps
 - 10.42 Mbps in slave mode (1/12 * APB clock frequency)
- Length of data frame:
 - 4 to 16 bits, adjustable
- Data bit transmission sequence:
 - MSB first (LSB first is not supported by SSPMS IP)
- DMA request signals
 - A DMA request signal becomes active if
 - The transmit FIFO is empty or
 - The receive FIFO contains at least one entry.
- Interrupts
 - A group interrupt is activated if
 - The transmit FIFO is empty,
 - The transmit FIFO is half full or emptier,
 - The transmit FIFO is not full,
 - The receive FIFO is not empty,
 - The receive FIFO is half full or fuller or
 - The receive FIFO overruns.
 - The interrupt sources can be masked individually.

Note on switching between master and slave mode during operation:

According to the specifications, the SSPMS IP does *not support dynamic* toggling between master and slave mode. Simulations have, however, shown that master → slave and slave → master toggling *during operation* is possible without resetting the SSPMS IP. There are no specific time requirements to be met, but the current SPI transfer must be completed for the operating mode in question (master or slave).

2.3.10.7.1.1 SSPMS IP Issues

If the SPI-IP is operated in TI mode (`SSPCR0.FRF = "01"`), the unused bits from the 16-bit data register whose used width is configured with `SSPCR0.DSS` are not defined in terms of data content and must therefore be masked out by the SW after being read.

2.3.10.7.2 SSPMS IP Extension

The SSPMS IP used for the ERTEC 200P SPI interface is not the original version; it has the following additional interface signals:

- "Receive FIFO not empty" signal (`SSP_SSPRXDMA`)
- "Transmit FIFO empty" signal (`SSP_TFE`)

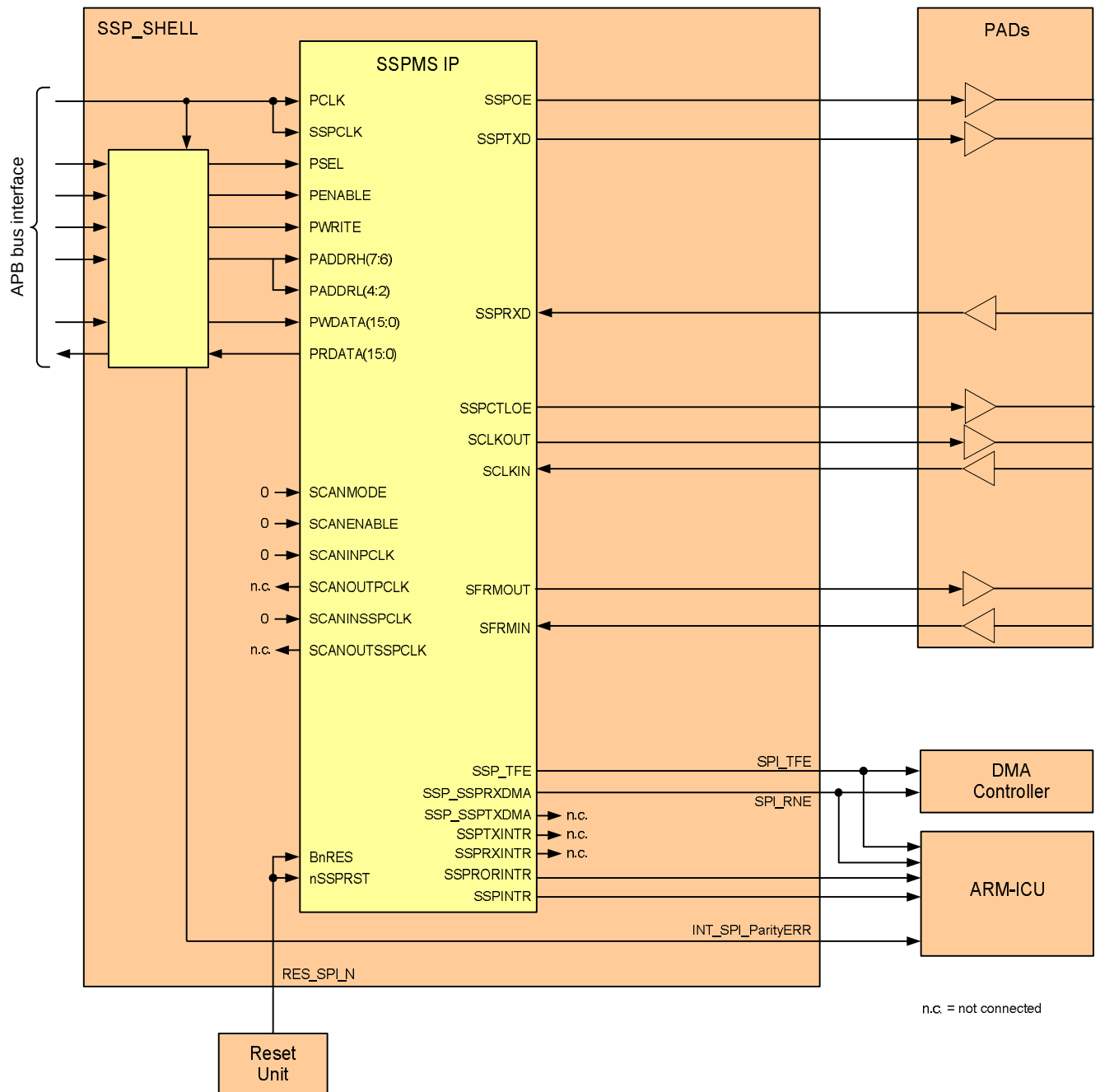
These signals contains the values of the RNE and TFE bits of the SSPSR register.

Note on implementation:

The additional interface signal `SSP_SSPRXDMA` is already available in the SSPMS IP version in the ClearCase directory `/vobs/ip_pool_mc/ssp/hdl`. The additional signal `SSP_TFE` needs to be added to the SSPMS top level file `Ssp.vhd`.

2.3.10.7.3 SSPMS IP Integration

The SSPMS IP is available either over the GPIOs (dedicated) (see xxx) in line with the SCRB register SPI_MODE (see 2.3.10.9.20),



or as MISO/MOSI at the same GPIOs, as shown in the diagram below:

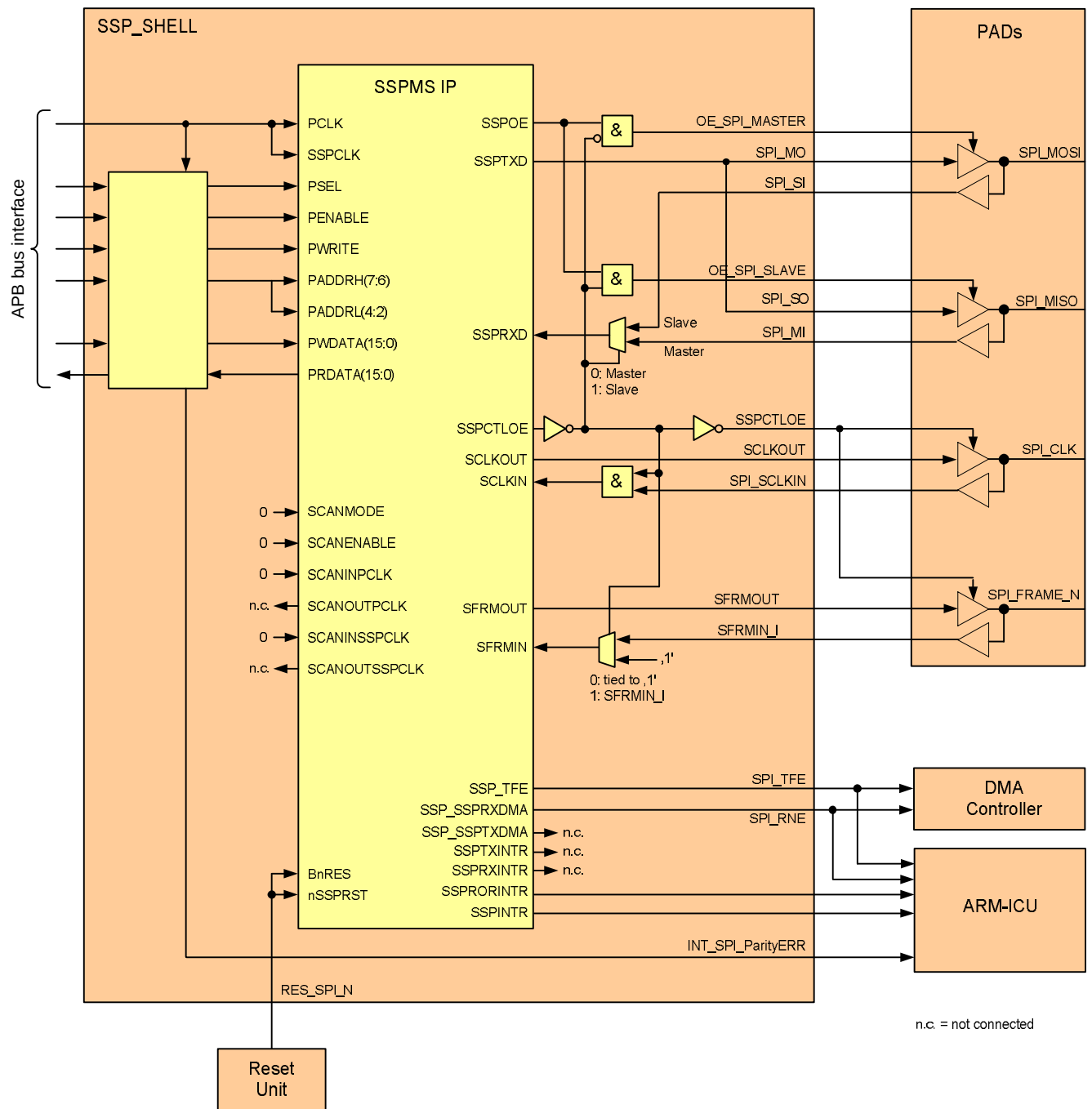


Figure 40: SPI Block Diagram

The following SPI1/2 signals can alternatively be assigned at the GPIOs (alternate functions) in line with SPI_MODE (see 2.3.10.9.20):

SPI1	SPI2	SPI_MODE = DIRECT	SPI_MODE = MISO_MOSI
GPIO16(A), GPIO40(C)	GPIO12(C), GPIO24(A),	SPI_1/2_SCLKOUT_a/_b	SPI1/2_CLK_a/_b
GPIO17(A), GPIO41(C)	GPIO13(C), GPIO25(A),	SPI_1/2_SFRMOUT_a/_b	SPI1/2_FRAME_N_a/_b
GPIO18(A), GPIO42(C)	GPIO14(C), GPIO26(A),	SPI_1/2_SSPTXD_a/_b	SPI1/2_MOSI_a/_b
GPIO19(A), GPIO43(C)	GPIO15(C), GPIO27(A),	SPI_1/2_SSPRXD_a/_b	SPI1/2_MISO_a/_b

Note: If `SPI1/2_FRAME_N_a/_b` is not used, the corresponding alternate function is assigned the blocking value = '0' (see 2.3.10.8.4.2).

Notes on integration

- **Implementation for SPI Interface**

The SSPMS IP does not have a special SPI interface. The IP interface signals SSPOE, SSPTXD, SSPRXD, SSPCTLOE, SCLKOUT, SCLKIN, SFRMOUT, SFRMIN are therefore implemented as SPI signals (`SPI_MASTER_OE_O_N`, `SPI_SLAVE_OE_O_N`, `SPI_OUT_O`, `SPI_S_IN_I`, `SPI_M_IN_I`, `SPI_SSPCTLOE_O_N`, `SPI_SCLKOUT_O`, `SPI_SCLKIN_I`, `SPI_SFRMOUT_O_N`, `SPI_SFRMIN_I_N`) as shown in the diagram above when `SPI_MODE = MISO_MOSI`.

- **Reset**

The SPI-IP can be reset by accessing a specific register. No SPI reset is required for switching from master to slave mode. This function has already been employed in previous projects as the IP has already been used in a number of cases. If a reset is nevertheless required, please remember that this will lead to performance losses (re-initialization).

A switchover from master to slave is required when the ERTEC 200P is connected to another ERTEC 200P over the SPI interface for redundant connection. As the SSPMS IP does not, according to the specifications, support dynamic toggling, it may be necessary first to reset the IP and then set the new operating mode (master or slave).

Operation without `SPI_FRAME_N`:

If the ERTEC 200P is connected as SPI slave to a standard controller (SPI master) that does not support the `SPI_FRAME_N` signal, the ERTEC 200P is to be set as follows:

- | | |
|--|-----|
| Configuration of Motorola format: <code>SSPCR0.FRF = '00'</code> (see 2.3.10.7.7) | AND |
| Configuration of the SPI clock phase: <code>SSPCR0.SPH = '1'</code> (see 2.3.10.7.7) | AND |
| GPIO pin <code>SPI_FRAME_N</code> (see 3.2) | |
| Connect with ext. pull-down | OR |
| Connect with int. pull-down (<code>PULLxx_yyGPIO = "11"</code> , see 2.3.10.9.15.2) | OR |
| Do not select the alternate function (blocking value = '0' is active) | |

Please note the following restrictions in this operating mode:

- In the SPI status register, `SSPSR.BSY` (see 2.3.10.7.7) is not reset at the end of the transfer
- Following a RESET (see 0), "zero data" are transferred when the first character (4-bit...16-bit, `SSPCR0.DSS`) is transferred from the SPI slave to the SPI master (return direction).

Note on communication between SPI master and SPI slave

Communication between SPI master and SPI slave is always in full-duplex mode, i.e. it is not possible for only the master or only the slave to send data. This means for example that when data are only to be transferred from the SPI master to the SPI slave, the master still clocks in the receive signal SSPRXD and enters the corresponding data words in the receive FIFO. If the receive FIFO is not emptied, it overruns after 8 data words at the latest and the receive overrun interrupt SSPRORINTR is triggered if enabled. (The re-

ceive overrun interrupt can be enabled and disabled with the RORIE bit in the SSPCR1 register.)

If data have been transferred from the SPI master to the SPI slave without the receive FIFO of the master being emptied, please note: before a data transfer in the opposite direction, i.e. from the slave to the master, the data from the previous transfer must be cleared from the receive FIFO of the master before the slave can read the "new" data from the receive FIFO of the master.

No new data received will be entered in the receive FIFO if it is full, but data transfer will not automatically be cancelled.

2.3.10.7.4 Send Support with Pause between Individual Characters

In one use case, the SPI transmitters are to transmit blocks of 64 bytes, for example, in a GDMA-compatible process. A set pause (e.g. 200 μ s) must however be maintained between the individual bytes. To ensure this pause, the two GDMA channels required (1 channel for SPI-TxD and 1 channel for SPI-RxD) are controlled by 2 timers.

A transfer record must be entered in the GDMA job list for each byte transfer (total: 64 + 1 transfer entries, see below). Timer 4 (SPI1) and timer 5 (SPI2) are also configured for cyclic, SW-triggered operation with an 1-active timer output signal. A 0 run for a given timer switches the timer output to 1 for one clock cycle in line with the TX-FIFO status (see below) and sets the corresponding 'SPI1/2_SSPTX_Delayed_Request' FF (located in the GDMA shell). The GDMA controller then runs a single-byte transfer (transfer record), generates GDMA acknowledge and resets the 'SPI1/2_SSPTX_Delayed_Request' FF (see 0).

Flow control for generating the timer-triggered GDMA requests can only prevent an SPI TX-FIFO overrun if the SPI TX interrupt status (TX-FIFO half full status, ≥ 4 bytes) is also evaluated. A TX-FIFO overrun is possible if too short a timer cycle time is configured with a short SPI TX transfer rate. The TX interrupt must therefore be enabled during configuration of the SPI-IP ($\rightarrow$ SSPCR1 Register(Bit 1), TIE = 1). This ensures that no characters are lost; certain timer trigger pulses are, if necessary, temporarily ignored.

To ensure equidistance when the above 64-byte transfer is repeated, the timer must be stopped after the 64th single transfer. This function is executed by the GDMA controller as the 65th transfer entry ($\rightarrow$ Write access to TIMER_TRIG_CONTROL_REG register $\rightarrow$ tim_4_clk_en or tim_5_clk_en bit reset) once the 64th single transfer is complete.

Timers 4 and 5 are also used for other functions and the timer output signals 'SPI1/2_SSPTX_Delayed_Request' can therefore also set FFs. The FFs must be specifically reset by the SW. This is done using the 'CLEAR_SPI1/2_SSPTX_Delayed_Request' bits in the 'CLEAR_DMA_SPI_REQ_REG' SCRB register (see 2.3.10.9.22). These register bits are not retentive; they simply activate a reset pulse when 1 is written. There is no reset when a 0 is written, so the 'SPI1/2_SSPTX_Delayed_Request' FFs can be operated separately.

Sequence for above scenario:

- 1) Write GDMA job list to GDMA job list RAM (e.g. 64 single transfers plus timer stop register access)
- 2) Stop timer in progress and then reset 'SPI1/2_SSPTX_Delayed_Request' FFs with write access to 'CLEAR_DMA_SPI_REQ_REG' (SCRB)
- 3) Enter pause + byte transfer time as load value in timer 4 or timer 5 LOAD register
- 4) Configure timer mode register for SW-triggered, cyclic operation (internal gating)

- 5) Start GDMA job (in HW-triggered request mode)
- 6) Start timer with write access to `TIMER_GATE_TRIG_CONTROL_REG` (sets `tim_4(5)_clk_en` and `tim_4(5)_int_gate_trig` bits)

Block transfer is repeated as follows:

- 5) Start GDMA job (in HW-triggered request mode)
- 6) Start timer with write access to `TIMER_GATE_TRIG_CONTROL_REG` (sets `tim_4(5)_clk_en` and `tim_4(5)_int_gate_trig` bits)

2.3.10.7.5 SPI Flash Boot

An SPI can also be used to boot I-TCM and D-TCM (256 KByte in total). This boot process can involve acyclic communication data or application code for rapid execution (e.g. for the digital IO of the ECO PN). To meet fast start-up requirements, booting must be completed within ca. 100 - 150 msec. High-speed SPI flash is therefore required. Figure 41: SPI flash: serial output timing and Figure 42: SPI flash: serial input timing show the timing for a 7-MHz SPI flash (in this case Winbond W25X10/20/40). Table 19: Timing parameters for SPI flash (75 MHz) lists the critical timing parameters.

At a frequency of 25 MHz, booting 256 KByte takes ca. 85 msec.

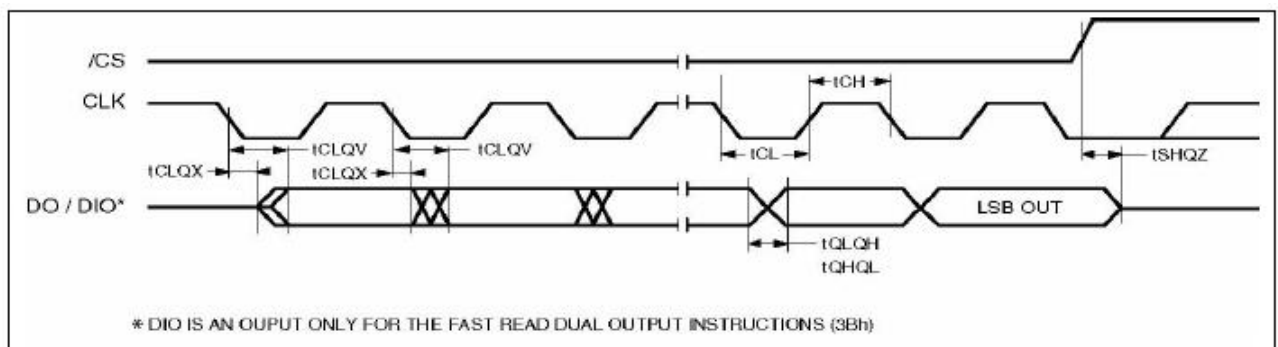


Figure 41: SPI flash: serial output timing

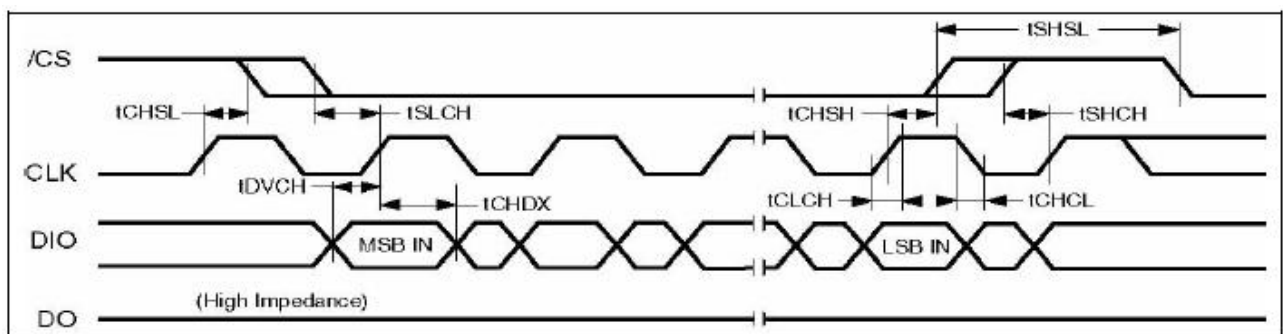


Figure 42: SPI flash: serial input timing

Description	Symbol	Specification	Unit
Clock High / Low Time for Fast Read	t_{CLH}, t_{CLL}	6	ns
Clock Low to Output Valid	t_{CLQV}	6	ns
Output Hold Time	t_{CLOX}	0	ns
Data In Setup Time	t_{DVCH}	2	ns
Data In Hold Time	t_{CHDX}	5	ns

Table 19: Timing parameters for SPI flash (75 MHz)

2.3.10.7.6 Address Mapping

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	14h	SPI	APB

Module	Register/Memory	Read	Write	Address
/SPI				
	SSPCR0	r	w	0h
	SSPCR1	r	w	4h
	SSPDR	rh	w	8h
	SSPSR	r		Ch
	SSPCPSR	r	w	10h
	SSPIIR_SSPICR	rh		14h

2.3.10.7.7 Register Description

Only the registers with addresses 0 – 14h are required for operating the SSPMS IP. The remaining registers (addresses 20h – B0h) are created by mirroring (address bit 5 of the APB bus is not connected to the SSPMS IP) or are only implemented for verification purposes.

Note on read values for unassigned register bits

In the description of the SSPMS IP, the read value of the unassigned register bits is given as "unpredictable". A read value of 0 is, however, implemented (see SspApbif.vhd, assignment to the NextPRDATA signal).

A description of the registers can be found in the SSPMS IP documentation:

The data-bus-width of all specified registers in this module is: 32.

A '0' is read from Software for each not specified Bit in the registers.

The addressing is organized: WORD – wise

Register:	SSPCR0				Address:	0h	
Bits:	15dt0	Reset value:	0h			Attributes:	r w
Description:		SSPCR0 is control register 0 and contains five bit fields that control various functions within the PrimeCell SSPMS.					
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	DSS	0h	r	w	Data size select: 0000 Reserved, undefined operation 0001 Reserved, undefined operation 0010 Reserved, undefined operation		

					0011 4-bit data 0100 5-bit data 0101 6-bit data 0110 7-bit data 0111 8-bit data 1000 9-bit data 1001 10-bit data 1010 11-bit data 1011 12-bit data 1100 13-bit data 1101 14-bit data 1110 15-bit data 1111 16-bit data
5dt4	FRF	0h	r	w	Frame format: 00 Motorola SPI frame format 01 TI synchronous serial frame format 10 National Mikrowire frame format 11 Reserved, undefined operation
6	SPO	0h	r	w	SCLKOUT polarity (applicable to Motorola SPI frame format only).
7	SPH	0h	r	w	SCLKOUT phase (applicable to Motorola SPI frame format only).
15dt8	SCR	00h	r	w	Serial Clock rate. The value SCR s used to generate the transmit and receive bit range of the PrimeCell SSPMS.

Register:		SSPCR1				Address:		4h			
Bits:		6dt0		Reset value:		0h		Attributes:		r	w
Description:			SSPCR1 is the control register 1 and contains five different bit fields, which control various functions within the PrimeCell SSPMS.								
Bit	Identifier			Reset	Attr.		Function / Description				
0	RIE			0h	r	w	Receive FIFO interrupt enable: 0 =Receive FIFO half-full or more condition does not generate the SSPRXINTR interrupt. 1 =Receive FIFO half-full or more condition generates the SSPRXINTR interrupt.				
1	TIE			0h	r	w	Transmit FIFO interrupt enable: 0 =Transmit FIFO half-full or less condition does not generate the SSPTXINTR interrupt. 1 =Transmit FIFO half-full or less condition generates the SSPTXINTR interrupt.				
2	RORIE			0h	r	w	Receive FIFO overrun interrupt enable: 0 =Overrun detection is disabled. Overrun condition does not generate the SSPRORINTR interrupt.				

					Clearing this bit to zero also clears the SSPRORINTR interrupt if it is already asserted. 1 =Overrun detection is enabled. Overrun condition generates the SSPRORINTR interrupt.
3	LBM	0h	r	w	Loop back mode 0 =Normal serial port operation enabled 1 =Output of transmit serial shifter is connected to input of receive serial shifter internally.
4	SSE	0h	r	w	Synchronous serial port enable: 0 =SSP operation disabled 1 = operation enabled
5	MS	0h	r	w	Master/slave mode select. This bit can be modified only when the PrimeCell SSPMS is disabled (SSE =0). 0 =Device configured as master (default). 1 =Device configured as slave.
6	SOD	0h	r	w	Slave mode output disable.This bis is relevant only in the slave mode (MS =1). In multiple-slave systems, it is possible for an SSPMS master to broadcast a message to all slaves in the system while ensuring that only the slave drives data onto ist serial output line. In such systems the RXD lines from multiple slaves could be tied together. To operate in such systems, the SOD may be set if the SSP slave is not supposed to drive the SSPTXD line. 0 =SSP may drive the SSPTXD output in slave mode 1 =SSP must not drive the SSPTXD output in slave mode.

Register:	SSPDR			Address:	8h	
Bits:	15dt0	Reset value:	0000h	Attributes:	rh	w
Description:	Transmit/Receive FIFO (additional wrapper for bit endianness and parity bit)					

Register:	SSPSR			Address:	Ch	
Bits:	4dt0	Reset value:	3h	Attributes:	r	
Description:	SSPSR is a read-only status register wich contains bits that indicate the FIFO fill status and the SSP busy status.					
Bit	Identifier	Reset	Attr.	Function / Description		
0	TFE	1h	r	Transmit FIFO empty		
1	TNF	1h	r	Transmit FIFO not full		
2	RNE	0h	r	Receive FIFO not empty		

3	RFF	0h	r		Receive FIFO full
4	BSY	0h	r		SSP busy flag

Register:	SSPCPSR				Address:	10h	
Bits:	7dt0	Reset value:	00h		Attributes:	r	w
Description:	Clock prescale divisor						

Register:	SSPIIR_SSPICR					Address:	14h	
Bits:	2dt0	Reset value:	0h			Attributes:	rh	
Description:		The interrupt status is read from the SSP interrupt identification register (SSPIIR)						
Bit	Identifier	Reset	Attr.		Function / Description			
0	RIS	0h	rh		Read: SSP transmit FIFO service request interrupt status			
1	TIS	0h	rh		Read: SSP Receive FIFO service request interrupt status			
2	RORIS	0h	rh		Read: SSP Receive FIFO overrun interrupt status			

2.3.10.8 GPIO (General Purpose Input / Output)

There are a maximum of 96 general purpose input/outputs (GPIOs). They are divided into 2 blocks:

- GPIO31-0: 32-bit (available as default GPIO port)
- GPIO95-32: 64-bit (available as an alternative to XHIF)

GPIO31-0 is available as a default block and is multiplexed with the interface signals from the PNPLL, UART2/3, SPI1/2, I²C_1_3, timer 0 – 5 and ARM926 watchdog . Following a reset, this GPIO block is selected and connected to all inputs. The alternate functions are set by the SW using the GPIO registers (see 2.3.10.8.5).

The internal MII debug interface (internal PHYs) is also multiplexed to GPIO31-0. This function is set upon an XRESET PowerOn reset with the configuration pins CONFIG(6-3) (see 2.3.10.9.3) or by the SW in the SCRB register 'Config-REG(6-3)' (see 2.3.10.9.22).

Rules for the SW: The default values of the alternate functions **must not** be changed if the internal MII debug interfaces are selected at the corresponding pins with the configuration pins.

The GPIO15-0 inputs are also connected to an I-filter (see 2.3.10.1) and to the ARM interrupt controller (IRQ32-47, see 2.3.2.14). The polarity of these signals to the ARM interrupt controller can be set in the GPIO_INT_POLSEL register (see 2.3.10.9.22). In line with the polarity of the external signals, this ensures a high-active interrupt level.

The GPIO7-4 inputs are also connected to PN-ICU 1 over an I-filter (see 2.3.10.1). The polarity of these signals to the PN-ICU 1 can be set in the GPIO_INT_POLSEL register (see 2.3.10.9.22). In line with the polarity of the external signals, this ensures a high-active interrupt level.

The GPIO3-0 inputs are connected to an I-filter (see 2.3.10.1) and can control HW jobs with GDMA (see 2.3.4.2.1.1). The polarity of these signals to the GDMA can be set in the GPIO_INT_POLSEL register (see 2.3.10.9.22). A high-active level to the GDMA must always be set. As these input signals are not synchronized with the GDMA, "Synchronize" mode (mode 2) must be set selectively in the I-filter for these inputs.

GPIO31-0 have integrated pulls that can be selected by the SW (see 2.3.10.9.15). These pulls have a default assignment following a reset (see corresponding Pull31_0GPIO register in the SCRB, 2.3.10.9.22). In GPIO11-0, the pulls are disabled by default as the alternate functions mean that there is no suitable setting for the application.

GPIO95-32 is completely unavailable if the parallel 32-bit host interface (XHIF) is used. If XHIF is operated in 16-bit mode, there are 18 GPIOs remaining. If the external host is coupled over the serial host interface, there are 57 GPIOs remaining (XHIF_XIRQ is required as well as the serial interface). GPIO95-32 is multiplexed with the interface signals from the local IO (64-bit parallel, 2x serial SPI master), SPI1, UART1/4, ARM926 watchdog, I²C_1_2 and PNPLL. These alternate functions are set by the SW using the GPIO registers (see 2.3.10.8.5).

The MII/GMII interfaces (if external PHYs are used) and the ARM926 trace port are also multiplexed to the GPIO95-32. These functions and the corresponding host interface enable are set upon an XRESET PowerOn reset with the configuration pins CONFIG(6-3) (see 2.3.10.9.3) or by the SW in the SCRB register 'Config-REG(6-3)' (see 2.3.10.9.22).

Rules for the SW: The default values of the alternate functions **must not** be changed if the XHIF interface, the MII/GMII interfaces or the ARM926 trace port is selected at the corresponding pins with the configuration pins.

GPIO95-32 have integrated pulls that can be selected by the SW (see 2.3.10.9.15). These pulls have a default assignment following a reset (see corresponding Pull95_32GPIO registers in the SCRB, 2.3.10.9.22). GPIO55-54 (XHIF_IRQ, XHIF_XRDY) have **no** internal pulls.

2.3.10.8.1 GPIO Module Function Description

Figure 43 is a block diagram of a GPIO module.

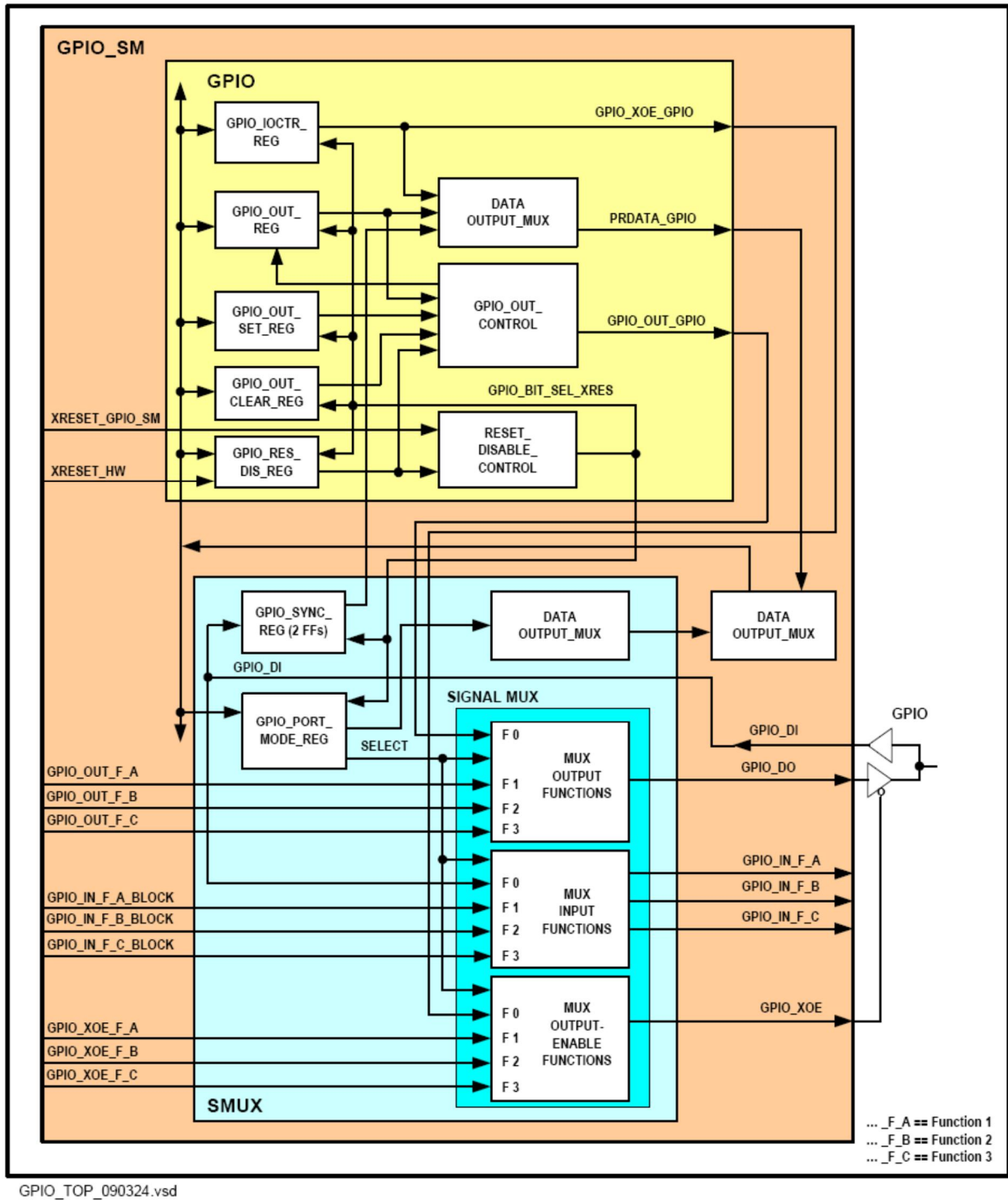


Figure 43: Block diagram of a GPIO module

3 GPIO modules with 32 GPIOs each are used for the GPIOs. These GPIO modules have the following properties:

- The number of general purpose input/outputs (GPIOs) is configurable with the hardware.
- Each GPIO can be programmable as input/output (GPIO function).
- Each GPIO input (pin) can be read by the SW.

- Each GPIO output can be set and reset by the SW (bit-selective programming is possible).
- The GPIO function is multiplexed with up to 3 alternative functions (function A-C).
- 2 reset signals affect the GPIO registers (XRESET_HW as asynchronous reset and XRESET_GPIO_SM as selective reset).
- Most GPIO pins have integrated pull-ups / pull-downs to prevent floating.
- The GPIO pins can be connected over an integrated filter (see X5.5.2X) (required for local IOs, external interrupts, alarms, etc.)
- Up to 16 GPIOs (GPIO15-0) can trigger interrupts at the ARM processor and are connected to the ARM interrupt controller for this purpose (IRQ32-47) (see 2.3.2.14). The polarity of these signals to the ARM-ICU (ARM interrupt controller) can be set in the GPIO_INT_POLSEL register (see 2.3.10.9.22). In line with the polarity of the external signal, this ensures a high-active interrupt level. The signals must be at least 2 clocks (125 MHz) long.
- Up to 6 GPIOs can be used as the gate / trigger signal for the internal timers.
- Up to 4 GPIOs (GPIO3-0) can control HW jobs with GDMA (see 2.3.4.2.1.1). The polarity of these signals to the GDMA can be set in the GPIO_INT_POLSEL register (see 2.3.10.9.22). A high-active level to the GDMA must always be set.

Using the GPIO module, either one of three signals or the value of the corresponding bit in the GPIO_OUT_0 register can be connected to each GPIO pin ("*GPIO function*"). The GPIO_PORT_MODE_L/H_0 registers define what is connected to the GPIO pins.

If a signal is connected to a GPIO pin, a corresponding output enable signal specifies whether the pin is the input or output. If, on the other hand, the GPIO function is used for a pin, this is specified with the GPIO_IOCTL_0 register.

With the GPIO function, the output value of a GPIO pin can be defined in three different ways:

- By writing the corresponding bit in the GPIO_OUT_0 register,
- By writing a '1' to the corresponding bit in the GPIO_OUT_SET_0 register
(=> The pin is set to 1.)
- By writing a '1' to the corresponding bit in the GPIO_OUT_CLEAR_0 register
(=> The pin is set to 0.)

The values can be read at the GPIO pins over the GPIO_IN_0 register.

2.3.10.8.2 Integration of the GPIO Module

The GPIO module connects the top level modules to the GPIO pins. It is also connected to the APB bus. The GPIO module is operated with the APB clock.

The feature of an internal GPIO "reset disable" is activated in ERTEC 200P (module input: XRESET_HW connected to reset signal). Alongside the global reset (XRESET_GPIO_SM), a selective reset in GPIO_SM is also possible.

The connection of the internal modules to the GPIO module (GPIO_XOE_F_A/B/C, GPIO_OUT_F_A/B/C, GPIO_IN_F_A/B/C, GPIO_IN_F_A/B/C_BLOCK signals) is described in the corresponding modules and in *GPIO assignment*.

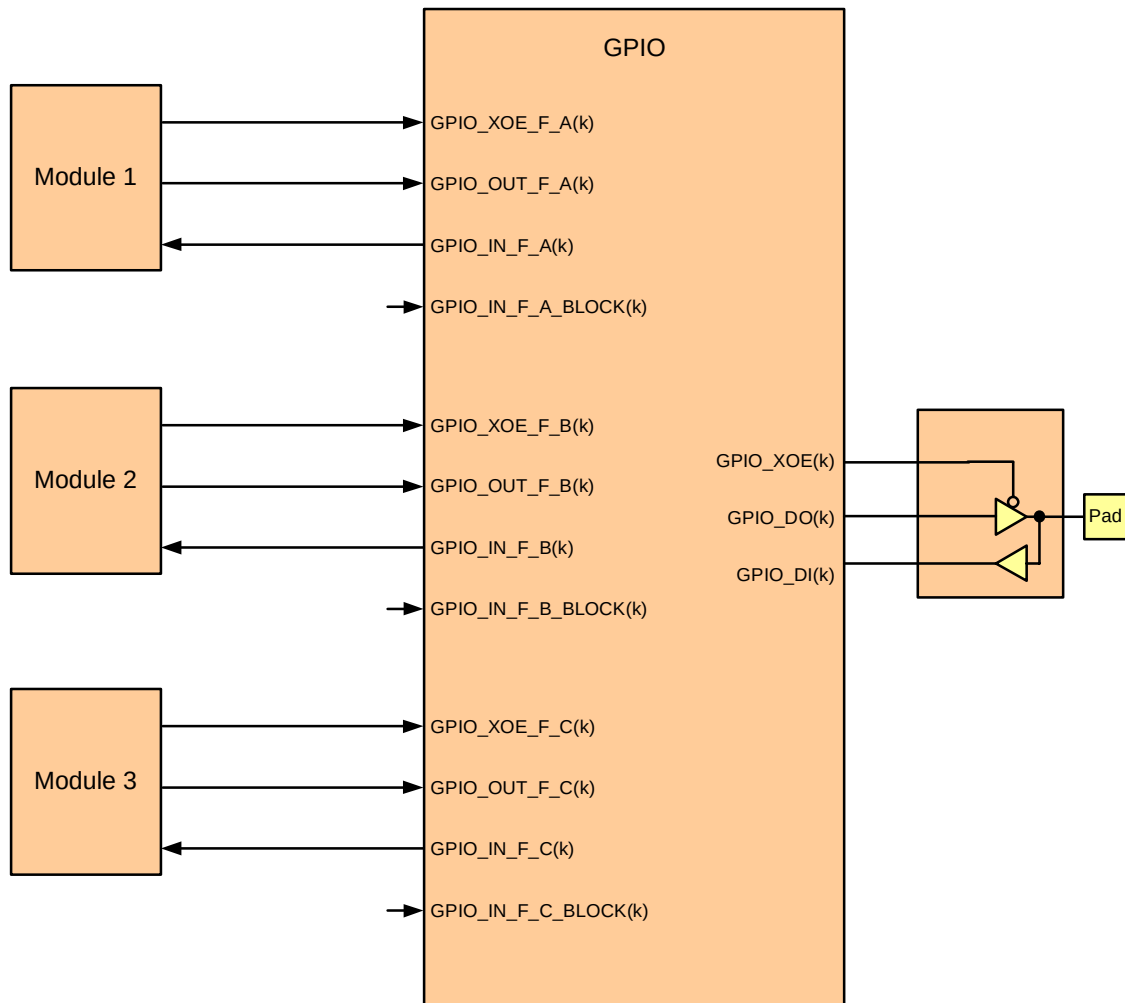


Figure 44: GPIO, IO Circuit

2.3.10.8.3 GPIO Pad Multiplexing

Alongside the alternate functions (A-C) integrated in the GPIO modules, the **GPIO95-32** block has a number of other alternate functions that are set with the configuration pins CONFIG(6-3) (see 2.3.10.9.3) or the SCRB register 'CONFIG_REG' (see 2.3.10.9.22, host interface, MII/GMII interface, ARM926 trace port).

Alongside the alternate functions (A-C) integrated in the GPIO module, the **GPIO31-0** block has another alternate function that is set with the configuration pins CONFIG(6-3) (see 2.3.10.9.3) or the SCRB register 'CONFIG_REG' (see 2.3.10.9.22, PHY Debugging).

2.3.10.8.4 GPIO Assignment

2.3.10.8.4.1 IO Multiplexing

For all pins designated GPIO.., the software can connect the alternate functions (A – C) downstream (see Table 20: Overview of alternate functions (A – C)). All other pins have fixed coding set by the function groups. TRACEPKT outputs of the trace port at GPIO53-38 (when CONFIG6:3 = 1110, function 7) that are not used by the debugger remain high-impedance.

2.3.10.8.4.2 GPIO Alternate Function

Table 20 lists alternate functions A - C that can be set for the GPIO pins by the SW. Some alternate function inputs are driven by two different GPIO pads (inputs with "_a/_b" versions). The GPIO alternate functions set which signal is assigned to each GPIO pad. In the case of input signals with _a/_b versions, the user must however ensure that **only one of the two versions** is activated over the GPIO alternate functions.

