SIEMENS

ERTEC 200P-3

Enhanced Real-Time Ethernet Controller

Manual

Edition 10/2023

Disclaimer

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions. Suggestions for improvement are welcome.

This manual contains hyperlinks to the websites as well as to electronic documents of third-party companies. Siemens cannot be held responsible for the contents of these websites and electronic documents, nor does Siemens adopt these web sites and electronic documents and their contents as its own. You therefore use these links at your own risk. Since Siemens is not responsible for linked contents and information on the websites and electronic documents of third parties, this information is not checked by Siemens.

Copyright © Siemens AG 2023. 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 owners and must be treated as such.

Technical data subject to change.

Changelog

Version	Changes
1.1	Power consumption of ERTEC 200P-3 is 0.78 W

Preface

Target audience of this manual

This manual is intended for hardware and software developers who want to use the ERTEC 200P for new products. Experience working with processors and designing embedded systems and knowledge of Ethernet are required for this.

The manual describes the ERTEC function groups in detail and provides information that you must consider when configuring your own PROFINET IO device hardware.

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

Structure of this manual

Chapter 1 Introduction

Chapter 2 Description of functions

Chapter 3 Miscellaneous

Scope of the manual

This manual is valid only for ERTEC 200P-3. For information on ERTEC 200P-1/2, refer to the respective manual.

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 figures and tables in the manual are provided at the beginning of the manual.
- A list of abbreviations used in the manual is located at the end of the manual.
- 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.

Conventions

ERTEC 200P, ERTEC 200P-3: We refer to "ERTEC 200P" or "ERTEC 200P-3" in this documentation as a synonym for ERTEC 200P Step 3.

PN IO stack, PN stack: We refer to "PN IO stack" or "PN stack" in this documentation as a synonym for PROFINET stack.

This documentation contains pictures of the devices described. The figures may differ slightly from the device supplied.

You should also pay particular attention to notes such as the ones shown below:

Note

A note contains important information on the product, on handling of the product and on the section of the documentation to which you should pay particular attention.

NOTICE

indicates that property damage can result if proper precautions are not taken.

Copyright © Siemens AG 2023. All rights reserved

What's new in the ERTEC 200P-3 manual, Edition 03/2023 compared to Edition 10/2016 of the ERTEC 200P-2 manual?

The ERTEC 200P Step 3 described in this manual is designed as a functionally compatible replacement for ERTEC 200P Step 1, ERTEC 200P Step 2 and ERTEC 200, and can supersede all ERTEC 200P Step 1/2 and ERTEC 200 applications in the future.

ERTEC 200P Step 3 supports the following additional features as compared to ERTEC 200P Step 1/2:

Function / Module	Content / Benefits	Where can I find this information?
Integrated Reed-Solomon Coder/Decoder	Increases reliability of boot media.	See chapter 2.3.9
Interface for Octal-SPI flash	Supports only Quad-SPI boot.	See chapter 2.3.14.8 and ERTEC 200P-3 data sheet
Same PN-IP as in ERTEC 200P-2	With PROFINET Device Development Kit V5.1, the same software can be used for ERTEC 200P-2 and ERTEC 200P-3.	-
Reduction in power dissipation	Lower power consumption: ERTEC 200P-2: 1.23 W ERTEC 200P-3: 0.78 W.	Refer to ERTEC 200P-3 data sheet
Power supply	Simplified chip supply: ERTEC 200P-2: needs 4 different supply rails ERTEC 200P-3: needs only 3 different supply rails.	Refer to ERTEC 200P-3 data sheet
Package size	Same as ERTEC 200P-2.	Refer to ERTEC 200P-3 data sheet
Pinning	Optimized pinout – new PCB layout.	Refer to ERTEC 200P-3 data sheet
Bridge delays	Same as ERTEC 200P-2.	Refer to PROFINET Device Development Kit for ERTEC 200P (6ES7195-3BE00-0YA1)
Port delays	ERTEC 200P-2: - MaxPortRxDelay = 198 ns - MaxPortTxDelay = 6 ns ERTEC 200P-3: - MaxPortRxDelay = 237 ns - MaxPortTxDelay = 21 ns	Refer to PROFINET Device Development Kit for ERTEC 200P (6ES7195-3BE00-0YA1)
LPSDRAM	Same as ERTEC 200P-2.	Refer to ERTEC 200P-3 data sheet
Packing unit	Optimized packing units, only Tape and Reel.	Refer to ERTEC 200P-3 data sheet
SCRB2	Additional System Control Register Block.	See chapter 2.3.14.11 and ERTEC 200P-3 data sheet

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Security information

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

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

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g., firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.

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

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under https://www.siemens.com/cert.

Technical Support

Technical information and support on development packages are provided by Support Team:

Siemens Sanayi ve Ticaret A.Ş.
Office Address:
Yakacık Caddesi No 111
34870 Istanbul, Turkey
Email: profinet.devkits.industry@siemens.com

PROFINET Certification

Comprehensive consulting prior to starting the development project is particularly important for an efficient and successful development process. The final step in field device development is certification. For PROFINET, Siemens also offers you independent, accredited test laboratories like "ComDeC" in Germany, "PIC" in the USA, "ITEI" in China, and "an testlabs" in the Czech Republic.

Germany and Europe

Siemens AG
Communication, Development & Certification (ComDeC)
Breslauer Strasse 5
90766 Fürth, Germany
Tel.: +49 (911) 750-2080
Email: comdec@siemens.com
Web: www.siemens.com/comdec

Europe

An testlabs
Zelený pruh 1560/99
140 00, Praha 4, Czech Republic
Tel.: +420 244 091 111
Email: antestlabs.cz@siemens.com
Web: https://new.siemens.com/cz/cs/products/sluzby-a-servis/an-testlabs.html

Copyright © Siemens AG 2023. All rights reserved

USA

PROFI Interface Center (PIC)
One Internet Plazza
PO Box 4991
Johnson City, TN 37602-4991, USA
Tel.: +1 (423) - 262 - 2576
Email: pic.industry@siemens.com
Web: www.profiinterfacecenter.com

China

Siemens Ltd., China
Factory Automation Department
7, Wang Jing Zhong Huan Nan Lu, Chaoyang District, Beijing 100102, PR. China
Email: profinet.cn@siemens.com

Table of contents

1 INTRODUCTION 15

1.1 Overview of the ERTEC 200P 15
1.1.1 NRT (Non-Real-Time) communication 15
1.1.2 RT (Real-Time) communication 16
1.1.3 IRT (Isochronous Real-Time) communication 16

1.2 ERTEC 200P data 16
1.2.1 System reliability 17
1.2.2 Electromagnetic compatibility (EMC) 17

1.3 ERTEC 200P use cases 17
1.3.1 Use Case 1 (UC1): Operation with external host 18
1.3.2 Use Case 2 (UC2): Operation without external host 19

1.4 Application notes 20
1.4.1 EMC SDRAM interface 20
1.4.2 No free-running frequency at quartz break 20
1.4.3 16-bit data access over XHIF in buffered mode 21

2 DESCRIPTION OF FUNCTIONS 22

2.1 Block diagram 22

2.2 General function description 22

2.3 Description of individual ERTEC 200P blocks 22
2.3.1 Overview 22
2.3.2 Register descriptions 23
2.3.3 Processor subsystem (ARM926) 24
2.3.3.1 Features 24
2.3.3.2 Block diagram 25
2.3.3.3 ARM926EJ-S processor 25
2.3.3.3.1 Cache structure of ARM926EJ-S 25
2.3.3.3.2 ARM926 Tightly Coupled Memories (ARM926_TCM) 26
2.3.3.3.3 Memory Management Unit (MMU) 27
2.3.3.3.4 Bus interface of the ARM926 Processor 28
2.3.3.3.5 ARM926 Embedded Trace Macrocell (ETM9), Trace Buffer (ETB11) 28
2.3.3.4 Debug support 29
2.3.3.4.1 Debug configuration 29
2.3.3.4.2 ARM926EJ-S Debug interface 29
2.3.3.5 Boot process for the ERTEC 200P 30
2.3.3.5.1 Primary Boot Loader 32
2.3.3.5.2 Boot mode 1 … 2 (booting with NOR flash) in compile mode 34
2.3.3.5.3 Boot mode 7 (Booting with External Host) 35
2.3.3.5.4 Boot mode 9 (Booting with Octal-SPI – with RS coding) 37
2.3.3.6 Secondary Boot Loader 40
2.3.3.6.1 Memory swapping by the Secondary Boot Loader 40
2.3.4 AMBA (Internal Bus System) 41
2.3.4.1 Characteristics of the bus system in ERTEC 200P 41
2.3.4.2 Endianness 41
2.3.4.3 AHB subsystem (Multi-Layer AHB) 41
2.3.4.3.1 AHB Arbiters 42
2.3.4.3.2 AHB Multi-layer configuration 42
2.3.4.3.3 AHB Burst Breaker 43
2.3.4.4 APB subsystem (Peripheral Bus) 44
2.3.4.4.1 AHB/APB bridge 45

2.3.4.4.2 APB decoder 47
2.3.4.5 Address Range and Acknowledgement Delay Monitoring 47
2.3.4.5.1 Monitoring at the AHB side 47
2.3.4.5.2 Monitoring at the APB side 48
2.3.4.5.3 Monitoring at the EMC 48
2.3.4.5.4 Monitoring in the individual modules 49
2.3.5 ARM-ICU (ARM926 Interrupt Controller) 51
2.3.5.1 Overview 51
2.3.5.2 Interface 51
2.3.5.3 Interrupt acquisition 52
2.3.5.4 Interrupt masking / priorisation 53
2.3.5.5 Interrupt post-processing 54
2.3.5.6 Special functions 55
2.3.5.7 Debug functions 55
2.3.5.8 Miscellaneous 56
2.3.5.8.1 Synchronizing the inputs 56
2.3.5.8.2 Bus interface 56
2.3.5.8.3 Sequences 56
2.3.5.8.4 Operating rules 57
2.3.5.8.5 Startup/shutdown 58
2.3.5.9 Interrupt sources for ARM-IRQ 59
2.3.5.9.1 Interrupts for accesses to missing addresses 62
2.3.5.9.2 Confirmation delay in the Memory Controller (EMC) address area 62
2.3.5.10 Interrupt sources for ARM-FIQ 62
2.3.5.10.1 High-priority interrupts for debugging 63
2.3.5.11 Address mapping 64
2.3.5.12 Register description 68
2.3.6 Host ICU (PerIF: Event Unit) 88
2.3.7 GDMA (Direct Memory Access) 88
2.3.7.1 GDMA function description 88
2.3.7.1.1 AHB interfaces 89
2.3.7.1.2 Job priorities 89
2.3.7.1.3 Details 89
2.3.7.1.4 Usage 96
2.3.7.1.5 Result 97
2.3.7.1.6 Memory 98
2.3.7.1.7 Interrupts 98
2.3.7.2 ERTEC 200P GDMA use cases 99
2.3.7.2.1 Using the Job Start and HW_REQ Signal List 99
2.3.7.2.2 DMA with SPI interface 103
2.3.7.2.3 DMA with UART interface 106
2.3.7.2.4 DMA with Octal-SPI 107
2.3.7.3 GDMA-IP bugs 109
2.3.7.4 Address mapping 110
2.3.7.5 Register description 112
2.3.7.6 DMA RAM 131
2.3.8 EMC (External Memory Controller) 134
2.3.8.1 Block diagram of the EMC 135
2.3.8.2 EMC Reset and Clock 135
2.3.8.3 AHB slave interface 137
2.3.8.4 EMC notes 137
2.3.8.4.1 Remapping 137
2.3.8.4.2 Maximum number of wait cycles 138
2.3.8.4.3 Shift mode with the asynchronous EMC interface 138
2.3.8.4.4 Control of an external driver 138
2.3.8.4.5 QVZ Acknowledgement Delay 139
2.3.8.4.6 Configuration of asynchronous EMC interface 139
2.3.8.5 Timing of the Asynchronous Memory interfaces 139
2.3.8.5.1 Read access 139
2.3.8.5.2 Write access 140

Copyright © Siemens AG 2023. All rights reserved

2.3.8.6 Application examples 141
2.3.8.7 Address mapping 143
2.3.8.8 Register description 143
2.3.9 RSC 1 and 2 (Reed-Solomon Controllers) 153
2.3.9.1 Module description 153
2.3.9.2 Block diagram 154
2.3.9.3 Function description 154
2.3.9.3.1 Operating modes 155
2.3.9.3.2 Detailed description 159
2.3.9.4 Address mapping 164
2.3.9.5 Register description 164
2.3.10 Host Interface – Parallel (XHIF) 172
2.3.10.1 Block diagram of the Host Interface 172
2.3.10.2 Function description (XHIF) 173
2.3.10.2.1 AHB Master interface 176
2.3.10.2.2 APB Slave interface 176
2.3.10.2.3 XHIF configuration 176
2.3.10.2.4 Application information 178
2.3.10.3 Address mapping 180
2.3.10.4 Register description 181
2.3.11 Peripheral Interface 188
2.3.11.1 Interrupts for External Host 188
2.3.11.2 Peripheral (Application) AHB interface 189
2.3.11.2.1 Address mapping 189
2.3.11.2.2 Register description 190
2.3.11.3 APB interface 194
2.3.11.3.1 Address mapping 194
2.3.11.3.2 Register description 195
2.3.12 PROFINET-IP 199
2.3.12.1 PN-IP interfaces 199
2.3.12.1.1 AHB interface 199
2.3.12.1.2 Interrupt management 199
2.3.12.1.3 PN-ICU 200
2.3.12.1.4 PN-PLL (incl. 3 × Application Timer Blocks) 202
2.3.12.1.5 EDC in PN-IP RAMs 203
2.3.12.1.6 Recommended parameter assignment for ERTEC 200P 203
2.3.12.2 Ethernet PHY (integrated) 204
2.3.12.2.1 Status of the integrated PHY 205
2.3.13 CRU (Clock and Reset Unit) 206
2.3.13.1 Clock system 206
2.3.13.2 Reset system 206
2.3.14 APB Peripherals 209
2.3.14.1 Access modes and data widths 209
2.3.14.2 I-Filter 210
2.3.14.2.1 Operating principle of the RC filter 212
2.3.14.2.2 Address mapping 212
2.3.14.2.3 Register description 213
2.3.14.3 ARM926 Watchdog 223
2.3.14.3.1 Overview 223
2.3.14.3.2 Block diagram 223
2.3.14.3.3 Signal waveforms 225
2.3.14.3.4 Write protection of the Watchdog register 227
2.3.14.3.5 Starting the Watchdog 227
2.3.14.3.6 Notes 227
2.3.14.3.7 Address mapping 227
2.3.14.3.8 Register description 228
2.3.14.4 Timer 0 – 5 231
2.3.14.4.1 Overview 231
2.3.14.4.2 TIMER_TOP functionality 232
2.3.14.4.3 Timer module 235

Copyright © Siemens AG 2023. All rights reserved

2.3.14.4.4 Overview of the count modes 241
2.3.14.4.5 Timing requirements 244
2.3.14.4.6 Operating rules 244
2.3.14.4.7 Connections on ERTEC 200P top-level 245
2.3.14.4.8 Address mapping 246
2.3.14.4.9 Register description 248
2.3.14.5 F-Counter 261
2.3.14.5.1 Function description 261
2.3.14.5.2 Application information 261
2.3.14.5.3 Address mapping 262
2.3.14.5.4 Register description 262
2.3.14.6 UART1…4 (Universal Asynchronous Receiver/Transmitter) 263
2.3.14.6.1 UART baud rates 264
2.3.14.6.2 Address mapping 266
2.3.14.6.3 Register description 267
2.3.14.7 I²C (Inter-Integrated Circuit) 281
2.3.14.7.1 Baudrate generator 281
2.3.14.7.2 Important software rules 281
2.3.14.7.3 Address mapping 283
2.3.14.7.4 Register description 283
2.3.14.8 OSPI (Octal-SPI) 290
2.3.14.8.1 Configuration of OSPI 290
2.3.14.8.2 WR transfer monitoring 291
2.3.14.8.3 Clock shift 292
2.3.14.8.4 OSPI_IP is used without RXDS 293
2.3.14.8.5 OSPI_IP is used with RXDS 293
2.3.14.8.6 Memory swapping 295
2.3.14.8.7 Software usage in the OSPI 295
2.3.14.8.8 OSPI programming 296
2.3.14.8.9 Execution In Place (XIP) 299
2.3.14.8.10 Address mapping 299
2.3.14.8.11 Register description 301
2.3.14.9 SPI1/2 (Serial Peripheral Interface) 356
2.3.14.9.1 Features of the SPI interface 357
2.3.14.9.2 SSPMS IP extension 358
2.3.14.9.3 SSPMS IP integration 358
2.3.14.9.4 Send support with pause between individual characters 362
2.3.14.9.5 SPI Flash boot 363
2.3.14.9.6 Address mapping 364
2.3.14.9.7 Register description 364
2.3.14.10 GPIO (General Purpose Input/Output) 368
2.3.14.10.1 Features of the GPIO module 369
2.3.14.10.2 GPIO module function description 370
2.3.14.10.3 Integration of the GPIO module 371
2.3.14.10.4 GPIO pad multiplexing 372
2.3.14.10.5 GPIO assignment 372
2.3.14.10.6 Address mapping 377
2.3.14.10.7 Register description 378
2.3.14.11 SCRB (System Control Register Block) 387
2.3.14.11.1 Hardware Identifier register 387
2.3.14.11.2 Boot register 387
2.3.14.11.3 Config register 387
2.3.14.11.4 Reset registers 387
2.3.14.11.5 ACCESS_ERROR and QVZ register 388
2.3.14.11.6 PLL Status register 388
2.3.14.11.7 Memory swapping 388
2.3.14.11.8 ARM Control register 389
2.3.14.11.9 PHY register 389
2.3.14.11.10 AHB Burst Breaker register 389
2.3.14.11.11 GPIO Control register 389

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.11.12 I²C Clock Divider 390
2.3.14.11.13 EDC register 390
2.3.14.11.14 ARM926 mapping 391
2.3.14.11.15 PAD Control register 391
2.3.14.11.16 Clear DMA request 393
2.3.14.11.17 SPI parity error 394
2.3.14.11.18 XHIF mode 394
2.3.14.11.19 Ext. Driver Enable 394
2.3.14.11.20 SPI mode 394
2.3.14.11.21 SD signal handling 394
2.3.14.11.22 Extension mode 395
2.3.14.11.23 Setup register 395
2.3.14.11.24 Enabling of Octal-SPI 395
2.3.14.11.25 Reset register extensions 395
2.3.14.11.26 GDMA control 396
2.3.14.11.27 Address mapping 396
2.3.14.11.28 Register description 398

2.4 Memory mapping 444
2.4.1 Host IF memory mapping 444
2.4.2 Detailed address mapping 446

3 MISCELLANEOUS 451

3.1 Abbreviations 451

3.2 References 454

List of figures

Figure 1: ERTEC 200P use cases 18
Figure 2: Application operation with external host 19
Figure 3: Application operation without external host 20
Figure 4: ERTEC 200P Step 3 block diagram 22
Figure 5: ARM926 subsystem 25
Figure 6: TCM block assignment in the ARM926 subsystem 31
Figure 7: Flow chart for the external host for establishing XHIF interface enable 36
Figure 8: AHB/APB bridge, block diagram 46
Figure 9: APB bridge, timing 47
Figure 10: The principle of the DMA Job organization 88
Figure 11: DMA Jobs programming 91
Figure 12: Finite state machine diagram of the job control algorithm 95
Figure 13: GDMA interrupt in line with the user data to be transferred 105
Figure 14: DMA RAM address space 131
Figure 15: EMC block diagram 135
Figure 16: EMC interface with one SDRAM 136
Figure 17: EMC interface with asynchronous RAM 136
Figure 18: EMC, address space 137
Figure 19: Control of an external driver 138
Figure 20: EMC, notation definition for a read access (ASYNC) 140
Figure 21: EMC, notation definition for a write access (ASYNC) 140
Figure 22: EMC, application example: SDRAM 141
Figure 23: EMC, application example: Combination of SDRAM and asynchronous interfaces 142
Figure 24: RSC block diagram 154
Figure 25: Example: Storage of 6 data bytes (max. 238) and 16 check bytes in RSC-RAM 159
Figure 26: Example: Storage of 7 data bytes (max 238) and 16 check bytes in RSC-RAM 161
Figure 27: Block diagram of the host interface 172
Figure 28: Structure of the parallel XHIF with 8 pages 173
Figure 29: Diagram of interrupt management in the PN-IP 200
Figure 30: Block diagram of PN-ICU for PN_IRQx(0/1:0) group interrupts 201
Figure 31: Block diagram of PN-MUX for PN_IRQx(2:1/15:2) single interrupts 201
Figure 32: PNPLL with 3 application time blocks (application connection) 202
Figure 33: Block diagram of the IO filter 210
Figure 34: Block diagram of the filter structure of a channel 210
Figure 35: Block diagram of the central clock divider 211
Figure 36: Signal in signal filtering 212
Figure 37: Block diagram ARM926 Watchdog 223
Figure 38: XWD_OUT0/WD_INT signal sequence 225
Figure 39: XWD_OUT1 signal sequence 226
Figure 40: TIMER_TOP block diagram 232
Figure 41: TIMER block diagram 235
Figure 42: F-timer block diagram 261
Figure 43: OSPI clk shift 293
Figure 44: OSPI_IP without RXDS 293
Figure 45: OSPI reset sequence 296
Figure 46: Sequence of TX FIFO entries with a READ command 297
Figure 47: Sequence of TX FIFO entries with a PAGE_PROGRAM command 297
Figure 48: Sequence of TX FIFO entries with a FAST_READ command 298
Figure 49: Sequence of TX FIFO entries with a QUAD_INPUT_FAST_PROGRAM command 299
Figure 50: SSP_SHELL 359
Figure 51: SPI block diagram 360
Figure 52: SPI flash: serial output timing 363
Figure 53: SPI flash: serial input timing 363
Figure 54: Block diagram of a GPIO module 370
Figure 55: GPIO, IO circuit 372
Figure 56: SD signal handling 395

Copyright © Siemens AG 2023. All rights reserved

List of tables

Table 1: Types of ARM926EJ-S access to I/D-TCM / AHB 27
Table 2: Boot, ERTEC 200P boot modes 30
Table 3: Interrupt vector table 32
Table 4: Startup times 34
Table 5: Example 39
Table 6: Data size restriction 39
Table 7: Memory image in the I-TCM 39
Table 8: OSPI startup time 40
Table 9: Internal bus system, parameters of internal buses 41
Table 10: Access types to AHB slaves 42
Table 11: Fixed priority assignment (default) 42
Table 12: AHB master-slave coupling 43
Table 13: Access types to APB slaves 45
Table 14: 'Access_Error' register (of RS controller) 50
Table 15: Host INTA, interrupt sources 59
Table 16: Host INTB, interrupt sources 62
Table 17: HW trigger for GDMA 99
Table 18: HW flow control for GDMA 101
Table 19: RSC encoder mode 155
Table 20: RSC interrupt events as individual interrupts 163
Table 21: Possible page configuration of XHIF module 174
Table 22: XHIF pages with 1 MByte address range each 175
Table 23: XHIF pages with 2 MBytes address range each 175
Table 24: XHIF interface adjustment 177
Table 25: Interrupts for external host 188
Table 26: PNPLL_OUT connection matrix 202
Table 27: Recommended parameter assignment for time-triggered injection 203
Table 28: Recommended parameter assignment for CPM cut-through receipt 204
Table 29: ERTEC 200P reset matrix 207
Table 30: Data width of peripherals 209
Table 31: Sample filter times for I-Filter 211
Table 32: Assignment of the timer gate/trigger/event inputs to the timer multiplexers 246
Table 33: Baud rates UART when $F_{UARTCLK}$ = 125 MHz 265
Table 34: Configuration of OSPI 290
Table 35: Memory swapping 295
Table 36: Timing parameters for SPI flash (75 MHz) 363
Table 37: Overview of alternate functions (A-C) 373
Table 38: Driver power after PowerOn reset 393
Table 39: Paging register set within the XHIF module 444

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

1 INTRODUCTION

1.1 Overview of the ERTEC 200P

Reliability and determinism in the industrial communication network is very critical. Standard non-real-time Ethernet communications, e.g., TCP/IP, have delays in data processing due to layered OSI architecture. Packet forwarding can be delayed under high network load in network switches due to lacking prioritization in the frames.

In order to overcome those challenges, PROFINET RT and IRT have been introduced and established to be used as standard communication protocols for industrial network. The PROFINET communication protocols provide services for deterministic, reliable data exchange. Also, mechanisms for redundancy are available.

In PROFINET RT, real-time data is sent from Ethernet layer to application without any network layer process. PROFINET significantly decreases the Ethernet frame length since the frame does not have IP headers and related areas. Additionally, PROFINET uses VLAN Priority in real-time frames to prioritize the frames in network switches.

Some special industrial applications, e.g., motion control systems, require more deterministic behavior and higher speed in the communication. For such applications, PROFINET IRT is the perfect solution. In PROFINET IRT, communication bandwidth is reserved for process data. Via clock synchronization each node on the network can synchronize their clocks and use reserved periods to update the process data. To achieve such precision in communication, hardware support in the Ethernet switch is mandatory. In the ERTEC 200P, special ASIC support for IRT communication achieves update rates down to 125 µs and less than 1 µs jitter.

Furthermore, by means of additional PROFINET performance upgrade features, i.e., Dynamic Frame Packaging (DFP), Fast Forwarding and Fragmentation, communication speed and determinism can be increased.

ERTEC 200P was designed as a solution for PROFINET device vendors by providing the following capabilities:

- 2-port switch functionality that enables users to create a line topology by forwarding the Ethernet frames from its one port to the other
- PROFINET IRT communication capability through specific hardware support.

In the following sections, NRT, RT and IRT communication mechanisms in PROFINET and ERTEC 200P are summarized. For further information, please refer to the website of the PROFINET user organization: https://www.profinet.com/.

1.1.1 NRT (Non-Real-Time) communication

NRT communication is based on address-based forwarding of frames with virtual local area networking (VLAN) support which is specified in the IEEE 802.1Q standard.

ERTEC 200P provides VLAN supported switch functionality without external PHY support which is a mandatory hardware for sending and receiving of Ethernet network frames. ERTEC 200P has its own integrated PHYs that enable to send and receive Ethernet frames for 2 ports.

In addition to the pure switch technology, PROFINET also supports the coexistence of PROFINET and standard Ethernet traffic, i.e., in addition to a cyclic PROFINET application relation, web services or network diagnostics can also be transmitted via SNMP. The ERTEC 200P supports this communication; further options for implementing such functions can be found in the associated PROFINET Device Development Kit for ERTEC 200P.

Copyright © Siemens AG 2023. All rights reserved

1.1.2 RT (Real-Time) communication

Real-time communication uses a special EtherType (0x8892) for PROFINET. It enables the real-time process data to be streamlined when using standard Ethernet components, and it allows skipping the TCP/IP layers for processing the process data.

Support of PROFINET RT communication is mandatory for all devices that support PROFINET communication.

ERTEC 200P meets all PROFINET RT communication requirements by its specific hardware ASIC.

1.1.3 IRT (Isochronous Real-Time) communication

PROFINET RT already meets the majority of industry's communication requirements. Most of the PROFINET networks can work without any problem with cyclic data exchange over PROFINET RT. However, some high-performance applications require high speed data exchange with minimum jitter.

PROFINET IRT is usually used for less than 1 ms update times and less than 1 µs jitter. To be able to reach such precision in Ethernet communication, PROFINET IRT uses the PTCP protocol to synchronize clocks of the nodes on the network and to calculate the required bandwidth during the engineering of the network.

ERTEC 200P ASIC has integrated the so-called PN-IP. It enables clock synchronization via PTCP as well as communication over its two Ethernet ports using PROFINET IRT.

1.2 ERTEC 200P data

Note

For detailed information on e.g., package, operating conditions, interfaces, system reliability (FIT rates) and electromagnetic compatibility, see the ERTEC 200P-3 data sheet.

Processor system

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

General functions

- Internal clock generation (quartz oscillator, PLL)
- Integrated boot ROM (8 KByte)
- 6 × Timer (control signals as alternate function)
- Watchdog (control signals as alternate function)
- F-counter
- GDMA controller

Copyright © Siemens AG 2023. All rights reserved

- ARM926 interrupt controller
- PNPLL_OUT20:0: 3 × application timer blocks in PN-IP with free selection of 21 output functions (APPL1/2/3_CLK_INT, APPL1/2/3_CMP_INT5:0), of which:
  - up to 6 PLL signals at the ARM interrupt controller to control SW tasks
  - up to 9 PLL signals at the GPIO to control ext. applications
  - up to 10 PLL signals at the GDMA to control the internal I/O

1.2.1 System reliability

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

Logically adjacent bits are located in the manufacturer's single port RAMs in physically separate (different) words; 2-bit errors can therefore be corrected by the error correction in logically adjacent bits. RAMs with EDC can therefore be ignored in the calculation of the soft error rate. For a more precise breakdown of FIT rates see the ERTEC 200P-3 data sheet.

1.2.2 Electromagnetic compatibility (EMC)

The ERTEC 200P ASIC and the systems/devices implemented with it are designed for EMC in accordance with [EN] IEC 61000-6-2 (Electromagnetic compatibility (EMC) - Part 6-2: Generic standards - Immunity standard for industrial environments) and [EN] IEC 61000-6-4 (Electromagnetic compatibility (EMC) - Part 6-4: Generic standards - Emission standard for industrial environments).

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

See the ERTEC 200P-3 data sheet for further details.

1.3 ERTEC 200P use cases

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

Copyright © Siemens AG 2023. All rights reserved

Figure 1: ERTEC 200P use cases

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

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

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

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

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Figure 2: Application operation with external host

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

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

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

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

The following profile is supported:

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

Figure 3 shows the application of the standard IO.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Figure 3: Application operation without external host

1.4 Application notes

You should pay attention to the following application notes.

1.4.1 EMC SDRAM interface

The EMC of the ERTEC 200P-3 has two possible settings for clocked signal output (address, data and control signals) to an external SDRAM. These can be configured with EXTENDED_CONFIG.SODM (see chapter 2.3.8.8):

- *EXTENDED_CONFIG.SODM* = **'0'**
  SDRAM signal output with feedback clock at pin CLK_I_SDRAM (register value after reset)
- *EXTENDED_CONFIG.SODM* = **'1'**
  SDRAM signal output with internal system clock

Note

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

1.4.2 No free-running frequency at quartz break

With the PLL used and its current wiring, when there is no quartz clock, the PLL returns a free-running frequency of 0 Hz.

Background:

With the PLL used in ERTEC 200P, only the CLK output (CLKOA) generates a free-running frequency; the outputs CLKOB and CLKOC do not. As multiplication (factor of 20) of the quartz clock (25 MHz) to 500 MHz is only possible at the CLKOB output (a max. multiplication factor of 16 is possible at CLKOA) and CLKOB is therefore used to generate internal ASIC clocks, no free-running frequency is visible.

