

SIEMENS

ERTEC 400

Enhanced Real-Time Ethernet Controller

Manual

Disclaimer of Liability

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.

Copyright

© Siemens AG 2006. 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.

PROCENTEC

Turfschipper 41
2292 JC Wateringen
The Netherlands
tel. +31 (0)174 671 800
fax. +31 (0)174 671 801
e-mail info@procentec.com

Products

ASICs

LSPM2 RoHS (6-pack)	EUR 95,- per pack
LSPM2 RoHS (66-pack)	EUR 919,- per pack
SPC3 Step-C RoHS (6-pack)	EUR 75,- per pack
SPC3 Step-C RoHS (96-pack)	EUR 1052,- per pack
SPC3 Step-C Low Voltage RoHS (5-pack)	EUR 85,- per pack
SPC3 Step-C Low Voltage RoHS (160pack)	EUR 1600,- per pack
SPC4-2 RoHS (5-pack)	EUR 92,- per pack
SPC4-2 RoHS (160-pack)	EUR 2288,- per pack
DPC31 Step-B RoHS (6-pack)	EUR 95,- per pack
DPC31 Step-B RoHS (60-pack)	EUR 877,- per pack
DPC31 Step-C1 RoHS (6-pack)	EUR 95,- per pack
DPC31 Step-C1 RoHS (60-pack)	EUR 868,- per pack
SIM1-1 RoHS (160-pack)	EUR 4890,- per pack
SIM1-2 RoHS (60-pack)	EUR 900,- per pack
ERTEC 200 (5-pack)	EUR 120,- per pack
ERTEC 200 (70-pack)	EUR 950,- per pack
ERTEC 400 (5-pack)	EUR 250,- per pack
ERTEC 400 (70-pack)	EUR 2300,- per pack

Please contact us for larger quantities or check out: www.procentec.com

PROCENTEC

Turfschipper 41
2292 JC Wateringen
The Netherlands
tel. +31 (0)174 671 800
fax. +31 (0)174 671 801
e-mail info@procentec.com

Firmware and development kits

SPC3 ANSI-C software	EUR 680,-
DPC31 Development kit	EUR 2470,-
ERTEC 200 Development Kit	EUR 4500,-
ERTEC 200 Productline license	EUR 4900,-
ERTEC 400 Development Kit	EUR 4500,-
ERTEC 400 Productline license	EUR 4900,-

Tools

ProfiScript - Certification software	EUR 800,- (HW excl.) www.procentec.com/profiscrypt
ProfiTrace - Analyzer on USB	EUR 1400,- www.procentec.com/profitrace2
ProfiCaptain - DP Master	EUR 500,- (HW excl.) www.procentec.com/proficaptain

Training

PROFIBUS Product Development	EUR 995,- www.procentec.com/ppd
Certified PROFIBUS Engineer	EUR 1850,- www.procentec.com/cpe
Certified PROFIBUS Installer	EUR 750,- www.procentec.com/cpi

Preface

Target Audience of this Manual

This manual is intended for hardware developers who want to use the ERTEC 400 for new products. Experience working with processors and designing embedded systems and knowledge of Ethernet are required for this. It describes all ERTEC 400 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

- Section 1 Overview of the architecture and the individual function groups of the ERTEC 400.
- Section 2 ARM946E-S processor systems.
- Section 3 Bus system of the ERTEC 400.
- Section 4 I/O of the ERTEC 400.
- Section 5 General hardware functions.
- Section 6 External memory interface (EMIF).
- Section 7 Local bus unit (LBU).
- Section 8 PCI Interface.
- Section 9 Memory partitioning of the ERTEC 400.
- Section 10 HW tools for test, trace, and debugging.
- Section 11 List of terms and references

Scope of the Manual

This manual applies to the following product:

ERTEC 400 Version 01 and higher

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:

- 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.
- A glossary containing definitions of important terms used in the manual is located following the appendices.
- 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

Automation & Drives

ComDeC

Street address:

Würzburgerstr.121

90766 Fürth

Federal Republic of Germany

Phone: 0911/750-2736

Phone: 0911/750-2080

Fax: 0911/750-2100

E-mail: ComDeC@siemens.com

Mailing address:

P.O. Box 2355

90713 Fürth

Federal Republic of Germany

Technical Contacts for USA

PROFI Interface Center:

One Internet Plaza

PO Box 4991

Johnson City, TN 37602-4991

Fax: (423)- 262- 2103

Phone: (423)- 262- 2576

E-mail: profibus.sea@siemens.com

Contents

1	Introduction	9
1.1	Applications of the ERTEC 400.....	9
1.2	Features of the ERTEC 400.....	9
1.3	Structure of the ERTEC 400.....	10
1.4	ERTEC 400 Package	11
1.5	Signal Function Description.....	12
1.5.1	GPIO 0 to 31 and Alternative Functions.....	12
1.5.2	JTAG and Debug	13
1.5.3	Trace Port.....	13
1.5.4	Clock and Reset	13
1.5.5	TEST Pins.....	13
1.5.6	EMIF, Boot/Config	14
1.5.7	PCI/LBU	16
1.5.8	RMII/MII	18
1.5.9	Power Supply.....	19
2	ARM946E-S Processors	21
2.1	Structure of ARM946E-S.....	21
2.2	Description of ARM946E-S	22
2.3	Operating Frequency of ARM946E-S.....	22
2.4	Cache Structure of ARM946E-S.....	22
2.5	Tightly Coupled Memory (TCM)	22
2.6	Memory Protection Unit (MPU)	23
2.7	Bus Interface of ARM946E-S	23
2.8	ARM946E-S Embedded Trace Macrocell (ETM9).....	23
2.9	ARM Interrupt Controller (ICU).....	23
2.9.1	Prioritization of Interrupts	23
2.9.2	Trigger Modes.....	24
2.9.3	Masking the Interrupt Inputs	24
2.9.4	Software Interrupts for IRQ.....	24
2.9.5	Nested Interrupt Structure.....	24
2.9.6	EOI End-Of-Interrupt.....	24
2.9.7	IRQ Interrupt Sources	25
2.9.8	FIQ Interrupt Sources	25
2.9.9	IRQ Interrupts as FIQ Interrupt Sources	25
2.9.10	Interrupt Control Register.....	26
2.9.11	ICU Register Description	27
2.10	ARM946E-S Register	31
3	Bus System of the ERTEC 400.....	32
3.1	“Multilayer AHB” Communication Bus	32
3.1.1	AHB Arbiter.....	32
3.1.2	AHB Master-Slave Coupling	32
3.2	APB I/O Bus	33
4	I/O on APB bus	33
4.1	BOOT ROM.....	33
4.1.1	Booting from External ROM	34
4.1.2	Booting via SPI	34
4.1.3	Booting via UART1	34
4.1.4	Booting via PCI or LBU.....	34
4.2	General Purpose I/O (GPIO).....	35
4.2.1	Address Assignment of GPIO Registers.....	35
4.2.2	GPIO Register Description.....	36
4.3	Timer 0 and Timer 1	37
4.3.1	Mode of Operation of Timers	38
4.3.2	Timer Interrupts	38
4.3.3	Timer Prescaler	38
4.3.4	Cascading of Timers	38
4.3.5	Address Assignment of Timer 0/1 Registers.....	38
4.3.6	Timer 0/1 Register Description	39
4.4	F - Counter	41
4.4.1	Address Assignment of F-Timer Registers	42

4.4.2	F-Timer Register Description	42
4.5	Watchdog Timers	43
4.5.1	Watchdog Timer 0.....	43
4.5.2	Watchdog Timer 1.....	43
4.5.3	Watchdog Interrupt	43
4.5.4	WDOUT0_N.....	43
4.5.5	WDOUT1_N.....	43
4.5.6	Watchdog Registers.....	44
4.5.7	Address Assignment of Watchdog Registers.....	44
4.5.8	Watchdog Register Description	44
4.6	UART1/ UART2.....	46
4.6.1	Address Assignment of UART 1/2 Registers	48
4.6.2	UART 1/2 Register Description.....	48
4.7	Synchronous Interface SPI.....	52
4.7.1	Address Assignment of SPI Register	53
4.7.2	SPI Register Description.....	54
4.8	System Control Register	56
4.8.1	Address Assignment of System Control Registers	56
4.8.2	System Control Register Description	57
5	General Hardware Functions	62
5.1	Clock Generation and Clock Supply.....	62
5.1.1	Clock Supply in ERTEC 400	62
5.1.2	PCI Clock Supply	63
5.1.3	LBU Clock Supply.....	64
5.1.4	JTAG Clock Supply.....	64
5.1.5	Ethernet Interface Clock Supply	64
5.2	Reset Logic of the ERTEC 400	64
5.2.1	Hardware Reset	65
5.2.2	Watchdog Reset	65
5.2.3	Software reset.....	65
5.2.4	PCI bridge reset.....	65
5.2.5	Actions when HW Reset is Active	66
5.3	Address Space and Timeout Monitoring	67
5.3.1	AHB Bus Monitoring.....	67
5.3.2	APB Bus Monitoring.....	67
5.3.3	EMIF Monitoring	67
5.3.4	PCI Slave Monitoring	67
6	External Memory Interface (EMIF)	69
6.1	Address Assignment of EMIF Registers.....	70
6.2	EMIF Register Description	70
7	Local Bus Unit (LBU)	74
7.1	Page Range Setting	75
7.2	Page Offset Setting	75
7.3	Page Control Setting	76
7.3.1	LBU Read from ERTEC 400 with separate Read/Write line (LBU_RDY_N active low)	77
7.3.2	LBU Write to ERTEC 400 with separate Read/Write line (LBU_RDY_N active low).....	78
7.3.3	LBU Read from ERTEC 400 with common Read/Write line (LBU_RDY_N active low)	79
7.3.4	LBU Write to ERTEC 400 with common Read/Write line (LBU_RDY_N active low).....	80
7.4	Address Assignment of LBU Registers	81
7.5	LBU Register Description	82
8	PCI Interface	83
8.1	PCI Functionality	83
8.1.1	General Functions of the PCI Interface:.....	83
8.1.2	PCI Master Interface:.....	83
8.1.3	PCI Target Interface:.....	84
8.1.4	PCI Interrupt Handling:	84
8.1.5	PCI Power Management:.....	85
8.1.6	Accesses to the AHB Bus:.....	85
8.2	ERTEC 400 Applications with PCI:.....	85
8.2.1	ERTEC 400 in a PC System	85
8.2.2	ERTEC 400 as a Station on the Local PCI Bus	86
8.3	Address Assignment of PCI Register	87
8.4	PCI Register Description	88

9	Memory Description	89
9.1	Memory Partitioning of the ERTEC 400	89
9.2	Detailed Memory Description	90
10	Test and Debugging	92
10.1	ETM9 Embedded Trace Macrocell	92
10.1.1	Trace Modes	92
10.1.2	Features of the ETM9 Module	92
10.1.3	ETM9 Registers	92
10.2	Trace Interface	93
10.3	JTAG Interface	93
10.4	Debugging via UART1.....	93
11	Miscellaneous	94
11.1	Acronyms/Glossary:	94
11.2	References:	95

List of Figures

Figure 1: ERTEC 400 Block Diagram	10
Figure 2: ERTEC 400 Package Description	11
Figure 3: Structure of ARM946E-S Processor System	21
Figure 4: GPIO Cells of ERTEC 400	35
Figure 5: Block Diagram of F-Counter	41
Figure 6: Watchdog Timing.....	43
Figure 7: Block Diagram of UART	46
Figure 8: Block Diagram of SPI	52
Figure 9: Detailed Representation of Clock Unit.....	63
Figure 10: Clock Supply of Ethernet Interface	64
Figure 11: Power-Up Phase of the PLL	65
Figure 12: LBU-Read-Sequence with separate RD/WR line	77
Figure 13: LBU-Write-Sequence with separate RD/WR line.....	78
Figure 14: LBU-Read-Sequence with common RD/WR line.....	79
Figure 15: LBU Write Sequence with common RD/WR line	80
Figure 16: PCI Interrupt Handling	84

List of Tables

Table 1: ERTEC 400 Pin Assignment and Signal Description.....	19
Table 2: Overview of IRQ Interrupts	25
Table 3: Overview of FIQ Interrupts.....	25
Table 4: Overview of Interrupt Control Register.....	27
Table 5: CP15 Registers - Overview	31
Table 6: Overview of AHB Master-Slave Access.....	32
Table 7: Access Type and Data Bit Width of I/O.....	33
Table 8: Selection of Download Source.....	34
Table 9: Overview of GPIO Registers.....	35
Table 10: Overview of Timer Registers	38
Table 11: Overview of F-Timer Registers	42
Table 12: Overview of WD Registers.....	44
Table 13: Baud Rates for UART at $F_{UARTCLK}=50$ MHz	47
Table 14: Overview of UART 1/2 Registers.....	48
Table 15: Overview of SPI Registers.....	53
Table 16: Overview of System Control Registers	57
Table 17: Overview of ERTEC 400 Clocks	63
Table 18: Overview of EMIF Registers	70
Table 19: Setting of Various Page Sizes	75
Table 20: Setting of Various Offset Areas	75
Table 21: Overview of Accesses to Address Areas of ERTEC 400.....	76
Table 22: LBU read access timing with separate Read/Write line	77
Table 23: LBU write access timing with separate Read/Write line	78
Table 24: LBU read access timing with common Read/Write line	79
Table 25: LBU write access timing with common Read/Write line	80
Table 26: Overview of LBU Registers.....	81
Table 27: Overview of PCI Registers.....	88
Table 28: Partitioning of Memory Areas	89
Table 29: Detailed Description of Memory Segments.....	90
Table 30: Pin Assignment of JTAG Interface.....	93

1 Introduction

The ERTEC 400 is intended for the implementation of PROFINET devices with RT and IRT functionality. By virtue of its integrated **ARM946 processor**, integrated **4-port realtime Ethernet switch**, and available options for connecting external host processor systems to a **selectable bus system (PCI or LBU)**, the ERTEC 400 meets all of the requirements for implementing PROFINET devices with integrated switch functionality.

1.1 Applications of the ERTEC 400

- Interface module for high-accuracy closed-loop drive control, even for PC-based systems
- Distributed I/O with real-time Ethernet interfacing
- PROFINET RT and IRT functionality

1.2 Features of the ERTEC 400

The ERTEC 400 is a high-performance Ethernet controller with integrated function groups:

- High-performance ARM 946 processor with D-cache, I-cache, D-TCM memory
- Multilayer AHB bus master/slave with AHB arbiter
- 4-port IRT switch with 192 Kbytes of communication RAM
- PCI interface for connection to a PC system (bootable)
- Local bus unit (LBU) for connection to an external host processor (bootable)
- SDRAM controller
- SRAM controller
- 32 assignable I/O
- 2 x UART (UART 1 bootable)
- SPI (bootable)
- 2 x timer
- F-timer
- Watchdog
- IRQ and FIQ interrupt controllers
- PLL with clock generator
- 8 Kbytes of SRAM
- 8 Kbytes of BOOT ROM
- 304-pin FBGA housing
- Different test functions
- JTAG debug and trace interface

1.3 Structure of the ERTEC 400

The figure below shows the function groups with the common communication paths.

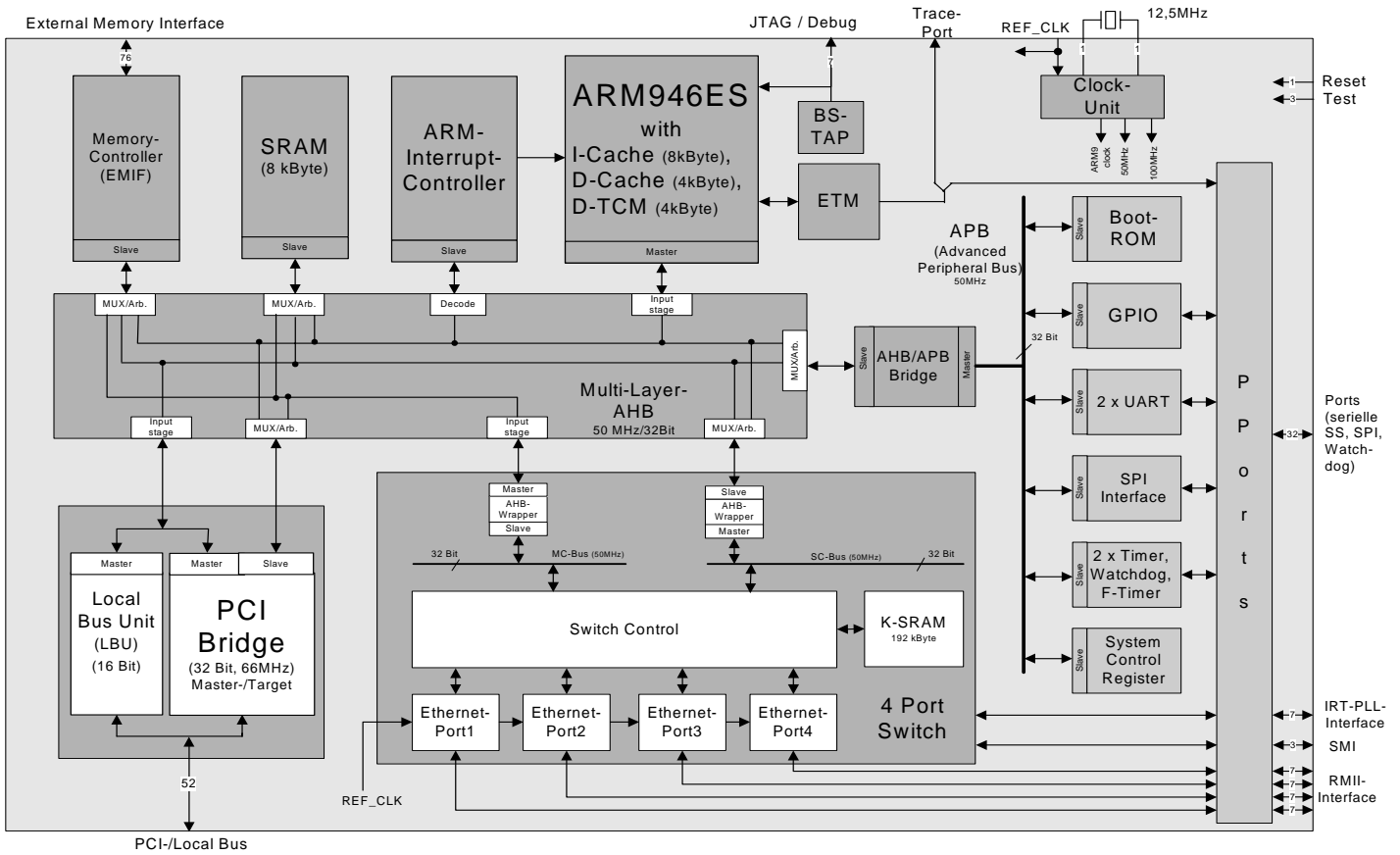


Figure 1: ERTEC 400 Block Diagram

1.4 ERTEC 400 Package

The ERTEC 400 is supplied in an FBGA package with 304 pins. The distance between the pins is 0.8 mm. The package dimensions are 19 mm x 19 mm.

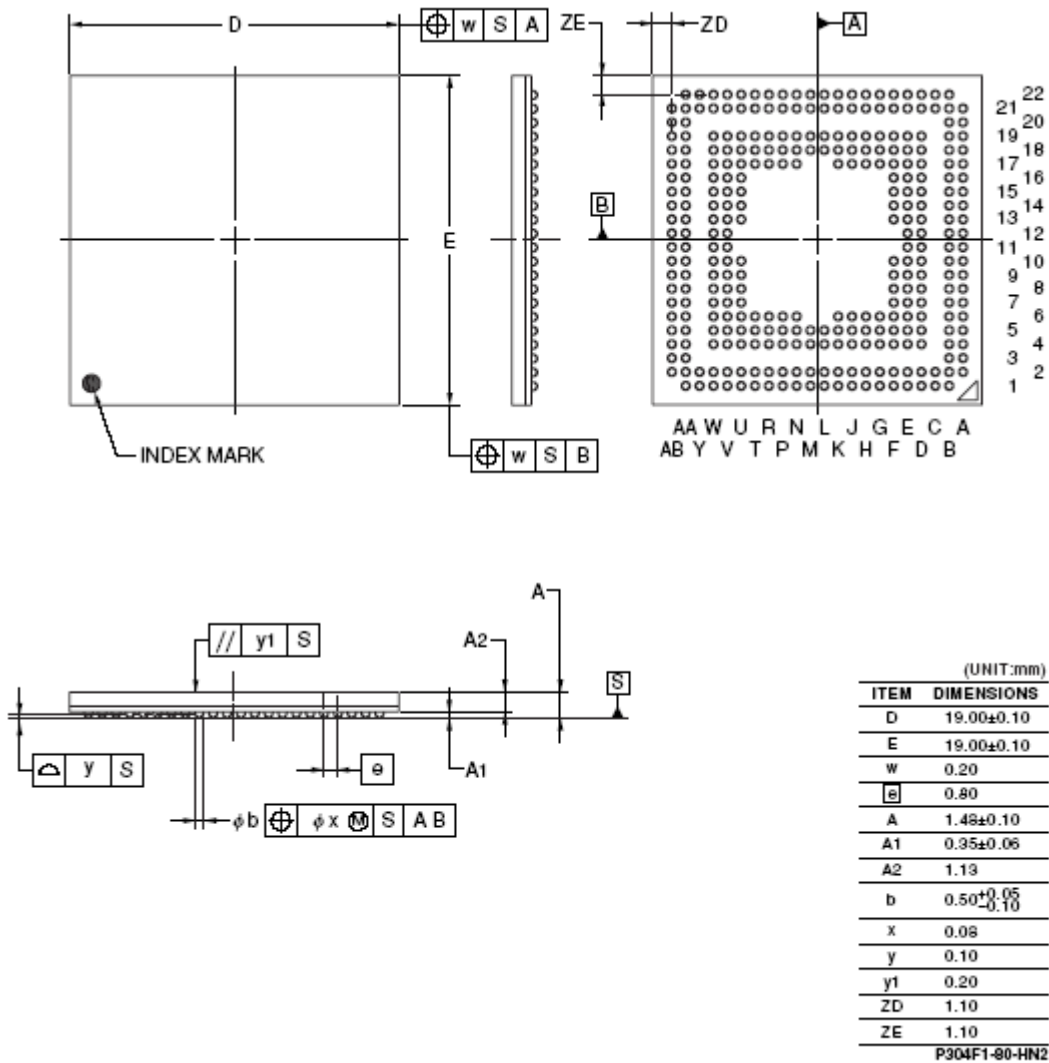


Figure 2: ERTEC 400 Package Description

The following documents contain the soldering instructions for the ERTEC 400:

- /10/ Soldering instructions for lead-based block.
- /11/ Soldering instructions for lead-free block.
- /12/ Code description for soldering.

When working with modules, **always take precautionary measures** against electrostatic charge (**ESD – Electrostatic Sensitive Devices**).

1.5 Signal Function Description

Pin Description for ERTEC 400

The ERTEC 400 Ethernet communication block is available in a 304-pin FBGA package: The signal names of the ERTEC 400 are described in this section.

1.5.1 GPIO 0 to 31 and Alternative Functions

Various signals are multiplexed on the same pin. These multiplexed signals can contain up to four different functions. The alternative functions are assigned in GPIO registers **GPIO_PORT_MODE_L** and **GPIO_PORT_MODE_H** (see Section 4.2.2). The table describes all signals with their different functions and associated pin numbers.

No.	Signal Name	Alternative Function 1	Alternative Function 2	Alternative Function 3	(5) I/O (Reset)	Pull-	Pin No.	Comment
General Purpose I/O / I/O								
1	GPIO0				B (I)	up	T21	GPIO (interruptible)
2	GPIO1				B (I)	up	U21	GPIO (interruptible)
3	GPIO2				B (I)	up	V22	GPIO
4	GPIO3				B (I)	up	V21	GPIO
5	GPIO4				B (I)	up	N17	GPIO
6	GPIO5				B (I)	up	P19	GPIO
7	GPIO6				B (I)	up	R19	GPIO
8	GPIO7				B (I)	up	T19	GPIO
9	GPIO8	TXD1		TRACEPKT0	B/O/-O (I)	up	AA20	GPIO or UART1 or ETM
10	GPIO9	RXD1		TRACEPKT1	B/I/-O (I)	up	AB20	GPIO or UART1 or ETM
11	GPIO10	DCD1_N		TRACEPKT2	B/I/-O (I)	up	AA19	GPIO or UART1 or ETM
12	GPIO11	DSR1_N		TRACEPKT3	B/I/-O (I)	up	AA18	GPIO or UART1 or ETM
13	GPIO12	CTS1_N		ETMEXTOUT	B/I/-O (I)	up	W13	GPIO or UART1 or ETM
14	GPIO13	TXD2			B/O (I)	up	V15	GPIO or UART2
15	GPIO14	RXD2			B/I (I)	up	U15	GPIO or UART2
16	GPIO15	DCD2_N	WDOUT0_N		B/I/O (I)	up	W16	GPIO or UART2 or watchdog
17	GPIO16	DSR2_N	SSPCTLOE	ETMEXTIN1	B/I/O/I (I)	up	AB14	GPIO or UART2 or SPI or ETM
18	GPIO17	CTS2_N	SSPOE	Reserved	B/I/O/O (I)	up	AA14	GPIO or UART2 or SPI
19	GPIO18	SSPRXD			B/I (I)	up	W22	GPIO or SPI
20	GPIO19	SSPTXD		TRACEPKT4	B/O/-O (I)	up	AB18	GPIO or SP1 or ETM
21	GPIO20	SCLKOUT		TRACEPKT5	B/O/-O (I)	up	AA17	GPIO or SP1 or ETM
22	GPIO21	SFRMOUT		TRACEPKT6	B/O/-O (I)	up	AB17	GPIO or SP1 or ETM
23	GPIO22	SFRMIN		TRACEPKT7	B/I/-O (I)	up	AA16	GPIO or SPI or ETM; if SPI boot → GPIO = Chip Select
24	GPIO23	SCLKIN		DBGACK	B/I/-O (I)	up	Y22	GPIO or SPI (I) or ARM9 debugging (O)
25	GPIO24	PLL_EXT_IN_N			B/I (I)	up	AA11	GPIO or MC_PLL
26	GPIO25	TGEN_OUT1_N *1			B/O (I)	up	AA10	GPIO or MC_PLL
27	GPIO26	TGEN_OUT2_N			B/O (I)	up	AB10	GPIO or MC_PLL
28	GPIO27	TGEN_OUT3_N			B/O (I)	up	W10	GPIO or MC_PLL
29	GPIO28	TGEN_OUT4_N			B/O (I)	up	W9	GPIO or MC_PLL
30	GPIO29	TGEN_OUT5_N			B/O (I)	up	V10	GPIO or MC_PLL
31	GPIO30	TGEN_OUT6_N			B/O (I)	up	V11	GPIO (interruptible) or MC_PLL
32	GPIO31				B (I)	up	V12	GPIO (interruptible)

*1 For an IRT application pin GPIO25 is default parameterized as alternate function1 (TGEN_OUT1_N). A synchronous clock is issued at this pin. During the certification process of a PROFINET IO DEVICE with IRT functionality this pin has to be accessible from outside (mandatory).

Different GPIO's are used on the Evaluation Board EB400. See Dokument /13/ Table 9.