GPIO Pin	GPIO Registers	GPIO Alternative Function A	GPIO Alternative Function B	GPIO Alternative Function C
GPIO0_INT	GPIO * 0.0	PNPLL_OUT0_a	TIM_OUT0_a	NOT_USED
GPIO1_INT	GPIO * 0.1	PNPLL_OUT1_a	TIM_OUT1_a	NOT_USED
GPIO2_INT	GPIO * 0.2	PNPLL_OUT2_a	TIM_OUT2	DBGREQ
GPIO3_INT	GPIO * 0.3	PNPLL_OUT3_a	TIM_OUT3	DBGACK
GPIO4_INT	GPIO * 0.4	PNPLL_OUT4_a	TIM_OUT4	I2C_SDO1_1_a
GPIO5_INT	GPIO * 0.5	PNPLL_OUT5_a	TIM_OUT5	I2C_SCLK_1_a
GPIO6_INT	GPIO * 0.6	PNPLL_OUT6_a	TIM_TRIG0_a	I2C_SDO1_2_a
GPIO7_INT	GPIO * 0.7	PNPLL_OUT7_a	TIM_TRIG1_a	I2C_SCLK_2_a
GPIO8_INT	GPIO * 0.8	PNPLL_OUT8_a	TIM_TRIG2	SPI_2_SCLKIN_b
GPIO9_INT	GPIO * 0.9	PNCLKA_IN_a	TIM_TRIG3	SPI_2_SFRMIN_b
GPIO10_INT	GPIO * 0.10	NOT_USED	TIM_TRIG4	SPI_2_SSPOE_b
GPIO11_INT	GPIO * 0.11	PNTIME_OUT_a	TIM_TRIG5	SPI_2_SSPCTLOE_b
GPIO12_INT	GPIO * 0.12	PNTIME_IN	U2_CTS	SPI_2_SCLKOUT_b
GPIO13_INT	GPIO * 0.13	WD_XWDOUT0_a	U2_RTS	SPI_2_SFRMOUT_b
GPIO14_INT	GPIO * 0.14	I2C_SCLK_3	U2_TXD	SPI_2_SSPTXD_b
GPIO15_INT	GPIO * 0.15	I2C_SDO1_3	U2_RXD	SPI_2_SSPRXD_b
GPIO16	GPIO * 0.16	SPI_1_SCLKOUT_a	NOT_USED	NOT_USED
GPIO17	GPIO * 0.17	SPI_1_SFRMOUT_a	NOT_USED	NOT_USED
GPIO18	GPIO * 0.18	SPI_1_SSPTXD_a	NOT_USED	NOT_USED
GPIO19	GPIO * 0.19	SPI_1_SSPRXD_a	NOT_USED	NOT_USED
GPIO20	GPIO * 0.20	SPI_1_SSPOE_a	NOT_USED	NOT_USED
GPIO21	GPIO * 0.21	SPI_1_SSPCTLOE_a	NOT_USED	NOT_USED
GPIO22	GPIO * 0.22	SPI_1_SFRMIN_a	NOT_USED	NOT_USED
GPIO23	GPIO * 0.23	SPI_1_SCLKIN_a	PNCLKA_IN_b	NOT_USED
GPIO24	GPIO * 0.24	SPI_2_SCLKOUT_a	TIM_OUT0_b	NOT_USED
GPIO25	GPIO * 0.25	SPI_2_SFRMOUT_a	TIM_OUT1_b	NOT_USED
GPIO26	GPIO * 0.26	SPI_2_SSPTXD_a	TIM_TRIG0_b	NOT_USED
GPIO27	GPIO * 0.27	SPI_2_SSPRXD_a	TIM_TRIG1_b	NOT_USED
GPIO28	GPIO * 0.28	SPI_2_SCLKIN_a	U3_TXD	PNTIME_OUT_b
GPIO29	GPIO * 0.29	SPI_2_SFRMIN_a	U3_RXD	PNPLL_OUT6_b
GPIO30	GPIO * 0.30	SPI_2_SSPOE_a	NOT_USED	PNPLL_OUT7_b
GPIO31	GPIO * 0.31	SPI_2_SSPCTLOE_a	NOT_USED	PNPLL_OUT8_b

GPIO Pin	GPIO Registers	GPIO Alternative Function A	GPIO Alternative Function B	GPIO Alternative Function C
XHIF_A1	GPIO * 1.0	NOT_USED	NOT_USED	NOT_USED
XHIF_A2	GPIO * 1.1	NOT_USED	NOT_USED	NOT_USED
XHIF_A3	GPIO * 1.2	NOT_USED	NOT_USED	NOT_USED
XHIF_A4	GPIO * 1.3	NOT_USED	NOT_USED	NOT_USED
XHIF_A5	GPIO * 1.4	NOT_USED	NOT_USED	NOT_USED
XHIF_A6	GPIO * 1.5	NOT_USED	NOT_USED	NOT_USED
XHIF_A7	GPIO * 1.6	NOT_USED	NOT_USED	NOT_USED
XHIF_A8	GPIO * 1.7	NOT_USED	NOT_USED	NOT_USED
XHIF_A9	GPIO * 1.8	NOT_USED	NOT_USED	SPI_1_SCLKOUT_b
XHIF_A10	GPIO * 1.9	NOT_USED	NOT_USED	SPI_1_SFRMOUT_b
XHIF_A11	GPIO * 1.10	NOT_USED	NOT_USED	SPI_1_SSPTXD_b
XHIF_A12	GPIO * 1.11	NOT_USED	NOT_USED	SPI_1_SSPRXD_b
XHIF_A13	GPIO * 1.12	NOT_USED	NOT_USED	SPI_1_SSPOE_b
XHIF_A14	GPIO * 1.13	NOT_USED	NOT_USED	SPI_1_SSPCTLOE_b
XHIF_A15	GPIO * 1.14	NOT_USED	NOT_USED	SPI_1_SCLKIN_b
XHIF_A16	GPIO * 1.15	NOT_USED	NOT_USED	SPI_1_SFRMIN_b
XHIF_A17	GPIO * 1.16	NOT_USED	NOT_USED	U1_CTS
XHIF_A18	GPIO * 1.17	NOT_USED	NOT_USED	U1_DCD
XHIF_A19	GPIO * 1.18	NOT_USED	NOT_USED	U1_DSR
XHIF_SEG_2	GPIO * 1.19	NOT_USED	NOT_USED	U1_RI
XHIF_SEG_0	GPIO * 1.20	NOT_USED	NOT_USED	U1_RTS
XHIF_SEG_1	GPIO * 1.21	NOT_USED	NOT_USED	U1_DTR
XHIF_XRDY	GPIO * 1.22	NOT_USED	NOT_USED	U4_TXD
XHIF_XRQ	GPIO * 1.23	NOT_USED	NOT_USED	U4_RXD
XHIF_XWR	GPIO * 1.24	NOT_USED	NOT_USED	PNPLL_OUT0_b
XHIF_XRD	GPIO * 1.25	NOT_USED	NOT_USED	PNPLL_OUT1_b
XHIF_XCS_R_A20	GPIO * 1.26	NOT_USED	NOT_USED	PNPLL_OUT2_b
XHIF_XCS_M	GPIO * 1.27	NOT_USED	NOT_USED	PNPLL_OUT3_b
XHIF_XBE0	GPIO * 1.28	NOT_USED	NOT_USED	PNPLL_OUT4_b
XHIF_XBE1	GPIO * 1.29	NOT_USED	NOT_USED	PNPLL_OUT5_b
XHIF_XBE2	GPIO * 1.30	NOT_USED	NOT_USED	U1_TXD

GPIO Pin	GPIO Registers	GPIO Alternative Function A	GPIO Alternative Function B	GPIO Alternative Function C
XHIF_XBE3	GPIO * 1.31	NOT_USED	NOT_USED	U1_RXD
XHIF_D0	GPIO * 2.0	NOT_USED	NOT_USED	NOT_USED
XHIF_D1	GPIO * 2.1	NOT_USED	NOT_USED	NOT_USED
XHIF_D2	GPIO * 2.2	NOT_USED	NOT_USED	NOT_USED
XHIF_D3	GPIO * 2.3	NOT_USED	NOT_USED	NOT_USED
XHIF_D4	GPIO * 2.4	NOT_USED	NOT_USED	NOT_USED
XHIF_D5	GPIO * 2.5	NOT_USED	NOT_USED	NOT_USED
XHIF_D6	GPIO * 2.6	NOT_USED	NOT_USED	NOT_USED
XHIF_D7	GPIO * 2.7	NOT_USED	NOT_USED	XSSI_DATA
XHIF_D8	GPIO * 2.8	NOT_USED	NOT_USED	SSI_CLK
XHIF_D9	GPIO * 2.9	NOT_USED	NOT_USED	SSI_DATA
XHIF_D10	GPIO * 2.10	NOT_USED	NOT_USED	NOT_USED
XHIF_D11	GPIO * 2.11	NOT_USED	NOT_USED	WD_XWDOUT0_b
XHIF_D12	GPIO * 2.12	NOT_USED	NOT_USED	NOT_USED
XHIF_D13	GPIO * 2.13	NOT_USED	NOT_USED	NOT_USED
XHIF_D14	GPIO * 2.14	NOT_USED	NOT_USED	NOT_USED
XHIF_D15	GPIO * 2.15	NOT_USED	NOT_USED	NOT_USED
XHIF_D16	GPIO * 2.16	NOT_USED	NOT_USED	I2C_SDO1_1_b
XHIF_D17	GPIO * 2.17	NOT_USED	NOT_USED	I2C_SCLK_1_b
XHIF_D18	GPIO * 2.18	NOT_USED	NOT_USED	I2C_SDO1_2_b
XHIF_D19	GPIO * 2.19	NOT_USED	NOT_USED	I2C_SCLK_2_b
XHIF_D20	GPIO * 2.20	NOT_USED	NOT_USED	NOT_USED
XHIF_D21	GPIO * 2.21	NOT_USED	NOT_USED	NOT_USED
XHIF_D22	GPIO * 2.22	NOT_USED	NOT_USED	NOT_USED
XHIF_D23	GPIO * 2.23	NOT_USED	NOT_USED	NOT_USED
XHIF_D24	GPIO * 2.24	NOT_USED	NOT_USED	NOT_USED
XHIF_D25	GPIO * 2.25	NOT_USED	NOT_USED	NOT_USED
XHIF_D26	GPIO * 2.26	NOT_USED	NOT_USED	NOT_USED
XHIF_D27	GPIO * 2.27	NOT_USED	NOT_USED	NOT_USED
XHIF_D28	GPIO * 2.28	NOT_USED	NOT_USED	NOT_USED
XHIF_D29	GPIO * 2.29	NOT_USED	NOT_USED	NOT_USED
XHIF_D30	GPIO * 2.30	NOT_USED	NOT_USED	NOT_USED
XHIF_D31	GPIO * 2.31	NOT_USED	NOT_USED	NOT_USED

Table 20: Overview of alternate functions (A – C)

2.3.10.8.5 Address Mapping

Start Address	End Address	Modul/Memory Name	Interface	Fill Mode
0h	5Ch	gpio	APB	

Module	Register/Memory	Read	Write	Address	Revision
/gpio					
	GPIO_IOCTL_0	r	w	0h	
	GPIO_OUT_0	rh	w	4h	
	GPIO_OUT_SET_0	rh	wt	8h	
	GPIO_OUT_CLEAR_0	rh	wt	Ch	
	GPIO_RES_DIS_0	r	w	10h	
	GPIO_IN_0	rh		14h	
	GPIO_PORT_MODE_0_L	r	w	18h	
	GPIO_PORT_MODE_0_H	r	w	1Ch	
	GPIO_IOCTL_1	r	w	20h	
	GPIO_OUT_1	rh	w	24h	
	GPIO_OUT_SET_1	rh	wt	28h	
	GPIO_OUT_CLEAR_1	rh	wt	2Ch	
	GPIO_RES_DIS_1	r	w	30h	
	GPIO_IN_1	rh		34h	
	GPIO_PORT_MODE_1_L	r	w	38h	
	GPIO_PORT_MODE_1_H	r	w	3Ch	
	GPIO_IOCTL_2	r	w	40h	
	GPIO_OUT_2	rh	w	44h	
	GPIO_OUT_SET_2	rh	wt	48h	
	GPIO_OUT_CLEAR_2	rh	wt	4Ch	
	GPIO_RES_DIS_2	r	w	50h	
	GPIO_IN_2	rh		54h	
	GPIO_PORT_MODE_2_L	r	w	58h	
	GPIO_PORT_MODE_2_H	r	w	5Ch	

2.3.10.8.6 Register Description

The data-bus-width of all specified registers in this module is: 32.

A '0' is read from Software for each not specified Bit in the registers.

The addressing is organized: WORD – wise

Module: /gpio

Register:	GPIO_IOCTL_0			Address:	0h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:	Configuration register for General Purpose IOs (31:0) 0: GPIOx = output, 1: GPIOx = input					

Register:	GPIO_OUT_0			Address:	4h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	w
Description:	Output register for General Purpose IOs (31:0) 0: GPIO outputx = 0, 1: GPIO outputx = 1					

Register:	GPIO_OUT_SET_0			Address:	8h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	wt
Description:	Bit-selective setting of the output register for General Purpose IOs (31:0) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 1 Read always returns 0					

Register:	GPIO_OUT_CLEAR_0			Address:	Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	wt
Description:	Bit-selective reset of the output register for General Purpose IOs (31:0) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 0 Read always returns 0					

Register:	GPIO_RES_DIS_0			Address:	10h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Bit-selective reset disabling of XRESET_GPIO_SM signal for registers of General Purpose IOs (31:0)					

	For writing: 0: XRESET_GPIO_SM resets the corresponding register bit of all registers except of GPIO_RES_DIS 1: XRESET_GPIO_SM has no effect on the corresponding register bit
--	--

Register:	GPIO_IN_0			Address:	14h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Input register for General Purpose IOs (31:0)					

Register:	GPIO_PORT_MODE_0_L			Address:	18h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Configuration register for GPIO port (15:0) Function assignment: 00 =GPIO function 01 =alternate function A 10 =alternate function B 11 =alternate function C					
Bit	Identifier	Reset	Attr.		Function / Description	
1dt0	GPIO_0_MODE_0_L	0h	r	w	Port GPIO(0)	
3dt2	GPIO_1_MODE_0_L	0h	r	w	Port GPIO(1)	
5dt4	GPIO_2_MODE_0_L	0h	r	w	Port GPIO(2)	
7dt6	GPIO_3_MODE_0_L	0h	r	w	Port GPIO(3)	
9dt8	GPIO_4_MODE_0_L	0h	r	w	Port GPIO(4)	
11dt10	GPIO_5_MODE_0_L	0h	r	w	Port GPIO(5)	
13dt12	GPIO_6_MODE_0_L	0h	r	w	Port GPIO(6)	
15dt14	GPIO_7_MODE_0_L	0h	r	w	Port GPIO(7)	
17dt16	GPIO_8_MODE_0_L	0h	r	w	Port GPIO(8)	
19dt18	GPIO_9_MODE_0_L	0h	r	w	Port GPIO(9)	
21dt20	GPIO_10_MODE_0_L	0h	r	w	Port GPIO(10)	
23dt22	GPIO_11_MODE_0_L	0h	r	w	Port GPIO(11)	
25dt24	GPIO_12_MODE_0_L	0h	r	w	Port GPIO(12)	
27dt26	GPIO_13_MODE_0_L	0h	r	w	Port GPIO(13)	
29dt28	GPIO_14_MODE_0_L	0h	r	w	Port GPIO(14)	
31dt30	GPIO_15_MODE_0_L	0h	r	w	Port GPIO(15)	

Register:	GPIO_PORT_MODE_0_H				Address:	1Ch	
Bits:	31dt0		Reset value:		00000000h		Attributes: r w
Description:		Configuration register for GPIO port (31:16) Function assignment: 00 =GPIO function 01 =alternate function A 10 =alternate function B 11 =alternate function C					
Bit	Identifier		Reset	Attr.		Function / Description	
1dt0	GPIO_16_MODE_0_H		0h	r	w	Port GPIO(16)	
3dt2	GPIO_17_MODE_0_H		0h	r	w	Port GPIO(17)	
5dt4	GPIO_18_MODE_0_H		0h	r	w	Port GPIO(18)	
7dt6	GPIO_19_MODE_0_H		0h	r	w	Port GPIO(19)	
9dt8	GPIO_20_MODE_0_H		0h	r	w	Port GPIO(20)	
11dt10	GPIO_21_MODE_0_H		0h	r	w	Port GPIO(21)	
13dt12	GPIO_22_MODE_0_H		0h	r	w	Port GPIO(22)	
15dt14	GPIO_23_MODE_0_H		0h	r	w	Port GPIO(23)	
17dt16	GPIO_24_MODE_0_H		0h	r	w	Port GPIO(24)	
19dt18	GPIO_25_MODE_0_H		0h	r	w	Port GPIO(25)	
21dt20	GPIO_26_MODE_0_H		0h	r	w	Port GPIO(26)	
23dt22	GPIO_27_MODE_0_H		0h	r	w	Port GPIO(27)	
25dt24	GPIO_28_MODE_0_H		0h	r	w	Port GPIO(28)	
27dt26	GPIO_29_MODE_0_H		0h	r	w	Port GPIO(29)	
29dt28	GPIO_30_MODE_0_H		0h	r	w	Port GPIO(30)	
31dt30	GPIO_31_MODE_0_H		0h	r	w	Port GPIO(31)	

Register:	GPIO_IOCTL_1				Address:	20h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w	
Description:	Configuration register for General Purpose IOs (63:32) 0: GPIOx = output, 1: GPIOx = input						

Register:	GPIO_OUT_1				Address:	24h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	rh	w
Description:	Output register for General Purpose IOs (63:32) 0: GPIO outputx = 0, 1: GPIO outputx = 1						

Register:	GPIO_OUT_SET_1			Address:	28h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	wt
Description:	Bit-selective setting of the output register for General Purpose IOs (63:32) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 1 Read always returns 0					

Register:	GPIO_OUT_CLEAR_1			Address:	2Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	wt
Description:	Bit-selective reset of the output register for General Purpose IOs (63:32) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 0 Read always returns 0					

Register:	GPIO_RES_DIS_1			Address:	30h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Bit-selective reset disabling of XRESET_GPIO_SM signal for registers of General Purpose IOs (31:0) For writing: 0: XRESET_GPIO_SM resets the corresponding register bit of all registers except of GPIO_RES_DIS 1: XRESET_GPIO_SM has no effect on the corresponding register bit					

Register:	GPIO_IN_1			Address:	34h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Input register for General Purpose IOs (63:32) 0: GPIO inputx = 0, 1: GPIO inputx = 1					

Register:	GPIO_PORT_MODE_1_L			Address:	38h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Configuration register for GPIO port (47:32) Function assignment: 00 =GPIO function 01 =alternate function A 10 =alternate function B					

11 =alternate function C					
Bit	Identifier	Reset	Attr.		Function / Description
1dt0	GPIO_32_MODE_1_L	0h	r	w	Port GPIO(32)
3dt2	GPIO_33_MODE_1_L	0h	r	w	Port GPIO(33)
5dt4	GPIO_34_MODE_1_L	0h	r	w	Port GPIO(34)
7dt6	GPIO_35_MODE_1_L	0h	r	w	Port GPIO(35)
9dt8	GPIO_36_MODE_1_L	0h	r	w	Port GPIO(36)
11dt10	GPIO_37_MODE_1_L	0h	r	w	Port GPIO(37)
13dt12	GPIO_38_MODE_1_L	0h	r	w	Port GPIO(38)
15dt14	GPIO_39_MODE_1_L	0h	r	w	Port GPIO(39)
17dt16	GPIO_40_MODE_1_L	0h	r	w	Port GPIO(40)
19dt18	GPIO_41_MODE_1_L	0h	r	w	Port GPIO(41)
21dt20	GPIO_42_MODE_1_L	0h	r	w	Port GPIO(42)
23dt22	GPIO_43_MODE_1_L	0h	r	w	Port GPIO(43)
25dt24	GPIO_44_MODE_1_L	0h	r	w	Port GPIO(44)
27dt26	GPIO_45_MODE_1_L	0h	r	w	Port GPIO(45)
29dt28	GPIO_46_MODE_1_L	0h	r	w	Port GPIO(46)
31dt30	GPIO_47_MODE_1_L	0h	r	w	Port GPIO(47)

Register:	GPIO_PORT_MODE_1_H				Address:	3Ch	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r w
Description:		Configuration register for GPIO port (63:48) Function assignment: 00 =GPIO function 01 =alternate function A 10 =alternate function B 11 =alternate function C					
Bit	Identifier		Reset	Attr.		Function / Description	
1dt0	GPIO_48_MODE_1_H		0h	r	w	Port GPIO(48)	
3dt2	GPIO_49_MODE_1_H		0h	r	w	Port GPIO(49)	
5dt4	GPIO_50_MODE_1_H		0h	r	w	Port GPIO(50)	
7dt6	GPIO_51_MODE_1_H		0h	r	w	Port GPIO(51)	
9dt8	GPIO_52_MODE_1_H		0h	r	w	Port GPIO(52)	
11dt10	GPIO_53_MODE_1_H		0h	r	w	Port GPIO(53)	
13dt12	GPIO_54_MODE_1_H		0h	r	w	Port GPIO(54)	

15dt14	GPIO_55_MODE_1_H	0h	r	w	Port GPIO(55)
17dt16	GPIO_56_MODE_1_H	0h	r	w	Port GPIO(56)
19dt18	GPIO_57_MODE_1_H	0h	r	w	Port GPIO(57)
21dt20	GPIO_58_MODE_1_H	0h	r	w	Port GPIO(58)
23dt22	GPIO_59_MODE_1_H	0h	r	w	Port GPIO(59)
25dt24	GPIO_60_MODE_1_H	0h	r	w	Port GPIO(60)
27dt26	GPIO_61_MODE_1_H	0h	r	w	Port GPIO(61)
29dt28	GPIO_62_MODE_1_H	0h	r	w	Port GPIO(62)
31dt30	GPIO_63_MODE_1_H	0h	r	w	Port GPIO(63)

Register:	GPIO_IOCTL_2			Address:	40h
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r w
Description:	Configuration register for General Purpose IOs (95:64) 0: GPIOx = output, 1: GPIOx = input				

Register:	GPIO_OUT_2			Address:	44h
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh w
Description:	Output register for General Purpose IOs (95:64) 0: GPIO outputx = 0, 1: GPIO outputx = 1				

Register:	GPIO_OUT_SET_2			Address:	48h
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh wt
Description:	Bit-selective setting of the output register for General Purpose IOs (95:64) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 1 Read always returns 0				

Register:	GPIO_OUT_CLEAR_2			Address:	4Ch
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh wt
Description:	Bit-selective reset of the output register for General Purpose IOs (95:64) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 0 Read always returns 0				

Register:	GPIO_RES_DIS_2			Address:	50h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Bit-selective reset disabling of XRESET_GPIO_SM signal for registers of General Purpose IOs (31:0) For writing: 0: XRESET_GPIO_SM resets the corresponding register bit of all registers except of GPIO_RES_DIS 1: XRESET_GPIO_SM has no effect on the corresponding register bit					

Register:	GPIO_IN_2			Address:	54h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Input register for General Purpose IOs (95:64)					

Register:	GPIO_PORT_MODE_2_L			Address:	58h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Configuration register for GPIO port (79:64) Function assignment: 00 =GPIO function 01 =alternate function A 10 =alternate function B 11 =alternate function C					
Bit	Identifier	Reset	Attr.		Function / Description	
1dt0	GPIO_64_MODE_2_L	0h	r	w	Port GPIO(64)	
3dt2	GPIO_65_MODE_2_L	0h	r	w	Port GPIO(65)	
5dt4	GPIO_66_MODE_2_L	0h	r	w	Port GPIO(66)	
7dt6	GPIO_67_MODE_2_L	0h	r	w	Port GPIO(67)	
9dt8	GPIO_68_MODE_2_L	0h	r	w	Port GPIO(68)	
11dt10	GPIO_69_MODE_2_L	0h	r	w	Port GPIO(69)	
13dt12	GPIO_70_MODE_2_L	0h	r	w	Port GPIO(70)	
15dt14	GPIO_71_MODE_2_L	0h	r	w	Port GPIO(71)	
17dt16	GPIO_72_MODE_2_L	0h	r	w	Port GPIO(72)	
19dt18	GPIO_73_MODE_2_L	0h	r	w	Port GPIO(73)	
21dt20	GPIO_74_MODE_2_L	0h	r	w	Port GPIO(74)	
23dt22	GPIO_75_MODE_2_L	0h	r	w	Port GPIO(75)	

25dt24	GPIO_76_MODE_2_L	0h	r	w	Port GPIO(76)
27dt26	GPIO_77_MODE_2_L	0h	r	w	Port GPIO(77)
29dt28	GPIO_78_MODE_2_L	0h	r	w	Port GPIO(78)
31dt30	GPIO_79_MODE_2_L	0h	r	w	Port GPIO(79)

Register:	GPIO_PORT_MODE_2_H				Address:	5Ch	
Bits:	31dt0		Reset value:	00000000h		Attributes:	r w
Description:		Configuration register for GPIO port (95:80) Function assignment: 00 =GPIO function 01 =alternate function A 10 =alternate function B 11 =alternate function C					
Bit	Identifier		Reset	Attr.		Function / Description	
1dt0	GPIO_80_MODE_2_H		0h	r	w	Port GPIO(80)	
3dt2	GPIO_81_MODE_2_H		0h	r	w	Port GPIO(81)	
5dt4	GPIO_82_MODE_2_H		0h	r	w	Port GPIO(82)	
7dt6	GPIO_83_MODE_2_H		0h	r	w	Port GPIO(83)	
9dt8	GPIO_84_MODE_2_H		0h	r	w	Port GPIO(84)	
11dt10	GPIO_85_MODE_2_H		0h	r	w	Port GPIO(85)	
13dt12	GPIO_86_MODE_2_H		0h	r	w	Port GPIO(86)	
15dt14	GPIO_87_MODE_2_H		0h	r	w	Port GPIO(87)	
17dt16	GPIO_88_MODE_2_H		0h	r	w	Port GPIO(88)	
19dt18	GPIO_89_MODE_2_H		0h	r	w	Port GPIO(89)	
21dt20	GPIO_90_MODE_2_H		0h	r	w	Port GPIO(90)	
23dt22	GPIO_91_MODE_2_H		0h	r	w	Port GPIO(91)	
25dt24	GPIO_92_MODE_2_H		0h	r	w	Port GPIO(92)	
27dt26	GPIO_93_MODE_2_H		0h	r	w	Port GPIO(93)	
29dt28	GPIO_94_MODE_2_H		0h	r	w	Port GPIO(94)	
31dt30	GPIO_95_MODE_2_H		0h	r	w	Port GPIO(95)	

2.3.10.9 SCRB (System Control Register Block)

This block contains central registers for controlling the ERTEC 200P.

2.3.10.9.1 Hardware Identifier Register

Contains the identification number of the ERTEC 200P and the target platform (ASIC/FPGA), and the versioning in ClearCase (ID_REG, ERTEC 200PLUS_TAG).

32-bit register				
31 .. 21	20 .. 19	18 .. 16	15 .. 11	10 .. 0
11	2	3	5	11
Identification	Target Platform	Patch Label	Increment	R Label
IRTE,ICU,DSA,...	ASIC,FPGA,....	Gray-coded	ClearCase increment	ClearCase HDL label

Unique number per module, managed centrally in the A&D IP list. A number can consist of a "major" and "minor" number.		For labeling metal fixes in the ASIC flow; only valid if platform = ASIC. Increment/R Label is then invalid	Represents the HDL increment value of the ClearCase label	Represents the HDL counter value of the ClearCase label	
	00 = ASIC	000 = no patch yet			
	01 = FPGA	001 = 1st patch			
	10 = reserved	etc. (Gray-coded)			
	11 = user defined				

Table 21: ERTEC 200P, ID register

2.3.10.9.2 Boot Register

The required boot mode is saved in the BOOT_REG boot register. Some of the pins of the EMC interface (see table) are assigned to the boot register bits 0...4. The EMC pins are latched to the boot register boot register during an active XRESET reset and return to their normal function with an inactive reset.

BootPin(3) = A[16]	BootPin(2) = A[15]	BootPin(1) =XOE_DRIVER	BootPin(0) = DTXR	Boot mode	Booting...	BootPin(4) = XAV_BF
1	0	0	1	1	External NOR flash (16-bit) ¹⁾ , ASYNC_ADDR_MODE = 1 → <i>Compile mode</i> → <i>Copy mode</i>	1 0
1	0	1	0	2	External NOR flash (32-bit) ¹⁾ , ASYNC_ADDR_MODE = 1 → <i>Compile mode</i> → <i>Copy mode</i>	1 0
1	1	0	1	5	SPI master (RD Cmd: 0xE8)	X
1	1	1	0	6	SPI master (RD Cmd: 0x03)	X
1	1	1	1	7	XHIF (ext. host)	X

¹⁾ The secondary boot loader is run straight from the NOR flash and not from the TCM. The default mode, i.e. without external resistors, is a NOR flash with an access width of 32 bits and is selected with the internal pull circuit (highlighted in **blue**).

²⁾ Code mode is not supported

Table 22: Bootmodi adjustment

2.3.10.9.3 Config Register

Global use cases and different test modes can be set with EMC pins that are latched in the CONFIG_REG register during an XRESET active PowerOn reset. These pins resume their EMC function in normal mode once the reset is cleared.

CONFIG(0): Enable REF_CLK output (25 MHz) or disable (tristate).

CONFIG(1): ARM clock 125 / 250 MHz.

CONFIG(2): PLL bypass: A 125 MHz clock is supplied straight over BYP-CLK and the PLL is bypassed. BYP-CLK is usually used as the F-counter clock source. All internal clocks that operate at 125/250 MHz are set to 125 MHz (fixed setting). CONFIG(1) must also be set to '0'.

CONFIG(6-3): XHIF interface setting: (off, 16/32-bit, ARM926 trace port)

XHIF_XRDY setting: (low-active, high-active)

XHIF_XWR setting: (Wr or Read/Write Control)

GPIO31-0 setting: GPIO, PHY debug port

CONFIG (6) PIN: A23	CONFIG (5) PIN: A22	CONFIG (4) PIN: A21	CONFIG (3) PIN: A20	CONFIG (2) PIN: A19	CONFIG (1) PIN: A18	CONFIG (0) PIN: A17	Description
-	-	-	-	-	-	1	REF_CLK tristate
-	-	-	-	-	-	0	REF_CLK output (25 MHz)
-	-	-	-	-	0	-	ARM-Clock 125 MHz
-	-	-	-	-	1	-	ARM-Clock 250 MHz
-	-	-	-	0	-	-	CONFIG (2) must be tied to zero
0	0	-	0	-	-	-	XHIF = on, 16 bit Mode, GPIO94-79 and GPIO61-60 on (all inputs), XHIF_XWR has Read/Write-Control
0	0	-	1	-	-	-	XHIF = on, 16 bit Mode, GPIO94-79 and GPIO 61-60 on (all inputs), XHIF_XRD / XHIF_XWR separated
0	0	0	-	-	-	-	XHIF = on, 16 bit Mode, GPIO94-79 and GPIO 61-60 on (all inputs), XHIF_XRDY is high-active
0	0	1	-	-	-	-	XHIF = on, 16 bit Mode, GPIO94-79 and GPIO61-60 on (all inputs), XHIF_XRDY is low-active
0	1	-	0	-	-	-	XHIF = on, 32 bit mode, GPIO95-32 off, XHIF_XWR has Read/Write-Control
0	1	-	1	-	-	-	XHIF = on, 32 bit mode, GPIO95-32 off, XHIF_XRD / XHIF_XWR separated
0	1	0	-	-	-	-	XHIF = on, 32 bit mode, GPIO95-32 off, XHIF_XRDY is high-active
0	1	1	-	-	-	-	XHIF = on, 32 bit mode, GPIO95-32 off, XHIF_XRDY is low-active
1	0	0	1	-	-	-	XHIF = off, GPIO95-32 on (all inputs)
1	1	1	0	-	-	-	XHIF = off, ARM926 Trace Port = on (only at ARM926 Clock = 125 MHz)
Remainder				-	-	-	reserved

Blue: Default setting through the internal pulls

Table 23: Configuration adjustment

2.3.10.9.4 Reset Registers

Reset control register: With the control bits in the ASYN_RES_CTRL_REG and SYN_RES_CTRL_REG registers:

1. An enable for triggering a WD reset for ARM926 and PN-IP can be configured
2. Triggering a **synchronous** module reset for PerIF, HostIF, PN-IP
3. Triggering an **asynchronous** block or module reset in response to the following events can be activated/deactivated:
 - Expiry of the watchdog
 - SW activation of a reset (RES_SOFT, RES_SOFT_...) → One-shot bit

Note:

All reset sources are reset with a PowerOn reset (XRESET) only. This prevents a combinational loop in the design. The register can **only be written word by word**. 8-bit and 16-bit accesses produces access errors.

Reset status register: Active reset events that have occurred since the last PowerOn reset are saved in RES_STAT_REG. The SW can delete individual bits by writing a '1' to the bit position. The register is only reset with a PowerOn reset (XRESET).

The vector address for the secondary boot loader following an ARM926EJ-S SW reset is saved in RES_SOFT_RETURN_ADDR.

2.3.10.9.5 ACCESS_ERROR and QVZ Register

Access errors by the individual modules are combined here (OR logic operation) and forwarded to the interrupt controller (ARM-ICU) as AHB_Address_Error (IRQ51), APB_Address_Error (IRQ52), EMC_Address_Error (IRQ53) or IP_Address_Error (IRQ86):

1. Master access error interrupt (AHB_Address_Error): QVZ error at **ML-AHB** caused by master access by ARM_I, ARM_D, PN, GDMA and HostIF.
2. Slave access error interrupt (APB_Address_Error): QVZ error at **APB** caused by master access by ARM_I, ARM_D, PN, GDMA and HostIF.
3. EMC access error interrupt (EMC_Address_Error): QVZ error at **EMC** caused by master access by ARM_I, ARM_D, PN, GDMA, HostIF.
4. Slave access error interrupt (IP_Address_Error): Slave module **access error** at the APB or AHB bus (PN-IP, PerIF, I-filter, HostIF, SCRB).

The QVZ_AHB_M contains all the modules (AHB masters) that have caused an access error. Following an activated interrupt, the CPU can read the QVZ_AHB_M register and identify the trigger module on the basis of the bit position in the register. The software can then read the QVZ registers of the AHB masters in the SCRB (QVZ_AHB_ADR; QVZ_APB_ADR; QVZ_AHB_CTRL) and identify the address and access type that led to the error.

As a general rule, an access error sets the corresponding bit in QVZ_AHB_M, freezes the corresponding address and control information and generates an interrupt.

- Until the bit of an AHB master in QVZ_AHB_M is deleted by the SW, another AHB access error (from any AHB master) will **not** be saved. This allows unique assignment of the address and control information in the QVZ_AHB_ADR, QVZ_APB_ADR and QVZ_AHB_CTRL registers to the corresponding AHB master, even when multiple access errors from different AHB masters occur at the same time.
- Until the bit of an AHB/APB slave MODUL_ACCESS_ERR is cleared by the SW, another access error for the same APB slave will **not** be saved. An access error for another APB slave can however be locked at the same time, as the address and control information for the erroneous access is buffered in the corresponding APB module or in the SCRB register (ACCESS_ERROR_SCRB) and locked.

2.3.10.9.6 PLL Status Register

You can read the status of the PLL LOCK bit and the oscillator loss information by reading the PLL status register.

2.3.10.9.7 Memory Swapping

The reset vector of the ARM926 processor points to address 0x0000_0000. That is why the mirrored area of the boot ROM starts at address 0x0000_0000 (*MEM_SWAP* = "00")

after reset (HW, SW or ARM926 watchdog reset). The boot ROM can also be addressed in its original address range.

At the end of the boot operation (primary boot or secondary boot), the EMC interface maps the asynchronous memory (*MEM_SWAP* = "01") or SDRAM (*MEM_SWAP* = "10") to address 0x0000_0000h to allow the exception vector table for ARM926EJ-S to be created in address range 0x0000_0000 – 0x0000_001F. The original address ranges for boot ROM (in segment 4), EMC SDRAM (segment 2) and EMC memory (segment 3) are not affected by memory swapping.

Swapping is done by programming the *MEM_SWAP* register. There is no swapping in *MEM_SWAP* with reserved coding (or to be precise, no memory space is shown in segment 0). If these addresses are then accessed at the AHB, a QVZ (see 2.3.2.5.1) is triggered.

2.3.10.9.8 ARM Control Register

The following control information can be configured in this register:

1. ARB: Setting for the arbitration procedure for ML-AHB (see 2.3.10.9.8.1)
2. ARM926 AHB-LOCK

2.3.10.9.8.1 AHB Arbiters

Each of the ML-AHB AHB arbiters uses the same arbitration procedure. The default setting is round robin. A fixed priority assignment for the AHB masters could be used as an alternative arbitration algorithm by programming the ARB bit in the SCRB register. Switching to "fixed priority" fixes the priority set at ML-AHB when the arbitration type was configured. This option should not, however, be implemented in the light of the dynamic processes at the multi-layer AHB.

Using round robin as the arbitration procedure prevents the AHB masters from blocking each other at the multi-layer AHB for a prolonged period of time.

2.3.10.9.9 PHY Register

The configuration of both integrated PHYs can be set in the **PHY control register** *PHY_CONFIG*. This configuration is possible alongside the PHY configuration over SMI (PN-IP). With *PHY_LED_CONTROL*, you can control whether the PHY-LED link status and activity are to be controlled by the integrated PHY or SW. The LED status of the SW is also saved.

The readiness of both integrated PHYs can be read from the **PHY status register** *PHY_STATUS*.

2.3.10.9.10 AHB Burst Breaker Register

This register limits the maximum number of addresses of an AHM master in an ARM926 burst. A maximum limit of 256 addresses can be set. After the set number of addresses, burst access by the ARM926 is interrupted by an IDLE sequence. This function is required if an AHB master is blocking the bus for too long with burst access and preventing another AHB master from accessing the slave in time (for details, see 2.3.2.3.3).

2.3.10.9.11 GPIO Control Register

With LOCAL_OUT_READ_L and LOCAL_OUT_READ_H, you can configure whether the value of the GPIO block or of the data IO value is returned for GPIOs 32...95 (see 2.3.10.8.4.2).

To adapt the interrupt-compatible GPIOs 0...15 to an external circuit that supplies both low-active and high-active signals, they can be connected to the following modules over GPIO_INT_POLSEL either unchanged (not inverted) or inverted:

- ARM-ICU (GPIO0...15 → IRQ32..47, see 2.3.2.14)
- PN-ICU1 (GPIO4...7, see 2.3.8.1.2)
- GDMA (GPIO0...3 → HW jobs 16...19, see 2.3.4.2.1.1)

. POLSEL_GPIOx must be configured to ensure that these modules always receive a high-active signal. The value in the GPIO_IN_0 register is also saved unchanged or inverted from the the input signal.

2.3.10.9.12 I²C Clock Divider

The bit rate of the I²C bus is configured with the CCR_I2C register. I²C transmission rates of 100 kb/s max require a 1 MHz enable for I²C-IP (MI2C: Inventra). This enable is implemented in CCR_I2C.CDIV_VAL configuration. The value to be configured is calculated using the following formula:

CLK [MHz] / (ClockDivider + 1) = 1 MHz → with a 125 MHz system clock → CDIV_VAL = 0x7C (Default)

Important: A max. transmission rate of 100 kb/s is supported.

2.3.10.9.13 EDC Register

The EDC logic for D-TCM in ARM926 can be enabled and disabled in the EDC register EDC_PARITY_EN. The parity logic for the I and D cache for ARM926 can also be activated.

One-bit and multi-bit errors for all EDC-protected RAM are saved in EDC_EVENT and the corresponding level-triggered interrupt event *EDC_Event* (IRQ48) triggered in ARM-ICU (see 2.3.2.14). The register bits (i.e. the active interrupt level) can only be reset with write access by the FW (writing a '0').

Note: The register bits can also be set by the FW for debugging with write access (writing a '1').

If a one-bit or multi-bit error is saved at the same time as the FW deletes the error with write access, the one-bit or multi-bit error will not be deleted.

The end of RAM initialization (INIT_DONE) for all EDC-protected RAM is signaled in EDC_INIT_DONE.

2.3.10.9.14 ARM926 Mapping

The partitioning of ARM926 subsystem TCM into I-TCM and D-TCM can be configured in steps of 64 KByte in TCM926_MAP.

2.3.10.9.15 PAD Control Register

The IO buffers at the XHIF / GPIO interface support both 3.3 V and 1.8 V IO. The IO buffers of the EMC interface are designed for an IO voltage of 1.8 V. The **driver power** of

these special IO buffers depends on the IO voltage and can be configured in the SCRB registers DRIVE_EMC and DRIVExx_yyGPIO.

The **pull properties** of the EMC interface and GPIO31..0 und GPIO95..32 block signal pads can also be set in the SCRB register PULLxx_yyGPIO.

2.3.10.9.15.1 Driver Power (DRIVE)

The following registers are used to set the driver power of the outputs (see 2.3.10.9.22):

- DRIVE_EMC: Contains all EMC interface signal pads, divided into signal groups G1 – G11 (see register); settings for 1.8 V possible
- DRIVE15_0GPIO: Contains the GPIO15..0 pads; settings only possible for 3.3 V
- DRIVE31_15GPIO: Contains the GPIO31..15 pads; settings only possible for 3.3 V
- DRIVE47_32GPIO: Contains the GPIO47..32 pads; settings for 1.8 V/3.3 V possible
- DRIVE63_48GPIO: Contains the GPIO63..48 pads; settings for 1.8 V/3.3 V possible
- DRIVE79_64GPIO: Contains the GPIO79..64 pads; settings for 1.8 V/3.3 V possible
- DRIVE95_80GPIO: Contains the GPIO95..80 pads; settings for 1.8 V/3.3 V possible

The following combinations can be set for each GPIO pad / signal group (EMC for 1.8 V only; GPIO31:0 for 3.3 V only; GPIO95:32 for 1.8 V / 3.3 V). Following a PowerOn reset, the driver power is by default set to:

- 6 mA for GPIO31..0 pads at 3.3 V
- 9 mA for GPIO95..32/XHIF pads at 3.3 V
- 4 mA for GPIO95..32/XHIF pads at 1.8 V
- 12 mA for EMC pads at 1.8 V

For the 1.8 V XHIF interface, the external host may need to increase the driver power to 9 mA before it executes the first instance of read access. 4 mA is ok for the XHIF_XRDY pad, as the connection is a simple point-to-point one. The alternative functions (trace port, ext. PHY port, ...) are operated at 3.3 V.

Coding	GPIO31..0 (3.3 V)	GPIO96..32 / XHIF		EMC (1.8 V)
		(3.3 V)	(1.8 V)	
00b	4 mA	6 mA	3 mA (not used)	4 mA
01b	6 mA (default)	9 mA (default)	4 mA (default)	6 mA
10b	8 mA	18 mA (not used)	9 mA	8 mA
11b	12 mA	24 mA (not used)	12 mA	12 mA (default)

Following a reset, the driver power settings in the registers are preassigned. The settings can be changed by the SW at any time.

2.3.10.9.15.2 Pull setting (PULL)

The following registers are used to set the pull for the inputs (see 2.3.10.9.22):

- PULL15_0GPIO: Covers the GPIO15..0 pads
- PULL31_15GPIO: Covers the GPIO31..15 ads
- PULL47_32GPIO: Covers the GPIO47..32 pads
- PULL63_48GPIO: Covers the GPIO63..48 pads
- PULL79_64GPIO: Coves the GPIO79..64 pads
- PULL95_80GPIO: Covers the GPIO95..80 pads

The following combinations can be set for each GPIO pad:

Coding	Pull
00b	highZ
01b	Pull-Up
10b	highZ
11b	Pull-Down

The pull setting is only effective if the pad is connected to the input. If the pad is operating as an output, the pull is deactivated. The pulls have a resistance of 35 – 65 kΩ (typ. 50 kΩ, see 0).

During / following a reset, the pull settings in the registers are preassigned. The reset values can be found in the register description. With GPIO95..0, there is also a dependency on the CONFIG6:3 pins. Different pull settings are used during / after a reset for the 7 different modes. The settings can be changed by the SW at any time.

Important: *The pull setting does not change automatically following reconfiguration of the CONFIG6..3 bits in CONFIG_REG by the SW. It must be changed by the SW.*

2.3.10.9.15.3 Reset Behavior of EMC Pins without Pull Circuit

Some of the ERTEC 200P EMC output pins have no internal pull resistors:

- CLK_O_SDRAM2 ^{*)}
- CLK_O_BF0 – 2

^{*)} CLK_O_SDRAM0-1 are by default connected in line with DRIVE_EMC(22) (see 2.3.10.9.22) and are therefore not high-impedance during a reset

Remember this when connecting external components, as these pins become high-impedance during a reset and produce floating input signals at the other components. Blocks whose inputs are connected to these ERTEC 200P EMC output pins without internal pull resistors must if necessary be protected from a floating input voltage with external pulls.

2.3.10.9.16 Clear DMA Request

→ See 2.3.10.7.4

2.3.10.9.17 SPI Parity Error

A parity error at the group interrupt INT_SPI_ParityERR for the ARM-ICU (IRQ17) is generated in SPI_PARITY_ERROR.

2.3.10.9.18 XHIF Mode

Eight different address ranges (pages) of the ERTEC 200P can be addressed individually over the parallel host interface in HostIF (XHIF, see 2.3.6.2). The size of the max. address ranges (pages) can be configured with XHIF_MODE. Following a reset, a default max. address range per page is 1 MByte; the ASIC pin XHIF_XCS_R_A20 is connected to XHIF_XCS_R, so the pages can be reconfigured at the XHIF interface by the ext. host. When XHIF_MODE = '1', the ASIC pin XHIF_XCS_R_A20 is connected to XHIF_A20, expanding the address range for access to the ERTEC 200P to 2 MByte per page.