1.4.3 16-bit data access over XHIF in buffered mode

If the parallel host interface (XHIF) is used with a 16-bit data bus (8-bit data bus is not supported in ERTEC 200P), buffered mode must be configured in *XHIF_0/1_Px_CFG* for consistent 32-bit access over ML-AHB (see chapter 2.3.10.3).

Note

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

Background:

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

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

Copyright © Siemens AG 2023. All rights reserved

2 DESCRIPTION OF FUNCTIONS

2.1 Block diagram

Figure 4: ERTEC 200P Step 3 block diagram

2.2 General function description

PROFINET is an Ethernet-based communication system for industrial automation.

The main requirements governing the development of ERTEC 200P are:

- Usability
- Performance

2.3 Description of individual ERTEC 200P blocks

2.3.1 Overview

The following chapters go into detail on the ERTEC 200P modules:

- Processor subsystem (ARM926, TCM, debug support, booting)
  - ARM-ICU
- Multi-layer AHB (internal bus system)
- PN-IP
  - PerIF (Peripheral Interface)

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

- EMC (External Memory Controller)
- RSC (Reed-Solomon Controller) 1 and 2
- HostIF
  - XHIF (External Host Interface)
- GDMA (Global DMA)
- APB peripherals
  - SPI 1 – 2
  - I^2C
  - UART 1 - 4
  - Timer 0-5
  - Watchdog
  - F-timer
  - GPIOs 0…95
  - I-Filter
  - SCRB (System Control Register Block)
  - Octal-SPI (OSPI)
- CRU (Clock and Reset Unit)

2.3.2 Register descriptions

All bits marked "reserved" in the register descriptions must not be evaluated in read access and must not be written.

If no specific information is given, the registers can be changed during operation.

Definition of "**Attributes**" used in the detailed register description:

r	(read)	The register value read cannot be modified by the HW.
rh	(read modified)	The register value read can be modified by the HW.
rht	(read modified trigger)	The register value read can be modified by the HW and triggers and HW function.
w	(write)	Write access applies the data value to the register.
wt	(write trigger)	Write access triggers a HW function that modifies the register content.
t	(trigger)	Access triggers a HW function.
wpt	(write protect)	Write access is protected and applies only with another HW event.
wk	(write key)	Write access applies the data value to the register only with a key inside the write data.
(..)	-	Only a part of register slices is used with this attribute.

Copyright © Siemens AG 2023. All rights reserved

2.3.3 Processor subsystem (ARM926)

The subsystem used is an ARM926 subsystem based on an ARM9E-S processor. The ARM926 subsystem is primarily for the application and non-runtime-critical routines of the PN stack. Figure 5 shows the structure of the ARM926 subsystem. It consists of a core system, the JTAG interface and a TCM_Block_926. Alongside the ARM926EJ-S processor, the core system contains a data and an instruction cache, a Memory Management Unit (MMU), separate connections to each AHB layer for instruction and data, an ETM9 (medium+) Trace Macrocell and an integrated trace buffer (ETB11). The Tightly Coupled Memories for data and instructions are located in TCM_Block_926. AHB access to D-TCM is with a 'DMA-DTCM Access Controller'. I-TCM is only operated by ARM926EJ-S and cannot be accessed by the AHB. ARM926EJ-S is described in detail in /7/.

Of the AHB masters, the host interface, PN-IP and GDMA can access D-TCM in the ARM926 subsystem. ARM926EJ-S has access to the AHB slaves, as detailed in chapter 2.3.4.3.2.

2.3.3.1 Features

- Macro Type:
  - Soft macro
- Function summary:
  - ARM926EJ-S, ARM926EJ-S16
  - 16 KByte data and 16 KByte instruction cache
  - 256 KByte Instruction/Data Tightly Coupled Memory incl. byte EDC, can be set in 64-KByte increments
    (I-TCM: 0 – 256 KByte, D-TCM: 256 – 0 KByte)
  - Debug compatibility through embedded ICE with JTAG interface, ETM cell with ETB (Embedded Trace Buffer)
  - Memory Management Unit (MMU)
  - Little-endian support only
- Operating frequency:
  - ARM926 only 250 MHz supported
  - 125 MHz AMBA bus system

Copyright © Siemens AG 2023. All rights reserved

2.3.3.2 Block diagram

Figure 5: ARM926 subsystem

2.3.3.3 ARM926EJ-S processor

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

The processor and TCMs (see chapter 2.3.3.3.2) can be operated at 125 or 250 MHz. Setting is with the CONFIG(1) pin (see 2.3.14.11.3). The AHB interfaces always operate at 125 MHz.

Note

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

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

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

2.3.3.3.1 Cache structure of ARM926EJ-S

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

- 16 KByte instruction cache
- 16 KByte data cache

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

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

The content of the cache segments can be "locked". This lock function makes it possible to retain permanently the instruction set for fast routines in the I-cache. In ARM926EJ-S, this mechanism can only be implemented on a segment-specific basis. More information on caching can be found in /7/ (see chapter 3.2).

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

The I-cache and D-cache have one byte of parity. The I-tag and D-tag RAM is also parity-protected. If an error occurs when reading the I-cache or the relevant I-tag entry, the cause of the error (I_Cache_Parity, I_Tag_Parity) is saved in the SCRB register 'EDC_EVENT' (see chapter 2.3.14.11.27.1) and the interrupt EDC_Event (IRQ48) is triggered (see 2.3.5.9). An error while reading the D-cache or the relevant D-tag entry generates the entry (D_Cache_Parity, D_Tag_Parity) in the 'EDC_EVENT' register with the EDC_Event interrupt. '0h' must be written to the 'EDC_EVENT' register to clear the error.

The parity bits are undefined after reset. If parity monitoring of the caches is to be used, the caches must be initialized by the SW (each cache line must be initialized in a loop). Until this point, the parity logic is disabled. Control is with the SCRB register 'EDC_PARITY_EN' (see 2.3.14.11.27.1). Bit 'I_CACHE_PAR_EN' enables the parity bits of the I-cache and I-tag, and bit 'D_CACHE_PAR_EN' the parity bits of the D-cache and D-tag. After reset, both bits are set to '0' and the parity bits are disabled. The SW can set these bits to '1' after cache initialization.

2.3.3.3.2 ARM926 Tightly Coupled Memories (ARM926_TCM)

The I-TCM and D-TCM are located in TCM_Block_926. The TCM configuration includes 256 KByte[1] for instruction and data TCM (I/D-TCM). The memory operates at the clock rate of the processor, 125/250 MHz. The memory consists of segments of 64 KByte, which can be assigned either to the instruction or to the data TCM. This results in a configurable memory size of 0 – 256 KByte for the I-TCM with 256 – 0 KByte for the D-TCM. After reset, the TCM926 configuration is 256 KByte D-TCM. The final setting is made during booting by the boot loader (SW), in the SCRB register 'TCM926_MAP'. After this reconfiguration, the boot loader must still display the TCM memories in the address range of the coprocessor interface (CP15 c9).

Note

I-TCM and D-TCM can only be displayed in the address range in increments of 2^n. If, for example, a physical size of I-TCM = 192 KByte / D-TCM = 64 KByte has been selected in 'TCM926_MAP' register, ARM926 can only assign I-TCM an address range of 256 KByte and D-TCM the required address range of 64 KByte. The values 256 KByte for I-TCM and 64 KByte for D-TCM are therefore to be read out in the coprocessor interface of ARM926 (CP15 c9).

The SW must then not access the unassigned area/hole (I-TCM: 192 – 256 KByte). If ARM926 accesses the unassigned area in the event of an error, either 'Invalid I-TCM926 Access Interrupt' (access to the I-TCM hole) or 'Invalid D-TCM926 Access Interrupt' (access to the D-TCM hole) is triggered (see chapter 2.3.5.9).

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

[1] 1 KByte is equal to 1024 bytes.

Copyright © Siemens AG 2023. All rights reserved

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

Only ARM926 D-TCM can be accessed from the AHB. In the event of access to the unassigned D-TCM address range from an AHB master, access is prevented and a QVZ interrupt is triggered (see chapter 2.3.5.9).

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

The I-TCM and D-TCM have an EDC code, which is assigned byte by byte (5 bits per byte, 1-bit error, correctable; 2-bit error, detectable). If an error occurs during reading, the cause of the error (I/D-TCM926-1B: 1-bit error corrected or I/D-TCM926-2B: 2-bit error detected) is saved in the SCRB register 'EDC_EVENT' (see chapter 2.3.14.11.27.1) and the interrupt EDC_Event (IRQ48) is triggered (see chapter 2.3.5.9). '0h' must be written to the 'EDC_EVENT' register to clear the error.

After ARM926-TCM reset, the I/D-TCM is automatically initialized by the HW. The initialization process (incl. EDC bits) for the complete I/D-TCM takes ca. 16.5 µs. The end of the initialization process is signaled in the SCRB register 'EDC_INIT_DONE' (see 2.3.14.11.27.1, I_TCM926_INIT_DONE, D_TCM926_INIT_DONE). The EDC logic can also be disabled using the SCRB register 'EDC_PARITY_EN' (see 2.3.14.11.27.1). The 'EDC_DISABLE_ARM926' bit deactivates the EDC logic. The EDC logic is enabled after reset.

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

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

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

EAS 926 , 250 MHz	access types								
Memory area	**AHB write access**	**AHB read access**	**AHB read getting an ECC detect**	**AHB read getting an ECC error**	**TCM write access**	**TCM read access**		**TCM read ECC detect**	**TCM read ECC error**
ITCM area (from AHB not possible)	na1	na1	na1	na1	ok	ok		I-TCM926-1B	I-TCM926-1B + I-TCM926-2B
ITCM area hole (from AHB not possible)	na1	na1	na1	na1	Invalid I-TCM926 Access	Invalid I-TCM926 Access		na2	na2
ITCM area mirror (not possible)	na1	na1	na1	na1	na1	na1		na1	na1
DTCM area	ok	ok	D-TCM926-1B	D-TCM926-1B + D-TCM926-2B	ok	ok		D-TCM926-1B	D-TCM926-1B + D-TCM926-2B
DTCM area hole (from AHB not possible)	na1	na1	na1	na1	Invalid D-TCM926 Access	Invalid D-TCM926 Access		na2	na2
DTCM area mirror (not possible)	na1	na1	na1	na1	na1	na1		na1	na1

Description:	
ok	no error
na1	not availiable / access is not possible
na2	not availiable / access never issues an ECC detect/error
AHB error	AHB error response via HRESP
D-TCM926-1B	AHB ECC detect signal
D-TCM926-2B	AHB ECC error signal 8ns width for the DTCM
I-TCM926-1B	AHB ECC detect signal 8ns width for the ITCM
I-TCM926-2B	AHB ECC error signal 8ns width for the ITCM

2.3.3.3.3 Memory Management Unit (MMU)

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

- Page size: 1 MByte, 64 KByte, 4 KByte and 1 KByte
- Separate TLBs for instruction and data
- Access attributes can be changed without a TLB flush

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

- A fast context switch enables virtual address remapping in a 0 – 32 MByte area
- TLB entries can be locked.

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

2.3.3.3.4 Bus interface of the ARM926 Processor

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

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

To support debugging, an ETM9 revision r2p2 (medium+[2]) module with integrated trace buffer (ETB11) is connected to the ARM926EJ-S system in the ERTEC 200P. This module allows an instruction and data trace. The trace buffer is 8 KB and supports both ARM926 clocks (125/250 MHz).

The ETM module receives all signals required for the data and instruction trace from the processor. The ETM9 module is operated over the JTAG interface.

Note

ARM926 trace port is not supported.

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

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

Recording in the trace buffer can be controlled by the external debugger over the JTAG interface or by the SW on the ARM926EJ-S[3]. A register set that can also be accessed from the AHB is available in ETB11 for this purpose. ARM926EJ-S has two interrupts for processing the trace (ETB_ACQCOMP -> Required trace is saved in the ETB, ETB_FULL -> Trace overrun in the ETB; (see chapter 2.3.5.9). The trace can, however, only be read from the trace buffer once tracing is complete (ETB_ACQCOMP interrupt). A detailed description of ETB11 and the corresponding programming interface can be found in /46/.

The trace cell used in ERTEC 200P is "Medium+". This version has the following features:

- 8 address comparators
- 4 address range comparators
- 2 data comparators with a filter function
- 4 trigger inputs: one 'ETMEXTIN0' input (available as alternate function over GPIO)
- 1 trigger output: 'ETMEXTOUT0' (available as alternate function over GPIO)
- 8 MMD areas for decoding the physical address range of the ERTEC 200P
- 1 3-stage sequencer

---

[2] The medium+ ETM9 cell has a larger FIFO (18 -> 45 bytes).
[3] ETM trace condition configuration is only possible from the external debugger and not over the AHB interface of the ETB11.

Copyright © Siemens AG 2023. All rights reserved

- 2 counters
- FIFO size = 45 Bytes

More information is available in specifications /4/ and /5/ for the ETM9 cell (see chapter 3.2).

In addition to these specifications, the following 8 MMD areas have been decoded in ERTEC 200P with the hardware.

- Seg 0, 0K – 256K: Instruction access to ARM926 I-TCM
  128 – 128M...256M: Data access to ARM926 D-TCM
- Seg 2, 0M - 256M: Instruction and data access to external SDRAM
- Seg 3, 0M – 64M: Instruction and data access to external CS0 (normal flash)
- Seg 3, 64M – 128M: Instruction and data access to external CS1 (normal SRAM)
- Seg 3, 128M – 256M: Instruction and data access to external CS2/3
- Seg 1, 6M – 8M: Data access to PN-IP
- Seg 1, 8M - 10M: Data access to PerIF
- Seg 1, 0M – 6M: Data access to ARM-ICU
  10M – 16M: Data access to GDMA, EMC register and ETB11
- Seg 4, 0K – 116K: Data access to APB

2.3.3.4 Debug support

See chapter 2.3.11 in /7/.

2.3.3.4.1 Debug configuration

JTAG Debugging is enabled if pin TAP_SEL is set to ‘0’. When set to ‘1’ JTAG is used for boundary scan functionality.

Two TAP controllers are connected in the JTAG chain (see chapter 2.3.3.4.2):

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

This produces the following queue:

- TDI → ETB → ARM926EJ-S → TDO

The number of instruction register bits is:

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

2.3.3.4.2 ARM926EJ-S Debug interface

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

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

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.3.5 Boot process for the ERTEC 200P

A Primary Boot Loader (PBL) is integrated into ERTEC 200P (64 KByte boot ROM at the APB). The ERTEC 200P can be booted from several boot media. Different boot modes are supported for the various configurations. The boot device will be enabled by the ARM926, if necessary.

The different boot modes boot from one of the following sources.

Table 2: Boot, ERTEC 200P boot modes

Boot mode	Booting	Use case
1 … 2	…from the external NOR flash connected to the EMC	Use case 2 (Standalone application)
7	… from an external host CPU that copies the firmware to the TCM via XHIF	Use case 1 (External host)
9	…from firmware copied to the TCM over the Octal-SPI master interface of the ERTEC 200P	Use case 2 (Standalone application)

Booting can be triggered by:

- HW reset (PowerOn reset, XRESET)
- Watchdog reset (watchdog timer expires)
- Software reset

The complete list of reset triggers can be found in chapter 2.3.14.11.4 and in the CRU (2.3.13.2).

From the boot ROM, ARM926 processes the primary boot loader, which loads a secondary boot loader from the corresponding boot medium to the D-TCM (Block3: 0x0803_0000) so that, after the switch to I-TCM (Block0: 0x0000_0000), the secondary boot loader is saved according to the interrupt jump table (see chapter 2.3.3.5.1). If an external host is responsible for booting, it transfers the secondary boot loader to the D-TCM (Block3: 0x0803_0000).

Once the secondary boot loader has been loaded, this 64 KByte segment is switched to the I-TCM and is shown at address 0x0000_0000. The secondary boot loader loaded is not lost when the system switches to the I-TCM. Following a PowerOn reset, the TCM926 configuration is 256 KByte D-TCM. As the assignment of the SRAM blocks in the I-TCM and D-TCM is completely reversed, the boot loader must copy the code image in 64 KByte blocks from the bottom D-TCM Block3 (0x0803_0000) → D-TCM Block 0 (0x0800_0000), i.e., in ascending order. The address order is not reversed in the TCM block. After download to the D-TCM, the final TCM926 configuration is set in 'TCM926_MAP' (see chapter 2.3.3.3.2). The boot loader must then display the TCM memories in the address range of the coprocessor interface (CP15 c9).

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

- The secondary boot loader sets the final TCM926 configuration required in 'TCM926_MAP'. The code image can then be transferred by the boot medium to the I-TCM after the secondary boot loader.
- First, the 64 KByte segment that is already assigned to the I-TCM is filled. The code image is then loaded to the D-TCM, in the correct order. After this download, the final TCM926 configuration is set in 'TCM926_MAP'. The secondary boot loader must also show the final configuration of the TCM in the coprocessor register (CP15 c9) address range.
  GDMA can also be used to load the code image to the D-TCM. This code image is loaded to the D-TCM following booting by the host.

For booting from an external host, the primary boot loader creates a page for the D-TCM (boot RAM) in the 8-page XHIF registers. The remaining pages can then be set up specifically by the

Copyright © Siemens AG 2023. All rights reserved

external host. It makes sense to set up additional pages for PN-IP, PerIF, EMC-SDRAM/EMC-SRAM, GDMA and APB peripherals.

Note

TCM block assignment in the ARM926 subsystem is reversed when you switch from D-TCM to I-TCM.

Figure 6: TCM block assignment in the ARM926 subsystem

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

To provide the routines (e.g., Reed-Solomon encoding) of the primary boot loader to the secondary boot loader, the primary boot code is located in the original address range of the boot ROM (address 0x4002_8000). The primary boot loader immediately branches from the mirrored area (address 0x0002_0000) to its original address range (address 0x4002_8000), where it processes the first-level boot code.

All boot modes with the exception of the boot mode from the external NOR flash in compile mode (see chapter 2.3.3.5.2) process the secondary boot loader stored in the TCM. As no instructions

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

can be processed from the D-TCM, the primary boot loader must map the data copied to D-TCM block 3 to I-TCM block 0.

2.3.3.5.1 Primary Boot Loader

The primary boot loader is a separate memory block (boot ROM, 64 KByte) which starts after a PowerOn reset from address 0x0. Depending on the boot mode settings (with boot pins or in the SCRB register 'BOOT_REG', see chapter 2.3.14.11.2, and refer to the ERTEC 200P-3 data sheet), the system recognizes the medium from which the software is to be loaded. For the download medium only, the required system settings are made and loaded to the TCM in line with the boot mode of the program code (secondary boot loader). The software can be processed after loading.

The following settings are made by the primary boot loader if SCRB.CONFIG.EMC_HOSTIF_MUX == 0:

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

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

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

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

Table 3: Interrupt vector table

Address	Data item	Comment	Priority
0x00	Reset vector	Jump command following power on/WD/SW reset → *Stack pointer from address: 0x0800_FD00*	1 (highest)
0x04	Undefined Instruction	Jump command after execution of an illegal opcode → *Stack pointer from address: 0x0800_FC00*	6 (lowest)
0x08	Software interrupt	Jump command after SWI triggered → *Stack pointer from address: 0x0800_FA00*	6 (lowest)
0x0c	Opcode from illegal address	Jump command after illegal opcode address is read → *Stack pointer from address: 0x0800_FB00*	5
0x10	Data from illegal address	Jump command after illegal data address is read → *Stack pointer from address: 0x0800_FB00*	2
0x14	*reserved*		*n.a.*
0x18	IRQ	Jump command after standard interrupt (IRQ) → *Stack pointer from address: 0x0800_F900*	4
0x1C	FIQ	Jump command after fast interrupt (FIQ) → *Stack pointer from address: 0x0800_F800*	3
0x20	Opcode	First possible opcode of the secondary boot loader	
...			

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

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

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

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

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

Note

In case of an error correction of the SBL by RSC, this information is stored in SERVICE_3_REG.

Note

The last 2 KB of the D-TCM in block 0 are reserved for internal variables and stack pointers and must not be used for a secondary boot loader. No data/code may be loaded here either before the secondary boot loader has set the system to its requirements and "enabled" this area (see Figure 6).

2.3.3.5.1.1 Boot mode settings

The EMC pins A[16:15], XOE_DRIVER, DTXR and XAV_BF are used to set the boot mode. The pins are latched to the boot register 'BOOT_REG' (refer to the ERTEC 200P-3 data sheet) during an active reset and return to their normal function once the reset is cleared. The ERTEC 200P and the XHIF interface are also configured with the EMC pins A[23:17].

2.3.3.5.1.2 Handling after ARM926EJ-S SW reset

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

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

2.3.3.5.1.3 Memory swapping

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

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

Swapping is done by programming the 'MEM_SWAP' register in the system control status register block SCRB (see chapter 2.3.14.11.7). With reserved coding, there is no swapping in the 'MEM_SWAP' register (or to be precise, no memory space is shown in segment 0). If these addresses are then accessed at the AHB, a QVZ (see 2.3.4.5.1) is triggered.

2.3.3.5.1.4 Startup times

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

Copyright © Siemens AG 2023. All rights reserved

Table 4: Startup times

Boot mode	Booting		t_1	t_2 (startup time)
1	External NOR flash (16-bit), *ASYNC_ADDR_MODE* = 1	Compile mode	485 µs	n.a. [2)]
2	External NOR flash (32-bit), *ASYNC_ADDR_MODE* = 1	Compile mode	485 µs	n.a. [2)]
7	XHIF (external host)		325 µs	41 µs per 1024 bytes (depends on external master speed)
9	Octal-SPI via external serial memory (slave)	8 bytes RSC [1)]	554 µs	454 µs per RSC page (236 bytes net), 515 µs for first RSC page

[1)] With error correction and recognition (Reed-Solomon) of max. 8 bytes under usage of the Reed-Solomon Controller (RSC, see chapter 2.3.9).
Copy mode: The SBL is RSC-protected and must be copied to a local memory of the ASIC. The SBL is executed by the processor from that internal memory.

[2)] In Compile mode the SBL execution starts immediately after t_1.
Compile mode: The SBL is executed by the processor directly from the source memory location (e.g., flash).

t_1: Time between XRESET end and loading of first SBL data.

t_2: Time from loading of first SBL data until the first command of the secondary boot loader is executed (startup time). This includes the time for copying data. If the alternative block is used or several data bytes are copied, the time may be longer.

The startup times in ERTEC 200P are calculated based on **t1+t2**.

2.3.3.5.2 Boot mode 1 … 2 (booting with NOR flash) in compile mode

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

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

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

- If the user program is to be processed straight from the NOR flash, the ARM926 interrupt vector table must be stored from NOR flash memory address 0x0000_0000 (see 2.3.3.5.1). The secondary boot loader must implement remapping (MEM_SWAP.SWAP = "010", see 2.3.14.11.7) to show the vector list in the NOR flash at address 0x0000_0000.
- If the user program is to be processed from the TCM (I-TCM) or SDRAM, the secondary boot loader must copy the data from its NOR flash firmware address to the required destination memory (D-TCM, SDRAM), configure the relevant D-TCM as I-TCM ('TCM926_MAP', see chapter 2.3.14.11.14) or remap the SDRAM to address 0x0000_0000 (MEM_SWAP.SWAP = "001", see chapter 2.3.14.11.7).

The complete interrupt vector table is available in the boot ROM to prevent system deadlock. When an interrupt or an exception occurs, a restart is always carried out if the vector address of the boot ROM is mapped at address 0x0000_0000 (MEM_SWAP.SWAP = "x00", see chapter 2.3.14.11.7) and no I-TCM has been configured (if the I-TCM is configured, this is always shown at address 0x0000_0000 irrespective of MEM_SWAP.SWAP).

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Note

The reset pin of the NOR flash must be connected to ERTEC 200P reset.

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

2.3.3.5.3 Boot mode 7 (Booting with External Host)

After reset, the ARM926 starts the instruction fetches from the primary boot loader. The basic XHIF configurations (16/32-bit bus, XHIF_ACC_MODE, XHIF_POL_RDY) are latched to the 'CONFIG_REG' register after reset (see chapter 2.3.14.11.3) and wired directly to the XHIF module. The parameter settings (for example page sizes, ...) at XHIF are made by the external host in the XHIF control registers.

The ERTEC 200P signals readiness for firmware upload to the host with a software handshake (boot loader functionality, semaphore) (ready for firmware upload). At address D-TCM+0x40 (address 0x0803_0040 in block 3 of the D-TCM), the OK identifier = 0x6879_9786 (software handshake) and the first OP code of the secondary boot loader is processed from address D-TCM+0x44 (address 0x0803_0044 in block 3 of the D-TCM).

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

SW code download from the XHIF must be actively executed by the external host processor. The boot code can be transferred from the XHIF host to the D-TCM (block 3) of the ERTEC 200P. At the end of data transfer, the XHIF host sets identifier 0x6879_9786 at address D-TCM+0x40 (semaphore address). Once the semaphore has been set, the ERTEC 200P maps the D-TCM to I-TCM ('TCM926_MAP', see chapter 2.3.14.11.14) and the secondary boot loader is started at address 0x0000_0044 (I-TCM). Access to the I-TCM over XHIF is not possible; only the D-TCM can be addressed.

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

For an external host processor to recognize whether the XHIF interface is ready, the following procedure must be observed:
The external host polls the XHIF range register of bank 0 until the value 0x0010_0000 (1 MByte range) appears. The reset value of the XHIF range register is 0x0000_0000. Access to this register is possible even during an active reset. If the external host reads the correct XHIF range register value (0x0010_0000), it then reads the offset for PAGE0, which must then contain the value 0x0800_0000. The host then reads the semaphore at D-TCM+0x40 (address 0x0803_0040 in block 3 of the D-TCM) until it returns the value 0x0000_0000.

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

Note

The interrupt vector table (see chapter 2.3.3.5.1) is to be stored from D-TCM+0x0 to D-TCM+0x39.

Copyright © Siemens AG 2023. All rights reserved

Figure 7: Flow chart for the external host for establishing XHIF interface enable

2.3.3.5.4 Boot mode 9 (Booting with Octal-SPI – with RS coding)

When booting from OSPI, the PBL enables X_MODE in EXTENSION_REG to gain access to the OSPI module.

Note

Please **note** that the ERTEC 200P-3 **only supports Quad-SPI mode**.

The Octal-SPI interface of ERTEC 200P-3 is configured with the following parameters:

- Octal-SPI of ERTEC 200P is the master (thus specifying the Octal-SPI clock).
- 8 data bits
- Transmit/Receive with MSB first
- Idle Clock Line = 0
- Data is latched with a rising edge and output with a falling edge.
- Baud rate is programmed fixed to 12.5 Mbps (conservative baud rate setting).
- The existing Octal-SPI module supports serial interfaces that are compatible with the Motorola, TI and NSC serial protocols.
- Starting is always effected in single-mode, meaning SPI-compatible Standard-SPI mode.
- SPI memory chip select is addressed via GPIO_32.
- The used Octal/Quad/Dual-SPI memory block has to support the following features:
  - Reset command sequence 0x66, then 0x99
  - Ramp-up after Power up in SPI-compatible Standard-SPI mode.

The Octal-SPI boot only supports Octal/Quad/Dual/Standard-SPI flash storage media that require the command 0x03 for reading.

After a SW or watchdog reset the operating mode (Standard/Dual/Quad/Octal) of the external storage block is not known. In order to set the external storage block in a defined state (Standard-SPI mode) so that data from the external storage block can be read, there are two reset sequences performed.

The first reset sequence 0x66 → 0x99 for SDR and for DDR 0x6699 and 0x9966 is first transferred in Octal-SPI mode (8 data lines). If the external storage block (SPI device) is in:

- Octal-SPI mode, it changes to the Standard-SPI mode after this reset sequence
- Quad-/Dual- or Standard-SPI mode, a command that is too short is recognized so that there is no response to this sequence by the storage block.

Now the reset sequence 0x66 → 0x99 is repeated for DDR and for SDR in Quad mode (4 data lines). If the external storage block (SPI device) is in:

- Quad-SPI mode, it changes to the Standard-SPI mode after this reset sequence
- Dual- or Standard-SPI mode, a command that is too short is recognized so that there is no response to this sequence by the storage block.

Now the reset sequence 0x66 → 0x99 is repeated in Dual mode (2 data lines). If the external storage block (SPI device) is in:

- Dual-SPI mode, it changes to the Standard-SPI mode after this reset sequence
- Standard-SPI mode, a command that is too short is recognized so that there is no response to this sequence by the storage block.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Subsequently the reset sequence 0x66 → 0x99 is repeated in the Standard-SPI mode (1 data line).

After this sequence it is ensured that the block is in Standard-SPI mode.

The timing used in ERTEC 200P is:

Parameter	Min (Spec.)	Max (Spec.)	ERTEC used	Unit
tCSL	500	--	1000	ns
tCSH	500	--	1000	ns

After the fourth CSn pulse, the slave triggers its internal reset. As a quick startup is required, the waiting time is only 5 ms, because this is the time that is needed when a program command is aborted. On Power up there is no pending command, except if a watchdog reset occurs. If the device is not available for more than 5 ms, and data cannot be read, an internal restart will be performed, because an SBL cannot be loaded.

Booting is always carried out with Standard-SPI mode. A change to Dual/Quad/Octal mode can be done with SBL by writing a volatile configuration register in the respective Dual/Quad/Octal SPI device. Controlling of the ChipSelect signal is effected via the GPIO functionality and is not controlled by the Octal-SPI IP, so that the ChipSelect signal to the external SPI device does not become inactive due to a readout of the data being too slow.

After configuration of the Octal-SPI master the read command 0x03 with the start address 0x0000 is sent to the SPI device. If =0x00 or =0xFF (no start byte) is found within 10 bytes (min. 3 address bytes + 7 other bytes), the read command 0x03 is sent to the SPI device with the start address 0x010000… .

If neither =0x00 nor =0xFF (no start byte) is found with this command, a restart is carried out by means of a watchdog reset.

If a start byte is read for the read command in original or alternative start address (!= 0x00 and != 0xFF), the position of the start byte in the SPI data stream indicates how many address bytes (3, 4, 5, …x bytes) are required for addressing the SPI device (distance between read command 1 and identifier = number of address bytes).

Next, the data bytes read via Octal-SPI are checked and corrected by means of Reed-Solomon coding. For this purpose, a fixed number of data bytes is read from the Octal-SPI flash, checked for errors, corrected, if necessary, and only then written to the D-TCM. Checking and correcting are carried out with the integrated RS controller (for a description of the functionality see chapter 2.3.9). In this boot mode, *NmbRSC* pages data blocks with 254/252 bytes are read from the SPI flash and, if necessary, corrected by the RS controller. There are 236 bytes of user data per data block. A total of 59 × 4 bytes with 32-bit access are therefore copied to the D-TCM. The check bytes are not copied.

Once the start byte is successfully checked as identifier, the information about the number of RSC pages (*NmbRSC*) is read out. The subsequent bytes are interpreted as data and are stored in ascending order to the D-TCM, from the address 0x00...00 until the number of RSC pages (*NmbRSC*) to be copied is reached.