1.5.2 JTAG and Debug

No.	Signal Name	I/O (Reset)	Pull-	Pin No.	Comment
Debug / JTAG (BOUNDARY SCAN)					
33	TRST_N	I (I)		V19	JTAG Reset
34	TCK	I (I)	up	W19	JTAG Clock
35	TDI	I (I)	up	W18	JTAG Data In
36	TMS	I (I)	up	U19	JTAG Test Mode Select
37	TDO	O (O)		V16	JTAG Data Out
38	DBGREQ	I (I)	dn	W21	ARM9 Debugging (PD)
39	TAP_SEL	I (I)	up	AB12	Select TAP Controller: 0: Boundary Scan TAP Controller selected 1: ARM-TAP Controller selected or Scan Clock (Scan mode)

1.5.3 Trace Port

No.	Signal Name	I/O (Reset)	Pull-	Pin No.	Comment
Trace Port (Basic)					
40	PIPESTA0	O (O)		Y21	Trace Pipeline Status(0) or TEST_N=0: Test input
41	PIPESTA1	O (O)		AA22	Trace Pipeline Status(1) or TEST_N=0: Test input
42	PIPESTA2	O (O)		AA21	Trace Pipeline Status(2) or TEST_N=0: Test input
43	TRACESYNC	O (O)		AB21	Trace Sync signal

1.5.4 Clock and Reset

No.	Signal Name	I/O (Reset)	Pull-	Pin No.	Comment
CLOCK / RESET GENERATION					
44	TRACECLK	O (O)		AB16	ETM Trace Clock or Scan Clock (Scan mode)
45	CLKP_A	Osc - I (I)		AA13	Quartz connection
46	CLKP_B	Osc - O (O)		W12	Quartz connection
47	F_CLK	I (I)		AA12	F_CLK for F-counter
48	REF_CLK	I (I)		R22	Reference clock for RMII
49	RESET_N	I (I)	up	AA15	HW reset

1.5.5 TEST Pins

No.	Signal Name	I/O (Reset)	Pull-	Pin No.	Comment
TEST					
50	TEST_N (3)	I	up	P17	Test mode
51	TMC1 (3)	I		R17	Test configuration
52	TMC2 (3)	I		T18	Test configuration

1.5.6 EMIF, Boot/Config

No.	Signal Name	Alternative Reset Function	(5) I/O (Reset)	Pull-	Pin No.	Comment
EMIF (External Memory Interface)						
53	DTR_N		O (O)		U8	Direction signal for external driver or scan clock (Scan mode)
54	OE_DRIVER_N		O (O)		V8	Enable signal for external driver or scan clock (Scan mode)
55	A0		O (O)		B1	Address bit 0 SDRAM: Bank address 0
56	A1		O (O)		C2	Address bit 1 SDRAM: Bank address 1
57	A2		O (O)		C1	Address bit 2 SDRAM: Address 0
58	A3		O (O)		D2	Address bit 3 SDRAM: Address 1
59	A4		O (O)		D1	Address bit 4 SDRAM: Address 2
60	A5		O (O)		E2	Address bit 5 SDRAM: Address 3
61	A6		O (O)		F2	Address bit 6 SDRAM: Address 4
62	A7		O (O)		F1	Address bit 7 SDRAM: Address 5
63	A8		O (O)		G2	Address bit 8 SDRAM: Address 6
64	A9		O (O)		G1	Address bit 9 SDRAM: Address 7
65	A10		O (O)		H2	Address bit 10 SDRAM: Address 8
66	A11		O (O)		H1	Address bit 11 SDRAM: Address 9
67	A12		O (O)		J2	Address bit 12 SDRAM: Address 10
68	A13		O (O)		J1	Address bit 13 SDRAM: Address 11
69	A14		O (O)		E4	Address bit 14 SDRAM: Address 12
70	A15		O (O)		F5	Address bit 15
71	A16	BOOT0 (1)	O (I)		F4	Address bit 16 / ERTEC 400 – boot mode (ext. PU/PD necessary)
72	A17	BOOT1 (1)	O (I)		G4	Address bit 17 / ERTEC400 – boot mode (ext. PU/PD necessary)
73	A18	BOOT2 (1)	O (I)		J4	Address bit 18 / ERTEC400 – boot mode (ext. PU/PD necessary)
74	A19	CONFIG0 (2)	O (I)		H6	Address bit 19 / ERTEC400 – system configuration (ext. PU/PD necessary)
75	A20	CONFIG1 (2)	O (I)		H5	Address bit 20 / ERTEC400 – system configuration (ext. PU/PD necessary)
76	A21	CONFIG2 (2)	O (I)		J6	Address bit 21 / ERTEC400 – system configuration (ext. PU/PD necessary)
77	A22	CONFIG3 (2)	O (I)		J5	Address bit 22 / ERTEC400 – system configuration (ext. PU/PD necessary)
78	A23	CONFIG4 (2)	O (I)		K5	Address bit 23 / ERTEC400 – system configuration (ext. PU/PD necessary)

No.	Signal Name	Alternative Reset Function	(5) I/O (Reset)	Pull-	Pin No.	Comment
EMIF (External Memory Interface)						
79	D0		B (I)	up	N2	Data bit 0
80	D1		B (I)	up	P1	Data bit 1
81	D2		B (I)	up	P2	Data bit 2
82	D3		B (I)	up	R2	Data bit 3
83	D4		B (I)	up	T1	Data bit 4
84	D5		B (I)	up	T2	Data bit 5
85	D6		B (I)	up	U2	Data bit 6
86	D7		B (I)	up	V2	Data bit 7
87	D8		B (I)	up	W1	Data bit 8
88	D9		B (I)	up	W2	Data bit 9
89	D10		B (I)	up	Y2	Data bit 10
90	D11		B (I)	up	AA1	Data bit 11
91	D12		B (I)	up	W4	Data bit 12
92	D13		B (I)	up	AA2	Data bit 13
93	D14		B (I)	up	AB2	Data bit 14
94	D15		B (I)	up	AA3	Data bit 15
95	D16		B (I)	up	N6	Data bit 16
96	D17		B (I)	up	M5	Data bit 17
97	D18		B (I)	up	N4	Data bit 18
98	D19		B (I)	up	P5	Data bit 19
99	D20		B (I)	up	R6	Data bit 20
100	D21		B (I)	up	R4	Data bit 21
101	D22		B (I)	up	R5	Data bit 22
102	D23		B (I)	up	T5	Data bit 23
103	D24		B (I)	up	V4	Data bit 24
104	D25		B (I)	up	W5	Data bit 25
105	D26		B (I)	up	AA4	Data bit 26
106	D27		B (I)	up	AB4	Data bit 27
107	D28		B (I)	up	AA5	Data bit 28
108	D29		B (I)	up	AB5	Data bit 29
109	D30		B (I)	up	AA6	Data bit 30
110	D31		B (I)	up	AB6	Data bit 31
111	WR_N		O (O)		AA9	Write strobe
112	RD_N		O (O)		AB9	Read strobe
113	CS_PER0_N		O (O)		AB8	Chip select bank 1 (ROM); boot area
114	CS_PER1_N		O (O)		AA7	Chip select bank 2
115	CS_PER2_N		O (O)		W6	Chip select bank 3
116	CS_PER3_N		O (O)		V6	Chip select bank 4
117	BE0_DQM0_N		O (O)		N1	Byte enable 0 for D(7:0)
118	BE1_DQM1_N		O (O)		V1	Byte enable 1 for D(15:8)
119	BE2_DQM2_N		O (O)		L5	Byte enable 2 for D(23:16)
120	BE3_DQM3_N		O (O)		T4	Byte enable 3 for D(31:24)
121	RDY_PER_N		I (I)	up	AA8	Ready signal
122	CLK_SDRAM		O (O)		L2	Clock SDRAM
123	CS_SDRAM_N		O (O)		K2	Chip select SDRAM
124	RAS_SDRAM_N		O (O)		K1	RAS line SDRAM
125	CAS_SDRAM_N		O (O)		L4	CAS line SDRAM
126	WE_SDRAM_N		O (O)		M2	RD/WR SDRAM

1.5.7 PCI/LBU

The PCI or LBU interface is selected by setting the "Config 2" configuration pin during the reset phase.

No	Signal Name PCI	Signal Name LBU	(5) I/O (Reset)	Pull-	Pin No.	Comment
PCI/LBU Interface						
127	AD00	LBU_DB00	B/B (I)		D4	<u>PCI: Address / Data Bit 0</u> <u>LBU: Data Bit 0</u>
128	AD01	LBU_DB01	B/B (I)		A2	<u>PCI: Address / Data Bit 1</u> <u>LBU: Data Bit 1</u>
129	AD02	LBU_DB02	B/B (I)		B2	<u>PCI: Address / Data Bit 2</u> <u>LBU: Data Bit 2</u>
130	AD03	LBU_DB03	B/B (I)		B3	<u>PCI: Address / Data Bit 3</u> <u>LBU: Data Bit 3</u>
131	AD04	LBU_DB04	B/B (I)		D5	<u>PCI: Address / Data Bit 4</u> <u>LBU: Data Bit 4</u>
132	AD05	LBU_DB05	B/B (I)		B4	<u>PCI: Address / Data Bit 5</u> <u>LBU: Data Bit 5</u>
133	AD06	LBU_DB06	B/B (I)		E7	<u>PCI: Address / Data Bit 6</u> <u>LBU: Data Bit 6</u>
134	AD07	LBU_DB07	B/B (I)		A4	<u>PCI: Address / Data Bit 7</u> <u>LBU: Data Bit 7</u>
135	CBE0_N	LBU_BE0_N	B/I (I)		D7	<u>PCI: Byte 0 Enable</u> <u>LBU: Byte 0 Enable</u>
136	AD08	LBU_DB08	B/B (I)		B5	<u>PCI: Address / Data Bit 8</u> <u>LBU: Data Bit 8</u>
137	AD09	LBU_DB09	B/B (I)		D8	<u>PCI: Address / Data Bit 9</u> <u>LBU: Data Bit 9</u>
138	AD10	LBU_DB10	B/B (I)		B6	<u>PCI: Address / Data Bit 10</u> <u>LBU: Data Bit 10</u>
139	AD11	LBU_DB11	B/B (I)		E8	<u>PCI: Address / Data Bit 11</u> <u>LBU: Data Bit 11</u>
140	AD12	LBU_DB12	B/B (I)		A6	<u>PCI: Address / Data Bit 12</u> <u>LBU: Data Bit 12</u>
141	AD13	LBU_DB13	B/B (I)		E9	<u>PCI: Address / Data Bit 13</u> <u>LBU: Data Bit 13</u>
142	AD14	LBU_DB14	B/B (I)		B7	<u>PCI: Address / Data Bit 14</u> <u>LBU: Data Bit 14</u>
143	AD15	LBU_DB15	B/B (I)		F9	<u>PCI: Address / Data Bit 15</u> <u>LBU: Data Bit 15</u>
144	CBE1_N	LBU_BE1_N	B/I (I)		B8	<u>PCI: Byte 1 Enable</u> <u>LBU: Byte 1 Enable</u>
145	PAR	LBU_WR_N	B/I (I)		E10	<u>LBU mode: LBU CFG=0: Write control (Low active)</u> <u>LBU CFG=1: RD/WR Control (0: WR: 1: RD)</u>
146	SERR_N	LBU_POL_RDY	B/I (I)		B9	<u>PCI mode: Bidirection, open drain ; ext. PU necessary;</u> <u>LBU mode: Setting of polarity for pin: LBU RDY N:</u> <u>0: LBU RDY N Low active</u> <u>1: LBU RDY N High active</u> <u>Input; static (input value must not be changed after power-up).</u>
147	PERR_N	LBU_RD_N	B/I (I)		A9	<u>LBU mode:</u> <u>LBU CFG=0: Read Control (Low active)</u> <u>LBU CFG=1: No function</u>
148	STOP_N	LBU_AB00	B/I (I)		D10	<u>LBU: Address Bit 0</u>
149	DEVSEL_N	LBU_AB01	B/I (I)		B10	<u>LBU: Address Bit 1</u>
150	TRDY_N	LBU_AB02	B/I (I)		D11	<u>LBU: Address Bit 2</u>

No	Signal Name PCI	Signal Name LBU	(5) I/O (Reset)	Pull-	Pin No.	Comment
PCI/LBU Interface						
151	IRDY_N	LBU_AB03	B/I (I)		A10	<u>LBU</u> : Address Bit 3
152	FRAME_N	LBU_AB04	B/I (I)		A12	<u>LBU</u> : Address Bit 4
153	CBE2_N	LBU_AB05	B/I (I)		A13	<u>PCI</u> : Byte 2 Enable <u>LBU</u> : Address Bit 5
154	AD16	LBU_AB06	B/I (I)		B12	<u>PCI</u> : Address / Data Bit 16 <u>LBU</u> : Address Bit 6
155	AD17	LBU_AB07	B/I (I)		B13	<u>PCI</u> : Address / Data Bit 17 <u>LBU</u> : Address Bit 7
156	AD18	LBU_AB08	B/I (I)		E11	<u>PCI</u> : Address / Data Bit 18 <u>LBU</u> : Address Bit 8
157	AD19	LBU_AB09	B/I (I)		A14	<u>PCI</u> : Address / Data Bit 19 <u>LBU</u> : Address Bit 9
158	AD20	LBU_AB10	B/I (I)		E12	<u>PCI</u> : Address / Data Bit 20 <u>LBU</u> : Address Bit 10
159	AD21	LBU_AB11	B/I (I)		B14	<u>PCI</u> : Address / Data Bit 21 <u>LBU</u> : Address Bit 11
160	AD22	LBU_AB12	B/I (I)		E13	<u>PCI</u> : Address / Data Bit 22 <u>LBU</u> : Address Bit 12
161	AD23	LBU_AB13	B/I (I)		B15	<u>PCI</u> : Address / Data Bit 23 <u>LBU</u> : Address Bit 13
162	IDSEL	LBU_AB14	I/I (I)		D13	<u>PCI</u> : IDSEL <u>LBU</u> : Address Bit 14
163	CBE3_N	LBU_AB15	B/I (I)		A16	<u>PCI</u> : Byte 3 Enable <u>LBU</u> : Address Bit 15
164	AD24	LBU_AB16	B/I (I)		F14	<u>PCI</u> : Address / Data Bit 24 <u>LBU</u> : Address Bit 16
165	AD25	LBU_AB17	B/I (I)		B16	<u>PCI</u> : Address / Data Bit 25 <u>LBU</u> : Address Bit 17
166	AD26	LBU_AB18	B/I (I)		E14	<u>PCI</u> : Address / Data Bit 26 <u>LBU</u> : Address Bit 18
167	AD27	LBU_AB19	B/I (I)		A17	<u>PCI</u> : Address / Data Bit 27 <u>LBU</u> : Address Bit 19
168	AD28	LBU_AB20	B/I (I)		F15	<u>PCI</u> : Address / Data Bit 28 <u>LBU</u> : Address Bit 20
169	AD29	LBU_SEG_0	B/I (I)		B17	<u>PCI</u> : Address / Data Bit 29 <u>LBU</u> : Segment 0
170	AD30	LBU_SEG_1	B/I (I)		D14	<u>PCI</u> : Address / Data bit 30 <u>LBU</u> : Segment 1
171	AD31	LBU_CS_R_N	B/I (I)		B18	<u>PCI</u> : Address / Data bit 31 <u>LBU</u> : Chip select for accesses to paging configuration register
172	PME_N	LBU_RDY_N	B/O (I)		D16	<u>PCI mode</u> : Open drain; ext. PU necessary <u>LBU mode</u> : Ready signal; polarity dependent on <u>LBU_POL_RDY</u> input; output active while <u>LBU_CS_R/M_N</u> is active;
173	REQ_N	LBU_CS_M_N	O/I (T)		A19	<u>LBU mode</u> : Chip select for accesses to ERTEC 400-internal resources
174	GNT_N	LBU_CFG	I/I (I)		D17	<u>LBU mode</u> : 0 : Separate RD/WR line 1: <u>LBU_WR_N</u> has read/write control
175	CLK_PCI	---	I/I (I)		B19	<u>PCI bus clock</u>
176	RES_PCI_N	---	I/I (I)		E17	<u>PCI bus reset</u>
177	INTA_N	LBU_IRQ0_N	O/O (T)		D18	<u>PCI mode</u> : Open drain; ext. PU necessary <u>LBU mode</u> : No open drain
178	INTB_N	LBU_IRQ1_N	O/O (T)		B20	<u>PCI mode</u> : Open drain; ext. PU necessary <u>LBU mode</u> : No open drain
179	M66EN	---	B/I (I)		D19	Selection of 66/33 MHz <u>PCI</u> clock

1.5.8 RMII/MII

No.	Signal Name RMII	Signal Name MII	(5) I/O (Reset)	Pull-	Pin No.	Comment
PHY Management						
180	SMI_MDC	SMI_MDC	O/O (O)		M18	SMI clock
181	SMI_MDIO	SMI_MDIO	B/B (I)		N18	SMI input/output
182	RES_PHY_N	RES_PHY_N	O/O (O)		R21	Reset PHY
RMII-0/MII-0						
183	TXD_P0(0)	TXD_P0(0)	O/O (O)		P22	<u>RMII: Transmit data Port 0 Bit 0</u> <u>MII: Transmit data Port 0 Bit 0</u>
184	TXD_P0(1)	TXD_P0(1)	O/O (O)		N21	<u>RMII: Transmit data Port 0 Bit 1</u> <u>MII: Transmit data Port 0 Bit 1</u>
185	RXD_P0(0)	RXD_P0(0)	I/I (I)	dn	N22	<u>RMII: Receive data Port 0 Bit 0</u> <u>MII: Receive data Port 0 Bit 0</u>
186	RXD_P0(1)	RXD_P0(1)	I/I (I)	dn	M21	<u>RMII: Receive data Port 0 Bit 1</u> <u>MII: Receive data Port 0 Bit 1</u>
187	TX_EN_P0	TX_EN_P0	O/O (O)		P21	<u>RMII: Transmit enable Port 0</u> <u>MII: Transmit enable Port 0</u>
188	CRS_DV_P0	CRS_P0	I/I (I)	dn	H19	<u>RMII: Carrier sense/data valid Port 0</u> <u>MII: Carrier sense Port 0</u>
189	RX_ER_P0	RX_ER_P0	I/I (I)	dn	K18	<u>RMII: Receive error Port 0</u> <u>MII: Receive error Port 0</u>
RMII-1/MII-0						
190	TXD_P1(0)	TXD_P0(2)	O/O (O)		L22	<u>RMII: Transmit data Port 1 Bit 0</u> <u>MII: Transmit data Port 0 Bit 2</u>
191	TXD_P1(1)	TXD_P0(3)	O/O (O)		L21	<u>RMII: Transmit data Port 1 Bit 1</u> <u>MII: Transmit data Port 0 Bit 3</u>
192	RXD_P1(0)	RXD_P0(2)	I/I (I)	dn	K21	<u>RMII: Receive data Port 1 Bit 0</u> <u>MII: Receive data Port 0 Bit 2</u>
193	RXD_P1(1)	RXD_P0(3)	I/I (I)	dn	J21	<u>RMII: Receive data Port 1 Bit 1</u> <u>MII: Receive data Port 0 Bit 3</u>
194	TX_EN_P1	TX_ERR_P0	O/O (O)		M19	<u>RMII: Transmit enable Port 1</u> <u>MII: Transmit Error Port0</u>
195	CRS_DV_P1	RX_DV_P0	I/I (I)	dn	G19	<u>RMII: Carrier sense/data valid Port 1</u> <u>MII: Receive data valid Port 0</u>
196	RX_ER_P1	COL_P0	I/I (I)	dn	M22	<u>RMII: Receive error Port 1</u> <u>MII: Collision Port 0</u>
197	-	RX_CLK_P0	I/I (I)	dn	K19	<u>MII: Receive clock Port 0</u>
198	-	TX_CLK_P0	I/I (I)	dn	K17	<u>MII: Transmit clock Port 0</u>
RMII-2/MII-1						
199	TXD_P2(0)	TXD_P1(0)	O/O (O)		H21	<u>RMII/MII: Transmit data Port 2 Bit 0</u> <u>MII: Transmit data Port 1 Bit 0</u>
200	TXD_P2(1)	TXD_P1(1)	O/O (O)		G22	<u>RMII: Transmit data Port 2 Bit 1</u> <u>MII: Transmit data Port1 Bit1</u>
201	RXD_P2(0)	RXD_P1(0)	I/I (I)	dn	F21	<u>RMII: Receive data Port 2 Bit 0</u> <u>MII: Receive data Port 1 Bit 0</u>
202	RXD_P2(1)	RXD_P1(1)	I/I (I)	dn	E22	<u>RMII: Receive data Port 2 Bit 1</u> <u>MII: Receive data Port 1 Bit 1</u>
203	TX_EN_P2	TX_EN_P1	O/O (O)		H22	<u>RMII: Transmit enable Port 2</u> <u>MII: Transmit enable Port 1</u>
204	CRS_DV_P2	CRS_P1	I/I (I)	dn	G18	<u>RMII: Carrier sense/data valid Port 2</u> <u>MII: Carrier sense Port 1</u>
205	RX_ER_P2	RX_ER_P1	I/I (I)	dn	J18	<u>RMII: Receive error Port 2</u> <u>MII: Receive error Port 1</u>

No.	Signal Name RMII	Signal Name MII	(5) I/O (Reset)	Pull-	Pin No.	Comment
RMII-3/MII-1						
206	TXD_P3(0)	TXD_P1(2)	O/O (O)		C22	<u>RMII</u> : Transmit data Port 3 Bit 0 <u>MII</u> : Transmit data Port 1 Bit 2
207	TXD_P3(1)	TXD_P1(3)	O/O (O)		C21	<u>RMII</u> : Transmit data Port 3 Bit 1 <u>MII</u> : Transmit data Port 1 Bit 3
208	RXD_P3(0)	RXD_P1(2)	I/I (I)	dn	B22	<u>RMII</u> : Receive data Port 3 Bit 0 <u>MII</u> : Receive data Port 1 Bit 2
209	RXD_P3(1)	RXD_P1(3)	I/I (I)	dn	B21	<u>RMII</u> : Receive data Port 3 Bit 1 <u>MII</u> : Receive data Port 1 Bit 3
210	TX_EN_P3	TX_ERR_P1	O/O (O)		D21	<u>RMII</u> : Transmit enable Port 3 <u>MII</u> : Transmit error Port 1
211	CRS_DV_P3	RX_DV_P1	I/I (I)	dn	F19	<u>RMII</u> : Carrier sense/data valid Port 3 <u>MII</u> : Receive data valid Port 1
212	RX_ER_P3	COL_P1	I/I (I)	dn	E21	<u>RMII</u> : Receive error Port 3 <u>MII</u> : Collision Port 1
213	-	RX_CLK_P1	I/I (I)	dn	H18	<u>MII</u> : Receive clock Port 1
214	-	TX_CLK_P1	I/I (I)	dn	J17	<u>MII</u> : Transmit clock Port 1

1.5.9 Power Supply

No.	Voltage Signal Name	I/O	Pin No.	Comment
Power Supply				
215 - 244	VDD Core	P	B11, D6, D9, D15, E5, E18, E19, F6, F17, H4, J19, K4, L19, M4, N5, N19, P4, P18, R18, U4, U6, U17, V5, V14, V18, W7, W8, W14, W15, W17	SV Core 1.5 V (30 pins)
245 - 261	GND Core	P	E6, F10, F16, G6, G17, K6, L18, T17, U1, U5, U7, U9, U13, U14, U18, V9, V17	GND CORE (17 pins)
262 - 280	VDD IO	P	A3, A5, A8, A11, A15, A18, A20, E1, L1, R1, Y1, AB3, AB7, AB11, AB15, AB19, D22, J22, T22	SV IO 3.3 V (19 pins)
281 - 296	GND IO	P	A7, A21, E16, F7, F13, F18, G5, K22, M1, P6, T6, U16, U22, V7, V13, AB13,	GND IO (16 pins)
297 - 299	P5V_PCI (4)	P	F8, D12, E15	5 V VDD for PCI (3 pins)
300	AVDD	P	W11	SV Analog 1.5 V (1 pin)
301	AGND	P	U10	GND Analog (1 pin)
302	AVDD_PCI	P	F22	SV Analog PCI 1.5 V (1 pin)
303	AGND_PCI	P	G21	GND Analog PCI (1 pin)
304	TACT_N		H17	Not used

Table 1: ERTEC 400 Pin Assignment and Signal Description

2 ARM946E-S Processors

The ARM946E-S processor is implemented in the ERTEC 400. This description is based on /1/ and /2/.

2.1 Structure of ARM946E-S

An ARM946E-S processor system is used. The figure below shows the structure of the processor. In addition to the processor core, the system contains one data cache, one instruction cache, a memory protection unit (MPU), a system control coprocessor, and a tightly coupled memory. The processor system has an interface to the integrated AHB bus.

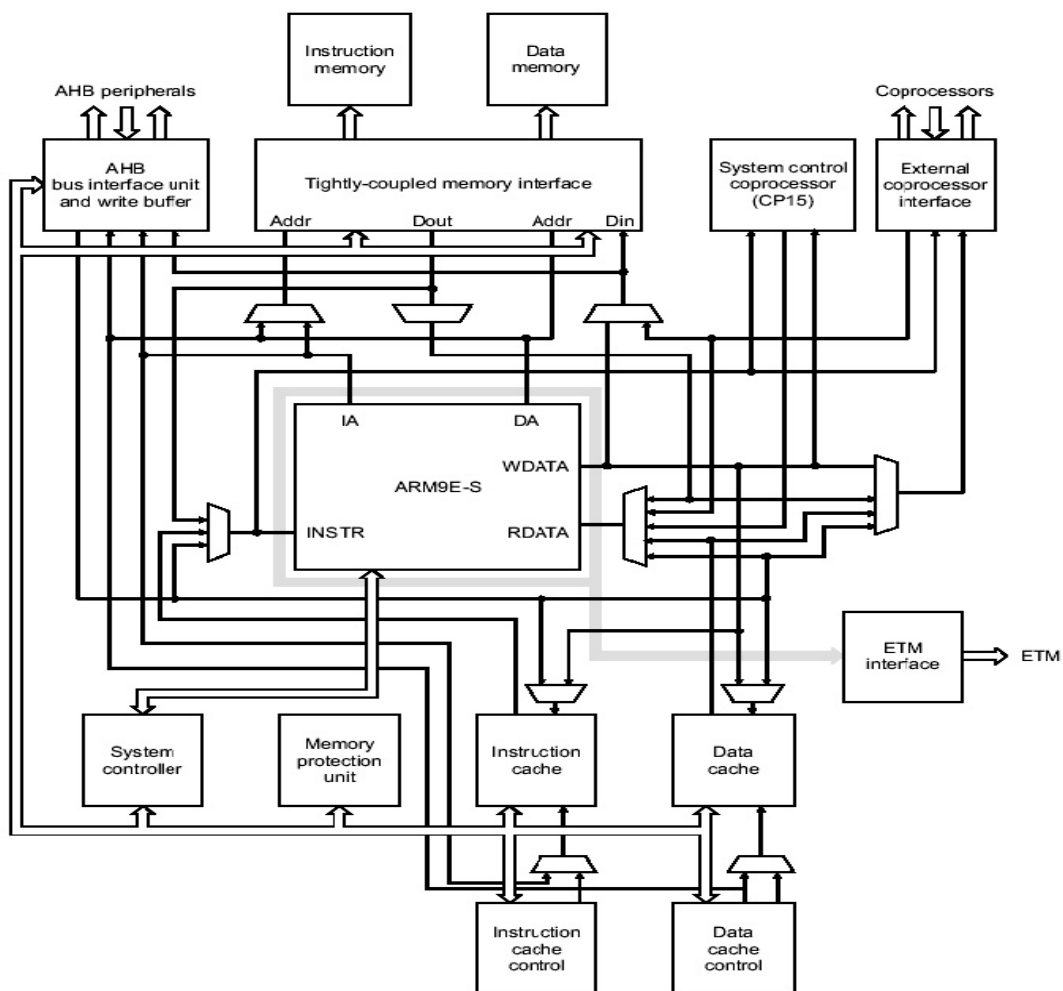


Figure 3: Structure of ARM946E-S Processor System

2.2 Description of ARM946E-S

The ARM946E-S processor system is a member of the ARM9 Thumb family. It has a processor core with Harvard architecture. Compared to the standard ARM9 family, the ARM946E-S has an enhanced V5TE architecture permitting faster switching between ARM and Thumb code segments and an enhanced multiplier structure. In addition, the processor has an integrated JTAG interface.

2.3 Operating Frequency of ARM946E-S

The processor can be operated at 50 MHz, 100 MHz, or 150 MHz. The operating frequency is set during the reset phase via the **CONFIG[3]** and **CONFIG[4]** configuration pins. Communication with the components of the ERTEC 400 takes place via the AHB bus at a frequency of 50 MHz.

2.4 Cache Structure of ARM946E-S

The following caches are integrated in the ARM946E-S.

- 8 Kbytes of instruction cache with lock function
- 4 Kbytes of data cache with lock function

Both caches are "Four-Way Set Associative" caches with 1-Kbyte segments. Each segment consists of 32 lines with 32 bytes (8 x 4 bytes). The D-cache has "write buffers" with write-back function.

The lock function enables the user to lock the contents of the cache segments ("LOCK"). This function enables the command set for fast routines to be maintained permanently in the instruction cache. This mechanism can only be implemented on a segment-specific basis in the ARM946E-S.

Both caches are locked after a reset. The caches can be enabled only if the "memory protection unit" is enabled at the same time.

The I-cache can be enabled by setting **Bit 12** of the **CP15 control register**.

The D-cache can be enabled by setting **Bit 2** of the **CP15 control register**.

Access to this area is blocked if the cache is not enabled.

For more information on caching, refer to **Section 3** of /1/.

For more information on the description of the ARM946 registers, refer to Section **2.10** of this document.

2.5 Tightly Coupled Memory (TCM)

A 4-Kbyte data TCM (D-TCM) is implemented in the ARM946E-S processor of the ERTEC 400. The memory is locked after a reset. The D-TCM can be placed anywhere in the address space of the ARM946E-S and must be used together with a region of the memory protection unit. Data of fast routines can be placed in the D-TCM.

The D-TCM can be enabled by setting **Bit 16** of the **CP15 control register**.

In addition, the address area of the D-TCM must be set in the **Tightly-Coupled Memory register**.

For more information on the D-TCM, refer to **Section 5** of /1/.

For more information on the description of the ARM946 registers, refer to Section **2.10** of this document.

2.6 Memory Protection Unit (MPU)

The memory protection unit enables the user to partition certain memory areas (I-cache, D-cache, or D-TCM) into various regions and to assign various attributes to these regions.

A maximum of 8 regions of variable size can be set. If regions overlap, the attributes of the higher region number apply.

Settings for each region:

- Base address of region
- Size of region
- Cache and “write buffer” configuration
- Read/write access enable for privileged users/users

Settings are made in the following registers of the ARM946E-S:

- Register 2 “Cache configuration register”
- Register 3 “Write buffer control register”
- Register 5 “Access permission register”
- Register 6 “Protection region/base size register”

The base address defines the start address of the region. It must always be a multiple of the size of the region. Example: The region size is 4 Kbytes. The starting address is then always a multiple of 4 Kbytes.

Before the MPU is enabled, at least one region must have been assigned. Otherwise, the ARM946E-S can assume a state that can only be cancelled by a reset.

The MPU can be enabled by setting **Bit 0** of the **CP15 control register**.

If the MPU is locked, neither an I-cache nor a D-cache can be accessed even if they are enabled.

For more information on the **MPU**, refer to **Section 4** of /1/.

For more information on the description of the ARM946 registers, refer to Section **2.10** of this document.

2.7 Bus Interface of ARM946E-S

The ARM946E-S uses an AHB bus master interface to the multilayer AHB bus for opcode fetches and data transfers. The interface operates at a fixed frequency of 50 MHz. The data bus and address bus each have a width of 32 bits.

For more information on the bus interface and for the various transfer types, refer to Section 6 of /1/.

2.8 ARM946E-S Embedded Trace Macrocell (ETM9)

An ETM9 module is connected at the ARM946E-S. This module permits debugging support for data and instruction traces in the ERTEC 400. The module contains all signals required by the processor for the data and instruction traces. The ETM9 module is operated by means of the JTAG interface. The trace information is provided outwards to the trace port via a FIFO memory. A more detailed description is in Section

2.9 ARM Interrupt Controller (ICU)

The interrupt controller supports the FIQ and IRQ interrupt levels of the ARM946 processor. An interrupt controller with 8 interrupt inputs is implemented for FIQ. Two interrupt inputs (FIQ6-7) are assigned internally and 6 interrupt inputs (FIQ0-5) are available for external events. An interrupt controller for 16 interrupt inputs is implemented for IRQ. Of the 16 IRQ inputs, 2 IRQ sources can be selected for processing as fast interrupt requests (FIQ6-7). The assignment is made by specifying the IRQ number of the relevant interrupt input in the FIQ1REG / FIQ2REG register. The interrupt inputs selected as FIQ must be disabled for the IRQ logic. All other interrupt inputs can continue to be processed as IRQs.

The interrupt controller is operated at a clock frequency of 50 MHz. Interrupt request signals that are generated at a higher clock frequency must be extended accordingly for error-free detection.