2.3.10.9.19 Ext. Driver Enable

With EXT_DRIVER_EN, you can configure the address range of the asynchronous EMC interface (CS0...CS3) for the chip select-specific control of an external driver at the EMC interface using the DTXR and XOE_DRIVER signals (for details, see 2.3.5.4.4).

It should also be possible to configure the CS0 address range (0x3000_0000 – 0x33FF_FFFF) at the EMC pin XRDY_BF by latching driver control information (EXT_DRIVER_DISABLE_CS0) when the reset (XRESET) is cleared. The integrated pull-up means that control for an external driver is deactivated by default and needs to be activated with a pull-down for the module.

2.3.10.9.20 SPI Mode

SPI1 or SPI2 operation as MISO/MOSI or SPI-IP wiring straight to the GPIOs can be set on an SPI-specific basis with SPI_MODE (for details, see 2.3.10.7.3)

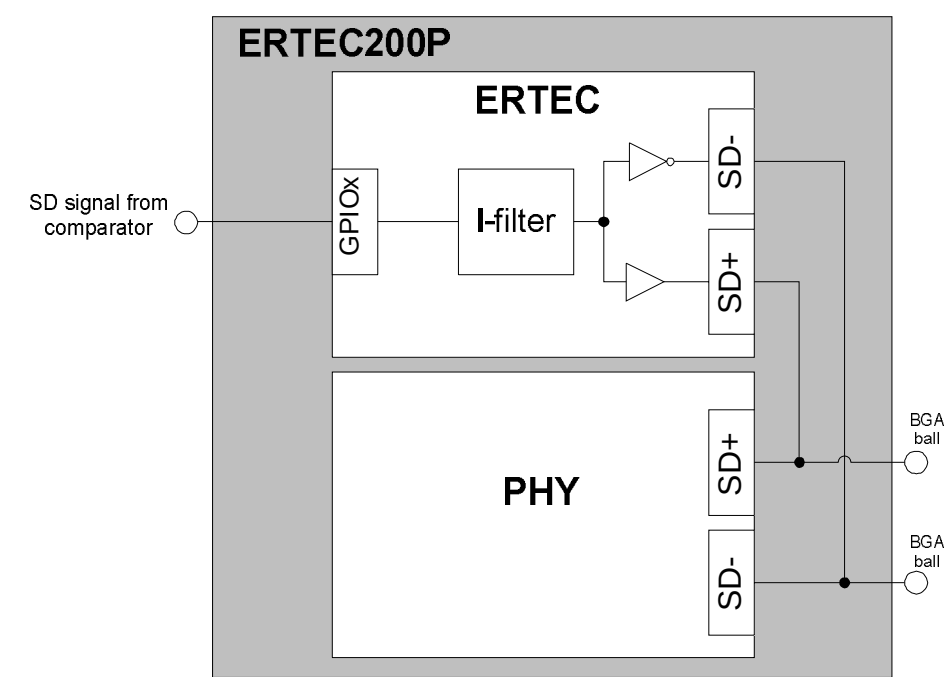
2.3.10.9.21 SD Signal Handling

When the ERTEC 200P is operated with fiber-optics, the external fiber-optic transceiver signals with signal detect (SD) whether a fiber-optic signal has been received. This SD signal is to be filtered on a port-specific basis and output as a differentiated signal for the ERTEC 200P integrated PHY. To avoid complex logic for preprocessing the SD signal in a given module, the ERTEC 200P can implement port-specific filtering and differentiation of the SD signal itself with SD_CONTROL.

A general enable for filtering and differentiation is implemented in SD_CONTROL.SD1/2_ENABLE for this purpose. If it is not, the ERTEC die port-specific SD+ and SD- are switched to high impedance to continue to allow the direct connection of an external differentiated SD signal over the ERTEC 200P BGA balls.

If SD_CONTROL.SD1/2_ENABLE = '1', filtering and differentiation are over the ERTEC die. With SD_CONTROL.SD1/2_MUX, you can select the GPIO over which the SD signal of a given port is received. A logical '0' can also be forced in place of the SD input signal for SW control of the SD signal when SD_CONTROL.SD1/2_MUX = 96...112. A logical '1' is used when SD_CONTROL.SD1/2_MUX = 113...127.

Note:
 When SD_CONTROL.SD1/2_MUX = 16...31, there is differentiation but generally **no** filtering (see 2.3.10.1 and 2.3.10.8.3).



→ For external wiring over GPIO, see 4.6.2.2.2.

2.3.10.9.22 Address Mapping

Start-Address	End-Address	Modul/Memory-Name
0h	C4h	SCRB

Module	Register/Memory	Read	Write	Address
/SCRB				
	ID_REG	r		0h
	BOOT_REG	rh		4h
	CONFIG_REG	rh	(w)	8h
	ASYN_RES_CTRL_REG	rh	w	Ch
	SYN_RES_CTRL_REG	r	w	10h

	RES_STAT_REG	rh		14h
	PLL_STAT_REG	rh		18h
	QVZ_AHB_ADR	rh		1Ch
	QVZ_AHB_CTRL	rh		20h
	QVZ_AHB_M	rh		24h
	QVZ_APB_ADR	rh		28h
	QVZ EMC_ADR	rh		2Ch
	MEM_SWAP	r	w	30h
	M_LOCK_CTRL	(r)	(w)	34h
	ERTEC 200PLUS_TAG	r		38h
	PHY_CONFIG	(r)(h)	(w)	3Ch
	PHY_STATUS	rh		40h
	AHB_BURSTBREAKER	r	w	44h
	LOCAL_OUT_READ_L	r	w	48h
	LOCAL_OUT_READ_H	r	w	4Ch
	CCR_I2C	r	w	50h
	EDC_EVENT	rh	w	54h
	EDC_INIT_DONE	rh		58h
	TCM926_MAP	r	w	5Ch
	GPIO_INT_POLSEL	r	w	60h
	EDC_PARITY_EN	r	w	64h
	MODUL_ACCESS_ERR	rh	w	68h
	RES_SOFT_RETURN_ADDR	r	w	6Ch
	PHY_LED_CONTROL	(r)(h)	(w)	70h
	ACCESS_ERROR_SCRB	(r)(h)	(w)	74h
	DRIVE EMC	r	w	78h
	DRIVE15_0GPIO	r	w	7Ch
	DRIVE31_16GPIO	r	w	80h
	DRIVE47_32GPIO	r	w	84h
	DRIVE63_48GPIO	r	w	88h
	DRIVE79_64GPIO	r	w	8Ch
	DRIVE95_80GPIO	r	w	90h
	PULL15_0GPIO	rh	w	94h

	PULL31_16GPIO	rh	w	98h
	PULL47_32GPIO	rh	w	9Ch
	PULL63_48GPIO	rh	w	A0h
	PULL79_64GPIO	rh	w	A4h
	PULL95_80GPIO	rh	w	A8h
	CLEAR_DMA_SPI_REQ_REG	rh	w	ACH
	SPI_PARITY_ERROR	rh	w	B0h
	SSI_DATA_INVERSION	r	w	B4h
	XHIF_MODE	r	w	B8h
	EXT_DRIVER_EN	r(h)	w	BCh
	SPI_MODE	r	w	C0h
	SD_CONTROL	r	w	C4h

2.3.10.9.23 Register Description

Module: SCRB

Register:		ID_REG			Address:		0h
Bits:	31dt0	Reset value:	40280200h	Attributes:	r		
Description:		Identification ERTEC 200P					
Bit	Identifier	Reset	Attr.	Function / Description			
7dt0	MET_FIX	01h	r	Metal fix: 01h			
15dt8	HW_R	2h	r	HW release: 02h			
31dt16	COMP	4028h	r	ERTEC 200P identifier: 4028h			

Register:		BOOT_REG			Address:		4h
Bits:	31dt0	Reset value:	0h	Attributes:	rh		
Description:		Boot mode pins BOOT(4:0) readable					
Bit	Identifier	Reset	Attr.	Function / Description			
0	BOOT_0	xh	r h	The value pending at the Boot(0) pin is latched when PowerOn reset is deactivated			
1	BOOT_1	xh	r h	The value pending at the Boot(1) pin is latched when PowerOn reset is			

				deactivated
2	BOOT_2	xh	r h	The value pending at the Boot(2) pin is latched when PowerOn reset is deactivated
3	BOOT_3	xh	r h	The value pending at the Boot(3) pin is latched when PowerOn reset is deactivated
4	BOOT_4	xh	r h	The value pending at the Boot(4) pin is latched when PowerOn reset is deactivated

Register:		CONFIG_REG			Address: 8h	
Bits:		31dt0	Reset value:	0h	Attributes:	rh (w)
Description:		ERTEC 200P ConfigPins CONFIG(6:0) readable, config REG(6-3) writable				
Bit	Identifier	Reset	Attr.	Function / Description		
0	CONF_0	xh	r h	The value pending at the Config(0) pin is latched when PowerOn reset is deactivated		
1	CONF_1	xh	r h	The value pending at the Config(1) pin is latched when PowerOn reset is deactivated		
2	CONF_2	xh	r h	The value pending at the Config(2) pin is latched when PowerOn reset is deactivated		
6dt3	CONF_6_3	xh	r h w	The value pending at the Config(6-3) pins is latched when PowerOn reset is deactivated and added to the register following deactivation of the internally extended system reset; the value can then be modified with write access 00-0: XHIF = on, 16-bit mode, GPIO94-79, 61-60 on (all inputs), XHIF_XWR has read/write control 00-1: XHIF = on, 16-bit mode, GPIO94-79, 61-60 on (all inputs), XHIF_XRD / XHIF_XWR separate 000-: XHIF = on, 16-bit mode, GPIO94-79, 61-60 on (all inputs), XHIF_XRDY is high-active 001-: XHIF = on, 16-bit mode, GPIO94-79, 61-60 on (all inputs),		

					XHIF_XRDY is low-active 01-0: XHIF = on, 32-bit mode, GPIO95-32 off, XHIF_XWR has read/write control 01-1: XHIF = on, 32-bit mode, GPIO95-32 off, XHIF_XRD / XHIF_XWR separate 010-: XHIF = on, 32-bit mode, GPIO95-32 off, XHIF_XRDY is high-active 011-: XHIF = on, 32-bit mode, GPIO95-32 off, XHIF_XRDY is low-active
--	--	--	--	--	--

Register:		ASYN_RES_CTRL_REG		Address:	Ch	
Bits:	31dt0	Reset value:	28h	Attributes:	rh	w
Description:		Control register for ERTEC 200P reset Reset for slice EN_WD_RES_PN is with PNIP reset. The remaining slices are reset with a system reset.				
Bit	Identifier	Reset	Attr.	Function / Description		
0	WD_RES_FREI_ARM926	0h	r h w	1: Watchdog reset enable for ARM926		
1	RES_SOFT	0h	r h w	1: Asynchronous software reset (not retentive, PULSE_DUR is reset length), resets everything except PN-IP		
2	RES_SOFT_PN	0h	r h w	1: Asynchronous software reset (not retentive, PULSE_DUR is reset length), resets PN-IP only		
3	EN_WD_RES_PN	1h	r h w	0: PN-IP is not reset by an ARM926 watchdog reset 1: PN-IP is reset by an ARM926 watchdog reset		
4	reserved	0h	r h			
5	reserved	1h	r h			
6	RES_SOFT_ARM926_CORE	0h	r h w	1: Asynchronous software reset (not retentive, PULSE_DUR is reset length), resets ARM926EJ-S core system only		

7	reserved	0h	r	w	
31dt1 6	PULSE_DUR	0h	r	w	Pulse duration for SW resets RES_SOFT, RES_SOFT_PN, ARM926 watchdog reset $TRES_PULSE = (8 \times n + 8) \times TCLK$ TCLK: APB clock period (1/125 MHz = 8 ns) n: Value of PULSE_DUR (0 .. 65535) The integrated PHYs require a reset duration of - 100 us. n - 1562 must therefore be set

Register:		SYN_RES_CTRL_REG			Address:		10h
Bits:		31dt0	Reset value:	0h	Attributes:	r	w
Description:		Control register for synchronous ERTEC 200P reset					
Bit	Identifier	Reset	Attr.	Function / Description			
0	SYN_RES_PER_IF	xh	r w	0: No synchronous reset with PER-IF 1: Synchronous reset with PER-IF (retentive)			
1	SYN_RES_HOST	xh	r w	0: No synchronous reset with host interface 1: Synchronous reset with host interface (retentive)			
2	SYN_RES_PN_IP	xh	r w	0: No synchronous reset with PN-IP 1: Synchronous reset with PN-IP (retentive)			
3	SYN_RES_RS_CONTROLLER	xh	r w	0: No synchronous reset with RS controller 1: Synchronous reset with RS controller (retentive)			

Register:		RES_STAT_REG			Address:		14h
Bits:		31dt0	Reset value:	4h	Attributes:	rh	
Description:		Status register for ERTEC 200P reset Only the bit for the most recent reset event is ever set. The two other bits are reset.					
Bit	Identifier	Reset	Attr.	Function / Description			
0	ARM926_WDOG_RES	xh	r	1: The last reset was with the			

			h	ARM926 watchdog
1	SW_RES	xh	r h	1: The last reset was a software reset
2	PWRON_HW_RES	1h	r h	1: The last reset was a power on or hardware reset
3	SW_RES_ARM926	xh	r h	1: The last reset was an ARM926 core software reset

Register:		PLL_STAT_REG		Address:		18h
Bits:		31dt0	Reset value:	1h	Attributes:	rh
Description:		Status register for ERTEC 200P PLL				
Bit	Identifier	Reset	Attr.	Function / Description		
0	LOCK	1h	r h	Lock: Lock at operating frequency status of PLL: 0: PLL is unlocked 1: PLL is locked This bit represents the current lock status of the PLL. Read-only		
1	LOSS	0h	r h	Loss: PLL input clock monitoring status 1: PLL input clock not recognized 0: PLL input clock This bit shows the current monitoring status of the PLL input clock. Read-only		
31dt2	reserved	0h				

Register:		QVZ_AHB_ADR		Address:		1Ch
Bits:		31dt0	Reset value:	0h	Attributes:	rh
Description:		Address of incorrect addressing at the multi-layer AHB				
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	QVZ_AHB_ADR	00000000h	r h	Address of incorrect addressing at the multi-layer AHB		

Register:		QVZ_AHB_CTRL		Address:		20h
Bits:		31dt0	Reset value:	0h	Attributes:	rh

Description:		Control signals of incorrect addressing at the multi-layer AHB		
Bit	Identifier	Reset	Attr.	Function / Description
0	R_W	xh	r h	Read/Write 0: read 1: write
3dt1	HSIZE	xh	r h	HSIZE
6dt4	HBURST	xh	r h	HBURST

Register:		QVZ_AHB_M			Address:	24h	
Bits:		31dt0	Reset value:	0h	Attributes:	rh	
Description:		Master detection for incorrect addressing at the multi-layer AHB					
Bit	Identifier	Reset	Attr.	Function / Description			
0	ARM_I	xh	r h	ARM926-I			
1	ARM_D	xh	r h	ARM926-D			
2	reserved	xh	r h	reserved			
3	PN	xh	r h	PROFINET_IP			
4	GDMA	xh	r h	GDMA			
5	HOSTIF	xh	r h	Host Interface			
6	reserved	xh	r h	reserved			

Register:		QVZ_APB_ADR			Address:	28h	
Bits:		31dt0	Reset value:	0h	Attributes:	rh	
Description:		Address of incorrect addressing at the APB					
Bit	Identifier	Reset	Attr.	Function / Description			
31dt0	QVZ_APB_ADR	00000000	r	Address of incorrect addressing at the			

		h	h	APB
--	--	---	---	-----

Register:	QVZ_EMC_ADR			Address:	2Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:		Address that resulted in QVZ at EMC				
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	QVZ_EMC_ADR	00000000h	rh	Address that resulted in QVZ at EMC		

Register:	MEM_SWAP			Address:	30h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:		Memory swapping in segment 0 on the AHB (ROM, EMC-SDRAM, EMC standard memory, I/D-TCM)				
Bit	Identifier	Reset	Attr.	Function / Description		
1dt0	SWAP	xh	rw	Selection of memory in segment 0 on the AHB: 00: Boot ROM from addr 0h 01: EMC-SDRAM from addr 0h (only the first 64 MByte) 10: EMC standard memory from addr 0h (only the first 64 MByte) 11: reserved (no memory range is to addr 0h; QVZ is generated upon access)		

Register:	M_LOCK_CTRL			Address:	34h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:		AHB master lock enable. Master-specific enable of AHB lock function.				
Bit	Identifier	Reset	Attr.	Function / Description		
0	LE_ARM926_I	0h	rw	Lock enable, AHB master ARM926-I: (I-master does not support locked transfers) 0: Lock disabled 1: Lock enabled		
1	LE_ARM926_D	0h	rw	Lock enable, AHB master ARM926-D: 0: Lock disabled		

					1: Lock enabled
2	reserved	0h	r	w	
5dt3	reserved				
6	ARB	0h	r	w	Select arbitration algorithm for AHB arbiter (ARB_MODE): 0: Round-robin 1: Fixed priority assignment This bit should not be changed (default: round robin)! (see 5.3.1.1)

Register:		ERTEC 200PLUS_TAG			Address: 38h	
Bits:		31dt0	Reset value:	00001013h	Attributes:	r
Description:		Tag number of current revision				
Bit	Identifier	Reset	Attr.	Function / Description		
10dt0	R_LABEL	13h	r	R-Label		
15dt1 1	INKREMENT_NR	2h	r	Increment no. (see ID_REG slice HW_R)		
18dt1 6	PATCH_NR	0h	r	Patch no. (see ID_REG slice MET_FIX)		
20dt1 9	PLATFORM	0h	r	Platform: 00 = ASIC, 01 = FPGA, 10 = reserved, 11 = user-defined		
31dt2 1	IDENTIFICATION	0h	r	Identification (not used)		

Register:		PHY_CONFIG			Address: 3Ch	
Bits:		31dt0	Reset value:	0h	Attributes:	(r)(h) () (w)
Description:		Configuration of PHY1 and PHY2 Important: PHY_CONFIG register is only reset with an asynchronous PN-IP reset.				
Bit	Identifier	Reset	Attr.	Function / Description		
0	P1_PHY_ENB	xh	r h w	0: PHY1 disabled (power down mode) - Comment: If the PHY is 'Disabled' and then 'Enabled' again, the SW must comply with a disable time of - 100?s (see Figure 37 - FSpec ERTEC 200P). 1: PHY1 enabled -		

				Comment: P1/2_PHYENABLE = 1 triggers a reset extension of 5.2 msec in the PHY. During this time, the PLL and all analog and digital components are started up. Readiness is signaled in the PHY_Status register with P1/2_PWRUPRST = 1 (see Figure 37 - FSpec ERTEC 200P). Comment: When CONFIG(6..3)="1101", the bit is not writable and is fixed at the default value.
1	P1_FX_MODE	xh	r h w	1: Enables 100BASE-FX interface (only useful when P1_PHY_Mode = 0010 or 0011) 0: The 100BASE-FX interface is enabled
5dt2	P1_2_PHY_MODE	xh	r h w	Setting applies to Phy P1 and Phy P2 together. 0000: 10BASE-T HD, Auto-Neg disabled 0001: 10BASE-T FD, Auto-Neg disabled 0010: 100BASE-TX/FX HD, Auto-Neg disabled 0011: 100BASE-TX/FX FD, Auto-Neg disabled 0100: 100BASE-TX HD signaled, Auto-Neg enabled 0101: 100BASE-TX HD signaled, Auto-Neg enabled, repeater mode 0110: PHY starts in power down mode 0111: Auto-Neg enabled, AutoMDIX enabled, all possible 1000: All capable. Quick Auto-Negotiation enabled. Forced Full Duplex in Parallel Detect mode. AutoMDIX enabled. Bits 1 and 0 determine timing 1001: 1010: 1011: 1100: 1101: 1110: 1111: Loopback mode. The Phy starts in loopback mode, with the 0.14 bit

				set. In this mode the Phy must be configured through the SMI interface and manually enabled by clearing bit 0.14. Until then it runs in loopback mode and is isolated from the line. Apart from Bit 0.14 the Phy is configured as with Phymode = 0111.
6	P1_AUTOMDIXEN	xh	r h w	1: Enable AutoMDIX state machine 0: Disable AutoMDIX state machine
8	P2_PHY_ENB	xh	r h w	0: PHY2 disabled (power down mode), see comment on bit 0 1: PHY2 enabled, see comment on bit 0
9	P2_FX_MODE	xh	r h w	1: Enables 100BASE-FX interface (only useful when P2_PHY_Mode=0010 or 0011) 0: The 100BASE-FX interface is enabled
13dt1 0	reserved	xh		
14	P2_AUTOMDIXEN	xh	r h w	1: Enable AutoMDIX state machine 0: Disable AutoMDIX state machine

Register:		PHY_STATUS			Address:		40h
Bits:		31dt0	Reset value:	0h	Attributes:	rh	
Description:		Status of PHY1 and PHY2 Important: PHY_STATUS register is only reset with an asynchronous PN-IP reset					
Bit	Identifier	Reset	Attr.	Function / Description			
0	P1_PWRUPRST	xh	r h	0: PHY1 in power down mode or internal reset still active 1: PHY1 is ready			
8	P2_PWRUPRST	xh	r h	0: PHY2 in power down mode or internal reset still active 1: PHY2 is ready			

Register:		AHB_BURSTBREAKER			Address:		44h
Bits:		31dt0	Reset value:	80808h	Attributes:	r	w
Description:		Maximal number of beats in an ARM9 burst. Longer bursts are split by inserting an IDLE transfer.					

Bit	Identifier	Reset	Attr.	Function / Description
7dt0	NR_ADDR_ARM926_D	8h	r w	0: bypass 1-255: max. number of addresses at ARM926-D AHB port
15dt8	NR_ADDR_ARM926_I	8h	r w	0: bypass 1-255: max. number of addresses at ARM926-I AHB port
23dt16		8h	r w	0: bypass

Register:		LOCAL_OUT_READ_L		Address: 48h	
Bits:	31dt0	Reset value:	0h	Attributes:	r w
Description:		Selection of GPIO_In path			
Bit	Identifier	Reset	Attr.	Function / Description	
0	GPIO32_IN_MUX	xh	r w	0: Reads GPIO32 pad 1: Reads LOC_IO0 Out	
1	GPIO33_IN_MUX	xh	r w	0: Reads GPIO33 pad 1: Reads LOC_IO1 Out	
2	GPIO34_IN_MUX	xh	r w	0: Reads GPIO34 pad 1: Reads LOC_IO2 Out	
3	GPIO35_IN_MUX	xh	r w	0: Reads GPIO35 pad 1: Reads LOC_IO3 Out	
4	GPIO36_IN_MUX	xh	r w	0: Reads GPIO36 pad 1: Reads LOC_IO4 Out	
5	GPIO37_IN_MUX	xh	r w	0: Reads GPIO37 pad 1: Reads LOC_IO5 Out	
6	GPIO38_IN_MUX	xh	r w	0: Reads GPIO38 pad 1: Reads LOC_IO6 Out	
7	GPIO39_IN_MUX	xh	r w	0: Reads GPIO39 pad 1: Reads LOC_IO7 Out	
8	GPIO40_IN_MUX	xh	r w	0: Reads GPIO40 pad 1: Reads LOC_IO8 Out	
9	GPIO41_IN_MUX	xh	r w	0: Reads GPIO41 pad 1: Reads LOC_IO9 Out	
10	GPIO42_IN_MUX	xh	r w	0: Reads GPIO42 pad 1: Reads LOC_IO10 Out	
11	GPIO43_IN_MUX	xh	r w	0: Reads GPIO43 pad 1: Reads LOC_IO11 Out	

12	GPIO44_IN_MUX	xh	r	w	0: Reads GPIO44 pad 1: Reads LOC_IO12 Out
13	GPIO45_IN_MUX	xh	r	w	0: Reads GPIO45 pad 1: Reads LOC_IO13 Out
14	GPIO46_IN_MUX	xh	r	w	0: Reads GPIO46 pad 1: Reads LOC_IO14 Out
15	GPIO47_IN_MUX	xh	r	w	0: Reads GPIO47 pad 1: Reads LOC_IO15 Out
16	GPIO48_IN_MUX	xh	r	w	0: Reads GPIO48 pad 1: Reads LOC_IO16 Out
17	GPIO49_IN_MUX	xh	r	w	0: Reads GPIO49 pad 1: Reads LOC_IO17 Out
18	GPIO50_IN_MUX	xh	r	w	0: Reads GPIO50 pad 1: Reads LOC_IO18 Out
19	GPIO51_IN_MUX	xh	r	w	0: Reads GPIO51 pad 1: Reads LOC_IO19 Out
20	GPIO52_IN_MUX	xh	r	w	0: Reads GPIO52 pad 1: Reads LOC_IO20 Out
21	GPIO53_IN_MUX	xh	r	w	0: Reads GPIO53 pad 1: Reads LOC_IO21 Out
22	GPIO54_IN_MUX	xh	r	w	0: Reads GPIO54 pad 1: Reads LOC_IO22 Out
23	GPIO55_IN_MUX	xh	r	w	0: Reads GPIO55 pad 1: Reads LOC_IO23 Out
24	GPIO56_IN_MUX	xh	r	w	0: Reads GPIO56 pad 1: Reads LOC_IO24 Out
25	GPIO57_IN_MUX	xh	r	w	0: Reads GPIO57 pad 1: Reads LOC_IO25 Out
26	GPIO58_IN_MUX	xh	r	w	0: Reads GPIO58 pad 1: Reads LOC_IO26 Out
27	GPIO59_IN_MUX	xh	r	w	0: Reads GPIO59 pad 1: Reads LOC_IO27 Out
28	GPIO60_IN_MUX	xh	r	w	0: Reads GPIO60 pad 1: Reads LOC_IO28 Out
29	GPIO61_IN_MUX	xh	r	w	0: Reads GPIO61 pad 1: Reads LOC_IO29 Out
30	GPIO62_IN_MUX	xh	r	w	0: Reads GPIO62 pad 1: Reads LOC_IO30 Out
31	GPIO63_IN_MUX	xh	r	w	0: Reads GPIO63 pad

				1: Reads LOC_IO31 Out
--	--	--	--	-----------------------

Register:	LOCAL_OUT_READ_H			Address:	4Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w

Description:	Selection of GPIO_In path					
--------------	---------------------------	--	--	--	--	--

Bit	Identifier	Reset	Attr.	Function / Description
0	GPIO64_IN_MUX	xh	r w	0: Reads GPIO64 pad 1: Reads LOC_IO32 Out
1	GPIO65_IN_MUX	xh	r w	0: Reads GPIO65 pad 1: Reads LOC_IO33 Out
2	GPIO66_IN_MUX	xh	r w	0: Reads GPIO66 pad 1: Reads LOC_IO34 Out
3	GPIO67_IN_MUX	xh	r w	0: Reads GPIO67 pad 1: Reads LOC_IO35 Out
4	GPIO68_IN_MUX	xh	r w	0: Reads GPIO68 pad 1: Reads LOC_IO36 Out
5	GPIO69_IN_MUX	xh	r w	0: Reads GPIO69 pad 1: Reads LOC_IO71 Out
6	GPIO70_IN_MUX	xh	r w	0: Reads GPIO70 pad 1: Reads LOC_IO38 Out
7	GPIO71_IN_MUX	xh	r w	0: Reads GPIO71 pad 1: Reads LOC_IO39 Out
8	GPIO72_IN_MUX	xh	r w	0: Reads GPIO72 pad 1: Reads LOC_IO40 Out
9	GPIO73_IN_MUX	xh	r w	0: Reads GPIO73 pad 1: Reads LOC_IO41 Out
10	GPIO74_IN_MUX	xh	r w	0: Reads GPIO74 pad 1: Reads LOC_IO42 Out
11	GPIO75_IN_MUX	xh	r w	0: Reads GPIO75 pad 1: Reads LOC_IO43 Out
12	GPIO76_IN_MUX	xh	r w	0: Reads GPIO76 pad 1: Reads LOC_IO44 Out
13	GPIO77_IN_MUX	xh	r w	0: Reads GPIO77 pad 1: Reads LOC_IO45 Out
14	GPIO78_IN_MUX	xh	r w	0: Reads GPIO78 pad 1: Reads LOC_IO46 Out
15	GPIO79_IN_MUX	xh	r w	0: Reads GPIO79 pad

					1: Reads LOC_IO47 Out
16	GPIO80_IN_MUX	xh	r	w	0: Reads GPIO80 pad 1: Reads LOC_IO48 Out
17	GPIO81_IN_MUX	xh	r	w	0: Reads GPIO81 pad 1: Reads LOC_IO49 Out
18	GPIO82_IN_MUX	xh	r	w	0: Reads GPIO82 pad 1: Reads LOC_IO50 Out
19	GPIO83_IN_MUX	xh	r	w	0: Reads GPIO83 pad 1: Reads LOC_IO51 Out
20	GPIO84_IN_MUX	xh	r	w	0: Reads GPIO84 pad 1: Reads LOC_IO52 Out
21	GPIO85_IN_MUX	xh	r	w	0: Reads GPIO85 pad 1: Reads LOC_IO53 Out
22	GPIO86_IN_MUX	xh	r	w	0: Reads GPIO86 pad 1: Reads LOC_IO54 Out
23	GPIO87_IN_MUX	xh	r	w	0: Reads GPIO87 pad 1: Reads LOC_IO55 Out
24	GPIO88_IN_MUX	xh	r	w	0: Reads GPIO88 pad 1: Reads LOC_IO56 Out
25	GPIO89_IN_MUX	xh	r	w	0: Reads GPIO89 pad 1: Reads LOC_IO57 Out
26	GPIO90_IN_MUX	xh	r	w	0: Reads GPIO90 pad 1: Reads LOC_IO58 Out
27	GPIO91_IN_MUX	xh	r	w	0: Reads GPIO91 pad 1: Reads LOC_IO59 Out
28	GPIO92_IN_MUX	xh	r	w	0: Reads GPIO92 pad 1: Reads LOC_IO60 Out
29	GPIO93_IN_MUX	xh	r	w	0: Reads GPIO93 pad 1: Reads LOC_IO61 Out
30	GPIO94_IN_MUX	xh	r	w	0: Reads GPIO94 pad 1: Reads LOC_IO62 Out
31	GPIO95_IN_MUX	xh	r	w	0: Reads GPIO95 pad 1: Reads LOC_IO63 Out

Register:	CCR_I2C			Address:	50h	
Bits:	31dt0	Reset value:	7Ch	Attributes:	r	w
Description:	Clock Control register for the I2C_3 interface divider value for determining the bit rate.					

Bit	Identifier	Reset	Attr.	Function / Description
7dt0	CDIV_VAL	7Ch	r w	Divider value for determining the bit rate $fBR = fCLK/(CCR_I2C+1)$ - $fCLK = fAPB=125\text{ MHz}$ in the I2C fBR: Bit rate clock (I2C) fCLK: I2C interface system cycle clock for fBR = 1MHz and fCLK=125MHz: CCR_I2C=124(dec.)

Register:		EDC_EVENT			Address: 54h	
Bits:		31dt0	Reset value:	0h	Attributes:	rh w
Description:		EDC event register - '0h' must be written to the register to clear.				
Bit	Identifier	Reset	Attr.	Function / Description		
0	I_TCM926_1B	xh	r h w	A 1-bit error has occurred in the I-TCM of ARM926 and been corrected		
1	I_TCM926_2B	xh	r h w	A 2-bit error has occurred in the I-TCM of ARM926		
2	D_TCM926_1B	xh	r h w	A 1-bit error has occurred in the D-TCM of ARM926 and been corrected		
3	D_TCM926_2B	xh	r h w	A 2-bit error has occurred in the D-TCM of ARM926		
4	GDMA_1B	xh	r h w	A 1-bit error has occurred in the GDMA memory and been corrected		
5	GDMA_2B	xh	r h w	A 2-bit error has occurred in the GDMA memory		
6	PN_1B	xh	r h w	A 1-bit error has occurred in one of the PN-IP memories and been corrected		
7	PN_2B	xh	r h w	A 2-bit error has occurred in one of the PN-IP memories		
8	PERIF_1B	xh	r h w	A 1-bit error has occurred in the PerIF memory and been corrected		
9	PERIF_2B	xh	r h w	A 2-bit error has occurred in the PerIF memory		
10	I_CACHE_PAR	xh	r h w	A parity error has occurred in the I-cache during reading		

11	D_CACHE_PAR	xh	r	w	A parity error has occurred in the D-cache during reading
12	I_TAG_PAR	xh	r	w	A parity error has occurred in the I-Tag RAM during reading
13	D_TAG_PAR	xh	r	w	A parity error has occurred in the D-Tag RAM during reading
14		xh	r	w	
15		xh	r	w	
16		xh	r	w	
17		xh	r	w	

Register:		EDC_INIT_DONE			Address: 58h	
Bits:		31dt0	Reset value:	0h	Attributes:	rh
Description:		EDC Init Done register - the status can be read				
Bit	Identifier	Reset	Attr.	Function / Description		
0	I_TCM926_INIT_DONE	xh	r	1: Initialization of the EDC bits in the I-TCM of ARM926 is complete		
1	D_TCM926_INIT_DONE	xh	r	1: Initialization of the EDC bits in the D-TCM of ARM926 is complete		
2	GDMA_INIT_DONE	xh	r	1: Initialization of the EDC bits in the GDMA is complete		
3	PN_IP_INIT_DONE	xh	r	1: Initialization of the EDC bits in the PN_IP is complete		
4	PERIF_INIT_DONE	xh	r	1: Initialization of the EDC bits in PER-IF is complete		
5		xh	r			
6		xh	r			

Register:		TCM926_MAP			Address: 5Ch	
Bits:		31dt0	Reset value:	0h	Attributes:	r w

Description:		TCM926 Map Register		
		Division into I / D-TCM and INITRAM		
Bit	Identifier	Reset	Attr.	Function / Description
2dt0	PARTITION_TCM926	xh	r w	I-TCM / D-TCM ARM926 000b: 0 KByte / 256 KByte 001b: 64 KByte / 192 KByte 010b: 128 KByte / 128 KByte 011b: 192 KByte / 64 KByte 100b: 256 KByte / 0 KByte
3	INITRAM	xh	r w	Booting I-TCM 0: I-TCM not booted. Booting in accordance with MEM_SWAP configuration. 1: I-TCM is activated after ARM926 core reset

Register:		GPIO_INT_POLSEL		Address:	60h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:		Interrupt polarity for GPIO interrupts (15:0)				
Bit	Identifier	Reset	Attr.	Function / Description		
0	INT_POLSEL_GPIO0	xh	r w	0: GPIO0 is not inverted from the A-ICU (IRQ32) and GDMA 1: GPIO0 is inverted from the A-ICU (IRQ32) and GDMA		
1	INT_POLSEL_GPIO1	xh	r w	0: GPIO1 is not inverted from the A-ICU (IRQ33) and GDMA 1: GPIO1 is inverted from the A-ICU (IRQ33) and GDMA		
2	INT_POLSEL_GPIO2	xh	r w	0: GPIO2 is not inverted from the A-ICU (IRQ34) and GDMA 1: GPIO2 is inverted from the A-ICU (IRQ34) and GDMA		
3	INT_POLSEL_GPIO3	xh	r w	0: GPIO3 is not inverted from the A-ICU (IRQ35) and GDMA 1: GPIO3 is inverted from the A-ICU (IRQ35) and GDMA		
4	INT_POLSEL_GPIO4	xh	r w	0: GPIO4 is not inverted from the A-ICU (IRQ36) and PN-ICU 1 1: GPIO4 is inverted from the A-ICU (IRQ36) and PN-ICU 1		
5	INT_POLSEL_GPIO5	xh	r w	0: GPIO5 is not inverted from the A-		

				ICU (IRQ37) and PN-ICU 1 1: GPIO5 is inverted from the A-ICU (IRQ37) and PN-ICU 1
6	INT_POLSEL_GPIO6	xh	r w	0: GPIO6 is not inverted from the A-ICU (IRQ38) and PN-ICU 1 1: GPIO6 is inverted from the A-ICU (IRQ38) and PN-ICU 1
7	INT_POLSEL_GPIO7	xh	r w	0: GPIO7 is not inverted from the A-ICU (IRQ39) and PN-ICU 1 1: GPIO7 is inverted from the A-ICU (IRQ39) and PN-ICU 1
8	INT_POLSEL_GPIO8	xh	r w	0: GPIO8 is not inverted from the A-ICU (IRQ40) 1: GPIO8 is inverted from the A-ICU (IRQ40)
9	INT_POLSEL_GPIO9	xh	r w	0: GPIO9 is not inverted from the A-ICU (IRQ41) 1: GPIO9 is inverted from the A-ICU (IRQ41)
10	INT_POLSEL_GPIO10	xh	r w	0: GPIO10 is not inverted from the A-ICU (IRQ42) 1: GPIO10 is inverted from the A-ICU (IRQ42)
11	INT_POLSEL_GPIO11	xh	r w	0: GPIO11 is not inverted from the A-ICU (IRQ43) 1: GPIO11 is inverted from the A-ICU (IRQ43)
12	INT_POLSEL_GPIO12	xh	r w	0: GPIO12 is not inverted from the A-ICU (IRQ44) 1: GPIO12 is inverted from the A-ICU (IRQ44)
13	INT_POLSEL_GPIO13	xh	r w	0: GPIO13 is not inverted from the A-ICU (IRQ45) 1: GPIO13 is inverted from the A-ICU (IRQ45)
14	INT_POLSEL_GPIO14	xh	r w	0: GPIO14 is not inverted from the A-ICU (IRQ46) 1: GPIO14 is inverted from the A-ICU (IRQ46)
15	INT_POLSEL_GPIO15	xh	r w	0: GPIO15 is not inverted from the A-ICU (IRQ47) 1: GPIO15 is inverted from the A-ICU (IRQ47)

Register:	EDC_PARITY_EN			Address:	64h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:		EDC and Parity Enable (2:0)				
Bit	Identifier	Reset	Attr.	Function / Description		
0	I_CACHE_PAR_EN	xh	r w	0: Parity logic for ARM926 I-cache / I-tag is disabled (default) 1: Parity logic for ARM926 I-cache / I-tag is enabled		
1	D_CACHE_PAR_EN	xh	r w	0: Parity logic for ARM926 D-cache / D-tag is disabled (default) 1: Parity logic for ARM926 D-cache / D-tag is enabled		
2	EDC_DISABLE_ARM926	xh	r w	0: The EDC logic in ARM926 I/D-TCM is enabled (default) 1: The EDC logic in ARM926 I/D-TCM is disabled		
3	reserved	xh	r w	0: reserved (default) 1: reserved		

Register:	MODUL_ACCESS_ERR			Address:	68h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:		Module Access Error register (5:0) - '0'h must be written to the register to clear				
Bit	Identifier	Reset	Attr.	Function / Description		
0	PN_IP_ACCESS_ERR	xh	r h w	0: No access error has occurred in the PN-IP 1: An access error has occurred in the PN-IP		
1	PER_IF_ACCESS_ERR	xh	r h w	0: No access error has occurred in the PER-IF 1: An access error has occurred in the PER-IF		
2	I_FILTER_ACCESS_ERR	xh	r h w	0: No access error has occurred in the I-filter 1: An access error has occurred in the I-filter		
3	HOST_IF_ACCESS_ERR	xh	r h w	0: No access error has occurred in the HOST IF 1: An access error has occurred in the		

				HOST IF
4	SCRB_ACCESS_ERR	xh	r h w	0: No access error has occurred in the SCRB 1: An access error has occurred in the SCRB
5	reserved	xh	r h w	
6	reserved	xh	r h w	

Register:		RES_SOFT_RETURN_ADDR		Address:		6Ch	
Bits:		31dt0	Reset value:	0h	Attributes:	r	w
Description:		Vector address for the secondary boot loader following an ARM926EJ-S SW reset					
Bit	Identifier	Reset	Attr.	Function / Description			
31dt0	RETURN_ADDRESS	xh	r w	Vector address for the secondary boot loader following RES_SOFT_ARM926_CORE. This SW reset is executed in the TCM926_MAP register after TCM 926 configuration.			

Register:		PHY_LED_CONTROL		Address:		70h	
Bits:		31dt0	Reset value:	0h	Attributes:	(r)(h))	(w)
Description:		PHY-LED control					
Bit	Identifier	Reset	Attr.	Function / Description			
0	EN_P1_XLINK_STATUS	xh	r w	0: LED: P1_XLINK_STATUS is controlled by the PHY 1: LED: P1_XLINK_STATUS is controlled by the SW			
1	EN_P1_XACTIVITY	xh	r w	0: LED: P1_XLINK_XACTIVITY is controlled by the PHY 1: LED: P1_XLINK_XACTIVITY is controlled by the SW			
2	EN_P2_XLINK_STATUS	xh	r w	0: LED: P2_XLINK_STATUS is controlled by the PHY 1: LED: P2_XLINK_STATUS is controlled by the SW			
3	EN_P2_XACTIVITY	xh	r w	0: LED: P2_XLINK_XACTIVITY is			

				controlled by the PHY 1: LED: P2_XLINK_XACTIVITY is controlled by the SW
7dt4	reserved	xh		No function, should always be '0000'
8	P1_XLINK_STATUS_SW	xh	r w	0: LED: P1_XLINK_STATUS is '0', set by the SW 1: LED: P1_XLINK_STATUS is '1', set by the SW
9	P1_XACTIVITY_SW	xh	r w	0: LED: P1_XLINK_XACTIVITY is '0', set by the SW 1: LED: P1_XLINK_XACTIVITY is '1', set by the SW
10	P2_XLINK_STATUS_SW	xh	r w	0: LED: P2_XLINK_STATUS is '0', set by the SW 1: LED: P2_XLINK_STATUS is '1', set by the SW
11	P2_XACTIVITY_SW	xh	r w	0: LED: P2_XLINK_XACTIVITY is '0', set by the SW 1: LED: P2_XLINK_XACTIVITY is '1', set by the SW
15dt1 2	reserved	xh		No function, should always be '0000'
16	P1_XLINK_STATUS_PHY	xh	r h	P1_XLINK_STATUS controlled by the PHY
17	P1_XACTIVITY_PHY	xh	r h	P1_XACTIVITY controlled by the PHY
18	P2_XLINK_STATUS_PHY	xh	r h	P2_XLINK_STATUS controlled by the PHY
19	P2_XACTIVITY_PHY	xh	r h	P2_XACTIVITY controlled by the PHY

Register:	ACCESS_ERROR_SCRB			Address:	74h	
Bits:	31dt0	Reset value:	20000000h	Attributes:	(r)(h)	(w)
Description:	SCRB Access Error					
Bit	Identifier	Reset	Attr.	Function / Description		
7dt0	APB_ADDRESS	xh	r h w	Erroneous APB address in SCRБ		
27dt8	Reserved	xh		READ_DATA = (others = '0')		
29dt2	APB_SIZE	2h	r	Clamped at 10 (word access) as only		

8				word access is permitted
30	APB_WRITE	xh	r h w	0: Read access 1: Write access
31	ERR_LOCK	xh	r h w	Set to '1' by the HW if erroneous SCRB access is detected. Further HW entries are then blocked. The SW must reset the bit to '0' to enable new entries.