Subsequently the primary boot loader then configures the D-TCM (192 KB) and I-TCM (64 KB) ('TCM926_MAP', see chapter 2.3.14.11.14), and the I-TCM is shown at address 0x0000_0000 (see Figure 6). Processing then continues with a jump to address 0x0000_0044. The length of data to be copied is limited to the maximum size of the D-TCM.

The maximum data quantity to be copied into the D-TCM is limited based on the maximum number of RSC pages (*NmbRSC* = 0xFF): 255 × 236 bytes = 60180 bytes.

If the start byte is not successfully checked as identifier, a watchdog is activated. When the watchdog expires (HW reset activation), device detection is run again.

Copyright © Siemens AG 2023. All rights reserved

Table 5: Example

Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	...
Send	0x03	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	…
	RD													
Re-ceive	*0xFF*	*0xFF*	*0xFF*	*0xFF*	*0x5A*	*0x10*	*0x00*	*0x00*	*0x01*	*0xFF*	*0xF0*	*0x00*	*0x00*	…
					Start Byte	*Nmb RSC*	*Data 0*	*Data 1*	*Data 2*	*Data 3*	*Data 4*	*Data 5*	*Data 6*	…

In the example above, 16 RSC pages with 236 user data bytes = 3776 bytes each, are read from the SPI flash device with 0 – 16 Mbyte (3 address bytes).

Due to the ID byte and the number of RSC pages (*NmbRSC*), the first page is read out with 238 user bytes + 16 bytes Reed-Solomon code. All additional pages are checked with the user data size of 236 bytes + 16 bytes Reed-Solomon code.

The data size restriction of 236 bytes is necessary so that the D-TCM can be written permanently with Word access (32-bit) without an additional data shift having to be carried out due to an unaligned 4-byte quantity (236 user bytes → 59 × 4 bytes write access).

Table 6: Data size restriction

Page No.	Total number of bytes (row)	User bytes	Quantity of the 32-bit write access to D-TCM
1	254	238	59
2	252	236	59
…	252	236	59
255	252	236	59

Table 7: Memory image in the I-TCM

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

Note

The interrupt vector table (see chapter 2.3.3.5.1) is to be stored from D-TCM+0x0 to D-TCM+0x39.

2.3.3.5.4.1 Startup time

The startup time of OSPI boot depends on the OSPI-Mode (ERTEC 200P Step 3: only Quad-SPI), the usage of RSC, and the DMA. See table below for details (times are simulation results).

Note

Time required to load SBL and setup OSPI:

- Reset until SBL (3 KB) is loaded: 7 ms
- Initphase OSPI: 0.035 ms

Table 8: OSPI startup time

ERTEC 200P (125 MHz)	1 MByte		PBL + 5 MByte	
	without RSC	**with RSC**	**without RSC**	**with RSC**
GDMA without EDC (64 Byte Prefetch)	21.8 ms	23.6 ms	116.0 ms	125.0 ms
GDMA with EDC (64 Byte Prefetch)	22.3 ms	23.8 ms	118.5 ms	126.0 ms
GDMA without EDC (256 Byte Prefetch)	17.8 ms	19.6 ms	96.0 ms	105.0 ms
GDMA with EDC (256 Byte Prefetch)	18.0 ms	19.6 ms	97.0 ms	105.0 ms
	Quad DDR		Quad DDR	

2.3.3.6 Secondary Boot Loader

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

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

2.3.3.6.1 Memory swapping by the Secondary Boot Loader

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

The reset vector of the ARM926 processor points to address 0x0000_0000. That is why the boot ROM starts at address 0x0000_0000 after reset (power on, HW, SW or watchdog reset). The boot ROM can also be addressed in its mirrored area (0x4002_8000).

It is left to the secondary boot loader to change the TCM mapping. To allow the secondary boot loader to be run, the primary boot loader switches the first page (64 KByte) of the D-TCM to I-TCM ('TCM926_MAP', see chapter 2.3.14.11.14). If additional I-TCM is required, the secondary boot loader must switch again.

Swapping is done by programming the MEM_SWAP.SWAP register (see chapter 2.3.14.11.7). This procedure ensures that the exception vectors are always assigned and prevents a deadlock.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.4 AMBA (Internal Bus System)

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

All masters (ARM926-I, ARM926-D, PN-IP-M1, PN-IP-M2, Host interface, GDMA, RSC 1 and RSC 2) and the slaves (D-TCM926, ETB, interrupt controller (ARM-ICU), GDMA register/memory, external memory controller (EMC), two APB bridges, peripheral interface (2 connections), PN-IP) are connected at the multi-layer AHB, which has a high data transfer rate and high bus availability.

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

2.3.4.1 Characteristics of the bus system in ERTEC 200P

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

Table 9: Internal bus system, parameters of internal buses

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

2.3.4.2 Endianness

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

2.3.4.3 AHB subsystem (Multi-Layer AHB)

The multi-layer AHB bus is a 32-bit bus with multi-master capability. It runs at 125 MHz and has the functionality of the ARM-AHB-Lite bus.

Connecting multiple AHB segments in a multi-layer AHB allows AHB masters to access different slaves simultaneously.

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

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

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

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

Copyright © Siemens AG 2023. All rights reserved

Definition of AHB-Lite bus

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

The AHB slave modules allow the following types of access.

Table 10: Access types to AHB slaves

AHB module (slave)	Access permitted	Comment
PN-IP [1)]	8/16/32-bit	Data bus: 32-bit
Peripheral interface	8/16/32-bit	Data bus: 32-bit
RSC 1	8/16/32-bit	Data bus: 32-bit
RSC 2	8/16/32-bit	Data bus: 32-bit
EMC memory	8/16/32-bit	Data bus: 32-bit
EMC register	8/16/32-bit	Data bus: 32-bit
GDMA (RAM + register)	Only 32-bit	Data bus: 32-bit
ARM-ICU	Only 32-bit	Data bus: 32-bit
ARM926 - TCM	8/16/32-bit	Data bus: 32-bit
ARM926 - ETB	8/16/32-bit	Data bus: 32-bit
AHB/APB bridge	Depends on APB peripheral	Data bus: 32-bit
AHB/APB bridge (OSPI)	Only 32-bit	Data bus: 32-bit

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

2.3.4.3.1 AHB Arbiters

Each of the AHB arbiters uses the same arbitration procedure. Round robin or a fixed priority assignment for the AHB masters (see Table 11) could be used as an alternative arbitration algorithm by programming the ARB_MODE bit in the SCRB register 'M_LOCK_CTRL'. This option shall not, however, be implemented in the light of the dynamic processes at the multi-layer AHB.

Using round robin as the arbitration procedure prevents the AHB masters from blocking each other at the multi-layer AHB for a prolonged period of time (see chapter 2.3.4.3.2).

Table 11: Fixed priority assignment (default)

Priorities	Master	Notes
10	RSC 2	Highest priority
9	----	
8	----	
7	Host interface	
6	GDMA	
5	PN-IP-M1	[1)]
4	RSC 1	
3	----	
2	ARM926-D	
1	ARM926-I	Lowest priority

[1)] Due to backward compatibility, the default setting is fixed priority. We recommend changing the setting to round robin.

2.3.4.3.2 AHB Multi-layer configuration

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

Copyright © Siemens AG 2023. All rights reserved

Table 12: AHB master-slave coupling

	AHB slaves											
	S1	S2	S3	S5	S6	S7	S9	S10	S12	S13	S14	S15
	D-TCM926	ETB	ARM-ICU	GDMA	EMC	AHB/APB bridge 1	PerIF host	PerIF-PN	PN-IP	RSC 1	RSC 2	AHB/APB bridge 2
AHB masters												
ARM926-D [1)]		x	x	x	x	x	x		x	x	x	x
ARM926-I [1)]					x	x						x
GDMA	x	x		x	x	x	x			x	x	x
Host interface	x			x	x	x	x		x	x	x	x
RSC 1					x							x
RSC 2					x							x
PN-IP-M1	x				x							
PN-IP-M2								x				

[1)] Burst access by the AHB master can be interrupted with the AHB burst breaker (see chapter 2.3.4.3.3).

Note

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

The ARM926-D and ARM926-I masters have burst breakers fitted at the AHB access point. A burst breaker generates an idle phase for a clock after a configured number of consecutive address phases (see chapter 2.3.4.3.3). A different AHB master can then be assigned the AHB slave during this idle phase. This is possible because re-arbitration at an AHB slave can only be carried out if a master releases that AHB slave (idle phase).

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

2.3.4.3.3 AHB Burst Breaker

To improve the real-time capabilities of ERTEC 200P, three burst breakers are used to prevent ARM926-D and ARM926-I to block resources by consecutive burst transfers without IDLE cycles in between, which are necessary for re-arbitration of AHB masters.

The burst breaker counts the address phases of following ARM926-D/ARM926-I AHB transfers:

- SINGLE
- INCR unspecified length

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

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

If the number of consecutive address phases at ARM926-D AHB port / ARM926-I AHB port / AHB port without an IDLE cycle in between exceeds NR_ADDR_ARM926-D / NR_ADDR_ARM926-I / in the 'AHB_BURSTBREAKER' register (see chapter 2.3.14.11.27.1, configured by SW), the burst breaker will insert an IDLE cycle immediately if the current transfer is SINGLE or INCR unspecified length. In the latter case the INCR will be split. If the current transfer is any kind of burst with specified length the IDLE insertion will be done at the end of the ongoing burst transfer (to not interrupt any cache transfers).

NR_ADDR_ARM926-D/NR_ADDR_ARM926-I set to ‘0’ means burst breaker at ARM926-D AHB port/ARM926-I AHB port is set to transparent mode. Any transfer is passed through as is; no IDLE insertion.

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

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

2.3.4.4 APB subsystem (Peripheral Bus)

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

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

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

Copyright © Siemens AG 2023. All rights reserved

- PerIF (APB interface)
- I-Filter

A second AHB/APB bridge is used to connect the OSPI to the AHB interconnect.

The APB bus is a 32-bit wide bus. The bus frequency is 125 MHz. The APB bus in the ERTEC 200P has a wider functionality than the ARM-APB bus. Unlike the standard APB bus, it offers the option of variable timing (wait states) and byte enables. Wait state and byte enable capability is achieved by introducing sideband signals. This allows byte-by-byte access to peripherals.

The APB modules allow the following types of access.

Table 13: Access types to APB slaves

APB module	Access permitted [1)]	Comment
SCRB	Only 32-bit	Data bus: 32-bit
Boot ROM	8/16/32-bit	Data bus: 32-bit
Host interface (XHIF)	Only 32-bit	Data bus: 32-bit
UART	16/32-bit	Data bus: 16-bit, i.e., the top 16 bits are ignored
F-timer	Only 32-bit	Data bus: 32-bit
Timer 0-5	Only 32-bit	Data bus: 32-bit
Watchdog	Only 32-bit	Data bus: 32-bit
I2C	8/16/32-bit	Data bus: 8-bit, i.e., the top 24 bits are ignored
SPI	16/32-bit	Data bus: 16-bit, i.e., the top 16 bits are ignored
GPIO	Only 32-bit	Data bus: 32-bit
I-Filter	8/16/32-bit	Data bus: 32-bit
Peripheral interface	8/16/32-bit	Data bus: 32-bit
SCRB	Only 32-bit	Data bus: 32-bit
Octal SPI	Only 32-bit	Data bus: 32-bit

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

Note

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

Note

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

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

2.3.4.4.1 AHB/APB bridge

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

Copyright © Siemens AG 2023. All rights reserved

Figure 8: AHB/APB bridge, block diagram

The key features of the bridge are as follows:

- On the AHB side the AHB_APB_Bridge fully supports the AMBA 2.0 AHB protocol. It only uses the AHB-Lite subset of this protocol. The bridge is connected on the AHB layer as a slave.
- On the APB side the AHB_APB_Bridge supports the AMBA 2.0 APB protocol. Accesses on the AHB side are one by one translated into APB accesses, write accesses are handled as posted, i. e. AHB_APB_Bridge completes the access on the AHB side as soon as it is able to start it on the APB side.
- The AHB_APB_Bridge is not able to produce back-to-back accesses on the APB side. In difference to the APB specification, it uses an IDLE state after every APB access even when there already is an AHB access waiting to be translated. AHB bursts are treated as sequences of single accesses.
- Expanding the AMBA 2.0 - APB specification, the AHB_APB_Bridge provides the option of using a variable timing of APB accesses. By introducing a ready signal, wait states can be implemented. The standard AMBA-APB bus is expanded by the PRDY sideband signal (see Figure 9, timing with sideband signals PRDY and PBE).
- Expanding the AMBA 2.0 - APB specification, the AHB_APB_Bridge provides the option of using "byte enables" on APB bus. Therefore, the standard AMBA-APB bus is expanded by the sideband signals PBExx which exist in parallel to PADDR, i.e., PADDR and PBE timing are identical (see Figure 9, timing with sideband signals PRDY and PBE).
- All signals are clocked. There are no snake paths from AHB side to APB side and vice versa.
- AHB frequency has to be the same as the APB frequency.
- The AHB data bus width is the same as the APB data bus width (32 bits).
- The AHB_APB_Bridge supports only one slave. Address decoding and multiplexing for several slaves are carried out outside the bridge, e.g., APB decoder.
- The AHB_APB_Bridge is implemented fully synchronously. The rising clock edge is the active one.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

- The reset is designed asynchronously. The reset is not synchronized in the AHB_APB_Bridge therefore synchronous reset has to be applied to meet recovery/removal time requirements.

Bridge timing with the additional signals PBE and PRDY

Figure 9: APB bridge, timing

2.3.4.4.2 APB decoder

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

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

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

2.3.4.4.2.1 Boot ROM accessibility

The Boot ROM is only accessible by the instruction master of the ARM926.

2.3.4.5 Address Range and Acknowledgement Delay Monitoring

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

2.3.4.5.1 Monitoring at the AHB side

Each AHB master is assigned its own address range monitoring. If an AHB master accesses an address range that is not in use, the access is acknowledged with an error response and an interrupt IRQ51 (see chapter 2.3.5.9) is triggered in the ARM interrupt controller. The address of the

Copyright © Siemens AG 2023. All rights reserved

erroneous access is also saved in SCRB register 'QVZ_AHB_ADR' and the corresponding access type (AHB control signals: Read/Write, HBURST, HSIZE) in SCRB register 'QVZ_AHB_CTRL'.

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

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

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

If multiple AHB masters cause an access violation at the same time (within one AHB clock cycle), only the violation by the highest priority AHB master (prioritization as detailed in chapter 2.3.4.3.1) is displayed.

Note

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

Address gaps in the APB address range are **not** visible at the AHB side; the monitoring mechanisms in chapter 2.3.4.5.2 apply.

2.3.4.5.2 Monitoring at the APB side

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

An IRQ52 interrupt (see chapter 2.3.5.9) is triggered in the ARM interrupt controller. The address of the erroneous access is saved in SCRB register 'QVZ_APB_ADR' (see 2.3.14.11.5).

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

2.3.4.5.3 Monitoring at the EMC

The external ready signal XRDY_PER is monitored in the EMC. If one of the 4 external memory areas selected with the output pins XCS_PER(3:0) is addressed, the memory controller waits for the input signal XRDY_PER (provided ready control is activated in the corresponding configuration register 'ASYNC_BANKx'; see chapter 2.3.8.7). Acknowledgement delay monitoring is activated in the 'EXTENDED_CONFIG' register (see 2.3.8.7) and generates a ready signal internally for the memory controller and IRQ53 (see 2.3.5.9) in the ARM interrupt controller after a max. of (1048575 + 1) × 16 AHB clock cycles. The duration of monitoring is set in the 'ASYNC_WAIT_CYCLE_CONFIG' register (see 2.3.8.7). For further information, see 2.3.8.4.5.

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

Copyright © Siemens AG 2023. All rights reserved

2.3.4.5.4 Monitoring in the individual modules

Implementation within the modules (IPs)

The following modules have monitoring for unassigned addresses:

- PN-IP (PROFINET-IP)
- PerIF (peripheral interface)
- I-Filter
- Host IF (host interface)
- SCRB
- RSC 1 / 2

Access over the AHB and APB that addresses an undefined address range within a module (IP):

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

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

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

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

- Write access to read-only registers
- Write and read access to address gaps in register and RAM address ranges
- Byte and halfword write access to addresses that can be written word by word.

Copyright © Siemens AG 2023. All rights reserved

As an example, here is the 'Access_Error' register of the RS controller.

Table 14: 'Access_Error' register (of RS controller)

Register:	Access_Error				Address:	2Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	r(h)	(w)
Description:	Access Error Register						
Bit	Identifier	Reset	Attr.		Function / Description		
8dt0	ahb_address	0h	rh		AHB address in RS IP where the error occurred		
26dt9	reserved	00000h					
27	ahb_apb_select	0h	rh		0 = AHB-Access, 1 = APB_acces (not relevant)		
29dt28	ahb_size	0h	rh		00 = Byte access (8 bit), 01 = HW access (16 bit), 10 = W access (32 bit), 11 = reserved		
30	ahb_rd_wr	0h	rh		0 = RD access, 1 = WR access		
31	err_lock	0h	r	w	Set by HW to 1 on access error, locked for futher entries until cleared by SW (write 0 to clear) 0 = no access error 1 = access error occurred		

Implementation in ERTEC 200P top-level

The access_err_irq_o events of the individual modules are high-active for **a 125 MHz system clock** in the top-level and are all set out in the 'MODUL_ACCESS_ERR' register (see chapter 2.3.14.11.5).

An entry in 'MODUL_ACCESS_ERR' triggers the ARM926 interrupt Modul_Access_Error (see 2.3.5.9). Following a Modul_Access_Error interrupt, the SW must read the 'MODUL_ACCESS_ERR' register to identify the module affected. The SW must then read the detailed data for the erroneous access from the access error register in the relevant module and clear the register (ERR_LOCK = '0'). The SW must subsequently reset the 'MODUL_ACCESS_ERROR' register in the SCRB by writing the value 0x0.

In the event of read access to unassigned addresses in AHB modules, an AHB error response is also generated on the AHB (PN-IP, PerIF). This triggers an exception in ARM926EJ-S.

Copyright © Siemens AG 2023. All rights reserved

2.3.5 ARM-ICU (ARM926 Interrupt Controller)

2.3.5.1 Overview

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

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

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

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

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

2. The second functional unit is responsible for decoding the priorities (priority resolving) for each interrupt. It is determined in each clock:
   - Whether a pending interrupt will be forwarded to the third functional unit
   - If several interrupts are pending, which of these have the highest priority.
3. This interrupt (or its priority) is compared in the third functional unit (postprocessing, ICU-ISR) with any interrupt currently being processed by the software (using the priority).
   - If the pending interrupt is valid, namely has the appropriate priority, it will be reported to the CPU.
   - The CPU must confirm each interrupt with an access to the Acknowledge register ('ACK'). This causes the appropriate bit for the interrupt number to be set in the In-Service-Register (ISR) and cleared in the IRR.

After completion of the Interrupt Service Routine by the CPU, the ICU requires an End-of-Interrupt command (EOI, a write access of the CPU to the register of the same name) to inform the ICU about the end of the interrupt processing and to initiate the clearing of the corresponding bits from the ISR.

2.3.5.2 Interface

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

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

Copyright © Siemens AG 2023. All rights reserved

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

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

Each interrupt source of the INTA subblock can be placed on each interrupt input of the INTB subblock, where the selected interrupt sources also continue to remain active in the INTA subblock. The "INTB0" is the exception, because no INTA can be placed on it. For the configuration of the assignment by the software, the numbers of the selected INTA inputs are entered in the FIQ_SEL_1 – FIQ_SEL_<NUM_OF_FINT-1> registers. Although the FIQ_SEL_0 register can be read and written, it is not further processed by the hardware.

Note

Both INTAs and INTBs are edge-triggered as initial setting. If level triggering is set, the interrupt at the source must be removed, otherwise it will issue a new interrupt (INTA or INTB) to the CPU after the acknowledge.

Note further that **only high active level (in level triggering mode) is supported by ICU whereas in edge triggering mode both edges are supported**. INTA and INTB must always be set to the same triggering. Because this is not checked by the ICU, it must be ensured by the software. In addition, the software must mask the INTA input in the INTA subblock used as INTB, because otherwise both an INTA and an INTB will be signaled to the CPU when the interrupt occurs.

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

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

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

2.3.5.3 Interrupt acquisition

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

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

nevertheless performs an acknowledge, the ICU returns the default vector (null vector) as interrupt vector.

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

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

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

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

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

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

NOTICE

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

2.3.5.4 Interrupt masking / priorisation

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

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

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

Copyright © Siemens AG 2023. All rights reserved

Note

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

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

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

Where the "0" value means the highest priority. After reset, all registers have the lowest priority. The ICU treats all interrupts in their priority order. If several interrupts are present, the ICU selects the interrupt with the highest priority. If in this case two priorities are identical, the interrupt with the smaller number will be processed first.

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

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

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

This also means, in the same clock in which the information is written from the AHB bus into the 'LOCKREG' register. If the interrupt request has already been pending longer without it being acknowledged and the software now writes this LOCKREG with the same or higher priority, the default vector will be entered as interrupt vector also two clocks after the LOCKREG was written.

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

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

2.3.5.5 Interrupt post-processing

The acknowledge of interrupts by the CPU is performed with a read access to the INTAACK or INTBACK register, where the ICU returns the number of the current pending interrupt with the highest priority. The bit with the corresponding number will be set in the In-Service register (ISR) and the INTA or INTB signal to the CPU is removed.
The ICU outputs a default vector when an acknowledge is performed although no interrupt is present on the CPU. The null vector is used as default vector. The internal status of the ICU with regard to IRR and ISR does not change.

A default vector can occur:

1. When the ICU receives an acknowledge command without it previously outputting a pending interrupt.
2. When the source of a level-triggered interrupt removes it before being confirmed by the CPU.

Copyright © Siemens AG 2023. All rights reserved

3. When an interrupt input is masked or a clear command is performed for an interrupt input while concurrently an interrupt occurs at this input. Because the CPU commands arrive at the interrupt controller with a time delay, the interrupt is initially forwarded to the CPU, however, removed after the command takes effect. If, however, the CPU responds with acknowledge, the interrupt controller can only assign the default vector, because in the meantime a valid interrupt is no longer present.
4. If the time between acknowledgment and the End of Interrupt event is less than 3 ICU clock cycles, the Interrupt line to the CPU may rise again, although there is no interrupt present. In this case the CPU will read the default vector.

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

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

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

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

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

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

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

2.3.5.6 Special functions

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

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

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

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

2.3.5.7 Debug functions

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

Copyright © Siemens AG 2023. All rights reserved

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

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

2.3.5.8 Miscellaneous

2.3.5.8.1 Synchronizing the inputs

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

Note

All level-triggered interrupts have to be high active. **Low Active level-triggered interrupts are not supported by ICU** and must be inverted on ERTEC 200P top-level.

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

2.3.5.8.2 Bus interface

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

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

2.3.5.8.3 Sequences

The following is a typical interrupt cycle:

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

Copyright © Siemens AG 2023. All rights reserved

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

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

2.3.5.8.4 Operating rules

Reconfiguration of priorities

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

Level-sensitive interrupts

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

Acknowledgement of the INTBs with write access to acknowledge register

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

Using INTAs as INTBs

INTA and INTB always have to be set to the same triggering. This is not checked by the ICU and has to be ensured by the software. Moreover, the SW must mask the INTA input in the INTA subblock used as INTB, because otherwise both an INTA and an INTB will be signaled to the CPU when the interrupt occurs.

Identical priorities

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

Copyright © Siemens AG 2023. All rights reserved

Hardware latencies in certain back2back software accesses

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

- IRCLVEC -> ACK
- IRCLVEC -> IRR0-2
- IRCLVEC -> SWIRREG0-2
- LOCKREG -> ACK
- MASKREG0-2 -> ACK
- ACK -> IRR0-2

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

Deletion of interrupts during the reset phase

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

Procedure for locking priorities via LOCKREG

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

2.3.5.8.5 Startup/shutdown

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

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

The INTA applied to the INTB always must be set to the same triggering. This is not checked by the ICU and must be ensured by the software. Moreover, the SW must mask the INTA input in the

Copyright © Siemens AG 2023. All rights reserved

INTA subblock used as INTB, because otherwise both an INTA and an INTB will be signaled to the CPU when the interrupt occurs.

2.3.5.9 Interrupt sources for ARM-IRQ

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

Table 15: Host INTA, interrupt sources

IRQ no.	Interrupt name	Type	Signal	Cause	Interrupt source
0	Default interrupt	-	Default Vector	Interrupt which is never assigned (must be tied to '0' on top-level)	
1	DMA_IRQ_O	LT	DMA_IRQ_O	GDMA Interrupt	GDMA
2	INT_I2C	LT	Intr	Combined Interrupt	I²C
3	reserved		-----		
4	reserved		-----		
5	INT_UART1	LT	UARTINTR (group interrupt)	Receive FIFO half-full or full Transmit FIFO half-empty or empty Receive Timeout	UART1
6	INT_UART1_ERR	LT	UARTEINTR (error interrupt)	Error interrupt	
7	INT_UART2	LT	UARTINTR (group interrupt)	Receive FIFO half-full or full Transmit FIFO half-empty or empty Receive Timeout	UART2
8	INT_UART2_ERR	LT	UARTEINTR (error interrupt)	Error interrupt	
9	INT_UART3	LT	UARTINTR (group interrupt)	Receive FIFO half-full or full Transmit FIFO half-empty or empty Receive Timeout	UART3
10	INT_UART3_ERR	LT	UARTEINTR (error interrupt)	Error interrupt	
11	INT_UART4	LT	UARTINTR (group interrupt)	Receive FIFO half-full or full Transmit FIFO half-empty or empty Receive Timeout	UART4
12	INT_UART4_ERR	LT	UARTEINTR (error interrupt)	Error interrupt	
13	INT_SPI1	LT	SSPINTR (group interrupt)	Transmit FIFO empty Transmit FIFO half-empty or empty Transmit FIFO not full Receive FIFO not empty Receive FIFO half-full or full Receive FIFO overrun	SPI1
14	INT_SPI1_Overrun	LT	SSPRORINTR (error interrupt)	Overrun Error Interrupt	
15	INT_SPI2	LT	SSPINTR (group interrupt)	Transmit FIFO empty Transmit FIFO half-empty or empty Transmit FIFO not full Receive FIFO not empty Receive FIFO half-full or full Receive FIFO overrun	SPI2
16	INT_SPI2_Overrun	LT	SSPRORINTR (error interrupt)	Overrun Error Interrupt	
17	INT_SPI_ParityERR	ET/R	SPI_PAR_ERR_IRQ	Interrupt for SPI1/2_PARITY_ERR	SCRB
18	reserved		-----		

Copyright © Siemens AG 2023. All rights reserved

IRQ no.	Interrupt name	Type	Signal	Cause	Interrupt source
19	reserved		-----		
20	reserved		-----		
21	INT_TIMER0	*)	TIM_OUT(0)	Timer expired (value 0 reached)	TIMER
22	INT_TIMER1	*)	TIM_OUT(1)	Timer expired (value 0 reached)	
23	INT_TIMER2	*)	TIM_OUT(2)	Timer expired (value 0 reached)	
24	INT_TIMER3	*)	TIM_OUT(3)	Timer expired (value 0 reached)	
25	INT_TIMER4	*)	TIM_OUT(4)	Timer expired (value 0 reached)	
26	INT_TIMER5	*)	TIM_OUT(5)	Timer expired (value 0 reached)	
27	WD_INT_ARM926	LT	WD_INT	ARM926 Watchdog Interrupt	WD
28	SPI1_TFE	LT	SSP_TFE	Transmit FIFO empty	SPI1
29	SPI1_RNE	LT	SSP_SSPRXDMA	Receive FIFO not empty (corresponds to SPI1_SSPRXDMA)	
30	SPI2_TFE	LT	SSP_TFE	Transmit FIFO empty	SPI2
31	SPI2_RNE	LT	SSP_SSPRXDMA	Receive FIFO not empty (corresponds to SPI2_SSPRXDMA)	
32	INT_GPIO0	**)	GPIO0	External interrupt over input pin GPIO 0	GPIO
33	INT_GPIO1	**)	GPIO1	External interrupt over GPIO 1 input pin	
34	INT_GPIO2	**)	GPIO2	External interrupt over GPIO 2 input pin	
35	INT_GPIO3	**)	GPIO3	External interrupt over GPIO 3 input pin	
36	INT_GPIO4	**)	GPIO4	External interrupt over GPIO 4 input pin	
37	INT_GPIO5	**)	GPIO5	External interrupt over GPIO 5 input pin	
38	INT_GPIO6	**)	GPIO6	External interrupt over GPIO 6 input pin	
39	INT_GPIO7	**)	GPIO7	External interrupt over GPIO 7 input pin	
40	INT_GPIO8	**)	GPIO8	External interrupt over GPIO 8 input pin	
41	INT_GPIO9	**)	GPIO9	External interrupt over GPIO 9 input pin	
42	INT_GPIO10	**)	GPIO10	External interrupt over GPIO 10 input pin	
43	INT_GPIO11	**)	GPIO11	External interrupt over GPIO 11 input pin	
44	INT_GPIO12	**)	GPIO12	External interrupt over GPIO 12 input pin	
45	INT_GPIO13	**)	GPIO13	External interrupt over GPIO 13 input pin	
46	INT_GPIO14	**)	GPIO14	External interrupt over GPIO 14 input pin	
47	INT_GPIO15	**)	GPIO15	External interrupt over GPIO 15 input pin	
48	EDC_Event	LT	EDC_IRQ_O	Combined Interrupt ('EDC_EVENT' register)	SCRB
49	INT_PLL_UNLOCK	ET/R	**Not** LOCK_PLL_MONI	Interrupt if PLL is not locked.	PLL

Copyright © Siemens AG 2023. All rights reserved

IRQ no.	Interrupt name	Type	Signal	Cause	Interrupt source
50	INT_PLL_LOSS	ET/R	LOSS_PLL_MONI	Interrupt if oscillator clock is not in the monitored range.	
51	AHB Address Error	ET/R	QVZ_AHB_IRQ_O (Group interrupt for the individual masters at the AHB)	Access to unavailable address at the AHB	SCRB (QVZ)
52	APB Address Error	ET/R	QVZ_APB_IRQ_O (Group interrupt for the individual slaves at the AHB/APB)	Access to unavailable address at the APB	
53	EMC Address Error	ET/R	qcz_irq	Access to unavailable address in the EMC address range	AHB_EMC
54	PER_IF_ARM_IRQ	ET/R	ARM_IRQ	Event unit group interrupt	PerIF
55	PHY1/2_INT	ET/R	IRQ	PHY1/2 interrupt (ORed)	PHY1, 2
56	PN_IRQ2(0)	ET/R	pn_irq2_o(0)	PN-ICU2 Combined Interrupt 1	PN-IP
57	PN_IRQ2(1)	ET/R	pn_irq2_o(1)	PN-ICU2 Combined Interrupt 2	
58	PN_IRQ2(2)	ET/R	pn_irq2_o(2)	Selectable PN-ICU Interrupt	
59	PN_IRQ2(3)	ET/R	pn_irq2_o(3)	Selectable PN-ICU Interrupt	
60	PN_IRQ2(4)	ET/R	pn_irq2_o(4)	Selectable PN-ICU Interrupt	
61	PN_IRQ2(5)	ET/R	pn_irq2_o(5)	Selectable PN-ICU Interrupt	
62	PN_IRQ2(6)	ET/R	pn_irq2_o(6)	Selectable PN-ICU Interrupt	
63	PN_IRQ2(7)	ET/R	pn_irq2_o(7)	Selectable PN-ICU Interrupt	
64	PN_IRQ2(8)	ET/R	pn_irq2_o(8)	Selectable PN-ICU Interrupt	
65	PN_IRQ2(9)	ET/R	pn_irq2_o(9)	Selectable PN-ICU Interrupt	
66	PN_IRQ2(10)	ET/R	pn_irq2_o(10)	Selectable PN-ICU Interrupt	
67	PN_IRQ2(11)	ET/R	pn_irq2_o(11)	Selectable PN-ICU Interrupt	
68	PN_IRQ2(12)	ET/R	pn_irq2_o(12)	Selectable PN-ICU Interrupt	
69	PN_IRQ2(13)	ET/R	pn_irq2_o(13)	Selectable PN-ICU Interrupt	
70	PN_IRQ2(14)	ET/R	pn_irq2_o(14)	Selectable PN-ICU Interrupt	
71	PN_IRQ2(15)	ET/R	pn_irq2_o(15)	Selectable PN-ICU Interrupt	
72	PNPLL_OUT9	ET/R	pnip_pn_pll_o(9)	Selectable PN-PLL Event	
73	PNPLL_OUT10	ET/R	pnip_pn_pll_o(10)	Selectable PN-PLL Event	
74	PNPLL_OUT11	ET/R	pnip_pn_pll_o(11)	Selectable PN-PLL Event	
75	PNPLL_OUT12	ET/R	pnip_pn_pll_o(12)	Selectable PN-PLL Event	
76	PNPLL_OUT13	ET/R	pnip_pn_pll_o(13)	Selectable PN-PLL Event	
77	PNPLL_OUT14	ET/R	pnip_pn_pll_o(14)	Selectable PN-PLL Event	
78	SW_INT_0	-	not used in HW		(SW)
79	SW_INT_1	-	not used in HW		
80	reserved	-			
81	reserved	-			
82	reserved	-			
83	reserved	-			
84	OSPI_INT	LT	ospi_ssi_intr	Accumulated interrupt	Octal-SPI
85	reserved	-			(SW)
86	IP Access Error	ET/R	ACCESS_ERR_IRQ	Combined Interrupt for address mismatches in modules	SCRB (QVZ)
87	ETB_ACQCOMP	ET/R	IR87_ETB_ACQCOMP	Combined Interrupt for address mismatches in modules	ARM926
88	Invalid I-TCM926 Access	ET/R	IR88_INV_ACCESS_ITCM926	ARM926 has accessed an I-TCM address gap	
89	Invalid D-TCM926 Access	ET/R	IR89_INV_ACCESS_DTCM926	ARM926 has accessed a D-TCM address gap	
90	ETB_FULL	ET/R	IR90_ETB_FULL	Trace overrun in trace buffer	
91	RSC_INT	ET/R	rsc_int_o (group interrupt from RSC 1) **or** rsc_int_o (group interrupt from RSC 2)	Group interrupt Encoder ready Decoder ready w/o error Decoder ready w error Decoder response error	RSC 1 **or** RSC 2
92	reserved		-----		
93	reserved		-----		

IRQ no.	Interrupt name	Type	Signal	Cause	Interrupt source
94	COMMRX926	LT	IR94_COMMRX926	Interrupt for Real-time Debugger (receive Buffer contains data)	ARM926
95	COMMTX926	LT	IR95_COMMTX926	Interrupt for Real-time Debugger (transmit buffer is empty)	

Types:

ET/R Edge-triggered interrupt, on the rising edge
ET/F Edge-triggered interrupt, on the falling edge
LT Level-triggered interrupt
*) This type depends on the selected mode of the module
**) This type is dependent on the external switch (ASIC-Pins). It can be ET/F or ET/R or LT, depending on the external connection. Parameterization must be done depending on the request of the application about the Host-ICU (see chapter 2.3.5.12).