2.9.1 Prioritization of Interrupts

It is possible to set the priorities of the IRQ and FIQ interrupts. Priorities 0 to 15 can be assigned to IRQ interrupts while priorities 0 to 7 can be assigned to FIQ interrupts. The highest priority is 0 for both interrupt levels. After a reset, all IRQ interrupt inputs are set to priority 15 and all FIQ interrupt inputs are set to priority 7. A priority register is associated with each interrupt input. PRIOREG0 to PRIOREG15 are for the IRQ interrupts and FIQPR0

to FIQPR7 are for the FIQ interrupts. A priority must not be assigned more than once. The interrupt control logic does not check for assignment of identical priorities. All interrupt requests with a lower or equal priority can be blocked at any time in the IRQ priority resolver by assigning a priority in the LOCKREG register. If an interrupt that is to be blocked is requested at the same time as the write access to the LOCKREG register, an IRQ signal is output. However, the signal is revoked after two clock cycles. If an acknowledgement is to be generated nonetheless, the transferred interrupt vector is the default vector.

2.9.2 Trigger Modes

There are two modes available for each interrupt input: "edge-triggered" and "level-triggered".

The trigger type is defined by means of the assigned bit in the TRIGREG register. For the "Edge-triggered" mode setting, differentiation can be made between a positive and negative edge evaluation. This is made in the EDGEREG register. "Edge-triggered" with positive edge is the default mode assignment for all interrupts. "High" is the active level in the "level-triggered" mode.

The interrupt input signal must be present for at least one clock cycle in "edge-triggered" mode. The input signal must be present up until confirmation of the ARM946E-S CPU in "level-triggered" mode. Shorter signals result in loss of the event.

2.9.3 Masking the Interrupt Inputs

Each IRQ interrupt can be enabled or disabled individually. The MASKREG register is available for this purpose. The interrupt mask acts only after the IRREG interrupt request register. That is, an interrupt is entered in the IRREG register in spite of the block in the MASKREG register. After a reset, all mask bits are set and, thus, all interrupts are disabled. At a higher level, all IRQ interrupts can be disabled globally via a command. When IRQ interrupts are enabled globally via a command, only those IRQ interrupts that are enabled by the corresponding mask bit in the MASKREG register are enabled.

For the FIQ interrupts, only selective masking by the mask bits in the FIQ_MASKREG register is possible. After a reset, all FIQ interrupts are disabled. A detected FIQ interrupt request is entered in the FIQ interrupt request register. If the interrupt is enabled in the mask register, processing takes place in the priority logic. If the interrupt request is accepted by the ARM946 CPU and an entry is made in the in-service request register (ISR), the corresponding bit is reset in the IRREG register. Each bit that is set in the IRREG register can be deleted via software. For this purpose, the number of the bit to be reset in the IRCLVEC register is transferred to the interrupt controller.

2.9.4 Software Interrupts for IRQ

Each IRQ interrupt request can be triggered by setting the bit corresponding to the input channel in the SWIRREG software interrupt register. Multiple requests can also be entered in the 16-bit SWIRREG register. The software interrupt requests are received directly in the IRREG register and, thus, treated like a hardware IRQ. Software interrupts can only be triggered by the ARM946E-S processor because only this processor has access rights to the interrupt controller.

2.9.5 Nested Interrupt Structure

When enabled by the interrupt priority logic, an IRQ interrupt request causes an IRQ signal to be output. Similarly, an FIQ interrupt request causes the FIQ signal to be output to the CPU.

If the request is accepted by the CPU (in the IRQACK or FIQACK register), the bit corresponding to the physical input is set in the ISREG or FIQISR register. The IRQ/FIQ signal is revoked. The ISR bit of the accepted interrupt remains set until the CPU returns an "End-of-interrupt" command to the interrupt controller. As long as the ISR bit is set, interrupts with lower priority in the priority logic of the interrupt controller are disabled. Interrupts with a higher priority are allowed by the priority logic to pass and generate an IRQ/FIQ signal to the CPU. As soon as the CPU accepts this interrupt, the corresponding ISR bit in the ISREG or FIQISR register is also set. The CPU then interrupts the lower-priority interrupt routine and executes the higher interrupt routine first. Lower-priority interrupts are not lost. They are entered in the IRREG register and are processed at a later time when all higher-priority interrupt routines have been executed.

2.9.6 EOI End-Of-Interrupt

A set ISR bit is deleted by the End-of-Interrupt command. The CPU must use the EOI command to communicate this to the interrupt controller after the corresponding interrupt service routine is processed. To communicate the EOI command to the interrupt controller, the CPU writes any value to the IRQEND/FIQEND register. The interrupt controller autonomously decides which ISR bit is reset with the EOI command. If several ISR bits are set, the interrupt controller deletes the ISR bit of the interrupt request with the highest priority at the time of the EOI command. The interrupt controller regards the interrupt cycle as ended when all of the set ISR bits have been reset by the appropriate number of EOI commands. Afterwards, low priority interrupts that occurred in the meantime and were entered in the IRREG register can be processed in the priority logic.

During one or more accepted interrupts, the priority distribution of the IRQ/FIQ interrupt inputs must not be changed because the ICU can otherwise no longer correctly assign the EOI commands.

An IRQ/FIQ request is accepted by the CPU by reading the IRVEC/FIVEQ register. This register contains the binary-coded vector number of the highest priority interrupt request at the moment. Each of the two interrupt vector registers can be referenced using two different addresses. The interrupt controller interprets the reading of

the vector register with the first address as an “interrupt acknowledge”. This causes the sequences for this interrupt to be implemented in the ICU logic. Reading of the vector register with the second address is not linked to the “acknowledge function”. This is primarily useful for the debugging functions in order to read out the content of the interrupt vector register without starting the acknowledge function of the interrupt controller.

2.9.7 IRQ Interrupt Sources

Int. No.	Function Block	Signal Name	Default Setting	Comment
0	Timer	TIM_INT0	Rising edge	Timer 0
1	Timer	TIM_INT1	Rising edge	Timer 1
3:2	GPIO	GPIO (1:0)	Assignable	External input ERTEC 400 GPIO[1:0]
5:4	GPIO	GPIO (31:30)	Assignable	External input ERTEC 400 GPIO[31:30]
6	ARM-CPU	COMM_Rx	Rising edge	Debug receive communications channel interrupt
7	ARM-CPU	COMM_Tx	Rising edge	Debug transmit communications channel interrupt
8	UART_1	UART_INTR1	High level	Group interrupt UART1
9	UART_2	UART_INTR2	High level	Group interrupt UART2
10	SPI	SSP_INTR	Rising edge	Group interrupt SPI
11	SPI	SSP_ROR_INTR	Rising edge	Receive overrun interrupt SPI
12	IRT switch	IRQ0_SP	Rising edge	High-priority IRT interrupt
13	IRT switch	IRQ1_SP	Rising edge	Low-priority IRT interrupt
14	IRT switch	IRQ0_IRT_API_ERR	Rising edge	Synchronization error in IRT API (cannot be masked in IRT)
15	PCI	AHB_INT_L	Rising edge	AHB PCI bridge

Table 2: Overview of IRQ Interrupts

2.9.8 FIQ Interrupt Sources

Int. No.	Function Block	Signal Name	Default Setting	Comment
0	Watchdog		Rising edge	
1	APB bus		Rising edge	Access to non-existing address at the APB (1)
2	Multilayer AHB		Rising edge	Access to non-existing address at the AHB (1)
3	PLL-Status-Register		Rising edge	Group interrupt of: EMIF: I/O QVZ PCI: Slave QVZ PLL: Loss state PLL: Lock State see system control register "PLL_STAT_REG"
4	IRT switch	IRQ_IRT_END	Rising edge	End of the IRT phase (cannot be masked in IRT)
5	IRT switch	IRQ_LOW_WATER	Rising edge	“Low water mark” is violated
6	Optional	Optional from IRQ	Rising edge	default: IRQ0_SP
7	Optional	Optional from IRQ	Rising edge	User-programmable by IRQ

Table 3: Overview of FIQ Interrupts

- (1) Access to non-existing addresses is detected by the individual function groups of the ERTEC 400 and triggers a pulse with duration $T_p = 2/50$ MHz. For evaluation of this interrupt, the connected FIQ input must be specified as an edge-triggered input.

2.9.9 IRQ Interrupts as FIQ Interrupt Sources

The interrupts of the FIQ interrupt controller are used for debugging, monitoring of address space accesses, and high-priority switch functions.

High-priority IRT interrupt **IRQ0_SP** can be used as an FIQ by means of interrupt input **FIQ 6**. Selection of the high-priority interrupt source is specified in the IRQ0_SP interrupt controller of the IRT switch.

IRQ interrupts No. 6 and 7 are the interrupts of embedded ICE RT communication. The UART can also be used as a debugger in place of the ICE. An effective realtime debugging is possible if the IRQ interrupt sources of the

UART or the ICE communication channel is mapped to the FIQs with numbers 6 or 7. This enables debugging of interrupt routines.

2.9.10 Interrupt Control Register

The interrupt control registers are used to specify all aspects of control, prioritization, and masking of the IRQ/FIQ interrupt controllers.

ICU (Base Address 0x5000_0000)					
Register Name	Offset Address	Address Area	Access	Default	Description
IRVEC	0x0000	4 bytes	R	0x00000000	Interrupt vector register
FIVEC	0x0004	4 bytes	R	0x00000000	Fast interrupt vector register
LOCKREG	0x0008	4 bytes	R/W	0x00000000	Priority lock register
FIQ1SREG	0x000C	4 bytes	R/W	0x00000000	Fast int. request 1 select register (FIQ6 on FIQ interrupt controller)
FIQ2SREG	0x0010	4 bytes	R/W	0x00000000	Fast int. request 2 select register (FIQ7 on FIQ interrupt controller)
IRQACK	0x0014	4 bytes	R	0x00000000	Interrupt vector register with IRQ acknowledge
FIQACK	0x0018	4 bytes	R	0x00000000	Fast interrupt vector register with FIQ acknowledge
IRCLVEC	0x001C	4 bytes	W	0x----	Interrupt request clear vector
MASKALL	0x0020	4 bytes	R/W	1	Mask for all interrupts
IRQEND	0x0024	4 bytes	W	0x----	End of IRQ interrupt
FIQEND	0x0028	4 bytes	W	0x----	End of FIQ interrupt
FIQPR0	0x002C	4 bytes	R/W	0x00000007	FIQ priority register on input FIQ0 of the FIQ interrupt controller
FIQPR1	0x0030	4 bytes	R/W	0x00000007	FIQ priority register on input FIQ1 of the FIQ interrupt controller
FIQPR2	0x0034	4 bytes	R/W	0x00000007	FIQ priority register on input FIQ2 of the FIQ interrupt controller
FIQPR3	0x0038	4 bytes	R/W	0x00000007	FIQ priority register on input FIQ3 of the FIQ interrupt controller
FIQPR4	0x003C	4 bytes	R/W	0x00000007	FIQ priority register on input FIQ4 of the FIQ interrupt controller
FIQPR5	0x0040	4 bytes	R/W	0x00000007	FIQ priority register on input FIQ5 of the FIQ interrupt controller
FIQPR6	0x0044	4 bytes	R/W	0x00000007	FIQ priority register on input FIQ6 of the FIQ interrupt controller
FIQPR7	0x0048	4 bytes	R/W	0x00000007	FIQ priority register on input FIQ7 of the FIQ interrupt controller
FIQISR	0x004C	4 bytes	R	0x00000000	FIQ in-service register
FIQIRR	0x0050	4 bytes	R	0x00000000	FIQ request register
FIQ_MASKREG	0x0054	4 bytes	R/W	0x000000FF	FIQ interrupt mask register
IRREG	0x0058	4 bytes	R	0x00000000	Interrupt request register
MASKREG	0x005C	4 bytes	R/W	0x0000FFFF	Interrupt mask register
ISREG	0x0060	4 bytes	R	0x00000000	In-service register
TRIGREG	0x0064	4 bytes	R/W	0x00000000	Trigger select register
EDGEREG	0x0068	4 bytes	R/W	0x00000000	Edge select register
SWIRREG	0x006C	4 bytes	R/W	0x00000000	Software interrupt register
PRIOREG 0	0x0070	4 bytes	R/W	0x0000000F	Priority register 0
PRIOREG 1	0x0074			
...
...
PRIOREG15	0x00AC	4 bytes	R/W	0x0000000F	Priority register 15

Table 4: Overview of Interrupt Control Register

2.9.11 ICU Register Description

IRVEC		R	Addr.: 0x5000_0000	Default: 0x0000_0000
Description		Interrupt vector register Input with highest priority pending interrupt request		
Bit No.	Name	Description		
3:0	IRVEC	For pending, valid interrupt: Binary code of input number. Default vector: Bit[3:0] = 1		
31:4	Vector ID	For pending, valid interrupt: Bit[31:4] = 0. Default vector: Bit[31:4] = 1		

FIVEC		R	Addr.: 0x5000_0004	Default: 0x0000_0000
Description		Fast interrupt vector register Number of the highest-priority pending fast interrupt request		
Bit No.	Name	Description		
2:0	FIVEC	For pending, valid interrupt: Binary code of FIQ number. Default vector: Bit[2:0] = 1		
31:3	Vector ID	For pending valid Bit[31:3] = 0. Default vector: Bit[31:3] = 1		

LOCKREG		R/W	Addr.: 0x5000_0008	Default: 0x0000_0000
Description		Priority lock register Specification of a priority for blocking interrupt requests of lower and equal priority		
Bit No.	Name	Description		
3 – 0	LOCKPRIO	Binary code of lock priority.		
7	LOCKENABLE	0=Lock inactive / 1=Lock active		

FIQ1SREG		R/W	Addr.: 0x5000_000C	Default: 0x0000_0000
Description		Fast interrupt request 1 select register Declaration of an IRQ input as FIQ6 (input FIQ6 on FIQ interrupt controller)		
Bit No.	Name	Description		
3 – 0	FIQ1SREG	Number of the input to be selected (binary code)		
7	FIQ1SENABLE	0=Ignore FIQ declaration 1=Take into account FIQ declaration		

FIQ2SREG		R/W	Addr.: 0x5000_0010	Default: 0x0000_0000
Description		Fast interrupt request 2 select register Declaration of an IRQ input as FIQ7 (input FIQ7 on FIQ interrupt controller)		
Bit No.	Name	Description		
3 – 0	FIQ2SREG	Number of the input to be selected (binary code)		
7	FIQ2SENABLE	0=Ignore FIQ declaration 1=Take into account FIQ declaration		

IRQACK		R	Addr.: 0x5000_0014	Default: 0x0000_0000
Description		Interrupt vector register with IRQ acknowledge Confirmation of highest-priority pending interrupt request by reading the associated interrupt vector		
Bit No.	Name	Description		
3 – 0	IRVEC	Binary code of input number		
31 - 4	Vector ID	Valid IRQ vector: always '0'. Default vector: always '1' (also bits 3 – 0).		

FIQACK		R	Addr.: 0x5000_0018	Default: 0x0000_0000
Description		Fast interrupt vector register with FIQ acknowledge Confirmation of fast interrupt request by reading the associated interrupt vector		
Bit No.	Name	Description		
2 – 0	FIVEC	Binary code of FIQ number		
31 – 3	Vector ID	Valid FIQ vector: always '1'. Default vector: always '1' (also bits 2 – 0).		

IRCLVEC		W	Addr.: 0x5000_001C	Default: ----
Description		Interrupt request clear vector Immediate deletion of an interrupt request in the interrupt request register		
Bit No.	Name	Description		
3 – 0	IRCLVEC	Binary code of the input number of the request to be deleted		
7	Not used			

MASKALL		R/W	Addr.: 0x5000_0020	Default: '1'
Description		Mask all Interrupts Global disable for all IRQ interrupt inputs		
Bit No.	Name	Description		
0	MASKALL	'0' = Enable all non-masked IRQ interrupt inputs (consideration given to set mask bits) '1' = Global disable for all IRQ interrupt inputs (independent of the interrupt mask)		

IRQEND		W	Addr.: 0x5000_0024	Default: ----
Description		End-of-interrupt (IRQ) Communicates the completion of the interrupt service routine associated with the current request to the IRQ interrupt controller		
Bit No.	Name	Description		
	Not used			

FIQEND		W	Addr.: 0x5000_0028	Default: ----
Description		End-of-interrupt (FIQ) Communicates the completion of the interrupt service routine associated with the fast interrupt request to the FIQ interrupt controller		
Bit No.	Name	Description		
	Not used			

FIQPR0		R/W	Addr.: 0x5000_002C	Default: 0x0000_0007
...			
FIQPR7		R/W	Addr.: 0x5000_0048	Default: 0x0000_0007
Description		FIQ priority registers Priority of the fast interrupt request at inputs FIQ0 to FIQ7 of the FIQ interrupt controller		
Bit No.	Name	Description		
2 – 0	FIQPR0 to FIQPR7	Binary code of the priority		
7 – 3	Not used			

FIQISR		R	Addr.: 0x5000_004C	Default: 0x0000_0000
Description		FIQ in-service register Indication of the fast interrupt requests confirmed by the CPU		
Bit No.	Name	Description		
7 – 0	FIQISR	Inputs 0 to 7 of the FIQ interrupt controller '0' = Fast interrupt request not confirmed '1' = Fast interrupt request has been confirmed		

FIQIRR		R	Addr.: 0x5000_0050	Default: 0x0000_0000
Description		FIQ request register Indication of the fast interrupt requests detected on the basis of a positive edge		
Bit No.	Name	Description		
7 – 0	FIQIRR	Inputs 0 to 7 of the FIQ interrupt controller '0' = No request '1' = Request is occurred		

FIQ_MASKREG		R/W	Addr.: 0x5000_0054	Default: 0x0000_00FF
Description		Interrupt mask register for FIQ Enable/disable of FIQ interrupt inputs		
Bit No.	Name	Description		
7 – 0	FIQ_MASKREG	FIQ interrupt input 0 to 7 '0' = Interrupt input enabled '1' = Interrupt input disabled		

IRREG		R	Addr.: 0x5000_0058	Default: 0x0000_0000
Description		Interrupt request register Storage of interrupt requests that have occurred		
Bit No.	Name	Description		
15 – 0	IRREG	Interrupt input 0 to 15 0 =Interrupt request inactive/ 1 =Interrupt request active		

MASKREG		R/W	Addr.: 0x5000_005C	Default: 0x0000_FFFF
Description		Interrupt mask register Enable/disable of interrupt inputs		
Bit No.	Name	Description		
15 – 0	MASKREG	Interrupt input 0 to 15 0 =Interrupt input enabled/ 1 =Interrupt input disabled		

ISREG		R	Addr.: 0x5000_0060	Default: 0x0000_0000
Description		In-service register Indication of the interrupt requests confirmed by the CPU		
Bit No.	Name	Description		
15 – 0	ISREG	Interrupt input 0 to 15 0 =Interrupt request not confirmed 1 =Interrupt request has been confirmed		

TRIGREG		R/W	Addr.: 0x5000_0064	Default: 0x0000_0000
Description		Trigger select register Selection of interrupt detection		
Bit No.	Name	Description		
15 – 0	TRIGREG	Interrupt input 0 to 15 0 =Interrupt detection via edge 1 =Interrupt detection via level		

EDGEREG		R/W	Addr.: 0x5000_0068	Default: 0x0000_0000
Description		Edge select register Edge selection for interrupt detection (only if edge detection is specified for the associated input)		
Bit No.	Name	Description		
15 – 0	EDGEREG	Interrupt input 0 to 15 0 =Interrupt detection via positive edge 1 =Interrupt detection via negative edge		

SWIRREG		R/W	Addr.: 0x5000_006C	Default: 0x0000_0000
Description		Software interrupt register Specification of interrupt requests		
Bit No.	Name	Description		
15 – 0	SWIRREG	Interrupt input 0 to 15 0=No interrupt request 1=Set interrupt request		

PRIOREG 0		R/W	Addr.: 0x5000_0070	Default: 0x0000_000F
...			
PRIOREG 15		R/W	Addr.: 0x5000_00AC	Default: 0x0000_000F
Description		Priority register Specification of priority of an interrupt request at the associated input		
Bit No.	Name	Description		
3 – 0	PRIOREG	Binary code of the priority		

2.10 ARM946E-S Register

The ARM946E-S uses CP15 registers for system control.

Consequently, the following settings are possible:

- Configure cache type and cache memory area
- Configure tightly coupled memory area
- Configure memory protection unit for various regions and memory types
- Assign system option parameters
- Configure “Little Endian” or “Big Endian” operations

The following table shows all CP15 registers with their access options.

Register	Access	Description
0	R	ID code register (1) Cache type register (1) Tightly coupled memory size register (2)
1	W/R	Control register
2	W/R	Cache configuration register (2)
3	W/R	Write buffer control register
4	xxx	Undefined
5	W/R	Access permission register (2)
6	W/R	Protection region base/size register (2)
7	W	Cache operation register
8	xxx	Undefined
9	W/R	Cache lockdown register (2)
10	xxx	Undefined
11	xxx	Undefined
12	xxx	Undefined
13	W/R	Trace process ID register
14	xxx	Undefined
15	W/R	RAM/TAG-BIST test register (1) Test state register (1) Cache debug index register (1) Trace control register

Table 5: CP15 Registers - Overview

(1) Registers contain multiple information entries that are selected by the “opcode_2” or “CRm” fields.

(2) Separate registers for instruction and data (see detailed description of registers).

Undefined means:

When this register is read, the read value is undefined.

When this register is written to, unforeseeable configuration changes can occur in the ARM946.

Refer to documents /1/ and /2/ for a detailed description of the ARM946 registers.

3 Bus System of the ERTEC 400

The ERTEC 400 has two different buses internally.

- **High-performance communication bus (multilayer AHB bus)**
- **I/O bus (APB bus)**

The following function blocks are connected directly to the multilayer AHB bus:

- ARM946E-S (Master/Slave)
- IRT switch (Master)
- LBU (Master)
- PCI (Master/Slave)
- Interrupt controller (Slave)
- Local SRAM (Slave)
- EMIF interface (Slave)

The master can access the remaining I/O connected to the low-performance APB bus via an AHB/APB bridge. Of the PCI and LBU masters, **only one can be active**. The selection is made using the **CONFIG[2]** input pin.

3.1 “Multilayer AHB” Communication Bus

The multilayer AHB bus is characterized by a high bus availability and data transmission. The multilayer AHB bus is a 32-bit wide bus with multiple master capability. It runs at a frequency of 50 MHz and has the functionality of the ARM AHB bus (see Section 3 of document /4/). Through interconnection of multiple AHB segments in the multilayer AHB bus, three masters can access various slaves simultaneously. Of the PCI bridge and LBU masters, only one is active at a time. The active node is set by the CONFIG[2] hardware pin during the reset phase.

3.1.1 AHB Arbiter

Arbiters control the access when multiple masters access a slave simultaneously. Each of the AHB arbiters uses the same arbitration procedure. “Round robin” is specified. Alternatively, a fixed priority assignment of the AHB master can be set through assignment of the ARB_MODE bit in the M_LOCK_CNTL system control register. Fixed priority assignment should be avoided due to the dynamic sequences on the multilayer AHB bus. The round robin arbitration procedure prevents mutual blocking of the AHB master over a long period on the multilayer AHB bus.

If the priority assignment is fixed, the ARM has the highest priority, followed by IRT and finally, PCI/LBU.

3.1.2 AHB Master-Slave Coupling

Not every AHB master is connected to each AHB slave. The table below shows which AHB masters can communicate with which AHB slaves.

AHB Master-Slave Coupling						
Slave Master	APB Slave 1	EMIF Slave 2	PCI Slave 3	IRT Slave 4	Local RAM Slave 5	INT Control Slave 6
ARM	X	X	X	X	X	X
IRT		X	X		X	
PCI/LBU	X	X		X	X	

Table 6: Overview of AHB Master-Slave Access

For closed-loop control applications, attention must be paid that AHB masters do not block each other over a long period. This would be possible if, for example, a PCI master and ARM master want to access the same IRT slave with a time lag. In this case, the ARM master would have to pause in a “Wait” until the PCI master enables the IRT slave again. To prevent this situation, monitoring is integrated into the PCI/LBU AHB masters and IRT switch, which enables the slave momentarily via an IDLE state after 8 consecutive data transfers (burst or single access). In this phase, another AHB master can access this slave.

3.2 APB I/O Bus

The APB bus is connected by means of an AHB/APB bridge on the multilayer AHB bus. The APB bus has a width of 32 bits and operates at a frequency of 50 MHz.

4 I/O on APB bus

The ERTEC 400 block has multiple I/O function blocks. They are connected to the 32-bit APB I/O bus. The ARM946E-S or PCI/LBU interface can access the I/O. The following I/O are available.

- 8 Kbyte Boot ROM
- 32-bit GPIO (*)
- UART1, UART2
- SPI interface
- Timer 0, Timer 1
- F-timer
- Watchdog
- System control register

(*) The complete 32 bits for GPIO input/output are only available if alternative functions are not assigned.

The I/O function blocks connected to the APB bus have data interfaces of different widths. The data bit width and the supported access mechanisms are shown in the table below. Non-permitted access types such as byte-by-byte loading of timer reload registers are not intercepted on the hardware side.

SUPPORTED ACCESS TYPES				Function Block
Bit 31:24	Bit 23:16	Bit 15:8	Bit 7:0	
32 bit				Timer, F-counter, Watchdog, System control register Boot_Rom
8 bit	8 bit	8 bit	8 bit	GPIO
16 bit		16 bit		
32 bit				
		16 bit		SPI
			8 bit	UART1/2

Table 7: Access Type and Data Bit Width of I/O

Accesses to non-decoded-out memory or register areas trigger an FIQ interrupt. Access by a generated "Ready" signal from the APB address decoder is closed. Write accesses do not affect the system. Read accesses supply undefined data.

4.1 BOOT ROM

The ERTEC 400 is implemented with a BOOT ROM whose integrated opcode enables software to be downloaded from an external storage medium. Various routines are available for the different boot and download modes. In order to select the source and the mode, three BOOT[2:0] inputs are available on the ERTEC 400. During the active reset phase, the boot pins are read in and stored in the BOOT_REG register in the system control register area.

After startup of the processor, the system branches to the appropriate BOOT routine based on the coding and the download is performed. After the download is complete, the newly loaded functions are executed.

The following actions lead to a boot operation:

- HW reset
- Watchdog Reset
- Software reset caused by setting the **XRES_SOFT** bit in the reset control register (system control register area)

The following download modes are supported:

BOOT[2]	BOOT[1]	BOOT[0]	MEANING
0	0	0	External ROM/NOR Flash 8-bit data width
0	0	1	External ROM/NOR Flash 16-bit data width
0	1	0	External ROM/NOR Flash 32-bit data width
0	1	1	Reserved
1	0	0	Reserved
1	0	1	SPI (e.g. EEPROMS with ser. interface)
1	1	0	UART1 (bootstrap method)
1	1	1	PCI slave/LBU interface (from ext. host)

Table 8: Selection of Download Source

- Booting from NOR Flash or EEPROM with 8/16/32-bit data width via EMIF I/O Bank 0 (CSPER_0_N).
- Booting from serial EEPROMs/Flashes via the SPI interface. The GPIO[22] control cable is used as the chip select for the serial BOOT ROM. The storage medium is selected by means of the GPIO[23] control cable.
- Booting from PCI slave interface or a host processor system via the LBU bus.
- Booting from UART1. With the bootstrap method, a routine for operation of the serial interface is first downloaded. This routine then controls the actual program download.

During the boot operation, the address area of the communication RAM from 0x10100000 to 0x1010102F is reserved for the boot sequence. During the boot sequence, the IRT switch can only be assigned in the area 0x10101030 to 0x10102FFFF.

4.1.1 Booting from External ROM

This boot mode is provided for applications for which the majority of the user firmware runs on the ARM946E-S.

4.1.2 Booting via SPI

SPI-compatible EEPROMs as well as SPI-compatible Data Flash memories can be used as an SPI source. GPIO cable GPIO[23] is used to select the type.

- GPIO[23] = 0 → SPI-compatible Data Flash e.g., AT45DB011B
- GPIO[23] = 1 → SPI-compatible EEPROM e.g., AT25HP256

The serial protocols by Motorola, Texas Instruments, and NSC are supported in principle.

4.1.3 Booting via UART1

The UART interface is set at a fixed baud rate of 115200 baud during the boot operation. The boot loader performs the serial download of the second-level loader to the USER RAM. The USER RAM is mapped to address 0x00000000 and the second-level loader is started. The second-level loader downloads the user firmware to the various memory areas of the ERTEC 400 and starts the firmware once the download is complete.

4.1.4 Booting via PCI or LBU

Booting of user software via PCI must be actively performed from an external PCI master. For this purpose, the PCI slave macro is enabled during the boot operation in the ARM946E-S. This enables the user software to be loaded from the PCI master to the various memory areas of the ERTEC 400. At the end of the data transfer, the PCI master sets an identification bit in the SRAM in order to communicate to the ARM processor that the download is complete.

An external host can perform a boot via the LBU the same as via the PCI. The primary difference lies in the larger memory area available for selection via the PCI interface.

The PCI/LBU selection is made via the system control register CONFIG_REG.

4.2 General Purpose I/O (GPIO)

A maximum of 32 general purpose inputs/outputs are available in the ERTEC 400. After a reset, these are set as GPIO inputs.

GPIOs [31 : 7 : 0] are always available as I/O because no additional functions can be assigned.

GPIOs [30 : 8] have additional function features in the form of interfaces to watchdog, F-counter, UARTs, SPI, ETM, and MC PLL in the IRT macro.

The direction of the IO can be programmed bit-by-bit in the "GPIO_IOCTRL" register.

The special I/O function selection can be programmed in the "GPIO_PORT_MODE_L" and "GPIO_PORT_MODE_H" registers and the direction (input or output) in the "GPIO_IOCTRL" register.

GPIO inputs [1 : 0] and [31 : 30] can also be used as external interrupt inputs. They are connected at the IRQ interrupt controller of the ARM946. An interrupt can be generated only with an active High input level, rising edge, or falling edge (for parameter assignment, refer to Section 2.9.11).

The following figure shows the structure of a GPIO pin as a standard I/O function or as an alternative function.