Register:		DRIVE EMC			Address: 78h	
Bits:	31dt0	Reset value:	FFFFFFh	Attributes:	r	w
Description:		SCRB Drive Current of Dedicated Signals (1.8V) Each GPIO bit is set to the drive current on the basis of the following coding -----1.8V----- 00 - 4 mA 01 - 6 mA 10 - 8 mA 11 - 12 mA and clock enable/disable for BF and SDRAM				
Bit	Identifier	Reset	Attr.	Function / Description		
1dt0	G1: DR_EMC_C	3h	r w	Signal list: DTXR, XOE_DDRIVER		
3dt2	G2: DR_EMC_AL	3h	r w	Signal list: A14 - A0		
5dt4	G3: DR_EMC_AH	3h	r w	Signal list: A23 - A15		
7dt6	G4: DR_EMC_DL	3h	r w	Signal list: D15 - D0, XBE0_DQM0, XBE1_DQM1		
9dt8	G5: DR_EMC_DH	3h	r w	Signal list: D31 - D16, XBE2_DQM2, XBE3_DQM3		
11dt10	G6: DR_EMC_RW	3h	r w	Signal list: XWR, XRD		
13dt12	G7: DR_EMC_PER	3h	r w	Signal list: XCS_PER0 - 3		
15dt14	G8: DR_EMC_CLK_SDRAM	3h	r w	Signal lists: CLK_O_SDRAM0, CLK_O_SDRAM1, CLK_O_SDRAM2		
17dt16	G9: DR_EMC_SDRAM	3h	r w	Signal list: XCS_SDRAM, XRAS_SDRAM, XCAS_SDRAM, XWE_SDRAM		
19dt18	G10: DR_EMC_CLK_BF	3h	r w	Signal list: CLK_O_BF0 - 2		

21dt2 0	G11: DR_EMC_XAV_BF	3h	r	w	Signal list: XAV_BF
22	EMC_SDRAM_CLK_SD0SD1_EBL	1h	r	w	enable(=1) / disable(=0) CLK_O_SDRAM0 and CLK_O_SDRAM1 (one SDRAM block)
23	EMC_SDRAM_CLK_SD2_EBL	1h	r	w	enable(=1) / disable(=0) CLK_O_SDRAM2 (with 2nd SDRAM block)
24	EMC_BF_CLK_BF0BF1_EBL	0h	r	w	enable(=1) / disable(=0) CLK_O_BF0 and CLK_O_BF1 (one flash block)
25	EMC_BF_CLK_BF2_EBL	0h	r	w	enable(=1) / disable(=0) CLK_O_BF2 (with 2nd flash block)

Register:		DRIVE15_0GPIO			Address: 7Ch	
Bits:		31dt0	Reset value:	55555555h	Attributes:	r w
Description:		SCRB Drive Current of GPIO Signals Each GPIO bit is set to the drive current on the basis of the following coding 00 - 4 mA 01 - 6 mA 10 - 8 mA 11 - 12 mA				
Bit	Identifier	Reset	Attr.	Function / Description		
1dt0	DR_GPIO0	1h	r w	drive current GPIO0		
3dt2	DR_GPIO1	1h	r w	drive current GPIO1		
5dt4	DR_GPIO2	1h	r w	drive current GPIO2		
7dt6	DR_GPIO3	1h	r w	drive current GPIO3		
9dt8	DR_GPIO4	1h	r w	drive current GPIO4		
11dt10	DR_GPIO5	1h	r w	drive current GPIO5		
13dt12	DR_GPIO6	1h	r w	drive current GPIO6		
15dt14	DR_GPIO7	1h	r w	drive current GPIO7		
17dt16	DR_GPIO8	1h	r w	drive current GPIO8		
19dt1	DR_GPIO9	1h	r w	drive current GPIO9		

8					
21dt2 0	DR_GPIO10	1h	r	w	drive current GPIO10
23dt2 2	DR_GPIO11	1h	r	w	drive current GPIO11
25dt2 4	DR_GPIO12	1h	r	w	drive current GPIO12
27dt2 6	DR_GPIO13	1h	r	w	drive current GPIO13
29dt2 8	DR_GPIO14	1h	r	w	drive current GPIO14
31dt3 0	DR_GPIO15	1h	r	w	drive current GPIO15

Register:		DRIVE31_16GPIO			Address: 80h	
Bits:	31dt0	Reset value:	55555555h	Attributes:	r	w
Description:	SCRB Drive Current of GPIO Signals Each GPIO bit is set to the drive current on the basis of the following coding 00 - 4 mA 01 - 6 mA 10 - 8 mA 11 - 12 mA					
Bit	Identifier	Reset	Attr.	Function / Description		
1dt0	DR_GPIO16	1h	r w	drive current GPIO16		
3dt2	DR_GPIO17	1h	r w	drive current GPIO17		
5dt4	DR_GPIO18	1h	r w	drive current GPIO18		
7dt6	DR_GPIO19	1h	r w	drive current GPIO19		
9dt8	DR_GPIO20	1h	r w	drive current GPIO20		
11dt1 0	DR_GPIO21	1h	r w	drive current GPIO21		
13dt1 2	DR_GPIO22	1h	r w	drive current GPIO22		
15dt1 4	DR_GPIO23	1h	r w	drive current GPIO23		
17dt1 6	DR_GPIO24	1h	r w	drive current GPIO24		

19dt1 8	DR_GPIO25	1h	r	w	drive current GPIO25
21dt2 0	DR_GPIO26	1h	r	w	drive current GPIO26
23dt2 2	DR_GPIO27	1h	r	w	drive current GPIO27
25dt2 4	DR_GPIO28	1h	r	w	drive current GPIO28
27dt2 6	DR_GPIO29	1h	r	w	drive current GPIO29
29dt2 8	DR_GPIO30	1h	r	w	drive current GPIO30
31dt3 0	DR_GPIO31	1h	r	w	drive current GPIO31

Register:		DRIVE47_32GPIO			Address:84h		
Bits:		31dt0	Reset value:	55555555h	Attributes:	r	w
Description:		SCRB Drive Current of GPIO Signals (3.3V / 1.8V) Each GPIO bit is set to the drive current on the basis of the following coding -----3.3V-----					

6					
19dt1 8	DR_GPIO41	1h	r	w	drive current GPIO41
21dt2 0	DR_GPIO42	1h	r	w	drive current GPIO42
23dt2 2	DR_GPIO43	1h	r	w	drive current GPIO43
25dt2 4	DR_GPIO44	1h	r	w	drive current GPIO44
27dt2 6	DR_GPIO45	1h	r	w	drive current GPIO45
29dt2 8	DR_GPIO46	1h	r	w	drive current GPIO46
31dt3 0	DR_GPIO47	1h	r	w	drive current GPIO47

Register:		DRIVE63_48GPIO			Address: 88h	
Bits:		31dt0	Reset value:	55555555h	Attributes:	r w
Description:		SCRB Drive Current of GPIO Signals (3.3V / 1.8V) Each GPIO bit is set to the drive current on the basis of the following coding -----3.3V----- -----1.8V----- 00 - 6 mA 00 - 3 mA (n.a.) 01 - 12 mA 01 - 6 mA 10 - 18 mA (n.a.) 10 - 9 mA 11 - 24 mA (n.a.) 11 - 12 mA				
Bit	Identifier		Reset	Attr.	Function / Description	
1dt0	DR_GPIO48		1h	r w	drive current GPIO48	
3dt2	DR_GPIO49		1h	r w	drive current GPIO49	
5dt4	DR_GPIO50		1h	r w	drive current GPIO50	
7dt6	DR_GPIO51		1h	r w	drive current GPIO51	
9dt8	DR_GPIO52		1h	r w	drive current GPIO52	
11dt10	DR_GPIO53		1h	r w	drive current GPIO53	
13dt12	DR_GPIO54		1h	r w	drive current GPIO54	
15dt14	DR_GPIO55		1h	r w	drive current GPIO55	

17dt1 6	DR_GPIO56	1h	r	w	drive current GPIO56
19dt1 8	DR_GPIO57	1h	r	w	drive current GPIO57
21dt2 0	DR_GPIO58	1h	r	w	drive current GPIO58
23dt2 2	DR_GPIO59	1h	r	w	drive current GPIO59
25dt2 4	DR_GPIO60	1h	r	w	drive current GPIO60
27dt2 6	DR_GPIO61	1h	r	w	drive current GPIO61
29dt2 8	DR_GPIO62	1h	r	w	drive current GPIO62
31dt3 0	DR_GPIO63	1h	r	w	drive current GPIO63

Register:		DRIVE79_64GPIO			Address:8Ch		
Bits:		31dt0	Reset value:	55555555h	Attributes:	r	w
Description:		SCRB Drive Current of GPIO Signals (3.3V / 1.8V) Each GPIO bit is set to the drive current on the basis of the following coding -----3.3V-----					

4					
17dt1 6	DR_GPIO72	1h	r	w	drive current GPIO72
19dt1 8	DR_GPIO73	1h	r	w	drive current GPIO73
21dt2 0	DR_GPIO74	1h	r	w	drive current GPIO74
23dt2 2	DR_GPIO75	1h	r	w	drive current GPIO75
25dt2 4	DR_GPIO76	1h	r	w	drive current GPIO76
27dt2 6	DR_GPIO77	1h	r	w	drive current GPIO77
29dt2 8	DR_GPIO78	1h	r	w	drive current GPIO78
31dt3 0	DR_GPIO79	1h	r	w	drive current GPIO79

Register:		DRIVE95_80GPIO			Address:90h	
Bits:		31dt0	Reset value:	55555555h	Attributes:	r w
Description:		SCRB Drive Current of GPIO Signals (3.3V / 1.8V) Each GPIO bit is set to the drive current on the basis of the following coding -----3.3V----- -----1.8V----- 00 - 6 mA 00 - 3 mA (n.a.) 01 - 12 mA 01 - 6 mA 10 - 18 mA (n.a.) 10 - 9 mA 11 - 24 mA (n.a.) 11 - 12 mA				
Bit	Identifier		Reset	Attr.	Function / Description	
1dt0	DR_GPIO80		1h	r w	drive current GPIO80	
3dt2	DR_GPIO81		1h	r w	drive current GPIO81	
5dt4	DR_GPIO82		1h	r w	drive current GPIO82	
7dt6	DR_GPIO83		1h	r w	drive current GPIO83	
9dt8	DR_GPIO84		1h	r w	drive current GPIO84	
11dt10	DR_GPIO85		1h	r w	drive current GPIO85	
13dt12	DR_GPIO86		1h	r w	drive current GPIO86	

15dt1 4	DR_GPIO87	1h	r	w	drive current GPIO87
17dt1 6	DR_GPIO88	1h	r	w	drive current GPIO88
19dt1 8	DR_GPIO89	1h	r	w	drive current GPIO89
21dt2 0	DR_GPIO90	1h	r	w	drive current GPIO90
23dt2 2	DR_GPIO91	1h	r	w	drive current GPIO91
25dt2 4	DR_GPIO92	1h	r	w	drive current GPIO92
27dt2 6	DR_GPIO93	1h	r	w	drive current GPIO93
29dt2 8	DR_GPIO94	1h	r	w	drive current GPIO94
31dt3 0	DR_GPIO95	1h	r	w	drive current GPIO95

Register:		PULL15_0GPIO				Address: 94h	
Bits:		31dt0	Reset value:	0h	Attributes:	rh	w
Description:		SCRB Pull GPIO Signals Each GPIO bit is set to the pull resistors on the basis of the following coding 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistors are automatically deactivated Reset value does not apply as the register is loaded in accordance with the function set with CONFIG.					
Bit	Identifier		Reset	Attr.	Function / Description		
1dt0	PR_GPIO0		xh	r h	w	pull control GPIO0	
3dt2	PR_GPIO1		xh	r h	w	pull control GPIO1	
5dt4	PR_GPIO2		xh	r h	w	pull control GPIO2	

7dt6	PR_GPIO3	xh	r	w	pull control GPIO3
9dt8	PR_GPIO4	xh	r	w	pull control GPIO4
11dt10	PR_GPIO5	xh	r	w	pull control GPIO5
13dt12	PR_GPIO6	xh	r	w	pull control GPIO6
15dt14	PR_GPIO7	xh	r	w	pull control GPIO7
17dt16	PR_GPIO8	xh	r	w	pull control GPIO8
19dt18	PR_GPIO9	xh	r	w	pull control GPIO9
21dt20	PR_GPIO10	xh	r	w	pull control GPIO10
23dt22	PR_GPIO11	xh	r	w	pull control GPIO11
25dt24	PR_GPIO12	xh	r	w	pull control GPIO12
27dt26	PR_GPIO13	xh	r	w	pull control GPIO13
29dt28	PR_GPIO14	xh	r	w	pull control GPIO14
31dt30	PR_GPIO15	xh	r	w	pull control GPIO15

Register:	PULL31_16GPIO			Address:	98h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	SCRB Pull GPIO Signals Each GPIO bit is set to the pull resistors on the basis of the following coding 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistors are automatically deactivated Reset value does not apply as the register is loaded in accordance with the function set with CONFIG.					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	PR_GPIO16	xh	r	w	pull control GPIO16
3dt2	PR_GPIO17	xh	r	w	pull control GPIO17
5dt4	PR_GPIO18	xh	r	w	pull control GPIO18
7dt6	PR_GPIO19	xh	r	w	pull control GPIO19
9dt8	PR_GPIO20	xh	r	w	pull control GPIO20
11dt10	PR_GPIO21	xh	r	w	pull control GPIO21
13dt12	PR_GPIO22	xh	r	w	pull control GPIO22
15dt14	PR_GPIO23	xh	r	w	pull control GPIO23
17dt16	PR_GPIO24	xh	r	w	pull control GPIO24
19dt18	PR_GPIO25	xh	r	w	pull control GPIO25
21dt20	PR_GPIO26	xh	r	w	pull control GPIO26
23dt22	PR_GPIO27	xh	r	w	pull control GPIO27
25dt24	PR_GPIO28	xh	r	w	pull control GPIO28
27dt26	PR_GPIO29	xh	r	w	pull control GPIO29
29dt28	PR_GPIO30	xh	r	w	pull control GPIO30
31dt30	PR_GPIO31	xh	r	w	pull control GPIO31

Register:	PULL47_32GPIO			Address:	9Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	SCRB Pull GPIO Signals Each GPIO bit is set to the pull resistors on the basis of					

		the following coding 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistors are automatically deactivated Reset value does not apply as the register is loaded in accordance with the function set with CONFIG.			
Bit	Identifier	Reset	Attr.		Function / Description
1dt0	PR_GPIO32	xh	r h	w	pull control GPIO32
3dt2	PR_GPIO33	xh	r h	w	pull control GPIO33
5dt4	PR_GPIO34	xh	r h	w	pull control GPIO34
7dt6	PR_GPIO35	xh	r h	w	pull control GPIO35
9dt8	PR_GPIO36	xh	r h	w	pull control GPIO36
11dt10	PR_GPIO37	xh	r h	w	pull control GPIO37
13dt12	PR_GPIO38	xh	r h	w	pull control GPIO38
15dt14	PR_GPIO39	xh	r h	w	pull control GPIO39
17dt16	PR_GPIO40	xh	r h	w	pull control GPIO40
19dt18	PR_GPIO41	xh	r h	w	pull control GPIO41
21dt20	PR_GPIO42	xh	r h	w	pull control GPIO42
23dt22	PR_GPIO43	xh	r h	w	pull control GPIO43
25dt24	PR_GPIO44	xh	r h	w	pull control GPIO44
27dt26	PR_GPIO45	xh	r h	w	pull control GPIO45
29dt28	PR_GPIO46	xh	r h	w	pull control GPIO46

31dt3 0	PR_GPIO47	xh	r h	w	pull control GPIO47
------------	-----------	----	--------	---	---------------------

Register:		PULL63_48GPIO			Address: A0h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:		SCRB Pull GPIO Signals Each GPIO bit is set to the pull resistors on the basis of the following coding 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistors are automatically deactivated Reset value does not apply as the register is loaded in accordance with the function set with CONFIG.				
Bit	Identifier	Reset	Attr.	Function / Description		
1dt0	PR_GPIO48	xh	r h	w	pull control GPIO48	
3dt2	PR_GPIO49	xh	r h	w	pull control GPIO49	
5dt4	PR_GPIO50	xh	r h	w	pull control GPIO50	
7dt6	PR_GPIO51	xh	r h	w	pull control GPIO51	
9dt8	PR_GPIO52	xh	r h	w	pull control GPIO52	
11dt10	PR_GPIO53	xh	r h	w	pull control GPIO53	
13dt12	PR_GPIO54	xh	r h	w	pull control GPIO54	
15dt14	PR_GPIO55	xh	r h	w	pull control GPIO55	
17dt16	PR_GPIO56	xh	r h	w	pull control GPIO56	
19dt18	PR_GPIO57	xh	r h	w	pull control GPIO57	
21dt20	PR_GPIO58	xh	r h	w	pull control GPIO58	

23dt2 2	PR_GPIO59	xh	r h	w	pull control GPIO59
25dt2 4	PR_GPIO60	xh	r h	w	pull control GPIO60
27dt2 6	PR_GPIO61	xh	r h	w	pull control GPIO61
29dt2 8	PR_GPIO62	xh	r h	w	pull control GPIO62
31dt3 0	PR_GPIO63	xh	r h	w	pull control GPIO63

Register:		PULL79_64GPIO			Address:		A4h
Bits:		31dt0	Reset value:	0h	Attributes:	rh	w
Description:		SCRB Pull GPIO Signals Each GPIO bit is set to the pull resistors on the basis of the following coding 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistors are automatically deactivated Reset value does not apply as the register is loaded in accordance with the function set with CONFIG.					
Bit	Identifier	Reset	Attr.	Function / Description			
1dt0	PR_GPIO64	xh	r h	w	pull control GPIO64		
3dt2	PR_GPIO65	xh	r h	w	pull control GPIO65		
5dt4	PR_GPIO66	xh	r h	w	pull control GPIO66		
7dt6	PR_GPIO67	xh	r h	w	pull control GPIO67		
9dt8	PR_GPIO68	xh	r h	w	pull control GPIO68		
11dt1 0	PR_GPIO69	xh	r h	w	pull control GPIO69		
13dt1 2	PR_GPIO70	xh	r h	w	pull control GPIO70		

15dt1 4	PR_GPIO71	xh	r h	w	pull control GPIO71
17dt1 6	PR_GPIO72	xh	r h	w	pull control GPIO72
19dt1 8	PR_GPIO73	xh	r h	w	pull control GPIO73
21dt2 0	PR_GPIO74	xh	r h	w	pull control GPIO74
23dt2 2	PR_GPIO75	xh	r h	w	pull control GPIO75
25dt2 4	PR_GPIO76	xh	r h	w	pull control GPIO76
27dt2 6	PR_GPIO77	xh	r h	w	pull control GPIO77
29dt2 8	PR_GPIO78	xh	r h	w	pull control GPIO78
31dt3 0	PR_GPIO79	xh	r h	w	pull control GPIO79

Register:		PULL95_80GPIO			Address:		A8h
Bits:		31dt0	Reset value:	0h	Attributes:	rh	w
Description:		SCRB Pull GPIO Signals Each GPIO bit is set to the pull resistors on the basis of the following coding 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistors are automatically deactivated Reset value does not apply as the register is loaded in accordance with the function set with CONFIG.					
Bit	Identifier	Reset	Attr.	Function / Description			
1dt0	PR_GPIO80	xh	r h	w	pull control GPIO80		
3dt2	PR_GPIO81	xh	r h	w	pull control GPIO81		
5dt4	PR_GPIO82	xh	r h	w	pull control GPIO82		

7dt6	PR_GPIO83	xh	r	w	pull control GPIO83
9dt8	PR_GPIO84	xh	r	w	pull control GPIO84
11dt10	PR_GPIO85	xh	r	w	pull control GPIO85
13dt12	PR_GPIO86	xh	r	w	pull control GPIO86
15dt14	PR_GPIO87	xh	r	w	pull control GPIO87
17dt16	PR_GPIO88	xh	r	w	pull control GPIO88
19dt18	PR_GPIO89	xh	r	w	pull control GPIO89
21dt20	PR_GPIO90	xh	r	w	pull control GPIO90
23dt22	PR_GPIO91	xh	r	w	pull control GPIO91
25dt24	PR_GPIO92	xh	r	w	pull control GPIO92
27dt26	PR_GPIO93	xh	r	w	pull control GPIO93
29dt28	PR_GPIO94	xh	r	w	pull control GPIO94
31dt30	PR_GPIO95	xh	r	w	pull control GPIO95

Register:		CLEAR_DMA_SPI_REQ_REG		Address:		ACh	
Bits:		31dt0	Reset value:	0h	Attributes:	rh	w
Description:		Reset SPI1/2_SSPTX_Delayed_Request (SPI1/2 pause mode)					
Bit	Identifier	Reset	Attr.	Function / Description			
0	CLEAR_SPI1_SSPTX_DELAYED_REQUEST	xh	r h w	1: SPI1_SSPTX_Delayed_Request reset in GDMA shell, non-retentive			
1	CLEAR_SPI2_SSPTX_DELAYED_REQUEST	xh	r h w	1: SPI2_SSPTX_Delayed_Request reset in GDMA shell, non-retentive			

Register:	SPI_PARITY_ERROR			Address:	B0h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	SPI1/2 parity error (there is only a parity check when SPI data are read in the address range with APB address bits(10dt9) = 01)					
Bit	Identifier	Reset	Attr.	Function / Description		
0	SPI1_PARITY_ERR	0h	rh	w	1: Parity-Error SPI1 (Even Parity)	
1	SPI2_PARITY_ERR	0h	rh	w	1: Parity-Error SPI2 (Even Parity)	

Register:	XHIF_MODE			Address:	B8h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Umschalten XHIF-Pin zwischen XHIF_A20 und XHIF_XCS_R					
Bit	Identifier	Reset	Attr.	Function / Description		
0	XHIF_MODE	0h	r	w	The input pin XHIF_XCS_R_A20 is used as 0: Page register chip select (XHIF_XCS_R) 1: Address line (XHIF_A20) (The XHIF input not used is switched to inactive, i.e. XHIF_A20='0', XHIF_XCS_R='1')	

Register:	EXT_DRIVER_EN			Address:	BCh	
Bits:	31dt0	Reset value:	0h	Attributes:	r(h)	w
Description:						
Bit	Identifier	Reset	Attr.	Function / Description		
0	CS0_ENABLE	0h	rh	w	1: CS0 activates XOE_DRIVER 0: CS0 has no effect on XOE_DRIVER When the reset is cleared, the inverted value of XRDY_BF is latched.	

1	CS1_ENABLE	0h	r	w	1: CS1 activates XOE_DRIVER 0: CS1 has no effect on XOE_DRIVER
2	CS2_ENABLE	0h	r	w	1: CS2 activates XOE_DRIVER 0: CS2 has no effect on XOE_DRIVER
3	CS3_ENABLE	0h	r	w	1: CS3 activates XOE_DRIVER 0: CS3 has no effect on XOE_DRIVER

Register:		SPI_MODE			Address:		C0h
Bits:		31dt0	Reset value:	0h	Attributes:	r	w
Description:		Toggling SPI outputs between direct mode and MISO_MOSI mode					
Bit	Identifier	Reset	Attr.	Function / Description			
0	SPI1_MODE	0h	r w	0: DIRECT Mode for SPI1 1: MISO_MOSI Mode for SPI1			
1	SPI2_MODE	0h	r w	0: DIRECT Mode for SPI2 1: MISO_MOSI Mode for SPI2			

Register:		SD_CONTROL			Address:		C4h
Bits:		31dt0	Reset value:	0h	Attributes:	r	w
Description:		Differentiating and filtering the SignalDetect signal (SD signal) of an external fiber-optic transceiver for the integrated PHY over the ERTEC die					
Bit	Identifier	Reset	Attr.	Function / Description			
0	SD1_ENABLE	0h	r w	Differentiation/filtering of the SD1 signal for port 1 is 0: direct, over the balls for the integrated PHY 1: over ERTEC die for the integrated PHY			
7dt1	reserved						
14dt8	SD1_MUX	00h	r w	only relevant when SD1_ENABLE = '1' 0 ...95: GPIO pin 0...95 over which the SD1 signal of the fiber-optic transceiver is received.			

				96...112, no GPIO selection, input signal of '0' is assumed 113...127, no GPIO selection, input signal of '1' is assumed
15	reserved			
16	SD2_ENABLE	0h	r w	Differentiation/filtering of the SD2 signal for port 2 is 0: direct, over the balls for the inte- grated PHY 1: over ERTEC die for the integrated PHY
23dt1 7	reserved			
30dt2 4	SD2_MUX	00h	r w	only relevant when SD2_ENABLE = '1' 0 ...95: GPIO pin 0...95 over which the SD2 signal of the fiber-optic transceiver is received. 96...112, no GPIO selection, input signal of '0' is assumed 113...127, no GPIO selection, input signal of '1' is assumed

2.4 Memory Mapping

2.4.1 Memory Mapping ARM926-I

Seg	Address-Range	Memory/ Peripheral	Size /Byte	Description
0	0x0000_0000h ... 0x0000_1FFFh 0x03FF_FFFFh	← Boot ROM (8 KByte) ← EMC (64 MByte)	8 KByte 64 MByte	MEM_SWAP parameter assignment: 00b: -> Boot ROM (0 - 8 KByte), imaged ⁷ . 01b: -> EMC SDRAM (0 - 64 MByte) 10b: -> EMC asyn memory (0 - 64 MByte) 11b: -> QVZ Error ARM926 I-TCM (0 – 256 KByte / Step 64 KByte) ⁸
	0x0400_0000h ... 0x0FFF_FFFFh	<i>Not used</i>	<i>192 MByte</i>	
1	0x1000_0000h ... 0x1FFF_FFFFh	<i>Not used</i>	<i>256 MByte</i>	
2	0x2000_0000h ... 0x2FFF_FFFFh	EMC SDRAM	256 MByte	
3	0x3000_0000h ... 0x3FFF_FFFFh	EMC Asyn Memory	256 MByte	
	0x4000_0000h ... 0x4000_7FFFh	<i>Not used</i>	<i>32 KByte</i>	
	0x4000_8000h ... 0x4000_9FFFh	Boot ROM	8 KByte	
	0x4000_A000h ... 0x4FFF_FFFFh	<i>Not used</i>	<i>< 256 MByte</i>	
5-F	0x5000_0000h ... 0xFFFF_FFFFh	<i>Not used</i>	<i>2816 MByte</i>	

⁷ After a reset, the boot ROM is at address 0. The first 64 MByte of EMC-SDRAM or EMC asyn memory (chip select Bank0) can also be mapped to address 0 with the MEM_SWAP register in the SCRB (see 2.3.10.9.22).

⁸ The ARM926 I-TCM is 0 - 256 KByte (adjustable in 64 KByte increments) and can be displayed by the SW in the ARM926 coprocessor register CP15 c9 at address 0 or at another address in segment 0 (except in the D-TCM926 / D-TCM966 range). The I-TCM is disabled after reset. After display, ARM926 accesses its I-TCM and not the AHB in this address range (ARM926 cannot access its I-TCM over the AHB). Please note that ARM926 I-TCM can only be displayed in the ARM address range in increments of 2ⁿ. If, for example, a physical size of I-TCM = 192 KByte has been selected in 'TCM926_Map Register', ARM926 can only assign I-TCM an address range of 256 KByte.

2.4.2 ARM926-D Memory Mapping

Seg	Address-Range	Memory/ Peripheral	Size /Byte	Description
0	0x0000_0000h ... 0x0000_1FFFh 0x03FF_FFFFh	← Boot ROM (8 KByte) ← EMC (64 MByte)	8 KByte 64 MByte	MEM_SWAP parameter assignment: 00b: -> Boot ROM (0 - 8 KByte), imaged ⁹ 01b: -> EMC-SDRAM (0 - 64 MByte) ¹⁰ 10b: -> EMC asyn memory (0 - 64 MByte) 11b: -> QVZ Error
	0x0404_0000h ... 0x07FF_FFFFh	Not used	< 64 MByte	
	0x0800_0000h ... 0x0803_FFFFh	ARM926 D-TCM	256 KByte	ARM 926 D-TCM (0 – 256 KByte physical. / Step 64 KByte; not imaged) ¹¹ ;
	0x0804_0000h ... 0x0FFF_FFFFh	Not used	< 128 MByte	
	0x1000_0000h ... 0x110F_FFFFh 0x1110_0000h ... 0x1FFF_FFFFh	AHB peripherals Not used	17 MByte 239 MByte	
2	0x2000_0000h ... 0x2FFF_FFFFh	EMC SDRAM (256 MByte)	256 MByte	
3	0x3000_0000h ... 0x3FFF_FFFFh	EMC Asyn Memory	256 MByte	
4	0x4000_0000h ... 0x405F_FFFFh	APB peripherals	128 KByte	
	0x4100_0000h ... 0x4FFF_FFFFh	Not used	< 256 MByte	
5-F	0x5000_0000h ... 0xFFFF_FFFFh	Not used	2816 MByte	

⁹ After a reset, the boot ROM is at address 0. The first 64 MByte of EMC-SDRAM or EMC asyn memory (chip select Bank0) can also be mapped to address 0 with the MEM_SWAP register in the SCRB (see 2.3.10.9.22).

¹⁰ After a reset, no memory is assigned to address 0. The first 64 MByte of EMC-SDRAM or EMC asyn memory (chip select Bank0) can also be mapped to address 0 with the MEM_SWAP register in the SCRB (see 2.3.10.9.22).

¹¹ The ARM926 D-TCM is 0 - 256 KByte (adjustable in 64 KByte increments) and can be shown at address 0x0800 0000h by the SW in the ARM926 coprocessor register CP15 c9. ARM926 then accesses its D-TCM and not the AHB in this address range. Please note that ARM926 D-TCM can only be displayed in the ARM address area in increments of 2ⁿ. If, for example, a physical size of D-TCM = 192 KByte has been selected in 'TCM926_Map Register', ARM926 can only assign D-TCM an address range of 256 KByte.

2.4.3 PN-IP Memory Mapping

Seg	Address-Range	Memory/ Peripheral	Size /Byte	Description
0	0x0000_0000h ... 0x0000_1FFFh 0x03FF_FFFFh	← Boot ROM (8 KByte) ← EMC (64 MByte)	8 KByte 64 MByte	MEM_SWAP parameter assignment: 00b: -> Boot ROM (0 - 8 KByte), imaged 01b: -> EMC SDRAM (0 - 64 MByte) 10b: -> EMC asyn memory (0 - 64 MByte) 11b: -> QVZ Error
	0x0404_0000h ... 0x07FF_FFFFh	Not used	< 64 MByte	
	0x0800_0000h ... 0x0803_FFFFh	ARM926 D-TCM	256 KByte	ARM 926 D-TCM (0 – 256 KByte physical. / Step 64 KByte; not imaged);
	0x0804_0000h ... 0x0FFF_FFFFh	Not used	< 128 MByte	
	0x1000_0000h ... 0x107F_FFFFh	Not used	8 MByte	
1	0x1080_0000h ... 0x109F_FFFFh	PerIF (consistency buffer)	2 MByte	
	0x10A0_0000h ... 0x1FFF_FFFFh	Not used	246 MByte	
	0x2000_0000h ... 0x2FFF_FFFFh	EMC SDRAM (256 MByte)	256 MByte	
3	0x3000_0000h ... 0x3FFF_FFFFh	EMC Asyn Memory	256 MByte	
4-F	0x4000_0000h ... 0xFFFF_FFFFh	Not used	3072 MByte	

2.4.4 Host IF Memory Mapping

Seg	Address-Range	Memory/ Peripheral	Size /Byte	Description
0	0x0000_0000h ... 0x0000_1FFFh 0x03FF_FFFFh	← Boot ROM (8 KByte) ← EMC (64 MByte)	8 KByte 64 MByte	MEM_SWAP parameter assignment: 00b: -> Boot ROM (0 - 8 KByte), imaged ¹² 01b: -> EMC SDRAM (0 - 64 MByte) 10b: -> EMC asyn memory (0 - 64 MByte) 11b: -> QVZ Error
	0x0404_0000h ... 0x07FF_FFFFh	Not used	< 64 MByte	
	0x0800_0000h ... 0x0803_FFFFh	ARM926 D-TCM	256 KByte	ARM 926 D-TCM (0 – 256 KByte physical. / Step 64 KByte; not imaged);
	0x0804_0000h ... 0x0FFF_FFFFh	Not used	< 128 MByte	
1	0x1000_0000h ... 0x105F_FFFFh	Not used	6 MByte	
	0x1060_0000h ... 0x107F_FFFFh	PN-IP	2 MByte	
	0x1080_0000h ... 0x109F_FFFFh	PerIF (consistency buffer)	2 MByte	
	0x10A0_0000h ... 0x10AF_FFFFh	GDMA (register, JOB SRAM)	1 MByte	
	0x10C0_0000h ... 0x10CF_FFFFh	Not used	1 MByte	
	0x10D0_0000h ... 0x10DF_FFFFh	EMC register	1 MByte	
	0x10E0_0000h ... 0x10FF_FFFFh	Not used	2 MByte	
	0x1110_0000h ... 0x1FFF_FFFFh	Not used	239 MByte	
2	0x2000_0000h ... 0x2FFF_FFFFh	EMC SDRAM (256 MByte)	256 MByte	
3	0x3000_0000h ... 0x3FFF_FFFFh	EMC asyn memory	256 MByte	
4	0x4000_0000h ... 0x405F_FFFFh	APB peripherals	128 KByte	
	0x4100_0000h ... 0x4FFF_FFFFh	Not used	< 256 MByte	
5-F	0x5000_0000h ... 0xFFFF_FFFFh	Not used	2816 MByte	

¹² The boot ROM can be accessed from the host but not used from an applicative perspective.

For booting from an external host, the primary boot loader must create a page for the D-TCM (boot RAM) in the page registers of the HostIF (2x XHIF). The remaining pages can then be set up specifically by the external host. It makes sense to set up additional pages for PN-IP, PER-IF, EMC-SDRAM / EMC-SRAM, GDMA and APB peripherals .

The following table shows the paging register set within the XHIF module.			Address			Default
Name	CPU-16bit data width	Description	Name	CPU-32bit data width	Description	
XHIF_P0_RG_L	00h	Lower 16 bit of the range configuration 15:8 write- and readable 7:0 only readable	XHIF_P0_RG	00h	32 bit of the range configura- tion	00h
XHIF_P1_RG_L	10h		XHIF_P1_RG	10h	31:22 only readable	00h
XHIF_P2_RG_L	20h		XHIF_P2_RG	20h	21:8 write and readable	00h
XHIF_P3_RG_L	30h		XHIF_P3_RG	30h	7:0 only readable	00h
XHIF_P0_RG_H	02h	Upper 16 bit of the range configuration 15:6 only readable 5:0 write- and readable				00h
XHIF_P1_RG_H	12h					00h
XHIF_P2_RG_H	22h					00h
XHIF_P3_RG_H	32h					00h
XHIF_P0_OF_L	04h	Lower 16 bit of the offset configuration 15:8 write- and readable 7:0 only readable	XHIF_P0_OF	04h	32 bit of the offset configura- tion	00h
XHIF_P1_OF_L	14h		XHIF_P1_OF	14h	31:8 write- and readable	00h
XHIF_P2_OF_L	24h		XHIF_P2_OF	24h	7:0 only readable	00h
XHIF_P3_OF_L	34h		XHIF_P3_OF	34h		00h
XHIF_P0_OF_H	06h	Upper 16 bit of the offset configuration 15:0 read/writeable				00h
XHIF_P1_OF_H	16h					00h
XHIF_P2_OF_H	26h					00h
XHIF_P3_OF_H	36h					00h
XHIF_P0_CFG	08h	Config. of the buffering mode for each single page 1: Page is 32-Bit Page 0: Page is 16-Bit Page	XHIF_P0_CFG	08h	Config. of the buffering mode for each single page 1: Buffering mode on read enabled 0 : Buffering mode on read disabled	00h
XHIF_P1_CFG	18h		XHIF_P1_CFG	18h		00h
XHIF_P2_CFG	28h		XHIF_P2_CFG	28h		00h
XHIF_P3_CFG	38h		XHIF_P3_CFG	38h		00h
XHIF_VERSION_L	3Ch	Version of the XHIF	XHIF_VERSION	3Ch	IA&DT IP version compliant	
XHIF_VERSION_H	3Eh					

The 8-page register of HostIF (2x XHIF with 4 pages each) can be configured either over the APB or externally with the XHIF_XCR_R_A20 signal (default after reset). The selection of the two XHIF modules is controlled by the pin XHIF_SEG_2 (= 0b → XHIF0, = 1b → XHIF1).

2.4.5 GDMA Memory Mapping

Seg	Address-Range	Memory/ Peripheral	Size /Byte	Description
0	0x0000_0000h ... 0x0000_1FFFh 0x03FF_FFFFh	← Boot ROM (8 KByte) ← EMC (64 MByte)	8 KByte 64 MByte	MEM_SWAP parameter assignment: 00b: -> Boot ROM (0 - 8 KByte), imaged ¹³ 01b: -> EMC SDRAM (0 - 64 MByte) 10b: -> EMC asyn memory (0 - 64 MByte) 11b: -> QVZ Error
	0x0404_0000h ... 0x07FF_FFFFh	Not used	< 64 MByte	
	0x0800_0000h ... 0x0803_FFFFh	ARM926 D-TCM	256 KByte	ARM 926 D-TCM (0 – 256 KByte physical. / Step 64 KByte; not imaged);
	0x0804_0000h ... 0x0FFF_FFFFh	Not used	< 128 MByte	
1	0x1000_0000h ... 0x107F_FFFFh	Not used	8 MByte	
	0x1080_0000h ... 0x109F_FFFFh	PerIF (consistency buffer)	2 MByte	
	0x10A0_0000h ... 0x10AF_FFFFh	GDMA (register, JOB SRAM)	1 MByte	
	0x10B0_0000h ... 0x10BF_FFFFh	Not used	1 MByte	
	0x10D0_0000h ... 0x10DF_FFFFh	EMC register	1 MByte	
	0x10E0_0000h ... 0x10EF_FFFFh	ETB11 memory	1 MByte	
	0x10F0_0000h ... 0x10FF_FFFFh	ETB11 register	1 MByte	
	0x1110_0000h ... 0x1FFF_FFFFh	Not used	239 MByte	
2	0x2000_0000h ... 0x2FFF_FFFFh	EMC SDRAM (256 MByte)	256 MByte	
3	0x3000_0000h ... 0x3FFF_FFFFh	EMC asyn memory	256 MByte	
4	0x4000_0000h ... 0x405F_FFFFh	APB peripherals	128 KByte	
	0x4100_0000h ... 0x4FFF_FFFFh	Not used	< 256 MByte	
5-F	0x5000_0000h ... 0xFFFF_FFFFh	Not used	2816 MByte	

¹³ The boot ROM can be accessed from the GDMA but not used from an applicative perspective.

2.5 Detailed Address Mapping

Seg	Address-Range	Memory/ Peripheral	Size /Byte	Description
ARM926 subsystem				
0	0x0000_0000h ... 0x0000_1FFFh 0x03FF_FFFFh	← Boot ROM (8 KByte) ← EMC (64 MByte)	8 KByte 64 MByte	MEM_SWAP parameter assignment: 00b: -> Boot ROM (0 - 8 KByte), imaged ¹⁴ 01b: -> EMC SDRAM (0 - 64 MByte) 10b: -> EMC asyn memory (0 - 64 MByte) 11b: -> QVZ Error
	0x0000_0000h ... 0x0003_FFFFh	ARM926 I-TCM ¹⁵	256 KByte	After ARM926 I-TCM activation (in the CP15 c9 register): ARM 926 I-TCM (0 – 256 KByte physical. / Step 64 KByte; not imaged);
	0x0404_0000h ... 0x07FF_FFFFh	Not used	< 64 MByte	
	0x0800_0000h ... 0x0803_FFFFh	ARM926 D-TCM	256 KByte	ARM 926 D-TCM (0 – 256 KByte physical. / Step 64 KByte; not imaged);
	0x0804_0000h ... 0x0FFF_FFFFh	Not used	< 128 MByte	
AHB peripherals				
1	0x1000_0000h ... 0x105F_FFFFh	ARM-ICU	6 MByte	ARM926 Interrupt Controller; 5 MByte physical; 2 ⁰ * imaged;
	0x1060_0000h ... 0x107F_FFFFh	PN-IP	2 MByte	2 MByte physical; 2 ⁰ * imaged;
	0x1080_0000h ... 0x109F_FFFFh	PerIF (consistency buffer)	2 MByte	com. access: 128 KByte physical; 2 ⁴ * imaged appl. access: 64 KByte physical; 2 ⁵ * imaged
	0x10A0_0000h ... 0x10AF_FFFFh	GDMA (register, JOB SRAM)	1 MByte	GDMA Register and internal GDMA Job SRAM (Size: 4608 bytes) 1 MByte physical; 2 ⁰ * imaged; Register: 10A0_0000..10A0_00AF JOB SRAM: 10A0_00B0..10A0_12AF Not used: 10A0_12B0..10AF_FFFF
	0x10D0_0000h ... 0x10DF_FFFFh	EMC register	1 MByte	EMC register; 1 MByte physical; not imaged;
	0x10E0_0000h ... 0x10EF_FFFFh	ETB11 memory	1 MByte	ETB11 memory; 8 KByte physical; 2 ⁷ * imaged;
	0x10F0_0000h ... 0x10FF_FFFFh	ETB11 register	1 MByte	ETB11 register; 512 byte physical; 2 ¹¹ * imaged;
	0x1100_0000h ... 0x110F_FFFFh	Not used	1 MByte	1 MByte physical; not imaged;
	0x1110_0000h	Not used	239 MByte	

¹⁴ After a reset, the boot ROM is at address 0. The first 64 MByte of EMC-SDRAM or EMC asyn memory (chip select Bank0) can also be mapped to address 0 with the MEM_SWAP register in the SCRB (see 2.3.10.9.22).

¹⁵ ARM926 I-TCM can only be accessed internally from the ARM926 and not over the AHB. After activation, this address range is no longer shown at the AHB. This range can also be shown at other addresses with alignment in segment 0, however not in the ARM926 D-TCM or ARM966 D-TCM area.

	0x1FFF_FFFFh			
Synchronous External RAM Area				
2	0x2000_0000h ... 0x2FFF_FFFFh	EMC SDRAM (256 MByte)	256 MByte	256 MByte physical;
Asynchronous External RAM Area				
3	0x3000_0000h ... 0x33FF_FFFFh	EMC asyn memory Chip Select Bank 0	64 MByte	64 MByte physical; (if less is selected, the area is mirrored)
	0x3400_0000h ... 0x37FF_FFFFh	EMC asyn memory Chip Select Bank 1	64 MByte	64 MByte physical; (if less is selected, the area is mirrored)
	0x3800_0000h ... 0x3BFF_FFFFh	EMC asyn memory Chip Select Bank 2	64 MByte	64 MByte physical; (if less is selected, the area is mirrored)
	0x3C00_0000h ... 0x3FFF_FFFFh	EMC asyn memory Chip Select Bank 3	64 MByte	64 MByte physical; (if less is selected, the area is mirrored)
APB peripherals				
4	0x4000_0000h ... 0x4000_7FFFh	PerIF (Register, RAM)	32 KByte	32 KByte physical;
	0x4000_8000h ... 0x4000_9FFFh	Boot ROM	8 KByte	8 KByte physical;
	0x4000_A000h ... 0x4000_AFFFh	UART1	4 KByte	4 KByte physical;
	0x4000_B000h ... 0x4000_BFFFh	UART2	4 KByte	4 KByte physical;
	0x4000_C000h ... 0x4000_CFFFh	UART3	4 KByte	4 KByte physical;
	0x4000_D000h ... 0x4000_DFFFh	UART4	4 KByte	4 KByte physical;
	0x4000_E000h ... 0x4000_EFFFh	I²C	4 KByte	256 byte physical; imaged
	0x4000_F000h ... 0x4000_FFFFh	SCRB	4 KByte	256 byte physical; imaged
	0x4001_0000h ... 0x4001_00FFh	SPI1	256 byte	No 8/16-bit reversal of endianness No parity generation/check
	0x4001_0100h ... 0x4001_01FFh		256 byte	8-bit reversal of endianness No parity generation/check
	0x4001_0200h ... 0x4001_02FFh		256 byte	No 8-bit reversal of endianness Parity generation/check
	0x4001_0300h ... 0x4001_03FFh		256 byte	8-bit reversal of endianness Parity generation/check
	0x4001_0400h ... 0x4001_04FFh		256 byte	16-bit reversal of endianness No parity generation/check
	0x4001_0500h ... 0x4001_0FFFh		2.75 KByte	reserved corresponds to 0x4001_0000 - 0x4001_00FF
	0x4001_1000h	SPI2	256 byte	No 8/16-bit reversal of endianness

	...			No parity generation/check
	0x4001_10FFh			
	0x4001_1100h		256 byte	8-bit reversal of endianness No parity generation/check
	...			
	0x4001_11FFh			
	0x4001_1200h		256 byte	No 8-bit reversal of endianness Parity generation/check
	...			
	0x4001_12FFh			
	0x4001_1300h		256 byte	8-bit reversal of endianness Parity generation/check
	...			
	0x4001_13FFh			
	0x4001_1400h		256 byte	16-bit reversal of endianness No parity generation/check
	...			
	0x4001_14FFh			
	0x4001_1500h			
	...		2.75 KByte	reserved corresponds to 0x4001_1000 - 0x4001_10FF
	0x4001_1FFFh			
	0x4001_2000h	Timer 0 – 5	4 KByte	256 bytes physical; imaged;
	...	Watchdog		
	0x4001_2FFFh		4 KByte	32 KByte physical; imaged;
	0x4001_3000h	F-counter		
	...			
	0x4001_3FFFh		4 KByte	8 bytes physical; imaged;
	0x4001_4000h			
	...			
	0x4001_4FFFh			
	0x4001_8000h	GPIO		
	...		4 KByte	256 bytes physical; imaged;
	0x4001_8FFFh	I-Filter		
	0x4001_9000h		4 KByte	128 bytes physical; imaged;
	...	HostIF (Host Interface)		
	0x4001_9FFFh			
	0x4001_A000h		4 KByte	256 bytes physical; imaged;
	...			
	0x4001_AFFFh			
	0x4002_0000h	Not used		
	...		< 256 MByte	
	0x4FFF_FFFFh			
Unused				
5-F	0x5000_0000h	Not used	2816 MByte	
	...			
	0xFFFF_FFFFh			

2.6 Register Description

All bits marked "reserved" in the register descriptions should be assigned a value of 0 by the SW/FW in write access to allow future extensions. These bits must not be evaluated in read access.