Note

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

2.3.5.9.1 Interrupts for accesses to missing addresses

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

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

In the address area of the EMC, the XCE_PER(3:0) outputs become active and the Memory Controller waits for the XRDY_PER input signal. The confirmation delay monitoring activated in the 'ASYNC_WAIT_CYCLE_CONFIG' register (see chapter 2.3.8.7) after ((MAX_EXT_WAIT + 1) × 16) AHB clock pulses creates an internal ready signal for the Memory Controller and an IRQ. The IRQ will be removed when the QVZ monitoring is disabled.

2.3.5.10 Interrupt sources for ARM-FIQ

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

Table 16: Host INTB, interrupt sources

IRQ no.	Interrupt name	Type	Signal	Cause	Interrupt source
0	Default interrupt	-	Default Vector	Interrupt which is never assigned (must be tied to '0' on top-level)	
1	FIQ_SEL_1	*)		Selectable interrupt from the INTA sources about *FIQ_SEL_1* register	
2	FIQ_SEL_2	*)		Selectable interrupt from the INTA sources about *FIQ_SEL_2* register	
3	FIQ_SEL_3	*)		Selectable interrupt from the INTA sources about *FIQ_SEL_3* register	
4	FIQ_SEL_4	*)		Selectable interrupt from the INTA sources about *FIQ_SEL_4* register	
5	FIQ_SEL_5	*)		Selectable interrupt from the INTA sources about *FIQ_SEL_5* register	
6	FIQ_SEL_6	*)		Selectable interrupt from the INTA sources about *FIQ_SEL_6* register	

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

IRQ no.	Interrupt name	Type	Signal	Cause	Interrupt source
7	FIQ_SEL_7	*)		Selectable interrupt from the INTA sources about *FIQ_SEL_7* register	

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

Note

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

2.3.5.10.1 High-priority interrupts for debugging

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

Copyright © Siemens AG 2023. All rights reserved

2.3.5.11 Address mapping

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

Module	Register/Memory	Read	Write	Address
/icu_ertec_addr_dec_top/icu96_inst				
	ID_REGISTER	r		4000h
	IRVEC	rh		4004h
	ACK	rht		4008h
	IRCLVEC		wt	400Ch
	MASKALL	r	w	4010h
	EOI		t	4014h
	UNLOCK_RD_ONLY_ACK	r	w	4018h
	MASK_ALL_INPUT_EN	r	w	401Ch
	LOCKREG	r	w	4020h
	MASKREG0	r	w	5000h
	MASKREG1	r	w	5004h
	MASKREG2	r	w	5008h
	IRR0	rh		5100h
	IRR1	rh		5104h
	IRR2	rh		5108h
	ISR0	rh		5200h
	ISR1	rh		5204h
	ISR2	rh		5208h
	TRIGREG0	r	w	5300h
	TRIGREG1	r	w	5304h
	TRIGREG2	r	w	5308h
	EDGEREG0	r	w	5400h
	EDGEREG1	r	w	5404h
	EDGEREG2	r	w	5408h
	SWIRREG0	rh	w	5500h
	SWIRREG1	rh	w	5504h
	SWIRREG2	rh	w	5508h
	PRIOREG0	r	w	6000h
	PRIOREG1	r	w	6004h
	PRIOREG2	r	w	6008h
	PRIOREG3	r	w	600Ch
	PRIOREG4	r	w	6010h
	PRIOREG5	r	w	6014h
	PRIOREG6	r	w	6018h
	PRIOREG7	r	w	601Ch
	PRIOREG8	r	w	6020h
	PRIOREG9	r	w	6024h
	PRIOREG10	r	w	6028h
	PRIOREG11	r	w	602Ch
	PRIOREG12	r	w	6030h

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

	PRIOREG13	r	w	6034h
	PRIOREG14	r	w	6038h
	PRIOREG15	r	w	603Ch
	PRIOREG16	r	w	6040h
	PRIOREG17	r	w	6044h
	PRIOREG18	r	w	6048h
	PRIOREG19	r	w	604Ch
	PRIOREG20	r	w	6050h
	PRIOREG21	r	w	6054h
	PRIOREG22	r	w	6058h
	PRIOREG23	r	w	605Ch
	PRIOREG24	r	w	6060h
	PRIOREG25	r	w	6064h
	PRIOREG26	r	w	6068h
	PRIOREG27	r	w	606Ch
	PRIOREG28	r	w	6070h
	PRIOREG29	r	w	6074h
	PRIOREG30	r	w	6078h
	PRIOREG31	r	w	607Ch
	PRIOREG32	r	w	6080h
	PRIOREG33	r	w	6084h
	PRIOREG34	r	w	6088h
	PRIOREG35	r	w	608Ch
	PRIOREG36	r	w	6090h
	PRIOREG37	r	w	6094h
	PRIOREG38	r	w	6098h
	PRIOREG39	r	w	609Ch
	PRIOREG40	r	w	60A0h
	PRIOREG41	r	w	60A4h
	PRIOREG42	r	w	60A8h
	PRIOREG43	r	w	60ACh
	PRIOREG44	r	w	60B0h
	PRIOREG45	r	w	60B4h
	PRIOREG46	r	w	60B8h
	PRIOREG47	r	w	60BCh
	PRIOREG48	r	w	60C0h
	PRIOREG49	r	w	60C4h
	PRIOREG50	r	w	60C8h
	PRIOREG51	r	w	60CCh
	PRIOREG52	r	w	60D0h
	PRIOREG53	r	w	60D4h
	PRIOREG54	r	w	60D8h
	PRIOREG55	r	w	60DCh
	PRIOREG56	r	w	60E0h
	PRIOREG57	r	w	60E4h
	PRIOREG58	r	w	60E8h
	PRIOREG59	r	w	60ECh
	PRIOREG60	r	w	60F0h
	PRIOREG61	r	w	60F4h
	PRIOREG62	r	w	60F8h

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

	PRIOREG63	r	w	60FCh
	PRIOREG64	r	w	6100h
	PRIOREG65	r	w	6104h
	PRIOREG66	r	w	6108h
	PRIOREG67	r	w	610Ch
	PRIOREG68	r	w	6110h
	PRIOREG69	r	w	6114h
	PRIOREG70	r	w	6118h
	PRIOREG71	r	w	611Ch
	PRIOREG72	r	w	6120h
	PRIOREG73	r	w	6124h
	PRIOREG74	r	w	6128h
	PRIOREG75	r	w	612Ch
	PRIOREG76	r	w	6130h
	PRIOREG77	r	w	6134h
	PRIOREG78	r	w	6138h
	PRIOREG79	r	w	613Ch
	PRIOREG80	r	w	6140h
	PRIOREG81	r	w	6144h
	PRIOREG82	r	w	6148h
	PRIOREG83	r	w	614Ch
	PRIOREG84	r	w	6150h
	PRIOREG85	r	w	6154h
	PRIOREG86	r	w	6158h
	PRIOREG87	r	w	615Ch
	PRIOREG88	r	w	6160h
	PRIOREG89	r	w	6164h
	PRIOREG90	r	w	6168h
	PRIOREG91	r	w	616Ch
	PRIOREG92	r	w	6170h
	PRIOREG93	r	w	6174h
	PRIOREG94	r	w	6178h
	PRIOREG95	r	w	617Ch
/icu_ertec_addr_dec_top/icu8_inst				
	ID_REGISTER	r		8000h
	IRVEC	rh		8004h
	ACK	rht		8008h
	IRCLVEC		wt	800Ch
	MASKALL	r	w	8010h
	EOI		t	8014h
	UNLOCK_RD_ONLY_ACK	r	w	8018h
	MASK_ALL_INPUT_EN	r	w	801Ch
	LOCKREG	r	w	8020h
	MASKREG0	r	w	9000h
	IRR0	rh		9100h
	ISR0	rh		9200h
	TRIGREG0	r	w	9300h
	EDGEREG0	r	w	9400h
	SWIRREG0	rh	w	9500h

Copyright © Siemens AG 2023. All rights reserved

	PRIOREG0	r	w	A000h
	PRIOREG1	r	w	A004h
	PRIOREG2	r	w	A008h
	PRIOREG3	r	w	A00Ch
	PRIOREG4	r	w	A010h
	PRIOREG5	r	w	A014h
	PRIOREG6	r	w	A018h
	PRIOREG7	r	w	A01Ch
/icu_ertec_addr_dec_top/icu_general_registers_inst				
	reserved	r	w	48000h
	FIQ_SEL_1	r	w	48004h
	FIQ_SEL_2	r	w	48008h
	FIQ_SEL_3	r	w	4800Ch
	FIQ_SEL_4	r	w	48010h
	FIQ_SEL_5	r	w	48014h
	FIQ_SEL_6	r	w	48018h
	FIQ_SEL_7	r	w	4801Ch

2.3.5.12 Register description

Module: /icu_ertec_addr_dec_top/icu96_inst

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

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

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

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

Register:	MASKALL			Address:	4010h	
Bits:	0	Reset value:	1h	Attributes:	r	w
Description:	Mask all interrupts Global lock of all IRQ interrupt inputs '0' = Enable all unmasked IRQ interrupt inputs (use the set mask bits) '1' = Global lock of all IRQ interrupt inputs (independent of the interrupt mask)					

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

Register:	UNLOCK_RD_ONLY_ACK	Address:	4018h

Bits:	0	Reset value:	0h	Attributes:	r	w
Description:	Unlocks the read only acknowledge mechanism. By setting this bit to 1 one can reach an acknowledge via a write access to ACK 0: Read only mechanism is active to acknowledge an Interrupt Request 1: Also a write access to ACK acknowledges the Interrupt Request. Be aware that this write access is destructive.					

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

Register:	LOCKREG			Address:	4020h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Priority Lock Register Specification of a priority to lock interrupt requests with lower or equal priority					

Bit	Identifier	Reset	Attr.		Function / Description
6dt0	LOCK_PRIO	00h	r	w	Binary code of the locking priority
31	LOCK-REG_ENABLE	0h	r	w	0 = lock inactive / 1 = lock active

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

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

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

Register:	IRR0			Address:	5100h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Request Register Flag for the Interrupt Request detected as result of a positive edge					

	0 – 31 inputs of the interrupt controller '0' = No request '1' = Request has occurred

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

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

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

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

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

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

Register:	TRIGREG1	Address:	5304h

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

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

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

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

Register:	EDGEREG2			Address:	5408h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Edge Select Register Select the edge for the interrupt detection (only when edge detection has been set for the associated input) Interrupt inputs 64 – 95 0 = Interrupt detection for positive edge 1 = Interrupt detection for negative edge					

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

Register:	SWIRREG1			Address:	5504h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	w

Copyright © Siemens AG 2023. All rights reserved

Description:	Software Interrupt Register Specification of interrupt requests Interrupt inputs 32 – 63 0 = No interrupt request 1 = Set interrupt request

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

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

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

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

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

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

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

Register:	PRIOREG6			Address:	6018h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w

Copyright © Siemens AG 2023. All rights reserved

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

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

Register:	PRIOREG8			Address:	6020h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

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

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

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

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

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

Register:	PRIOREG14			Address:	6038h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Copyright © Siemens AG 2023. All rights reserved

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

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

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

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

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

Register:	PRIOREG20			Address:	6050h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

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

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

Register:	PRIOREG23			Address:	605Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

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

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

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

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

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

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

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

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

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

Copyright © Siemens AG 2023. All rights reserved

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

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

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

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

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

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

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

Register:	PRIOREG39			Address:	609Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Register:	PRIOREG40			Address:	60A0h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w

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

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

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

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

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

Register:	PRIOREG45			Address:	60B4h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

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

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

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

Copyright © Siemens AG 2023. All rights reserved

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

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

Register:	PRIOREG51			Address:	60CCh	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

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

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

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

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

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

Register:	PRIOREG57			Address:	60E4h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w

Copyright © Siemens AG 2023. All rights reserved

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

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

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

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

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

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

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

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

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

Copyright © Siemens AG 2023. All rights reserved

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

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

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

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

Register:	PRIOREG70			Address:	6118h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

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

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

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

Register:	PRIOREG74			Address:	6128h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

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

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

Register:	PRIOREG76			Address:	6130h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

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

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

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

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

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

Register:	PRIOREG82			Address:	6148h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w
Description:	Priority Register Specification of the priority of an interrupt request at the associated Binary code of the priority input					

Copyright © Siemens AG 2023. All rights reserved

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

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

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

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

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

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

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

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

Register:	PRIOREG91			Address:	616Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w

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

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

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

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

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

Module: /icu_ertec_addr_dec_top/icu8_inst

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

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

Register:	ACK			Address:	8008h	
Bits:	2dt0	Reset value:	0h	Attributes:	rht	

Description:	Interrupt Vector Register with IRQ Acknowledge Acknowledge the highest priority pending interrupt request by reading the associated interrupt vector For valid request: binary code of the input number Otherwise: Default vector 0h

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

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

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

Register:	UNLOCK_RD_ONLY_ACK			Address:	8018h	
Bits:	0	Reset value:	0h	Attributes:	r	w
Description:	Unlocks the read only acknowledge mechanism. By setting this bit to 1 one can reach an acknowledge via a write access to ACK 0: Read only mechanism is active to acknowledge an Interrupt Request 1: Also a write access to ACK acknowledges the Interrupt Request. Be aware that this write access is destructive.					

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

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

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

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

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

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

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

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

Register:	EDGEREG0			Address:	9400h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:	Edge Select Register Select the edge for the interrupt detection (only when edge detection has been set for the associated input) Interrupt inputs 0 – 7 0 = Interrupt detection for positive edge 1 = Interrupt detection for negative edge					

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

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

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

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

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

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

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

Register:	PRIOREG5			Address:	A014h	
Bits:	2dt0	Reset value:	7h	Attributes:	r	w
Description:	Priority Register for IRQ5, the higher the number, the lower the priority					

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

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

Module: /icu_ertec_addr_dec_top/icu_general_registers_inst

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

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

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

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

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

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

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

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

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

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.6 Host ICU (PerIF: Event Unit)

The Host Interrupt Controller is included in the Peripheral Interface.

2.3.7 GDMA (Direct Memory Access)

2.3.7.1 GDMA function description

The GDMA controller is programmable to perform DMA jobs. A DMA job is a collection of single transfers which are organized in a transfer list. The transfer list contains a parameterizable maximum number of single transfers. In the GDMA, the parameter is set to allow maximally 256 single transfers. A single DMA transfer represents copying of a programmable count of data elements, with programmable element size, from source address to destination address. The number of data elements can be programmed within the range of 0 to 65535, thus the count of data elements can be set within the range of 1 to 65536 elements. A transfer record of the transfer list defines these parameters of the single DMA transfer. Each transfer record holds information, whether it is the last one in the job. The principle of the DMA jobs organization is shown in Figure 10.

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

Figure 10: The principle of the DMA Job organization

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.7.1.1 AHB interfaces

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

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

2.3.7.1.2 Job priorities

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

2.3.7.1.3 Details

2.3.7.1.3.1 Blocks

The GDMA controller consists of the following blocks:

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

2.3.7.1.3.2 Interfaces

The GDMA controller contains the following interfaces:

- AHB Slave Interface – Serves for access to the GDMA registers as well as to the DMA RAM, if the GDMA controller is configured for an internal RAM.
- AHB Master Interface – Serves for transferring data via a DMA channel (read and write accesses). The AHB Master also is used to access the DMA RAM from the GDMA controller.
- Job-starting input signals – Start jobs by HW, they are connected to PROFINET-IP (3 × Application Timer), Timer and GPIO input signals.

Copyright © Siemens AG 2023. All rights reserved

Note

The start of a job by HW is enabled by means of bit HW_JOB_START_EN of the Job Control register.

- DMA request by HW signals – For a job dedicated to data transfer from a peripheral, the HW DMA request signal must inform, if data is available on the peripheral. For a job dedicated to data transfer to the peripheral, the HW DMA request signal must inform, if the peripheral is ready to receive the data.
- According to these needs, the HW DMA request signals of the GDMA controller are connected to the corresponding outputs of the peripheral devices. The number of DMA request signals is parameterizable and equals to the maximum number of jobs.

Note

The ready-checking of the peripheral is enabled by means of bit HW_FLOW_EN of the Job Control register. For every HW DMA request signal input an HW DMA acknowledge output exists, which is asserted at the end of a DMA transfer, if this is enabled in the DMA Transfer Record and the bit HW_FLOW_EN in the job control register is set.

- DMA Interrupt Request (DMA_IRQ) output – If a job is finished or a DMA error occurs (and if the other conditions, described later, are fulfilled), then the GDMA controller generates a DMA interrupt request.
- Error Detection and Correction (EDC) interrupt request – these signals (GDMA-1B: 1Bit Error corrected, GDMA-2B: 2Bit Error recognized) are wired to the 'EDC_EVENT' register in the SCRB Register Block (see chapter 2.3.14.11.13). Additionally, the interrupt EDC_Event (IRQ48) is set (see Table 15). To clear the 'EDC_EVENT' register a ‘0’ must be written.

  After Reset an initialisation of the EDC-Bits is performed automatically. The conclusion of the initialisation can be read in the SCRB register 'EDC_INIT_DONE' (see chapter 2.3.14.11.27.1). These GDMA outputs are internally connected to the outputs of the internal DMA RAM.

Note

This interrupt is used only when the GDMA controller is configured for an internal DMA RAM.

2.3.7.1.3.3 DMA Jobs programming

DMA jobs are programmed by means of GDMA Job Control registers and the transfer records of the DMA RAM, as illustrated in Figure 11.

Copyright © Siemens AG 2023. All rights reserved

Figure 11: DMA Jobs programming

Note

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

2.3.7.1.3.4 DMA Job Control registers

A Job Control register consists of the following fields:

- First Transfer Number of the Job (TRANSFER_PTR) is a pointer to the transfer number of the first transfer of the job. (The pointer is shown in Figure 11.)
- Job Priority (JOB_PRIO) defines the priority of the job. A higher value of JOB_PRIO means higher priority. If two or more jobs have the same value of JOB_PRIO, then the job priority is defined by the job number. In such a case, the smaller job number means the higher priority.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

If, for example, the JOB_PRIO field has the same value for each job, the highest priority has the started job with the smallest job number.

- HW Job Start Selector (HW_SELECT) defines, which of „n" HW job-starting signals is chosen to start the job by HW.
  Note that the maximum number of job-starting input signals is parameterizable up to 64. The HW parameter has no influence on the register width.
- Job reset (JOB_RESET) cancels the running job.
- Interrupt request generation enable (INTR_EN) enables GDMA controller to generate interrupt request when the job has finished.
- Hardware Triggered Flow Enable (HW_FLOW_EN) enables/disables the check whether the HW peripheral is ready.
- Start Job by HW enable (HW_JOB_START_EN).
- Enable Job (JOB_EN) enables the run of the Job or interrupts the running job.
- Start Job by SW (SW_JOB_START).

2.3.7.1.3.5 Transfer list

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

- "Source Address" defines the address of source data for the DMA transfer. This can be either a memory or a peripheral address.
- "Destination Address" defines the address of the destination for the DMA transfer. This can be either a memory or a peripheral address. The addresses of the GDMA registers and DMA RAM are protected against write accesses by the DMA transfers.
  The reason for this is as follows: When the GDMA controller is programmed incorrectly, the data in the GDMA registers and DMA RAM could be corrupted and subsequently the system might fail. When a wrong destination address is programmed, then the DMA destination address error is assumed. When at the same time bit "Error Interrupt Enable" (ERR_INT_EN) in the 'GDMA_MAIN_CTRL' register is "1", the relevant error bit in the GDMA Error Interrupt State register is set to "1" and the GDMA interrupt request (DMA_IRQ) is generated.
- "Transfer Control" consists of three 2-bit fields: Source Address Mode, Destination Address Mode and Burst Mode.
  - The Source Address Mode and Destination Address Mode can be either "increment" or "hold". **Note** that the second bit of this field serves as a reserve bit.
  - The Burst Mode holds the information about the AHB burst type which will be used, and can be Single, INCR4, INCR8 and INCR16
    (see necessary selection under HW_DMA_REQ signals in chapter 2.3.7.2.1.2).
- "Transfer Count" consists of 4 fields: 1-bit flag "Last Transfer of the Job", 1-bit flag "Enable DMA Acknowledge", 2-bit field "Element Size" and 16-bit field "Transfer Count".
  - Last Transfer of the Job – Indicates the last transfer record in a job (thus the transfers in a job are defined by pointer TRANSFER_PTR in the Job Control register and the bit "Last Transfer of the Job".) When no transfer in the Transfer List is marked as the last one, the transfers will be executed in an infinite loop.
  - Enable DMA Acknowledge – The output HW_DMA_ACK of a job is set to '1', if the input HW_DMA_REQ of this job is 1, and the bit HW_FLOW_EN of the job register is 1, and the current transfer is finished, and the bit DMA_ACK_EN in the Transfer Record is 1. The output HW_DMA_ACK of a job is reset to '0', if the input HW_DMA_REQ of this job is 0.
  - Element Size – Sets the size of elements to be copied in the transfer and can be 8, 16 or 32 bits. The number of bytes to tranfer is (element size) × (transfer count).

Copyright © Siemens AG 2023. All rights reserved

- Transfer Count – Holds the information on how many elements have to be copied in this transfer. It can be programmed from 0 to 65535 elements, representing the range of 1 to 65536 elements.

2.3.7.1.3.6 DMA Control registers

The global SW configuration of the GDMA controller is located in the GDMA Control registers. They are organized in a set of four 32-bit registers: “GDMA Registers Base Address”, “DMA Transfer List Base Address”, “GDMA Main Control” and “Enable Job Counter for the Job”.
The DMA Control registers configure the address space of the GDMA registers and the transfer list as well as control the functions of the GDMA controller, as described below:

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

2.3.7.1.3.7 Job control by SW or by HW

SW control of the job

- Start of job by SW:
  The job is started by SW when bit SW_JOB_START is set in the Job Control register. Start of DMA job is disabled, when bit SW_RESET in the 'GDMA_MAIN_CTRL' register is set to “1” or when bit HW_JOB_START_EN in the Job Control register is set to “1”.
- Make a job running:
  When one or more jobs are started, the started job with the highest priority begins to run (becomes active). The running job can be interrupted in three ways:
  1. When a job is running and a job with higher priority is started, the running job is interrupted and the job with higher priority begins to run. After the job with the higher priority is finished, the interrupted job will continue.
  2. When bit JOB_EN in the Job Control register is reset from ‘1’ to “0”, a running job is interrupted and a started job with the next, smaller priority begins to run. After this bit is

changed back to “1”, the interrupted job will be reactivated. JOB_EN = 0 does not cancel a job.

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

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

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

HW control of the job

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

The job control is illustrated in Figure 12.

Copyright © Siemens AG 2023. All rights reserved

Figure 12: Finite state machine diagram of the job control algorithm

2.3.7.1.3.8 GDMA Status registers

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

- Job State register (GDMA_JOB_STATUS) indicates via its 32 bits, which job is active (i.e., started, running or interrupted).
- Finished Jobs State register (GDMA_FINISHED_JOBS) contains one “job finished” bit for each job. This bit is set to “1” each time after the job is completed. Writing a “1” clears the bit.
- Actual Job State register (GDMA_ACTUAL_STATUS) contains 1-bit flag “Active Job Number Valid” (ACT_JOB_VAL) and 5-bit field “Active Job Number” (ACT_JOB). Active Job Number indicates the number of the running job. Bit “Active Job Number Valid” indicates whether the job number is a valid number, which represents active DMA transfers.
- GDMA Interrupt State register (GDMA_IRQ_STATUS) contains a 1-bit “job finished” bit for each job. This bit is set when a job is finished and a interrupt request (DMA_IRQ) is generated. Writing a “1” clears the bit.
- Error State register (GDMA_ERR_STATUS) contains 32 error bits, one bit for each type of error. The error bit is set to “1” when the corresponding type of error occurs and when the ERR_INT_EN bit in the 'GDMA_MAIN_CTRL' register is set to “1”. Writing a “1” clears the error bit. If an error bit is set to “1”, then the interrupt request (DMA_IRQ) is generated.

Copyright © Siemens AG 2023. All rights reserved

- Job counter register (GDMA_JOB_COUNT) counts clock cycles for a selected running job. Selection of the jobs for which the clocks shall be counted is done with 32-bit register Enable Job Counter for the Job (GDMA_JC_EN), one of the GDMA Control registers. Reset of the job counter is done by bit “Reset Job Counter” (JC_RESET) of the GDMA Main Control register (GDMA_MAIN_CTRL).

2.3.7.1.4 Usage

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

2.3.7.1.4.1 First step

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

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

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

2.3.7.1.4.2 Second step

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

- Source Address
- Destination Address
- Transfer Control
  - consists of three 2-bit fields: Source Address Mode, Destination Address Mode and Burst Mode, as described above.
- Transfer Count
  - consists of 4 fields: 1-bit flag “Last Transfer of the Job”, 1-bit flag “Enable DMA Acknowledge”, 2-bit field “Element Size” and 16-bit field “Transfer Count”, as described above.

2.3.7.1.4.3 Third step

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

Copyright © Siemens AG 2023. All rights reserved

2.3.7.1.5 Result

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

2.3.7.1.5.1 DMA job started by SW or by HW

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

Start of the job by SW

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

Start of the job by HW

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

2.3.7.1.5.2 DMA job controlled by SW or by HW

Control of the job by SW

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

- When bit DMA_EN in the 'GDMA_MAIN_CTRL' register is reset while a DMA job is running, the job is interrupted and no new job begins to run. When subsequently bit DMA_EN is set again, the interrupted job will continue.
- The started jobs can be reset and the running job as well as all interrupted ones can be cancelled when bit SW_RESET in 'GDMA_MAIN_CTRL' register is set.
- The running job can be interrupted when the job-enabling bit (JOB_EN) of the Job Control register is reset from ‘1’ to ‘0’.
- The running and all interrupted jobs can be cancelled, when the job-resetting bit (JOB_RESET) in the Job Control register is set.

Control of the job by HW

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

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

2.3.7.1.5.3 Job finished

When a DMA job is finished, the GDMA controller generates a DMA interrupt request. This interrupt request generation can be enabled/disabled for each job, using bit INTR_EN of the Job

Copyright © Siemens AG 2023. All rights reserved

Control registers. When the DMA interrupt is generated, a “job finished” bit in the Interrupt State register is set.

2.3.7.1.5.4 Monitoring of GDMA controller status

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

2.3.7.1.6 Memory

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

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

The internal GDMA RAM is accessible from the CPU through the AHB Slave interface of the GDMA only with word accesses (byte and halfword write accesses not possible). From the GDMA controller it is accessible through its AHB Master interface.