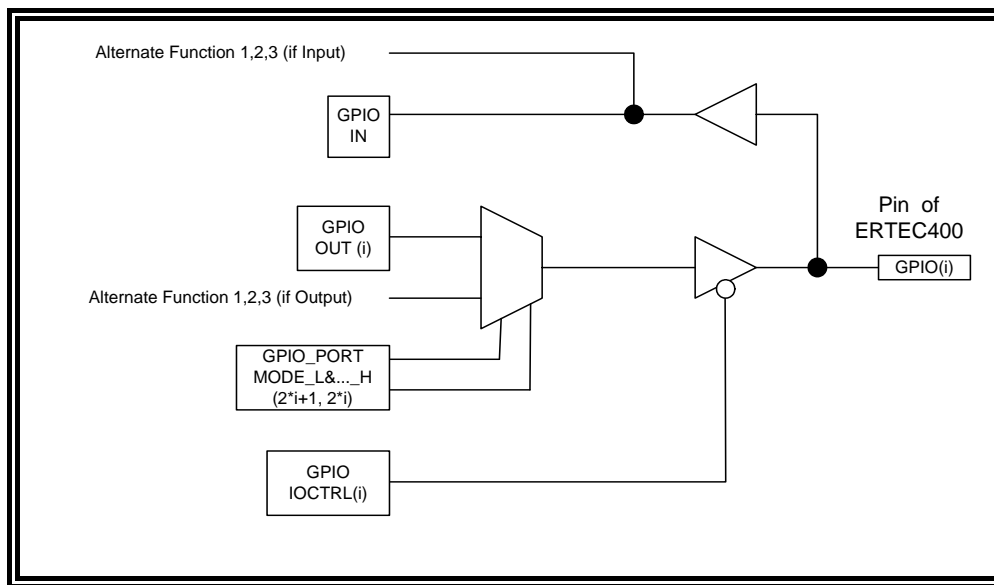


Figure 4: GPIO Cells of ERTEC 400

4.2.1 Address Assignment of GPIO Registers

The GPIO registers are 32 bits in width. The registers can be read or written to with 8-bit, 16-bit, or 32-bit accesses.

GPIO (Base Address 0x4000_2500)					
Register Name	Offset Address	Address Area	Access	Default	Description
GPIO_IOCTRL	0x0000	4 bytes	W/R	0xFFFFFFFF	Configuration register for GPIO
GPIO_OUT	0x0004	4 bytes	W/R	0x00000000	Output register for GPIO
GPIO_IN	0x0008	4 bytes	R	0x00000000	Input register for GPIO
GPIO_PORT_MODE_L	0x000C	4 bytes	W/R	0x00000000	Function assignment of GPIO port 0 to 15
GPIO_PORT_MODE_H	0x0010	4 bytes	W/R	0x00000000	Function assignment of GPIO port 16 to 31

Table 9: Overview of GPIO Registers

4.2.2 GPIO Register Description

GPIO_IOCTLL			W/R	Addr.: 0x4000_2500	Default: 0xFFFF_FFFF
Description		Configuration register for general purpose I/O (31:0)			
Bit No.	Name	Description			
31 - 0	GPIO_IOCTLL[31:0]	0 : GPIOx is output 1: GPIOx is input x = Bit 0 to 31			

GPIO_OUT			W/R	Addr.: 0x4000_2504	Default: 0x0000_0000
Description		Output register for general purpose I/O (31:0)			
Bit No.	Name	Description			
31..0	GPIO_OUT[31:0]	0: GPIO outputx = 0, 1: GPIO outputx = 1			

GPIO_IN			R	Addr.: 0x4000_2508	Default: Port assignment
Description		Input register for general purpose I/O (31:0)			
Bit No.	Name	Description			
31..0	GPIO_IN[31:0]	0: GPIO inputx = 0, 1: GPIO inputx = 1			

GPIO_PORT_MODE_L			W/R	Addr.: 0x4000_250C	Default: 0x0000_0000
Description		Configuration register for GPIO Port 0 to 15 Function assignment: 00 = Function 0; 01 = Function 1; 10 = Function 2; 11 = Function 3			
Bit No.	Name	Description			
1:0	GPIO0_PORT_MODE	Port GPIO(0);			
3:2	GPIO1_PORT_MODE	Port GPIO(1);			
5:4	GPIO2_PORT_MODE	Port GPIO(2);			
7:6	GPIO3_PORT_MODE	Port GPIO(3);			
9:8	GPIO4_PORT_MODE	Port GPIO(4);			
11:10	GPIO5_PORT_MODE	Port GPIO(5);			
13:12	GPIO6_PORT_MODE	Port GPIO(6);			
15:14	GPIO7_PORT_MODE	Port GPIO(7);			
17:16	GPIO8_PORT_MODE	Port GPIO(8); If the ETM9 module is activated via debug SW <u>and</u> it is set as a 4- to 8-bit wide ETM port, function 3 (ETM:TRACEPKT(0) is set regardless of the value of GPIO_PORT_MODE_L-bits (17:16). This applies analogously to GPIO_PORT_MODE_L-bits (23:18) and ETM pins TRACEPKT(3:1).			
19:18	GPIO9_PORT_MODE	Port GPIO(9); See note for bits(17:16)			
21:20	GPIO10_PORT_MODE	Port GPIO(10); See note for bits(17:16)			
23:22	GPIO11_PORT_MODE	Port GPIO(11); See note for bits(17:16)			
25:24	GPIO12_PORT_MODE	Port GPIO(12);			
27:26	GPIO13_PORT_MODE	Port GPIO(13);			
29:28	GPIO14_PORT_MODE	Port GPIO(14);			
31:30	GPIO15_PORT_MODE	Port GPIO(15);			

GPIO_PORT_MODE_H		W/R	Addr.: 0x4000_2510	Default: 0x0000_0000
Description		Configuration register for GPIO Port 16 to 31 Function assignment: 00 = Function 0; 01 = Function 1; 10 = Function 2; 11 = Function 3		
Bit No.	Name	Description		
1:0	GPIO16_PORT_MODE	Port GPIO(16);		
3:2	GPIO17_PORT_MODE	Port GPIO(17);		
5:4	GPIO18_PORT_MODE	Port GPIO(18);		
7:6	GPIO19_PORT_MODE	Port GPIO(19); If the ETM9 module is activated via debug SW <u>and</u> it is set as an 8-bit wide ETM port, function 3 (ETM:TRACEPKT(6)) is set regardless of the value of GPIO_PORT_MODE_H-bits (7:6). This applies analogously to GPIO_PORT_MODE_H-bits (13:8) and ETM pins TRACEPKT(7:5).		
9:8	GPIO20_PORT_MODE	Port GPIO(20); See note for bits(7:6)		
11:10	GPIO21_PORT_MODE	Port GPIO(21); See note for bits(7:6)		
13:12	GPIO22_PORT_MODE	Port GPIO(22); See note for bits(7:6)		
15:14	GPIO23_PORT_MODE	Port GPIO(23);		
17:16	GPIO24_PORT_MODE	Port GPIO(24);		
19:18	GPIO25_PORT_MODE	Port GPIO(25);		
21:20	GPIO26_PORT_MODE	Port GPIO(26);		
23:22	GPIO27_PORT_MODE	Port GPIO(27);		
25:24	GPIO28_PORT_MODE	Port GPIO(28);		
27:26	GPIO29_PORT_MODE	Port GPIO(29);		
29:28	GPIO30_PORT_MODE	Port GPIO(30);		
31:30	GPIO31_PORT_MODE	Port GPIO(31);		

4.3 Timer 0 and Timer 1

Two independent timers are integrated in the ERTEC 400. They can be used for internal monitoring of diverse software routines. Each timer has an interrupt output that is connected to the IRQ interrupt controller of the ARM946. Access to the timers is always 32 bits in width.

Both timers have the following functionality:

- 32-bit count register
- Input clock can be switched to:
 - 50 MHz clock (default setting)
 - 8-bit prescaler per timer (can be assigned separately)
- Down-counting
- Load/reload function
- Start, stop and continue functions
- Interrupt when counter state 0 is reached
- Count register can be read/write-accessed

4.3.1 Mode of Operation of Timers

Both timers are deactivated after a reset. The timers are enabled by setting the "RUN/XStop" bit in the status/control register of the respective timer. The timer then counts downwards from its loaded 32-bit starting value. When the timer value reaches 0, a timer interrupt is generated. The interrupt can then be evaluated by the IRQ interrupt controller.

If Reload mode = 0, the timer stops.

If Reload mode = 1, the timer is reloaded with the 32-bit reload value and automatically restarted.

The timer can also be reloaded with the reload value during normal timer function (count value <>0). This occurs by setting the "LOAD" bit in the status/control register of the timer.

Normally, the timer clock operates at 50 MHz, which is generated by the internal PLL. Each timer can also be operated with an 8-bit prescaler. This can be used to increase the timer time accordingly.

4.3.2 Timer Interrupts

The timer interrupt is active (High) starting from the point at which the timer value is counted down to 0.

The timer interrupt is deactivated (Low) when the reload value is automatically reloaded or the "LOAD" bit is set by the user. The interrupt is not reset if the loaded reload value is 0. If the timer is deactivated (Run/XStop = 0), the interrupt is also deactivated.

If the timer operates in Reload mode without a prescaler, the interrupt is present for only one 50 MHz cycle. This must be taken into account when assigning the relevant interrupt input (level/edge evaluation).

4.3.3 Timer Prescaler

An 8-bit prescaler is available for each timer. Settings can be made independently for each prescaler. Each prescaler has its own 8-bit reload register. If the reload value or starting value of the prescaler is 0, prescaling does not occur. The current prescaler value cannot be read out. In addition, there are no status bits for the prescalers. The prescalers always run in Reload mode.

4.3.4 Cascading of Timers

If the "Cascading" bit is set, both timers can be cascaded to form one 64-bit timer.

The cascaded timer is enabled via the status/control register of Timer 1. The interrupt of Timer 1 is active. The interrupt of Timer 0 must be disabled when the timers are cascaded. When prescalers are specified, the prescaler of Timer 1 is used.

The user must provide for data consistency in the user software when reading out the 64-bit timer.

4.3.5 Address Assignment of Timer 0/1 Registers

The timer registers are 32 bits in width. For read/write access of the timer registers to be meaningful, a 32-bit access is required. However, an 8-bit or 16-bit access is not intercepted by the hardware.

Timer (Base Address 0x4000_2000)					
Register Name	Offset Address	Address Area	Access	Default	Description
CTRL_STAT0	0x0000	4 bytes	R/W	0x00000000	Control/status register timer 0
CTRL_STAT1	0x0004	4 bytes	R/W	0x00000000	Control/status register timer 1
RELD0	0x0008	4 bytes	R/W	0x00000000	Reload register timer 0
RELD1	0x000C	4 bytes	R/W	0x00000000	Reload register timer 1
CTRL_PREDIV	0x0010	4 bytes	R/W	0x00000000	Control register for both prescalers
RELD_PREDIV	0x0014	4 bytes	R/W	0x00000000	Reload register for both prescalers
TIM0	0x0018	4 bytes	R	0x00000000	Timer 0 value register
TIM1	0x001C	4 bytes	R	0x00000000	Timer 1 value register

Table 10: Overview of Timer Registers

4.3.6 Timer 0/1 Register Description

CTRL_STAT0		R/W	Addr.: 0x4000_2000	Default: 0x0000_0000
Description		Control/status register 0. Configuration and control bits for Timer No. 0 .		
Bit No.	Name	Description		
0	Run/xStop *)	Stop/start of timer: 0: Timer is stopped 1: Timer is running Note: If this bit = 0, the timer interrupt is inactive (0) and the status bit (Bit 5) is reset (0).		
1	Load	Trigger=Load the timer with the reload register value: 0: Not relevant 1: Timer is loaded with the value of the reload register (irrespective of Bit 0=Run/xStop) While this bit can be read back, it only has an effect at the instant of writing. Writing a value of 1 to this bit is sufficient to trigger the timer; a 0/1 edge is not needed.		
2	Reload mode *)	Reload mode (continuous mode) of the timer. 0: Timer stops at value 00000000h 1: Timer is loaded with the reload register value when the timer value is 00000000h and the timer continues running Important note: If timers 0 and 1 are cascaded, the Reload mode setting of Timer 0 is irrelevant.		
3	Reserved	Not relevant (can be read/write-accessed)		
4	Reserved	Not relevant (read=0)		
5	Status	Timer status (writing is ignored) 0: Timer has not expired 1: Timer has expired (count is 0 and Run/xStop=Bit 0=1) Note: This bit can only be read as 1 if Run/xStop (Bit 0) is active (1).		
31-6	Reserved	Not relevant (read=0)		

Important note: The bits designated with *) are not applicable if the timers are cascaded! See CTRL_STAT1

CTRL_STAT1		R/W	Addr.: 0x4000_2004	Default: 0x0000_0000
Description		Control/status register 1. Configuration and control bits for Timer No. 1.		
Bit No.	Name	Description		
0	Run/xStop *)	Stop/start of timer: 0: Timer is stopped 1: Timer is running Note: If this bit = 0, the timer interrupt is inactive (0) and the status bit (Bit 5) is reset (0).		
1	Load	Trigger=Load the timer with the reload register value: 0: Not relevant 1: Timer is loaded with the value of the reload register (irrespective of Bit 0=Run/xStop) While this bit can be read back, it only has an effect at the instant of writing. Writing a value of 1 to this bit is sufficient to trigger the timer; a 0/1 edge is not needed.		
2	Reload mode *)	Reload mode (continuous mode) of the timer. 0: Timer stops at value 00000000h 1: Timer is loaded with the reload register value when the timer value is 00000000h and the timer continues running		
3	Reserved	Not relevant (can be read/write-accessed)		
4	Reserved	Not relevant (read=0)		

5	Status	Timer status (writing is ignored) 0: Timer has not expired 1: Timer has expired (count is 0 and Run/xStop=Bit 0=1) Note: This bit can only be read as 1 if Run/xStop (Bit 0) is active (1).
6	Cascading	Cascading of timer 0: Not relevant 1: Cascading of timers 0 and 1
31-7	Reserved	Not relevant (read=0)

Important note: The bits designated with *) are relevant to Timer 0 as well if the timers are cascaded!

RELD0	R/W	Addr.: 0x4000_2008	Default: 0x0000_0000
RELD1	R/W	Addr.: 0x4000_200C	Default: 0x0000_0000
Description	Reload registers 0 to 1. Reload value for timers 0 to 1.		
Bit No.	Name	Description	
31:0	Reload (31:0)	Reload value of timer	

CTRL_PREDIV	R/W	Addr.: 0x4000_2010	Default: 0x0000_0000
Description	Control register for the two prescalers		
Bit No.	Name	Description	
0	Run/xStop_V0	Stop/start of prescaler 0: 0: Prescaler 0 is stopped 1: Prescaler 0 is running	
1	Load_V0	Trigger = loading of prescaler 0 with the reload register value: 0: Not relevant 1: Timer is loaded with the value of the reload register While this bit can be read back, the trigger only has an effect at the instant of writing. The prescaler is loaded independently of the status of Run/xStop_V0.	
2	Run/xStop_V1	Stop/start of prescaler 1: 0: Prescaler 1 is stopped 1: Prescaler 1 is running	
3	Load_V1	Trigger = loading of prescaler 1 with the reload register value: 0: Not relevant 1: Timer is loaded with the value of the reload register While this bit can be read back, the trigger only has an effect at the instant of writing. The prescaler is loaded independently of the status of Run/xStop_V1.	
31-4	Reserved	Not relevant (read=0)	

Note regarding prescalers: The current counter value of the prescalers cannot be read. In addition, there are no status bits for the prescalers indicating when the counter state is 0. The prescalers always run cyclically (in Reload mode).

RELD_PREDIV	R/W	Addr.: 0x4000_2014	Default: 0x0000_0000
Description	Reload register for the two prescalers		
Bit No.	Name	Description	
7:0	Prediv (7:0)	Reload value of Prescaler 0	
15:8	Prediv (15:8)	Reload value of Prescaler 1	
31-16	Reserved	Not relevant (read=0)	

TIM0	R	Addr.: 0x4000_2018	Default: 0x0000_0000
TIM1	R	Addr.: 0x4000_201C	Default: 0x0000_0000
Description	Timer registers 0 to 1. Values of timers 0 to 1.		
Bit No.	Name	Description	
31:0	Timer (31:0)	Value of timer	

4.4 F - Counter

An F-counter is integrated in the ERTEC 400 in addition to the system timers. This counter works independently of the system clock and can be used for fail-safe applications, for example. The F-counter is triggered via the alternative "F_CLK" function at the external "BYP_CLK" input. External triggering is not possible if the ARM946E-S is operated in a reserved test mode (Config[4:3] = 11).

The following signal pins are available for the F-counter on the ERTEC 400.

- External counter cable 1 **F_CLK**

Function description:

The asynchronous input signal of the external independent time base is applied at a synchronization stage via the BYP_CLK input pin (alternative F_CLK function). To rule out occurrences of metastable states at the counter input, the synchronization stage is implemented with three flip-flop stages. The count pulses are generated in a series-connected edge detection. All flip-flops run at the APB clock of 50 MHz.

The F_COUNTER_VAL register is reset using an asynchronous block reset or by writing the value 0x XXXX 55AA (X means "don't care") to the F-counter register "FCOUNT_RES". The next count pulse sets the counter to 0xFFFF FFFF and the counter is decremented at each additional count pulse. The FCOUNT_RES register is cleared again at the next clock cycle.

The count value can be read out by a 32-bit read access. While an 8-bit or 16-bit read access is possible, it is not useful because it can result in an inconsistency in the read count values.

Note on input frequency:

The maximum input frequency for the F-CLK is one-quarter of the APB clock. In the event of a quartz failure on the ERTEC 400, a minimum output frequency between 40 and 90 MHz is set at the PLL. This yields a minimum APB-CLK frequency of $PLLOUT_{min} 40 \text{ MHz} / 6 = 6.6666 \text{ MHz}$. To rule out a malfunction in the edge evaluation, the F-CLK can not exceed $APB-CLK_{min} 6.66 \text{ MHz} / 4 = 1.6666 \text{ MHz}$.

The figure below shows the function blocks of the F-counter.

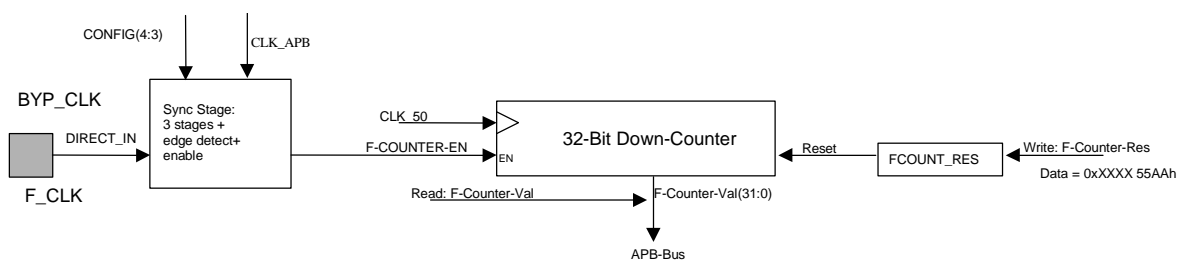


Figure 5: Block Diagram of F-Counter

4.4.1 Address Assignment of F-Timer Registers

The F-timer registers are **32 bits in width**. The registers can be read or written to with 32-bit accesses only.

F-Counter (Base Address 0x4000_2700)					
Register Name	Offset Address	Address Area	Access	Default	Description
F-COUNTER-VAL	0x0000	4 bytes	R	0x00000000	F-counter value register
F-COUNTER-RES	0x0004	4 bytes	W	0x00000000	Reset register for F-counter

Table 11: Overview of F-Timer Registers

4.4.2 F-Timer Register Description

F-COUNTER-VAL		R	Addr.: 0x4000_2700	Default: 0x0000_0000
Description		Timer value of F-counter		
Bit No.	Name	Description		
31:0	F-CNT-VAL (31:0)	Timer value of F-timer		

F-COUNTER-RES		W	Addr.: 0x4000_2704	Default: 0x0000_0000
Description		Reset register for F-counters. A reset of the F-counter is performed only if 0xFFFF55AAh is entered in this register. Resets are thus possible via 16-bit and 32-bit accesses.		
Bit No.	Name	Description		
31:16	F-CNT-RES (31:16)	More significant word of F-counter reset (any value)		
15:0	F-CNT-RES (15:0)	Less significant word of F-counter reset		

4.5 Watchdog Timers

Two watchdog timers are integrated in the ERTEC 400. The watchdog timers are intended for stand-alone monitoring of processes. The working clock of 50 MHz is derived from the PLL the same as the processor clock.

4.5.1 Watchdog Timer 0

Watchdog timer 0 is a 32-bit down-counter to which the WDOOUT0_N output is assigned. This output can be used at the GPIO[15]-pin as an alternative function (see GPIO and signal descriptions). The timer is locked after a reset. It is started by setting the "Run/XStop_Z0" bit in the "CTRL/STATUS" watchdog register. A maximum monitoring time of 85.89 s (at a resolution of 20 ns) can be assigned.

4.5.2 Watchdog Timer 1

Watchdog timer 1 is a 36-bit down-counter in which only the upper 32 bits can be programmed. The WDOOUT1_N output is assigned to watchdog timer 1. This output is not routed to the outside. Rather, it triggers a hardware reset internally. The timer is locked after a reset. It is started by setting the "Run/XStop_Z1" bit in the "CTRL/STATUS" watchdog register. A maximum monitoring time of 1374.3 s (at a resolution of 320 ns) can be assigned.

When the "LOAD" bit is set in the "CTRL/STATUS" watchdog register, both watchdog timers are reloaded with the applicable reload values of their reload registers. In the case of watchdog timer 1, bits 35:4 are loaded with the reload value. Bits 3:0 are set to 0.

The count values of the watchdog timers can also be read. When watchdog timer 1 is read, bits 35:4 are read out. The status of the two watchdog timers can be read out in the "CTRL/STATUS" register.

4.5.3 Watchdog Interrupt

The WDINT interrupt of the watchdog is routed to the FIQ interrupt controller. The interrupt is only active (High) if watchdog timer 0 is in "RUN mode" and watchdog timer 0 has reached zero. The exception to this is a load operation with reload value = 0.

4.5.4 WDOOUT0_N

The WDOOUT0_N output is at Low after a reset. If watchdog timer 0 is set in RUN mode and the timer value does not equal zero, the output changes to High. The output changes to Low again when the count has reached zero. The output can also be reset by stopping and then restarting watchdog timer 0.

The signal can be used as an external output signal at the GPIO[15] port if the alternative function is assigned for this pin. The output can thus inform an external host about an imminent watchdog event.

4.5.5 WDOOUT1_N

The WDOOUT1_N signal is at High after a reset or when watchdog timer 1 goes to Stop. If watchdog timer 1 is started, WDOOUT1_N changes to Low when the timer reaches zero. It remains Low until watchdog timer 1 is loaded with the reset value again by setting the "LOAD" bit. The exception is when reload value = 0 is loaded. A hardware reset is triggered internally with WDOOUT1_N.

The figure below shows the time sequence of the watchdog interrupt and the two watchdog signals:

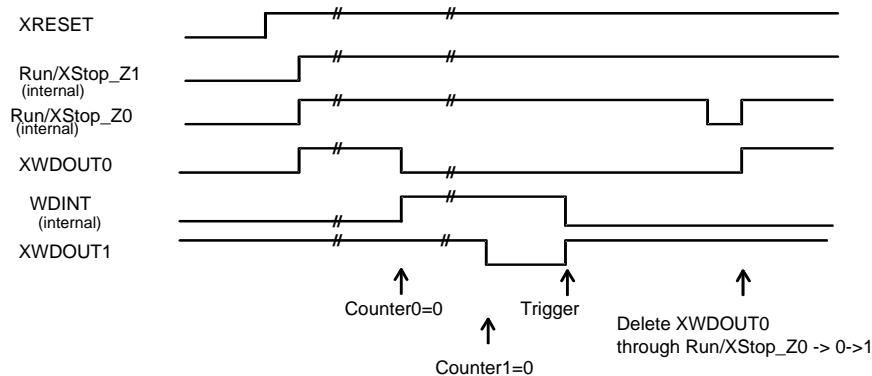


Figure 6: Watchdog Timing

4.5.6 Watchdog Registers

The watchdog registers are 32 bits in width. For read/write access of the watchdog registers to be meaningful, a 32-bit access is required. However, a byte-by-byte write operation is not intercepted by the hardware. To prevent the watchdog registers from being written to inadvertently, e.g., in the event of an undefined computer crash, the writable watchdog registers are provided with write protection. The upper 16 bits of the registers are so-called **key bits**. In order to write a valid value in the lower 16 bits, the key bits must be set to 0x**9876** yyyy (yyyy is the 16-bit value to be written).

4.5.7 Address Assignment of Watchdog Registers

Watchdog (Base Address 0x4000_2100)					
Register Name	Offset Address	Address Area	Access	Default	Description
CTRL/STATUS	0x0000	4 bytes	R/W	0x00000000	Control/status register WD
RELD0_LOW	0x0004	4 bytes	R/W	0x0000FFFF	Reload register 0_Low Bits 0-15
RELD0_HIGH	0x0008	4 bytes	R/W	0x0000FFFF	Reload register 0_High Bits 16-31
RELD1_LOW	0x000C	4 bytes	R/W	0x0000FFFF	Reload register 1_Low Bits 4-19
RELD1_HIGH	0x0010	4 bytes	R/W	0x0000FFFF	Reload register 1_High Bits 20-35
WDOG0	0x0014	4 bytes	R	0xFFFFFFFF	Watchdog timer 0 value register
WDOG1	0x0018	4 bytes	R	0xFFFFFFFF	Watchdog timer 1 value register

Table 12: Overview of WD Registers

4.5.8 Watchdog Register Description

CTRL/STATUS		R/W	Addr.: 0x4000_2100	Default: 0x0000_0000
Description	Control/status register Configuration and control bits for the watchdog.			
Bit No.	Name	Description		
0	Run/xStop_V0	Enable/disable watchdog counter 0: 0: Watchdog counter 0 disabled 1: Watchdog counter 0 enabled Note: If this bit = 0, the WDOOUT0_n output of the ERTEC 400 is active (0), the interrupt of the watchdog (WDINT) is "0", and the status bit of counter 0 (Bit 3) is "0".		
1	Run/xStop_Z1	Enable/disable watchdog counter 1: 0: Watchdog counter 1 disabled 1: Watchdog counter 1 enabled Note: If this bit = 0, the WDOOUT1_N output of the ERTEC 400 is passive (1) and the status bit of counter 1 (Bit 4) is "0".		
2	Load(Trigger)	Watchdog trigger (load watchdog counters 0 and 1 with the value of the reload registers): 0: Do not trigger watchdog 1: Trigger watchdog While this bit can be read back, it only has an effect at the instant of writing. Writing a value of 1 to this bit is sufficient to trigger the watchdog counter; a 0/1 edge is not needed. The trigger signal acts on both watchdog counters.		
3	Status_Counter 0	Watchdog status counter 0 (writing is ignored): 0: Watchdog counter 0 has not expired 1: Watchdog counter 0 has expired Note: This bit can only be read as '1' if Run/xStop_Z0 is active (1).		
4	Status_Counter 1	Watchdog status counter 1 (writing is ignored): 0: Watchdog counter 1 has not expired 1: Watchdog counter 1 has expired Note: This bit can only be read as '1' if Run/xStop_Z1 is active (1).		
15-5	Reserved	Not relevant (read=0)		
31-16	Key bits	Key bits for writing to this register (read=0). If bits 31-16=9876h, writing of bits 0-4 of this register has an effect; otherwise, no effect.		

RELD0_LOW			R/W	Addr.: 0x4000_2104	Default: 0x0000_FFFF
Description		Reload register 0_Low. Reload value for bits 15:0 of watchdog counter 0.			
Bit No.	Name	Description			
15-0	Reload0 (15:0)	Reload value for bits 15:0 of watchdog counter 0.			
31-16	Key bits	Key bits for writing to this register (read=0). If bits 31-16=9876h, writing of bits 0-15 of this register has an effect; otherwise, no effect.			

RELD0_HIGH			R/W	Addr.: 0x4000_2108	Default: 0x0000_FFFF
Description		Reload register 0_High. Reload value for bits 31:16 of watchdog counter 0.			
Bit No.	Name	Description			
15-0	Reload0 (31:16)	Reload value for bits 31-16 of watchdog counter 0.			
31-16	Key bits	Key bits for writing to this register (read=0). If bits 31-16=9876h, writing of bits 0-15 of this register has an effect; otherwise, no effect.			

RELD1_LOW			R/W	Addr.: 0x4000_210C	Default: 0x0000_FFFF
Description		Reload register 1_Low. Reload value for bits 19:4 of watchdog counter 1.			
Bit No.	Name	Description			
15-0	Reload1 (19:4)	Reload value for bits 19:4 of watchdog counter 1.			
31-16	Key bits	Key bits for writing to this register (read=0). If bits 31-16=9876h, writing of bits 0-15 of this register has an effect; otherwise, no effect.			

RELD1_HIGH			R/W	Addr.: 0x4000_2110	Default: 0x0000_FFFF
Description		Reload register 1_High. Reload value for bits 35:20 of watchdog counter 1.			
Bit No.	Name	Description			
15-0	Reload1 (35:20)	Reload value for bits 35-20 of watchdog counter 1.			
31-16	Key bits	Key bits for writing to this register (read=0). If bits 31-16=9876h, writing of bits 0-15 of this register has an effect; otherwise, no effect.			

WDOG0			R	Addr.: 0x4000_2114	Default: 0xFFFF_FFFF
Description		Watchdog value 0. Value of watchdog counter 0.			
Bit No.	Name	Description			
31-0	WDOG0(31:0)	Bit (31:0) of watchdog counter 0.			

WDOG1			R	Addr.: 0x4000_2118	Default: 0xFFFF_FFFF
Description		Watchdog value 1. Value of watchdog counter 1.			
Bit No.	Name	Description			
31-0	WDOG1(36:4)	Bit (36:4) of watchdog counter 1.			

4.6 UART1/ UART2

Two UARTs are implemented in the ERTEC 400. The inputs and outputs of the UARTs are available as an alternative function at GPIO port [13:9] (UART1) and GPIO port [18:14] (UART2). For this purpose, the I/O must be assigned to the relevant inputs and outputs and the alternative function must be assigned (see [GPIO register description](#)). If the UARTs are used, the pins are no longer available as standard I/O. The baud rate generation is derived from the internal 50 MHz APB clock. The data bit width for read/write access on the APB bus is 8 bits.

The following signal pins are available for UART 1 and UART 2 on the ERTEC 400.

- | | | |
|------------------|------------|----------------------------------------------------------------------|
| • Transmit cable | 1 per UART | TXD1/TXD2 |
| • Receive cable | 1 per UART | RXD1/ RXD2 |
| • Handshake | 3 per UART | DCD1_N/DCD2_N
CTS1_N/CTS2_N
DSR1_N/DSR2_N |

Both UARTs are implemented as ARM Prime Cell™ (PL010) macros. These are similar to standard UART 16C550. For a detailed description, refer to /5/. The figure below shows the structure of the UART.