If no specific information is given, the registers can be changed during operation.

All register descriptions are taken from the XRSL descriptions of the individual modules.
The registers used are described in the individual function blocks and in the relevant IP specifications.

3 IO INTERFACE

3.1 Overview

Description	Voltage	BGA Balls
Functional signals		
Standard signals	3.3V	45
XHIF/GPIOs	3.3V/1.8V	64
EMC	1.8V	83
	Subtotal	192
Power-Supply		
VDD_XHIF	3.3V/1.8V	12
VDD33	3.3V	17
VDD18 (EMC)	1.8V	15
VDD15	1.5V	9
VDD12	1.2V	16
GND		82
	Subtotal	151
PHY Interface		
AVDD 3,3V (voltage supply)	3.3V (analog)	2
AVDD 1,5V (voltage supply)	1.5V (analog)	3
AGND (voltage supply – ground)		6
DVDD 1,5V (voltage supply)	1.5V (digital)	2
DGND (voltage supply – ground)		2
VDDY (PECL)	analog	2
MDI - TX	analog	9
MDI - FX	3.3V	14
ATP (Test Function)	analog	1
TESTDOUT	3.3V	3
	Subtotal	44
PLL 500 MHz		
AVDD (voltage supply - connects to VOUT10)	1.2V	1
AGND (voltage supply – ground)		1
	Subtotal	2
Oscillator		
CLKP_A(in)	3.3V	1
CLKP_B(out)	3.3V	1
REF_CLK, BYP_CLK	3.3V	2
	Subtotal	4
REE Test pins		
TMC1, TMC2 (inputs)	3.3V	2
TEST, TACT	3.3V	2
	Subtotal	4
CTRL-STBY pins		
CTRL_STBY0 (inputs)	3.3V	1
CTRL_STBY1 (inputs)	1.8V/3.3V	1
CTRL_STBY2 (inputs)	1.8V/3.3V	1
	Subtotal	3
	Total	400

Table 24: IO – pin count overview

3.2 Detailed Signal Description

Active level low = Signal name X....;

*) Driver power of the EMC signals can be set on a group-specific basis (G1 – G11) in the SCRB register SCRB_DRIVER_EMC (see 2.3.10.9.15.1)

**) Driver power of GPIO(31:0) and XHIF interface (GPIO95:32); can be set on a signal-specific basis with the SCRB_DRIVEx_yGPIO SCRB registers (see 2.3.10.9.15.1)

**) Pull circuit for GPIO(31:0) and XHIF interface (GPIO95:32); can be set on a signal-specific basis with the SCRB_PULLx_yGPIO SCRB registers (see 2.3.10.9.15.2)

ERTEC 200P	CTRL _{STBYx}	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
EMC														
A0	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 0 <i>REE DFT Input</i>	1
A1	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 1 <i>REE DFT Input</i>	1
A2	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 2 <i>REE DFT Input</i>	1
A3	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 3 <i>REE DFT Input</i>	1
A4	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 4 <i>REE DFT Input</i>	1
A5	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 5 <i>REE DFT Input</i>	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
A6	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 6 <i>REE DFT Input</i>	1
A7	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 7 <i>REE DFT Input</i>	1
A8	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 8 <i>REE DFT Input</i>	1
A9	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 9 <i>REE DFT Input</i>	1
A10	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 10 <i>REE DFT Input</i>	1
A11	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 11 <i>REE DFT Input</i>	1
A12	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 12 <i>REE DFT Input</i>	1
A13	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 13 <i>REE DFT Input</i>	1
A14	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G2 ^{*)}	42	-	42	0.1		-	EMC Address Bus Pin 14 <i>REE DFT Input</i>	1
A15	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G3 ^{*)}	42	DN	42	0.1		-	EMC Address Bus Pin 15 <i>Boot(2)</i>	1
A16	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G3 ^{*)}	42	UP	42	0.1		-	EMC Address Bus Pin 16 <i>Boot(3)</i>	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
A17	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G3 ^{*)}	42	UP	42	0.1		-	EMC Address Bus Pin 17 <i>Config(0)</i>	1
A18	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G3 ^{*)}	42	UP	42	0.1		-	EMC Address Bus Pin 18 <i>Config(1)</i>	1
A19	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G3 ^{*)}	42	DN	42	0.1		-	EMC Address Bus Pin 19 <i>Config(2)</i>	1
A20	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G3 ^{*)}	42	UP	42	0.1		-	EMC Address Bus Pin 20 <i>Config(3)</i>	1
A21	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G3 ^{*)}	42	DN	42	0.1		-	EMC Address Bus Pin 21 <i>Config(4)</i>	1
A22	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G3 ^{*)}	42	DN	42	0.1		-	EMC Address Bus Pin 22 <i>Config(5)</i>	1
A23	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G3 ^{*)}	42	UP	42	0.1		-	EMC Address Bus Pin 23 <i>Config(6)</i>	1
D0	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 0	1
D1	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 1	1
D2	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 2	1
D3	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 3	1
D4	-	bidi	1V8	TWC1BC18ALV04SZ	bid	G4 ^{*)}	42	UP	62.5	0.1		-	EMC Data Bus Pin 4	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
					i					6				
D5	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 5	1
D6	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 6	1
D7	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 7	1
D8	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 8	1
D9	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 9	1
D10	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 10	1
D11	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 11	1
D12	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 12	1
D13	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 13	1
D14	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 14	1
D15	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 15	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
D16	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 16	1
D17	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 17	1
D18	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 18	1
D19	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 19	1
D20	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 20	1
D21	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 21	1
D22	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 22	1
D23	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 23	1
D24	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 24	1
D25	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 25	1
D26	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 26	1
D27	-	bidi	1V8	TWC1BC18ALV04SZ	bid	G5 ^{*)}	42	UP	62.5	0.1		-	EMC Data Bus Pin 27	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
					i					6				
D28	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 28	1
D29	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 29	1
D30	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 30	1
D31	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	UP	62.5	0.1 6		-	EMC Data Bus Pin 31	1
XBE0_DQM0	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	-	42	0.1		-	EMC Byte 0 Enable <i>REE DFT Input</i>	1
XBE1_DQM1	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G4 ^{*)}	42	-	42	0.1		-	EMC Byte 1 Enable <i>REE DFT Input</i>	1
XBE2_DQM2	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	-	42	0.1		-	EMC Byte 2 Enable <i>REE DFT Input</i>	1
XBE3_DQM3	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G5 ^{*)}	42	-	42	0.1		-	EMC Byte 3 Enable <i>REE DFT Input</i>	1
XCS_PER0	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G7 ^{*)}	42	-	1	1		-	EMC Bank 0 Chip Select <i>REE DFT Input</i>	1
XCS_PER1	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G7 ^{*)}	42	-	1	1		-	EMC Bank 1 Chip Select <i>REE DFT Input</i>	1
XCS_PER2	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G7 ^{*)}	42	-	1	1		-	EMC Bank 2 Chip Select <i>REE DFT Input</i>	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
XCS_PER3	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G7 ^{*)}	42	-	1	1		-	EMC Bank 3 Chip Select <i>REE DFT Input</i>	1
XWR	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G6 ^{*)}	42	-	42	0.1		-	EMC Write Signal <i>REE DFT Input</i>	1
XRD	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G6 ^{*)}	42	-	42	0.1		-	EMC Read Signal <i>REE DFT Input</i>	1
XRDY_PER	-	in	1V8	TWC1IC18AS	bid i	-	-	UP	-	-		-	EMC Ready Signal	1
DTXR	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G1 ^{*)}	42	DN	42	0.1		-	EMC Direction for ext. Driver <i>Boot(0)</i>	1
XOE_DRIVER	-	bid	1V8	TWC1BC18ALV04SZ	bid i	G1 ^{*)}	42	UP	1	1		-	EMC Enable for ext. Driver <i>Boot(1)</i>	1
CLK_O_SDRAM0	-	out	1V8	TWC1BC18ALV04SZ	bid i	G8 ^{*)}	42	-	125	1		-	Feedback clock output	1
CLK_O_SDRAM1	-	out	1V8	TWC1BC18ALV04SZ	bid i	G8 ^{*)}	42	-	125	1		-	clock for the SDRAM device	1
CLK_O_SDRAM2	-	out	1V8	TWC1BC18ALV04SZ	bid i	G8 ^{*)}	42	-	125	1		-	clock for the SDRAM device	1
CLK_I_SDRAM	-	in	1V8	TWC1IC18AS	bid i	-	-	-	-	-		-	Feedback clock for syn- chronization of read data. MUST be connected, even if SDRAM is not	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
													used!	
XCS_SDRAM	-	out	1V8	TWC1BC18ALV04SZ	bid i	G9 ^{*)}	42	-	5	0.1		-	Chip select for SDRAM	1
XRAS_SDRAM	-	out	1V8	TWC1BC18ALV04SZ	bid i	G9 ^{*)}	42	-	5	0.1		-	Row address strobe	1
XCAS_SDRAM	-	out	1V8	TWC1BC18ALV04SZ	bid i	G9 ^{*)}	42	-	5	0.1		-	Column address strobe	1
XWE_SDRAM	-	out	1V8	TWC1BC18ALV04SZ	bid i	G9 ^{*)}	42	-	5	0.1		-	Write enable for SDRAM	1
XAV_BF	-	bidi	1V8	TWC1BC18ALV04SZ	bid i	G11 [*])	42	UP	5	0.1		-	Address Valid BurstFlash <i>Boot(4)</i>	1
XRDY_BF	-	in	1V8	TWC1IC18AS		-	-	UP	-	-		-	Ready BurstFlash <i>EXT_DRIVER_DISABLE</i> <i>_CS0</i>	1
CLK_O_BF0	-	out	1V8	TWC1BC18ALV04SZ	bid i	G10 [*])	42	-	125	1		-	Feedback clock output	1
CLK_O_BF1	-	out	1V8	TWC1BC18ALV04SZ	bid i	G10 [*])	42	-	125	1		-	clock for the BurstFlash device	1
CLK_O_BF2	-	out	1V8	TWC1BC18ALV04SZ	bid i	G10 [*])	42	-	125	1		-	clock for the BurstFlash device	1
CLK_I_BF	-	in	1V8	TWC1IC18AS	bid i	-	-	-	-	-		-	Feedback clock for syn- chronization of read data. MUST be connected, even if BurstFlash is not	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
													used!	
sum														8 2
GPIO														
GPIO0_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 0 (interruptible) <i>I-filter: IN_Delay_0</i>	1
GPIO1_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 1 (interruptible) <i>I-filter: IN_Delay_1</i>	1
GPIO2_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 2 (interruptible) <i>I-filter: IN_Delay_2</i>	1
GPIO3_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 3 (interruptible) <i>I-filter: IN_Delay_3</i>	1
GPIO4_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 4 (interruptible) <i>I-filter: IN_Delay_4</i>	1
GPIO5_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 5 (interruptible) <i>I-filter: IN_Delay_5</i>	1
GPIO6_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 6 (interruptible) <i>I-filter: IN_Delay_6</i>	1
GPIO7_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 7 (interruptible) <i>I-filter: IN_Delay_7</i>	1
GPIO8_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 8 (interruptible) <i>I-filter: IN_Delay_8</i>	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
GPIO9_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 9 (interruptible) <i>I-filter: IN_Delay_9</i>	1
GPIO10_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 10 (interruptible) <i>I-filter: IN_Delay_10</i>	1
GPIO11_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 11 (interruptible) <i>I-filter: IN_Delay_11</i>	1
GPIO12_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 12 (interruptible) <i>I-filter: IN_Delay_12</i>	1
GPIO13_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 13 (interruptible) <i>I-filter: IN_Delay_13</i>	1
GPIO14_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 14 (interruptible) <i>I-filter: IN_Delay_14</i>	1
GPIO15_INT	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 15 (interruptible) <i>I-filter: IN_Delay_15</i>	1
GPIO16	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 16	1
GPIO17	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 17	1
GPIO18	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 18	1
GPIO19	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 19	1
GPIO20	0	bidi	3V3	TWF1BC33ASLV04SZ	bid	**))	-	**))	-	-		ST	GPIO 20	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
					i									
GPIO21	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 21	1
GPIO22	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 22	1
GPIO23	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 23	1
GPIO24	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 24	1
GPIO25	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 25	1
GPIO26	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 26	1
GPIO27	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 27	1
GPIO28	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 28	1
GPIO29	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 29	1
GPIO30	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 30	1
GPIO31	0	bidi	3V3	TWF1BC33ASLV04SZ	bid i	**))	-	**))	-	-		ST	GPIO 31	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
sum														3 2
XHIF														
XHIF_A1	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 1 <i>I-filter: IN_Delay_16</i>	1
XHIF_A2	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 2 <i>I-filter: IN_Delay_17</i>	1
XHIF_A3	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 3 <i>I-filter: IN_Delay_18</i>	1
XHIF_A4	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 4 <i>I-filter: IN_Delay_19</i>	1
XHIF_A5	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 5 <i>I-filter: IN_Delay_20</i>	1
XHIF_A6	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 6 <i>I-filter: IN_Delay_21</i>	1
XHIF_A7	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 7 <i>I-filter: IN_Delay_22</i>	1
XHIF_A8	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 8 <i>I-filter: IN_Delay_23</i>	1
XHIF_A9	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 9 <i>I-filter: IN_Delay_24</i>	1
XHIF_A10	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 10 <i>I-filter: IN_Delay_25</i>	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
XHIF_A11	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 11 <i>I-filter: IN_Delay_26</i>	1
XHIF_A12	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 12 <i>I-filter: IN_Delay_27</i>	1
XHIF_A13	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 13 <i>I-filter: IN_Delay_28</i>	1
XHIF_A14	2	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 14 <i>I-filter: IN_Delay_29</i>	1
XHIF_A15	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 15 <i>I-filter: IN_Delay_30</i>	1
XHIF_A16	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 16 <i>I-filter: IN_Delay_31</i>	1
XHIF_A17	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 17 <i>I-filter: IN_Delay_32</i>	1
XHIF_A18	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 18 <i>I-filter: IN_Delay_33</i>	1
XHIF_A19	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 19 <i>I-filter: IN_Delay_34</i>	1
XHIF_SEG_0	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 20 <i>I-filter: IN_Delay_35</i>	1
XHIF_SEG_1	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Address Bus Pin 21 <i>I-filter: IN_Delay_36</i>	1
XHIF_SEG_2	1	in	3V3/1	TWF1ZE1410BASNV6SZ	bid	**))	-	***)	-	-		-	XHIF Address Bus Pin 22	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
			V8		i								<i>I-filter: IN_Delay_37</i>	
XHIF_XBE0	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**)	-	***)	-	-		-	XHIF Byte 0 Enable, low active <i>I-filter: IN_Delay_38</i>	1
XHIF_XBE1	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**)	-	***)	-	-		-	XHIF Byte 1 Enable, low active <i>I-filter: IN_Delay_39</i>	1
XHIF_XBE2	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**)	-	***)	-	-		-	XHIF Byte 2 Enable, low active <i>I-filter: IN_Delay_40</i>	1
XHIF_XBE3	2	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**)	-	***)	-	-		-	XHIF Byte 3 Enable, low active <i>I-filter: IN_Delay_41</i>	1
XHIF_XCS_M	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**)	-	***)	-	-		-	XHIF Chip Select for AHB access <i>I-filter: IN_Delay_42</i>	1
XHIF_XCS_R_A20	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**)	-	***)	-	-		-	XHIF Chip Select for Page Config. Registers <i>I-filter: IN_Delay_43</i>	1
XHIF_XRD	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**)	-	***)	-	-		-	XHIF Read Signal, low active <i>I-filter: IN_Delay_44</i>	1
XHIF_XWR	1	in	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**)	-	***)	-	-		-	XHIF Write Signal, low active	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
													<i>I-filter: IN_Delay_45</i>	
XHIF_XRDY	1	out	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Ready Signal, polarity adjustable <i>I-Ffilter: IN_Delay_46</i>	1
XHIF_XIRQ	1	out	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Interrupt Output <i>I-filter: IN_Delay_47</i>	1
XHIF_D0	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 0 <i>I-filter: IN_Delay_48</i>	1
XHIF_D1	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 1 <i>I-filter: IN_Delay_49</i>	1
XHIF_D2	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 2 <i>I-filter: IN_Delay_50</i>	1
XHIF_D3	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 3 <i>I-filter: IN_Delay_51</i>	1
XHIF_D4	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 4 <i>I-filter: IN_Delay_52</i>	1
XHIF_D5	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 5 <i>I-filter: IN_Delay_53</i>	1
XHIF_D6	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 6 <i>I-filter: IN_Delay_54</i>	1
XHIF_D7	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 7 <i>I-filter: IN_Delay_55</i>	1
XHIF_D8	1	bidi	3V3/1	TWF1ZE1410BASNV6SZ	bid	**))	-	***)	-	-		-	XHIF Data Bus Pin 8	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
			V8		i								<i>I-filter: IN_Delay_56</i>	
XHIF_D9	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	**))	-	-		-	XHIF Data Bus Pin 9 <i>I-filter: IN_Delay_57</i>	1
XHIF_D10	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	**))	-	-		-	XHIF Data Bus Pin 10 <i>I-filter: IN_Delay_58</i>	1
XHIF_D11	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	**))	-	-		-	XHIF Data Bus Pin 11 <i>I-filter: IN_Delay_59</i>	1
XHIF_D12	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	**))	-	-		-	XHIF Data Bus Pin 12 <i>I-filter: IN_Delay_60</i>	1
XHIF_D13	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	**))	-	-		-	XHIF Data Bus Pin 13 <i>I-filter: IN_Delay_61</i>	1
XHIF_D14	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	**))	-	-		-	XHIF Data Bus Pin 14 <i>I-filter: IN_Delay_62</i>	1
XHIF_D15	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	**))	-	-		-	XHIF Data Bus Pin 15 <i>I-filter: IN_Delay_63</i>	1
XHIF_D16	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	**))	-	-		-	XHIF Data Bus Pin 16 <i>I-filter: IN_Delay_64</i>	1
XHIF_D17	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	**))	-	-		-	XHIF Data Bus Pin 17 <i>I-filter: IN_Delay_65</i>	1
XHIF_D18	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	**))	-	-		-	XHIF Data Bus Pin 18 <i>I-filter: IN_Delay_66</i>	1
XHIF_D19	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	**))	-	-		-	XHIF Data Bus Pin 19 <i>I-filter: IN_Delay_67</i>	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
XHIF_D20	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 20 <i>I-filter: IN_Delay_68</i>	1
XHIF_D21	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 21 <i>I-filter: IN_Delay_69</i>	1
XHIF_D22	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 22 <i>I-filter: IN_Delay_70</i>	1
XHIF_D23	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 23 <i>I-filter: IN_Delay_71</i>	1
XHIF_D24	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 24 <i>I-filter: IN_Delay_72</i>	1
XHIF_D25	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 25 <i>I-filter: IN_Delay_73</i>	1
XHIF_D26	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 26 <i>I-filter: IN_Delay_74</i>	1
XHIF_D27	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 27 <i>I-filter: IN_Delay_75</i>	1
XHIF_D28	1	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 28 <i>I-filter: IN_Delay_76</i>	1
XHIF_D29	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 29 <i>I-filter: IN_Delay_77</i>	1
XHIF_D30	2	bidi	3V3/1 V8	TWF1ZE1410BASNV6SZ	bid i	**))	-	***)	-	-		-	XHIF Data Bus Pin 30 <i>I-filter: IN_Delay_78</i>	1
XHIF_D31	2	bidi	3V3/1	TWF1ZE1410BASNV6SZ	bid	**))	-	***)	-	-		-	XHIF Data Bus Pin 31	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
			V8		i								I-filter: IN_Delay_79	
sum														64
PHY LEDs														
L_PHY_1	0	out	3V3	TWF1BC33ALV04SZ	bid i	8	20	-	0.1	0.1		-	Link LED PHY Port 1	1
A_PHY_1	0	out	3V3	TWF1BC33ALV04SZ	bid i	8	20	-	0.1	0.1		-	Activity LED PHY Port 1	1
L_PHY_2	0	out	3V3	TWF1BC33ALV04SZ	bid i	8	20	-	0.1	0.1		-	Link LED PHY Port 2	1
A_PHY_2	0	out	3V3	TWF1BC33ALV04SZ	bid i	8	20	-	0.1	0.1		-	Activity LED PHY Port 2	1
sum														4
PHY MDIO														
P1RXN	-	AIO	analog	NBTHPHYC2GV02	IO	18	0	-	125	1		-	Port1 differential receive input	1
P1RXP	-	AIO	analog	NBTHPHYC2GV02	IO	18	0	-	125	1		-		1
P1TXN	-	AIO	analog	NBTHPHYC2GV02	IO	18	0	-	125	1		-	Port1 differential transmit output	1
P1TXP	-	AIO	analog	NBTHPHYC2GV02	IO	18	0	-	125	1		-		1
P1RDXP	-	in	3V3	TDIPAPECNR	in	-	-	-	125	1		-	Port1 FX differential receive input (PECL)	1
P1RDXN	-	in	3V3	TDIPAPECNR	in	-	-	-	125	1		-		1
P1TDXP	-	out	3V3	TDOTAC33NN12	ou	12	30	-	0	1		-	Port1 FX differential	1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
					t								transmit output	
P1TDXN	-	out	3V3	TDOTAC33NN12	ou t	12	30	-	0	1		-		1
P1SDXP	-	in	3V3	TDIPAECN	in	-	-	-	0	1		-	Port1 FX differential SD	1
P1SDXN	-	in	3V3	TDIPAECR	in	-	-	-	0	1		-	input (PECL)	1
P2RXN	-	AIO	analog	NBTHPHYC2GV02	IO	18	0	-	125	1		-	Port2 differential receive	1
P2RXP	-	AIO	analog	NBTHPHYC2GV02	IO	18	0	-	125	1		-	input	1
P2TXN	-	AIO	analog	NBTHPHYC2GV02	IO	18	0	-	125	1		-	Port2 differential transmit	1
P2TXP	-	AIO	analog	NBTHPHYC2GV02	IO	18	0	-	125	1		-	output	1
P2RDXP	-	in	3V3	TDIPAECNR	in	-	-	-	125	1		-	Port2 FX differential	1
P2RDXN	-	in	3V3	TDIPAECNR	in	-	-	-	125	1		-	receive input (PECL)	1
P2TDXP	-	out	3V3	TDOTAC33NN12	ou t	12	30	-	0	1		-		1
P2TDXN	-	out	3V3	TDOTAC33NN12	ou t	12	30	-	0	1		-	Port2 FX differential	1
P2SDXP	-	in	3V3	TDIPAECN	in	-	-	-	0	1		-	transmit output	
P2SDXN	-	in	3V3	TDIPAECR	in	-	-	-	0	1		-	Port2 FX differential SD	1
													input (PECL)	1
P1FXEN	-	out	3V3	TDOTAC33NN12	ou t	12	30	-	0	1		-		1
P2FXEN	-	out	3V3	TDOTAC33NN12	ou t	12	30	-	0	1		-		1

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
EXTRES	-	AIO	analog	NBTHPHYC2GV02	IO	-	-	-	-	-		-	Reference Resistor of 12.4 kΩ	1
ATP	-	AIO	analog	NBTHPHYC2GV02	IO	-	-	-	-	-		-	Analog Test Function	1
TESTOUT	-	out	3V3	TDOTAC33NN12	out	12	30	-	0	1		-		3
sum														27
CRU, Oscillator, PLL														
REF_CLK	0	out	3V3	TWF1BC33ALV04SZ	bid i	6	50	-	25	0.5		-	Reference Clock MII (ext. PHY)	1
BYP_CLK	-	in	3V3	TWF1IC33AS	in	-	-	-	-	-		-	Bypass Clock for F-Timer	1
XRESET	-	in	3V3	TWF1IC33ASS	in	-	-	UP	-	-		ST	externer Reset-Eingang (lowaktiv)	1
CLKP_A	-	in	3V3	TDOSAC33N32M	in	-	-	-	-	-		-	Oscillator in / Clk_in for oscillator bypass	1
CLKP_B	-	out	3V3	TDOSAC33N32M	bid i	12	30	-	25	1		-	Oscillator	1
sum														5
Debug, JTAG														
TMS	-	in	3V3	TWF1IC33ASS	in	-	-	-	-	-		ST	JTAG: Test Mode Select	1
XTRST	-	in	3V3	TWF1IC33ASS	in	-	-	-	-	-		ST	JTAG: Reset	1

ERTEC 200P	CTRL-STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
TCK	-	in	3V3	TWF1IC33ASS	in	-	-	DN	-	-		ST	JTAG: Clock	1
RTCK	0	out	3V3	TWF1BC33ALV04SZ	bid i	6	50	-	32	1		ST	JTAG: Sync TCK	1
TDI	-	in	3V3	TWF1IC33ASS	in	-	-	UP	-	-		ST	JTAG: Data In	1
TDO	0	out	3V3	TWF1BC33ALV04SZ	bid i	6	50	-	32	0.5		ST	JTAG: Data Out	1
XSRST	0	bidi	3V3	TWF1BC33ASLV04SL	bid i	6	50	UP	0.1	0.1		ST	System-Reset for De- bugging	1
TAP_SEL	-	in	3V3	TWF1IC33ASS	in	-	-	-	-	-		ST	TAP Select Signal: 0 = JTAG for debug 1 = JTAG for BS	1
CHAIN_CTRL	-	in	3V3	TWF1IC33ASS	in	-	-	-	-	-		-	Debugging: 0 = ARM926	1
sum														9
Vendor Test														
TEST	-	in	-	TWF1BC33ASNV02SL	-	-	-	DN	-	-		ST	Test pin	1
TMC1	-	in	-	TWF1ITE1C33ND	-	-	-	DN	-	-		-	Test Mode	1
TMC2	-	in	-	TWF1ITE2C33ND	-	-	-	DN	-	-		-	Test Mode	1
TACT	-	in	-	TWF1BC33ASNV02SL	-	-	-	DN	-	-		ST	Test pin	1
sum														4

ERTEC 200P	CTRL-STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
CTRL-STBY														
CTRL_STBY0	-	in	-	TWF1ISTBC33SN	-	-	-	-	-	-		ST	Tristate of 3.3 V	1
CTRL_STBY1	-	in	-	TWF1ISTBC33SN	-	-	-	-	-	-		ST	Tristate of 1.8 V / 3.3 V XHIF	1
CTRL_STBY2	-	in	-	TWF1ISTBC33SN	-	-	-	-	-	-		ST	Tristate of 1.8 V / 3.3 V XHIF	1
sum														3
Total Signal Pins														230

ERTEC 200P	CTRL- STBYx	Dir	3V3 1V8	Buffer	Buffer Dir	DS (mA)	CL (pF)	int. Pul I	fout (MHz)	activity	Low Noise	Schmitt Trigger	Description	#
VDD / GND														
VDD_XHIF													3.3V / 1.8V	12
VDD33													3.3 V	17
VDD18													1.8V (EMC IO Supply)	15
VDD15													1.5 V (PHY IO Supply)	9
VDD12													1.2 V (Core Supply)	16
GND													GND	82
AVDD33_PHY				analog									3.3V (analog)	2
AVDD15_PHY				analog									1.5V (analog)	3
AGND_PHY				analog									analog GND PHY	6
DVDD15_PHY				analog									1.5V (digital)	2
DGND_PHY				analog									digital GND PHY	2
VDDY_PECCL				analog										2
AVDD_PLL				analog									analog voltage supply for 500 MHz PLL	1
AGND_PLL				analog									analog GND for 500 MHz PLL	1
Total VDD/GND														170
Total for pins														400

Table 25: Signal Description and Pinning ERTEC 200P

3.2.1 Strapping Pins

During XRESET is active, the Input-Value of following Pins are LATCHED in:

Package Ball Name	Strapping Information	Remark
A_17	Config(0)	see chapter 2.3.10.9.3
A_18	Config(1)	
A_19	Config(2)	
A_20	Config(3)	
A_21	Config(4)	
A_22	Config(5)	
A_23	Config(6)	see chapter 2.3.10.9.2
DTXR	Boot(0)	
XOE_DRIVER	Boot(1)	
A_15	Boot(2)	
A_16	Boot(3)	
XAV_BF	Boot(4)	see chapter 2.3.10.9.19
XRDY_BF	<i>EXT_DRIVER_DISABLE_CS0</i>	

Remark:

After XRESET is released, the values are stored. The values must stay stable 120ns after XRESET is released.

3.3 IO Timing

General rule:

For IO signals whose timing response is not detailed in the following sections, there are no specific IO timing requirements. These signals therefore have default constraining, i.e. the constraining selected achieves IO timing closure but without relaxing IO timing too much.

Definition of time reference:

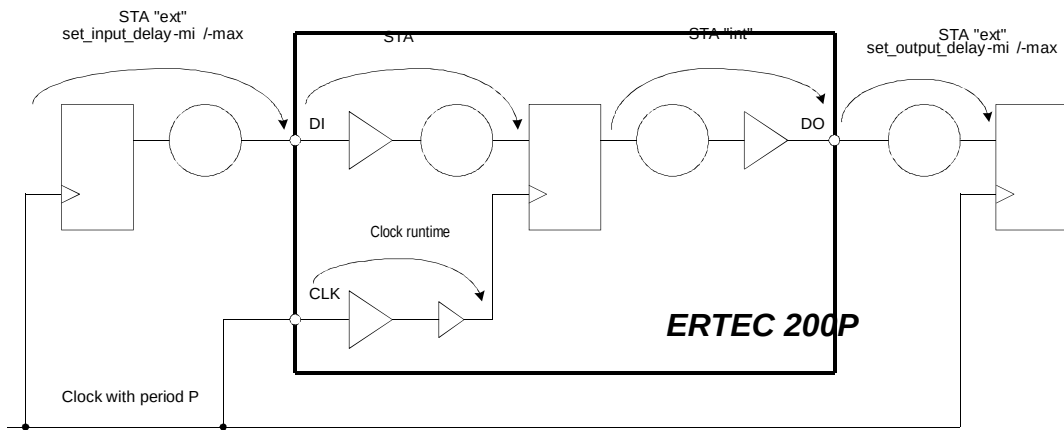


Figure 45: Definition of time reference

The constraint specifications correspond to the data sheet values as follows:

Input delay –min X	Input signal hold time => X
Input delay –max X	Input signal setup time => Period - X
Output delay –min X	Output signal hold time => X
Output delay –max X	Clock-to-output time of output signal => Period - X

3.3.1 EMC Timing

Important:

To achieve the timing below, configure EXTENDED_CONFIG.SODM = '0' (default value).
EXTENDED_CONFIG.SODM = '1' is not permitted

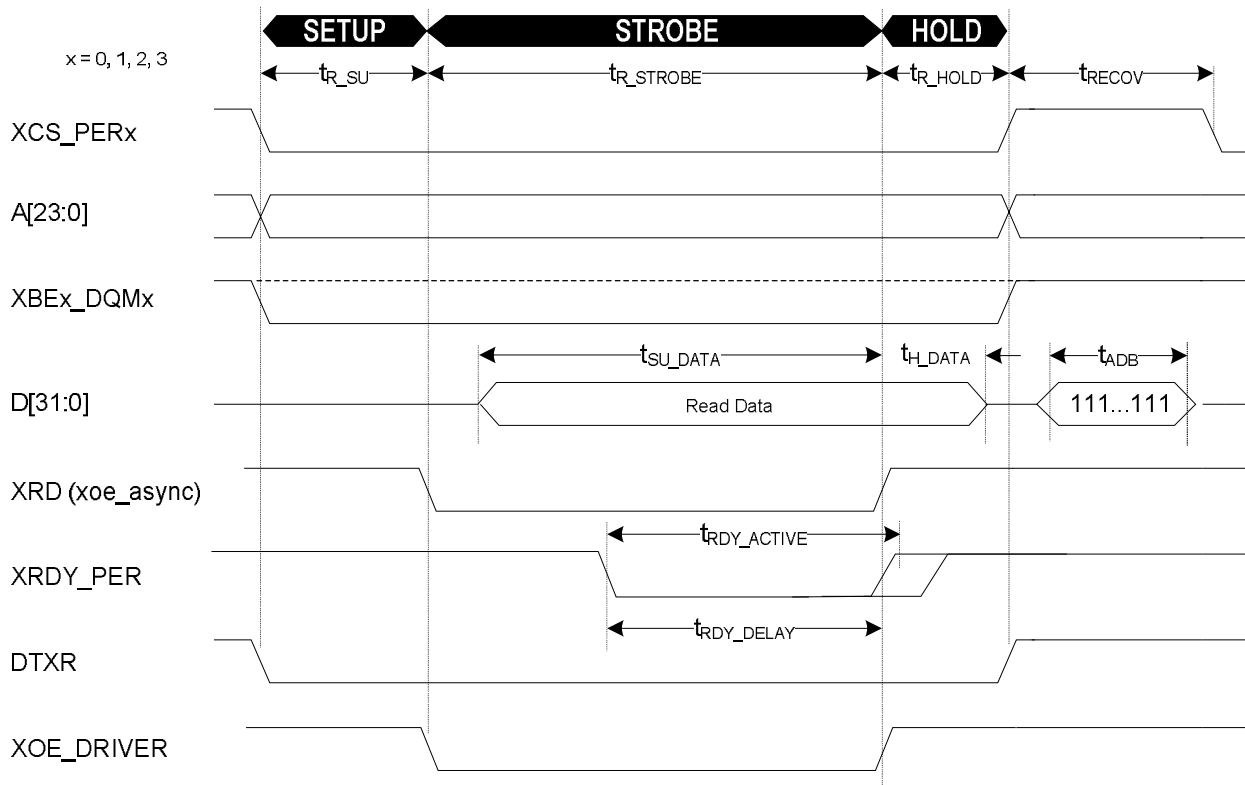
3.3.1.1 SRAM Timing

The use of XRDY_PER is optional and can be enabled with ASYNC_BANKx.EW. Wait-States can be inserted if XRDY_PER is used.

For the asynchronous SRAM interface an “**active interface**” is selectable. Active interface means that at the end of a transfer the data bus is actively driven high for one AHB clock cycle. This is necessary together with the use of internal PullUps to speed up the reloading of the wiring capacity.

After the active phase the internal PullUps are driving the data bus and there is no need for external PullUps on the board.

3.3.1.1.1 SRAM Timing for read access



Parameter	Description	Min	Max	depends on Register	Note
t _{R_SU}	Read Setup Time	0 Tc – 4.5 ns	15 Tc + 4.8 ns	ASYNC_BANKx.R_SU	1)
t _{R_STROBE}	Read Strobe Time	1 Tc – 4.2 ns	64 Tc + 4.4 ns	ASYNC_BANKx.R_STROBE	1)
t _{R_HOLD}	Read Hold-Time	1 Tc – 4.8 ns	8 Tc + 4.4 ns	ASYNC_BANKx.R_HOLD	1)
t _{SU_DATA}	Data Setup Time	5.4ns			
t _{H_DATA}	Data Hold Time	2.1 ns			

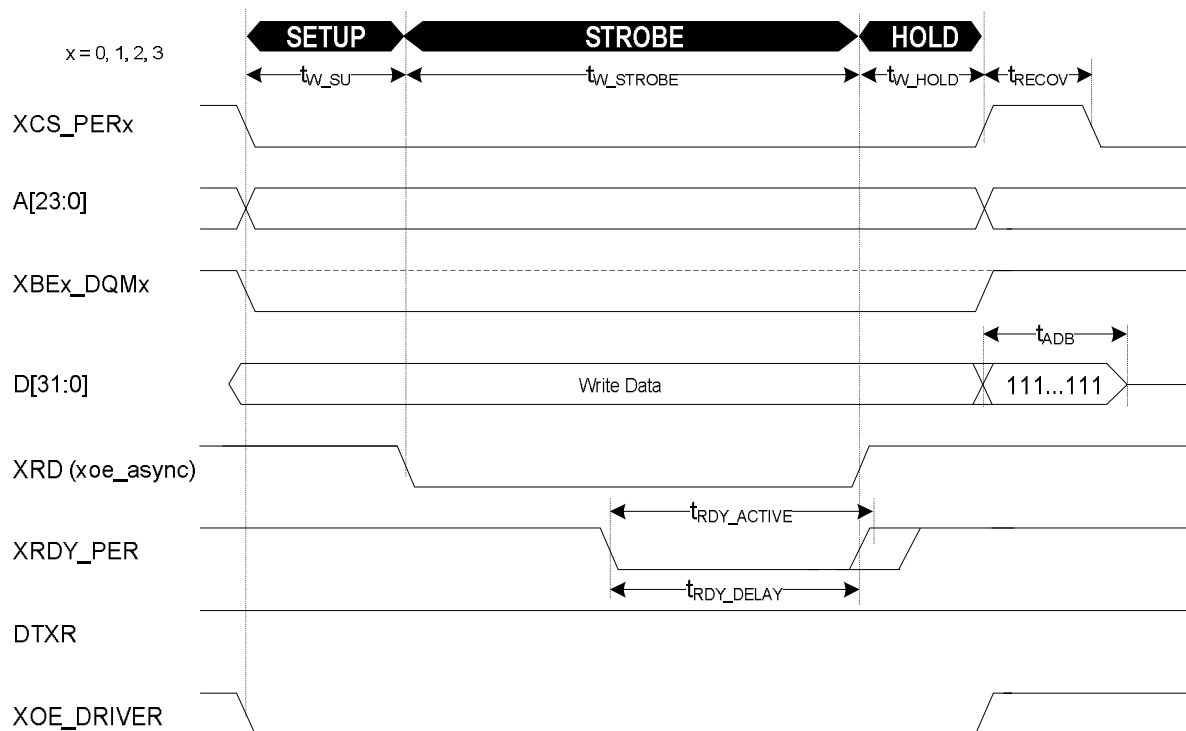
t_{RDY_ACTIVE}	Ready Time	Active	9.5 ns		ASYNC_BANKx.EW	
t_{RDY_DELAY}	Ready Time	Delay	$2 T_c + 2.6$ ns	$3 T_c + 10.1$ ns	ASYNC_BANKx.EW	2)
t_{RDY_DELAY}	Ready Time	Delay	$1 T_c + 2.6$ ns	$2 T_c + 10.1$ ns	ASYNC_BANKx.EW	3)
t_{ADB}	Active data bus		8 ns	8 ns	EXTENDED_CONFIG.ADB	
t_{RECOV}	Recovery Phase		0 ns	120 ns	RECOV_CONFIG.RECOVx	
Based on	$T_c = 8$ ns (AHB Clock = 125 MHz); Load-value for Timing = 20pF Buffer Driverstrength = 12mA IO-Voltage = 1.8V					

1) BANK_0-4_CONFIG register

2) in ASYNC_BANK0...4 register; Bit 31 – WSM = '0'

3) in ASYNC_BANK0...4 register; Bit 31 – WSM = '1'

3.3.1.1.2 SRAM Timing for write access



Parameter	Description	Min	Max	depends on Register	Note
t_{W_SU}	Write Time Setup-	$0 T_c - 1.3$ ns	$15 T_c + 5.1$ ns	ASYNC_BANKx.W_SU	1)
t_{W_STROBE}	Write Time Strobe-	$1 T_c - 1.3$ ns	$64 T_c + 1.6$ ns	ASYNC_BANKx.W_STROBE	1)
t_{W_HOLD}	Write Hold-Time	$1 T_c - 5.1$ ns	$8 T_c + 1.2$ ns	ASYNC_BANKx.W_HOLD	1)
t_{RDY_ACTIVE}	Ready Time Active	9.5 ns		ASYNC_BANKx.EW	
t_{RDY_DELAY}	Ready Time Delay	$2 T_c + 6.2$ ns	$3 T_c + 10.4$ ns	ASYNC_BANKx.EW	2)

t _{RDY_DELAY}	Ready Delay Time	1 Tc + 6.2 ns	2 Tc + 10.4 ns	ASYNC_BANKx.EW	³⁾
t _{ADB}	Active data bus	8 ns	8 ns	EXTENDED_CONFIG.ADB	
t _{RECOV}	Recovery Phase	0 ns	120 ns	RECOV_CONFIG.RECOVx	
Based on	Tc = 8 ns (AHB Clock = 125 MHz); Load-value for Timing = 20pF Buffer Driverstrength = 12mA IO-Voltage = 1.8V				

¹⁾ BANK_0-4_CONFIG register

²⁾ in ASYNC_BANK0...4 register; Bit 31 – WSM = '0'

³⁾ in ASYNC_BANK0...4 register; Bit 31 – WSM = '1'

3.3.1.2 BurstMode Flash Timing

Important:

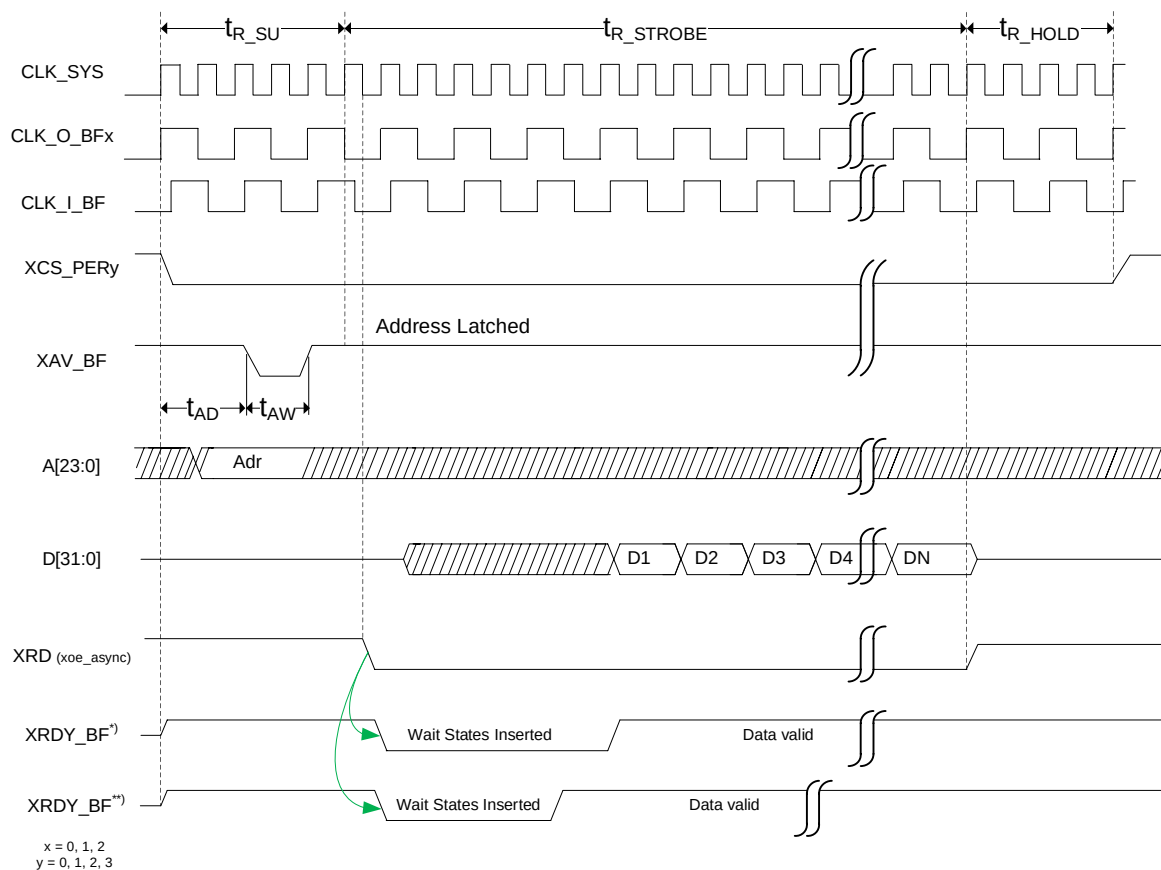
To achieve the correct timing

configure EXTENDED_CONFIG.BFODM = '1'. EXTENDED_CONFIG.BFODM = '0' (default value) is not permitted

configure / retain BF_CONFIG.BF_CLK_F = '0' / (default value). BF_CONFIG.BF_CLK_F = '1' is not permitted.