Note

The data path from the GDMA controller core to the GDMA RAM is: GDMA controller core -> AHB Master of the GDMA -> Multi-layer AHB Bus -> AHB Slave of the GDMA -> RAM Wrapper -> RAM. Thus, the access algorithm is the same for both GDMA RAM configurations. The address space of the internal GDMA RAM is also configurable by means of GDMA Control registers (LIST_ADDR and LIST_SIZE).

The GDMA RAM has the size of ((n+1) × 16 Byte) + ((j+1) × 16 Byte), where “n” is the index of the last transfer from the transfer list and “j” is the index of the last job stack. The configuration for ERTEC 200P is n = 255 and j = 31. **For this configuration the internal GDMA SRAM is organized in 1152 × 32 Bit**.

The GDMA RAM has additionally EDC bits (7 bit for a 32-bit word, 1-bit error, correctable; 2-bit error, detectable). If an EDC error is detected:

- in the SCRB register 'EDC_EVENT' the appropriate reason is stored (GDMA-1B: 1-bit error corrected or GDMA-2B: 2-bit error detected) and
- the interrupt EDC_Event (IRQ48) is generated (see chapter 2.3.5.9). The 'EDC_EVENT' register is reset by writing the register with '0h'.

2.3.7.1.7 Interrupts

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

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

The two interrupt requests are:

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

- Job is finished and bit “Interrupt Request Generation Enable” (INTR_EN) in the Job Control register is set to “1”. When Interrupt request is generated, a “job finished” bit is set in the Interrupt State register (GDMA_IRQ_STATUS). This status register must be read by the interrupt controller to figure out which job has caused the interrupt. The register is cleared by writing a “1” to the related bit position.

Copyright © Siemens AG 2023. All rights reserved

- A monitored error occurs at the DMA controller and bit “Error Interrupt Enable” (ERR_INT_EN) in the 'GDMA_MAIN_CTRL' register is set to “1”. In this case the relevant error bit in the GDMA Error Interrupt State register is set to “1”. This status register must be read by the interrupt controller to figure out which type of error has caused the interrupt. The register is cleared by writing a “1” to the related bit position.

The monitored types of error are:

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

2.3.7.2 ERTEC 200P GDMA use cases

2.3.7.2.1 Using the Job Start and HW_REQ Signal List

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

2.3.7.2.1.1 HW Trigger with HW_JOB_START input signals (edge-sensitive)

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

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

Table 17: HW trigger for GDMA

Port	Source	Active level	Description
HW_JOB_START (starts the job in HW by rising edge):			
HW_JOB_START_0	**PNPLL_OUT11 (PN-IP)**	high	coming from PNPLL in PN-IP (see Table 26)
HW_JOB_START_1	**PNPLL_OUT12 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_2	**PNPLL_OUT13 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_3	**PNPLL_OUT14 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_4	**PNPLL_OUT15 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_5	**PNPLL_OUT16 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_6	**PNPLL_OUT17 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_7	**PNPLL_OUT18 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_8	**PNPLL_OUT19 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_9	**PNPLL_OUT20 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_10	**TIM_OUT0 (Timer Unit)**	high	coming from Timer 0 in the Timer Unit
HW_JOB_START_11	**TIM_OUT1 (Timer Unit)**	high	coming from Timer 1 in the Timer Unit

Copyright © Siemens AG 2023. All rights reserved

Port	Source	Active level	Description
HW_JOB_START_12	**TIM_OUT2 (Timer Unit)**	high	coming from Timer2 in the Timer Unit
HW_JOB_START_13	**TIM_OUT3 (Timer Unit)**	high	coming from Timer 3 in the Timer Unit
HW_JOB_START_14	**TIM_OUT4 (Timer Unit)**	high	coming from Timer 4 in the Timer Unit
HW_JOB_START_15	**TIM_OUT5 (Timer Unit)**	high	coming from Timer 5 in the Timer Unit
HW_JOB_START_16	**GPIO0-INT (GPIO31-0)**	high	coming from GPIO0 input pin (see chapter 2.3.14.10), in I-Filter this input signal needs Mode 2 “Synchronize”, because GDMA has no synchronized input behavior
HW_JOB_START_17	**GPIO1-INT (GPIO31-0)**	high	coming from GPIO1 input pin, in I-Filter this input signal needs Mode 2 “Synchronize”, because GDMA has no synchronized input behavior
HW_JOB_START_18	**GPIO2-INT (GPIO31-0)**	high	coming from GPIO2 input pin, in I-Filter this input signal needs Mode 2 “Synchronize”, because GDMA has no synchronized input behavior
HW_JOB_START_19	**GPIO3-INT (GPIO31-0)**	high	coming from GPIO3 input pin, in I-Filter this input signal needs Mode 2 “Synchronize”, because GDMA has no synchronized input behavior
HW_JOB_START_20	**RSC_ENC_FINISHED**	high	coming from RSC1
HW_JOB_START_21	**RSC_DEC_FINISHED_GOOD**	high	
HW_JOB_START_22	**RSC_DEC_FINISHED_CORR**	high	
HW_JOB_START_23	**RSC_DEC_FINISHED_BAD**	high	
HW_JOB_START_24	**RSC_ENC_FINISHED**	high	coming from RSC2
HW_JOB_START_25	**RSC_DEC_FINISHED_GOOD**	high	
HW_JOB_START_26	**RSC_DEC_FINISHED_CORR**	high	
HW_JOB_START_27	**RSC_DEC_FINISHED_BAD**	high	
HW_JOB_START_28	**RSC_DEC_FINISHED_GOOD or RSC_DEC_FINISHED_CORR or RSC_DEC_FINISHED_BAD**	high	coming from RSC1
HW_JOB_START_29	**RSC_DEC_FINISHED_GOOD or RSC_DEC_FINISHED_CORR or RSC_DEC_FINISHED_BAD**	high	coming from RSC2
HW_JOB_START_ 30..63	Tied to inactive values		Not used for ERTEC 200P

Not all HW_JOB_START signals are used in the ERTEC 200P. The input signals for the HW_JOB_START inputs that are not used are set to inactive level. The GPIO inputs for GDMA

(GPIO3...GPIO0) must be synchronized in the I-Filter with the system clock and can be inverted/not inverted with GPIO_POLSEL (see chapter 2.3.14.11.11).

2.3.7.2.1.2 HW Flow Control with REQ and DMA_ACK handshake (level-sensitive)

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

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

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

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

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

In this mode, the programming rules for the GDMA and the connected IP (SPI, UART) must be followed. The GDMA IP connection only works if these programming rules are followed.

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

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

The following table shows the use of request signals in the ERTEC 200P.

Table 18: HW flow control for GDMA

HW_DMA_REQ (controls DMA transfers to / from serial peripherals):			
HW_DMA_REQ_0 **(JOB0)**	UART1_TX-FIFO half full or less	high	UART1 Tx-FIFO not half full (set GDMA to INCR8 mode)
HW_DMA_REQ_1 **(JOB1)**	UART1_RX-FIFO not empty	high	UART1 Rx-FIFO not empty (set GDMA to single mode)
HW_DMA_REQ_2 **(JOB2)**	UART2_TX-FIFO half full or less	high	UART2 Tx-FIFO not half full (set GDMA to INCR8 mode)
HW_DMA_REQ_3 **(JOB3)**	UART2_RX-FIFO not empty	high	UART2 Rx-FIFO not empty (set GDMA to single mode)
HW_DMA_REQ_4 **(JOB4)**	UART3_TX-FIFO half full or less	high	UART3 Tx-FIFO not half full (set GDMA to INCR8 mode)
HW_DMA_REQ_5 **(JOB5)**	UART3_RX-FIFO not empty	high	UART3 Rx-FIFO not empty (set GDMA to single mode)
HW_DMA_REQ_6 **(JOB6)**	UART4_TX-FIFO half full or less	high	UART4 Tx-FIFO not half full (set GDMA to INCR8 mode)
HW_DMA_REQ_7 **(JOB7)**	UART4_RX-FIFO not empty	high	UART4 Rx-FIFO not empty (set GDMA to single mode)
HW_DMA_REQ_8 **(JOB8)**	SPI1_SSPRXDMA	high	SPI1 Rx-FIFO not empty - DMA request (set GDMA to single mode)
HW_DMA_REQ_9 **(JOB9)**	SPI1_SSPTXINTR	high	SPI1 Tx-FIFO not half full - DMA request (the SSPTXINTR interrupt (Transmit FIFO is half

Copyright © Siemens AG 2023. All rights reserved

HW_DMA_REQ (controls DMA transfers to / from serial peripherals):			
			full or less) must be enabled; set GDMA to INCR4 mode)
HW_DMA_REQ_10 **(JOB10)**	SPI1_SSPTX_ Delayed_Request	high	SPI1 transmit delayed – DMA request (mapped in the GDMA Shell) Expiry of timer 4 when 'SPI1 Tx-FIFO not half full' sets this bit; set GDMA to single byte transfer. DMA_ACK is used, resets this bit.
HW_DMA_REQ_11 **(JOB11)**	SPI2_SSPRXDMA	high	SPI2 Rx-FIFO not empty - DMA request (set GDMA to single mode)
HW_DMA_REQ_12 **(JOB12)**	SPI2_SSPTXINTR	high	SPI2 Tx-FIFO not half full - DMA request (the SSPTXINTR interrupt (Transmit FIFO is half full or less) must be enabled; set GDMA to INCR4 mode)
HW_DMA_REQ_13 **(JOB13)**	SPI2_SSPTX_ Delayed_Request	high	SPI2 transmit delayed – DMA request (mapped in the GDMA Shell) Expiry of timer 5 when 'SPI2 Tx-FIFO not half full' sets this bit; set GDMA to single byte transfer. DMA_ACK is used, resets this bit.
HW_DMA_REQ_14 **(JOB14)**	Not used for ERTEC 200P; tied to inactive values		
HW_DMA_REQ_15 **(JOB15)**	Not used for ERTEC 200P; tied to inactive values		
HW_DMA_REQ_16 **(JOB16)**			
HW_DMA_REQ_17 **(JOB17)**			
HW_DMA_REQ_18 **(JOB18)**	OctalSPI_BREQ_TX (dma_tx_req)	high	Transmit FIFO DMA Request - Asserted when the transmit FIFO requires service from the DMA Controller; that is, the transmit FIFO is at or below the watermark level. DMA_ACK is not used, ACK for Request (dma_tx_ack)
HW_DMA_REQ_19 **(JOB19)**	OctalSPI_SREQ_TX (dma_tx_single)	high	DMA Transmit FIFO Single Signal - This DMA status output informs the DMA Controller that there is at least one free entry in the transmit FIFO. DMA_ACK is not used, ACK for Request (dma_tx_ack)
HW_DMA_REQ_20 **(JOB20)**	OctalSPI_BREQ_RX (dma_rx_req)	high	Receive FIFO DMA Request - Asserted when the receive FIFO requires service from the DMA Controller; that is, the receive FIFO is at or above the watermark level. DMA_ACK is not used, ACK for Request (dma_rx_ack)
HW_DMA_REQ_21 **(JOB21)**	OctalSPI_SREQ_RX (dma_rx_single)	high	DMA Receive FIFO Single Signal - This DMA status output informs the DMA Controller that

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

HW_DMA_REQ (controls DMA transfers to / from serial peripherals):			
			there is at least one valid data entry in the receive FIFO. DMA_ACK is not used, ACK for Request (dma_rx_ack)
HW_DMA_REQ_22…_23 **(JOB22…JOB23)**	Not used for ERTEC 200P Step 3; tied to '0'		
HW_DMA_REQ_24 **(JOB24)**	*not* JOB24 suspended	high	*not* SCRB.GDMA_CTRL.JOB_SUSPEND(0), see chapter 2.3.14.11.26.
HW_DMA_REQ_25 **(JOB25)**	*not* JOB25 suspended	high	*not* SCRB.GDMA_CTRL.JOB_SUSPEND (1), see chapter 2.3.14.11.26.
HW_DMA_REQ_26 **(JOB26)**	*not* JOB26 suspended	high	*not* SCRB.GDMA_CTRL.JOB_SUSPEND (2), see chapter 2.3.14.11.26.
HW_DMA_REQ_27 **(JOB27)**	*not* JOB27 suspended	high	*not* SCRB.GDMA_CTRL.JOB_SUSPEND (3), see chapter 2.3.14.11.26.
HW_DMA_REQ_28 **(JOB28)**	*not* JOB28 suspended	high	*not* SCRB.GDMA_CTRL.JOB_SUSPEND (4), see chapter 2.3.14.11.26.
HW_DMA_REQ_29 **(JOB29)**	*not* JOB29 suspended	high	*not* SCRB.GDMA_CTRL.JOB_SUSPEND (5), see chapter 2.3.14.11.26.
HW_DMA_REQ_30 **(JOB30)**	*not* JOB30 suspended	high	*not* SCRB.GDMA_CTRL.JOB_SUSPEND (6), see chapter 2.3.14.11.26**.**
HW_DMA_REQ_31 **(JOB31)**	*not* JOB31 suspended	high	*not* SCRB.GDMA_CTRL.JOB_SUSPEND (7), see chapter 2.3.14.11.26**.**

2.3.7.2.2 DMA with SPI interface

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

2.3.7.2.2.1 Receive data

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

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

Copyright © Siemens AG 2023. All rights reserved

Configuring the JOB_CTRL register and Transfer_Record

Receive data can have different lengths from **2 to 256 bytes**. DMA traffic is controlled with SPI DMA-REQ. The data is read out by the receive FIFO of the SPI and written to the PB-RAM, for example. The data bus width of the SPI is 16 bits. JOB_CTRL register 0 is linked to DMA_ACK0, which is connected to SPI_RFNE (Receive FIFO not empty).

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

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

TRANSFER_PTR =	0-255	First Transfer Number of job in Transfer List (0-255)
JOB_PRIO =	0-31	Job Priority(0-31)
HW_SELECT =	0	HW Job Start Selector (selects one of 64 inputs)
JOB_RESET =	0	Reset job (cancel running job): 0 = don't care, 1 = cancel
INTR_EN =	0	Interrupt Request Generation Enable: 0 = disabled
HW_FLOW_EN =	1	HW-triggered Flow enabled
HW_JOB_START_EN =	0	Start job by hardware: 0 = disabled
JOB_EN =	1	Job Enable
SW_JOB_START =	1	Start job from software enabled

Parameter assignment of DMA Transfer_Record for SPI operation (receive data)

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

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

Only one transfer record is likely to be needed.

2.3.7.2.2.2 Transmit data

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

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

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

Copyright © Siemens AG 2023. All rights reserved

Configuring the JOB_CTRL register and Transfer_Record

Send data can have different lengths from **2 to 256 bytes**. DMA traffic is controlled with SPI DMA-REQ. Data is read by the DMA from PN-RAM, for example, and written to the transmit FIFO of the SPI. The data bus width of the SPI is 16 bits. JOB_CTRL register 1 is linked to DMA_ACK1, which is connected to SPI_TFE (Transmit Buffer empty).

Entries required in Job Control register 1 (SPI transmit data)

TRANSFER_PTR =	0-255	First Transfer Number of job in Transfer List (0-255)
JOB_PRIO =	0-31	Job Priority (0-31)
HW_SELECT =	0	HW Job Start Selector (selects one of 64 inputs)
JOB_RESET =	0	Reset job (cancel running job) 0 = don't care, 1 = cancel
INTR_EN =	0	Interrupt Request Generation Enable: 0 = disabled
HW_FLOW_EN =	1	HW-triggered Flow enabled
HW_JOB_START_EN =	0	Start job by hardware disabled
JOB_EN =	1	Job Enable
SW_JOB_START =	1	Start job from software enabled

Parameter assignment of DMA Transfer_Record for SPI operation (send data)

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

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

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

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

Figure 13: GDMA interrupt in line with the user data to be transferred

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

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

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

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

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

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

  - when the number of data transfers is even, the GDMA interrupt always will be active 3 SPI transfers (8…16 bits) before the end of the SPI transfer

    → 1 SPI transfer in the SPI shift register + 2 SPI transfers in the SPI Transmit FIFO:

  - when the number of data transfers is odd, the GDMA interrupt always will be active 2 SPI transfers (8…16 bits) before the end of the SPI transfer

    → 1 SPI transfer in the SPI shift register + 1 SPI transfer in the SPI Transmit FIFO:

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

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

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

2.3.7.2.3 DMA with UART interface

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

- UART_RX-FIFO not empty:
  The RX FIFO contains at least one character and is not empty. As the UART cannot know when the last character (byte) was read in, the GDMA must operate in SINGLE-byte access

Copyright © Siemens AG 2023. All rights reserved

mode (AHB). 1 to 65536 characters can therefore be transferred per DMA request; the job consists of a transfer entry.

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

Note

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

2.3.7.2.4 DMA with Octal-SPI

The Octal-SPI controller provides DMA-Handshaking signals for HW-Flow-Control read/write transfer from/to external memories that are connected to the Octal-SPI controller.

In ERTEC 200P, there is a fixed mapping of job numbers (see chapter 2.3.7.2.1.2) to read/write transfers, which is:

- JOB 18 and JOB 19 only for write transfers
- JOB 20 and JOB 21 only for read transfers.

Note

Please **note** that the ERTEC 200P Step 3 **only supports Quad-SPI mode**.

The following two chapters describe how to configure the GDMA for Quad-SPI transfers.

2.3.7.2.4.1 Receive data

For reading a page, JOB 20/21 first, after being triggered by software, executes the read command (i.e., writing the command and the read address which are stored, for instance, in the data RAM) into the Octal-SPI controller, then JOB 20/21 reads out the read data from the Octal-SPI controller and writes the data into the RS controller. With the last write access to the RS controller, the Octal-SPI controller automatically starts the decoding process within the RS controller.

By polling the *JOB_Finished* register, the software detects the end of JOB 20/21. Then, the software polls DEC_Status.run within the RS controller for detecting the end of the decoding process. After that, the software can either copy the decoded data to, for instance, the CodeRAM, or can alternatively start a dedicated GDMA job for that task.

If further pages need to be read, after the end of the copy process the software needs to increment the read address in, e.g., the data RAM which is read by JOB 20/21 and passed to the Octal-SPI controller, and finally re-trigger JOB 20/21.

Necessary values for the JOB20/21_Control register

TRANSFER_PTR =	0-255	Pointer to the first Transfer record
JOB_PRIO =	0-31	Some priority, jobs are executed in a sequential manner

HW_SELECT =	0	HW Job Start Selector (selects one of 64 inputs)
JOB_RESET =	0	Do not reset the job
INTR_EN =	0	Deactivate interrupts
HW_FLOW_EN =	0	Deactivate HW-Flow-Control initially
HW_JOB_START_EN =	0	Job will be triggered by software
JOB_EN =	1	Enable the job
SW_JOB_START =	1	Trigger the job by software

JOB 20/21 consists of 7 Transfer records:

1. The GDMA writes the read command and address into the Octal-SPI controller with word access.
2. A word write access on the **JOB20/21_Control** register HW_FLOW_EN = 1 now activates HW-Flow-Control.
3. **Note**:
   The GDMA writes the register value to an unused memory region. Due to an internal interruption delay between activating HW-Flow-Control and the actual interruption, the GDMA would immediately read data from the still empty RX FIFO inside the Octal-SPI controller and would thus cause an RX underflow. For preventing that RX underflow, it is required to parameterize a Transfer record that reads back the written value from JOB20/21_Control register and writes it to an unsued memory destination, e.g., a service register. With this delay, the internal interruption delay is covered and JOB20/21 is interrupted in time, i.e., before any read access to an empty FIFO is executed. As soon as the first read data is available in the RX FIFO inside the Octal-SPI controller, the signals dma_rx_req and dma_rx_single will become active and consequently JOB 20/21 is continued with next Transfer record.
4. The next Transfer record reads 1 word less from Octal-SPI controller DR register.
5. A word write access on the **JOB20/21_Control** register with *HW_FLOW_EN* = 0 now deactivates HW-Flow-Control.
6. Readback the register value to an unused memory region. Without doing this, the GMA will not set *JOB_Finished* register and remain in an endless loop.
7. Last Transfer record reads the missing Data, e.g., 1 word.

Note

Hint for parameterization:

- *TransferRecord.ESIZE* must match the Octal-SPI controller FrameSize (OCTAL_SPI.CTRLR0.DFS_32).
- *TransferRecord.BURST_MODE* = 0x0 (Single Access) is mandatory.

2.3.7.2.4.2 Transmit data

JOB 18/19 first executes the write command (i.e., writes command and write address) within the Octal-SPI controller and then writes the write data into the Octal-SPI controller.
If further pages need to be read, after the end of the write process the software needs to increment the write address in, e.g., the data RAM which is read by JOB 18/19 and passed to the Octal-SPI controller, and finally re-trigger JOB 18/19.

Necessary values for the JOB18/19_Control register

TRANSFER_PTR =	0-255	Pointer to the first Transfer record
JOB_PRIO =	0-31	Some priority, jobs are executed in a sequential manner
HW_SELECT =	0	HW Job Start Selector (selects one of 64 inputs)
JOB_RESET =	0	Do not reset the Job
INTR_EN =	0	Deactivate interrupts
HW_FLOW_EN =	0	Activate HW-Flow-Control
HW_JOB_START_EN =	0	Job will be triggered by software

Copyright © Siemens AG 2023. All rights reserved

JOB_EN =	1	Enable the job
SW_JOB_START =	1	Trigger the job by software

JOB 18/19 consists of 3 Transfer records:

1. The GDMA writes the write command into the Octal-SPI controller with one byte access.
2. The GDMA writes the write address into the Octal-SPI controller with word access.
3. Standard-DMA: This Transfer record writes the actual write data into the Octal-SPI controller. One Transfer record without CS Monitoring, two Transfer records with CS Monitoring, where the last Transfer record with 1 word is a normal access (see chapter 2.3.14.8.2).

Note

Hint for parameterization:

- *TransferRecord.ESIZE* must match the Octal-SPI controller FrameSize (*OCTAL_SPI.CTRLR0.DFS_32*).
- *TransferRecord.BURST_MODE* = 0x0 (Single Access) is mandatory.
- Watermark in the Octal-SPI controller (*DMATDLR.DMATDL*) must be set to 0x4.

JOB 18 corresponds with the DMA-Handshaking signal **dma_tx_req**. The Octal-SPI controller activates dma_tx_req as soon as the fill level of the TX FIFO is equal or lower than the parameterized watermark (= DMATDLR.DMATDL). If the fill level is lower than the watermark, the Octal-SPI controller activates dma_tx_req.
JOB 19 corresponds with the DMA-Handshaking signal **dma_tx_single**. The Octal-SPI controller activates dma_tx_single, if there is at least on free entry in the TX FIFO. If the TX FIFO is full, the Octal-SPI controller deactivates dma_tx_single.

2.3.7.3 GDMA-IP bugs

JOB_Reset

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

Error description: JOB31 is to be canceled and restarted

1. JOB31 is disabled with:

   a) JOB_EN = 0 (bit 1 in GDMA_JOB31_CTRL) OR

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

Workaround:

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

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

4. The JOB is enabled (JOB_EN = 1), but not started.

Errors in JOB Stop with GDMA_MAIN_CTRL.DMA_EN = 0

Conditions:

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

If you stop processing globally with GDMA_MAIN_CTRL.DMA_EN = 0 (GDMA_JOB0 _CTRL.JOB_EN = 1), GDMA_JOB_COUNT continues to be incremented cyclically and GDMA_ACTUAL_STATUS = 0x00000000 (no job running, actual job number 0) is set.

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

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

Workaround:

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

2.3.7.4 Address mapping

Start-Address	End-Address	Modul/Memory-Name
B0h	10AFh	LIST_RAM
10B0h	12AFh	JOB_STACK_RAM
0h	12AFh	gdma

Module	Register/Memory	Read	Write	Access	Address
/gdma				32-Bit	
	/GDMA_REG_ADDR	r	(w)	32-Bit	0h
	/GDMA_LIST_ADDR	r	(w)	32-Bit	4h
	/GDMA_MAIN_CTRL	r	(w)	32-Bit	8h
	/GDMA_JC_EN	r	w	32-Bit	Ch
	/GDMA_JOB0_CTRL	(r)	(w)(t)	32-Bit	10h
	/GDMA_JOB1_CTRL	(r)	(w)(t)	32-Bit	14h
	/GDMA_JOB2_CTRL	(r)	(w)(t)	32-Bit	18h
	/GDMA_JOB3_CTRL	(r)	(w)(t)	32-Bit	1Ch
	/GDMA_JOB4_CTRL	(r)	(w)(t)	32-Bit	20h
	/GDMA_JOB5_CTRL	(r)	(w)(t)	32-Bit	24h
	/GDMA_JOB6_CTRL	(r)	(w)(t)	32-Bit	28h
	/GDMA_JOB7_CTRL	(r)	(w)(t)	32-Bit	2Ch
	/GDMA_JOB8_CTRL	(r)	(w)(t)	32-Bit	30h
	/GDMA_JOB9_CTRL	(r)	(w)(t)	32-Bit	34h
	/GDMA_JOB10_CTRL	(r)	(w)(t)	32-Bit	38h

Copyright © Siemens AG 2023. All rights reserved

	/GDMA_JOB11_CTRL	(r)	(w)(t)	32-Bit	3Ch
	/GDMA_JOB12_CTRL	(r)	(w)(t)	32-Bit	40h
	/GDMA_JOB13_CTRL	(r)	(w)(t)	32-Bit	44h
	/GDMA_JOB14_CTRL	(r)	(w)(t)	32-Bit	48h
	/GDMA_JOB15_CTRL	(r)	(w)(t)	32-Bit	4Ch
	/GDMA_JOB16_CTRL	(r)	(w)(t)	32-Bit	50h
	/GDMA_JOB17_CTRL	(r)	(w)(t)	32-Bit	54h
	/GDMA_JOB18_CTRL	(r)	(w)(t)	32-Bit	58h
	/GDMA_JOB19_CTRL	(r)	(w)(t)	32-Bit	5Ch
	/GDMA_JOB20_CTRL	(r)	(w)(t)	32-Bit	60h
	/GDMA_JOB21_CTRL	(r)	(w)(t)	32-Bit	64h
	/GDMA_JOB22_CTRL	(r)	(w)(t)	32-Bit	68h
	/GDMA_JOB23_CTRL	(r)	(w)(t)	32-Bit	6Ch
	/GDMA_JOB24_CTRL	(r)	(w)(t)	32-Bit	70h
	/GDMA_JOB25_CTRL	(r)	(w)(t)	32-Bit	74h
	/GDMA_JOB26_CTRL	(r)	(w)(t)	32-Bit	78h
	/GDMA_JOB27_CTRL	(r)	(w)(t)	32-Bit	7Ch
	/GDMA_JOB28_CTRL	(r)	(w)(t)	32-Bit	80h
	/GDMA_JOB29_CTRL	(r)	(w)(t)	32-Bit	84h
	/GDMA_JOB30_CTRL	(r)	(w)(t)	32-Bit	88h
	/GDMA_JOB31_CTRL	(r)	(w)(t)	32-Bit	8Ch
	/GDMA_JOB_STATUS	rh		32-Bit	90h
	/GDMA_FINISHED_JOBS	rh	w	32-Bit	94h
	/GDMA_ACTUAL_STATUS	r(h)		32-Bit	98h
	/GDMA_IRQ_STATUS	rh	w	32-Bit	9Ch
	/GDMA_ERR_IRQ_STATUS	r(h)	(w)	32-Bit	A0h
	/GDMA_JOB_COUNT	rh		32-Bit	A4h
	/REVISION_CODE	r		32-Bit	A8h
	LIST_RAM	r	w	32-Bit	B0h-10AFh
	JOB_STACK_RAM	r	w	32-Bit	10B0h-12AFh

Copyright © Siemens AG 2023. All rights reserved

2.3.7.5 Register description

Module: gdma

Register:	GDMA_REG_ADDR					Address:	0h	
Bits:	31dt0		Reset value:		0h	Attributes:	r	(w)
Description:	GDMA Register Base Address							
Bit	Identifier		Reset	Attr.		Function / Description		
3dt0	reserved		0h					
31dt4	REG_ADDR		0000000h	r	w	DMA Register Base Address		

Register:	GDMA_LIST_ADDR					Address:	4h	
Bits:	31dt0		Reset value:		0h	Attributes:	r	(w)
Description:	DMA Transfer List Base Address							
Bit	Identifier		Reset	Attr.		Function / Description		
3dt0	reserved		0h					
31dt4	LIST_ADDR		000h	r	w	DMA Transfer List Base Address		

Register:	GDMA_MAIN_CTRL					Address:	8h	
Bits:	31dt0		Reset value:		0h	Attributes:	r	(w)
Description:	Main Control register							
Bit	Identifier		Reset	Attr.		Function / Description		
0dt0	DMA_EN		0h	r	w	Global Enable		
1dt1	SW_RESET		0h	r	w	Software Reset '0' = don't care, '1' = Reset		
14dt2	reserved		0000h					
15dt15	JC_RESET		0h	r	w	Reset Job Counter '0' = don't care, '1' = Reset		
16dt16	ERR_INT_EN		0h	r	w	Error Interrupt Enable		
23dt17	reserved		00h					
31dt24	LIST_SIZE		000h	r	w	Total Number of Transfers in Transfer List (0-255) 0 means 1, 255-- 256		

Register:	GDMA_JC_EN					Address:	Ch	
Bits:	31dt0		Reset value:		0h	Attributes:	r	w
Description:	Enable Job Counter for Job							
Bit	Identifier		Reset	Attr.		Function / Description		
31dt0	GDMA_JC_EN		00000000h	r	w	Enable Job Counter for Job		

Register:	GDMA_JOB0_CTRL					Address:	10h	
Bits:	31dt0		Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 0							
Bit	Identifier		Reset	Attr.		Function / Description		
0dt0	SW_JOB_START		0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN		0h	r	w	Job Enable 0 = disable, 1 = enable		

Copyright © Siemens AG 2023. All rights reserved

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

Register:	GDMA_JOB1_CTRL				Address:	14h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 1						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	reserved	0h					
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	reserved	0h					
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	reserved	0h					
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB2_CTRL				Address:	18h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 2						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

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

Register:	GDMA_JOB4_CTRL				Address:	20h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 4						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

Register:	GDMA_JOB5_CTRL				Address:	24h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 5						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	reserved	0h					
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	reserved	0h					
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	reserved	0h					
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB6_CTRL				Address:	28h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 6						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

Register:	GDMA_JOB7_CTRL				Address:	2Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 7						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	reserved	0h					
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	reserved	0h					
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	reserved	0h					
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB8_CTRL				Address:	30h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 8						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

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

Register:	GDMA_JOB10_CTRL				Address:	38h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 10						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

Register:	GDMA_JOB11_CTRL				Address:	3Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 11						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	reserved	0h					
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	reserved	0h					
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	reserved	0h					
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB12_CTRL				Address:	40h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 12						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

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

Register:	GDMA_JOB14_CTRL				Address:	48h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 14						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

Register:	GDMA_JOB15_CTRL				Address:	4Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 15						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	reserved	0h					
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	reserved	0h					
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	reserved	0h					
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB16_CTRL				Address:	50h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 16						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

Register:	GDMA_JOB17_CTRL				Address:	54h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 17						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	reserved	0h					
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	reserved	0h					
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	reserved	0h					
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB18_CTRL				Address:	58h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 18						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

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

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

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

Register:	GDMA_JOB21_CTRL				Address:	64h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 21						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	reserved	0h					
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	reserved	0h					
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	reserved	0h					
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB22_CTRL				Address:	68h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 22						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

Register:	GDMA_JOB23_CTRL				Address:	6Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 23						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	reserved	0h					
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	reserved	0h					
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	reserved	0h					
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB24_CTRL				Address:	70h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 24						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

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

Register:	GDMA_JOB26_CTRL				Address:	78h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 26						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

Register:	GDMA_JOB27_CTRL				Address:	7Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 27						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	reserved	0h					
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	reserved	0h					
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	reserved	0h					
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB28_CTRL				Address:	80h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 28						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

Register:	GDMA_JOB29_CTRL				Address:	84h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 29						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	reserved	0h					
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	reserved	0h					
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	reserved	0h					
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB30_CTRL				Address:	88h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 30						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		wt	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		

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

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

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

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

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

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

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

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

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

Register:	GDMA_JOB_COUNT			Address:	A4h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:	Counts clock cycles, when selected Jobs are active. Selection of the counted Jobs is done through the GDMA_JC_EN register. Each Job has own enable bit here. The reset of Job Counter is done through “Reset Job Counter” flag of the GDMA_MAIN_CTRL register.					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	GDMA_JOB_COUNT	00000000h	rh		Counts clock cycles, when selected Jobs are active. Selection of the counted Jobs is done through the GDMA_JC_EN register. Each Job has own enable bit here. The reset of Job Counter is done through “Reset Job Counter” flag of the GDMA_MAIN_CTRL register.	