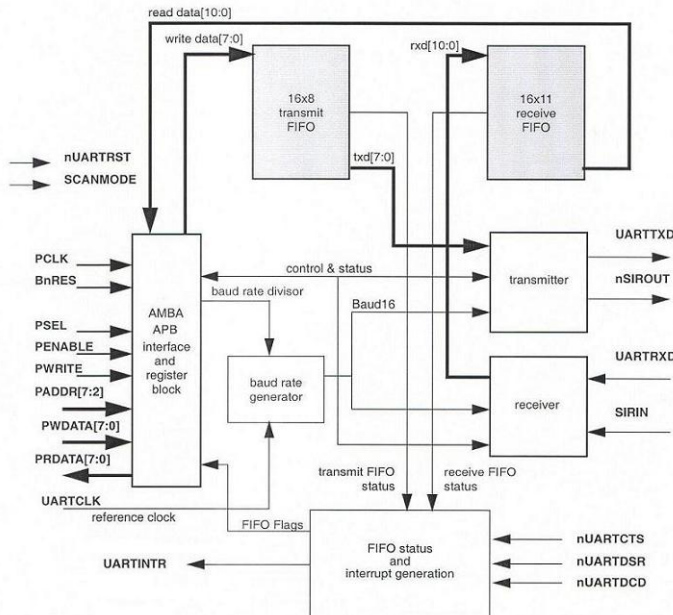


Figure 7: Block Diagram of UART

The UARTs differ from standard UART 16C550 as follows:

- Receive FIFO trigger level is set permanently to 8 bytes.
- Receive errors are stored in the FIFO.
- Receive errors do **not** generate an interrupt.
- The internal register address mapping and the register bit functions are different.

The following standard UART 16C550 features are not supported:

- 1.5 Stop bits
- “Forcing stick parity” function

Each UART has an interrupt source:

- UARTINTR 1 UART1 – group interrupt
- UARTINTR 2 UART2 – group interrupt

Both interrupts are available on the IRQ interrupt controller of the ARM946E-S.

The baud rate is calculated according to the following formula:

$$BR = \frac{F_{UARTCLK}}{(BAUDDIV+1) \times 16} \quad \text{or} \quad BAUDDIV = \left(\frac{F_{UARTCLK}}{BR \times 16} \right) - 1$$

This yields the following error tolerance calculation:

$$E_p = \frac{(BR - BRI)}{BRI} \times 100\% \quad \text{where BRI is the ideal baud rate}$$

The following table shows the baud rate values to be set and the deviations from the standard baud rates. The associated error percentages are within the baud rate tolerance range.

BRI	BAUDDIV	BR	E _p %
115200	26	115740	+0.47
76800	40	76219	-0.76
57600	53	57870	+0.47
38400	80	38580	+0.47
19200	162	19171	-0.15
14400	216	14400.9	+0.006
9600	325	9585.9	-0.15
2400	1301	2400.15	+0.006
1200	2603	1200.077	+0.006
110	28408	110.0004	+0.0003

Table 13: Baud Rates for UART at F_{UARTCLK}=50 MHz

UART 1 can also be used as a BOOT medium if, for example, functions from an external PC are to be loaded to the ERTEC 400 and executed. The BOOT medium is selected by the BOOT[2:0] inputs during the active reset phase.

The BOOT loader then takes over setting of the UART 1 signal pins and loading of the program code. The "Boot strap loader" functionality is also used.

If the user does not utilize UART 1, it can also be used as a debugging interface.

4.6.1 Address Assignment of UART 1/2 Registers

The UART registers are 8 bits in width.

UART1 (Base Address 0x4000_2300)					
UART2 (Base Address 0x4000_2400)					
Register Name	Offset Address	Address Area	Access	Default	Description
UARTDR	0x0000	1 byte	R/W	0x--	Read/write data from interface
UARTRSR/UARTECR	0x0004	1 byte	R/W	0x00	Receive status register (read) Error clear register (write)
UARTLCR_H	0x0008	1 byte	R/W	0x00	Line control register high byte
UARTLCR_M	0x000C	1 byte	R/W	0x00	Line control register middle byte
UARTLCR_L	0x0010	1 byte	R/W	0x00	Line control register low byte
UARTCR	0x0014	1 byte	R/W	0x00	Control register
UARTFR	0x0018	1 byte	R	0x9-	Flag register
UARTIIR/UARTICR	0x001C	1 byte	R/W	0x00	Int identification register (read) Interrupt clear register (write)
UARTILPR	0x0020	1 byte	R/W	0x00	IrDA low power counter register (not supported in ERTEC400)
	0x0024 - 0x003C				Reserved
	0x0040 - 0x0098				Reserved for test purposes
	0x009C - 0x00FF				Reserved for future extension

Table 14: Overview of UART 1/2 Registers

4.6.2 UART 1/2 Register Description

UARTDR (1)	R/W	Addr.: 0x4000_2300	Default: 0x--
UARTDR (2)	R/W	Addr.: 0x4000_2400	Default: 0x--
Description	UART data registers		
Bit No.	Name	Description	
7 - 0	-----	WRITE: - If FIFO is enabled, the written data are entered in the FIFO. - If FIFO is disabled, the written data are entered in the Transmit holding register (the first word in the Transmit FIFO). READ: - If FIFO is enabled, the received data are entered in the FIFO. - If FIFO is disabled, the received data are entered in the Receive holding register (the first word in the RECEIVE FIFO).	

NOTE: When data are received, the UARTDR data register must be read out first and then the UARTRSR error register.

UARTRSR/UARTECR (1)	R/W	Addr.: 0x4000_2304	Default: 0x00
UARTRSR/UARTECR (2)	R/W	Addr.: 0x4000_2404	Default: 0x00
Description	UART receive status register (read) UART receive error clear register (write)		
Bit No.	Name	Description	
7 - 0	----- (Write)	Framing errors, parity errors, break errors, and overrun errors are deleted.	
0	FE (Read)	Framing error = 1 Received character does not have a valid stop bit	
1	PE (Read)	Parity error = 1 Parity of received character does not match the assigned parity in the UARTLCR_H register Bit 2.	

2	BE (Read)	Break error = 1 A break was detected. A break means that the received data are at LOW for longer than a standard character with all control bits.
3	OE (Read)	Overrun-Error = 1 If the FIFO is full and a new character is received.
7 - 4	----- (Read)	Reserved Value is undefined

NOTE: When new data are displayed, the UARTDR data register must be read out first and then the UARTRSR error register. The error register is not updated until the data register is read.

UARTLCR_H (1)		R/W	Addr.: 0x4000_2308	Default: 0x00
UARTLCR_H (2)		R/W	Addr.: 0x4000_2408	Default: 0x00
Description	UART line control register high byte bit rate and control register bits 22 to 16			
Bit No.	Name	Description		
0	BRK	Send break = 1 A LOW level is sent continuously at the Transmit output.		
1	PEN	Parity enable = 1 Parity check and generation are enabled.		
2	EPS	If PEN = 1 Even parity select = 1 Even parity (1) for check and generation. Even parity select = 0 Odd parity (0) for check and generation.		
3	STP2	Two stop bit select = 1 Two stop bits are appended at the end of the frame when sending. Two stop bit select = 0 One stop bit is appended at the end of the frame when sending.		
4	FEN	FIFO enable = 1 FIFO modes for sending and receiving are enabled. FIFO enable = 0 FIFO is disabled. Sending/receiving is then performed via 1-byte holding registers.		
6 - 5	WLEN	Word length indicates the number of data bits within a frame. 00 5-bit data 01 6-bit data 10 7-bit data 11 8-bit data		
7	-----	Reserved Value is undefined		

UARTLCR_M (1)		R/W	Addr.: 0x4000_230C	Default: 0x00
UARTLCR_M (2)		R/W	Addr.: 0x4000_240C	Default: 0x00
Description	UART line control register middle byte baud rate high byte bits 15 - 8			
Bit No.	Name	Description		
7 - 0	BAUD DIVMS	Baud rate divisor high byte		

UARTLCR_L (1)		R/W	Addr.: 0x4000_2310	Default: 0x00
UARTLCR_L (2)		R/W	Addr.: 0x4000_2410	Default: 0x00
Description	UART line control register low byte baud rate low byte bits 7 - 0			
Bit No.	Name	Description		
7 - 0	BAUD DIVLS	Baud rate divisor low byte		

NOTE: The baud rate divisor is calculated according to the following formula:

$$\text{BAUDDIV} = \frac{\text{FUARTCLK}}{16 * \text{baud rate}} - 1$$

Zero is not a valid divisor.

UARTLCR consists of 3 bytes. Writing of bytes is complete when UARTLCR_H has been written. If one of the first two bytes is to be changed, UARTLCR_H must be written at the end following the change.
 Example: Write UARTLCR_L and/or UARTLCR_M, write UARTLCR_H as acceptance.
 Write UARTLCR_H only means write and accept UARTLCR_H bits.

UARTCR (1)		R/W	Addr.: 0x4000_2314	Default: 0x00
UARTCR (2)		R/W	Addr.: 0x4000_2414	Default: 0x00
Description		UART control registers		
Bit No.	Name	Description		
0	UARTEN	UART Enable = 1 UART Data can be transmitted/received		
1	SIREN	SIR enable = 1 IrDA SIR Endec is enabled. The bit can only be changed if UARTEN = 1		
2	SIRLP	IrDA SIR Low power mode		
3	MSIE	Modem status interrupt enable = 1 Interrupt is enabled		
4	RIE	Receive interrupt enable = 1 Receive interrupt is enabled		
5	TIE	Transmit interrupt enable = 1 Transmit interrupt is enabled		
6	RTIE	Receive timeout interrupt enable = 1 Receive timeout interrupt is enabled		
7	LBE	Loop back enable		

UARTFR (1)		R	Addr.: 0x4000_2318	Default: 0x9-
UARTFR (2)		R	Addr.: 0x4000_2418	Default: 0x9-
Description		UART flag registers		
Bit No.	Name	Description		
0	CTS	<u>Clear To Send</u> This bit is the inverse signal of UART input CTS.		
1	DSR	<u>Data Set Ready</u> This bit is the inverse signal of UART input DSR.		
2	DCD	<u>Data Carrier Detect</u> This bit is the inverse signal of UART input DCD.		
3	BUSY	<u>UART Busy</u> The bit is set if send data are in progress or if the Transmit FIFO is not empty.		
4	RXFE	<u>Receive FIFO Empty</u> = 1 if <ul style="list-style-type: none"> ▪ FIFO is disabled and Receive holding register is empty ▪ FIFO is disabled and Receive FIFO buffer is empty 		
5	TXFF	<u>Transmit FIFO Full</u> = 1 if <ul style="list-style-type: none"> ▪ FIFO is disabled and Transmit holding register is full ▪ FIFO is enabled and Transmit FIFO buffer is full 		
6	RXFF	<u>Receive FIFO Full</u> = 1 if <ul style="list-style-type: none"> ▪ FIFO is disabled and Receive holding register is full ▪ FIFO is enabled and Receive FIFO buffer is full 		
7	TXFE	<u>Transmit FIFO Empty</u> = 1 if <ul style="list-style-type: none"> ▪ FIFO is disabled and Transmit holding register is empty ▪ FIFO is enabled and Transmit FIFO buffer is empty 		

UARTIIR/UARTICR (1)		R/W	Addr.: 0x4000_231C	Default: 0x00
UARTIIR/UARTICR (2)		R/W	Addr.: 0x4000_241C	Default: 0x00
Description		UART interrupt identification register (read) UART interrupt clear register (write)		
Bit No.	Name	Description		
0	MIS (Read)	<u>Modem Interrupt Status</u> This bit is set if UARTMSINTR is active.		
1	RIS (Read)	<u>Receive Interrupt Status</u> This bit is set if UARTRXINTR is active.		
2	TIS (Read)	<u>Transmit Interrupt Status</u> This bit is set if UARTRXINTR is active.		
3	RTIS (Read)	<u>Receive Timeout Interrupt Status</u> This bit is set if UARTRTINTR is active.		
7 - 4	----- (Read)	Reserved Value is undefined		
7 - 0	----- (Write)	Writing to this register deletes the MIS bit irrespective of the value written.		

UARTILPR (1)		R/W	Addr.: 0x4000_2320	Default: 0x00
UARTILPR (2)		R/W	Addr.: 0x4000_2420	Default: 0x00
Description		UART IrDA low power counter registers (not supported in ERTEC400)		
Bit No.	Name	Description		
7 - 0	ILPDVSR	8-bit low power divisor value		

NOTE: The low power divisor is calculated according to the following formula:

$$ILPDVSR = \frac{FUARTCLK}{F_{IrLPBAUD16}} - 1$$

F_{IrLPBAUD16} is nominally 1.8432 MHz

Zero is not a valid divisor.

4.7 Synchronous Interface SPI

An SPI interface is implemented in the ERTEC 400. The inputs and outputs of the SPI interface are available as an alternative function at GPIO port [23:16]. For this purpose, the I/O must be assigned to the relevant inputs and outputs and the alternative function must be assigned (see [GPIO register description](#)). If the SPI interface is used, the pins are no longer available as standard GPIO. The base frequency for the internal bit rate generation is the 50 MHz APB clock. The data bit width for read/write access is 16 bits.

The following signal pins are available for the SPI interface on the ERTEC 400.

- Transmit cable 1 **SSPTXD**
- Receive cable 1 **SSPRXD**
- Clock cable 2 **SCLKIN/ SCLKOUT**
- Enables 2 **SSPCTLOE/SSPOE**
- SFRs 2 **SFRMIN/SFRMOUT**

The SPI interface is implemented as ARM Prime Cell™ (PL021) macros. For a detailed description, refer to /6/. The figure below shows the structure of the SPI macro.

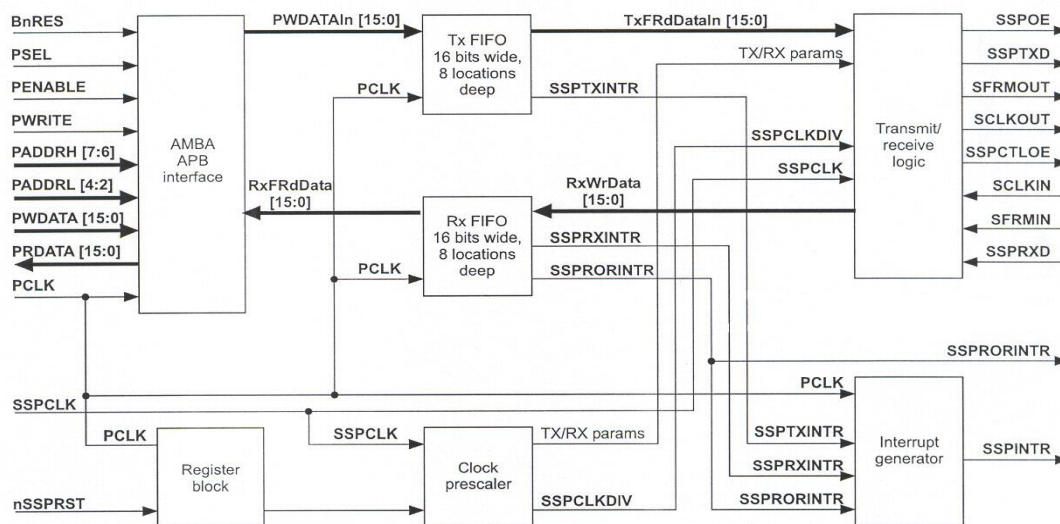


Figure 8: Block Diagram of SPI

The SPI interface supports the following modes:

- Motorola SPI-compatible mode
- Texas Instruments synchronous serial interface
- National Semiconductor microwire interface

The SPI interface has the following features:

- Separate send and receive FIFOs for 8 entries with 16-bit data width
- Data frame of 4 to 16 bits can be assigned
- The following bit rates can be assigned
 - 769 Hz to 25 MHz in master mode
 - Maximum of 4.16 MHz in slave mode

The SPI interface has the following interrupt sources:

- SSPINTR Group interrupt
- SSPPRORINTR Overrun error interrupt

Both interrupts are available on the IRQ interrupt controller of the ARM946E-S.

For the synchronous clock output of the SPI interface, the following frequencies are calculated according to the assigned SPI registers:

$$\text{SCLKOUT} = \frac{50 \text{ MHz}}{\text{CPSDRV} * (1 + \text{SCR})}$$

The SPI parameters can assume the following values:

CPSDRV From 2 to 254
 SCR From 0 to 255

This yields a frequency range of

- 769 Hz (CPSDRV = 254, SCR = 255) to
- 25 MHz (CPSDRV = 2, SCR = 0)

The SPI interface can also be used as a BOOT medium if, for example, functions from a serial EEPROM are to be loaded to the ERTEC 400 and executed. The BOOT medium is selected by the BOOT[2:0] inputs during the active reset phase. (See BOOT ROM description).

The BOOT loader then takes over setting of the SPI signal bins and loading of the program code. For BOOT mode with SPI interface, the GPIO[22] is used as a chip select signal.

4.7.1 Address Assignment of SPI Register

The SPI registers are **16 bits in width**. For read/write access of the SPI registers to be meaningful, a 16-bit access is required. However, a byte-by-byte write operation is not intercepted by the hardware.

SPI (Base Address 0x4000_2200)					
Register Name	Offset Address	Address Area	Access	Default	Description
SSPCR0	0x0000	2 bytes	R/W	0x0000	SSP control register 0
SSPCR1	0x0004	2 bytes	R/W	0x0000	SSP control register 1
SSPDR	0x0008	2 bytes	R/W	0x----	Rx/Tx FIFO data register
SSPSR	0x000C	2 bytes	R	0x0000	SSP status register
SSPCPSR	0x0010	2 bytes	R/W	0x0000	SSP clock prescale register
SSPIIR/SSPICR	0x0014	2 bytes	R/W	0x0000	Int identification register (read) Interrupt clear register (write)
	0x0018 - 0x003C				Reserved
	0x0040 - 0x0090				Reserved for test purposes
	0x0094 - 0x00FF				Reserved for future extension

Table 15: Overview of SPI Registers

4.7.2 SPI Register Description

SSPCR0			R/W	Addr.: 0x4000_2200	Default: 0x0000
Description		Control register 0. Configuration frame format and baud rate for SPI.			
Bit No.	Name	Description			
3 - 0	DSS Data Size Select	0000	Reserved (undefined)	1000	9-bit data
		0001	Reserved (undefined)	1001	10-bit data
		0010	Reserved (undefined)	1010	11-bit data
		0011	4-bit data	1011	12-bit data
		0100	5-bit data	1100	13-bit data
		0101	6-bit data	1101	14-bit data
		0110	7-bit data	1110	15-bit data
		0111	8-bit data	1111	16-bit data
5 - 4	FRF Frame Format	00	Motorola SPI frame format		
		01	TI synchronous serial frame format		
		02	National Microwire frame format		
		03	Reserved (undefined operation)		
6	SPO Serial Clock Output Polarity	Can only be used in Motorola SPI frame format. 0 Received bits are engaged on the rising edge of SCLKIN/OUT. Sent bits are switched on the falling edge of SCLKIN/OUT. 1 Received bits are engaged on the falling edge of SCLKIN/OUT. Sent bits are switched on the rising edge of SCLKIN/OUT.			
7	SPH Phase of Transmission Bit	Can only be used in Motorola SPI frame format. 0 Received MSB is expected after frame signal has gone to Low 1 Received MSB is expected ½ clock cycle after frame signal has gone to Low			
15-8	SCR Serial Clock Rate	The serial clock rate is taken for calculation of the Transmit/Receive bit rate. The calculation formula is as follows: F_{SPCLK} ----- $CPSDVSr \times (1 + SCR)$ SCR := 1 to 255 CPSDVSr := 2 to 254 (for a description, refer to SSPCSR Register)			

SSPCR1			R/W	Addr.: 0x4000_2204	Default: 0x0000
Description		Control register 1. Configuration frame format and baud rate for SPI.			
Bit No.	Name	Description			
0	RIE	Receive FIFO interrupt enable: 0 = Receive FIFO half full or more interrupt SSPRXINTR is disabled 1 = Receive FIFO half full or more interrupt SSPRXINTR is enabled			
1	TIE	Transmit FIFO interrupt enable: 0 = Transmit FIFO half full or less interrupt SSPTXINTR is disabled 1 = Transmit FIFO half full or less interrupt SSPTXINTR is enabled			
2	RORIE	Receive FIFO overrun interrupt enable: 0 = FIFO overrun display interrupt SSPRORINTR is disabled (When this bit is deleted, the SSPRORINTR interrupt is also deleted if this interrupt was currently being enabled) 1 = FIFO overrun display interrupt SSPRORINTR is enabled			
3	LBM	Loop back mode 0 = Normal serial operation is active 1 = Loop back mode is active. (The output of the Transmit serial shifter is connected internally to the input of the Receive serial shifter.)			
4	SSE	Synchronous serial port enable: 0 = SPI port is disabled 1 = SPI port is enabled			
5	MS	Master/slave mode select (This bit can only be changed if Bit 4 SSE = 0) 0 = Device is master (default) 1 = Device is slave			

6	SOD	Slave mode output disable (This bit is only relevant in slave mode MS = 1) In "Multiple slave systems," the master can send a broadcast message to all slaves in the system in order to ensure that only one slave drives data at its Transmit output 0 = SPI can drive the SSPTXD output in slave mode 1 = SPI must not drive the SSPTXD output in slave mode
15-7	-----	Reserved Read: Value is undefined Write: Should always be written with zero

SSPDR		R/W	Addr.: 0x4000_2208	Default: 0x----
Description		SPI data register		
Bit No.	Name	Description		
15-0	DATA [15:0]	Transmit/Receive FIFO Read = Receive FIFO Write = Transmit FIFO (If < 16 bits of data, the user must write the data to the Transmit FIFO in the proper format. When data are read, they are read out correctly from the Receive FIFO.)		

SSPSR		R	Addr.: 0x4000_220C	Default: 0x0000
Description		SPI status register		
Bit No.	Name	Description		
0	TFE	Transmit FIFO empty 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty		
1	TNF	Transmit FIFO not full 0 = Transmit FIFO is full 1 = Transmit FIFO is not full		
2	RNE	Receive FIFO not empty 0 = Receive FIFO is empty 1 = Receive FIFO is not empty		
3	RFF	Receive FIFO full 0 = Receive FIFO is not full 1 = Receive FIFO is full		
4	BSY	SPI busy flag 0 = SPI is 1 = SPI is sending and/or receiving a frame or the Transmit FIFO is not empty.		
15-5	-----	Reserved Read: Value is undefined Write: Should always be written with zero		

SSPCPSR		R/W	Addr.: 0x4000_2210	Default: 0x0000
Description		SPI clock prescale register		
Bit No.	Name	Description		
7 - 0	CPSDVSR	Clock prescale divisor (Value between 2 and 254. For formula, refer to SSPCR0 Register.) When the value is read, bit 0 is always zero.		
15-5	-----	Reserved Read: Value is undefined Write: Should always be written with zero		

SSPIIR/SSPICR		R/W	Addr.: 0x4000_2214	Default: 0x0000
Description		SPI interrupt identification register (read) SPI interrupt clear register (write)		
Bit No.	Name	Description		
0	RIS (Read)	SPI Receive FIFO service request interrupt status 0 = SSPRXINTR is not active 1 = SSPRXINTR is active		
1	TIS (Read)	SPI Transmit FIFO service request interrupt status 0 = SSPTXINTR is not active 1 = SSPTXINTR is active		
2	RORIS (Read)	SPI Receive FIFO overrun interrupt status 0 = SSPRORINTR is not active 1 = SSPRORINTR is active		
15-3	----- (Read)	Read: Reserved - value is undefined		
15-0	----- (Write)	Write: Receive overrun interrupt is deleted without check to determine whether data are currently being written.		

4.8 System Control Register

The system control registers are ERTEC 400-specific control registers that can be read and written to from the PCI/LBU side or from the ARM946. A listing of all system control registers and their address assignments as well as a detailed description are included in the following sections.

4.8.1 Address Assignment of System Control Registers

The system control registers are **32 bits in width**.

System Control Registers (Base address 0x4000_2600)					
Register Name	Offset Address	Address Area	Access	Default	Description
ID_REG	0x0000	4 bytes	R	0x40260100	ID ERTEC 400
BOOT_REG	0x0004	4 bytes	R	0x----	Boot mode pins Boot[0:2]
CONFIG_REG	0x0008	4 bytes	R	0x----	ERTEC 400 config pins Config[0:4]
RES_CTRL_REG	0x000C	4 bytes	W/R	0x00000100	Control register for reset of ERTEC 400
RES_STAT_REG	0x0010	4 bytes	R	0x00000004	Status register for reset of ERTEC 400
PLL_STAT_REG	0x0014	4 bytes	R/W	0x00070005	Status register for PLL/FIQ3
CLK_CTRL_REG	0x0018	4 bytes	W/R	0x00000001	Control register for clock of ERTEC 400
PM_STATE_REQ_REG	0x001C	4 bytes	R	0x00000000	Required power state of the PCI host
PM_STATE_ACK_REG	0x0020	4 bytes	R/W	0x00000000	Current power state of ERTEC 400
PME_REG	0x0024	4 bytes	R/W	0x00000000	Power management event PME
QVZ_AHB_ADR	0x0028	4 bytes	R	0x00000000	Address of incorrect addressing on multilayer AHB
QVZ_AHB_CTRL	0x002C	4 bytes	R	0x00000000	Control signals of incorrect addressing on multilayer AHB
QVZ_AHB_M	0x0030	4 bytes	R	0x00000000	Master detection of incorrect addressing on multilayer AHB
QVZ_APB_ADR	0x0034	4 bytes	R	0x00000000	Address of incorrect addressing on AHB
QVZ_EMIF_ADR	0x0038	4 bytes	R	0x00000000	Address that leads to timeout on EMIF
PCI_RES_REQ	0x003C	4 bytes	R/W	0xFFFF0002	Request register for placing a SW reset request on the PCI bridge

PCI_RES_ACK	0x0040	4 bytes	R	0x00000000	ACK for display of an implemented SW reset request
MEM_SWAP	0x0044	4 bytes	R/W	0x00000000	Memory swapping in Segment 0 between ROM and RAM
PCI_INT_CTRL	0x0048	4 bytes	R/W	0x00000000	Control PCI interrupts
M_LOCK_CTRL	0x004C	4 bytes	R/W	0x00000000	AHB master lock enable. Master-selective enable of AHB lock functionality
ARM9_CTRL	0x0050	4 bytes	R/W	0x00001939	Controller of ARM9 and ETM inputs
ARM9_WE	0x0054	4 bytes	R/W	0x00000000	Write protection register for ARM9_CTRL
-----	0x0058	76 bytes			Reserved

Table 16: Overview of System Control Registers

4.8.2 System Control Register Description

ID_REG		R	Addr.: 0x4000_2600	Default: 0x4026_0100
Description	Identification of ERTEC 400			
Bit No.	Name	Description		
31..16	ERTEC400-ID	ERTEC 400 identifier: 4026h (corresponds to the device ID of the AHB-PCI bridge)		
15..8	HW-RELEASE	HW release: 01h		
7..0	METALL-FIX	Metal fix: 00h		

BOOT_REG		R	Addr.: 0x4000_2604	Default: Bootpins[2:0]
Description	Boot mode pins BOOT[2:0] can be read			
Bit No.	Name	Description		
31..3	----	Reserved		
2 .. 0	BOOT[2:0]	Reading of BOOT[2:0] pin		

CONFIG_REG		R	Addr.: 0x4000_2608	Default: Configpins[4:0]
Description	ERTEC 400 config pins CONFIG[4:0] can be read.			
Bit No.	Name	Description		
31..5	----	Reserved		
4 .. 0	CONFIG[4:0]	Reading of CONFIG[4:0] pin		

RES_CTRL_REG			W/R	Addr.: 0x4000_260C	Default: 0x0000_0100
Description		Control register for reset of ERTEC 400			
Bit No.	Name	Description			
31..10	----	Reserved			
9	XRES_PCI_STATE	Status of PCI reset (read only) 0 : PCI reset is active 1: PCI reset is inactive			
8	XRES_PCI_AHB_SOFT	SW reset for the AHB side of the PCI-AHB bridge. 0: Reset active 1: Reset inactive			
7:3	PULSE_DUR	Pulse duration of SW or watchdog reset. $T_{RES_PULSE} = (8 \times n + 8) \times T_{CLK}$; T_{CLK} : APB clock period (1/50 MHz = 20 ns) n: Value of PULSE_DUR (0 to 31)			
2	----	Reserved, must be assigned 0			
1	XRES_SOFT	1: Trigger software reset (not latching) 0: Software reset not active			
0	WD_RES_FREI	1: Enable watchdog reset			

RES_STAT_REG			R	Addr.: 0x4000_2610	Default: 0x0000_0004
Description		Status register for reset of ERTEC 400. Only the bit of the last reset event occurrence is set. The two other bits are reset.			
Bit No.	Name	Description			
31..3	----	Reserved			
2	HW_RESET	1: Last reset was a hardware reset			
1	SW_RESET	1: Last reset was via a software reset			
0	WD_RESET	1: Last reset was via a watchdog			

PLL_STAT_REG			R/W	Addr.: 0x4000_2614	Default: 0x0007_0005
Description		Status register for PLL of ERTEC 400 and interrupt control for FIQ3			
Bit No.	Name	Description			
31..19	----	Reserved			
18	INT_MASK_QVZ_PCI_SLAVE	Interrupt masking for INT_QVZ_PCI_SLAVE_STATE 0: Interrupt is enabled 1: Interrupt is masked Read/write accessible			
17	INT_MASK_LOSS	Interrupt masking for INT_LOSS_STATE 0: Interrupt is enabled 1: Interrupt is masked Read/write accessible			
16	INT_MASK_LOCK	Interrupt masking for INT_LOCK_STATE 0: Interrupt is enabled 1: Interrupt is masked Read/write accessible			
15..6	----	Reserved			
5	INT_QVZ_EMIF_STATE	Interrupt timeout on EMIF: 0: Interrupt request is inactive 1: Interrupt request is active Read access only; This bit represents the value of Bit 7 of EMIF register Extended_Config.			
4	INT_QVZ_PCI_STATE	Interrupt timeout on the PCI slave 0: Interrupt request is inactive 1: Interrupt request is active Read/write accessible			

3	INT_LOSS_STATE	Interrupt loss state: 0: Interrupt request is inactive 1: Interrupt request is active This bit indicates whether the PLL input clock has failed (latching). Read/write accessible
2	INT_LOCK_STATE	Interrupt lock state: 0: Interrupt request is inactive 1: Interrupt request is active This bit indicates whether the PLL was in unlocked state (latching). Read/write accessible
1	PLL_INPUT_CLK_LOSS	Loss: Monitoring status of PLL input clock 1: PLL input clock not detected 0: PLL input clock available This bit indicates the current monitoring status of the PLL input clock. Read access only
0	PLL_LOCK	Lock: Engages at operating frequency; status of PLL: 0: PLL is unlocked 1: PLL is locked This bit represents the current lock state of the PLL. Read access only

CLK_CTRL_REG W/R Addr.: 0x4000_2618 Default: 0x0000_0001		
Description		Control register for ERTEC 400 clock
Bit No.	Name	Description
31..1	----	Reserved
0	CLK_CTRL	Enable of AHB clock for PCI bridge. 0: Clock is disabled 1: Clock is enabled

PM_STATE_REQ_REG R Addr.: 0x4000_261C Default: 0x0000_0000		
Description		Required power state of the PCI host
Bit No.	Name	Description
31: 2	----	Reserved
1:0	PM_STATE_REQ	Required power state of the PCI host

PM_STATE_ACK_REG R/W Addr.: 0x4000_2620 Default: 0x0000_0000		
Description		Current power state of the ERTEC 400
Bit No.	Name	Description
31:2	----	Reserved
1:0	PM_STATE_ACK	Current power state of the ERTEC 400

PME_REG R/W Addr.: 0x4000_2624 Default: 0x0000_0000		
Description		Power management event PME
Bit No.	Name	Description
31:1	----	Reserved
0	PME	Power management event. If this bit is set (rising edge), the PCI signal PME_N can be activated. However, this requires that the PME_ENABLE bit be set in the power management configuration register of the AHB-PCI bridge (see /3/).