The default-configuration for BurstFlash is asynchronous. The Timing is the same as for SRAM-Interface (see chapter 3.3.1.1.1 and 3.3.1.1.2). For burst-Operation a setup is needed in the EMC-Controller (e.g. set BF_CONFIG.SYNC_READ to 1) and Burst-ModeFlash Device.

The Input-Signals are latched in with CLK_I_BF.



Notes:

- (1) XRDY_BF active with BF_CONFIG.RDY_DELAY = 0
- (2) XRDY_BF active with BF_CONFIG.RDY_DELAY = 1

Parameter	Description	Min	Max	depends on Register
t_{R_SU}	Read Setup-Time	$t_{AD} + t_{AW}$	120 ns	ASYNC_BANKx.R_SU
t_{R_STROBE}	Read Strobe-Time	8 ns	512 ns	ASYNC_BANKx.R_STROBE
t_{R_HOLD}	Read Hold-Time	8 ns	64 ns	ASYNC_BANKx.R_HOLD
t_{AD}	Address Valid Delay	8 ns	128 ns ¹⁾	BF_CONFIG.AVD_DELAY
t_{AW}	Address Valid Pulse Width	8 ns	64 ns ¹⁾	BF_CONFIG.AVD_PW

¹⁾ $t_{AD} + t_{AW} \leq t_{R_SU}$ must be ensured

3.3.1.3 SDRAM Timing

The combination of the control signals XCS_DRAM, XRAS_SDRAM, XCAS_SDRAM, XWE_SDRAM and XBEy_DQMy together with the Address bus is defining SDRAM commands in the following way:

SDRAM command	XCS_SDRAM	M	M	XWE_SDRAM	XBEy_DQMy	A	Description
COMMAND INHIBIT	1	X	X	X	X	X	No operation

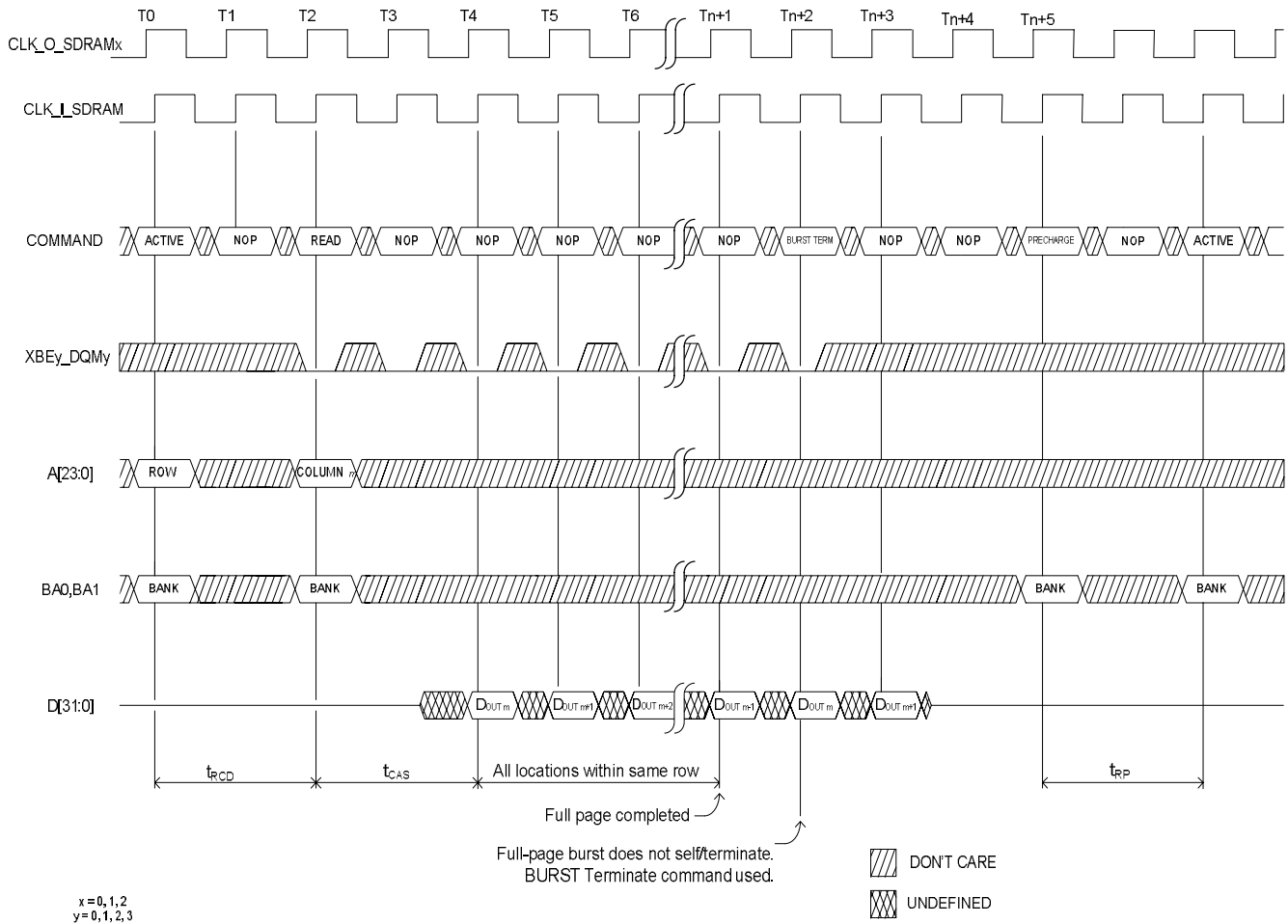
(NOP)							
NO OPERATION (NOP)	0	1	1	1	X	X	No operation
ACTIVE	0	0	1	1	X	BA/Ro w	Select bank and activate row
READ	0	1	0	1	1/0	BA/Co l	Select bank and column, and start READ Burst
WRITE	0	1	0	0	1/0	BA/Co l	Select bank and column, and start WRITE Burst
BURST TERMINATE	0	1	1	0	X	X	Terminate Burst sequence
PRECHARGE	0	0	1	0	X	BA/A1 0	Deactivate row in bank or banks
AUTO REFRESH	0	0	0	1	X	X	Start auto refresh cycle
LOAD MODE REGISTER	0	0	0	0	X	Op- Code1	Setup of the device specific configuration register
LOAD EXTENDED MODE REGISTER	0	0	0	0	X	Op- Code2	Setup of the device specific extended mode configuration register (e.g. mobile SDRAM devices)

X means don't care.

y = 0, 1, 2, 3

3.3.1.3.1 SDRAM Timing for read access

The Output-Signals are launched with CLK_O_SDRAMx, the Input-Signals are latched in with CLK_I_SDRAM.

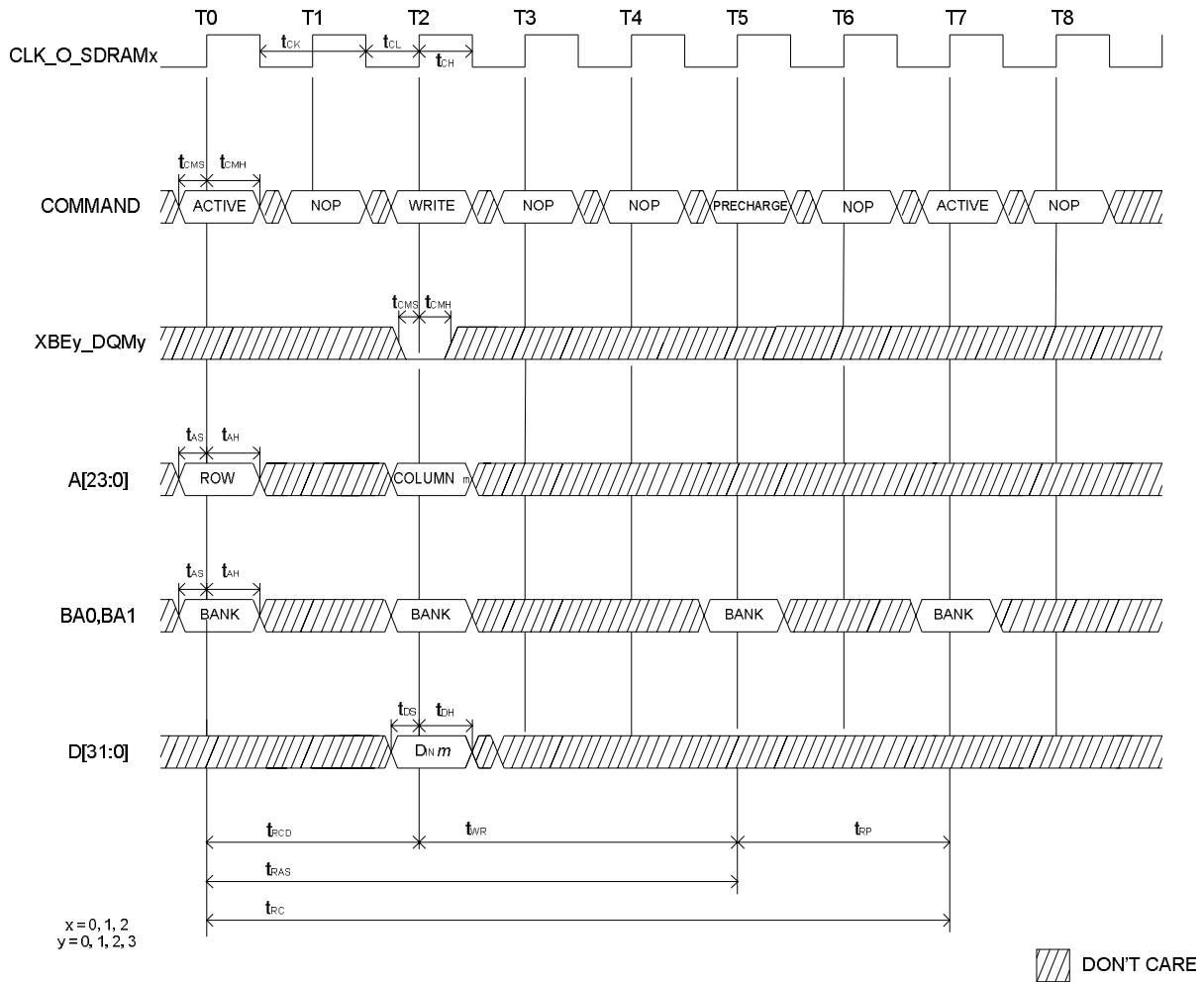


Note: The Bank signals BA0, BA1 are part of the address bus A. They are given here separately for a better understanding.

Parameter	Description	Min	Max	depends on Register	Note
t_{RCD}	RAS to CAS delay	16 ns	32 ns	EXTENDED_CONFIG.T RCD	
t_{CAS}	CAS Latency	16 ns	24 ns	SDRAM_CONFIG.CL	
t_{RP}	Row precharge latency	24 ns	24 ns		
t_{DS}	Data Setup Time	0.0 ns			CLK_I_SDRAM
t_{DH}	Data Hold Time	3.5 ns			CLK_I_SDRAM
Based on	Tc = 8 ns (AHB Clock = 125 MHz); Load-value for Timing = 20pF Buffer Driverstrength = 12mA IO-Voltage = 1.8V				

Setup and hold times for address, command and data are the same as by a write access. They can be found in chapter 7.2

3.3.1.3.2 SDRAM Timing for write access



Note: The Bank signals BA0, BA1 are part of the address bus A. They are given here separately for a better understanding.

Parameter	Description	Min	Max	depends on Register	Note
t_{CK}	Clock Period	7.8 ns	8.2 ns	-	
t_{CL}	Clock Low Time	3.8 ns	4.2 ns	-	
t_{CH}	Clock High Time	3.8 ns	4.2 ns	-	
t_{CMS}	Command Setup Time	3.5 ns	6.6 ns	-	
t_{CMH}	Command Hold Time	1.4 ns	4.5 ns	-	
t_{AS}	Address Setup Time	3.5 ns	6.6	-	
t_{AH}	Address Hold Time	1.4 ns	4.5 ns	-	
t_{DS}	Data Setup Time	3.5 ns	6.6 ns	-	
t_{DH}	Data Hold Time	1.4 ns	4.5 ns	-	
t_{RCD}	RAS to CAS delay	16 ns	40 ns	EXTENDED_CONFIG.TRC D + 1	
t_{RAS}	Row Address Strobe	$t_{RCD} + t_{WR}$	¹⁾		
t_{RC}	ROW cycle Time	$t_{RCD} + t_{WR} + t_{RP}$	-	-	
t_{WR}	Write to Precharge Time	16 ns	¹⁾	-	
t_{RP}	Row precharge time	24 ns	24 ns	-	

	tency				
Based on	Tc = 8 ns (AHB Clock = 125 MHz); Load-value for Timing = 20pF Buffer Driverstrength = 12mA IO-Voltage = 1.8V				

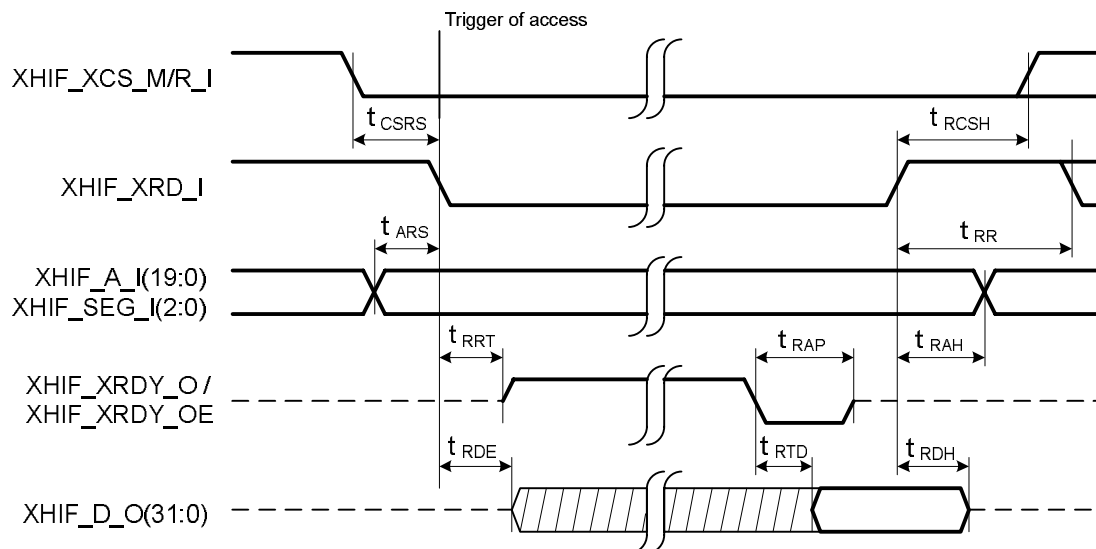
¹⁾ Depends on Refresh Cycle Time

3.3.2 Host-Interface Timing

3.3.2.1 XHIF Timing

3.3.2.1.1 Separate RDWR

The following figure shows the timing, when the External Host initiates a **Read Access**.

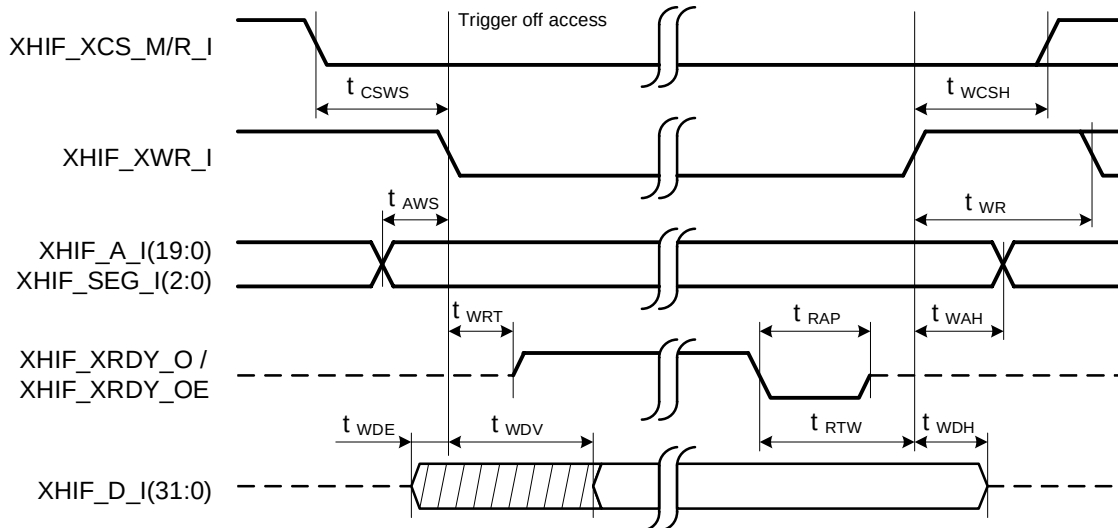


Parameter	Description	Min	Max
t _{CSRS}	chip select asserted to read pulse asserted delay	4.4 ns ¹⁾	
t _{ARCS}	address valid to read pulse asserted setup time	1.9 ns	
t _{RRT}	read pulse asserted to ready deasserted delay	4.0 ns	11.2 ns
t _{RDE}	read pulse asserted to data enable delay	3.7 ns	11.4 ns
t _{RAP}	ready active pulse width	6.1 ns	10.1 ns
t _{RTD}	ready asserted to data valid delay		10.6 ns
t _{RCSH}	read pulse deasserted to chip select deasserted delay	1.3 ns ²⁾	
t _{RAH}	address valid to read pulse deasserted hold time	1.1 ns	
t _{RDH}	data valid/enable to read pulse deasserted hold time	2.4 ns	11.7 ns
t _{RR}	read recovery time	12.4 ns	
Based on	Tc = 8 ns (AHB Clock = 125 MHz); Load-value for Timing = 20pF Buffer Driverstrength = 9mA IO-Voltage = 3,3V		

¹⁾ If t_{CSRS} < 0, t_{ARCS}, t_{RRT} and t_{RDE} are related to the falling edge of XHIF_XCS

²⁾ If t_{RCSH} < 0, t_{RAH} and t_{RDH} are related to the rising edge of XHIF_XCS

The following figure shows the timing, when the External Host initiates a **Write Access**:



Parameter	Description	Min	Max
t_{CSWS}	chip select asserted to write pulse asserted delay	0.8 ns ¹⁾	
t_{AWS}	address valid to write pulse asserted setup time	1.9 ns	
t_{WRT}	write pulse asserted to ready deasserted delay	4.0 ns	11.2 ns
t_{WDE}	write pulse asserted to data enable setup	0 ns	t.b.d ns ³⁾
t_{WDV}	write pulse asserted to data valid delay		15.7 ns
t_{RAP}	ready active pulse width	6.1 ns	10.1 ns
t_{WCSH}	write pulse deasserted to chip select deasserted delay	1.7 ns ²⁾	
t_{WAH}	address valid to write pulse deasserted hold time	1.6 ns	
t_{RTW}	ready asserted to write pulse deasserted delay	0 ns	
t_{WDH}	data valid/enabled to read pulse deasserted hold time	1.8 ns ²⁾	
t_{WR}	write recovery time	10.9 ns	
Based on	Tc = 8 ns (AHB Clock = 125 MHz); Load-value for Timing = 20pF Buffer Driverstrength = 9mA IO-Voltage = 3,3V		

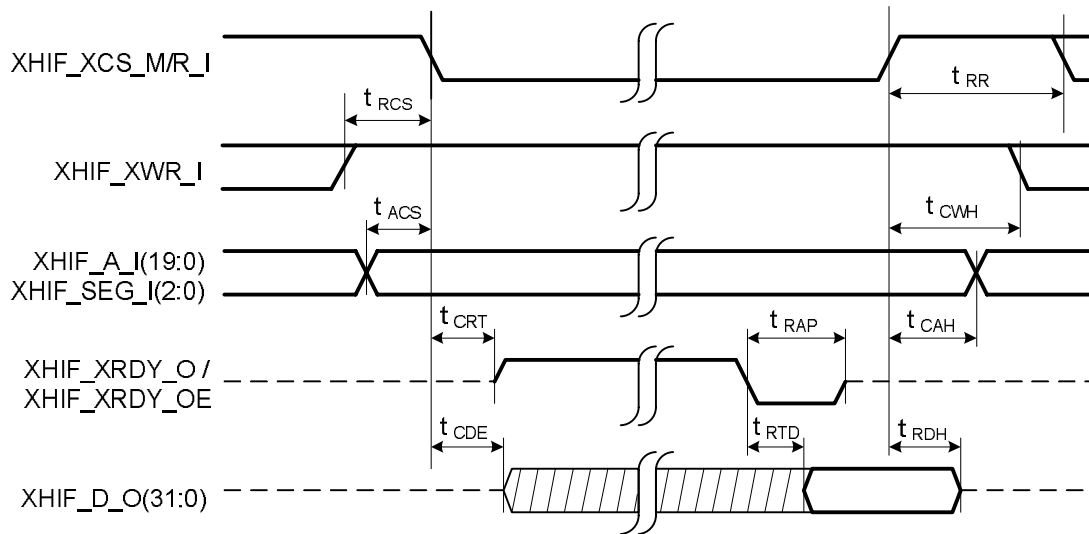
¹⁾ If $t_{CSWS} < 0$, t_{AWS} , t_{WRT} and t_{WDE} are related to the falling edge of XHIF_XCS

²⁾ If $t_{WCSH} < 0$, t_{WAH} and t_{WDH} are related to the rising edge of XHIF_XCS

³⁾ t_{WDE} may get any value, as long as it is assured, that there is 1 idle cycle (of the XHIF clock period) guaranteed between the end of the preceding access and the start of the current access (indicated by the falling edge of XCS/XWR). Within this idle cycle no access is allowed at all.

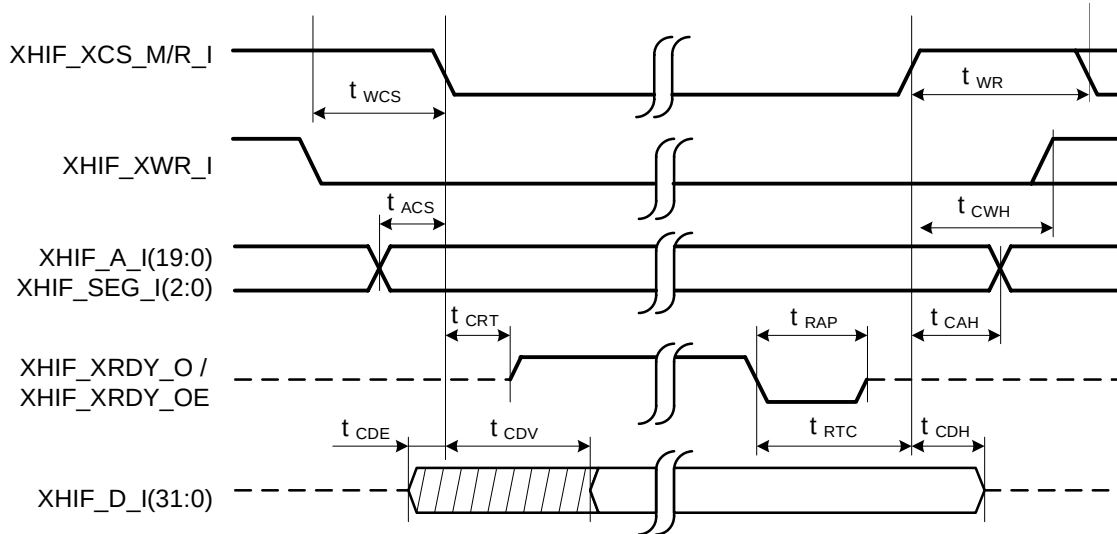
3.3.2.1.2 Common RD/WR

The following figure shows the timing, when the External Host initiates a **Common Read Access**.



Parameter	Description	Min	Max
t_{RCS}	Write signal deasserted to chip select asserted delay	6.2 ns	
t_{ACS}	address valid to chip select asserted setup time	2.1 ns	
t_{CRT}	chip select asserted to ready deasserted delay	2.8 ns	11.7 ns
t_{CDE}	chip select asserted to data enable delay	2.5 ns	11.9 ns
t_{RAP}	ready active pulse width	6.1 ns	10.2 ns
t_{RTD}	ready asserted to data valid delay		10.6 ns
t_{CWH}	chip select deasserted to write signal asserted delay	0.9 ns	
t_{CAH}	address valid to chip select deasserted hold time	1.2 ns	
t_{RDH}	data valid/enable to chip select deasserted hold time	2.5 ns	11.7 ns
t_{RR}	read recovery time	8.8 ns	
Based on	Tc = 8 ns (AHB Clock = 125 MHz); Load-value for Timing = 20pF Buffer Driverstrength = 9mA IO-Voltage = 3,3V		

The following figure shows the timing, when the External Host initiates a **Common Write Access**:



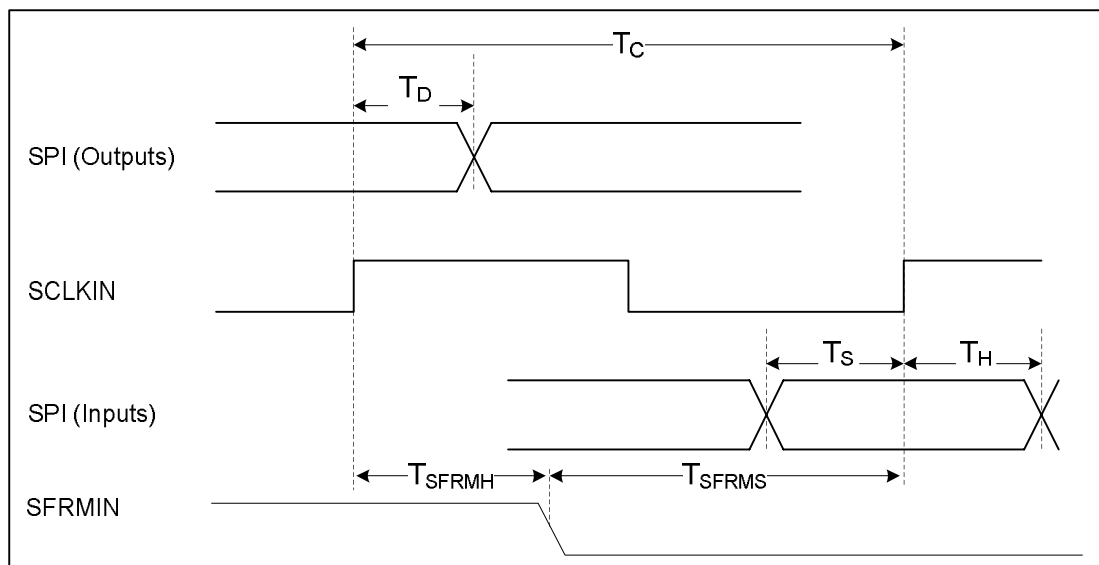
Parameter	Description	Min	Max
t_{WCS}	Write signal asserted to chip select setup time	1.6 ns ¹⁾	
t_{ACS}	address valid to chip select asserted setup time	2.1 ns	
t_{CRT}	chip select asserted to ready deasserted delay	2.8 ns	11.7 ns
t_{CDE}	chip select asserted to data enable setup	-1.7 ns	1.1 ns ²⁾
t_{CDV}	chip select asserted to data valid delay		15.6 ns
t_{RAP}	ready active pulse width	6.1 ns	10.2 ns
t_{CWH}	write signal deasserted to chip select deasserted delay	0.9 ns	
t_{CAH}	address valid to chip select deasserted hold time	1.2 ns	
t_{RTC}	ready asserted to chip select deasserted delay	0 ns	
t_{CDH}	data valid/enabled to chip select deasserted hold time	0 ns ²⁾	
t_{WR}	write recovery time	8.8 ns	
Based on	Tc = 8 ns (AHB Clock = 125 MHz); Load-value for Timing = 20pF Buffer Driverstrength = 9mA IO-Voltage = 3,3V		

¹⁾ It is important to meet the setup timing of the Write signals; otherwise the XHIF module is driving the data bus

²⁾ t_{CDE} may get any value, as long as it is assured, that there is 1 idle cycle (of the XHIF clock period) guaranteed between the end of the preceding access and the start of the current access (indicated by the falling edge of XHIF_XCS). Within this idle cycle no access is allowed at all.

3.3.2.2 SPI Timing

The SPI part of the host interface is a serial slave interface that operates at a max. of 31.25 MHz. The SPI clock is used for clocking internal registers. The signals are available as alternative functions at the pins of the XHIF.



Symbol	Parameter	Min.	Max.	Unit	Note
T_c	Baudrate	32		ns	Slave
T_D	Valid delay	3.7	9.9	ns	¹⁾
T_S	Setup Time	1.2		ns	
T_H	Hold Time	1.2		ns	
T_{SFRMS}	SFRMIN Setup Time	2.0		ns	
T_{SFRMH}	SFRMIN Hold Time	2.0		ns	
Based on		Buffer Driverstrength = 9mA			

¹⁾ $C_L = 20 \text{ pF}$

3.3.3 PN-IP Timing

3.3.3.1 MII Timing

In the receive path, the data are output by the PHY following a rising RX_CLK clock edge. The PHY guarantees a setup and hold time of 10 ns. This means that the PHY can output the receive signals with a delay of 10 ns to 30 ns.

In the transmit path, the transmit signals are output by the MAC following a TX_CLK rising edge. The MAC can output these data with a delay of 0 ns to 25 ns in accordance with the standard.

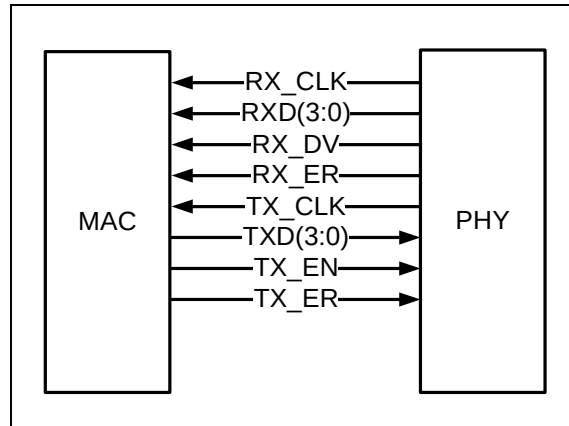


Figure 46: MII interface

3.3.3.1.1 MII Timing at Integrated PHY

Symbol	Parameter	Min.	Max.	Unit	Note
$T_{SU\ TX}$	Setup Zeit TX	28.7		ns	TX_CLK
$T_{H\ TX}$	Hold Zeit TX	3.9		ns	TX_CLK
$T_{SU\ RX}$	Setup Zeit RX	12.3		ns	RX_CLK
$T_{H\ RX}$	Hold Zeit RX	0		ns	RX_CLK
Based on					

3.3.3.2 MDIO Timing

The management interface has very relaxed timing and is based on a clock frequency of 2.5 MHz. The PHY can output the data with a delay of 0 ns to 300 ns. The MAC must output the data with a minimum delay of 10 ns and must have output the data within 390 ns at the latest.

A PHY reset implements a hardware reset for all connected PHYs. The timing of this signal is asynchronous.

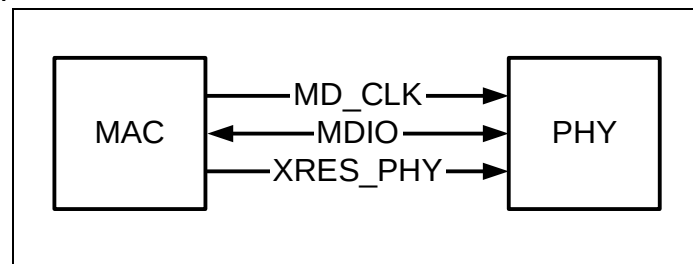


Figure 47: MDIO interface

The PN-IP in ERTEC 200P outputs both the clock and the SMI interface data with the system clock. The correct timing is ensured by the circuit – the SMI interface accepts data with a rising clock edge and outputs data with a falling clock edge.

As an example of constraining for the PN-IP in the ERTEC 200P, the following values should be used for a load of min.10pF / max. 20pF – relaxed default timing. The interface will in fact, like the MII interface, be internal to the package (SIP); the load capacity will no longer have a min/max value (which will also be much lower).

3.3.3.2.1 MDIO Timing at the Integrated PHY

Symbol	Parameter	Min.	Max.	Unit	Note
T _{CO}	Clock2out MDIO	4.8	21.5	ns	CLK_MDC
T _H	Hold Zeit MDIO (in)	0.0		ns	CLK_MDC
T _{SU}	Setup Zeit MDIO (in)	11.7		ns	CLK_MDC
Based on					

3.3.3.3 Integrated PHY Timing

3.3.3.3.1 PHY FX Timing

The Ethernet interfaces of the integrated PHYs are part of the PHY macro and do not have any constraints. The differential signals for controlling fiber-optic lines (FX interface) are not an integral part of the PHY macro; they are connected separately. The most important thing for these signals is as small as possible a skew between the differential pairs – there is no other reference.

Note:

3.3.3.3.2 PHY LED Timing

These outputs of ERTEC 200P are part of the GPIOs.

3.3.3.4 PNPLL Timing

A total of 21 PN-PLL signals are sent to the ERTEC 200P top level from the PN-IP. Of these signals, the **nine** output signals PNPLL_out(8..0) can be connected to ext. pins over GPIOs. PNPLL_Extin can also be connected as an input signal from an ext.pin to the PNPLL over a GPIO.

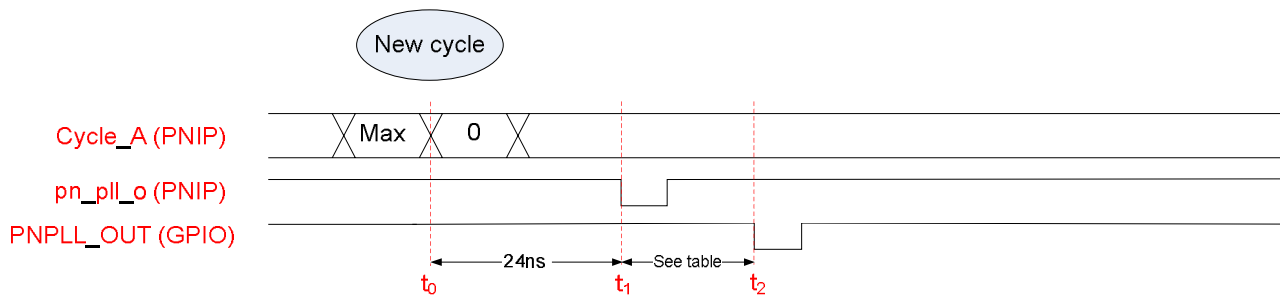
One of the nine output signals PNPLL_out(8..0) can be used as SyncOut for measuring the synchronism of individual nodes in a network topology.

This requires the following configuration in the PN-IP:

- Register *CycleCompare_A_1* = 0x0000_0000 (trigger pulse at start of cycle)
- Register *PLL_OUT_Control_0* = 0x0000_0016 (event appears at the signal pn_pll_o(0))

Multiplexers and GPIOs must be set accordingly on the ERTEC 200P top level (see 2.3.8.1.3).

PNPLL_OUT can be measured at the selected GPIO pin; the signal is generated as follows:



t_0 : Start of cycle, i.e. the time at which a clock edge switches the cycle timer from its maximum value to its minimum value.

t_1 : pn_pll_o is output by PNIP (output signal to entity).

t_2 : The signal PNPLL_OUT appears at the external GPIO pin

This produces the following time delay in the PN-IP:

	Time	Explanation	Min/Max (ns)
PN-IP	$t_0 \rightarrow t_1$	Time from Cycle_A transition until the internal PN-IP flip-flop of pn_pll_o(0) changes from 1 to 0.	24 ns (exactly three 8 ns clock cycles)
ERTEC 200P	$t_1 \rightarrow t_2$	Output_delay of one the following PN-PLL output signals from the PN-IP to the GPIO pin.	See table below

Signal (Input)	Ports	Alternative Function	Input delay		Load (pF) Min / Max	Reference	Functional characteristics
			-min	-max			
PNPLL_EXTIN_A	GPIO_9	A	1.5 ns	4.4 ns	-	CLK_SYS	asynchronous
	GPIO_23	B	1.5 ns	4.4 ns	-	CLK_SYS	asynchronous
Signal (Output)			Clock-to-Output delay			Reference	
			-min	-max			
PNPLL_OUT0	GPIO_0	A	5.0 ns	12.7 ns	10 / 30	CLK_SYS	asynchronous
	GPIO_56	C	4.6 ns	11.9 ns	10 / 30	CLK_SYS	asynchronous
PNPLL_OUT1	GPIO_1	A	5.0 ns	12.6 ns	10 / 30	CLK_SYS	asynchronous
	GPIO_57	C	4.5 ns	11.7 ns	10 / 30	CLK_SYS	asynchronous
PNPLL_OUT2	GPIO_2	A	4.9 ns	12.2 ns	10 / 30	CLK_SYS	asynchronous
	GPIO_58	C	4.5 ns	11.4 ns	10 / 30	CLK_SYS	asynchronous
PNPLL_OUT3	GPIO_3	A	4.9 ns	11.9 ns	10 / 30	CLK_SYS	asynchronous
	GPIO_59	C	4.5 ns	11.9 ns	10 / 30	CLK_SYS	asynchronous
PNPLL_OUT4	GPIO_4	A	4.8 ns	11.8 ns	10 / 30	CLK_SYS	asynchronous
	GPIO_60	C	4.4 ns	11.3 ns	10 / 30	CLK_SYS	asynchronous
PNPLL_OUT5	GPIO_5	A	4.7 ns	11.5 ns	10 / 30	CLK_SYS	asynchronous
	GPIO_61	C	4.4 ns	11.3 ns	10 / 30	CLK_SYS	asynchronous
PNPLL_OUT6	GPIO_6	A	4.7 ns	11.8 ns	10 / 30	CLK_SYS	asynchronous
	GPIO_29	C	4.5 ns	11.7 ns	10 / 30	CLK_SYS	asynchronous
PNPLL_OUT7	GPIO_7	A	4.6 ns	11.4 ns	10 / 30	CLK_SYS	asynchronous
	GPIO_30	C	4.6 ns	11.7 ns	10 / 30	CLK_SYS	asynchronous
PNPLL_OUT8	GPIO_8	A	4.8 ns	11.8 ns	10 / 30	CLK_SYS	asynchronous
	GPIO_31	C	4.6 ns	11.7 ns	10 / 30	CLK_SYS	asynchronous

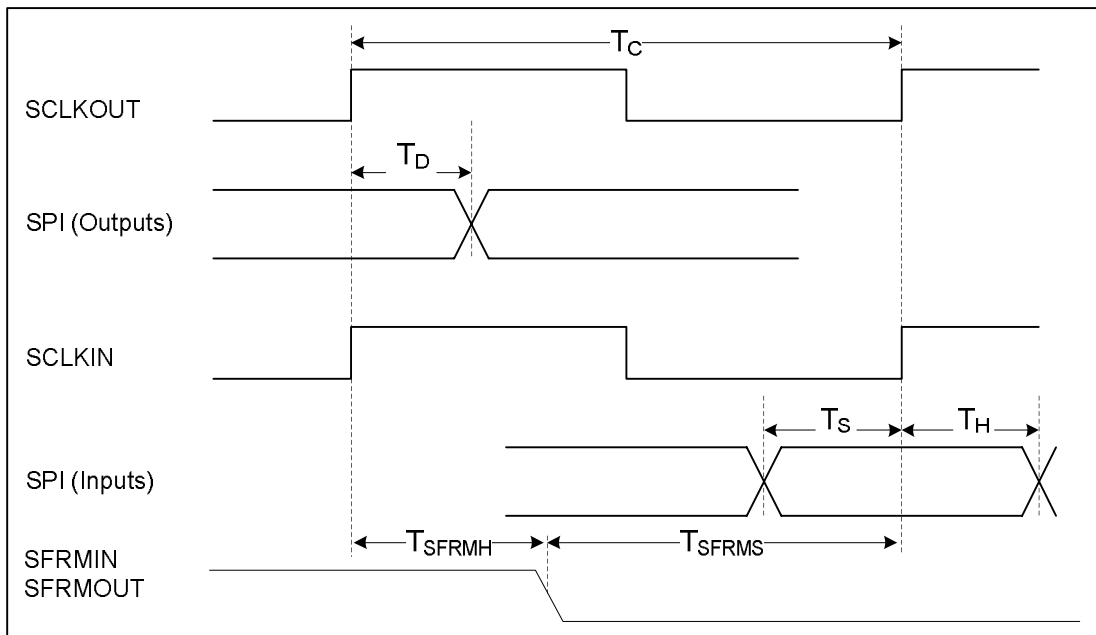
Table 26: Constraining PNPLL Interface

3.3.3.4.1 Timing for time synchronization

Signal (Input)	Ports	Alterna- te Func- tion	Input_delay		Load (pF) Min / Max	Refer- ence	Functional characteris- tics
			-min	-max			
PNPLL_EXTIN_Ti me	GPIO_12	A	1.5 ns	4.4 ns	-	CLK_SY S	asynchronous
Signal (Output)			Clock-to-Output_delay				
			-min	-max			
PNPLL_Time_Out	GPIO_11	A	4.9 ns	12.4 ns	10 / 30	CLK_SY S	asynchronous
	GPIO_28	C	4.8 ns	12.3 ns	10 / 30	CLK_SY S	asynchronous

3.3.4 SPI Timing

The frequency of SSPCLK (base frequency for SPI macro): $f_{SSPCLK} = 125 \text{ MHz}$.



3.3.4.1 SPI1

Symbol	Parameter	Min.	Max.	Unit	Note
T_C	Baudrate	25	-	ns	Master
		150	-	ns	Slave
T_D	Valid delay	-2.9	2.4	ns	Master
		21.2	37.7	ns	Slave
T_S	Setup Time	13.9	-	ns	Master
		-14.9	-	ns	Slave
T_H	Hold Time	-3.8	-	ns	Master
		26.7	-	ns	Slave
T_{SFRMS}	SFRMIN Setup Time	16.7	-	ns	
T_{SFRMH}	SFRMIN Hold Time	9.6	-	ns	
Based on Buffer Driver strength = 9mA					

¹⁾ $C_L = 20 \text{ pF}$

²⁾ $T_S, T_H > 1 \times 125 \text{ MHz period}$; Inputs are synchronized with APB-Clock (F_{SSPCLK}).

³⁾ Slave-Mode : SCLK_IN is synchronized in (2 internal clocks)

3.3.4.2 SPI2

Symbol	Parameter	Min.	Max.	Unit	Note
T_C	Baudrate	25	-	ns	Master
		150	-	ns	Slave
T_D	Valid delay	-3.0	2.2	ns	Master
		22.1	36.8	ns	Slave
T_S	Setup Time	14.6	-	ns	Master
		-15.1	-	ns	Slave
T_H	Hold Time	-4.2	-	ns	Master
		26.0	-	ns	Slave

T_{SFRMS}	SFRMIN Setup Time	17.5	-	ns	
T_{SFRMH}	SFRMIN Hold Time	9.0	-	ns	
Based on	Buffer Driverstrength = 9mA				

¹⁾ $C_L = 20 \text{ pF}$

²⁾ $T_S, T_h > 1 \times 125 \text{ MHz period}$; Inputs are synchronized with APB-Clock (F_{SSPCLK}).