Register:	REVISION_CODE			Address:	A8h	
Bits:	31dt0	Reset value:	101h	Attributes:	r	
Description:	Revison Code register					
Bit	Identifier	Reset	Attr.		Function / Description	
7dt0	MINOR_REVISION	1h	r		minor revision = 0x1	
15dt8	MAJOR_REVISION	1h	r		major revision = 0x1	
31dt16	reserved	0000h				

Memory:	LIST_RAM			Start-address:	B0h	
Bits:	31dt0	Size:	1000h	Attributes:	r	w
Description:						

Memory:	JOB_STACK_RAM			Start-address:	10B0h	
Bits:	31dt0	Size:	200h	Attributes:	r	w
Description:						

2.3.7.6 DMA RAM

-- stack begins after last Transfer List entry:

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

Figure 14: DMA RAM address space

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

DMA Transfer Record – Source Address SRC_ADDR

Bit no.	Name	Description
31:0	SRC	Source Address

DMA Transfer Record – Destination Address DEST_ADDR

Bit no.	Name	Description
31:0	DEST	Destination Address

DMA Transfer Record – Transfer Control

Bit no.	Name	Description
31:24		reserved
23:22	SRC_AMODE	Source Address Mode 00 = increment address 10 = hold address x1 = reserved for future implementation = hold address
21:20	DEST_AMODE	Destination Address Mode (reserved = hold address) 00 = increment address 10 = hold address x1 = reserved for future implementation = hold address
19:18	BURST_MODE	00 = Single 01 = INCR4 10 = INCR8 11 = INCR16
17:16		reserved

DMA Transfer Record – Transfer Count

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

DMA Stack – Source Address Stack

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

DMA Stack – Destination Address Stack

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

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

DMA Stack – Transfer Control Stack

Bit no.	Name	Description
31:24		reserved
23:22	SRC_AMODE	Source Address Mode 00 = increment address 10 = hold address x1 = reserved for future implementation = hold address
21:20	DEST_AMODE	Destination Address Mode 00 = increment address 10 = hold address x1 = reserved for future implementation = hold address
19:18	BURST_MODE	00 = Single 01 = INCR4 10 = INCR8 11 = INCR16
15:8		reserved
7:0	CTRANSFER	Current Transfer List Number to continue from after interruption

DMA Stack – Transfer Count Stack

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

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.8 EMC (External Memory Controller)

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

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

Note

Due to its unavailability on the market, Burst Mode Flash ROM is no longer supported.

The EMC interface is connected to the system bus using an AHB slave interface with the following features:

- Supports AHB 2.0 lite protocol (No SPLIT, no RETRY)
- Burst transactions are accepted at the AHB input side.
  - If the SDRAM controller is active, bursts of undefined length are split into bursts of length 16.
  - If the asynchronous controller is active, any burst is split into single transfers (exception to this rule: read access to Page Mode ROM allows 16 beat bursts).
- The EMC itself can be configured using the AHB slave interface.
- SDRAM controller features:
  - 16/32 Bit databus width
  - PC133 SDRAM-compatible (125 MHz synchronous is used in ERTEC 200P)
  - 1 Bank with max. 256 MByte SDRAM (32-bit databus)
  - SDRAM support for following parts:

    - CAS latency: 2 or 3 clocks

    - Bank address bits (1/2/4 internal banks), realized via the lowest two bits of the address bus MA(1:0)

    - 8/9/10/11 bits column-address MA(13), MA(11:2)

    - max. 14 bits row-address MA(15:2)

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

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

Copyright © Siemens AG 2023. All rights reserved

- A maximum of 64 MByte address area for each chip select
- Acknowledgement delay monitoring for external components can be set by software.

The EMC interface only supports little-endian mode. The block diagram of the EMC module is shown in chapter 2.3.8.1.

2.3.8.1 Block diagram of the EMC

Figure 15: EMC block diagram

The EMC interface can only be driven by 1.8 V (necessary for mobile SDRAM). The EMC IO pads are separated from the other IOs and have their own power supply (VDD 1.8 V and GND). This supply ring contains all 77 IO pads of the EMC interface. So, the EMC can work with 1.8 V and the other IOs (e.g., GPIOs) with 3.3 V.

The driver strength of the 1.8 V EMC pads is 12 mA after PowerOn reset by default.

2.3.8.2 EMC Reset and Clock

The maximum memory configuration at the EMC interface is 2x SDRAM devices, 1x Peripheral / Level Shifter for 3.3 V. Each memory device gets its own clock. The ERTEC 200P has 3 clocks for the SDRAM memory (CLK_O_SDRAM2/1/0).

The CLK_O_SDRAM0 must be used for feedback of the external memory clock to the ERTEC 200P inputs CLK_I_SDRAM.

CLK_O_SDRAM1 is used for the respective memory device no.1 and CLK_O_SDRAM2 for the respective memory device no.2.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Figure 16: EMC interface with one SDRAM

CLK_O_SDRAM0 must always be returned back to CLK_I_SDRAM, even if only one external SRAM or an EMC/XHIF coupling to a second ERTEC 200P is used (see Figure 17). The feedback clock stores the incoming read data inside ERTEC 200P. The storing time can be influenced by board layout through the length of feedback clock line.

Figure 17: EMC interface with asynchronous RAM

All clocks can be switched off if they are not used. In the 'DRIVE_EMC' register (see chapter 2.3.14.11.27.1) are the appropriate bits for this function. CLK_O_SDRAM0/1, CLK_O_SDRAM2 can be switched separately.

NOTICE

After PowerOn reset all CLK_O_SDRAM0…2 are switched on.

Copyright © Siemens AG 2023. All rights reserved

2.3.8.3 AHB slave interface

The AHB slave interface supports the following features:

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

The AHB-slave interface contains an address code which divides the EMC address range of the AHB into 6 sub-ranges. The first sub-range is designed for 256 MByte SDRAM. The second range is subdivided into 4 areas of 64 MByte each and is designed for the SRAM and other asynchronous memory modules. The last range contains the internal EMC registers which have an address range of 1 MByte.

The interface can be adapted to the RAM blocks used by the configuration registers.

The following diagram shows the address range allocation.

Figure 18: EMC, address space

2.3.8.4 EMC notes

2.3.8.4.1 Remapping

Remapping the AHB address range from 0x0000_0000 to 0x0FFF_FFFF to the range of the SDRAM or the flash/SRAM block: This is required for some masters (ARM CPUs) because their exception table is expected when you boot at address 0x0000_0000.

The address may be moved from 0 to another address range depending on the MEM-SWAP signal.

- "00" → no action (Default: Boot ROM)
- "01" → start address of SDRAM device (sub-range #1)

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

- "10" → start address of asynchronous device (sub-range #2)
- "11" → no action (QVZ error upon access to 0x0000_0000, see chapter 2.3.4.5.1)

There is no address remapping with 00 or 11. With 01, address 0x0000_0000 is forwarded to the start address of the SDRAM. With 10, address 0x0000_0000 is forwarded to the start address of the asynchronous RAM, ROM or flash. In the latter two cases, address conversion is carried out so that the required memory can be reached.

If, for example, the exception table of a CPU is located in the flash memory (sub-range#2), an ARM-CPU requires this table at 0x0000_0000 during booting. The address is therefore remapped (in line with MEM_SWAP).

In order to avoid unnecessary wait states at the AHB bus, the EMC can buffer a maximum of 4 instances of AHB write access and acknowledge them immediately (provided there are no more than 4). Read access is delayed until the write buffer is empty (all data have been written).

2.3.8.4.2 Maximum number of wait cycles

The maximum number of wait cycles should be configured at least 5 times bigger than the expected worst case delay time of the external wait signal. False QVZ interrupts are possible when the number of wait cycles and the delay of the external wait signal have the same size. There is no way to detect a false QVZ interrupt by software but to compare a read value with an expected value. If a false QVZ interrupt is generated then the stored QVZ address can be false, too.

2.3.8.4.3 Shift mode with the asynchronous EMC interface

Irrespective of the configuration of the boot pins, i.e., in every boot mode, the asynchronous EMC interface (CS0 – CS3) is configured in shift mode so that a max. address range of 64 MByte can be addressed per chip select. The EXTENDED_CONFIG.ASYNC_ADDR_MODE bit is set in the EMC interface by the primary boot loader (see chapter 2.3.3.5.1).

This parameter assignment allows you to address:

- 16 MByte with a data bus width of 8 bits (byte)
- 32 MByte with a data bus width of 16 bits (halfword)
- 64 MByte with a data bus width of 32 bits (word).

2.3.8.4.4 Control of an external driver

The signals DTXR and XOE_DRIVER at the EMC are used to control external drivers (timing, see chapter 2.3.8.5.1 and 2.3.8.5.2). These signals can only be used for asynchronous memory. The use of these signals is optional.

As the signals for controlling the drivers are always activated irrespective of the address range of the asynchronous EMC interface (CS0 – CS3) and not on a chip select-specific basis, the driver becomes active even when memory without/before the driver is accessed; in this case two devices would drive the data bus for the EMC.

Figure 19: Control of an external driver

To prevent this from happening, a chip select-specific enable for driver control can be configured with the SCRB register 'EXT_DRIVER_EN' (see chapter 2.3.14.11.19).

Driver control is disabled after a reset. To allow the flash block in the NOR flash boot (see chapter 2.3.3.5.2) to be operated both before and after the driver, the driver control information (*EXT_DRIVER_DISABLE_CS0*) for address range: 0x3000_0000 – 0x33FF_FFFF (chip select 0) is also latched to *EXT_DRIVER_EN.CS0_ENABLE* over EMC pin XRDY_BF when the reset is cleared. The integrated pull-up means that control for an external driver is deactivated by default and needs to be activated with a pull-down for the module.

2.3.8.4.5 QVZ Acknowledgement Delay

If the ready signal does not respond in time with asynchronous RAMs (...ROMs, all devices with ready), an interrupt is generated and the address that has generated the error is saved. This is known as QVZ (German: **Q**uittungs**v**er**z**ug, acknowledgement delay) monitoring. The signals ASYNC_WAIT and BF_RDY are monitored. The QVZ interrupt (EMC_QVZ_INT, see 2.3.5.9) is triggered synchronously to the AHB clock.

All read and write access to external asynchronous RAMs interrupts the SDRAM refresh cycles until access is ended. The value MAX_EXT_WAIT is the limit for one complete instance of burst access on the RAM interface, not at the AHB bus. For example, 32-bit read access to an 8-bit block is converted to a burst with 4 reads. All 4 reads must be carried out within the set time window (MAX_EXT_WAIT).

2.3.8.4.6 Configuration of asynchronous EMC interface

Configuration of EXTENDED_CONFIG.SODM = '1' during the operation and active asynchronous memory (EMC) interface is not allowed. Only an indirect configuration is possible when the instruction code for writing the EXTENDED_CONFIG.SODM is transferred in I-TCM of ARM926 and will be operated from there. This ensures that no access to the asynchronous memory interface is active during the configuration of EXTENDED_CONFIG.SODM.

2.3.8.5 Timing of the Asynchronous Memory interfaces

The asynchronous interface can be activated by writing a '1' to the ADB bit of the 'EXTENDED_CONFIG' register. The data bus is then kept at '1' for one cycle after access. With pull-up resistors, this speeds up capacity reloading (with wiring). The internal pull-ups drive the data bus after the active phase. No external pull-ups are therefore required.

For further explanation, SETUP, STROBE and HOLD phases are defined for asynchronous access.

The figures below detail the individual phases in read and write access.

2.3.8.5.1 Read access

The SETUP phase starts with an active CS signal and ends when the output enable signal becomes active. The STROBE phase then starts. The STROBE phase lasts until the output enable signal is deactivated again. This then starts the hold phase. The hold phase ends when the CS signal is deactivated.

Copyright © Siemens AG 2023. All rights reserved

Figure 20: EMC, notation definition for a read access (ASYNC)

2.3.8.5.2 Write access

CS and WE signals are key to the individual phases in write access. See the write timing below.

Figure 21: EMC, notation definition for a write access (ASYNC)

For the asynchronous interface (SRAM), the data is valid until the end of the HOLD phase.

2.3.8.6 Application examples

Figure 22 shows an example of connections between EMC and external SDRAM devices.

Figure 22: EMC, application example: SDRAM

Figure 23 shows an example of connections between EMC and external SDRAM device in combination with an external driver to separate SDRAM- and SRAM-like devices.

Figure 23: EMC, application example: Combination of SDRAM and asynchronous interfaces

2.3.8.7 Address mapping

Start_Address	End_Address	Modul/Memory_Name	Interface	Fill_Mode
10D0.0000h	10DF.FFFFh	EMC REGISTER		1 MB
2000.0000h	2FFF.FFFFh	SDRAM		256 MB
3000.0000h	33FF.FFFFh	ASYNC_CHIP_SELECT0		64 MB
3400.0000h	37FF.FFFFh	ASYNC_CHIP_SELECT1		64 MB
3800.0000h	3BFF.FFFFh	ASYNC_CHIP_SELECT2		64 MB
3C00.0000h	3FFF.FFFFh	ASYNC_CHIP_SELECT3		64 MB

Start-Address	End-Address	Modul/Memory-Name
0h	37h	emc_reg

Module	Register/Memory	Read	Write	Access	Address
/emc_reg				32-Bit	
	/REVISION_CODE	r		32-Bit	0h
	/ASYNC_WAIT_CYCLE_CONFIG	(r)	(w)	32-Bit	4h
	/SDRAM_CONFIG	(r)	(w)(t)(p)	32-Bit	8h
	/SDRAM_REFRESH	(r)(h)	(w)	32-Bit	Ch
	/ASYNC_BANK0	r	w	32-Bit	10h
	/ASYNC_BANK1	r	w	32-Bit	14h
	/ASYNC_BANK2	r	w	32-Bit	18h
	/ASYNC_BANK3	r	w	32-Bit	1Ch
	/EXTENDED_CONFIG	(r)	(w)	32-Bit	20h
	/AT_ADDR	rh		32-Bit	24h
	/LPEMR	(r)	(w)	32-Bit	28h
	/BF_CONFIG	(r)	(w)	32-Bit	2Ch
	/PM_CONFIG	(r)	(w)	32-Bit	30h
	/RECOV_CONFIG	(r)	(w)	32-Bit	34h

2.3.8.8 Register description

Note

The parameter "ASYNC_ADDR_MODE" in the EMC-IP is set to '1' by the ERTEC 200P boot code.

Module: emc_reg

Register:	REVISION_CODE			Address:	0h	
Bits:	31dt0	Reset value:	4200805h	Attributes:	r	
Description:	Revision Code					

Bit	Identifier	Reset	Attr.		Function / Description
10dt0	LABEL	5h	r		Release Label
15dt11	INCREMENT	1h	r		Increment

18dt16	PATCH	0h	r		Patch Version
20dt19	PLATFORM	0h	r		Platform code
31dt21	ID	21h	r		Identification code

Register:	ASYNC_WAIT_CYCLE_CONFIG			Address:	4h	
Bits:	31dt0	Reset value:	40000080h	Attributes:	(r)	(w)
Description:	Async. Wait Cycle Configuration register					
Bit	Identifier	Reset	Attr.		Function / Description	
19dt0	MAX_EXT_WAIT	80h	r	w	Maximum number of wait cycles. If an access is delayed by WAIT input for more than [(MAX_EXT_WAIT+1) x 16] clocks, the access is completed.	
23dt20	reserved	0h				
24	BE_CTRL1	0h	r	w	Byte Enable Control Async. Bank 1: 0 = all Byte Enables low during read 1 = Byte Enable reflect AHB HSIZE during read	
25	BE_CTRL2	0h	r	w	Byte Enable Control Async. Bank 2: 0 = all Byte Enables low during read 1 = Byte Enable reflect AHB HSIZE during read	
26	BE_CTRL3	0h	r	w	Byte Enable Control Async. Bank 3: 0 = all Byte Enables low during read 1 = Byte Enable reflect AHB HSIZE during read	
27	BE_CTRL4	0h	r	w	Byte Enable Control Async. Bank 4: 0 = all Byte Enables low during read 1 = Byte Enable reflect AHB HSIZE during read	
29dt28	reserved	0h				
30	WAIT_POLARITY	1h	r	w	WAIT input polarity 0 = low active 1 = high active	
31	reserved	0h				

Register:	SDRAM_CONFIG			Address:	8h	
Bits:	31dt0	Reset value:	2020h	Attributes:	(r)	(w)(t)(p)
Description:	SDRAM Configuration register A write on this register starts a Load Mode Register sequence. This register can only be written after the SDRAM Initialization sequence is done (Bit 29 of register SDRAM_REFRESH).					
Bit	Identifier	Reset	Attr.		Function / Description	
2dt0	PAGE_SIZE	0h	r	w	Page Size 000 = 8 Column address lines 001 = 9 Column address lines 010 = 10 Column address lines 011 = 11 Column address lines	

					100-111 = reserved
3	reserved	0h			
6dt4	SDRAM_BANKS	2h	r	w	Number of banks inside SDRAM 000 = 1 Bank 001 = 2 Bank 010 = 4 Bank 011-111 = reserved
7	reserved	0h			
10dt8	ROW_SIZE	0h	r	wpt	Row Size 000 = 8 Row address lines 001 = 9 Row address lines 010 = 10 Row address lines 011 = 11 Row address lines 100 = 12 Row address lines 101 = 13 Row address lines 110 = 14 Row address lines 111 = 15 Row address lines Write protected as long as bit29 in SDRAM_REFRESH = 0. A write starts a Mode Register Set to external SDRAM.
12dt11	reserved	0h			
13	CL	1h	r	wpt	CAS Latency 0 = 2 clocks 1 = 3 clocks Write protected as long as bit29 in SDRAM_REFRESH = 0. A write starts a Mode Register Set to external SDRAM.
31dt14	reserved	00000h			

Register:	SDRAM_REFRESH			Address:	Ch	
Bits:	31dt0	Reset value:	190h	Attributes:	(r)(h)	(w)
Description:	SDRAM Refresh Control register					

Bit	Identifier	Reset	Attr.		Function / Description
12dt0	REFRESH_RATE	190h	r	w	Refresh Rate Number of clocks between 2 SDRAM Refresh cycles = REFRESH_RATE + 1
28dt13	reserved	0000h			
29	INIT_DONE	0h	rh		SDRAM Initialization done 0 = default 1 = SDRAM initialization ofter Hardware Reset is done (needs 28670 clocks)
30	AT	0h	rh		Async. Timeout 0 = default 1 = Async. access was finished because Wait Cycle Counter expired.
31	reserved	0h			

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Register:	ASYNC_BANK0					Address:	10h	
Bits:	31dt0		Reset value:		3FFFFFF2h	Attributes:	r	w
Description:	Async. Bank 0 Configuration register							
Bit	Identifier		Reset	Attr.		Function / Description		
1dt0	SIZE		2h	r	w	Bank Size 00 = 8 bit device 01 = 16 bit device 10 = 32 bit device 11 = reserved		
3dt2	MODE		0h	r	w	Mode 00 = SRAM 01 = Pagemode ROM 10 = reserved 11 = Burstflash ROM		
6dt4	R_HOLD		7h	r	w	Read Hold Cycles Hold Phase of read access lasts (R_HOLD + 1) clocks.		
12dt7	R_STROBE		3Fh	r	w	Read Strobe Duration Cycles Strobe Phase of read access (Read Enable = 0) lasts (R_STROBE + 1) clocks. Value 0 is not supported.		
16dt13	R_SU		Fh	r	w	Read Setup Cycles Setup Phase of read access lasts R_SU clocks.		
19dt17	W_HOLD		7h	r	w	Write Hold Cycles Hold Phase of write access lasts (W_HOLD + 1) clocks.		
25dt20	W_STROBE		3Fh	r	w	Write Strobe Duration Cycles Strobe Phase of write access (Write Enable = 0) lasts (W_STROBE + 1) clocks.		
29dt26	W_SU		Fh	r	w	Write Setup Cycles Setup Phase of write access lasts W_SU clocks.		
30	EW		0h	r	w	Extend Wait mode 0 = WAIT input = don't care 1 = Wait input extends access (valid in SRAM mode only)		
31	WSM		0h	r	w	Wait input synchronisation mode 0 = 2 Flipflops 1 = 1 Flipflop		

Register:	ASYNC_BANK1					Address:	14h	
Bits:	31dt0		Reset value:		3FFFFFF2h	Attributes:	r	w
Description:	Async. Bank 1 Configuration register							
Bit	Identifier		Reset	Attr.		Function / Description		
1dt0	SIZE		2h	r	w	Bank Size 00 = 8 bit device 01 = 16 bit device 10 = 32 bit device 11 = reserved		
3dt2	MODE		0h	r	w	Mode 00 = SRAM 01 = Pagemode ROM		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

					10 = reserved 11 = Burstflash ROM
6dt4	R_HOLD	7h	r	w	Read Hold Cycles Hold Phase of read access lasts (R_HOLD + 1) clocks.
12dt7	R_STROBE	3Fh	r	w	Read Strobe Duration Cycles Strobe Phase of read access (Read Enable = 0) lasts (R_STROBE + 1) clocks. Value 0 is not supported.
16dt13	R_SU	Fh	r	w	Read Setup Cycles Setup Phase of read access lasts R_SU clocks.
19dt17	W_HOLD	7h	r	w	Write Hold Cycles Hold Phase of write access lasts (W_HOLD + 1) clocks.
25dt20	W_STROBE	3Fh	r	w	Write Strobe Duration Cycles Strobe Phase of write access (Write Enable = 0) lasts (W_STROBE + 1) clocks.
29dt26	W_SU	Fh	r	w	Write Setup Cycles Setup Phase of write access lasts W_SU clocks.
30	EW	0h	r	w	Extend Wait mode 0 = WAIT input = don't care 1 = Wait input extends access (valid in SRAM mode only)
31	WSM	0h	r	w	Wait input synchronisation mode 0 = 2 Flipflops 1 = 1 Flipflop

Register:	ASYNC_BANK2				Address:	18h	
Bits:	31dt0	Reset value:	3FFFFFF2h		Attributes:	r	w
Description:	Async. Bank 2 Configuration register						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	SIZE	2h	r	w	Bank Size 00 = 8 bit device 01 = 16 bit device 10 = 32 bit device 11 = reserved		
3dt2	MODE	0h	r	w	Mode 00 = SRAM 01 = Pagemode ROM 10 = reserved 11 = Burstflash ROM		
6dt4	R_HOLD	7h	r	w	Read Hold Cycles Hold Phase of read access lasts (R_HOLD + 1) clocks.		
12dt7	R_STROBE	3Fh	r	w	Read Strobe Duration Cycles Strobe Phase of read access (Read Enable = 0) lasts (R_STROBE + 1) clocks. Value 0 is not supported.		
16dt13	R_SU	Fh	r	w	Read Setup Cycles Setup Phase of read access lasts R_SU clocks.		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

19dt17	W_HOLD	7h	r	w	Write Hold Cycles Hold Phase of write access lasts (W_HOLD + 1) clocks.
25dt20	W_STROBE	3Fh	r	w	Write Strobe Duration Cycles Strobe Phase of write access (Write Enable = 0) lasts (W_STROBE + 1) clocks.
29dt26	W_SU	Fh	r	w	Write Setup Cycles Setup Phase of write access lasts W_SU clocks.
30	EW	0h	r	w	Extend Wait mode 0 = WAIT input = don't care 1 = Wait input extends access (valid in SRAM mode only)
31	WSM	0h	r	w	Wait input synchronisation mode 0 = 2 Flipflops 1 = 1 Flipflop

Register:	ASYNC_BANK3				Address:	1Ch	
Bits:	31dt0	Reset value:		3FFFFFF2h	Attributes:	r	w
Description:	Async. Bank 3 Configuration register						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	SIZE	2h	r	w	Bank Size 00 = 8 bit device 01 = 16 bit device 10 = 32 bit device 11 = reserved		
3dt2	MODE	0h	r	w	Mode 00 = SRAM 01 = Pagemode ROM 10 = reserved 11 = Burstflash ROM		
6dt4	R_HOLD	7h	r	w	Read Hold Cycles Hold Phase of read access lasts (R_HOLD + 1) clocks.		
12dt7	R_STROBE	3Fh	r	w	Read Strobe Duration Cycles Strobe Phase of read access (Read Enable = 0) lasts (R_STROBE + 1) clocks. Value 0 is not supported.		
16dt13	R_SU	Fh	r	w	Read Setup Cycles Setup Phase of read access lasts R_SU clocks.		
19dt17	W_HOLD	7h	r	w	Write Hold Cycles Hold Phase of write access lasts (W_HOLD + 1) clocks.		
25dt20	W_STROBE	3Fh	r	w	Write Strobe Duration Cycles Strobe Phase of write access (Write Enable = 0) lasts (W_STROBE + 1) clocks.		
29dt26	W_SU	Fh	r	w	Write Setup Cycles Setup Phase of write access lasts W_SU clocks.		
30	EW	0h	r	w	Extend Wait mode 0 = WAIT input = don't care		

					1 = Wait input extends access (valid in SRAM mode only)
31	WSM	0h	r	w	Wait input synchronisation mode 0 = 2 Flipflops 1 = 1 Flipflop

Register:	EXTENDED_CONFIG			Address:	20h	
Bits:	31dt0	Reset value:	1F70000h	Attributes:	(r)	(w)
Description:	Extended Configuration register					
Bit	Identifier	Reset	Attr.		Function / Description	
6dt0	reserved	00h				
7	AT_EN	0h	r	w	Async. Timeout Enable 0 = disabled 1 = enabled	
8	SDRAM_DW	0h	r	w	SDRAM interface data width 0 = 32 bit 1 = 16 bit	
9	MOBILE_SDRAM	0h	r	w	Mobile SDRAM Mode 0 = disabled (1 SDRAM mode register) 1 = enabled (2 SDRAM mode register)	
10	ASYNC_ADDR_MODE	0h	r	w	Async. Address Output Mode 0 = no address shift (ERTEC200 compatible) MA(23:0) = HADDR(23:0) 1 = address shift depending on configured memory datawidth datawidth = 8 bit : MA(23:0) = HADDR(23:0) datawidth = 16 bit : MA(23:0) = HADDR(24:1) datawidth = 32 bit : MA(23:0) = HADDR(25:2)	
13dt11	reserved	0h				
15dt14	TRCD	0h	r	w	TRCD (RAS to CAS delay) 00 = 2 clocks 01 = 3 clocks 10 = 4 clocks 11 = reserved	
18dt16	SDRAM_BL	7h	r	w	SDRAM Burst Length This value is written into SDRAM Mode Register. 000 = 1 (32 bit SDRAM interface only) 001 = 2 (16 bit SDRAM interface only) 010 = 4 (not supported) 011 = 8 100-110 = reserved 111 = continuous/full page (recommended)	
19	reserved	0h				
23dt20	TRFC	Fh	r	w	AUTO REFRESH period (cycle) time	

					Minimum time between to AUTO REFRESH commands in clocks 0x0 = reserved 0x1 = 4 clocks : 0xF = 18 clocks
24	ADB	1h	r	w	active data bus 0 = disabled 1 = enabled
25	reserved	0h			
26	reserved	0h	r	w	
27	SODM	0h	r	w	SDRAM Output Delay Mode 0 = outputs are delayed with register clocked by input SDRAM clock 1 = outputs are delayed with delay buffers
28	TM4	0h	r	w	Test Mode bit 4 0 = normal operation 1 = async write data outputs toggle 1 clock after address outputs
29	TM2	0h	r	w	Test Mode bit 2 0 = normal operation 1 = all SDRAM reads/writes are of type miss (32 bit SDRAM only)
30	TM1	0h	r	w	Test Mode bit 1 0 = normal operation 1 = SDRAM initialization delay after reset is cleared
31	reserved	0h			

Register:	AT_ADDR			Address:	24h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:	Async. Timeout Address register Contains after asyc. timeout address of the access, which was finished with timeout.					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	AT_ADDR	00000000h	rh		Async. Timeout Address register Contains after asyc. timeout address of the access, which was finished with timeout.	