QVZ_AHB_ADR		R	Addr.: 0x4000_2628	Default: 0x0000_0000
Description		Address of incorrect addressing on multilayer AHB		
Bit No.	Name	Description		
31:0	QVZ_AHB_ADR	Address		

QVZ_AHB_CTRL		R	Addr.: 0x4000_262C	Default: 0x0000_0000
Description		Control signals of incorrect addressing on multilayer AHB		
Bit No.	Name	Description		
31:7	----	Reserved		
6:4	HBURST	HBURST		
3:1	HSIZE	HSIZE		
0	HWRITE	HWRITE 0: HREAD 1: HWRITE		

QVZ_AHB_M		R	Addr.: 0x4000_2630	Default: 0x0000_0000
Description		Master identifier of an incorrect addressing on the multilayer AHB		
Bit No.	Name	Description		
31:3	----	Reserved		
2	QVZ_AHB_IRT	IRT		
1	QVZ_AHB_LBU/PCI	LBU/PCI		
0	QVZ_AHB_ARM946	ARM946		

QVZ_APB_ADR		R	Addr.: 0x4000_2634	Default: 0x0000_0000
Description		Address of incorrect addressing on AHB		
Bit No.	Name	Description		
31:0	QVZ_APB_ADR	Address		

QVZ_EMIF_ADR		R	Addr.: 0x4000_2638	Default: 0x0000_0000
Description		Address that leads to timeout on EMIF		
Bit No.	Name	Description		
31:0	QVZ_EMIF_ADR	Address		

PCI_RES_REQ		R/W	Addr.: 0x4000_263C	Default: 0xFFFF_0002
Description		Request register for placing a soft reset request on the PCI bridge		
Bit No.	Name	Description		
31:16	MAX_DELAY_PCI	Number of AHB clock cycles allowed to elapse without a response on the AHB-PCI bridge before a timeout is triggered.		
15:2	----	Reserved		
1	PCI_QVZ_EN	Enable of timeout monitoring of AHB accesses to the PCI bridge. 0: Monitoring inactive 1: Monitoring active		
0	PCI_SOFT_RES_REQ	PCI bridge requirement for soft reset 0: Request is inactive 1: Request is active		

PCI_RES_ACK			R	Addr.: 0x4000_2640	Default: 0x0000_0000
Description		Acknowledge register for display of an implemented soft reset request			
Bit No.	Name	Description			
31:1	----	Reserved			
0	PCI_SOFT_RES_ACK	Display of implemented soft reset by PCI bridge 0: Request was not implemented 1: Request was implemented			

MEM_SWAP			R/W	Addr.: 0x4000_2644	Default: 0x0000_0000
Description		Memory swapping in Segment 0 between ROM and RAM			
Bit No.	Name	Description			
31:1	----	Reserved			
0	MEM_SWAP	Selection of memory in Segment 0: 0: Boot ROM starting with Addr 0h 1: Internal user RAM starting with Adr 0h			

PCI_INT_CTRL			R/W	Addr.: 0x4000_2648	Default: 0x0000_0000
Description		Control of PCI interrupts			
Bit No.	Name	Description			
31:2	----	Reserved			
1	PCI_INT_CTRL[1]	PCI_INT_CTRL = 0x0000_0001 IRQ0_HP → SERR_N			
0	PCI_INT_CTRL[0]	PCI_INT_CTRL = 0x0000_0000 IRQ0_HP → INTB_N			

M_LOCK_CTRL			R/W	Addr.: 0x4000_264C	Default: 0x0000_0000
Description		AHB master lock enable. Master-selective enable of AHB lock functionality.			
Bit No.	Name	Description			
31:4	----	Reserved			
3	ARB_MODE	Select arbitration algorithm for AHB arbiter (ARB_MODE). 0: Round robin 1: Fixed priority assignment This bit should not be changed (default: round robin)!			
2	LOCK_ENA_IRT	Lock enable of AHB master IRT: 0: Lock disabled 1: Lock enabled			
1	LOCK_ENA_PCI/LBU	Lock enable of AHB master PCI/LBU: 0: Lock disabled 1: Lock enabled			
0	LOCK_ENA_ARM946	Lock enable of AHB master ARM9: 0: Lock disabled 1: Lock enabled			

ARM9_CTRL			R/W	Addr.: 0x4000_2650	Default: 0x0000_1939
Description		Check of ARM9 and ETM inputs that are not accessible from external pins. This register can only be written to if the Write enable bit is set in the ARM9_WE register. This register can only be changed for debugging purposes!			
Bit No.	Name	Description			
31:14	----	Reserved			
13	BIGENDIAN	BIGENDIAN (read only)			
12	DISABLE_GATE_THE_CLK	DisableGateTheClk: 1: ARM9 processor clock runs freely 0: ARM9 processor clock is paused by a Wait-for-Interrupt.			
11	DBGEN	DBGEN: Enable of embedded ARM9 debugger 1: Debugger is enabled. 0: Debugger is disabled.			
10	MICEBYPASS	MICEBYPASS: Bypass of TCK synchronization to the ARM9 clock. 0: TCK is synchronized to ARM 9 clock 1: TCK is not synchronized to ARM 9 clock			
9	INITRAM	INITRAM: Indicates whether the TCMs are enabled after a (SW) reset. 1: TCMs enabled 0: TCMs disabled This bit is only reset by the external RESET_N reset. SW and watchdog resets have no effect on this bit.			
8:0	SYSOPT[8:0]	SYSOPT(8:0): Indicates the implemented ETM options. For more detailed information: See /19/. Default value: 139H			

ARM9_WE			R/W	Addr.: 0x4000_2654	Default: 0x0000_0000
Description		Write access register for the ARM9_CTRL register			
Bit No.	Name	Description			
31:1	----	Reserved			
0	WE_ARM9_CTRL	Write enable for ARM9_CTL register 1: ARM9_CTRL can be write accessed. 0: ARM9_CTRL is read-only.			

5 General Hardware Functions

5.1 Clock Generation and Clock Supply

The clock system of the ERTEC 400 basically consists of four clock systems that are decoupled through asynchronous transfers:

- ARM946E-S together with AHB bus, APB bus, and IRT
- JTAG Interface
- PCI bus
- RMII/MII – interfacing of Ethernet MACs

5.1.1 Clock Supply in ERTEC 400

The required clocks are generated in the ERTEC 400 by means of internal PLL and/or through direct infeed. The following table provides a detailed list of the clocks:

MODULE	CLOCK SOURCE	FREQUENCY
ARM946ES	PLL	50/100/150 MHz (scalable)
AHB/EMIF/ICU	PLL	50 MHz
IRT (except MII/RMII)	PLL	50/100 MHz
APB	PLL	50 MHz
JTAG	JTAG clock	0-10 MHz
PCI	PCI clock	0-66 MHz
MII/RMII	RX/TX clock (MII) REF_CLK (RMII)	25 MHz (MII) 50 MHz (RMII)

Table 17: Overview of ERTEC 400 Clocks

Synchronous clocks CLK_50MHz and CLK_100MHz are used primarily in the ERTEC 400. These clocks are generated with an internal PLL that is, in turn, supplied by a quartz or oscillator.

The input clock is selected using the **CONFIG0** configuration pin.

CONFIG0 = 0 → Input clock is fed with a quartz via the **CLKP_A, CLKP_B** pin.

CONFIG0 = 1 → Input clock is fed with an oscillator clock via the **REF_CLK** pin.

In the case of direct infeed at the REF_CLK pin, the clock frequency can be set with the **CONFIG1** configuration pin.

CONFIG1 = 0 → 50 MHz input clock

CONFIG1 = 1 → 25 MHz input clock

The PLL generates the CLK_50MHz and CLK_100MHz system clocks as well as the clock for the ARM946E-S.

This clock can be scaled with the **CONFIG3** and **CONFIG4** configuration pins.

CONFIG4, CONFIG3 = 00 → ARM946 processor clock 50 MHz.

CONFIG4, CONFIG3 = 01 → ARM946 processor clock 100 MHz.

CONFIG4, CONFIG3 = 10 → ARM946 processor clock 150 MHz.

CONFIG4, CONFIG3 = 11 → Reserved.

The figure below shows the structure of the clock unit with the individual input and output clocks.

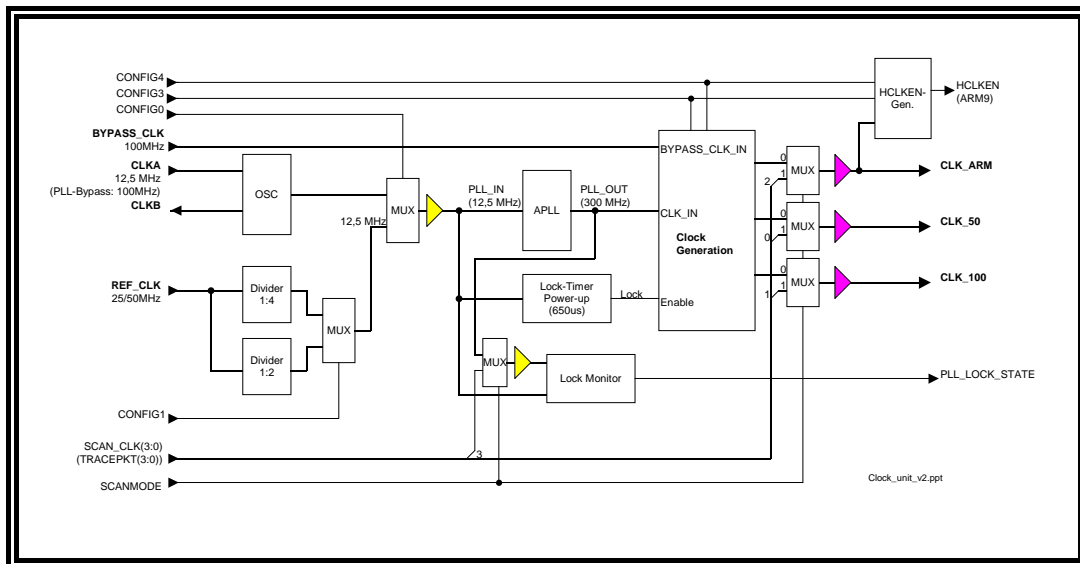


Figure 9: Detailed Representation of Clock Unit

5.1.2 PCI Clock Supply

The clock supply of the AHB PCI bridge is implemented using two different clock inputs.

- Using the external **CLK_PCI** pin at a frequency of 33 MHz or 66 MHz.
- Using the internal **CLK_50MHz** clock per SW via the **CLK_CTRL_REG** system control register.

After a power-up reset, AHB clock CLK_50MHz is enabled.

5.1.3 LBU Clock Supply

In LBU mode, AHB clock CLK_50MHz is enabled for the LBU clock supply. The clock supply for the LBU is disabled in PCI mode.

In LBU mode, it is recommended that the AHB clock for the PCI bridge be disabled.

Configuration pin **CONFIG2** is used to select PCI or LBU mode.

CONFIG2 = 0 → LBU mode
CONFIG2 = 1 → PCI mode

5.1.4 JTAG Clock Supply

The clock supply for the JTAG interface is implemented using the **JTAG_CLK** pin. The frequency range is between 0 and 10 MHz. The boundary scan and the ICE macro cell of the ARM946E-S are enabled via the JTAG interface.

5.1.5 Ethernet Interface Clock Supply

In the case of Ethernet ports, there are two interfaces to the PHY blocks:

- **MII mode** = 2 Ethernet ports to 2 MII-PHYs
- **RMII mode** = 4 Ethernet ports to quad RMII-PHY

In RMII mode, the Ethernet ports and the PHYs are supplied by the CLK_50MHz system clock. Communication between the ERTEC port and the PHYs is synchronous.

In MII mode, the two PHYs are supplied with one 25 MHz PHY clock. The clock for the Ethernet ports of the ERTEC 400 is supplied by the MII PHYs via the RX_CLK and TX_CLK clock cables.

The clock for the Ethernet ports is enabled/disabled via the **clock control register** in the IRT switch.

The following figure shows the two different Ethernet modes with the clock supply.

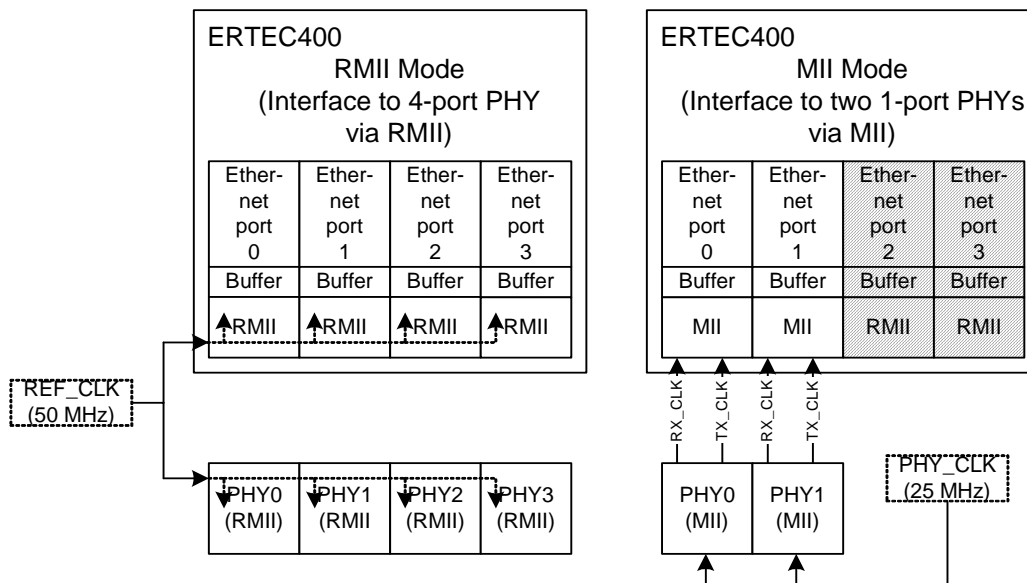


Figure 10: Clock Supply of Ethernet Interface

5.2 Reset Logic of the ERTEC 400

The reset logic resets the entire circuit of the ERTEC 400 except for the PCI portion of the AHB-PCI bridge. The reset system of the ERTEC 400 is enabled by the following events:

- Hardware reset via external **RESET_N** pin
- Software reset via **XRES_SOFT** bit in the reset control register
- Watchdog reset via watchdog timer overflow

The triggering reset event can be read out in the reset status register.

5.2.1 Hardware Reset

The external hardware reset circuitry is connected at the **RESET_N** pin of the ERTEC 400. If the hardware reset is enabled, the entire ERTEC 400 circuit except for the PCI portion is reset internally. The hardware reset must be present steadily for **at least 35 μ s** (see figure below). Afterwards, the PLL powers up within $t_{\text{Lock}} = 400 \mu\text{s}$. The lock status of the PLL is monitored. The state of the PLL can be read out in the **PLL_STAT_REG** status register. A filter is integrated at the **RESET_N** input, which suppresses spikes up to 10 ns. In the case of the hardware reset, a bit is set in the reset status register. This bit remains unaffected by the triggered reset function. This register can be evaluated after a restart. The following figure shows the power-up phase of the PLL after a reset.

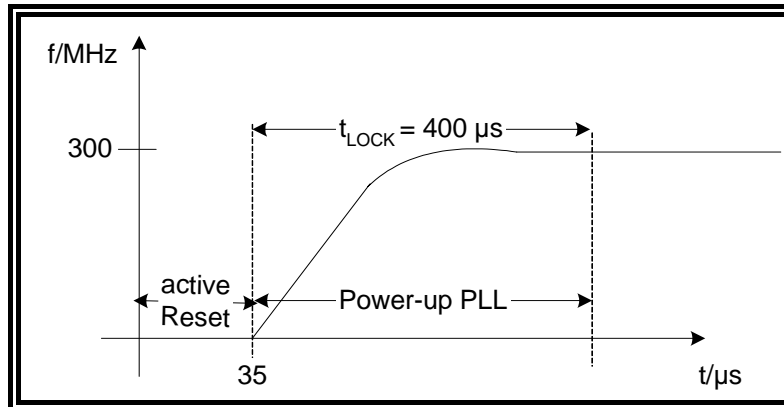


Figure 11: Power-Up Phase of the PLL

5.2.2 Watchdog Reset

The watchdog reset involves software monitoring by the hardware. Monitoring is based on a time setting in the watchdog timer. Retriggering the timer at a specified reload value prevents the watchdog reset from being triggered. If the timer is not retriggered, the watchdog reset is enabled after the timer expires if the watchdog function is active with the **WD_RES_FREI** bit. The watchdog reset is controlled in the ERTEC 400 by means of assignable pulse stretching (PV). The watchdog reset resets the complete ERTEC 400 circuit. The watchdog event is signaled to the host system via a GPIO pin.

As in the case of a hardware reset, a bit is set in the reset status register. This bit remains unaffected by the triggered reset function. This register can be evaluated after a restart.

5.2.3 Software reset

A software reset can be triggered in the ERTEC 400 by setting a bit in the reset control register. The **XRES_SOFT** bit is set in the reset status register when the reset is triggered.

5.2.4 PCI bridge reset

A hardware reset of the PCI bridge can be generated using the following functions:

- Activating hardware reset pins **RES_PCI_N** and **RESET_N**
- Activating the AHB PCI software reset by setting the **XRES_PCI_AHB_SOFT** bit in the reset control register

The PCI bridge side that is clocked with CLK_PCI (33/66 MHz) resets the **RES_PCI_N** output. The state of **RES_PCI_N** can be read in the reset control register.

The PCI bridge side that is clocked with CLK_50MHz is reset by the **XRES_SOFT** and **XRES_PCI_AHB_SOFT** software resets and the watchdog reset.

In order to ensure a defined state of the PCI bridge through the reset sources indicated above, you must make sure that both sides of the PCI bridge are reset. A hardware reset of the PCI bridge clears all bridge and configuration registers.

To prevent the PCI registers from being cleared, a warm reset is possible for the PCI bridge. This requires that the **PCI_SOFT_RESREQ** bit be set in the **PCI_RES_REQ** register. The PCI bridge terminates all transactions at this moment and issues an acknowledgement with **PCI_SOFT_RESACK**. This state can be scanned in the

PCI_RES_ACK register by the ARM946E-S user software. All new transactions from the AHB and PCI bus are rejected with Retry. This also pertains to the bridge and configuration registers on the AHB side. The state is retained until the **PCI_SOFT_RESREQ** bit is deleted again. All of the registers indicated above can be addressed in the system control register area.

5.2.5 Actions when HW Reset is Active

During the active HW reset phase, the states of the 3 boot pins (3) are read in to the **BOOT_REG** register and the states of the config pins (5) are read in to the **CONFIG_REG** register. After the hardware reset phase, these pins are available as normal EMIF function pins.

5.3 Address Space and Timeout Monitoring

Monitoring mechanisms are incorporated in the ERTEC 400 for detection of incorrect addressing, illegal accesses, and timeout. The following I/O are monitored:

- AHB bus
- APB bus
- EMIF
- PCI slave

5.3.1 AHB Bus Monitoring

Separate address space monitoring is assigned to each of the three AHB masters. If an AHB master addresses an unused address space, the access is acknowledged with an error response and an FIQ interrupt is triggered at the ARM946 interrupt controller. The incorrect access address is stored in the **QVZ_AHB_ADR** system control register and the associated access type (read, write, HTRANS, HSIZE) is stored in the **QVZ_AHB_CTRL** system control register. The master that caused the access error is stored in the **QVZ_AHB_M** system control register. In the case of an access violation by PCI or LBU as an AHB master, an interrupt request is also enabled and stored in the IRT macro. The interrupt is issued to the PCI/LBU bus as an INTA_N interrupt. In the case of an access violation by the PCI user, Bit 0 (for write accesses) or Bit 1 (for read accesses) is set in the AHB status register of the PCI bridge and the INTA_N interrupt is enabled.

If more than one AHB master causes an access violation simultaneously (accurate within one AHB clock cycle), only the violation of the highest priority AHB master is indicated in the registers (see Section 3.1.1).

Diagnostic registers **QVZ_AHB_ADR**, **QVZ_AHB_CTRL**, and **QVZ_AHB_M** remain locked for subsequent access violations until the **QVZ_AHB_CTRL** register has been read.

5.3.2 APB Bus Monitoring

The APB address space is monitored on the APB bus. If incorrect addressing is detected in the APB address space, access to the APB side and AHB side is terminated with an "OKAY" response because the APB bus does not recognize response-type signaling. An FIQ interrupt is triggered on the ARM945 interrupt controller. The incorrect access address is placed in the **QVZ_APB_ADR** system control register. The **QVZ_APB_ADR** system control register is locked for subsequent address violations until it has been read.

5.3.3 EMIF Monitoring

In the case of the EMIF, the external **RDY_PER_N** ready signal is monitored. In order to enable monitoring, "Extended_Wait_Mode" must be switched on in the **Async_Bank_0_Config** to **Async_Bank_3_Config** configuration registers. If one of the four memory areas that are selected via the CS_PER0_N to CS_PER3_N chip select outputs is addressed, the memory controller of the ERTEC 400 waits for the **RDY_PER_N** input signal. The monitoring duration is set in the **ASYNC_WAIT_CYCLE_CONFIG** EMIF register and is active if timeout monitoring (Bit 7) is set in the **EXTENDED_CONFIG** EMIF register. The specified value (maximum of 255) multiplied by 16 AHB clock cycles yields the monitoring time, i.e., the time that the memory controller waits for the Ready signal. After this time elapses, a Ready signal is generated for the memory controller and an FIQ interrupt is generated for the ARM946 interrupt controller. In addition, the address of the incorrect access is stored in the **QVZ_EMIF_ADR** system control register. The **QVZ_EMIF_ADR** system control register is locked for subsequent address violations until it has been read.

The set FIQ interrupt is then removed if timeout monitoring is reset.

5.3.4 PCI Slave Monitoring

The **HREADY** signal of the PCI slave is monitored with an 8-bit wide counter, i.e., monitoring is triggered after 256 AHB clock cycles. Monitoring can be enabled or disabled in the **PCI_RES_REQ** system control register. The monitoring counter is reset when **HREADY** = 1 and incremented when **HREADY** = 0.

There are three possible reasons for the timeout:

- Actual timeout in the slave

If **HREADY** is still 0 after a maximum of 255 AHB clock cycles, access to the master is terminated with an error response and the timeout interrupt is activated.

Access to the slave continues. As long as the slave does not supply **HREADY**=1, all other accesses to the slave must be blocked with an error response. The interrupt is triggered only once.

If the address phase of a non-IDLE access is pending in parallel to the extended data phase, this access is canceled and an IDLE address phase is output to the slave.

- Too many retrys in a row for the same access

Access to the master is terminated with an error response and the timeout interrupt is activated. Because there is no requirement that an access that is rejected with Retry has to be repeated, the next access of the master can be switched to the slave.

- HSPLIT is missing after a split response

Access to the master is terminated with an error response and the timeout interrupt is activated. The slave must continue to wait for signal HSPLIT=1. As long as the signal to the slave is missing, all other accesses to the slave must be blocked with an error response.

According to the AHB specification, once the slave outputs HSPLIT=1, access must be repeated. However, because access is already terminated for the master, the data phase can no longer be handled correctly.

6 External Memory Interface (EMIF)

In order to access an external memory area, an **External Memory Interface** is incorporated in the ERTEC 400. The interface contains one SDRAM memory controller and one SRAM memory control each for asynchronous memory. Both interfaces can be assigned separately as active interfaces. That is, the data bus is driven actively to High at the end of each access. The internal pull-ups keep the data bus actively at High. External pull-ups are not required. When writing, this occurs after the end of the strobe phase. When reading, this occurs after a specified time has elapsed after the end of the strobe phase to avoid driving against the externally read block. For the SDRAM controller, this time is equivalent to one AHB bus cycle. For the asynchronous controller, the time is equivalent to the time required for the hold phase to elapse, which corresponds to the time from the rising edge of RD_N to the rising edge of the chip select signal. By default, the active interface is switched on.

The following signal pins are available for the EMIF on the ERTEC 400.

• Data bus	32 bit	D[31 : 0]
• Address bus	24 bit	A[23 : 0]
• Memory CS	4	CS_PER0_N - CS_PER3_N
• Byte enable	4	BE0_DQM0_N – BE3_DQM3_N
• RD/WR Async.	2	RD_N/WR_N
• Ready	1	RDY_PER_N
• DIR	2	DTR_N/OE_DRIVER_N
• SDRAM	5	CLK_SDRAM/CS_SDRAM_N/RAS_SDRAM_N/ CAS_SDRAM_N/WE_SDRAM_N

The SDRAM controller has the following features:

- 16-bit or 32-bit data bus width can be assigned
- PC100 SDRAM-compatible (50 MHz clock frequency)
- A maximum of 256 Mbytes of SDRAM for 32-bit data bus width, or
- A maximum of 128 Mbytes of SDRAM for 16-bit data bus width
- Supports various SDRAMs with the following properties:
 - CAS latency 2 or 3 clock cycles
 - 1/2/4 internal banks can be addressed (A1 : 0)
 - 8/9/10/11 bits column address (A13, 11:2)
 - Maximum of 13 row addresses (A14 : 2)

SDRAMS with a maximum of 4 banks are supported. The SDRAM controller can keep all 4 banks open simultaneously. In terms of addresses, these four banks correspond to one quarter of the SDRAM address area on the AHB bus. As long as the alternating accesses are in the respective page, no page miss can occur.

The asynchronous memory controller has the following features:

- 8-bit, 16-bit, or 32-bit data bus width can be assigned
- 4 chip selects
- Maximum of 16 Mbytes per chip select can be addressed
- Different timing can be assigned for each chip select
- Ready signal can be assigned differently for each chip select
- Chip select CS_PER0_N can be used for a BOOT operation from external memory
- Data bus width of the external memory for a BOOT operation is selected via the BOOT[2:0] input pins
- Default setting "Slow timing" for BOOT operation
- Timeout monitoring can be assigned
- Supports the following asynchronous blocks
 - SRAM
 - Flash PROM
 - External I/O blocks

6.1 Address Assignment of EMIF Registers

The EMIF registers are **32 bits in width**. The registers can be written to with 32-bit accesses only.

EMIF (Base Address 0x7000_0000)					
Register Name	Offset Address	Address Area	Access	Default	Description
Revision_Code_and_Status	0x0000	4 bytes	R	0x00000000	Revision code and status register
Async_Wait_Cycle_Config	0x0004	4 bytes	W/R	0x40000080	Async wait cycle config register
SDRAM_Bank_Config	0x0008	4 bytes	R	0x000020A0	SDRAM bank config register
SDRAM_Refresh_Control	0x000C	4 bytes	W/R	0x00000190	Setting of refresh rate Indication for timeout
Async_BANK0_Config	0x0010	4 bytes	W/R	0x3FFFFFF2	Timing/data bus width for access via async. interface CS_PER0_N
Async_BANK1_Config	0x0014	4 bytes	W/R	0x3FFFFFF2	Timing/data bus width for access via async. interface CS_PER1_N
Async_BANK2_Config	0x0018	4 bytes	W/R	0x3FFFFFF2	Timing/data bus width for access via async. interface CS_PER2_N
Async_BANK3_Config	0x001C	4 bytes	W/R	0x3FFFFFF2	Timing/data bus width for access via async. interface CS_PER3_N
Extended_Config	0x0020	4 bytes	W/R	0x03030000	Setting of additional functionalities

Table 18: Overview of EMIF Registers

6.2 EMIF Register Description

Revision Code and Status			R	Addr.: 0x7000_0000	Default: 0x0000_0000
Description		Revision code and status register			
Bit No.	Name	Description			
31	----	Reserved			
30	----	Reserved			
29..16	----	Reserved			
15..8	Major_Revision	00h			
7..0	Minor_Revision	00h			

Async wait cycle config			W/R	Addr.: 0x7000_0004	Default: 0x4000_0080
Description		Async wait cycle config register			
Bit No.	Name	Description			
31	----	Reserved			
30	WP	Wait polarity 0: Wait if RDY_PER_N = 0 1: Wait if RDY_PER_N = 1			
29..8	----	Reserved			
7..0	MAX_EXT_WAIT	This value multiplied by 16 is equivalent to the number of AHB clock cycles that the asynchronous controller waits for RDY_PER_N before access is terminated with timeout IRQ.			