³⁾ Slave-Mode : SCLK_IN is synchronized in (2 internal clocks)

Operation without SPI_FRAME_N:

If the ERTEC 200P is connected as SPI slave to a standard controller (SPI master) that does not support the SPI_FRAME_N signal, the ERTEC 200P is to be set as follows:

- Configuration of Motorola format: $SSPCR0.FRF = '00'$ (see 2.3.10.7.7) AND
- Configuration of the SPI clock phase: $SSPCR0.SPH = '1'$ (see 2.3.10.7.7) AND
- GPIO pin SPI_FRAME_N (see 3.2)
 - Connect with ext. pull-down OR
 - Connect with int. pull-down ($PULL_{xx_yy}GPIO = "11"$, see 2.3.10.9.15.2) OR
 - Do not select the alternate function (blocking value = '0' is active)

Please note the following restrictions in this operating mode:

- In the SPI status register, $SSPSR.BSY$ (see 2.3.10.7.7) is not reset at the end of the transfer
- Following a RESET (see 0), "zero data" are transferred when the first character (4-bit...16-bit, $SSPCR0.DCR$) is transferred from the SPI slave to the SPI master (transfer direction)

3.3.5 UART Timing

3.3.5.1 UART1

Signal	Output Runtime (ns)		Input Runtime (ns)	
	T _{OR min}	T _{OR max}	T _{IR min}	T _{IR max}
GPIO48(C) (UART_CTS_I)			1.4	4.4
GPIO49(C) (UART_DCD_I)			1.6	5.0
GPIO50(C) (UART_DSR_I)			1.5	4.6
GPIO51(C) (UART_RI_I)			1.4	4.7
GPIO52(C) (UART_RTS_O)	3.1	8.5		
GPIO53(C) (UART_DTR_O)	2.9	7.9		
GPIO62(C) (UART_TXD_O)	2.7	7.2		
GPIO63(C) (UART_RXD_I)			1.4	4.2
Based on	Buffer Driverstrength = 9mA			

3.3.5.2 UART2

Signal	Output Runtime (ns)		Input Runtime (ns)	
	T _{OR min}	T _{OR max}	T _{IR min}	T _{IR max}
GPIO12(B) (UART_CTS_I)			1.3	3.9
GPIO13(B) (UART_RTS_O)	3.1	7.5		
GPIO14(B) (UART_TXD_O)	3.1	8.0		
GPIO15(B) (UART_RXD_I)			1.4	4.3
Based on	Buffer Driverstrength = 9mA			

3.3.5.3 UART3

Signal	Output Runtime (ns)		Input Runtime (ns)	
	T _{OR min}	T _{OR max}	T _{IR min}	T _{IR max}
GPIO28(B) (UART_TXD_O)	2.9	7.4		
GPIO29(B) (UART_RXD_I)			1.3	3.6
Based on	Buffer Driverstrength = 9mA			

3.3.5.4 UART4

Signal	Output Runtime (ns)	Input Runtime (ns)
--------	---------------------	--------------------

	T_{OR min}	T_{OR max}	T_{IR min}	T_{IR max}
GPIO54(C) (UART_TXD_O)	2.7	7.5		
GPIO55(C) (UART_RXD_I)			1.4	1.3
Based on	Buffer Driverstrength = 9mA			

3.3.6 I²C Timing

3.3.6.1 I²C – APB

Symbol	Parameter	Min.	Max.	Unit	Note
T _{out SCLK}	SCLK Output delay	1.5	8.3	ns	¹⁾
T _{out SDIO}	SDIO Output delay	1.5	8.7	ns	¹⁾
T _{in SCLK}	SCLK Output delay	1.5	4.4	ns	¹⁾
T _{in SDIO}	SDIO Output delay	1.4	4.6	ns	¹⁾
Based on	Buffer Driverstrength = 9mA				

¹⁾ C_L = 20 pF

3.3.6.2 I²C – PN-IP

Symbol	Parameter	Min.	Max.	Unit	Note
T _{out SCLK}	SCLK Output delay	1.5	8.4	ns	¹⁾
T _{out SDIO}	SDIO Output delay	1.2	8.0	ns	¹⁾
T _{in SCLK}	SCLK Output delay	1.4	4.8	ns	¹⁾
T _{in SDIO}	SDIO Output delay	1.3	4.4	ns	¹⁾
Based on	Buffer Driverstrength = 9mA				

¹⁾ C_L = 20 pF

3.3.7 GPIO Timing

The GPIO interface is asynchronous to external signals. External inputs will be synchronized internally.

All inputs have no timing relation from an external clock to internal clock of ERTEC 200P.

All Outputs have no timing relation from internal clock of ERTEC 200P to an external clock.

Symbol	Description	Min	Max	Unit	Note
T _{din}	Runtime of Input signals	0.9	4.3	ns	¹⁾
T _{dout}	Runtime of Output signals	2.4	8.2	ns	¹⁾
T _{din skew}	Skew of Input signals	-	2.5	ns	¹⁾
T _{dout skew}	Skew of Output signals	-	3.3	ns	¹⁾
Based on	Buffer Driverstrength = 9mA				

¹⁾ C_L = 20 pF

3.3.8 JTAG Timing

The JTAG interface consists of the TCK, TMS, TDI, TDO, and TRST signals. All input signals except TRST are clocked in with a TCK rising edge. The TDO signal is output with a TCK falling edge.

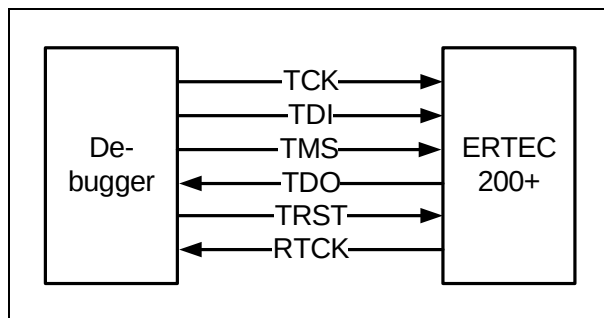


Figure 48: Debug interface

The JTAG clock is to be operated up to a frequency 32 MHz (RTCK evaluation in the debugger). A minimum hold time of 0 ns should be used for the inputs of the debug interface. Only a JTAG clock of up to 16 MHz can be used for debuggers that do not support RTCK.

3.3.9 ARM926 Trace Timing

The ARM926 trace is recorded in an integrated trace buffer (ETB). The buffer is 8 KByte (2K x 32 bits). The trace buffer can be read out over the JTAG interface after recording. Both ARM926 clock rates (125/250 MHz) are supported.

An external debugger can be connected to a trace port. In this case, however, only the ARM926 clock rate of 125 MHz is supported.

The trace interface consists of ERTEC 200P outputs and is specified relative to the trace clock. The trace clock operates at a maximum frequency of 62.5 MHz and the data are output with a rising and with a falling edge. The ERTEC 200P complies with a setup time of 2 ns and a hold time of 1 ns.

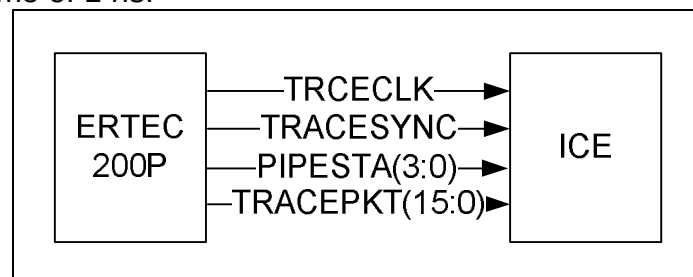
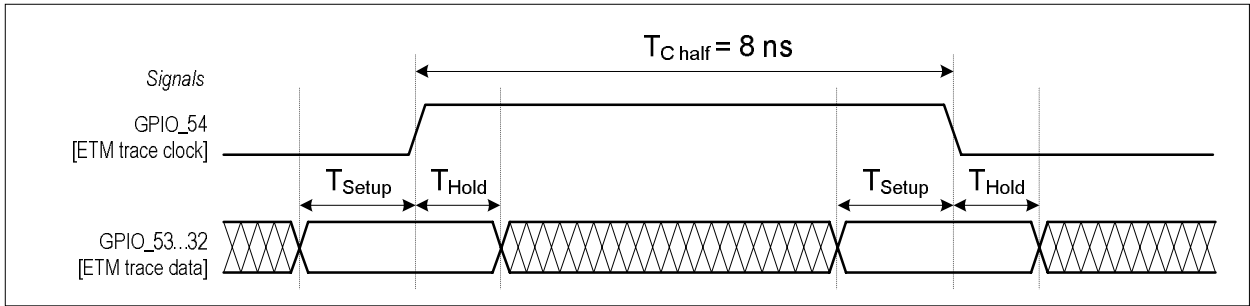


Figure 49: ARM926 trace interface

The constraints specified in the ETM specifications (3 ns setup time, 2 ns hold time) **cannot** be met in the ERTEC 200P. Assuming, however, that ERTEC 200P tracing is not implemented in maximum conditions (temperature and voltage), there is still sufficient tolerance. (To compare: Current debuggers require a setup time of 0.3 ns and a hold time of 0.3 ns for the trace clock).

The ETM trace is clocked in the ERTEC 200P at a fixed 125 MHz (ARM926 subsystem clock), so the trace information, including trace clock (TRACECLK), is output over GPIO_32..._54, SCRB.CONFIG_REG(6:3) = "1110" (see 2.3.10.8.4.1) at both edges at 62.5 MHz.



Symbol	Parameter	Min.	Max.	Unit	Note
T_{Setup}	Setup time to traceclock	4.1	-	ns	
T_{Hold}	Hold time from traceclock	2.9	-	ns	
Based on	Buffer Driverstrength = 6mA				

4 LAYOUT AND DESIGN HINTS

4.1 EMC Measures

4.1.1 ESD Protection

With the 3.3 V input cells, there is a sufficient clearance from the ESD structure triggering.

- o ESD characteristics CB55, 3.3 V I/O buffer:
 - Trigger voltage (Vt1): ca. 9 V
 - Hold voltage (Vhold): ca. 5 V
 - VDDmax: 3.6 V (3.3 V + 10%)

With the 1.8 V input cells, there is a sufficient clearance from the ESD structure triggering.

- o ESD characteristics CB55, 1.8 V I/O buffer:
 - Trigger voltage (Vt1): ca. 7 V
 - Hold voltage (Vhold): ca. 4 V
 - VDDmax: 1.95 V (1.8 V + 10%)

4.1.2 Immunity to ESD

- HBM: 1000 V
- SDM/CDM: 500 V

4.1.3 Package Power Distribution

There was a focus on "good distribution" of the supply pins in the ballout.

→ short discharge paths between protective structure and supply ball are guaranteed

Optimization of the I/O ring and the package PCB ensures better signal integrity and improves the discharge capacity of the supply pins

→ Leakage current and signal [return] current respond in a similar way.

4.1.4 Spike Filter

The same spike filters are implemented for the test inputs TAP_SEL and TACT at the input as for the reset inputs (XRESET, XSRST, XTRST). The filters ensure that spikes ≤ 40 ns (best case) are suppressed. This ensures a time constant of 40 ns between the pin and function (low-pass).

Note: A spike at TAP_SEL and/or TACT is not usually forwarded to the JTAG controller as this would require a sequence over TDI/TMS and TCK from the debugger.

The ERTEC 200P does contain a spike filter of 40 ns (best case condition) for following signals:

- XRESET (Power-On Reset)
- XSRST (Debugger-Reset)
- XTRST (JTAG Reset)
- TAP_SEL (TAP Select)
- TACT (Testmode)

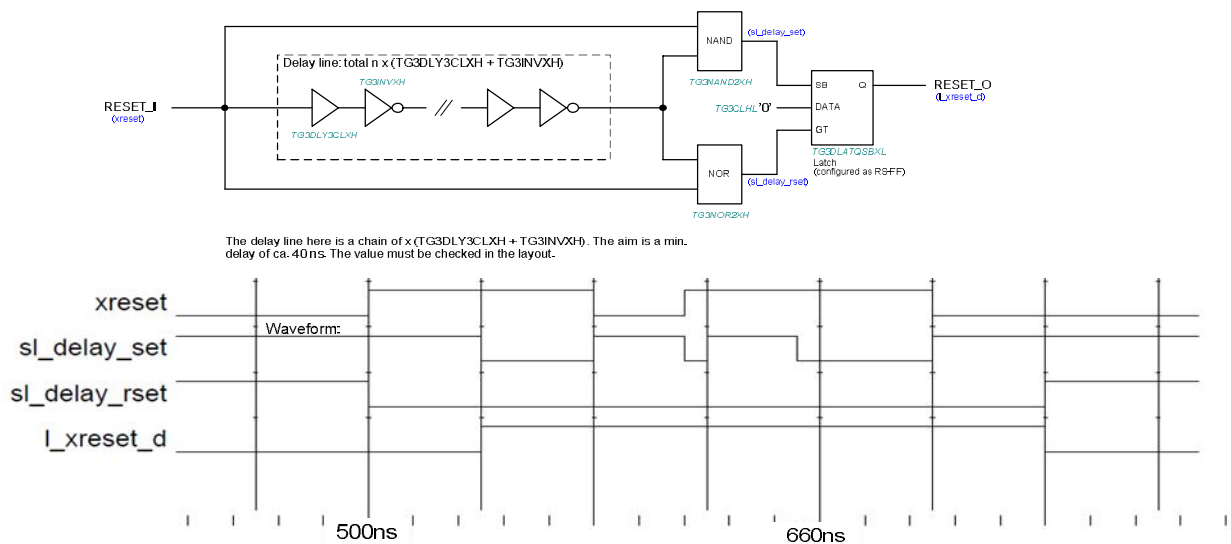
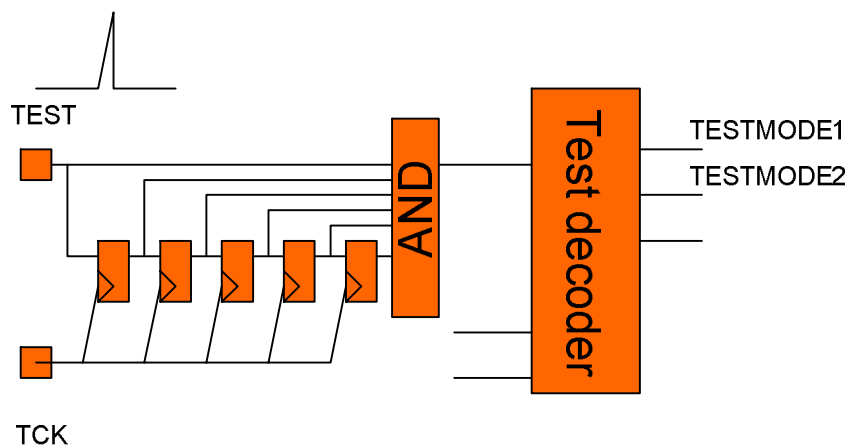


Figure 50: Spike Filter Implementation

A special spike filter from Renesas Electronics, clocked by TCK, is fitted for the test input TEST:



4.2 Oscillator Circuit

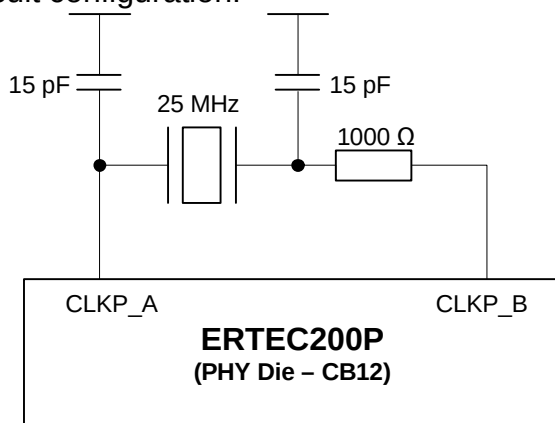
The oscillator on the PHY-Die (CB12 technology, see 2.3.9.2.2) has the following parameters:

Parameter	Description	Min	Typ	Max	Unit
C _{IN}	Input capacity (incl. package), CLKP_A		4,2		pF
C _{OUT}	Output capacity (incl. package), CLKP_B	3	4	5	pF
F	Frequency		25		MHz
V _{IH}	High Level Input Voltage	2	-	-	V
V _{IL}	Low Level Input Voltage	-	-	0,8	V
F _{Tolerance}	Frequency Tolerance over all (lifetime and temperature)	- 50	0	+ 50	ppm

The integrated oscillator is a pierce-gate oscillator.

4.2.1 Circuit with Ext. Crystal

The on-chip crystal oscillator (ASIC pins: CLKP_A, CLKP_B) of the ERTEC 200P has a pin input capacity (incl. package) of **4.2 pF** and requires a 25 MHz crystal with following circuit configuration:



Note:

The input capacity of the integrated oscillator TDOSAC33N32M in the ERTEC 200P is **4.2 pF** (this includes the effect of the package)

The following type of ext. quartz should be used:

- Epson TSX-3225 25MHz 10pF

Parameter	Description	Min
F	Nominal (fundamental) Frequency	25MHz
ESR _{max}	Maximal Equivalent Series Resistor	40Ω
C _L	Load Capacitance	10pF
DL _{max}	Maximal Drive Level	300μW
F _{Tolerance}	Frequency Tolerance	40ppm (@20 years)
T _{Operation}	Operating Temperature	-40°C to +85°C

Note that the oscillator frequency must be 25 MHz, otherwise the PLL will not operate properly. If a different crystal is used the following conditions have to be met:

- Crystal with 25MHz with a maximum of ± 50ppm over the whole lifetime and temperature range.

- The values for R_v , C_{in} , C_{out} has to be calculated accordingly.

PCB layout hints:

- Place the input and output pins of the oscillator and the resonator and external components close to each other, and keep wiring as short as possible.
- Make the wiring between the ground side of the capacitor and the ground pin of the ERTEC 200P as short and as thick as possible.
- Keep the lead wire of the resonator and capacitor as short as possible, and fix the resonator and capacitor to the printed circuit board to keep the influence of mechanical vibrations to a minimum.
- Lay out the external constant portion so that it is surrounded by GND insofar as possible.

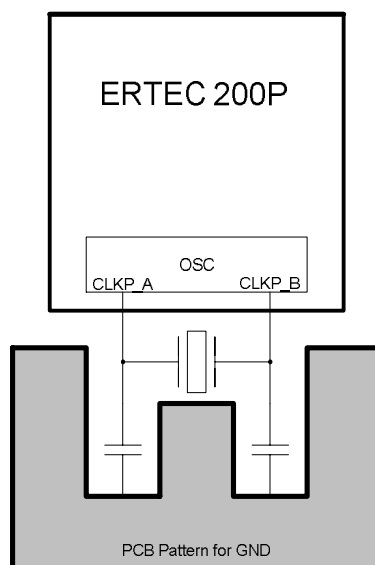


Figure 51: Oscillator Circuitry Layout Example

Note for HW-Developer:

Each PCB Design has to measure the oscillator startup time and must adjust the circuit to fulfill required timing or increase reset puls accordingly.

To meet the FSU (fast startup) requirement, the crystal startup time must be < 20 ms. The circuitry above requires a startup time of ~ 80 ms (worst case). To meet the FSU requirements, a circuit with an ext. crystal oscillator is required.

	Startup Time			
	t[ms] @ -20°C	t[ms] @ 25°C	t[ms] @ 60°C	t[ms] @ 85°C
ERTEC 200P	n.a.	9,04	13,5	82,4
ERTEC 200P Step2	5,6	9,28	15,6	78

4.2.2 Circuit with External Crystal Oscillator

If the ERTEC 200P is wired with an external crystal oscillator at CLKP_A instead of a crystal (see 4.2) (**important:** CLKP_B must **not** be connected - open), the external crystal oscillator must comply with the following parameters at CLKP_A:

Parameter	Description	Min	Typ	Max	Unit
f_{IN}	external clock source frequency	¹⁾	25	¹⁾	MHz
V_{IH}	CLKP_A high level voltage	2	3,3	$V_{DD_{actual}}$ ³⁾	V
V_{IL}	CLKP_A low level voltage	0	-	0,8	V
t_{RFC}	CLKP_A rise or fall time	0	1	4	ns
t_w	CLKP_A high or low time	16 ²⁾	20 ²⁾	24 ²⁾	ns
t_{JIT}	CLKP_A jitter tolerace	-	20	-	ps (RMS)
DuCy	CLKP_A duty cycle	40	50	60	%

¹⁾ +/- 50ppm (PN requirement)

²⁾ t_w has been calculated for $f_{IN(TYP)} = 25$ MHz, e.g. $t_w(MIN) = 10 * (DuCy(MIN) / f_{IN(TYP)})$

³⁾ $V_{DD_{actual}}$: actual IO voltage

Note:

When an ext. crystal oscillator is used with CB12 technology (the integrated oscillator is located on the PHY-Die, see 2.3.9.2.2), the integrated quartz oscillator does **not** need to be configured in bypass mode.

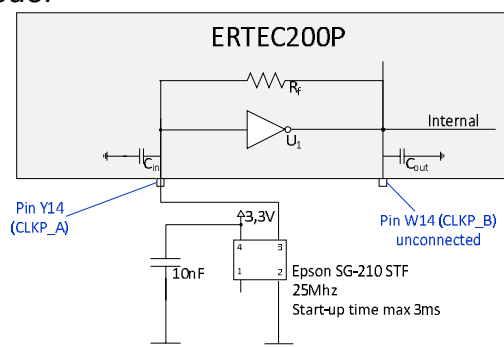


Figure 52: Connection of an external oscillator

We recommend the following quartz oscillators for connection to the ERTEC 200P:

		FSU (Fast Start-Up) Times				Unit
		- 20°C	+ 25°C	+ 85°C	+ 105°C	
Epson SG210STF-L 85°C	A5E34133756	1.72	2.08	2.28	-	ms
Epson SG210STF-Y 105°C	A5E35460986	1.68	2.12	2.36	2.52	ms

4.3 PLL Wiring (Power Supply)

Since GND noise may affect VDD through C1 and C2, GND must be stable.

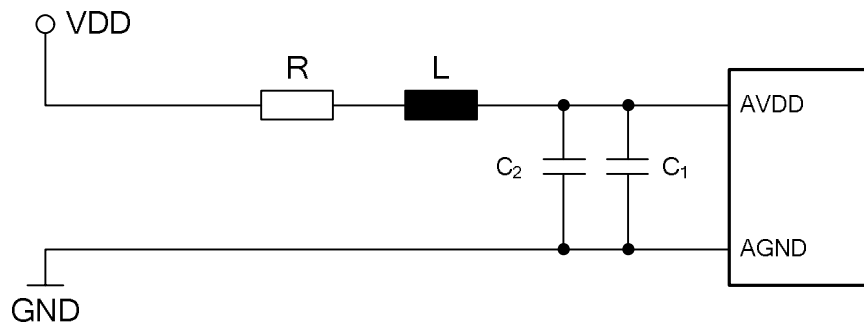


Figure 53: Recommended for PLL Power Supply Filter

Dimensioning the PLL Power Supply Filter

$$\xi = \frac{R + R_L}{2} \cdot \sqrt{\frac{C_1 + C_2}{L}} \quad (\text{damping factor})$$

$$f_0 = \frac{1}{2\pi\sqrt{L \cdot (C_1 + C_2)}} \quad (\text{cut off frequency in Hz})$$

where $R + R_L \leq 0.8 \, \Omega$, $C_2 \geq 10 \, \mu\text{F}$, $\xi \approx 0.7$ (should not be less than 0.7 to decrease the sensitivity for resonance), $f_0 \leq 5 \text{ kHz}$

Example:

With $R + R_L = 0.8 \, \Omega$, $L = 22 \, \mu\text{H}$, $C_2 = 68 \, \mu\text{F}$, $C_1 = 0.3 \, \mu\text{F}$ (ceramic capacitor) this results in $f_0 = 4.1 \text{ kHz}$ and $\xi = 0.70$.

C_1 should be an SMD capacitor (e.g. ceramic 300 nF) for elimination of high frequency components. Be sure to place it as close as possible to a power supply pin.

4.4 Test Signal Configuration

Test Pins	External Wiring in Function Mode	Internal Wiring
TEST	low-impedance to ground (GND)	- pull-down (ca. 50kΩ) - spike filter < 40 ns
TACT	low-impedance to ground (GND)	- pull-down (ca. 50kΩ) - spike filter < 40 ns
TAP_SEL	low-impedance to ground (GND)	- no pull! - spike filter < 40 ns
TMC1	straight to ground (GND)	- no pull! - no spike filter!
TMC2	straight to ground (GND)	- no pull! - no spike filter!

The module design must comply **strictly** with the external test signal configuration given in the table.

For the module test, the ERTEC 200P can be switched to boundary scan mode by setting the test signals as shown in the table below:

TACT	TAP_SE L	TEST	TMC1	TMC2	Description
0	0	0	0	0	Function mode
1	1	0	0	0	ERTEC 200P boundary scan mode

The boundary scan is controlled over the JTAG interface (see 3.3.8).

4.5 JTAG Wiring

The JTAG interface is an interface over which the boundary scan register can be controlled or which can be used for ERTEC 200P debugging. The JTAG reset is purposefully implemented without an internal pull resistor so that various debugger connections are possible (Hitex or MC). A filter integrated in ERTEC 200P ensures that spikes ≤ 40 ns (best case) at JTAG reset XTRST are suppressed (see 4.1.4). A spike at XTRST is not usually forwarded to the JTAG controller as this would require a sequence over TDI/TMS and TCK.

The table below shows the various recommendations for external pull-up/down configurations of the JTAG interface signals.

JTAG Signal	Signal Direction	ERTEC 200P internal pull (siehe Kap. 3.2 und 4.9)	Circuit for Production (ETB <u>not</u> accessible via AHB)	Circuit for Debugging, ARM recommended (ETB accessible via AHB)	recommended JTAG circuit	Recommendation from Debug supplier (Lauterbach)
XTRST	in	-	10k pulldown	4k7 Pullup	10k Pulldown default and 4k7 Pullup Assembly option	You should place a pull-down resistor (1k - 47k) on this signal on target side, although this is not JTAG conform. It ensures the on-chip debug logic is inactive when the debugger is not connected.
RTCK	in/out	-	<i>not necessary</i>	4k7 Pulldown	4k7 Pulldown	If this is not required, then it can be used to compensate the propagation delays on driver and cable. This allows to reach higher JTAG clock frequencies. Therefore you need to feed-back the TCK signal buffered or unbuffered to this line. On an unbuff-

						ered feed-back it might have negative effect on signal reflection. Better provide a chance to cut the connection on the target (jumper or solder bridge) in case problems arise.
TCK	In	Pulldown	<i>not necessary</i>	4k7 Pulldown	4k7 Pulldown	You should place a pull-up or pull-down resistor (1k - 47k) on this line in order to give it a defined state even when the line is not driven by the debugger.
TDI	In	-	<i>not necessary</i>	4k7 Pullup	4k7 Pullup	You can place a pull-up or pull-down resistor (1k - 47k) on this line to ensure a defined state even when the line is not driven by the debugger.
TMS	In	Pullup	<i>not necessary</i>	4k7 Pullup	4k7 Pullup	You can place a pull-up or pull-down resistor (1k - 47k) on this line in order to give it a defined state even when the line is not driven by the debugger.
XSRST	In	Pullup	<i>not necessary because of internal pullup</i>	<i>not necessary, because of internal Pullup</i>	<i>not necessary, because internal Pullup</i>	There might be the need to place a pull-up (1k - 47k) on target side to avoid unintentional resets when the debugger is not connected and probably to strengthen the weak 47k pull-up in the debug cable.
TDO	out	-	<i>not necessary</i>	<i>not necessary</i>	33Ohm serial resistor	You can place a 33 series resistor close to the processor for series termination. You can place a pull-up or pull-down resistor (1k - 47k) on this line.

Note on module development:

To achieve the best possible resistance to interference on the module (see "Circuit for Production" column in the table above), the XTRST pin on the module must have a 10 KOhm pull-down. This deactivates JTAG interface during operation, which means that noise pulses affecting the individual JTAG signals can no longer affect ERTEC 200P function. If a debugger is used at the JTAG interface, the pull-down has no effect as the debugger actively pulls the XTRST signal to '1'. Only AHB access to the ETM trace buffer SRAM (ETB, see 2.3.1.3.5 and 3.3.9) is not possible in this configuration. If access from an AHB master (e.g. ARM926) to this SRAM (ETB) is required for test purposes, the XTRST pin requires a 10 KOhm pull-up. Please note that AHB access to the ETM trace buffer SRAM is only possible in this case if no debugger is connected.

4.6 PHY Wiring

4.6.1 PHY-TX Wiring

Following design hints are recommended for the UTP interface:

- RX and TX pairs must be routed 100Ohm differential
- Differential length for each pair must match in length
- Transformer must be placed as close as possible to the ERTEC 200P
- Centre Tap of transformer must be connected with 100Ohm series resistor to 3.3V_Analog.
- 50Ohm termination resistors must be placed as close as possible to the ERTEC 200P
- An external 12.4kOhm pull down resistor on pin EXTRES must be placed as close as possible to ERTEC 200P
- Analog signals must be placed referenced to analog ground plane or common plane (same as digital) and must be not coupled with other signals.

The picture below shows a typical UTP circuit with separate centre taps on the magnetic:

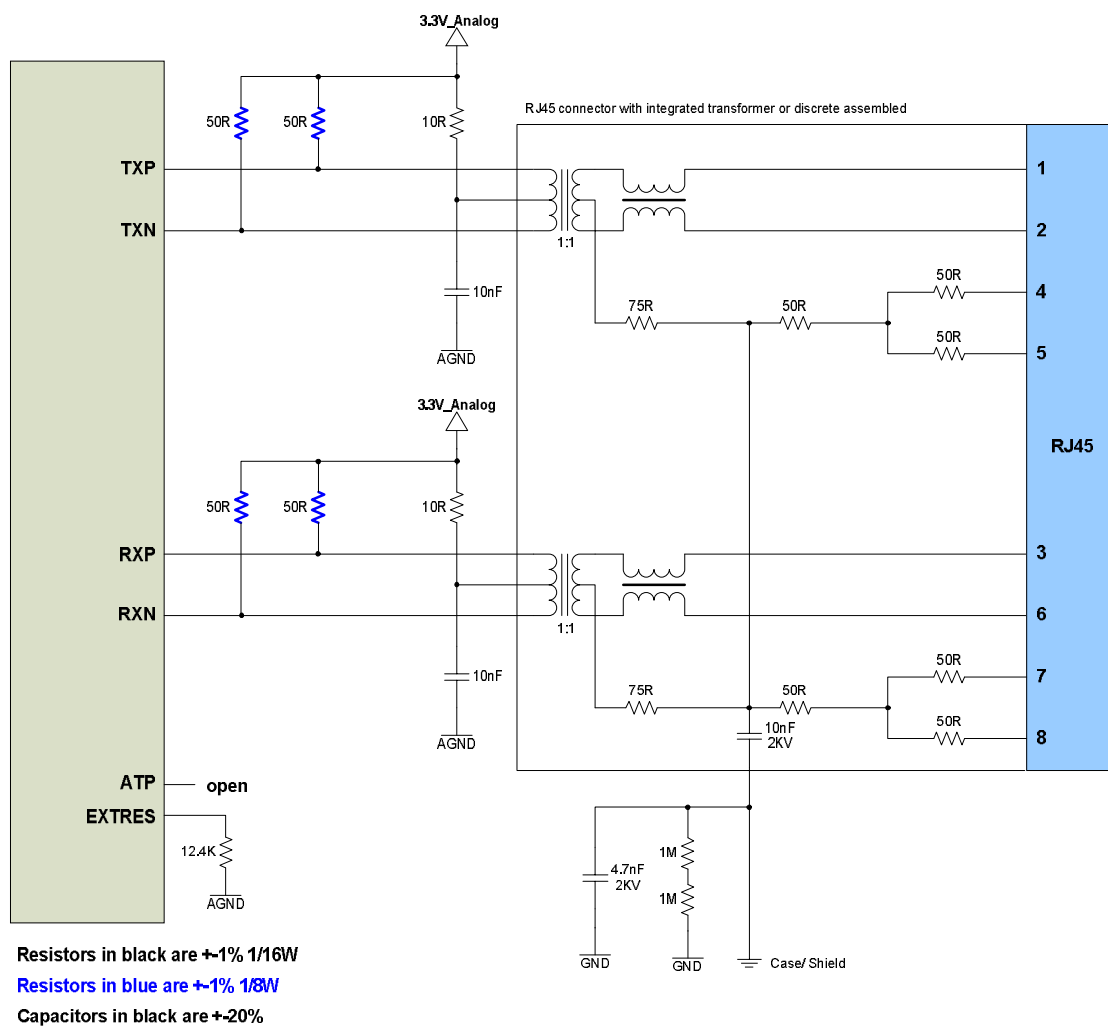
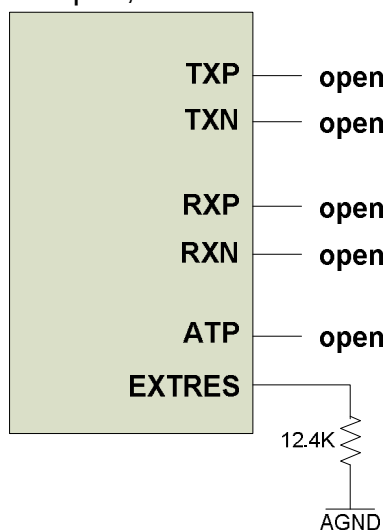


Figure 54: UTP circuit

Note: UTP interface must full fill ANSI X3.263-1995 FDDI specifications.

4.6.1.1 PHY-TX Wiring – not used

Unused UTP port should be left open, but **EXTRES** must be connected as recommended.



Resistors in black are $\pm 1\%$ 1/16W

UTP circuit unused

4.6.2 PHY-FX Wiring

Following design hints are recommended for the FX interface:

- It is strongly recommended to use Profinet compliant POF Transceiver QFBR-5978AZ from company Avago.
- RX and TX pairs (LVPECL) must be routed 100Ohm differential
- Differential length for each pair must match in length
- Transceiver must be placed as close as possible to the ERTEC 200P
- 150Ohm series resistors in TX path must be placed as close as possible to the ERTEC 200P
- The RX termination (130Ohm pull up and 82Ohm pull down) must be placed as close as possible to ERTEC 200P
- RX and TX Power supply on transceiver should separately filtered, see datasheet transceiver.
- Care must be taken on level translation between transceiver SD pin and ERTEC 200P PHY input, see recommended circuit below.
- Signals must be placed referenced to digital ground plane or common plane and must be not coupled with other signals.
- Don't connect analog VDD/ AGND (related pins on ERTEC 200P: VDD33ESD, VDDACB/ PVSSA, VSSPLLCB) to FX circuit.

The picture below shows the recommended FX circuit:

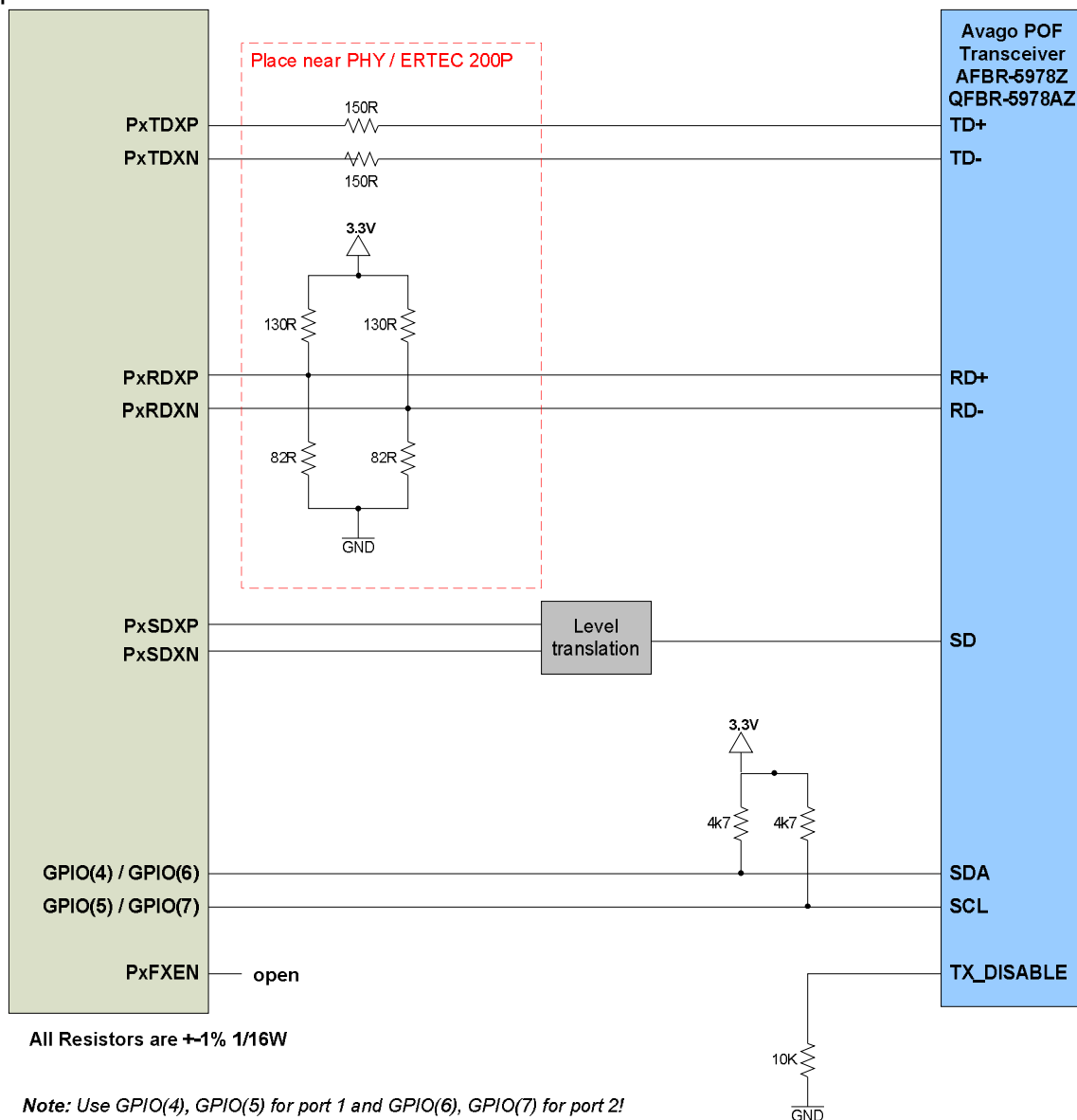


Figure 55: FX circuit

4.6.2.1 PHY-FX Wiring – not used

PxTDXP/N Signals on unused FX port should be left open, **PxRDXP/N** and **PxSDXP/N** inputs should directly connected to GND, the GPIOs could used for alternate function.

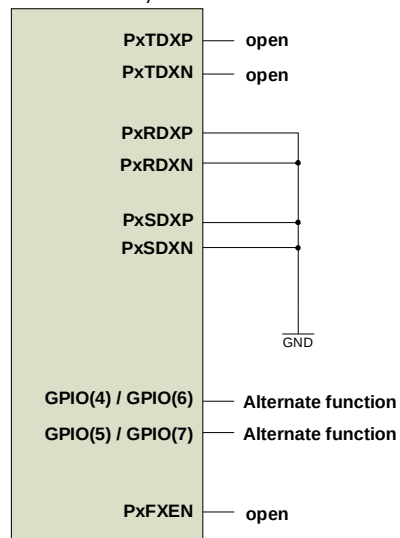


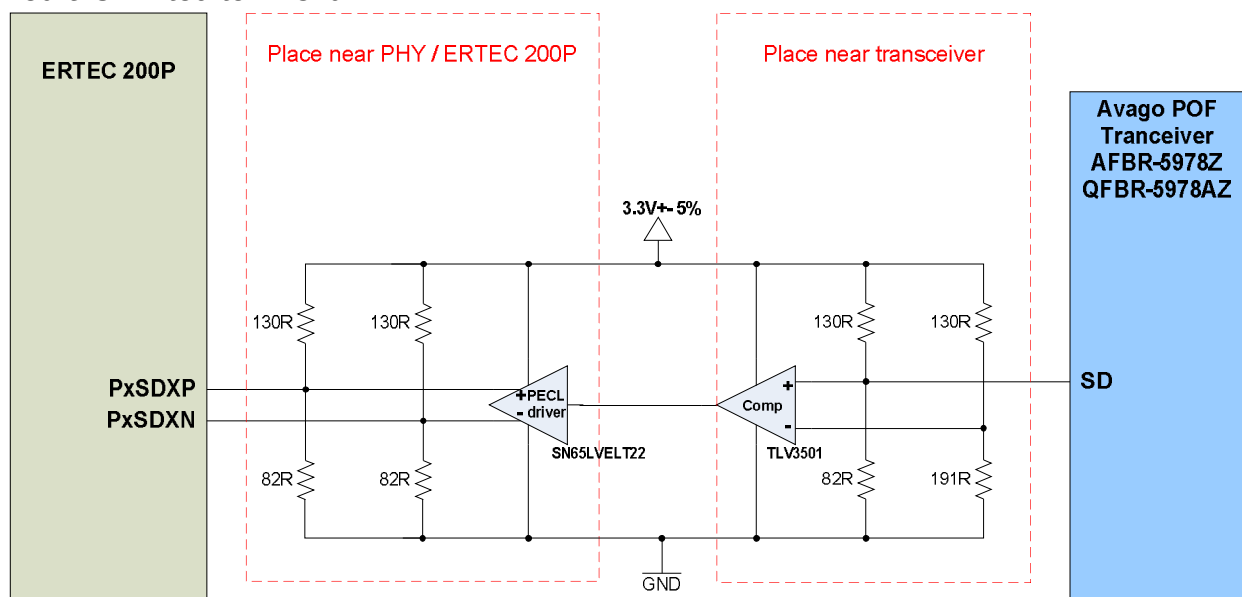
Figure 56: FX circuit unused pins

4.6.2.2 PHY-SD Wiring – Avago QFBR-5978AZ

The Avago QFBR-5978AZ has a single ended output and ERTEC 200P has a differential LVPECL input.

4.6.2.2.1 PxSD circuit

The following level translation circuit is recommended **by direct connecting of PxSD signals**. Comparator should be placed near transceiver and PECL driver near ERTEC 200P. The 3,3V supply voltage tolerance for **POF transceiver and SD level translation circuit** is limited to **+/- 5%**.



Second channel of comparator and PECL driver could used for other FX port!
All Resistors are +/-1% 1/16W

SD level translation circuit

4.6.2.2.2 GPIO circuit

The following level translation circuit is recommended **by using of GPIOs** (see chapter 2.3.10.9.21). Comparator should be placed near transceiver. The 3,3V supply voltage tolerance for **POF transceiver** is limited to **+ - 5%**.

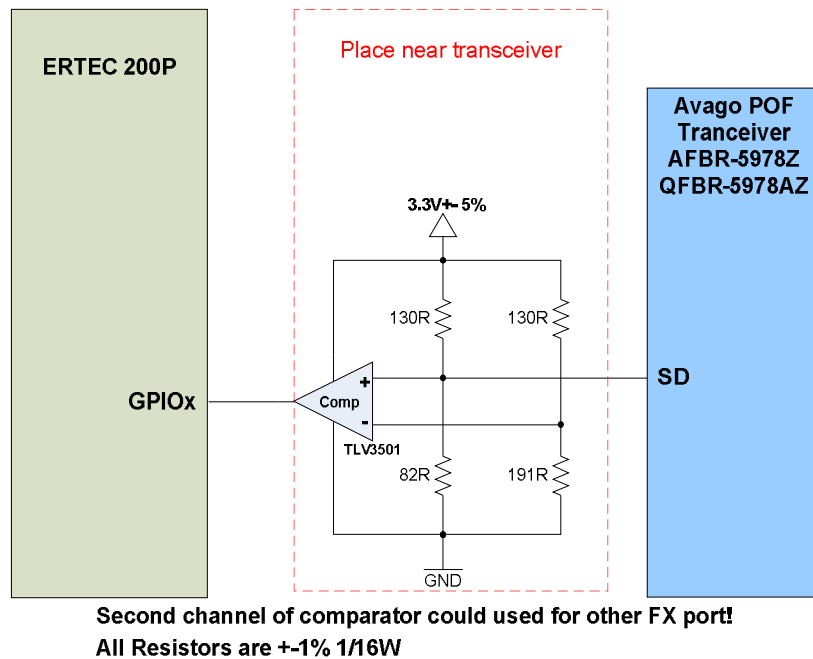


Figure 57: SD level translation circuit

4.7 Wiring of pins not used

The following applies in general:

- IN and INOUT pins that do **not** have an internal pull resistor must be connected to an external pull resistor.
- If the internal pull-up/pull-down resistor of the GPIOs is **deactivated**, it must be activated with the SW or an external pull-up/pull-down resistor is to be connected.