Register:	LPEMR			Address:	28h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	Low Power Extended Mode Register This value is written into Extended Mode Register of Mobile SDRAM					
Bit	Identifier	Reset	Attr.		Function / Description	
12dt0	LPEMR	0h	r	w	This value is written into Extended Mode Register of Mobile SDRAM. See Mobile SDRAM datasheet.	
31dt13	reserved	00000h				

Register:	BF_CONFIG	Address:	2Ch

Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	Burstflash ROM Configuration register					

Bit	Identifier	Reset	Attr.		Function / Description
0	BF_CLK_F	0h	r	w	Clock Frequency 0 = Burstflash clock = HCLK/2 1 = Burstflash clock = HCLK (HCLK = AHB clock)
1	BF_CLK_EN	0h	r	w	Clock Enable 0 = Burstflash Clock always at fixed level defined in bit 3 1 = Burstflash clock toggles
2	AVD_MODE	0h	r	w	Address Valid output mode 0 = AVD always low 1 = AVD toggles
3	BF_CDV	0h	r	w	Clock Disable Value 0 = Burstflash Clock fixed level = 0 1 = Burstflash Clock fixed level = 1
7dt4	reserved	0h			
10dt8	RM	0h	r	w	Read Mode 000 = Continuos 001 = 8-word linear without wrap around 010 = 16-word linear without wrap around 011 = 32-word linear without wrap around 100 = reserved 101 = 8-word linear with wrap around 110 = 16-word linear with wrap around 111 = 32-word linear with wrap around
13dt11	reserved	0h			
14	RDY_DELAY	0h	r	w	Ready Delay Mode 0 = RDY active with data 1 = RDY active one clock cycle before data
15	SYNC_READ	0h	r	w	Set Device Read Mode 0 = Asynchronous Read Mode 1 = Synchronous Read (Burst Mode) enabled
19dt16	AVD_DELAY	0h	r	w	AVD delay 0000 = 1 AHB clock .. 1111 = 16 AHB clocks Delay after falling edge of chip select to falling edge of AVD
22dt20	AVD_PW	0h	r	w	AVD pulse width 000 = 1 AHB clock .. 111 = 8 AHB clocks
31dt23	reserved	000h			

Register:	PM_CONFIG	Address:	30h

Bits:	31dt0	Reset value:	3Fh	Attributes:	(r)	(w)
Description:	Pagemode ROM Configuration register					

Bit	Identifier	Reset	Attr.		Function / Description
5dt0	PACC	3Fh	r	w	Page Access time Strobe Phase of page read access lasts (PACC + 1) clocks.
6	PAGE_MODE	0h	r	w	0 = 4 word page 1 = 8 word page
31dt7	reserved	0000000h			

Register:	RECOV_CONFIG			Address:	34h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	Recovery Phase Configuration register					

Bit	Identifier	Reset	Attr.		Function / Description
3dt0	RECOV0	0h	r	w	Async. Bank 0 Recovery Phase 0000 = no recovery Phase 0001 = 1 clock 0010 = 2 clocks .. 1111 = 15 clocks
7dt4	RECOV1	0h	r	w	Async. Bank 1 Recovery Phase 0000 = no recovery Phase 0001 = 1 clock 0010 = 2 clocks .. 1111 = 15 clocks
11dt8	RECOV2	0h	r	w	Async. Bank 2 Recovery Phase 0000 = no recovery Phase 0001 = 1 clock 0010 = 2 clocks .. 1111 = 15 clocks
15dt12	RECOV3	0h	r	w	Async. Bank 3 Recovery Phase 0000 = no recovery Phase 0001 = 1 clock 0010 = 2 clocks .. 1111 = 15 clocks
31dt16	reserved	0000h			

Copyright © Siemens AG 2023. All rights reserved

2.3.9 RSC 1 and 2 (Reed-Solomon Controllers)

In ERTEC 200P Step 3 there are two RS controllers available.

Note

Please **note** that the ERTEC 200P Step 3 does not support booting from the external NAND flash.

2.3.9.1 Module description

The RS controller performs the required EDC (Error Detection and Correction) autonomously for each kind of memory. Previously this was executed by means of software. Due to the increasingly smaller technologies and the resultant probability of possible errors, new memory modules are designed for an ever greater number of bytes that must be corrected via EDC.
In the past the flash manufacturers required that 2 bits in 512 bytes be detected as faulty and 1 bit be corrected. In the future the flash manufacturers will require that at least 6 bits in 512 bytes be detected as faulty and be corrected.

The previously used SW solution (usage of 6 check bytes, so that 3 bytes in 256 bytes were corrected safely and 4 faulty bytes were detected at a rate of 99 %) therefore no longer fulfills the requirements for new memory modules. Since the calculation / SW runtime increases exponentially during the calculation of the RSC check bytes in the software and the required higher correction rate (6 bytes in 256 data bytes), this can no longer be implemented in the SW. Therefore, this coding has to be performed in the hardware.

The RS controller can therefore recognize a **maximum of 8 bytes** in 256 data bytes as faulty and correct them. To this purpose 16 RCS check bytes are required so that the number of user data bytes is reduced to 238 bytes within a block (maximum of 254 bytes).

→ Gross code = (238 bytes + 16 bytes) / 238 bytes x net code

More than 8 faulty bytes cannot be detected reliably anymore (for example 9 faulty bytes are detected at a rate of 99 %) and can therefore not be corrected correctly. The behavior of the RS controller is therefore undefined as of > 8 faulty bytes.

The RS code has the advantage that it is memory independent.

In the case of an RSC code the maximum sum of the data bytes and RSC check bytes is 255. However, to reduce the required memory in a ROM, the sum of data and RSC check bytes can also be smaller.

As the number of RSC check bytes increases, the maximum number of data bytes that can be saved in a block is reduced. This means that while the net data amount remains the same, the gross data amount that has to be stored on a memory increases.

Copyright © Siemens AG 2023. All rights reserved

2.3.9.2 Block diagram

Figure 24: RSC block diagram

2.3.9.3 Function description

The RS controller provides the following operating modes:

- Encoder mode (AHB), see chapter 2.3.9.3.1.1:

  The data is written into the encoder via the AHB Slave Interface of the RS controller and transferred from the encoder via the AHB Master Interface (typically to the EMC interface). After the maximum number of data bytes (1…238 bytes) has been reached, the check bytes (16 bytes) are automatically transferred to the AHB Master Interface.

- Encoder mode (RSC-RAM), see chapter 2.3.9.3.1.2:

  The data is written into the encoder via the AHB Slave Interface of the RS controller and stored by the encoder in the RSC-RAM in ascending order from the address 0x00…00 onwards. After the maximum number of data bytes (1…238 bytes) has been reached, the check bytes (16 bytes) are written into the RSC-RAM.

- Decoder mode (AHB), see chapter 2.3.9.3.1.3:

  The data bytes (1…238 bytes) including the check bytes (16 bytes) are read from the AHB Master Interface of the RS controller (typically via the EMC interface) and written into the RSC-RAM. The RS controller then performs the decoding and the data is corrected, if necessary.

- Decoder mode (RSC-RAM), see chapter 2.3.9.3.1.4:

  The data bytes (1…238 bytes) incl. the check bytes (16 bytes) are written via the AHB Master interface of the RS controller into the RSC-RAM. The RS controller then performes the decoding and the data is corrected, if necessary.

The number of symbols to be corrected is determined by means of the HW settings of the RS controller. Thereby the number of check bytes is defined (always 16 bytes), which influences the number of max. data bytes (max. data bytes + check bytes = 254 bytes). However, fewer data bytes are also supported by the RS controller (1…238 data bytes).

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Note

Before starting a new job (encoder or decoder procedure), the firmware must ensure that the current procedure has been completed (*ENC_Status.run* = 0 or *DEC_Status.run* = 0). No hardware locking takes place between the jobs.

The access paths are displayed in the block diagram (see chapter 2.3.9.2). The control logic of the encoder or decoder selects the path to the AHB Master Interface or the RSC-RAM, depending on whether an encoder is active or a decoder. Only **one action can be active at one time**, this means, **either coding** is performed **or decoding** of the data takes place.

2.3.9.3.1 Operating modes

2.3.9.3.1.1 Encoder mode (AHB)

The data is written into the encoder via the AHB Slave Interface of the RS controller and transferred from the encoder directly via the AHB Master Interface (typically to the EMC interface). After the data bytes (1…238 bytes) configured in *ENC_Length* have been reached, the check bytes (16 bytes) are automatically transferred via the AHB Master Interface.

The number of data to be coded is determined by means of the AHB accesses (byte, halfword, word) on the AHB Slave Interface, this means on *ENC_Data*. While the number of data bytes configured in *ENC_Length* is being transferred, the coding procedure (encoder) takes place in parallel. Data bytes not required within the last AHB access (halfword, word) are ignored, meaning discarded (see also the following table).

The write data in the encoder, meaning on *ENC_Data*, can be transferred via 8-bit/16-bit/32-bit access. For the accesses which are allowed on *ENC_Data*, refer to the following table.

Table 19: RSC encoder mode

	AHB: WR control		AHB: WR data hwdata(31:0)	Expected data bytes in ENC_Data	Used data bytes in ENC_Data
	hsize(1:0)	haddr(1:0)			
Byte	00 (byte)	00	xxxx xxAA	≥ 1	AA
	00 (byte)	01	xxxx BBxx	Not permitted [1]	
	00 (byte)	10	xxCC xxxx	≥ 1	CC
	00 (byte)	11	DDxx xxxx	Not permitted [1]	
Halfword	01 (halfword)	00	xxxx BBAA	1	AA
				≥ 2	BBAA
	01 (halfword)	10	DDCC xxxx	Not permitted [1]	
Word	10 (word)	00	DDCC BBAA	1	AA
				2	BBAA
				3	CCBB AA
				≥ 4	DDCC BBAA

[1] AHB access is processed correctly, but *ENC_Data* contains incorrect data.

Note

A write burst access of *ENC_Data* is not possible at the AHB Slave Interface since the data must always be written to one AHB address only.

When the number of data bytes configured in *ENC_Length* has been transferred, the check bytes (16 bytes) are written by the encoder via the AHB Master Interface.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

NOTICE

If a further write access is initiated at the AHB Slave Interface while the check byte is being written, it is accepted at the AHB Slave Interface (posted WR). However, it is not processed in the RS controller so that a subsequent write access is not acknowledged (ready delay) by the AHB Slave Interface until the current coding procedure has been completed.

During the transfer of the coded data including check bytes via the AHB Master Interface of the RS controller, the write address can:

- **Remain the same**. This means the data and check bytes are always written to the same address. This is required for the access (writing) on a NAND flash. The configurable data width of the transfer can amount to 8, 16 or 32 bits. Missing bytes at the end of the transfer are always filled with 0xFF…FF.
- **Be incremented**. Depending on the configured data width (8, 16 or 32 bits), an address increment is performed as follows.
  - 8-bit data width (byte): address + 1
  - 16-bit data width (halfword): address + 2
  - 32-bit data width (word): address + 4

  This is required for the access (writing) on a NOR flash since the data is always stored there continuously. Missing bytes at the end of the transfer are always filled with 0xFF…FF.

The addressing of the AHB Master Interface is always limited to the configured data width and is therefore aligned in accordance with the configurable data width:

- All addresses are used for byte accesses (8-bit).
- The address bit A0 is set to zero for halfword accesses (16-bit).
- The address bits A0 and A1 are set to zero for word accesses (32-bit).

Only Single-WR accesses are supported by the encoder at the AHB Master Interface, **no** burst WR accesses. As a general rule read accesses are not performed by the encoder at the AHB Master Interface.

The Single-WR accesses are processed posted (bufferable), this means at least the first WR access on the address range configured in *ENC_DSTaddr* is acknowledged **immediately** without the WR data having to be applied in the target memory (register, RAM). This is accomplished by additionally setting the required sideband signal: ahbm_hprot_o(2). **Therefore, the access time and duration of the bus assignment can be reduced with regard to the AHB Master**.

The last write transfer is always performed as "not posted" as secure detection that the WR transfer has been performed at the ML-AHB.

2.3.9.3.1.2 Encoder mode (RSC-RAM)

The configured number of data bytes is written into the encoder via the AHB Slave Interface of the RS controller and from there stored in the RSC-RAM, ascending from RAM address 0x00…00 (AHB address: 0x0100). After the number of data bytes (1…238 bytes) configured in *ENC_Length* has been reached, the check bytes (16 bytes) are automatically written into the RSC-RAM.

The number of data to be coded is determined by means of the AHB accesses (byte, halfword, word) at the AHB Slave Interface. While the number of data bytes configured in *ENC_Length* is being transferred, the coding procedure (encoder) takes place in parallel. Data bytes not required within the last AHB access (halfword, word) are ignored, meaning discarded (see Table 19).
The write data in the encoder can be transferred via 8-bit-/16-bit/32-bit access. For details of the accesses see Table 19.

Copyright © Siemens AG 2023. All rights reserved

Note

A write burst access of *ENC_Data* is not possible at the AHB Slave Interface since the data can always only be written to one AHB address.

Note

Error scenario (has to be prevented via configuration):
If a further write access is initiated at the AHB Slave Interface while the check byte is being written into the RSC-RAM, it is accepted at the AHB Slave Interface (posted WR). However, it is not processed in the RS Controller so that a subsequent write access is not acknowledged (ready delay) by the AHB Slave Interface until the current coding procedure has been completed. As a result, the data bytes and, if applicable, additionally the check bytes generated in the current coding procedure are overwritten.

When the number of data bytes configured in 'ENC_Length' is transferred, the check bytes (16 bytes) are written by the encoder into the RSC-RAM. To achieve this, the encoder generates the checksum and stores it after the data in the RSC-RAM. If the coding procedure including writing of the check bytes is completed (*ENC_Status.run* = 0), the data including the check bytes can be read via the AHB Slave Interface (typically by a processor) and transferred into the respective target memory.

The read data from the RSC-RAM can be read by means of 8-bit/16-bit/32-bit access. A burst access is supported at the RSC-RAM at the AHB Slave Interface.

NOTICE

If a read access is initiated at the AHB Slave Interface on the RSC-RAM during the decoding procedure (ENC_Status.run = 1), this access is performed. Only a read access during the writing of the check bytes is not acknowledged (ready delay) as long as the writing of the check bytes is not completed by the encoder.

In this operating mode, **no transfer takes place via the AHB Master Interface**. The use case is to write the data into an SPI flash using an external host. The external host reads the coded SPI flash data from the RS controller and passes it to the SPI.

2.3.9.3.1.3 Decoder mode (AHB)

The data bytes (1…238 bytes) including the check bytes (16 bytes) are read from the AHB Master Interface of the RS controller (typically via the EMC interface) and written into the RSC-RAM. The RS controller then performes the decoding and the data is corrected, if required.

After reading of the configured data including number of check bytes via the AHB Master Interface, the decoding (decoder) starts.

At the AHB Master Interface of the RS controller the read address can:

- **Remain the same**. This means the data and check bytes are always read by the same address. This is required for the access (reading) on a NAND flash. The configurable data width of the transfer can amount to 8, 16 or 32 bits. Missing bytes at the end of the transfer are ignored.
- **Be incremented**. Depending on the configured data width (8, 16 or 32 bits), an address increment is performed as follows.
  - 8-bit data width (byte): address + 1
  - 16-bit data width (halfword): address + 2
  - 32-bit data width (word): address + 4

  This is required for the access (writing) on a NOR flash since the data is always stored there continuously. Too many read bytes at the end of the transfer are ignored.

The addressing of the AHB Master Interface is always limited to the configured data width and is therefore aligned in accordance with the configurable data width:

- All addresses are used for byte accesses (8-bit).
- The address bit A0 is set to zero for halfword accesses (16-bit).
- The address bits A0 and A1 are set to zero for word accesses (32-bit).

Only the Single-RD accesses at the AHB Master Interface are supported by the decoder, **no** burst RD accesses. As a general rule write accesses are not performed by the encoder at the AHB Master Interface.

The number of data to be decoded is determined by means of the AHB accesses (byte, halfword, word) at the AHB Master Interface. If the number of data bytes configured in *DEC_Length* has been transferred, the decoding procedure (decoder) starts. Data bytes not required within the last AHB access (halfword, word) are ignored, meaning discarded.

The end of the decoding is indicated by *DEC_Status.run* = 0. In addition, the number of corrected errors is stored in *DEC_Status.corr_count*. The SW may only read the data via the AHB Slave Interface after the decoding is completed. This is indicated by *DEC_Status.run* = 0 and in case of:

- **Successful** decoding through the interrupt *decoder_finished_good* (without correction) or *decoder_finished_corr* (with correction)
- **Unsuccessful** decoding by the interrupt *decoder_finished_bad*

The read data from the RSC-RAM can be read by means of 8-bit/16-bit/32-bit access. A burst access is performed at the AHB Slave Interface to the RSC-RAM.

NOTICE

If a read access is already initiated at the AHB Slave Interface on the RSC-RAM during the decoding procedure (*DEC_Status.run* = 1), the access is not acknowledged (ready delay), as long as the decoding procedure including error correction is not completed (*DEC_Status.run* = 0).

2.3.9.3.1.4 Decoder mode (RSC-RAM)

The configured number of data bytes and check bytes (16 bytes) are saved to the RSC-RAM over the AHB slave interface of the RS controller in ascending order from address 0x00...00 (typically by a processor). Decoding is carried out by the RS controller and data is corrected, if necessary, when *DEC_Status.run* = 1 is written.

The number of items of data to be decoded is defined based on the number of data bytes configured in *DEC_Length*. Excess data entered in the RSC-RAM is ignored.

Note

The decoder interprets excess data bytes entered as check bytes.

The end of decoding is shown with *DEC_Status.run* = 0. The number of errors corrected is also saved in *DEC_Status.corr_count*. The SW may not read the data over the AHB slave interface until decoding is complete. This is indicated by *DEC_Status.run* = 0 or the interrupt *decoder_finished_good* or *decoder_finished_bad*.

The data in the RSC-RAM can be written and read with 8-bit/16-bit/32-bit access. An instance of burst access to the RSC-RAM over the AHB slave interface is supported.

Note

Any write access to the RSC-RAM at the AHB slave interface during the decoding process (*DEC_Status.run* = 1) will not be acknowledged (ready delay) until the decoding process and error correction are complete (*DEC_Status.run* = 0).

Copyright © Siemens AG 2023. All rights reserved

There is **no transfer over the AHB master interface** in this operating mode. Use case example: an external host reading data from an SPI flash. The external host writes the SPI flash data read to the RS controller for decoding.

2.3.9.3.2 Detailed description

2.3.9.3.2.1 Encoder

The encoder encodes the number of data bytes set in the 'ENC_Length' register and writes the data into the selected memory (RSC-RAM or external memory). When the number of data bytes has been reached, the check bytes are also written into the selected memory (RSC-RAM or external memory). The task of writing the check bytes is automatically assumed by the encoder control.

Figure 25: Example: Storage of 6 data bytes (max. 238) and 16 check bytes in RSC-RAM

Via the 'ENC_DSTmode' register the configuration of the write accesses takes place on the target storage medium.

ENC_DSTmode.bypass can be used to set whether the data is written via the AHB Master Interface or directly transferred from the RAM address 0x00...00 (AHB-/AXI address: 0x0100) beginning in ascending order into the RSC-RAM.

If the coded data is to be written via the AHB Master Interface, the width of the AHB WR transfer is determined via ENC_DSTmode.access_size. It can be 8, 16, or 32 bits. The encoder waits until the number of data has been received and encoded via the AHB Slave Interface in accordance with the configured access width (*ENC_DSTmode.access_size*). Only then is the AHB Master transfer started. If the access width (halfword, word) cannot be reached during the last data transfer, the remaining data bytes are filled with the 0xFF...FF.

If the coded data including check bytes cannot be written via the AHB Master Interface because of the missing acknowledgment of the buffer, the next WR access at the AHB Slave Interface is also acknowledged with a ready delay. Buffering of the data to be encoded in RSC-RAM does not take place.

Different storage types can be supported via the AHB Master Interface by using *ENC_DSTmode.incr_addr*. No address increment may be performed for NAND flash. By contrast an address increment is required for the transfer in a NOR-Flash or in an SDRAM. *ENC_DSTmode.incr_addr* is not relevant for *ENC_DSTmode.bypass* = "1" (RSC-RAM), an address increment always takes place in this case.

In the case of *ENC_DSTmode.incr_addr* = "1" (perform address increment) and *ENC_DSTmode.bypass* = "0" (AHB Master Interface), after completed coding including transfer of the data and check bytes via the AHB Master Interface (ENC_Status.run: 1 → 0) the target address in 'ENC_DSTaddr' register is set to the next address to be transferred.

Note

The AHB write address is always limited to the data width. In case of a word access (32-bit), the address bits A0 and A1 are set to zero, in case of the halfword access (16-bit) only the address A0. This is independent of the content in *ENC_DSTaddr*.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

This allows continuous writing to an external storage, without reprogramming this register. *ENC_DSTmode.access_size* is incremented dependent on 'ENC_DSTaddr':

- = 0 to +1 bytes
- = 1 to +2 bytes
- = 2 to +4 bytes

'ENC_DSTaddr' is not relevant and the content remains unchanged if no increment is performed (*ENC_DSTmode.incr_addr* = "0") or during the use of RSC-RAMs (*ENC_DSTmode.bypass* = "1").

During the use of the RSC-RAM (*ENC_DSTmode.bypass* = "1"), with each newly started encoding procedure of the encoder the data is written into the RAM address 0x00…00 (AHB address: 0x0100), beginning in ascending order, into the RSC-RAM. Here the user must ensure that the previously encoded data have already been fetched.

Note

Configuration note:

The values for 'ENC_DSTaddr' and 'ENC_DSTmode' may only be written by the user if the *ENC_Status.run* = "0" is set, otherwise this can lead to an uncontrollable behavior (a HW lock does not exist).

The data to be coded is handed over at the AHB Slave Interface to the encoder via 'ENC_Data' register.

The encoder encodes the data and stores it in the RSC-RAM or transfers it via the AHB Master Interface, depending on the *ENC_DSTmode.bypass*. If the number of data to be coded configured in 'ENC_Length' register is reached by means of the AHB accesses (byte, halfword, word) via the AHB Slave Interface, the calculated check bytes are transferred automatically.

If no data to be encoded has been transferred to the encoder, *ENC_Status.run* = "0" in 'ENC_Status' register.

If the first data byte is written into the encoder via 'ENC_Data', the *ENC_Status.run* switches from 0 → 1 until all data and check bytes have been transferred to the AHB Master Interface or have been entered in the RSC-RAM.

If this is the case, *ENC_Status.run* is set from 1 → 0 and in addition the dedicated output signal int_encoder_finished_o is triggered (see chapter 2.3.9.3.2.3). In parallel *encoder_finished* is stored in the group interrupt register 'IRQstatus' (see chapter 2.3.9.3.2.3).

2.3.9.3.2.2 Decoder

The decoder reads its data to be decoded via the AHB Master Interface or the data to be decoded are already available in the RSC-RAM.

Via the register 'DEC_SRCmode' the configuration of the read accesses takes place on the destination storage.

DEC_SRCmode.bypass can be used to set whether the data has to be read before the decoding via the AHB Master Interface or directly transferred, beginning from the RAM address 0x00…00 (AHB address: 0x0100) in ascending order, into the RSC-RAM via the AHB Slave Interface.

If the data to be decoded has to be read via the AHB Master Interface (*DEC_SRCmode.bypass* = "0"), the width of the AHB WR transfer is determined by the *DEC_SRCmode.access_size.* It can be 8, 16, or 32 bits. If during the data transfer more data is read than required because of the access width (halfword, word), this data will be ignored, meaning not used.

Different storage types can be supported by means of *DEC_SRCmode.incr_addr*. No address increment may be performed for NAND flash. By contrast an address increment is required when reading from a NOR flash or from an SDRAM. *DEC_SRCmode.incr_addr* is not relevant for *DEC_SRCmode.bypass* = "1" (RSC-RAM).

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

In the case of *DEC_SRCmode.incr_addr* = "1" (perform address increment) and *DEC_SRCmode.bypass* = "0" (AHB Master Interface), after completed decoding including correction of the data in RSC-RAM (*DEC_Status.run*: 1 → 0) the source address in 'DEC_SRCaddr' register is set to the next address to be read.

Note

The AHB read address is always limited to the configured data width (*DEC_SRCmode.access_size*). In case of a word access (32-bit), the address bits A0 and A1 are set to zero, in case of the halfword access (16-bit) only the address A0. This is independent of the content in *DEC_SRCaddr*.

This allows continuous reading from an external storage, without reprogramming this register. 'DEC_SRCaddr' depends on *DEC_SRCmode.access_size*:

- = 0 to +1 bytes
- = 1 to +2 bytes
- = 2 to +4 bytes

'DEC_SRCaddr' is not relevant and the content remains unchanged if no increment is performed (*DEC_SRCmode.incr_addr* = "0") or during the use of RSC-RAMs (*DEC_SRCmode.bypass* = "0").

Note

Configuration note:

The values for 'DEC_SRCaddr' and 'DEC_SRCmode' may only be written by the user if the *DEC_Status.run* = "0" is set, otherwise this can lead to an uncontrollable behavior (a HW lock does not exist).

The decoder waits until the configured number of data to be decoded as well the 16 check bytes are read by means of the AHB accesses (byte, halfword, word) via the AHB Slave Interface and are stored as follows in the RSC-RAM as of RAM address 0x00…00 (AHB address: 0x0100).

Figure 26: Example: Storage of 7 data bytes (max 238) and 16 check bytes in RSC-RAM

Afterwards the decoding takes place automatically by means of the read check bytes (16 bytes) and if necessary the correction of the data in the RSC-RAM (*DEC_Status.run* = "1").

In general, the decoding and correction procedure in DEC_Status register has to be started by setting *DEC_Status.run* = "1". The decoder then reads its data to be decoded via the AHB Master Interface or the data to be decoded is already available in the RSC-RAM.

The decoder is active with *DEC_Status.run* = "1". When the decoding or correction procedure has been completed, the decoder sets *DEC_Status.run* = "0". In addition, the number of corrected symbols is stored in *DEC_Status.corr_count.*

If all error bytes can be corrected, the number of corrected bytes is communicated in *DEC_Status.corr_count*. If not all error bytes can be corrected, *DEC_Status.corr_count* = 0xF is set.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Note

The user has to read the decoded and corrected data from the RSC-RAM via the AHB Slave Interface **before a subsequent decoding and correction process can be started by means of** *DEC_Status.run* = "1".

While *DEC_Status.run* = "1", each read and write access is acknowledged by the AHB Slave Interface with ready delay, until the decoding or correction procedure is completed (*DEC_Status.run* = "0"). Writing to *DEC_Status.run* is ignored during an ongoing decoding and correction procedure.

When the decoding and correction procedure is completed, *DEC_Status.correction* indicates whether a byte correction was performed.

- If **no** byte correction was performed, the dedicated output signal int_decoder_finished_good is triggered (see chapter 2.3.9.3.2.3). In parallel *decoder_finished_good* is stored in the group interrupt register 'IRQstatus' (see chapter 2.3.9.3.2.3).
- If at least one byte correction was performed successfully, the dedicated output signal int_decoder_finished_corr is triggered (see chapter 2.3.9.3.2.3). In parallel *decoder_finished_corr* is stored in the group interrupt register IRQstatus (see chapter 2.3.9.3.2.3). The decoding and correction procedure has been performed successfully if:
  - *DEC_Status.all_one* = "1" OR
  - *DEC_Status.all_zero* = "1" OR
  - all detected byte errors could be corrected (*DEC_Status.corr_count* ≠ 0xF)
- If a correction was not performed successfully because of too many byte errors, the dedicated output signal int_decoder_finished_bad is triggered (see chapter 2.3.9.3.2.3). In parallel *decoder_finished_bad* is stored in the group interrupt register 'IRQstatus' (see chapter 2.3.9.3.2.3). The decoding and coding procedure has **not** been performed successfully if:
  - *DEC_Status.all_one* = "0" AND
  - *DEC_Status.all_zero* = "0" AND
  - if not all detected byte errors could be detected (*DEC_Status.corr_count* = 0xF)

DEC_Status.correction remains set until it is reset by the SW by writing DEC_Status.correction = "1". This behavior is required to be able to detect in a stream of 2 KB, 4 KB or 16 KB data whether an error exists and to not have to check the status bits in DEC_Status after each decoding and correction procedure (max. 254 byte). For this reason only the highest number of corrected errors is entered in *DEC_Status.corr_count*. This happens continuously until the user deletes *DEC_Status.corr_count*, meaning ≠ 0x0 is written.

Since deleted storage areas are also accessed, *DEC_Status.all_one* and *DEC_Status.all_zero* are available.

If all data bytes and check bytes are equal to the value 0xFF in a deleted storage area, one correction process is always performed as part of the decoding procedure. *DEC_Status.all_one* = "1" is set so that a user can see that a deleted storage area was accessed. In this case, the user must discard the data available in the RSC-RAM. A byte to be decoded ≠ 0xFF sets *DEC_Status.all_one* = "0".

Note

If a user wants to decode data with 0xFF, the check bytes are always ≠ 0xFF.

If all data bytes and check bytes are equal to the value 0x00 in a deleted memory area, *DEC_Status.all_zero* = "1" is set. A byte to be decoded ≠ 0x00 sets *DEC_Status.all_zero* = "0".

DEC_Status.all_one and *DEC_Status.all_zero* are set at every start of the decoder and correction procedure and deleted in case of a detected decoding byte ≠ 0xFF or ≠ 0x00.

Copyright © Siemens AG 2023. All rights reserved

2.3.9.3.2.3 RSC interrupt

The interrupts of the RS controller (see chapter 2.3.9.3.2.1 and 2.3.9.3.2.2) are summarized per interrupt event in a total register 'IRQstatus'.

If an interrupt event is generated, the FW can determine which interrupt event was the cause by reading 'IRQstatus'. An interrupt event only leads to the triggering of rsc_int_o if the respective interrupt event in the 'IRQmask' register is not masked.

With each enabled interrupt event that is not already set in 'IRQstatus', rsc_int_o is triggered for an impulse length that can be set (high-pulse, can be set via a VHDL constant: 1 - 16 system cycles). The activation of an interrupt event already set in 'IRQstatus' or an interrupt event already locked via 'IRQmask' is not visible at rsc_int_o and is therefore ignored.

The register entry is deleted by writing a "1" to the respective interrupt event in 'IRQstatus'.

Writing a '1' to the interrupt event in 'IRQstatus' deletes the register entry.

In addition, the RS controller makes the following interrupt events available as individual interrupts.

Table 20: RSC interrupt events as individual interrupts

Interrupt event		Description
Access_Err_IRQ	high-pulse *)	Error during access within the RSC address range
int_encoder_finished	high-pulse *)	Encoding procedure incl. writing of the check bytes is completed
int_decoder_finished_good	high-pulse *)	Decoding procedure has been successfully completed without correction of the data
int_decoder_finished_corr	high-pulse *)	Decoding procedure has been successfully completed with at least one correction of the data
int_decoder_finished_bad	high-pulse *)	Decoding procedure has **not** been successfully completed because of too many byte errors

*) The length of the interrupt event can be set via a VHDL constant: 1 – 16 system cycles.

2.3.9.3.2.4 Access error

Accesses via the AHB that address a non-defined address range within the RS controller or in case of impermissible AHB size:

- return the read data 0x0000 at READ
- are ignored at WRITE.

At AHB an AHB-Error Response is effected in the case of such RD accesses, write accesses are always acknowledged with OK-Response, due to the Posted behavior of the AHB Slave modules. In addition, the first faulty write or read access in the 'Access_Error' register is stored and reported as own output signal of the RS controller via the interrupt access_err_irq_o.

As long as Access_Error has not been enabled for a subsequent faulty access by the FW, meaning *Error_Lock* = "1", all the subsequent accesses are processed without storage and interrupt message. Writing a "0" to Bit *Error_Lock* only resets *Error_Lock* = "0" in the 'Access_Error' register and thus releases it again. The content of the remaining register fields remains.

An entry in the 'Access_Error' register is generated at *Error_Lock* = "0" when:

- Write accesses are only to readable registers
- Read accesses are only to readable address ranges (register)
- Write and read accesses are in the address gaps of the register and RAM address range
- Byte and halfword accesses are to wordwise writeable addresses.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

In addition the access_err_irq_o interrupt is triggered.

Note

Implementation note:

The length of the interrupt pulse can be set via a VHDL constant: 1 – 16 system cycles.