SDRAM Bank Config		W/R	Addr.: 0x7000_0008	Default: 0x0000_20A0
Description		SDRAM bank config register		
Bit No.	Name	Description		
31..15	----	Reserved		
14	WB	Write burst type 0: Programmed burst length 1: Single write (not permitted for 16-bit SDRAM bank)		
13*	CL	CAS latency 0: SDRAM is activated with CAS latency = 2 1: SDRAM is activated with CAS latency = 3		
12..11	----	Reserved		
10..8*	ROWS	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		
7	----	Reserved		
6..4	IBANK	Internal SDRAM bank setup (number of banks in the SDRAM) 000: 1 bank 001: 2 banks 010: 4 banks 011 .. 111: Reserved		
3	----	Reserved		
2..0	PAGESIZE	Page size 000: SDRAM with 8-column address lines 001: SDRAM with 9-column address lines 010: SDRAM with 10-column address lines 011: SDRAM with 11-column address lines 100..111: Reserved		

*) Attention: Writing to SDRAM_Bank_Config executes the Mode Register Set command on the SDRAM if Bit 29 (init_done) is set in the SDRAM_Refresh_Control register (i.e., the SDRAM power-up sequence has been executed).

SDRAM Refresh Control		W/R	Addr.: 0x7000_000C	Default: 0x0000_0190
Description		Setting of refresh rate, indication for timeout		
Bit No.	Name	Description		
31	----	Reserved		
30	AT (Read only)	Asynchronous timeout Set to 1 in event of timeout		
29	INIT_DONE (Read only)	SDRAM initialization done 0: SDRAM power-up sequence is running 1: SDRAM power-up sequence is complete		
28..13	----	Reserved		
12..0	REFRESH_RATE	Refresh rate Number of AHB clock cycles between 2 SDRAM refresh cycles		

Async Bank 0 Config	W/R	Addr.: 0x7000_0010	Default: 0x3FFF_FFF2
Async Bank 1 Config	W/R	Addr.: 0x7000_0014	Default: 0x3FFF_FFF2
Async Bank 2 Config	W/R	Addr.: 0x7000_0018	Default: 0x3FFF_FFF2
Async Bank 3 Config	W/R	Addr.: 0x7000_001C	Default: 0x3FFF_FFF2
Description	Settings for the timing and the data bus width for accesses via an asynchronous interface (CS_PER0_N to CS_PER3_N) (AHB clock cycle has a length of 20 ns)		
Bit No.	Name	Description	
31	----	Reserved	
30	EW	Extend Wait mode 0: RDY_PER_N = don't care 1: Wait until RDY_PER_N is active	
29..26	W_SU	Write strobe setup cycles (w_su + 1) AHB clock cycles between valid address, data, and chip select and falling edge of the write signal.	
25..20	W_STROBE	Write strobe duration cycles (w_strobe + 1) AHB clock cycles between falling and rising edges of the write signal.	
19..17	W_HOLD	Write strobe hold cycles (w_hold + 1) AHB clock cycles between rising edge of the write signal and change of address, data, and chip select.	
16..13	R_SU	Read strobe setup cycles (r_su + 1) AHB clock cycles between valid address and chip select and falling edge of the read signal (RD_N).	
12..7	R_STROBE	Read strobe duration cycles (r_strobe + 1) AHB clock cycles between falling and rising edges of the read signal.	
6..4	R_HOLD	Read strobe hold cycles (r_hold + 1) AHB clock cycles between rising edge of the read signal and change of address and chip select.	
3..2	----	Reserved	
1..0	ASIZE	Asynchronous bank size 00: 8-bit data bus 01: 16-bit data bus 1x: 32-bit data bus	

Extended Config		W/R	Addr.: 0x7000_0020	Default: 0x0303_0000
Description	Setting of additional functionalities			
Bit No.	Name	Description		
31	----	Reserved		
30	TEST_1	Test Mode 1 0: 200 µs delay after system reset (SDRAM power-up) 1: Delay after system reset is immediately terminated		
29	TEST_2	Test Mode 2 0: Normal function 1: All SDRAM accesses are misses		
28..26	----	Reserved		
25	ADB	Active data bus After each access to the SDRAM, the data bus is driven actively to 1 in order to support integrated pull-ups.		
24	ASDB	Asynchronous active data bus After each access to the asynchronous area, the data bus is driven actively to 1 at the end of the Hold phase in order to support integrated pull-ups.		
23..20	----	Reserved		
19	TEST_3	Test Mode 3 0: Normal function 1: DTR_N = Test Output		
18	----	Reserved		
17..16	BURST_LENGTH	SDRAM burst length 00: 1 01: 2 10: Full Page, Read INCR_S burst length = 4 11: Full Page, Read INCR_S burst length = 8		
15	----	Reserved		
14	TRCD/TCD	Time between the SDRAM commands Activate and read/write, precharge and activate 0: 2 AHB clock cycles 1: 1 AHB clock cycle		
13..9	----	Reserved		
8	SDSIZE	SDRAM bank size 0: 32-bit data bus 1: 16-bit data bus		
7	ATIRQ	0: Timeout watchdog for asynchronous accesses disabled 1: Timeout watchdog for asynchronous accesses enabled After the watchdog expires (256 AHB clock cycles), an interrupt is triggered. Setting Bit 7 to 0 deletes interrupt source.		
6..0	----	Reserved		

7 Local Bus Unit (LBU).

The ERTEC 400 can also be operated from an external host processor. Two different bus interfaces are available for this purpose:

- PCI bus
- Local bus unit

The bus system is selected using the CONFIG[2] input pin.

CONFIG[2] = 0 LBU bus system is active

The LBU is a 16-bit data interface.

The following signal pins are available for the LBU on the ERTEC 400.

- | | | |
|--------------------------|--------|-------------------------------|
| • Data bus | 16 bit | LBU_DB[15 : 0] |
| • Address bus | 21 bit | LBU_AB[20 : 0] |
| • Memory CS | 1 | LBU_CS_M_N |
| • Register CS | 1 | LBU_CS_R_N |
| • RD/WR config | 1 | LBU_CFG |
| • RD/WR | 2 | LBU_WR_N / LBU_RD_N |
| • Ready | 2 | LBU_POL_RDY, LBU_RDY_N |
| • Byte selection | 2 | LBU_BE[1 : 0] |
| • Page segment selection | 2 | LBU_SEG[1 : 0] |
| • Interrupt outputs | 2 | LBU_IRQ0_N, LBU_IRQ1_N |

Four different pages within the ERTEC 400 can be accessed via the LBU. Each page can be set individually.

The settings for the four pages are made via the LBU page registers. Five page registers are available per page. These registers are used for the size, offset, and access width settings of the page. The "LBU_CS_R_N" chip select signal can be used to access the page registers.

The following settings are possible for each page:

- Access size of a page between 256 bytes and 2 Mbytes with 2-page range register
- Offset (segment) of page in 4-Gbyte address area with 2-page offset register
- Access type (data bit width) with 1 page control register

The ERTEC 400-internal address area is accessed via the "LBU_CS_M_N" chip select signal.

The LBU supports accesses to the address area with two separate read and write cables or with a common read/write cable. The access type setting is made via the "LBU_CFG" input. The polarity of the ready signal is set via the "LBU_POL_RDY" input.

LBU_POL_RDY	LBU_RDY
0	Low active
1	High active

LBU_RDY_N is a tristate output and must be pulled to his "ready" level by an external pull-down or pull-up resistor. During an access from the LBU-Interface to the ERTEC 400 (CS with RD or WR activ) , the LBU_RDY_N switched to inactiv (Wait) first. LBU_RDY_N will be active for a 50 MHz-Clock if data can be read or write. After that LBU_RDY_N switched back to tristate. The external Pull- (up/down) resistor drives the ready state.

The four segments are addressed via the two LBU_SEG[1:0] inputs.

LBU_SEG[1 : 0]	Addressed Segment
00	LBU_PAGE0
01	LBU_PAGE1
10	LBU_PAGE2
11	LBU_PAGE3

7.1 Page Range Setting

The page size of each page is set in the PAGE_x_RANGE_HIGH and PAGE_x_RANGE_LOW range registers (x = 0 to 3). Together, the two page range registers yield a 32-bit address register. The size of the page varies between 256 bytes and 2 MBytes. Therefore, Bits 0 to 7 and Bits 22 to 31 of the PAGE_x_RANGE register remain unchanged at a value of 0 even if a value of 1 is entered. If no bit at all is set in one of the PAGE RANGE registers, the range of this page is set to 256 bytes, by default. If several bits are set to 1 in one of the PAGE RANGE registers, the range is always calculated based on the most significant bit (see Example 3).

PAGE _x _RANGE_HIGH			PAGE _x _RANGE_LOW				Size of pagex
31	2423	16	15	8	7	0	
00000000	00000000		00000001	00000000			256 bytes
00000000	00010000		01000000	00000000			1 Mbytes
00000000	00000110		00000000	00000000			256 Kbytes
00000000	00000010		00000000	00000000			128 Kbytes

Table 19: Setting of Various Page Sizes

The largest page determines the number of addresses that have to be connected to the LBU. In the page range table above, the largest page is 1 Mbyte (i = Bit 20). The maximum addresses are calculated from $A_{max} = 20 - 1$. In this case, address cables **A [19:0]** are required.

This addressing mechanism results in a mirroring of the specified page size in the total segment.

7.2 Page Offset Setting

The page offset of each page is set in the PAGE_x_OFFSET_HIGH and PAGE_x_OFFSET_LOW range registers (x = 0 to 3). Together, the two page offset registers yield a 32-bit address register. The register is evaluated in such a way that the offset is evaluated only up to the highest set bit of the associated page range register. These bits are then switched to the AHB bus as the highest address. The following table shows some examples for an offset calculation.

PAGE _x _OFFSET_HIGH			PAGE _x _OFFSET_LOW				Offset for Pagex
31	2423	16	15	8	7	0	
00000000	00000000		00000001	00000000			256 bytes
01000000	00000000		00000000	00000000			1 Gbyte
00010000	00000000		00000000	00000000			256 Mbytes
00000000	00000001		00000000	00000000			64 Kbytes

Table 20: Setting of Various Offset Areas

Because the host computer can always access the page registers, the pages can be reassigned at any time. This is useful, for example, if a page is to be used to initialize the I/O. If access to this address area is no longer required after the initialization, the page can then be reassigned in order to access other address areas of the ERTEC 400.

7.3 Page Control Setting

The user can use the page control register to set the type of access to the relevant page. Certain areas of the ERTEC 400 must be implemented with a 32-bit data access in order to ensure data consistency. For other areas, an 8-bit or 16-bit data access is permitted. The following table shows which ERTEC 400 address areas require 32-bit access.

ERTEC 400 Area	32-Bit Access Required	32-Bit Access Possible
System control register	X	-
Timer 0 / 1	X	-
F-counter	X	-
Watchdog	X	-
IRT register	X	-
SDRAM	-	X
User RAM	-	X
Communication RAM (as user RAM)	-	X
Communication RAM (switch RAM)	-	X
Residual APB I/O (UARTs, SPI, GPIO)	-	X

Table 21: Overview of Accesses to Address Areas of ERTEC 400

A setting is made in the paging control registers to indicate whether the relevant page area is addressed according to a 16-bit or 32-bit organization. In the case of a page with 16-bit organization, each 8-bit or 16-bit access is forwarded to the AHB bus. In the case of a page with 32-bit organization, 32-bit read access is implemented on the AHB bus when the LOW word is read. In addition, the LOW word is forwarded and the HIGH word is stored temporarily in the LBU. A subsequent read access to the HIGH word address outputs the temporarily stored value. This ensures consistent reading of 32-bit data on a 16-bit bus. In the case of 32-bit write access, the LOW word is first stored temporarily in the LBU area. When the HIGH word is write accessed, a 32-bit access to the AHB bus is implemented. Eight bit accesses are forwarded directly to the AHB bus and are therefore not useful for a 32-bit page.

When the host accesses address areas of the ERTEC 400, a distinction must be made between 16-bit and 32-bit host processors.

The data width of the variables is defined for a 16-bit host processor. The various compilers implement the accesses in any order. In the case of a 32-bit access by the user software, it must be ensured that LOW 16-bit access to the 32-bit address area precedes HIGH 16-bit access.

In the case of a 32-bit host processor, the access order is defined by setting the "external bus controller" of the host processor. In this case, the address area access must be assigned as "**Little Endian access.**"

Access by the host is asynchronous to the AHB clock of the ERTEC 400. For this reason, it is synchronized with the AHB clock. The following figures show different read- and write sequences with the timings:

7.3.1 LBU Read from ERTEC 400 with separate Read/Write line (LBU_RDY_N active low)

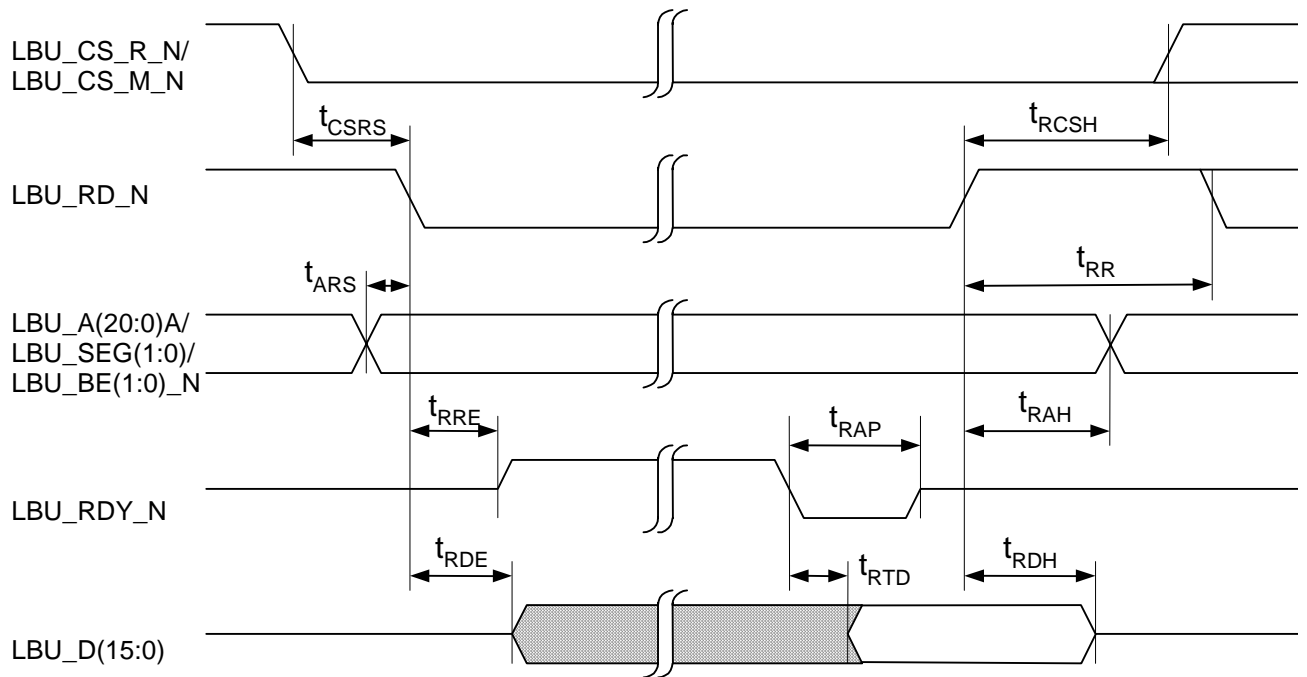


Figure 12: LBU-Read-Sequence with separate RD/WR line

Parameter	Description	Min	Max
t_{CSRS}	chip select asserted to read pulse asserted delay	0 ns	
t_{ARS}	address valid to read pulse asserted setup time	0 ns	
t_{RRE}	read pulse asserted to ready enabled delay	5 ns	12 ns
t_{RDE}	read pulse asserted to data enable delay	5 ns	12 ns
t_{RAP}	ready active pulse width	17 ns	23 ns
t_{RTD}	ready asserted to data valid delay		5 ns
t_{RCSH}	read pulse deasserted to chip select deasserted delay	0 ns	
t_{RAH}	address valid to read pulse deasserted hold time	0 ns	
t_{RDH}	data valid/enabled to read pulse deasserted hold time	0 ns	12 ns
t_{RR}	read recovery time	25 ns	

Table 22: LBU read access timing with separate Read/Write line

7.3.2 LBU Write to ERTEC 400 with separate Read/Write line (LBU_RDY_N active low)

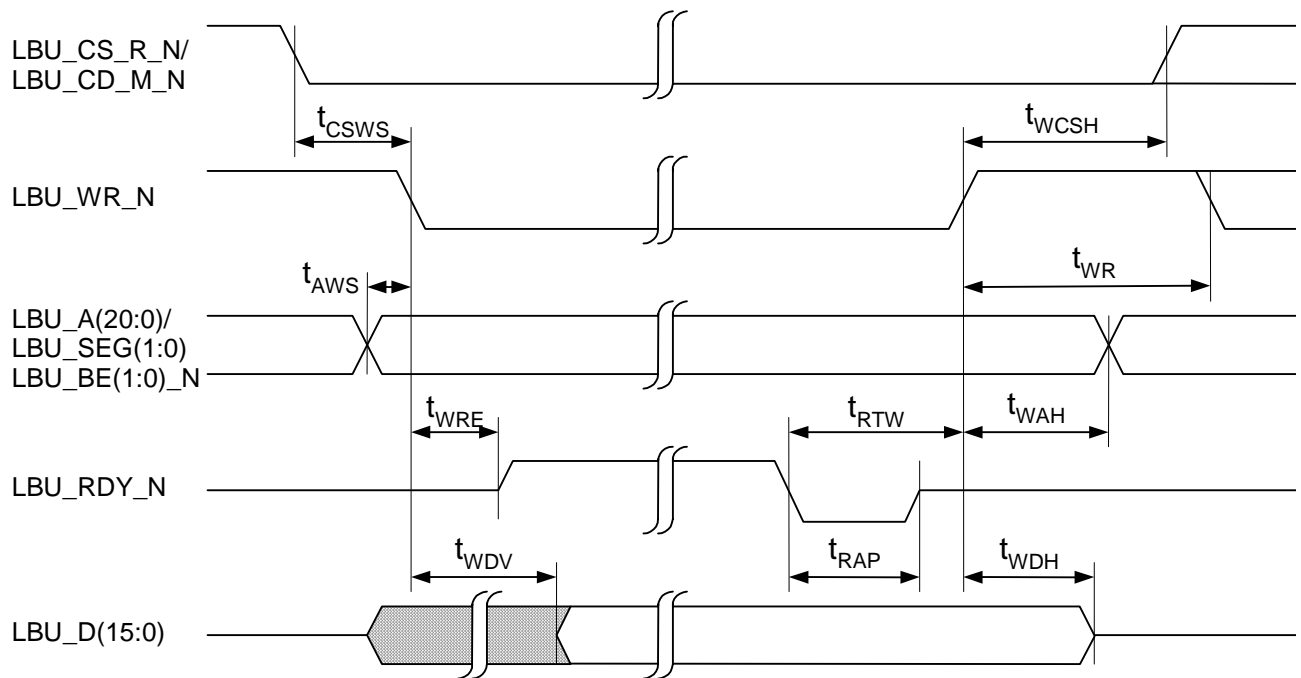


Figure 13: LBU-Write-Sequence with separate RD/WR line

Parameter	Description	Min	Max
t_{CSWS}	chip select asserted to write pulse asserted delay	0 ns	
t_{AWS}	address valid to write pulse asserted setup time	0 ns	
t_{WRE}	write pulse asserted to ready enabled delay	5 ns	12 ns
t_{WDV}	write pulse asserted to data valid delay		40 ns
t_{RAP}	ready active pulse width	17 ns	23 ns
t_{WCSH}	write pulse deasserted to chip select deasserted delay	0 ns	
t_{WAH}	address valid to write pulse deasserted hold time	0 ns	
t_{RTW}	ready asserted to write pulse deasserted delay	0 ns	
t_{WDH}	data valid/enabled to read pulse deasserted hold time	0 ns	
t_{WR}	write recovery time	25 ns	

Table 23: LBU write access timing with separate Read/Write line

7.3.3 LBU Read from ERTEC 400 with common Read/Write line (LBU_RDY_N active low)

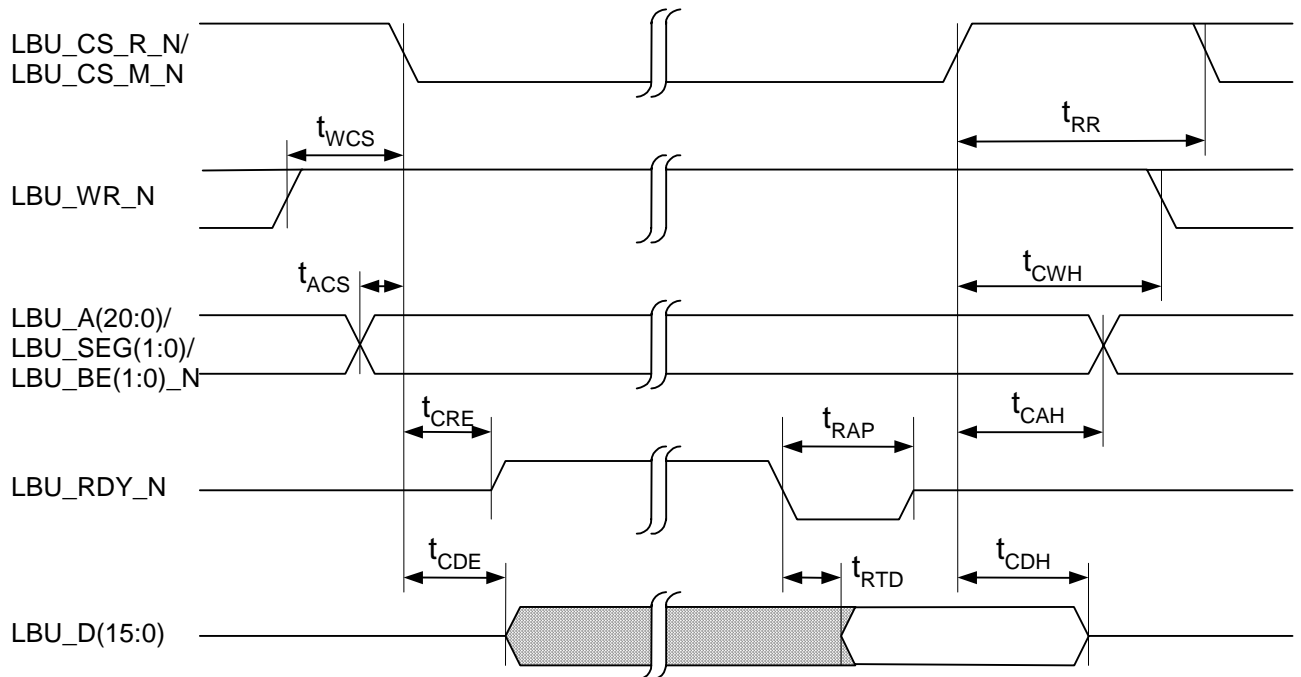


Figure 14: LBU-Read-Sequence with common RD/WR line

Parameter	Description	Min	Max
t_{WCS}	write signal deasserted to chip select asserted setup time	2 ns	
t_{ACS}	address valid to chip select asserted setup time	0 ns	
t_{CRE}	chip select asserted to ready enabled delay	5 ns	12 ns
t_{CDE}	chip select asserted to data enable delay	5 ns	12 ns
t_{RAP}	ready active pulse width	17 ns	23 ns
t_{RTD}	ready asserted to data valid delay		5 ns
t_{CWH}	write signal inactive to chip select deasserted hold time	0 ns	
t_{RAH}	address valid to chip select deasserted hold time	0 ns	
t_{RDH}	data valid/enabled to chip select deasserted hold time	0 ns	12 ns
t_{RR}	read recovery time	25 ns	

Table 24: LBU read access timing with common Read/Write line

7.3.4 LBU Write to ERTEC 400 with common Read/Write line (LBU_RDY_N active low)

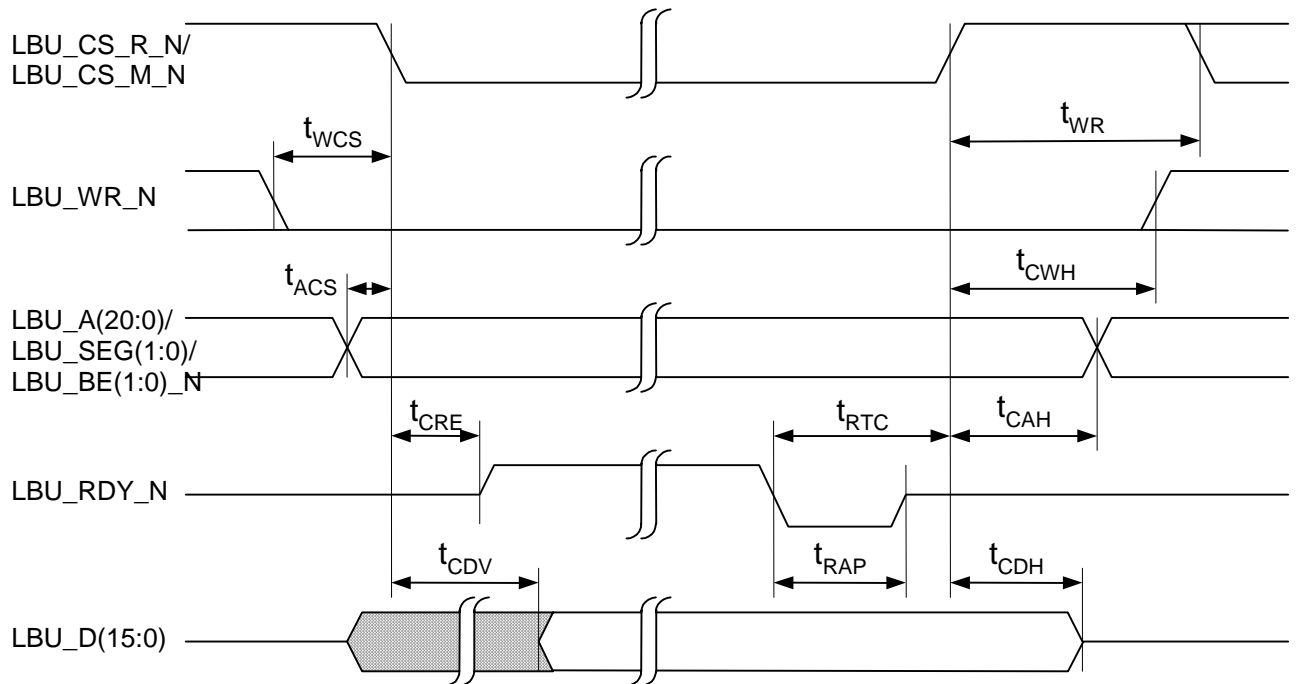


Figure 15: LBU Write Sequence with common RD/WR line

Parameter	Description	Min	Max
t_{WCS}	write signal asserted to chip select setup time	2 ns ¹	
t_{ACS}	address valid to chip select asserted setup time	0 ns	
t_{CRE}	chip select asserted to ready enabled delay	5 ns	12 ns
t_{CDV}	chip select asserted to data valid delay		40 ns
t_{RAP}	ready active pulse width	17 ns	23 ns
t_{CWH}	write signal deasserted to chip select deasserted hold time	0 ns	
t_{CAH}	address valid to chip select deasserted hold time	0 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	
t_{WR}	write recovery time	25 ns	

Table 25: LBU write access timing with common Read/Write line

¹ The setup time t_{WCS} must be maintained under all circumstances; otherwise the LBU unit drives the ERTEC 400 databus.

The ERTEC 400 has two LBU chip select inputs. One for access to the page configuration register (LBU_CS_R_N) and one to access to the ERTEC 400 memory address space (LBU_CS_M_N). Only one of these chip select signals may be active at a time and it is not allowed to change the chip select during the complete access.

The polarity of the signal LBU_RDY_N can be configured using the input LBU_POL_RDY:

LBU_POL_RDY = 0 (external pull down resistor used) → LBU_RDY_N is LOW active
 LBU_POL_RDY = 1 (external pull up resistor used) → LBU_RDY_N is HIGH active

The LBU access control can be configured using the input LBU_CFG:

LBU_CFG = 0 use separate read/write lines (LBU_RD_N and LBU_WR_N)
 LBU_CFG = 1 use common read/write lines (LBU_WD_N)

If common read/write line is selected, the unused input LBU_RD_N must be pulled to inactive level with a pull up resistor.

7.4 Address Assignment of LBU Registers

The LBU registers are **16 bits in width**. The registers can be written to with 16-bit accesses only. The LBU paging configuration registers are addressed via the "LBU_CS_R_N" input.