For wiring PHY pins that are not used, see xxx

Signal	Signal description	Dir	Function description	Ball
ATP	Analog Test Function	out	Analog Test Enable to monitor or drive specific nodes in the analog circuit during analog test. For debugging purposes it is recommended to have this pin accessible for an oscilloscope on a PCB. This pin is not used in normal operation.	K18
TEST	IC-Test-Mode	in	IC Test Mode Select signal for ASIC test. For normal operation this pin must be connected by a 1k Ohm Pull down resistor to GND.	R9
TMC1	Testmode_1	in	Test Mode Control Signal for ASIC test. For normal operation this pin must be connected directly to GND.	F12
TMC2	Testmode_2	in	Test Mode Control Signal for ASIC test. For normal operation this pin must be connected directly to GND.	R14
TACT	TESTACT-TAP-RESET	in	Special Test Mode TAP Controller Used for Boundary scan test. For normal operation this pin must be connected by a 1k Ohm Pull down resistor to GND.	R12
TAP_SEL	TAP Select	in	TAP Select	T7

Signal	Signal description	Dir	Function description	Ball
			Used for Boundary scan test. For normal operation this pin must be connected by a 1k Ohm Pull down resistor to GND.	
P1FXEN	Port 1 Fiber Optic Enable	out	Port 1 Fiber Optic Enable This pin is not used and should be unconnected. The Fiber Optic transceiver should be enabled all the time by proper pull resistor.	C9
P2FXEN	Port 2 Fiber Optic Enable	out	Port 2 Fiber Optic Enable This pin is not used and should be unconnected. The Fiber Optic transceiver should be enabled all the time by proper pull resistor.	Y8
TESTDOUT5	Test Output 5	out	Test Output 5 This pin is not used and should be unconnected.	B8
TESTDOUT6	Test Output 6	out	Test Output 6 This pin is not used and should be unconnected.	A8
TESTDOUT7	Test Output 7	out	Test Output 7 This pin is not used and should be unconnected.	V10
CLK_O_SDRAM2	EMC SDRAM Clock Out	bi	Clock Output SDRAM 2 In normal operation this signal is connected to second SDRAM, upper 16 Bit. In case where just one SDRAM is implemented an external 10k Ohm Pull up resistor must be connected.	H5
CLK_O_BF0	EMC Burst Flash CLK Out	bi	Clock Output Burst Mode Flash 0 In normal operation this signal is connected to CLK_I_BF. In case where no Burst Mode Flash is implemented an external 10k Ohm Pull up resistor must be connected.	M5
CLK_O_BF1	EMC Burst Flash CLK Out	bi	Clock Output Burst Mode Flash 1 In normal operation this signal is connected to first Burst Mode Flash, lower 16 Bit . In case where no Burst Mode Flash is implemented an external 10k Ohm Pull up resistor must be connected.	N5
CLK_O_BF2	EMC Burst Flash CLK Out	bi	Clock Output Burst Mode Flash 2 In normal operation this signal is connected to second Burst Mode Flash, upper 16 Bit . In case where just one Burst Mode Flash is implemented an external 10k Ohm Pull up resistor must be connected.	M6
CLK_I_BF	EMC Burst Flash CLK In	in	Clock Input Burst Mode Flash In normal operation this signal is connected to CLK_O_BF0 . In case where no Burst Mode Flash is implemented an external 10k Ohm Pull up resistor must be connected.	L5
NC	Reserved	in	Reserved This pin is not used and must be connected by a 1k Ohm Pull up resistor to VDD33.	P6

Figure 58: Recommendation for handling special function signals

4.8 Operating Conditions

	Min.	Max.
Ambient Temperature	- 40 °C	+ 85 °C
Junction Temperature	- 40 °C	+ 125 °C
VCC-IO	1,8 V – 10%	1,8 V + 10%
	3,3 V – 10%	3,3 V + 10%
Core ERTEC 200P Die	1,2 V – 0,1V	1,2 V + 0,1V
Core PHY Die	1,5 V – 10%	1,5 V + 10%

The supply voltage tolerance includes the supply voltage ripple, i.e. the supply voltage plus ripple must not exceed the upper limit / violate the lower limit.

4.8.1 Power Up

There is no set order for switches on the supply voltages. However, all must be activated within a period of 100 ms.

4.8.2 Wiring of CTRL-STBY

To ensure that the outputs remain at high-impedance upon power up (see 00), make sure that the inputs pins *CTRL_STBY0-2* are in the PCB layout are switched to '0' (GND) during this time (for example, link to XRESET pin), i.e. that the CTRL_STBY function is active. Please note that the CTRL_STBY pins control both 3.3 V and 1.8/3.3 V pads. **CTRL_STBY0-2 must** therefore be deactivated **with the corresponding supply voltage** of the controlled IO buffers e.g. operation of the XHIF interface with 1.8 V → deactivation of CTRL_STBY1, 2 = 1.8 V).

The table below shows which ERTEC 200P IO signals (see 3.2) are controlled with the CTRL_STBY0-2 input pins:

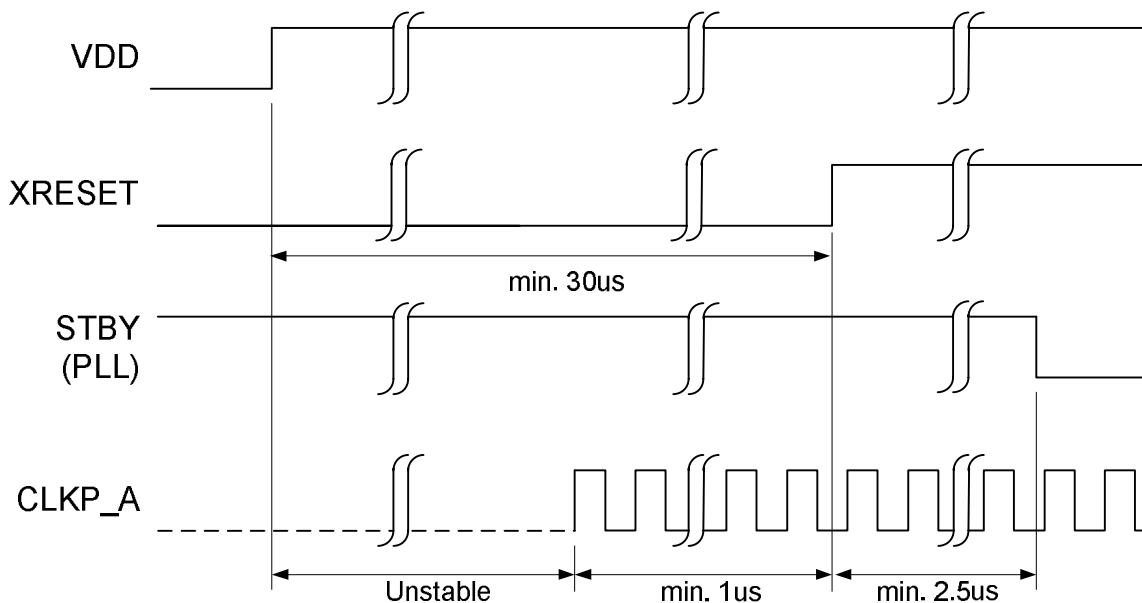
CTRL_STBY0 (3,3V)	CTRL_STBY1 (1,8V / 3,3V)		CTRL_STBY2 (1,8V / 3,3V)	
GPIO31:0	GPIO92	XHIF_D28	GPIO95:93	XHIF_D31-29
JTAG_RTCK JTAG_TDO	GPIO79:64	XHIF_D15-0	GPIO91:80	XHIF_D27-16
<i>P2XTXEN</i> ¹⁾ <i>P2XTXER</i> ¹⁾ <i>P2XTXD0-3</i> ¹⁾ <i>P2XPHYEN</i> ¹⁾ <i>P2XFXMODE</i> ¹⁾ <i>P2XAUTOMDIXEN</i> ¹⁾ <i>P2SDXN</i> ¹⁾ <i>P2SDXP</i> ¹⁾	GPIO62:46	XHIF_XBE2-0 XHIF_XCS_M XHIF_XCS_R XHIF_XRD XHIF_XWR XHIF_XIRQ XHIF_XRDY XHIF_SEG_2-0	GPIO63	XHIF_XBE3

XSRST		XHIF_A19-15		
A_PHY_1				
A_PHY_2				
L_PHY_1				
L_PHY_2				
REF_CLK	GPIO44:32	XHIF_A13-1	GPIO45	XHIF_A14

¹⁾ Not visible as IO signals at the ASIC, but wired in the interposer of the ASIC package between ERTEC-Die and PHY-Die

The other pins connected to the Die, CTRL_STBY3-9, control the remaining signal pins (EMC with 1.8 V; PHY with 3.3 V). These CTRL_STBY pins are not connected to package balls, but instead fixed to the corresponding VDD in the interposer of the ASIC package.

4.8.3 Power-Up Sequence (PLL)



The standby signal (STBY) to the PLL is an extension of XRESET by ca. 2.5 μ sec.

4.8.4 PLL Behavior

4.8.4.1 following crystal break

If the ext. crystal breaks, i.e. CLK_A, CLK_B are

- open
- clamped to '0'
- clamped to '1',

a frequency of 100 MHz ... 300 MHz is established at the PLL output (free-running frequency).

4.8.4.2 upon temporary clock failure

The period jitter is considerably less than 50 ps and is therefore within the specified range provided the recommended filter connections for AVDD decoupling are used.

If the reference clock for the ERTEC 200P PLL fails, the frequency changes continuously until it reaches the final value after ca. 30 μ s.

4.8.5 Readiness of internal resources once a reset is cleared

Please note that access, for example over the XHIF interface, to internal ERTEC 200P resources immediately after a reset is cleared is only permitted once all internal initialization actions are complete (see also boot mode startup times in 2.3.1.5.1.4).

Duration of initialization actions after XRESET deactivation:

PLL standby duration:	2.5 μ s (see 0)
PLL lockup duration:	1000.0 μ s (see 2.3.9.2)
EMC Init Done ¹⁾ :	233.0 μ s
Total duration:	1235.5 μs

¹⁾ Alongside EMC Init_Done (see also SDRAM_Refresh register in 2.3.5.8), a number of internal SRAMs are also initialized during this time (see also EDC_INIT_DONE register in 2.3.10.9.22). EMC INIT_Done is detailed here as it takes the longest.

4.9 Pull-up/Pull-down Resistor Values

($V_{DD} = 3.3 \text{ V} \pm 0.3 \text{ V}$, $V_{DD} = 1.8 \text{ V} \pm 0.15 \text{ V}$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$)

Parameter		Library expression	Min.	Typ.	Max.	Unit
Pull-up resistor	3.3 V buffer	50 k Ω	40.5	51.9	70.1	k Ω
	1.8 V buffer	50 k Ω	38.6	49.7	66.9	k Ω
Pull-down resistor	3.3 V buffer	50 k Ω	37.0	49.6	70.4	k Ω
	1.8 V buffer	50 k Ω	35.8	47.6	66.4	k Ω

4.10 Schmitt Trigger Characteristics

	Vifall min	Virise max	Vhyst min
3.3V spec (prelimi- nary)	0.8	2	0.3
1.8V target spec	0.4	1.3	0.2

4.11 Module and ASIC Code (Chip ID)

The ASIC code (version number of the ASIC) can be read from the ID register in the SCRB (see 2.3.10.9.1).

4.12 Power dissipation

Total ERTEC 200P power dissipation: **max. 1.23 W** (with: max. voltage, TA= 85°C; worst process)

typ. 0.72 W (with: typ. voltage, TA= 25°C; typ. process)

Power dissipation ERTEC 200P	Typ. voltage (max.)	Power dissipation		Current consump- tion	
		typ.	max.	typ.	max.
IO pads, 1.8 V (EMC) IO pads, 1.8 V / 3.3 V (XHIF) IO pads, 3.3 V (GPIO, ..)	1.8 V (+ 10%) 1.8 V / 3.3 V (+ 10%) 3.3 V (+ 10%)	72 mW 9 mW / 31 mW 24 mW	93 mW 22 mW / 73 mW 29 mW	40 mA 5 mA / 9 mA 7 mA	47 mA 11 mA / 20 mA 8 mA
Core (ARM926, clock, logic, memories)	1.2 V (+ 0.1 V)	228 mW	538 mW	190 mA	413 mA
Core (PLL)	1.2 V (+ 0.1 V)	10 mW	12 mW	8 mA	9 mA
Ethernet Double PHY 100Base TX	3.3 V (+ 10%) 1.5 V (+ 10%)	139 mW 225 mW	210 mW *) 277 mW	80 mA 152 mA	80 mA 168 mA
ERTEC 200P 100Base TX	1.2 V (+ 0.1 V) 1.5 V (+ 10%) 1.8 V (+ 10%) 3.3 V (+ 10%)	0.72 W	1.23 W	198 mA 152 mA 40 mA 96 mA	422 mA 168 mA 47 mA 108 mA
Ethernet Double PHY 100Base FX	3.3 V (+ 10%) 1.5 V (+ 10%)	5 mW 100 mW	6 mW 125 mW	1.5 mA 67 mA	1.7 mA 76 mA
ERTEC 200P 100Base FX	1.2 V (+ 0.1 V) 1.5 V (+ 10%) 1.8 V (+ 10%) 3.3 V (+ 10%)	0.47 W	0.87 W	198 mA 67 mA 40 mA 18 mA	422 mA 76 mA 47 mA 30 mA

Note:

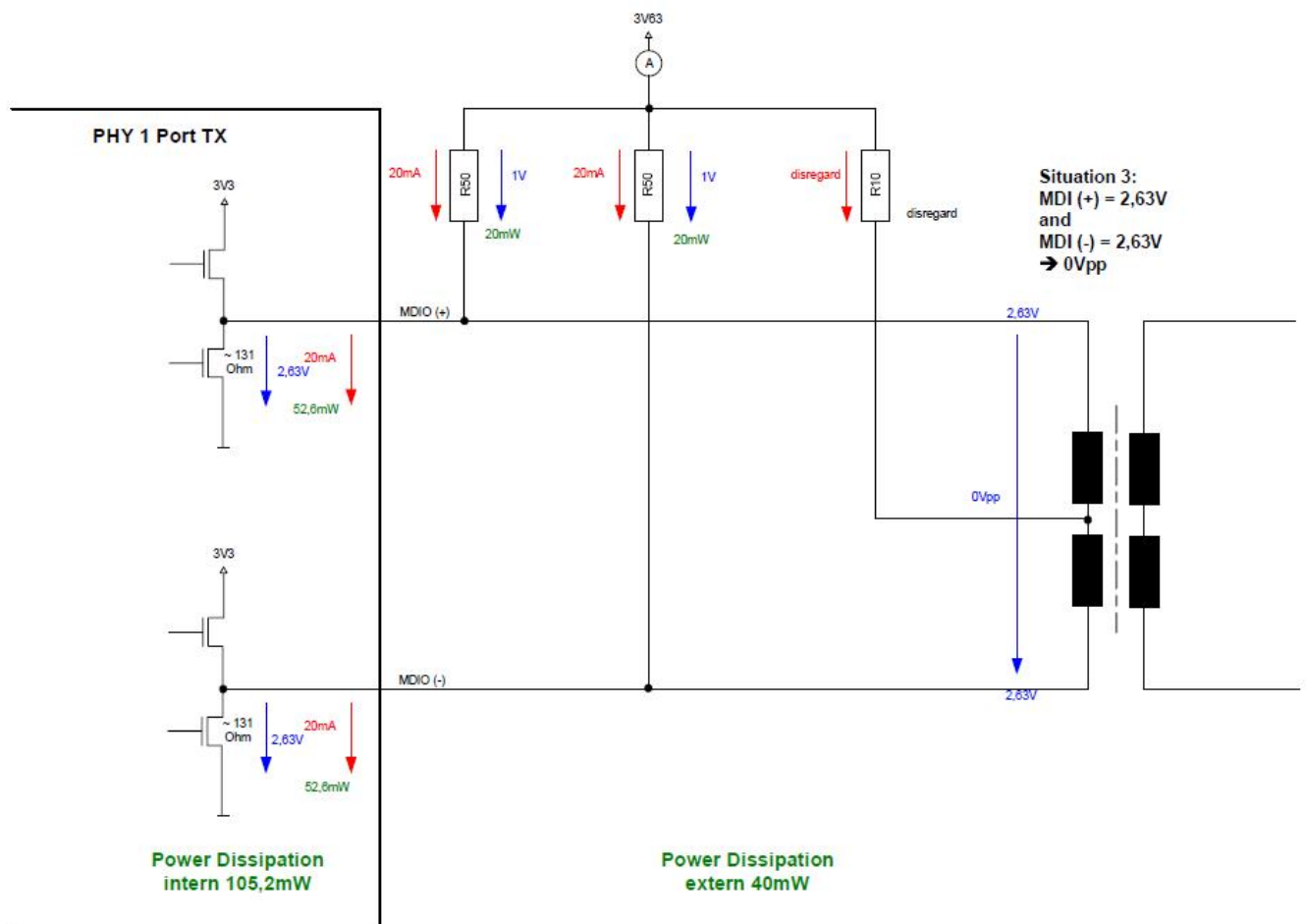
The typical and maximum values given here are taken from the measurements of corner samples for ERTEC 200P Step1 and the power dissipation simulations for ERTEC 200P-2 standardized on that basis.

*) The power dissipation shown is the power dissipation in the integrated PHY (105.2 mW per port= 210 mW), which cannot be calculated on the basis of the current consumption (constant 80 mA) as the external power dissipation (40 mW per port) must not be included in the power dissipation calculation for the ERTEC 200P ASIC.

The power dissipation relevant for the PHY supply is calculated as follows:

- internal ERTEC 200P power dissipation: $(2 \times 105.2 \text{ mW}) + 277 \text{ mW}$ +
- external power dissipation: $2 \times 40 \text{ mW}$

→ **567 mW** (see figure below)



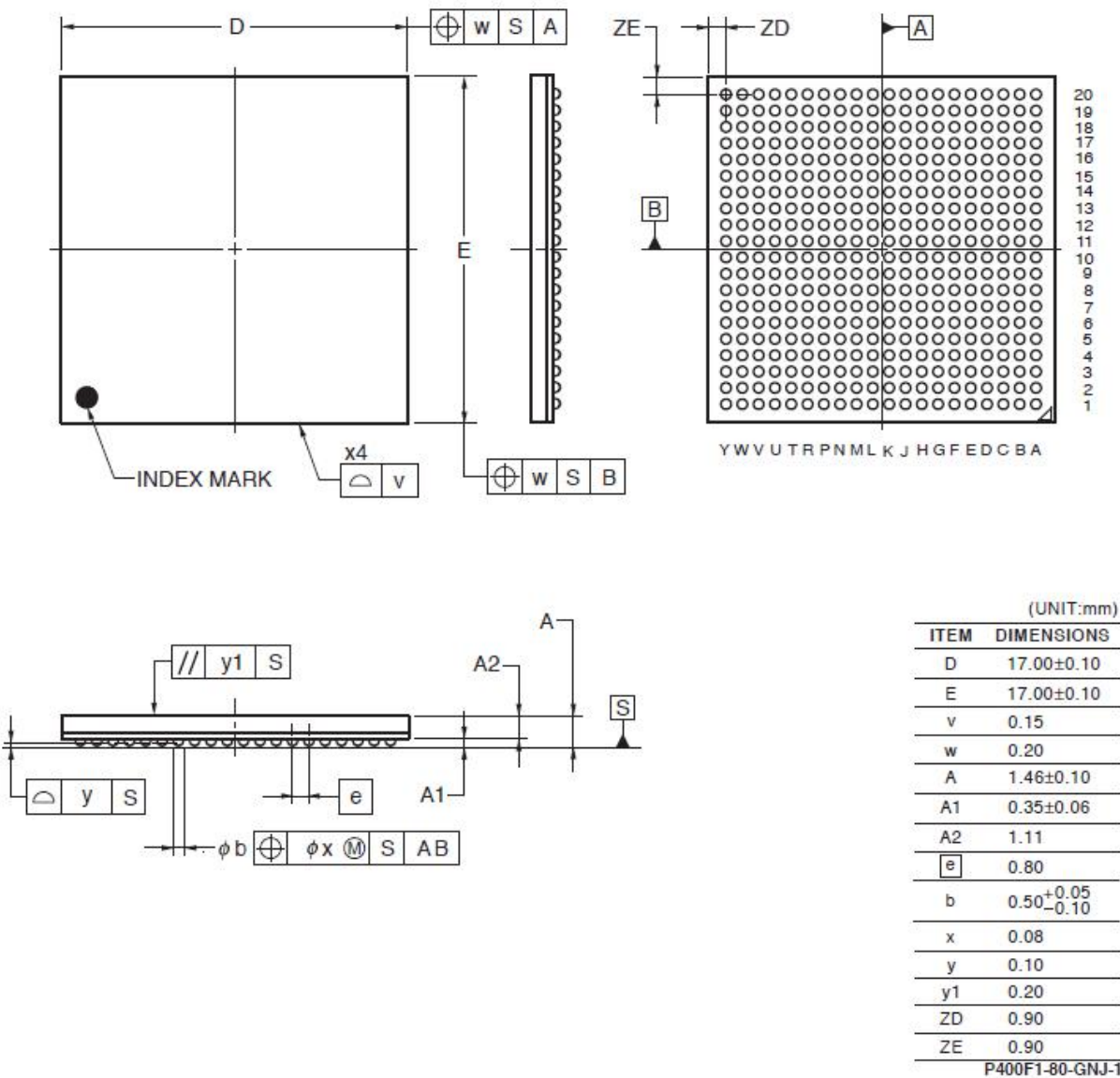
5 PACKAGE

ERTEC 200P and PHY Die wiring to the balls (see 0) in the ERTEC 200P package is over an interposer (package internal PCB) with 4 layers.

5.1 Package Drawing

The ERTEC 200P has a 400-ball full grid SIP-FPBGA package. The ball pitch is 0.8 mm. The package size is 17 mm x 17 mm. The inner rows of balls act as the voltage supply and as thermal balls (see 5.2).

400-PIN PLASTIC FBGA (17x17)



©2010 Renesas Electronics Corporation. All rights reserved.

Figure 59: 400-ball SIP-FPBGA

5.2 Ball Layout

Top-View

	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
Y	Y20(33) VDD33	Y19(34) GND	Y18(37) GPIO28	Y17(36) GPIO29	Y16(35) VDD33	Y15(36) GND	Y14(33) XCLK1	Y13(32) GND	Y12(31) P2RDXN	Y11(30) P2SDXN	Y10(29) GND	Y9(28) P2TDXN	Y8(27) P2FXEN	Y7(26) VDD33	Y6(25) GND	Y5(24) VDD_EMC	Y4(23) D0	Y3(22) D15	Y2(21) GND	Y1(20) VDD_EMC	Y
W	W20(40) GND	W19(111) GPIO24	W18(110) GPIO25	W17(107) GPIO26	W16(108) REF_CLK	W15(107) XRESET	W14(106) XCLK2	W13(105) VDD33	W12(104) P2RDXP	W11(103) P2SDXP	W10(102) GND	W9(101) P2TDXP	W8(99) TDI	W7(98) RTCK	W6(97) GND	W5(97) VDD_EMC	W4(96) D3	W3(95) D13	W2(94) D11	W1(93) GND	W
V	V20(41) GPIO27	V19(111) GPIO30	V18(115) GPIO31	V17(114) GPIO105_INT	V16(113) AVDD	V15(112) AGND	V14(111) VDD15	V13(110) GND	V12(109) VDD15	V11(108) TESTOUT7	V10(107) GPIO106_INT	V9(106) GND	V8(105) VDD15	V7(104) D4	V6(103) D2	V5(102) D1	V4(101) D14	V3(100) D9	V2(99) D12	V1(98) D17	V
U	U20(42) L_PHY_1	U19(113) L_PHY_2	U18(116) GPIO104_INT	U17(233) GPIO103_INT	U16(230) GPIO23	U15(229) GPIO21	U14(228) GPIO20	U13(227) GPIO107_INT	U12(226) GPIO106_INT	U11(225) GPIO105_INT	U10(224) GPIO104_INT	U9(223) GPIO103_INT	U8(222) GPIO102_INT	U7(221) DTXR	U6(220) D7	U5(219) D5	U4(218) D6	U3(159) D8	U2(92) D10	U1(17) D10	U
T	T20(43) GND	T19(114) VDD33	T18(117) GPIO102_INT	T17(118) GPIO101_INT	T16(119) GPIO109_INT	T15(122) GPIO19	T14(217) GPIO18	T13(216) A_PHY_1	T12(215) A_PHY_2	T11(214) BYP_CLK	T10(213) GPIO17	T9(212) GPIO16	T8(211) TMS	T7(210) TDO	T6(209) XOE_DRIVER	T5(208) XOE_DRIVER	T4(217) XOE_DRIVER	T3(158) XOE_DRIVER	T2(91) VDD_EMC	T1(16) VDD_EMC	T
R	R20(44) VDD33	R19(115) GND	R18(118) GPIO22	R17(233) GPIO21	R16(230) GPIO20	R15(229) GPIO19	R14(217) GPIO18	R13(216) A_PHY_1	R12(215) A_PHY_2	R11(214) BYP_CLK	R10(213) GPIO17	R9(212) GPIO16	R8(211) TMS	R7(210) TDO	R6(209) XOE_DRIVER	R5(208) XOE_DRIVER	R4(217) XOE_DRIVER	R3(157) XOE_DRIVER	R2(90) VDD_EMC	R1(5) VDD_EMC	R
P	P20(45) P2VDDARXTX	P19(116) VDD15	P18(117) GND	P17(234) XHF_D2	P16(281) XHF_D15	P15(320) VDD33	P14(359) GND	P13(350) VDD33	P12(349) GND	P11(348) VDD12	P10(347) VDD12	P9(346) GND	P8(345) VDD33	P7(344) GND	P6(303) CHAM_CTRL	P5(266) A21	P4(215) XTRST	P3(156) TCK	P2(83) A20	P1(14) A19	P
N	N20(46) P2RXN	N19(117) P2RXP	N18(118) PYSSA	N17(235) XHF_D10	N16(282) XHF_D13	N15(321) VDD_SHIF	N14(352) VDD_SHIF	N13(351) GND	N12(350) GND	N11(373) VDD12	N10(372) VDD12	N9(371) GND	N8(370) GND	N7(343) VDD_EMC	N6(308) VDD_EMC	N5(265) CLK_O_BF1	N4(214) A18	N3(155) VDD15	N2(88) A17	N1(13) XWR	N
M	M20(47) P2TXN	M19(118) P2TXP	M18(119) PYSSA	M17(236) XHF_D9	M16(283) XHF_D6	M15(322) CTRL_STBY1	M14(353) GND	M13(352) GND	M12(351) GND	M11(350) GND	M10(350) GND	M9(349) GND	M8(348) GND	M7(342) GND	M6(307) CLK_O_BF2	M5(264) CLK_O_BF3	M4(213) A15	M3(154) VDD33	M2(87) A16	M1(12) XRD	M
L	L20(48) VDDACB	L19(119) VDDACB	L18(120) XHF_D1	L17(237) XHF_D5	L16(284) XHF_D5	L15(323) XHF_D5	L14(354) XHF_D5	L13(377) XHF_D5	L12(376) XHF_D5	L11(375) XHF_D5	L10(374) XHF_D5	L9(373) XHF_D5	L8(368) XHF_D5	L7(341) XHF_D5	L6(306) CLK_O_BF4	L5(263) CLK_O_BF5	L4(212) A15	L3(153) VDD33	L2(86) A16	L1(11) XRD	L
K	K20(49) VDDAPLL	K19(120) EXTRES	K18(121) ATP	K17(238) XHF_D8	K16(285) XHF_D7	K15(324) VDD12	K14(355) VDD12	K13(378) VDD12	K12(383) VDD12	K11(400) VDD12	K10(399) VDD12	K9(398) VDD12	K8(367) VDD12	K7(340) VDD12	K6(305) XAV_BF	K5(262) XOE_DRIVER	K4(211) XOE_DRIVER	K3(152) XOE_DRIVER	K2(85) A13	K1(10) VDD_EMC	K
J	J20(50) PITXN	J19(121) PITXP	J18(122) PYSSA	J17(239) XHF_D14	J16(286) XHF_D11	J15(325) CTRL_STBY2	J14(356) GND	J13(379) GND	J12(384) GND	J11(383) GND	J10(382) GND	J9(381) GND	J8(354) GND	J7(337) GND	J6(304) CLK_O_BF6	J5(261) CLK_O_BF7	J4(210) A18	J3(151) XOE_DRIVER	J2(84) XOE_DRIVER	J1(9) A11	J
H	H20(51) PIRXN	H19(122) PIRXP	H18(123) PYSSA	H17(240) XHF_D13	H16(287) XHF_D12	H15(326) VDD12	H14(357) VDD12	H13(380) VDD12	H12(385) VDD12	H11(384) VDD12	H10(383) VDD12	H9(382) VDD12	H8(353) VDD12	H7(336) VDD12	H6(303) VDD_EMC	H5(260) VDD_EMC	H4(209) VDD_EMC	H3(150) A0	H2(83) A10	H1(8) A9	H
G	G20(52) FVDDARXTX	G19(123) VDD15	G18(124) VDD33	G17(241) XHF_D0	G16(288) XHF_D3	G15(327) XHF_D3	G14(358) XHF_D3	G13(359) XHF_D3	G12(360) XHF_D3	G11(361) XHF_D3	G10(362) XHF_D3	G9(361) XHF_D3	G8(364) XHF_D3	G7(335) XHF_D3	G6(302) XHF_D3	G5(259) XHF_D3	G4(208) XHF_D3	G3(149) XHF_D3	G2(82) XHF_D3	G1(7) XHF_D3	G
F	F20(53) VDD33	F19(124) VDD33	F18(125) VDD33	F17(242) XHF_D0	F16(289) XHF_D3	F15(328) XHF_D3	F14(329) XHF_D3	F13(330) XHF_D3	F12(331) XHF_D3	F11(332) XHF_D3	F10(331) XHF_D3	F9(334) XHF_D3	F8(335) XHF_D3	F7(336) XHF_D3	F6(301) XHF_D3	F5(258) XHF_D3	F4(207) XHF_D3	F3(148) XHF_D3	F2(81) XHF_D3	F1(6) XHF_D3	F
E	E20(54) GND	E19(125) VDD_SHIF	E18(126) VDD_SHIF	E17(243) XHF_A3	E16(290) XHF_A7	E15(291) XHF_A4	E14(292) XHF_A2	E13(293) XHF_A1	E12(294) XHF_D2	E11(295) XHF_D1	E10(296) XHF_D0	E9(297) XHF_D1	E8(298) XHF_D1	E7(299) XHF_D1	E6(300) XHF_D2	E5(257) XOE_DRIVER	E4(206) XOE_DRIVER	E3(147) XOE_DRIVER	E2(80) VDD_EMC	E1(5) VDD_EMC	E
D	D20(55) XHF_WVR	D19(126) XHF_A13	D18(127) XHF_A13	D17(244) XHF_A13	D16(245) XHF_A13	D15(246) XHF_A13	D14(247) XHF_A13	D13(248) XHF_A13	D12(249) XHF_A13	D11(250) XHF_A13	D10(251) XHF_A13	D9(252) XHF_A13	D8(253) XHF_A13	D7(254) XHF_A13	D6(255) XHF_A13	D5(256) XHF_A13	D4(205) XHF_A13	D3(146) XHF_A13	D2(79) XHF_A13	D1(4) XHF_A13	D
C	C20(56) XHF_WVR	C19(127) XHF_A12	C18(128) XHF_A12	C17(129) XHF_A12	C16(130) XHF_A12	C15(131) XHF_A12	C14(134) XHF_A12	C13(135) XHF_A12	C12(136) XHF_A12	C11(135) XHF_A12	C10(138) XHF_A12	C9(139) XHF_A12	C8(200) XHF_A12	C7(201) XHF_A12	C6(202) XHF_A12	C5(203) XHF_A12	C4(204) XHF_A12	C3(145) XHF_A12	C2(78) XHF_A12	C1(3) XHF_A12	C
B	B20(57) GND	B19(128) XHF_A10	B18(129) XHF_A18	B17(130) XHF_A18	B16(131) XHF_A3	B15(132) XHF_A3	B14(133) XHF_A8	B13(134) GND	B12(135) P1RDXP	B11(136) P1SDXP	B10(137) P1TDXP	B9(138) P1FXEN	B8(139) P1FXEN	B7(140) P1FXEN	B6(141) P1FXEN	B5(142) P1FXEN	B4(143) P1FXEN	B3(144) P1FXEN	B2(77) P1FXEN	B1(2) P1FXEN	B
A	A20(58) VDD12	A19(129) VDD12	A18(130) VDD12	A17(61) VDD12	A16(62) VDD12	A15(63) VDD12	A14(64) VDD12	A13(65) VDD12	A12(66) VDD12	A11(67) VDD12	A10(68) VDD12	A9(69) VDD12	A8(70) VDD12	A7(71) VDD12	A6(72) VDD12	A5(73) VDD12	A4(74) VDD12	A3(75) VDD12	A2(76) VDD12	A1(1) VDD12	A

5.3 Marking (Printed)

The ERTEC 200P Step2 ASIC package is printed as follows:



5.3.1 Order codes

The Part-Number of the ERTEC 200P Step 2:

6ES7195-0BH02-0XA0	ERTEC 200P Step 2, 10 pcs
6ES7195-0BH12-0XA0	ERTEC 200P Step 2, 90 pcs
6ES7195-0BH22-0XA0	ERTEC 200P Step 2, 450 pcs
6ES7195-0BH32-0XA0	ERTEC 200P Step 2, 1000 pcs Tape and Reel

5.4 SiP – FPBGA400 Thermal Characteristics

Unit :deg. / W

Layer (PCB)	Ambient Temperature	Air Flow unit [m/s]				
			0	0.2	1	2
6	25 deg	θ_{ja}	26.4	23.4	20.0	18.0
		Ψ_{jt}	0.14	0.17	0.24	0.29
		θ_{jc}	7.1			
		θ_{jb}	11.9			
6	85 deg	θ_{ja}	23.5	21.7	19.5	17.8
		Ψ_{jt}	0.2	0.2	0.3	0.3
		θ_{jc}	7.2			
		θ_{jb}	12.1			
8	25 deg	θ_{ja}	23.6	21.7	17.7	15.9
		Ψ_{jt}	0.14	0.16	0.22	0.26
		θ_{jc}	7.1			
		θ_{jb}	10.3			
8	85 deg	θ_{ja}	21.1	19.9	17.2	15.7
		Ψ_{jt}	0.16	0.17	0.23	0.27
		θ_{jc}	7.2			
		θ_{jb}	10.4			

Note : each value is estimated one

Condition	
PCB(Layer6)	: Layer construction by Siemens
PCB(Layer8)	: Layer construction by Siemens
PKG	: 400pinFPBGA

EIA/JEDEC51-1, 51-2 define thermal characters as follows,

θ_{ja} Thermal resistance, junction-to-ambient

Ψ_{jt} Thermal characterization parameter, junction to the top center of the package surface.

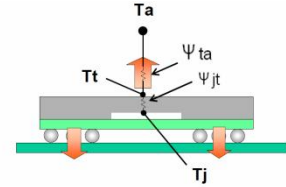
θ_{jc} Thermal resistance, junction-to-case

θ_{jb} Thermal resistance, junction-to-ball

$$T_j = T_a + \theta_{ja} \times P$$

$$T_t = T_j - \Psi_{jt} \times P$$

Where, no use of external heat sinks



The conditions are as follows:

- Chip size 5.44 mm x 5.44 mm
- Natural air cooling
- JEDEC standards conformal
- Thermal resistance of 400 mW around the working point

5.4.1 Max. junction temperature T_j

With a max. ERTEC 200P power dissipation of **1.23 W** (see 4.12) and $T_A = 85^\circ\text{C}$, the max. junction temperature T_j is: $85^\circ\text{C} + (1.23 \text{ W} * 23.5 \text{ K/W}) = \mathbf{114^\circ\text{C}}$

5.5 Solder Profile

Solder this product ERTEC 200P (lead free device) under the following recommended conditions:

Soldering Method	Soldering Condition	Symbol of Recommended Soldering Condition
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (220°C min.), Number of times: 3 max., Number of days: 7 (see Note)	IR60-107-3

Note:

The number of days refers to storage at 25°C, 65% RH MAX after the dry pack has been opened.

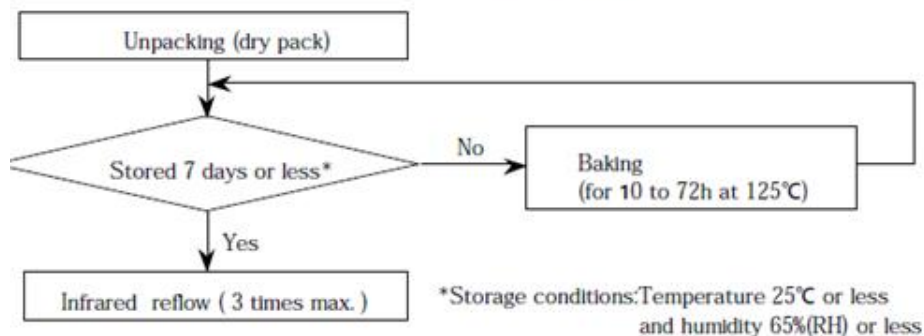
After that, prebaking is necessary at 125 °C for 10 to 72 hours.

RECOMMENDED SOLDERING CONDITIONS OF INFRARED REFLOW [INCLUDING CONVECTION, INFRARED/CONVECTION]

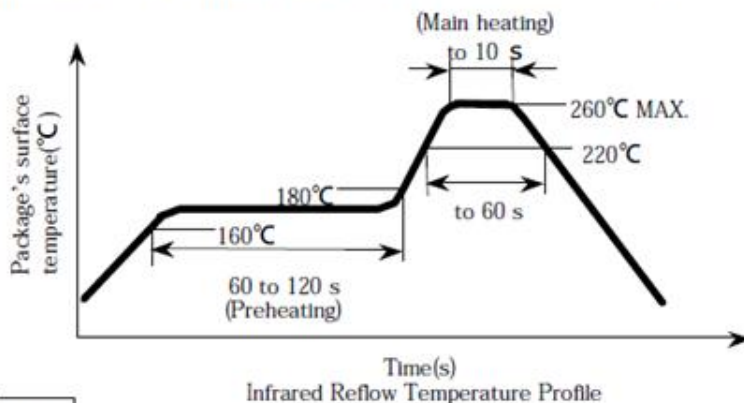
Moisture sensitive device

(IR60-107-3)

The following is recommended soldering conditions of infrared reflow.



Maximum temperature (Package's surface temperature) : 260°C or below
 Time at maximum temperature : 10s or less
 Time of temperature higher than 220°C : 60s or less
 Preheating time at 160 to 180°C : 60 to 120s
 Maximum number of reflow processes : 3 times
 Maximum chlorine content of rosin flux (percentage mass) : 0.2 % or less
 Exposure limit (Store until the final reflow process starts) : 7 days or less



Notice

For baking components, it is necessary to use heatproof type container. Plastic magazines, emboss tape/reels and some of trays are not heatproof type, so if the packing container is not heatproof type, please transfer them to a heatproof type container.

IR60-107-3SSD-A-M2207-1

IR60-107-3 (It may be changed after assembly check.)

And regarding the additional information, please see the following Home Page.

http://www.renesas.com/products/package/manual/4/4_3/index.jsp

5.6 Packing Information

5.6.1 Tray

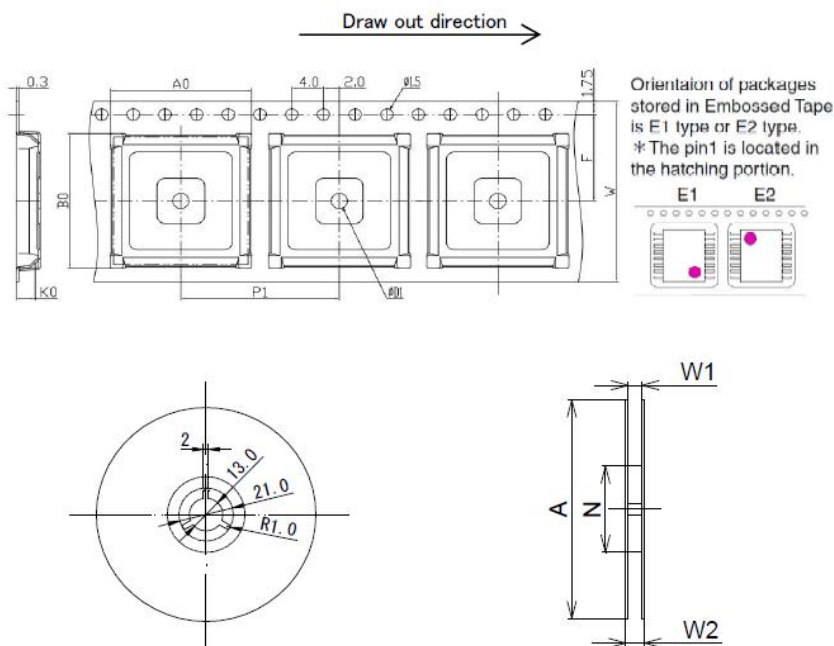
Max. 90 ASICs are packed into one tray.

The dimensions of a tray are: 135,9 x 322,6 x 7,62 mm

Max. 5 trays are packed into an inner box.

The dimensions of an inner box are: 175 x 375 x 75 mm

5.6.2 Tape&Reel



Tape Code		E24 * 20—88CA
Tape Dimensions (mm)	W	24.0
	P1	20.0
	A0	17.9
	B0	17.9
	K0	1.8
	F	11.5
	D1	2.0
Reel Dimensions (mm)	A	330
	N	100
	W1	25.5
	W2	29.5
Maximum storage Num. IC/ Reel		1000
Material		Carbon PS
Surface resistance		Less than $1 \times 10^{11} \Omega / \square$

Information: Orientation (pin 1) is E2 type

6 QUALITY

6.1 Hard-Error FIT rates

(Ea=0.7eV, C.L.=60%, 24h/10yrs)

Ten years operation FIT rate @125°C:

- 315 FIT (ERTEC-Die: 217FIT, PHY-Die: 98 FIT)

6.2 Soft-Error FIT rates

Double bit errors: The adjacent bits in single port RAMs are in different words, therefore they can be corrected by ECC. Based on literature values and simulation results double bit fails have a probability of 1/10th to 1/100th of single bit fails.

Soft error rate for the memories in the ERTEC 200P (alpha & neutron):

	0m	2000m	2300m	2500m	3000m	3500m	4000m	5000m
alpha	158 FIT							
neutron	142 FIT	747 FIT	933 FIT	1077,8 FIT	1525,1 FIT	2121,5 FIT	2901 FIT	5171,6 FIT
factor to neutron 0m ¹⁾	x 1,0	x 5,26	x 6,57	x 7,59	x 10,74	x 14,94	x 20,43	x 36,42
total SER	300 FIT	905 FIT	1091 FIT	1235,8 FIT	1683,1 FIT	2279,5 FIT	3059 FIT	5329,6 FIT

¹⁾ Location: NewYork, USA

Note:

The neutron factor for various difference heights can be calculated at the following WEB link:

<http://www.seutest.com/cgi-bin/FluxCalculator.cgi>

7 MISCELLANEOUS

7.1 Abbreviations

Abbreviation	Description
AHB	AMBA Advanced High-performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APB	AMBA Advanced Peripheral Bus
AR	Application Relationship (e.g. controller <-> device)
CR	Communication Relationship (e.g. input data / output data)
D_TCM	Data Tightly Coupled Memory
DFT	Design For Test
DFP	Dynamic Frame Packing
GDMA	Generic Direct Memory Access
EMC	External Memory Controller
ETB	Embedded Trace Buffer
ETM	Embedded Trace Macrocell
FIQ	Fast Interrupt Request
GPIO	General Purpose Input/Output
HW	Hardware
I_TCM	Instruction Tightly Coupled Memory
ICU	Interrupt Control Unit
IRQ	Interrupt Request
IRT	Isochronous Real Time
IRT High Performance	Isochronous Real Time with fast-forwarding, pack/unpack, and NRT fragmentation
ISR	Interrupt Service Register
IP	Intellectual Property
JTAG	Joint Test Action Group
MC	Motion Control
NMI	Non Maskable Interrupt
NRT	Non-Real Time
PCB	Printed Circuit Board
PCI	Peripheral Communication Interface
PHY	Physical Layer
PLL	Phase Locked Loop
PNPLL	PROFINET PLL
RT	Soft Real Time
SCRB	System Control Register Block

7.2 Literature list

Refer- ence	Title	
/1/	IEEE 802.1D – 1998 (MAC Bridges)	
/2/	IEEE 802.1Q – 1998 (VLANs)	
/3/	IEEE1149.1 (Boundary Scan)	
/4/	ETM9 Technical Reference Manual (Rev. 2a) (ARM DDI 0157E)	
/5/	ETM Specification (ARM IHI 0014 H)	
/6/	ARM9E-S Technical Reference Manual (ARM DDI 0198D)	
/7/	ARM926EJ-S Technical Reference Manual (ARM DDI 0201A)	
/9/	Using Embedded ICE; Appl. Note 31; Issue C	
/15/	DDI0183F_uart_pl011_r1p4_trm.pdf, UART (PL011) Technical Reference Manual	
/16/	IEC 61158-5-10 V2.3 (PNO) and IEC 61158-6-10 V2.3 (PNO)	
/29/	I ² C-Bus Specification, V2.1 of January 2000	
/46/	ETB11 Technical Reference Manual (ARM DDI0275B) of 2002/2003	