LBU					
Register Name	Offset Address	Address Area	Access	Default	Description
LBU_P0_RG_L	0x0000	2 bytes	W/R	0x0000	LBU pagex range register 0 Low
LBU_P0_RG_H	0x0002	2 bytes	W/R	0x0000	LBU pagex range register 0 High
LBU_P0_OF_L	0x0004	2 bytes	W/R	0x0000	LBU pagex offset register 0 Low
LBU_P0_OF_H	0x0006	2 bytes	W/R	0x0000	LBU pagex offset register 0 High
LBU_P0_CFG	0x0008	2 bytes	W/R	0x0000	LBU configuration register 0
LBU_P1_RG_L	0x0010	2 bytes	W/R	0x0000	LBU pagex range register 1 Low
LBU_P1_RG_H	0x0012	2 bytes	W/R	0x0000	LBU pagex range register 1 High
LBU_P1_OF_L	0x0014	2 bytes	W/R	0x0000	LBU pagex offset register 1 Low
LBU_P1_OF_H	0x0016	2 bytes	W/R	0x0000	LBU pagex offset register 1 High
LBU_P1_CFG	0x0018	2 bytes	W/R	0x0000	LBU configuration register 1
LBU_P2_RG_L	0x0020	2 bytes	W/R	0x0000	LBU pagex range register 2 Low
LBU_P2_RG_H	0x0022	2 bytes	W/R	0x0000	LBU pagex range register 2 High
LBU_P2_OF_L	0x0024	2 bytes	W/R	0x0000	LBU pagex offset register 2 Low
LBU_P2_OF_H	0x0026	2 bytes	W/R	0x0000	LBU pagex offset register 2 High
LBU_P2_CFG	0x0028	2 bytes	W/R	0x0000	LBU configuration register 2
LBU_P3_RG_L	0x0030	2 bytes	W/R	0x0000	LBU pagex range register 3 Low
LBU_P3_RG_H	0x0032	2 bytes	W/R	0x0000	LBU pagex range register 3 High
LBU_P3_OF_L	0x0034	2 bytes	W/R	0x0000	LBU pagex offset register 3 Low
LBU_P3_OF_H	0x0036	2 bytes	W/R	0x0000	LBU pagex offset register 3 High
LBU_P3_CFG	0x0038	2 bytes	W/R	0x0000	LBU configuration register 3

Table 26: Overview of LBU Registers

7.5 LBU Register Description

LBU_P0_RG_L	W/R	Addr.: LBU_CS_R_N+0x00	Default: 0x0000_0000
LBU_P1_RG_L	W/R	Addr.: LBU_CS_R_N+0x10	Default: 0x0000_0000
LBU_P2_RG_L	W/R	Addr.: LBU_CS_R_N+0x20	Default: 0x0000_0000
LBU_P3_RG_L	W/R	Addr.: LBU_CS_R_N+0x30	Default: 0x0000_0000
Description		Low word of LBU Pagex_Range_register	
Bit No.	Name	Description	
15..0		Lower 16 bits for area setting 15:8 are read/write accessible 7:0 are read-only (value: 00h)	

LBU_P0_RG_H	W/R	Addr.: LBU_CS_R_N+0x02	Default: 0x0000_0000
LBU_P1_RG_H	W/R	Addr.: LBU_CS_R_N+0x12	Default: 0x0000_0000
LBU_P2_RG_H	W/R	Addr.: LBU_CS_R_N+0x22	Default: 0x0000_0000
LBU_P3_RG_H	W/R	Addr.: LBU_CS_R_N+0x32	Default: 0x0000_0000
Description		High word of LBU Pagex_Range_register	
Bit No.	Name	Description	
15..0		Upper 16 bits for area setting 15:6 are read-only (value: 000h) 5:0 are read/write accessible	

LBU_P0_OF_L	W/R	Addr.: LBU_CS_R_N+0x04	Default: 0x0000_0000
LBU_P1_OF_L	W/R	Addr.: LBU_CS_R_N+0x14	Default: 0x0000_0000
LBU_P2_OF_L	W/R	Addr.: LBU_CS_R_N+0x24	Default: 0x0000_0000
LBU_P3_OF_L	W/R	Addr.: LBU_CS_R_N+0x34	Default: 0x0000_0000
Description		Low word of LBU Pagex_Offset_register	
Bit No.	Name	Description	
15..0		Lower 16 bits for offset setting 15:8 are read/write accessible 7:0 are read-only (value: 00h)	

LBU_P0_OF_H	W/R	Addr.: LBU_CS_R_N+0x06	Default: 0x0000_0000
LBU_P1_OF_H	W/R	Addr.: LBU_CS_R_N+0x16	Default: 0x0000_0000
LBU_P2_OF_H	W/R	Addr.: LBU_CS_R_N+0x26	Default: 0x0000_0000
LBU_P3_OF_H	W/R	Addr.: LBU_CS_R_N+0x36	Default: 0x0000_0000
Description		High word of LBU Pagex_Offset_register	
Bit No.	Name	Description	
15..0		Upper 16 bits for offset setting	

LBU_P0_CFG	W/R	Addr.: LBU_CS_R_N+0x08	Default: 0x0000_0000
LBU_P1_CFG	W/R	Addr.: LBU_CS_R_N+0x18	Default: 0x0000_0000
LBU_P2_CFG	W/R	Addr.: LBU_CS_R_N+0x28	Default: 0x0000_0000
LBU_P3_CFG	W/R	Addr.: LBU_CS_R_N+0x38	Default: 0x0000_0000
Description		Configuration for the individual pages	
Bit No.	Name	Description	
15..1		Reserved	
0	PAGE_X_32	1: Page is a 32-bit page 0: Page is a 16-bit page	

8 PCI Interface

The AHB-PCI bridge of Fujitsu-Siemens is used as the PCI interface. A 2-Gbyte segment starting at address 0x80000000 (offset = 2 Gbytes) is on the AHB bus. See also "Detailed Memory Description" in Section 9.2. The configuration area of the PCI macro is addressed here. Mapping of addresses of the AHB to the address area of the PCI bus can be set in the configuration area. For a description of the PCI bridge, refer to /3/.

The bus system is selected using the CONFIG[2] input pin.

CONFIG[2] = 1 PCI bus system is active

The 32-bit PCI interface operates at a maximum frequency of 66 MHz.

The following signal pins are available for the PCI interface on the ERTEC 400.

• Data/address bus	32-bit	AD[31 : 0]
• PCI power management	1	PME_N
• PCI control cables	11	PAR, FRAME_N, IRDY_N, TRDY_N, DEVSEL_N, STOP_N, IDSEL, PERR_N, REQ_N, GNT_N, M66EN
• PCI bus clock	1	CLOCK_PCI
• PCI bus reset	1	RES_PCI_N
• CS for 4 address spaces	4	CBE0_N – CBE3_N
• Interrupt outputs	3	INTA_N, INTB_N, SERR_N

8.1 **PCI Functionality**

The PCI functions are described in general terms in this section.

8.1.1 **General Functions of the PCI Interface:**

- Compliant with PCI Specification 2.2
- Host functionality
- Master/target interface
- 32-bit AHB interface
- 32-bit PCI interface
- 3.3 V supply (5 V-compatible)
- Maximum operating frequency of 66 MHz
- Loading of PCI configuration registers from ARM946 processor
- 2 PCI interrupt outputs (INTA_N and INTB_N)
- Power Management Version 1.1

- **No** PCI interrupt inputs
- **No** support of lock transfers

8.1.2 **PCI Master Interface:**

The following accesses are supported:

- Memory Read
- Memory Read Line
- Memory Read Multiple
- Memory Write Single/Burst
- Memory Write and Invalidate
- 1 delayed instruction queue
- Write-data FIFO with a depth of 16
- Delayed read-data FIFO with a depth of 16
- Configuration area is loaded from the ARM946 (not from an EPROM)

8.1.3 PCI Target Interface:

The following accesses are supported:

- Memory Read
- Memory Read Line
- Memory Read Multiple
- Memory Write
- Memory Write and Invalidate
- Configuration Read/Write (Type 0)
- I/O Read/Write
- 6 ERTEC 400-internal address spaces
- 1 IRT communication function
- 1 delayed instruction queue
- Write-data FIFO with a depth of 16
- Delayed read-data FIFO with a depth of 16
- Configuration area is loaded from the ARM946 (not from an EPROM)

8.1.4 PCI Interrupt Handling:

Interrupt outputs INTA_N, INTB_N, and SERR_N are available at the PCI interface.

An interrupt request from the ARM946 to the PCI bus takes place by write accessing the interrupt controller integrated in the IRT switch. The interrupts are masked and saved in the IRT switch. This enables operation of a mailbox from the ARM946 to the PCI host.

An interrupt request from the PCI bus to the ARM946 takes place by write accessing an IRT switch-internal register. The interrupts are masked and saved in the interrupt controller of the ARM946. This enables operation of a mailbox from the PCI host to the ARM946.

When PC-based systems are linked, only INTA_N is used (only 1 function of the PCI bridge). The INTB_N output is not used. When an embedded host processor is linked, both INTA_N and INTB_N can be used. The local ARM processor is linked via the IRQ0_SP and IRQ1_SP interrupts. Both are linked via the internal logic structure of the interrupt controller unit (ICU) to the IRQ input of the ARM946E-S. Optionally, the option exists to link these interrupts to the ARM946E-S by assigning the ICU as FIQ.

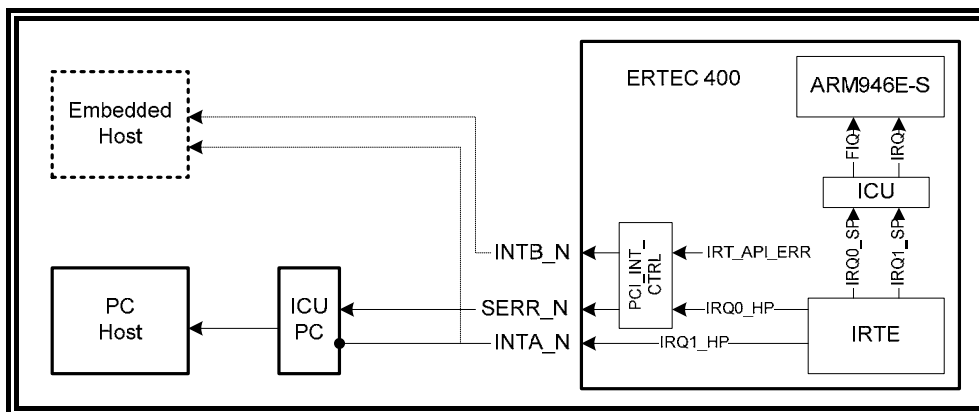


Figure 16: PCI Interrupt Handling

After a reset, PCI signal SERR_N is only generated from the PCI bridge (address parity error). IRT interrupts IRQ0_HP and IRQ_IRT_API_ERR (synchronization problems in the IRT-API) can be enabled according to the setting in the "PCI_INT_CNRL" system control register. Interrupt output SERR_N is a PCI-synchronous signal. When used, the IRQ0_HP and IRQ_IRT_API_ERR interrupts are synchronized to the PCI clock and kept active for the duration of one PCI clock cycle.

8.1.5 PCI Power Management:

Requests to change the power state are made by the PCI host in the “**PM_Control_Status**” register in the PCI configuration area. The ARM946 can read the requested power state in the “**PM-State-Req**” register in the system control register area. The current power state is stored in the “**PM_State_Ack**” register in the PCI configuration area. If the requested power state differs from the current power state, the PCI macro module issues an interrupt to the interrupt controller of the ARM946. The ARM946 can then change the local power state to the requested power state by writing to the “**PM_State_Ack**” register. The PCI host can then scan the current state by reading the “**PM_Control_Status**” register.

Output signal PME_N can be activated by writing to the “**PME_REG**” register in the system control register area. However, this requires that the “PME Enable” bit be set in the “**PM_Control_Status**” register in the PCI configuration area. The ERTEC 400 requires the PCI clock to initiate the “PME_N” signal.

If the power state of the ERTEC 400 is **not D0**, all PCI interrupts in the ERTEC 400 must be disabled via the software (according to the PCI specification, PCI interrupts can only be enabled in the D0 state). There is no hardware support for disabling the interrupts.

8.1.6 Accesses to the AHB Bus:

When accesses to the AHB bus are “not aligned,” the AHB-PCI bridge behaves as follows:

- Not-Aligned Posted Writes are separated on the AHB side into 2 individual accesses with a lock.
- Not-Aligned Delayed Reads are issued on the AHB side as full 32-bit accesses.
- All other not-aligned accesses are rejected with Target Abort.

8.2 ERTEC 400 Applications with PCI:

- ERTEC 400 on a PC card with use of power management functionality
- ERTEC 400 used with PCI bus as the local bus

8.2.1 ERTEC 400 in a PC System

In PC systems, the ERTEC 400 can be integrated into the system as a PC card with host and master functionality. The PC card can be operated on the 32-bit bus at a maximum of 66 MHz. Interrupt INTB_N must be disabled because the PC interface of the ERTEC 400 is only a “single function interface.” However, in order for a low-priority interrupt to be interrupted by a high-priority interrupt from the IRT switch, the IRQO_HP interrupt can be output on the SERR_N output. In the case of PC processors, this results in an NMI interrupt. Optionally, IRT interrupt IRQ_IRT_API_ERR can also be placed on the SERR_N output.

Power management support is an additional requirement. In such applications, the PC card can be operated in different power modes. The PC card is also able to generate certain events in the host system that “wake up” the host system (WAKE function). If the Wake function is employed for the PC card, it must be ensured that the PC card can still operate at “reduced” function in power-down state and an interrupt can be generated via the PME_N output.

The following sequence illustrates the functioning of a power management state:

- A power-down state (e.g., D3hot) is requested by the PC host.
- Interrupt is requested at the local processor.
- Current state is backed up with the corresponding registers and the requested state is set by the ERTEC 400.
- PCI interrupts are disabled.
- Requested state is confirmed.
- PCI bus is powered down and functions are disabled by the PC host.
- Data traffic is monitored for a certain event by the ERTEC 400.
- The “PME” interrupt is triggered when the monitored event is detected.
- PC host system is powered up when the individual devices request the Power-UP state.
- Backed-up state is restored by the ERTEC 400.
- The last state is established in the PC host system.
- Transition to requested “Power UP” power state occurs.

8.2.2 ERTEC 400 as a Station on the Local PCI Bus

In contrast to PC systems, the configuration can be changed during operation in local PCI systems where a host is active on the PCI system. In systems where a PCI interface is implemented only between several ERTEC 400s, one of the ERTEC 400s can also assume the host function. This enables very simple single master communication to be established between several ERTEC 400s via the PCI bus.

In the case of local onboard PCI bus systems, the IRQ0-HP interrupt (high-priority IRT interrupt) can be switched to the INTB_N output because an independent software structure is provided in this case.

For consistent non-aligned accesses where IRT switch data must be accessed, consistency assurance for non-aligned accesses is implemented. The alignment is signaled via accesses to the various mirror areas of the communication RAM. The IRT switch evaluates these mirror areas, thus ensuring consistency for unaligned 16-bit and 32-bit accesses.

8.3 Address Assignment of PCI Register

The PCI registers are **32 bits in width**. The registers can be read or written to with 8-bit, 16-bit, or 32-bit accesses.

PCI (Base Address 0x8000_0000)						
Offset Address	Byte 3	Byte 2	Byte 1	Byte 0	Access	Default
0x0000	Device_ID		Vendor_ID		R	0x4026110A
0x0004	Status		Command		R/W	0x02300000
0x0008	Class_Code			Revision_ID	R	0x000020A0
0x000C	BIST	Header_Type	Latency_Timer	Cache_Line_Size	R/W	0x00000000
0x0010	PCI_Base_Address_Register0				R/W	0x00000000
0x0014	PCI_Base_Address_Register1				R/W	0x00000000
0x0018	PCI_Base_Address_Register2				R/W	0x00000000
0x001C	PCI_Base_Address_Register3				R/W	0x00000000
0x0020	PCI_Base_Address_Register4				R/W	0x00000000
0x0024	PCI_Base_Address_Register5				R/W	0x00000000
0x0028	Cardbus_CIS_Pointer				R	0x00000000
0x002C	Subsystem_ID		Subsystem_Vendor_ID		R	0x00000000
0x0030	Expansion_ROM_Base_Address (Reserved)				R	0x00000000
0x0034	Reserved			Capability_Pointer	R	0x00000048
0x0038	Reserved				R	0x00000000
0x003C	Max_Lat	Min_Gnt	Interrupt_Pin	Interrupt_Line	R/W	0x00000000
0x0040	Device_ID		Vendor_ID		R/W	0x00000000
0x0044	Subsystem_ID		Subsystem_Vendor_ID		R/W	0x00000000
0x0048	PM_Capability		PM_next_item_ptr	PM_Capability_id	R	0x00020001
0x004C	PM_Data	PM_CSR_BSE	PM_Control_Status		R/W	0x00000000
0x0050	PCI_Base_Address_Mask_Register0				R/W	0x00000000
0x0054	PCI_Base_Address_Mask_Register1				R/W	0x00000000
0x0058	PCI_Base_Address_Mask_Register2				R/W	0x00000000
0x005C	PCI_Base_Address_Mask_Register3				R/W	0x00000000
0x0060	PCI_Base_Address_Mask_Register4				R/W	0x00000000
0x0064	PCI_Base_Address_Mask_Register5				R/W	0x00000000
0x0068	PCI_Base_Address_Translation_Register0				R/W	0x00000000
0x006C	PCI_Base_Address_Translation_Register1				R/W	0x00000000
0x0070	PCI_Base_Address_Translation_Register2				R/W	0x00000000
0x0074	PCI_Base_Address_Translation_Register3				R/W	0x00000000
0x0078	PCI_Base_Address_Translation_Register4				R/W	0x00000000
0x007C	PCI_Base_Address_Translation_Register5				R/W	0x00000000
0x0080	Reserved		PCI_Arbiter_Config_Register		R/W	0x00000000
0x0084	Max_Lat	Min_Gnt	INT_Pin	Reserved	R/W	0x00000000
0x0088	PM_Capability		Reserved		R/W	0x00020000
0x008C	Class_Code			Revision_ID	R/W	0x00000000
0x0090	AHB_Base_Address_Register0				R/W	0x00000001
0x0094	AHB_Base_Address_Register1				R/W	0x00000000
0x0098	AHB_Base_Address_Register2				R/W	0x00000000
0x009C	AHB_Base_Address_Register3				R/W	0x00000000
0x00A0	AHB_Base_Address_Register4				R/W	0x00000000

0x00A4	AHB_Base_Address_Mask_Register0				R	0xBFFF0001
0x00A8	AHB_Base_Address_Mask_Register1				R/W	0x3F000007
0x00AC	AHB_Base_Address_Mask_Register2				R/W	0x00000000
0x00B0	AHB_Base_Address_Mask_Register3				R/W	0x00000000
0x00B4	AHB_Base_Address_Mask_Register4				R/W	0x00000000
0x00B8	Reserved					
0x00BC	Reserved					
0x00C0	AHB_Base_Address_Translation_Register2				R/W	0x00000000
0x00C4	AHB_Base_Address_Translation_Register3				R/W	0x00000000
0x00C8	AHB_Base_Address_Translation_Register4				R/W	0x00000000
0x00CC	AHB_Status_Register		AHB_Function_Register		R/W	0x00000000
0x00D0	Wait_States_Bridge_as_PCI_Target	Wait_States_Bridge_PCI_Master	Wait_States_Bridge_as_AHB_Slave	Wait_States_Bridge_AHB_Master	R/W	0x00000000
0x00D4	Bridge_Interrupt_Status_Register				R	0x00000000
0x00D8	AHB_Interrupt_Enable_Register				R/W	0x00000000
0x00DC	PCI_Interrupt_Enable_Register				R/W	0x00000000
0x00E0 - 0x00F4					
0x00F8	SERR_Generation_By_Software				W	0x00000000
0x00FC	Enable_Configuration_From_PCI				R/W	0x00000000

Table 27: Overview of PCI Registers

8.4 PCI Register Description

A detailed description of the individual PCI registers can be found in /3/.

9 Memory Description

This section presents a detailed description of the memory areas of all integrated function groups.

9.1 Memory Partitioning of the ERTEC 400

The table below lists the AHB masters along with their options for accessing various memory areas.

Start and End Address	Segment	Function Area for ARM9	Function Area for IRT	Function Area for PCI/LBU
0000 0000	0	Internal Boot ROM or internal RAM	Internal Boot ROM or internal RAM	Internal Boot ROM or internal RAM
0FFF FFFF				
1000 0000	1	IRT switch	Not used	IRT switch
1FFF FFFF				
2000 0000	2	EMIF (SDRAM)	EMIF (SDRAM)	EMIF (SDRAM)
2FFF FFFF				
3000 0000	3	EMIF (asynchr. Memory Area Bank 0-3)	EMIF (asynchr. Memory Area Bank 0-3)	EMIF (asynchr. Memory Area Bank 0-3)
3FFF FFFF				
4000 0000	4	All APB macros incl. boot ROM	Not used	All APB macros incl. boot ROM
4FFF FFFF				
5000 0000	5	ARM-ICU	Not used	Not used
5FFF FFFF				
6000 0000	6	Internal SRAM	Internal SRAM	Internal SRAM
6FFF FFFF				
7000 0000	7	EMIF register	Not used	EMIF register
7FFF FFFF				
8000 0000	8 - 15	PCI	PCI	Not used
FFFF FFFF				

Table 28: Partitioning of Memory Areas

9.2 Detailed Memory Description

The table below presents a detailed description of the memory segments. Mirrored segments should not be used for addressing to ensure compatible memory expansion at a later date. When a locked I-cache and D-TCM are used, they can only be addressed by the ARM946E-S and not by PCI or IRT.

When the I-cache is used, it cross-fades the first 4 Kbytes (0-4k) of the memory area. The D-TCM memory can be placed anywhere in the address space of the ARM946E-S.

Segment	Contents	Size	Address Area	Description
0	Internal Boot ROM or internal RAM	256 Mbytes	0000_0000 - 0FFF_FFFF	After reset: Boot ROM (8 Kbytes, physical) After memory swap: internal USER RAM (8 Kbytes, physical) Note2
1	IRT switch	256 Mbytes	1000_0000 - 1FFF_FFFF	8 Mbytes, physical: * 0-1 Mbyte for registers * 1-2 Mbytes for communication RAM (192 Kbytes) Note1
2	EMIF (SDRAM)	256 Mbytes	2000_0000 - 2FFF_FFFF	When a smaller memory area is used, mirroring over the entire area
3	EMIF I/O Bank 0	16 Mbytes	3000_0000 - 30FF_FFFF	When a smaller device is interfaced, mirroring over the entire 16 Mbytes
	EMIF I/O Bank 1	16 Mbytes	3100_0000 - 31FF_FFFF	When a smaller device is interfaced, mirroring over the entire 16 Mbytes
	EMIF I/O Bank 2	16 Mbytes	3200_0000 - 32FF_FFFF	When a smaller device is interfaced, mirroring over the entire 16 Mbytes
	EMIF I/O Bank 3	16 Mbytes	3300_0000 - 33FF_FFFF	When a smaller device is interfaced, mirroring over the entire 16 Mbytes
	Not used		3400_0000 - 3FFF_FFFF	
4	Internal boot ROM	8 Kbytes	4000_0000- 4000_1FFF	8 Kbytes, physical
	Timer	256 bytes	4000_2000 - 4000_20FF	32 bytes, physical Note2
	Watchdog	256 bytes	4000_2100 - 4000_21FF	28 bytes, physical Note2
	SPI	256 bytes	4000_2200 - 4000_22FF	256 bytes, physical
	UART1	256 bytes	4000_2300 - 4000_23FF	256 bytes, physical
	UART2	256 bytes	4000_2400 - 4000_24FF	256 bytes, physical
	GPIO	256 bytes	4000_2500 - 4000_25FF	32 bytes, physical Note2
	System control register block	256 bytes	4000_2600 - 4000_26FF	164 bytes, physical General register block Note2
	F-counter	256 bytes	4000_2700 - 4000_27FF	8 bytes, physical Note2
	Not used		4000_2800-4FFF_FFFF	
5	ARM-ICU	256 Mbytes	5000_0000- 5FFF_FFFF	ARM interrupt controller 128 bytes, physical Note2
6	Internal RAM	256 Mbytes	6000_0000- 6FFF_FFFF	Mirror area of internal RAM 8 Kbytes, physical Note2
7	EMIF register	256 Mbytes	7000_0000- 7FFF_FFFF	Control register for external memory interface 64 bytes, physical Note2
8-15	PCI bus	2 Gbyte	8000_0000- FFFF_FFFF	Access to PCI area: Maximum of 4 regions * <= 1 Gbyte Memory Prefetch * <= 1 Gbyte Memory non Prefetch * <= 1 Gbyte IO * 64 Kbytes internal register * 16 Mbytes PCI configuration register Note: For size and location of areas, see "AHB-PCI Bridge" Rev.2.5, 2002, Fujitsu Siemens Computers

Table 29: Detailed Description of Memory Segments

Note:

1. Access to IRT registers and KRAM should only occur in the address areas indicated above (first 2 Mbytes). An access to areas within the 2 Mbytes that are not occupied by the IRT registers and KRAM result in undefined access (acknowledgement timeout). The read or written data are not valid. While the 2-Mbyte areas are mirrored within the 8-Mbyte physical address area, different access types are used:

- 2-4-Mbyte area for unaligned consistent 16-bit accesses to IRT
- 4-6-Mbyte area for unaligned consistent 32-bit accesses to IRT
- 6-8 Mbytes is not supported (supplies undefined values)

The 8-Mbyte address area is mirrored 32 times within the 256 Mbytes.

2. Memory areas are mirrored according to the following formula:

$$N = \frac{\text{Memory size}}{\text{Physical memory size}}$$

Physical memory size is limited to values of 2^n (2, 4, 8, ... 128, 256 etc.)

Example: The physical memory size of the watchdog is 28 bytes. However, 32 bytes are taken for calculating the number of mirrorings N. In this case, the number of mirrorings $N = 8$. Access to the 4 unused bytes does not result in an acknowledgement timeout, but the read or written values are undefined.

10 Test and Debugging

10.1 ETM9 Embedded Trace Macrocell

An ETM9 module is integrated in the ARM946E-S of the ERTEC 400 to enable instructions and data to be traced. The ARM946E-S supplies the ETM module with the signals needed to carry out the trace functions. The ETM9 module is operated by means of the Trace interface or JTAG interface. The trace information is stored in an internal FIFO and forwarded to the debugger via the interface.

10.1.1 Trace Modes

- Normal mode with 4- or 8-data bit width
- Transmission mode
 - Full rate mode at 50 or 100 MHz
 - Half rate mode at 150 MHz

10.1.2 Features of the ETM9 Module

The Medium version of the ETM9 module is in the ERTEC 400. That is, it has the following features:

- 4 address comparators
- 2 data comparators with filter function
- 4 direct trigger inputs, one of which can be connected via a GPIO port
- 1 trigger output that is also available at the GPIO port for external purposes
- 8 memory map decoders for decoding the physical address area of the ERTEC 400
- 1 sequencer
- 2 counters

Supplemental to the ETM specification, the 8 MMD regions have been decoded out via the hardware:

- 0k – 4k : Instruction cache (I-cache)
- 4k – 8k : Data cache (D-TCM)
- 8k – 256M : All accesses (User RAM in all mirrored areas)
- 256M – 257M : Data (IRT registers in all mirrored areas)
- 257M – 258M : All accesses (communication RAM in all mirrored areas)
- 512M – 768M : All accesses (SDRAM in all mirrored areas)
- 784M – 800M : All accesses (EMIF of CS1)
- 2G – 4G : Data (PCI in the entire 2-Gbyte area)

For more information on the ETM, refer to **Section 9** of /1/.

10.1.3 ETM9 Registers

The ETM registers are not described in this document because they are handled differently according the ETM version being used.

For a detailed description, refer to /7/.

10.2 Trace Interface

The trace interface is parameterized, enabled, and disabled by means of a connected debugger (e.g. by Lauterbach) on the JTAG interface.

A Trace port is provided in the ERTEC 400 for tracing internal processor states:

- PIPESTA [2:0]
- TRACESYNC
- TRACECLK
- TRACEPKT[7:0]

The TRACEPKT[7:0] signals are alternative signal pins at the GPIO port. The Trace interface can be assigned a data width of 4 bits or 8 bits in the debugger. If a data width of 4 bits is assigned, the TRACEPKT[3:0] signals at GPIO port[11:8] are automatically switched to trace function. If a data width of 8 bits is assigned, the TRACEPKT[7:4] signals at GPIO port[21:18] are also switched to trace function.

For connectors, pinning, and hardware circuitry for the Trace interface, refer to /7/.

10.3 JTAG Interface

Besides the debug function, the JTAG interface is also used for the boundary scan (see /9/). In addition to the JTAG interface, the DBGREQ and DBGACK signals are available as alternative function pins for debugging. Due to the different debuggers, an internal pull-up resistor at the TRST_N JTAG pin is not included. The user has to ensure the proper circuitry for the utilized debugger .

The standard connector for JTAG interfaces is a 20-pin connector with a pin spacing of 0.1 inch. All JTAG pins and the two additional DBGREQ and DBGACK pins are connected here. The connector is assigned as follows:

Function	Pin No.	Pin No.	Function
Vcc-Sense	1	2	Vcc
TRST_N	3	4	GND
TDI	5	6	GND
TMS	7	8	GND
TCK	9	10	GND
RTCK (#1)	11	12	GND
TDO	13	14	GND
RST (#1)	15	16	GND
DBGREQ	17	18	GND
DBGACK	19	20	GND

Table 30: Pin Assignment of JTAG Interface

For connectors, pinning, signal description, and hardware circuitry for a standard JTAG interface for the multi-ICE debugger, for example, refer to /8/.

In addition to the standard JTAG connector, the pins can also be connected to the Trace interface.

For connectors, pinning, and hardware circuitry for JTAG signals at the Trace interface, refer to /7/.

10.4 Debugging via UART1

If UART1 is not used for user-specific tasks, it can also be used as a debugging interface.

11 Miscellaneous

11.1 Acronyms/Glossary:

AHB	AMBA Advanced High Performance Bus (Multimaster, Bursts)
AMBA	Advanced Microcontroller Bus Architecture
APB	AMBA Advanced Peripheral Bus (Single master, bursts)
BIST	Built In Self Test
ComDeC	C ommunication, D evelopment & C ertification
DTCM	D ata T ightly C oupled M emory
ERTEC	Enhanced Real-Time Ethernet Controller
EMIF	External Memory Interface
ETM	Embedder Trace Macrocell
FIQ	Fast Interrupt Request
GPIO	General Purpose Input/Output
ICE	In Circuit Emulator
ICU	Interrupt Controller Unit
IRQ	Interrupt Request
IRT	Isochronous Real Time
ITCM	Instruction Tightly Coupled Memory
JTAG	Joint Test Action Group
LBU	Local Bus Unit
MAC	Media Access Controller
MII	Media Independent Interface
MPU	Memory Protection Unit
NRT	Non Real Time
PCI	Peripheral Communication Interface
PD	Pull Down
PU	Pull Up
RMII	Reduced Media Independent Interface
RT	Real Time
SPI	Standard Serial Peripheral Interface
SW	Software
UART	Universal Asynchronous Receiver / Transmitter
WS	Warteschlange (queue)

11.2 References:

- /1/ Technical Reference Manual ARM946E-S REV1 16 February 2001 (DDI_0201A_946ES.PDF);
- /2/ Technical Reference Manual ARM946E-S 16 December 1999 (DDI_0165A_9E-S_TRM.PDF);
- /3/ AHB PCI Bridge Revision2.5 08 July 2002 (amba2pci_rev2.5.pdf);
- /4/ AMBA Specification (Revision 2.0), 1999; ARM
- /5/ ARM Prime Cell™ UART (PL010) Technical Reference Manual; ARM
- /6/ ARM Prime Cell™ Synchronous Serial Port (PL021) Technical Reference Manual;
- /7/ Embedded Trace Macrocell Architecture Specification (ETM_Spec.PDF);
- /8/ Multi-ICE System Design Consideration Applic.-Note 72 (DAI0072A_Multiicedesign-Notes.PDF);
- /9/ IEEE Standard Test Access Port and Boundary-Scan Architecture (1149.1 IEEE Boundary Scan 2001.PDF);
- /10/ IR35-107-3.pdf
- /11/ LeadfreeIR50_60.pdf
- /12/ Codeexpl.pdf
- /13/ EB 400 Manual V1.2.0 (EB400_Manual_V120.PDF);