2.3.9.4 Address mapping

Start-Address	End-Address	Modul/Memory-Name
100h	1FFh	RSC_RAM
0h	1FFh	rs_ip

Module	Register/Memory	Read	Write	Access	Address
/rs_ip				all	
	/ENC_DSTaddr	rh	w	all	0h
	/ENC_DSTmode	r	w	all	4h
	/ENC_Length	rh	w	all	8h
	/ENC_Status	rh		all	Ch
	/DEC_SRCaddr	rh	w	all	10h
	/DEC_SRCmode	r	w	all	14h
	/DEC_Length	rh	w	all	18h
	/DEC_Status	rh	w(t)	all	1Ch
	/IRQmask	r	w	all	20h
	/IRQstatus	rh	wt	all	24h
	/Access_Error	r(h)	(w)	all	2Ch
	/IP_Development	r		all	30h
	/IP_Version	r		all	34h
	/ENC_Data		wt	all	80h
	RSC_RAM	r	w	all	100h-1FFh

2.3.9.5 Register description

Module: rs_ip

Register:	ENC_DSTaddr				Address:	0h	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	w
Description:	Destination address						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	addr	0h	rh	w	Address output over the AHB master - Halfword access: A(0) = '0' - Word access: A(1:0) = "00"		

Register:	ENC_DSTmode				Address:	4h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Mode register for						
Bit	Identifier	Reset	Attr.		Function / Description		
0	incr_addr	0h	r	w	Destination address 0 = Hold 1 = Increment		
1	bypass	0h	r	w	0: use AHB master interface 1: use RSC RAM For writing data incl. check bytes		
3dt2	access_size	0h	r	w	Access width 0: Byte (incr Addr +1) 1: Halfword (incr Addr +2) 2: Word (incr Addr +4)		
31dt4	reserved	0000000h					

Register:	ENC_Length				Address:	8h	
Bits:	31dt0	Reset value:		EEh	Attributes:	rh	w
Description:	Mode settings of encoder						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	length	eeh	rh	w	Number of data bytes, not including check bytes. Length must be within a range of between 238 and 1. - Length = 0 becomes 1 (0x01) - Length > 238 becomes 238 (0xEE)		
31dt8	reserved	000000h					

Register:	ENC_Status				Address:	Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	rh	
Description:	Status of encoder						
Bit	Identifier	Reset	Attr.		Function / Description		
0	run	0h	rh		Run / idle 0: Encoder is IDLE and ready to receive new data 1: Encoder is in operation		

31dt1	reserved	00000000h			

Register:	DEC_SRCaddr				Address:	10h	
Bits:	31dt0	Reset value:		0h	Attributes:	rh	w
Description:	Source address read by the data						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	addr	0h	rh	w	Address output over the AHB master - Halfword access: A(0) = '0' - Word access: A(1:0) = "00"		

Register:	DEC_SRCmode				Address:	14h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Mode register for						
Bit	Identifier	Reset	Attr.		Function / Description		
0	incr_addr	0h	r	w	Source address 0 = Hold 1 = Increment		
1	bypass	0h	r	w	0: use AHB master interface 1: use RSC-RAM For reading data incl. check bytes		
3dt2	access_size	0h	r	w	Access width 0: Byte (incr Addr +1) 1: Halfword (incr Addr +2) 2: Word (incr Addr +4)		
31dt4	reserved	0000000h					

Register:	DEC_Length				Address:	18h	
Bits:	31dt0	Reset value:		EEh	Attributes:	rh	w
Description:	Decoder mode settings						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	length	eeh	rh	w	Number of data bytes, not including check bytes. Length must be within a range of between 238 and 1. - Length = 0 becomes 1 (0x01) - Length > 238 becomes 238 (0xEE)		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

31dt8	reserved	000000h			

Register:	DEC_Status				Address:	1Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	rh	w(t)
Description:	Status of last decoding operation						
Bit	Identifier	Reset	Attr.		Function / Description		
0	run	0h	rh	wt	0: Decoder is IDLE (waiting for new data) 1: Decoder still running Writing 1 starts the decoder; the RSC clears this bit upon completion		
3dt1	reserved	0h					
7dt4	corr_count	0h	rh	w	Number of corrected symbols 0: No correction 1: 1 data byte corrected ... f: Cannot be corrected Updated after each decoding operation, but only if the current encoding has detected more errors than are already entered in the register. Must be reset by the firmware (by writing a value /= 0).		
8	correction	0h	rh	wt	Error while decoding data 0: No data correction required 1: Data correction performed. See number of corrected bytes in corr_count. If corr_count = 0xf, data could not be restored. Is not reset by the hardware if there are encoding errors - must be cleared by the firmware. (Detection of errors in a long stream) Writing a 1 resets the error.		
15dt9	reserved	00h					
16	all_one	0h	rh	w	If all data and check bytes have been read as 0xff. 1: All bytes read were 0xff. 0: At least 1 byte read was not equal to 0xff. Please **note**, decoding still takes place; corr_count and data must not be evaluated.		

17	all_zero	0h	rh	w	If all data and check bytes have been read as 0x00. 1: All bytes read were 0x00. 0: At least 1 byte read was not equal to 0x00. Please **note**, decoding still takes place; corr_count and data must not be evaluated.
30dt18	reserved	0000h			
31	response_err	0h	rh	wt	In decoder AHB RD access, 0: No error occurred (OK response) 1: An error occurred (error response) Writing a 1 resets the error

Register:	IRQmask				Address:	20h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Interrupt mask register						
Bit	Identifier	Reset	Attr.		Function / Description		
0	encoder_finished	0h	r	w	Mask for enabling encoder_finished for triggering the group interrupt 0: No interrupt 1: Interrupt enable		
3dt1	reserved	0h					
4	decoder_finished_good	0h	r	w	Mask for enabling decoder_finished_good for triggering the group interrupt 0: No interrupt 1: Interrupt enable		
5	decoder_finished_bad	0h	r	w	Mask for enabling decoder_finished_bad for triggering the group interrupt 0: No interrupt 1: Interrupt enable		
6	decoder_finished_corr	0h	r	w	Mask for enabling decoder_finished_corr for triggering the group interrupt 0: No interrupt 1: Interrupt enable		
7	decoder_response_err	0h	r	w	Mask for enabling response_err for triggering the group interrupt 0: No interrupt 1: Interrupt enable		
31dt8	reserved	000000h					

Register:	IRQstatus	Address:	24h

Bits:	31dt0	Reset value:	0h	Attributes:	rh	wt
Description:	Interrupt group register. If a new bit is set in this register and the corresponding bit in IRQmask is enabled, a group interrupt is triggered (pulse length can be set with VHDL constant).					

Bit	Identifier	Reset	Attr.		Function / Description
0	encoder_finished	0h	rh	wt	Display 1 if an encoder operation has been completed. If the enable flag is set in the IRQmask register, an interrupt is triggered. This bit is set by the hardware and can be reset by the firmware writing a 1.
3dt1	reserved	0h			
4	decoder_finis-hed_good	0h	rh	wt	Display 1 if the decoder operation has been completed without errors. If the enable flag is set in the IRQmask register, an interrupt is triggered. This bit is set by the hardware and can be reset by the firmware writing a 1.
5	decoder_finis-hed_bad	0h	rh	wt	Display 1 if the decoder operation is complete and the decoder has detected at least 1 erroneous byte. If the enable flag is set in the IRQmask register, an interrupt is triggered. This bit is set by the hardware and can be reset by the firmware writing a 1.
6	decoder_finis-hed_corr	0h	rh	wt	Display 1 if the decoder operation is complete at least one error was detected and corrected. If the enable flag is set in the IRQmask register, an interrupt is triggered. This bit is set by the hardware and can be reset by the firmware writing a 1.
7	deco-der_response_err	0h	rh	wt	Display 1 if the decoder operation is complete and a con-figuration error has occurred. If the enable flag is set in the IRQmask register, an interrupt is triggered. This bit is set by the hardware and can be reset by the firmware writing a 1.
31dt8	reserved	000000h			

Regis-ter:	Access_Error			Address:	2Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r(h)	(w)
Description:	Access Error Register					

Bit	Identifier	Reset	Attr.		Function / Description
8dt0	ahb_address	0h	rh		AHB address in RS IP where the error occurred
26dt9	reserved	00000h			
27	ahb_apb_select	0h	rh		0 = AHB-Access, 1 = APB_access (not relevant)
29dt28	ahb_size	0h	rh		00 = Byte access (8 bit), 01 = HW access (16 bit),

					10 = W access (32 bit), 11 = reserved
30	ahb_rd_wr	0h	rh		0 = RD access, 1 = WR access
31	err_lock	0h	r	w	Set by HW to 1 on access error, locked for further entries until cleared by SW (write 0 to clear) 0 = no access error 1 = access error occurred

Register:	IP_Development				Address:	30h	
Bits:	31dt0	Reset value:		7E01014h	Attributes:	r	
Description:	IP Development register						
Bit	Identifier	Reset	Attr.		Function / Description		
10dt0	r_label	14h	r		Revision Label		
15dt11	increment	2h	r		Increment		
18dt16	patch_label	0h	r		Patch-Label, if ASIC, else 0		
20dt19	target_play	0h	r		ASIC 00- FPGA 01		
31dt21	ident	3fh	r		Name: RS_Controller Description: Reed-Solomon encoder IP Identifier: 0x3F		

Register:	IP_Version				Address:	34h	
Bits:	31dt0	Reset value:		10100h	Attributes:	r	
Description:	IP Version register						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	Debug_Version	0h	r		incremeted for error correction (e.g. Metal-Fix)		
15dt8	Version	1h	r		version counter of RS IP: 0x01: initial		
31dt16	Configuration	1h	r		IP configuration: 0x0000: ERTEC200P Step 2 (AHB) 0x0001: HOF (AHB)		

Register:	ENC_Data				Address:	80h	
Bits:	31dt0	Reset value:		0h	Attributes:		wt
Description:	Encoder data						

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	data	0h		wt	Access to encoder (write-only)

Me-mory:	RSC_RAM			Startaddress:	100h	
Bits:	31dt0	Size:	100h	Attributes:	r	w
Description:	In RSC_RAM the encoded data of the encoder or the data to be decoded by the decoder is stored.					

Copyright © Siemens AG 2023. All rights reserved

2.3.10 Host Interface – Parallel (XHIF)

Parallel external host access is processed over the host interface.

A local bus unit (2x XHIF-IP) is implemented for access to a host system connected in parallel. The local bus unit (XHIF) has two internal XHIF-IPs and supports access from an external 16-bit and 32-bit host CPU. The complete ERTEC 200P address range can be accessed by the external host system over the host interface (over a max. of 8 configurable address windows (pages) of 256 bytes each – 1 / 2 MByte (see below)).

The host interface is configured in the "Parameter" module with APB access. The basic configuration (data width, ready polarity, read/write line) for the XHIF is set (jointly for the two IP XHIF_0/XHIF_1) with a PowerOn reset over HW_ConfigPins 5...3. Subsequent changes can be made to this configuration by the ARM926EJ-S in the 'XHIF_CONTROL' register over the APB interface. The external host must not access this register, as this could cause undefined states at the host interface. The basic configuration therefore cannot subsequently be changed by the external host.

2.3.10.1 Block diagram of the Host Interface

Figure 27: Block diagram of the host interface

The figure below shows the structure of the XHIF, which consists of two individual XHIF modules (XHIF_0/1) (2 × 4 pages of 2 MByte each).

The XHIF module (XHIF_0 or XHIF_1) is selected with the XHIF_SEG_2 pin. The XHIF interface is switched to the selected XHIF module. Access to the AHB interface is with arbitration. The pins XHIF_SEG_0 and XHIF_SEG_1 address the corresponding pages of an XHIF module.

The XHIF configuration (XHIF_ACC_MODE, XHIF_POL_RDY, XHIF_CPU_WIDTH) is the same for both XHIF modules and is connected straight to both modules (source: ConfigPins or 'XHIF_CONTROL' register). The XHIF interrupt (XHIFerr_IRQ) is connected from each module with an OR logic operation and connected externally (to the event unit in PerIF).

Figure 28: Structure of the parallel XHIF with 8 pages

2.3.10.2 Function description (XHIF)

XHIF0, 1 is an interface module that enables an external CPU to access the AHB bus as master over a memory interface. The basic features of the interface module are as follows:

- 32/16 and 8-bit CPUs can be connected with an address range (A0 to A22) of 10 to 23 address bits (1 k to 8 M) (**note** however: only 16/32-bit CPUs are supported in the ERTEC 200P). Addresses A21 (XHIF_SEG0) and A22 (XHIF_SEG1) are used to select the 4 possible pages with an address range of 256 bytes to 2 MByte.
- The interface module can be configured by the external CPU over a 16-bit or 32-bit data bus.

- The interface module can also always be configured with the internal ARM926 APB bus.
- XHIF interface timing is asynchronous to the AHB bus clock.
- Areas in the AHB address range (pages) can be configured for access by the external CPU (XHIF internal paging register). ERTEC 200P address mapping (see chapter 2.4.1) is designed to ensure that all relevant AHB address ranges can be configured over the 8 pages (8 × 2 MByte) during ERTEC 200P operation.

  Following is an example of possible page configuration.

Table 21: Possible page configuration of XHIF module

	XHIF address range	AHB module	AHB address range
XHIF0	Page 0	APB peripherals	1 MByte
	Page 1	PROFINET IP	2 MByte
	Page 2	Peripheral interface	1 MByte
	Page 3	GDMA	1 MByte
XHIF1	Page 4	Reserved	2 MByte
	Page 5	SDRAM (EMC)	2 MByte
	Page 6	D-TCM (ARM926)	2 MByte
	Page 7	SRAM (EMC)	2 MByte

- The AHB master interface is based on the AHB standard 2.0 AHB-Lite protocol.
- The XHIF module has only one clock source, which must correspond to the frequency of the AHB/APB system connected.
- Single read and write commands are supported, but burst mode is not.
- The XHIF module only supports little-endian mode (least significant byte at the lower memory address).

The address windows in the page registers (offset, range and buffer mode) can be configured by the ARM926 over the APB provided the XHIF is not yet active (enabled). The external host uses the register chip select XHIF_XCS_R for configuring the page registers.

The external host can also configure the page registers over the APB interface of the XHIF. In this case, however, the page for the APB address range must not be changed, as further changes to the page registers over APB will otherwise not be possible.

Information for page configuration

- The page size (range) can be configured between 256 Byte and 2 Mbyte.
- The page base address (offset) is always a multiple of page size (range).

Two functions have been implemented with the 'XHIF_XCS_R_A20' input pin in the light of restrictions on the ballout and package design of the ERTEC 200P. They allow the signal at this input to be used:

- as chip select for page configuration over the external host
  OR
- as address line A20 (→ page address range increased to 2 MByte)

Just how this input pin is used in the XHIF modules is set in the SCRB register 'XHIF_Mode' with the 'XHIF_Mode' bit:

- XHIF_Mode = 0b: Page register chip select XHIF_XCS_R
  → Page = max. 1 MByte

- XHIF_Mode = 1b: Address line XHIF_A20
  → Page = max. 2 MByte

There are therefore 2 use cases with different max. page sizes.

8 pages with a 1 MByte address range each

This is the default setting (XHIF_Mode = '0'). The external host and its address lines are wired to the XHIF interface as follows:

Table 22: XHIF pages with 1 MByte address range each

XHIF interface (ERTEC 200P)	Ext. host (ext. configuration)
XHIF_A1	A1 with 16-bit interface fixed at '0' with 32-bit interface
XHIF_A2 ... XHIF_A19	A2...A19
XHIF_XCS_R_A20	XCS_R
XHIF_SEG_0	A20
XHIF_SEG_1	A21
XHIF_SEG_2	A22

The 8 address windows (1 MByte each) are generally assigned as follows: 2x PN-IP, 1x PerIF, 2x EMC-SDRAM, 1x GDMA, 1x dyn. (to ARM926 D-TCM or EMC-SRAM or APB peripherals or EMC register).

8 pages with a 2 MBytes address range each

(XHIF_Mode = '1') must be set for this setting. The external host and its address lines are wired to the XHIF interface as follows:

Table 23: XHIF pages with 2 MBytes address range each

XHIF interface (ERTEC 200P)	Ext. host (ext. configuration)
XHIF_A1	A1 with 16-bit interface, fixed at '0' with 32-bit interface
XHIF_A2 ... XHIF_A19	A2 ... A19
XHIF_XCS_R_A20	A20
XHIF_SEG_0	A21
XHIF_SEG_1	A22
XHIF_SEG_2	A23

The procedure for XHIF configuration is as follows:

- The XHIF interface with 8 pages of 1 MByte each and page register configuration over the external host (XHIF_XCS_R) is available after a ERTEC 200P reset.
- Page register configuration from the host is with A20 = '0' (XHIF_XCS_R = '0').
- The page from which the APB area can be accessed is accessed from the host with A20 = '1' (XHIF_XCS_R = '1') and XHIF_XCS_M = '0'. The register 'XHIF_Mode' = '1' is set in the SCRB block.
- From now on, A20 acts from the host perspective as an address line (XHIF_A20) and 2 MByte of address range are therefore available for each page.

The XHIF interface can be operated at either 3.3 V or 1.8 V. The IO pads are connected separately with VDD_XHIF (either 1.8 V or 3.3 VM; no other setting possible) and supplied with GND.

Copyright © Siemens AG 2023. All rights reserved

This supply system includes all 64 IO pads of the host interface. One use case for 1.8 V is the connection of two ERTEC 200P. The ERTEC 200P EMC interface is always operated at VDD=1.8 V. The connection must therefore also be 1.8 V. The driver capacity of the pad cell is 25 pF with 1.8 V and 35 pF with 3.3 V.

The driver power of the 1.8 V XHIF pads in the ERTEC 200P is set to 4 mA after power on. An external host processor should increase this to 9 mA for the XHIF pads (configuration in 'DRIVE95-32GPIO' registers, see chapter 2.3.14.11.27.1) before its first read access to the XHIF interface. 4 mA is enough for the XHIF_XRDY pin as it is wired straight from the host to the input pin and only needs to recharge this one load.

2.3.10.2.1 AHB Master interface

The 32-bit data bus supports the AHB-Lite protocol. Erroneous access is indicated by an interrupt (AHBerror_INT). The lock functionality is not used. Burst access is not supported; only single access over the host interface is permitted.
(The following AHB signals are not used, but set static to HLOCK = "0", HPROT = "0001" (i.e., DATA_ACCESS), HBURST = "000").

2.3.10.2.2 APB Slave interface

The HostIF module contains an APB slave interface with a 32-bit data bus and APB protocol. The 8 pages and the serial interface can be configured over this slave interface. XHIF configuration parameter assignment is also possible in the 'XHIF_CONTROL' HostIF register.

2.3.10.2.3 XHIF configuration

The XHIF configurations (data width, ready polarity and read/write line) for the XHIF-IP are set with the ConfigPins(5:3) and latched to *SCRB_CONFIG_REG* when the PowerOn reset is cleared. ARM926 can also assign parameters for XHIF configuration by changing the XHIF configurations (data width, ready polarity or read/write line) in *SCRB_CONFIG_REG* over the APB interface. The external host must not access *SCRB_CONFIG_REG*, as this could cause undefined states at the host interface.

Copyright © Siemens AG 2023. All rights reserved

Table 24: XHIF interface adjustment

ConfigPin(5) = A[22] = XHIF_CPU_WIDTH(0)	ConfigPin(4) = A[21] = **not** XHIF_POL_RDY	ConfigPin(3) = A[20] = **not** XHIF_ACC_MODE	**XHIF interface**
0	0	0	16-bit, XRDY high-active, XWR only
0	0	1	16-bit, XRDY high-active, RD-WR separate
0	1	0	16-bit, XRDY low-active, XWR only
0	1	1	16-bit, XRDY low-active, RD-WR separate
1	0	0	32-bit, XRDY high-active, XWR only
1	0	1	32-bit, XRDY high-active, RD-WR separate
1	1	0	32-bit, XRDY low-active, XWR only
1	1	1	32-bit, XRDY low-active, RD-WR separate

Note

XHIF_CPU_WIDTH(1) is fixed at '0', as an external host with an 8-bit data interface is not supported.

Description of configuration pins

- XHIF_POL_RDY
  - 0: The XHIF module signals to the host that access with XHIF_XRDY = 0 is complete (ready).
  - 1: The XHIF module signals to the host that access with XHIF_XRDY = 1 is complete (ready).
- XHIF_ACC_MODE
  - 0: Separate read/write lines are provided
  - 1: A combined read/write signal is used.
- XHIF_CPU_WIDTH
  - 00: 16-bit data bus of the external host
  - 01: 32-bit data bus
  - 10: 8-bit data bus
  - 11: Not permitted

The configuration signals are synchronized with the XHIF clock.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.10.2.4 Application information

- The XHIF configurations (16/32-bit data width, ready polarity and read/write line) for the XHIF-IP are set with a PowerOn reset over ConfigPins 5...3. Subsequent changes can be made to these configurations by the ARM926EJ-S in the 'XHIF_CONTROL' HostIF register over the APB interface.
- 'XHIF_CONTROL' must not be changed by the external host. According to the XHIF-IP specifications, the external host must not address the configuration registers over the APB interface (AHB/APB bridge).
- The external host can only address the internal XHIF-IP page registers (offset, range and buffer mode) and XHIF_VERSION over the register chip select XHIF_XCS_R. Only the lowest-value address bits are considered for register selection (address ≥ 0x40h are invalid according to the XHIF-IP spec).

  Only halfword write/read access to the registers is permitted with a configured data width of 16 bits, and only word write/read access with a configured data width of 32 bits. According to the XHIF-IP spec, byte-specific access is not permitted.
- When the memory chip select XHIF_XCS_M is active, write/read access is forwarded straight to the HOSTIF AHB master interface and all aligned access types, byte, halfword and word, are permitted.

  If you configure a 32-bit data width, the XHIF_ADR(1) pin must be clamped to '0' for the ERTEC 200P (XHIF_ADR(0) internal static setting '0') so that there are always word addresses internally in the XHIF-IP.

  If a data width of 16 bits is configured, the XHIF_ADR(1) pin is required for halfword addressing and is driven by the external host.
- Initialization procedure for XHIF-IP after reset according to IP spec:
  - Power on reset cleared
  - XHIF configuration with ConfigPins 5...3, thereafter only the ARM926EJ-S can change configuration (access by the external host not yet permitted)
  - Page setting by ARM926EJ-S: access over the APB to the relevant registers or page setting by host: access over the XHIF_XCS_R chip select to the relevant registers
  - Write access (HOSTIF-AHB master) by the external host
  - The ARM926EJ-S must not change the XHIF configuration (data width, ready polarity or command mode) once the external host has started write access.
- The XHIF_XRDY signal is implemented as a bidi buffer. However, the output driver only becomes active with an XHIF CS, i.e., other devices can actively pull the signal when there is no access with XHIF. This basically provides open drain functionality.
- XHIF_XRDY must be pulled to "normally ready" with an external pull resistor, i.e., pull-up if XHIF_XRDY = high-active and pull-down if XHIF_XRDY = low-active, so that at the end of access the XHIF_XRDY signal can also be recognized by an external host operating with a lower clock rate than the ERTEC 200P (background: XHIF_XRDY is only actively driven by the XHIF module for **one clock cycle**).

  The XHIF_XRDY signal sends a ready message to the external XHIF host. XHIF_XRDY is implemented with a tristate driver whose Output_Enable does not become active until the start of access (as soon as CS and RD or WR are active) and becomes inactive again at the end of access (one clock cycle after active XHIF_XRDY).

Copyright © Siemens AG 2023. All rights reserved

Note

Implementation note:

XHIF_XRDY is a controlled **push-pull** output which requires an external pull resistor in accordance with its polarity (i.e., pull-down if XHIF_XRDY is low-active and vice versa). The pull resistor ensures that XHIF_XRDY is pending at the external host for long enough.

- When configuring the external host, please **note** that XHIF_XRDY briefly appears to the host as active at the start of access because of the pull resistor (it takes up to 11 ns for the pin to be driven by ERTEC 200P). To prevent the external host from interpreting access as acknowledged at this stage, there must be a delay before it evaluates XHIF_XRDY. When the external host is an ERTEC 200P, this delay would for example be implemented by configuring the appropriate number of "Read/Write Strobe Cycles" in the EMC interface (R_Strobe or W_Strobe parameter in the 'ASYNC_BANK0-3' EMC registers).
- How does the host detect that the ERTEC 200P is ready for host access after the reset phase?

  The host recognizes that the ERTEC 200P is ready for host access with the CS_R Page 0 range and offset registers. Page 0 range and offset registers have the default value 0x0000_0000; after configuration by ERTEC 200P, the registers of page 0 must contain values of ≠ 0x0000_0000.

  This means that the host must poll these registers in the CS-R area after an ERTEC 200P reset until the values initialized by the ERTEC 200P are read; the host is then able to recognize that the ERTEC 200P is ready for host access.
- Can deadlock occur during host access?

  Problems can only occur during the active reset phase of the ERTEC 200P. They may for example be triggered by a:

  - Software reset

  - Watchdog reset

  If the host accesses the CS_M area during the reset phase, access by ERTEC 200P will not be completed with READY. Access by the host to CS_R is completed during the reset phase, in other words there are no problems there.

  Please **note** the following points to detect/avoid the problem:

  - The host must detect an ERTEC 200P reset (at the host, use a GPIO that signals the reset state) and block access to CS_M.
  - The XHIF master must force READY, if READY is not output (i.e., cancel access) and initiate an error signal (interrupt).

Copyright © Siemens AG 2023. All rights reserved

2.3.10.3 Address mapping

Start_Address	End_Address	Modul/Memory_Name
0h	FCh	HOSTIF

Module	Register/Memory	Read	Write	AHB-Master Address	Ext. Host XHIF_XCS_R Address [1)]
/HOSTIF					
	HOST_CONTROL	r	w	0h	not accessible
	reserved	r	w	4h	
	reserved	r	w	8h	
	reserved	rh		Ch	
	reserved	rh		10h	
	reserved	rh		14h	
	reserved	r	w	18h	
	reserved	r	w	1Ch	
	reserved	r	w	20h	
	IP_VERSION	r		40h	
	IP_DEVELOPMENT	r		44h	
	ACCESS_ERROR	r	(w)	48h	
	XHIF_CONTROL	rh	w	70h	
	XHIF_0_P0_RG	r	(w)	80h	0h
	XHIF_0_P0_OF	r	(w)	84h	4h
	XHIF_0_P0_CFG	r	w	88h	8h
	XHIF_0_P1_RG	r	(w)	90h	10h
	XHIF_0_P1_OF	r	(w)	94h	14h
	XHIF_0_P1_CFG	r	w	98h	18h
	XHIF_0_P2_RG	r	(w)	A0h	20h
	XHIF_0_P2_OF	r	(w)	A4h	24h
	XHIF_0_P2_CFG	r	w	A8h	28h
	XHIF_0_P3_RG	r	(w)	B0h	30h
	XHIF_0_P3_OF	r	(w)	B4h	34h
	XHIF_0_P3_CFG	r	w	B8h	38h
	XHIF_0_VERSION	r		BCh	3Ch
	XHIF_1_P0_RG	r	(w)	C0h	40h
	XHIF_1_P0_OF	r	(w)	C4h	44h
	XHIF_1_P0_CFG	r	w	C8h	48h
	XHIF_1_P1_RG	r	(w)	D0h	50h
	XHIF_1_P1_OF	r	(w)	D4h	54h
	XHIF_1_P1_CFG	r	w	D8h	58h
	XHIF_1_P2_RG	r	(w)	E0h	60h
	XHIF_1_P2_OF	r	(w)	E4h	64h
	XHIF_1_P2_CFG	r	w	E8h	68h
	XHIF_1_P3_RG	r	(w)	F0h	70h
	XHIF_1_P3_OF	r	(w)	F4h	74h
	XHIF_1_P3_CFG	r	w	F8h	78h
	XHIF_1_VERSION	r		FCh	7Ch

[1)] The addresses also apply to access from an external host with a 32-bit data bus at the XHIF interface with *XHIF_XCS_R* (XHIF chip select register: offset address = 0x0000). An external host with a 16-bit data bus addresses on a 16-bit basis (address: 0h, 2h, 4h, 6h, …).

2.3.10.4 Register description

Module: HOSTIF

Register:	HOST_CONTROL				Address:	0h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Selection of the serial or parallel path						
Bit	Identifier	Reset	Attr.		Function / Description		
0	CONNECT_MODE	0h	r	w	0=parallel connection (XHIF) 1=serial connection (SPI)		
31dt1	reserved	00000000h					

Register:	IP_VERSION				Address:	40h	
Bits:	31dt0	Reset value:	100h		Attributes:	r	
Description:	Metal-fix register for IP Version						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	DEBUG_VERSION	0h	r		IP Debug Version		
15dt8	VERSION	1h	r		IP Version		
31dt16	IP_CONFIGURA-TION	0h	r		IP Configuration		

Register:	IP_DEVELOPMENT				Address:	44h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	
Description:	Metal-fix register for IP Development						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	LABEL_STRUC-TURE	0h	r		Design Label		

Register:	ACCESS_ERROR				Address:	48h	
Bits:	31dt0	Reset value:	20000000h		Attributes:	(r)	(w)
Description:	Access error register						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	APB_ADDRESS	0h	r		Erroneous APB address in the HostIF		
27dt8	reserved	00000h					
29dt28	APB_SIZE	2h	r		"10": Word access (32-bit), fixed value as only word access possible		
30	APB_RD_WR	0h	r		'0': RD-access '1': WR-access		

31	ERROR_LOCK	0h	r	w	Set to '1' by the HW if erroneous APB access is detected. Further HW entries are then blocked. The SW must reset the bit to '0' to enable new entries.

Register:	XHIF_CONTROL				Address:	70h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	XHIF interface settings						
Bit	Identifier	Reset	Attr.		Function / Description		
0	XHIF_ACC_MODE	0h	r	w	XHIF Handshake protocol: 0=Intel Mode, 1=Motorola Mode		
1	XHIF_POL_RDY	0h	r	w	'0': XHIF_XRDY ist low_aktiv ‘1': XHIF_XRDY ist high_aktiv		
3dt2	XHIF_CPU_WIDTH	0h	r	w	Data bus width: 00=8b, 01=16b, 10=32b		
31dt4	reserved	0000000h					

Register:	XHIF_0_P0_RG				Address:	80h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:	Range value of the page #0						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	ROSL2	0h	r		Read only value = 0		
21dt8	RW	0h	r	w			
31dt22	ROSL1	0h	r		Read only value = 0		

Register:	XHIF_0_P0_OF				Address:	84h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:	Offset value of the page #0						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	RO	0h	r		Read only value = 0		
31dt8	RW	0h	r	w			

Register:	XHIF_0_P0_CFG				Address:	88h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Configuration of the buffering mode for page #0						
Bit	Identifier	Reset	Attr.		Function / Description		
0	BUFMOD	0h	r	w	16bit access mode of page #0		

31dt1	reserved	00000000h			

Register:	XHIF_0_P1_RG				Address:	90h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Range value of the page #1						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	ROSL2	0h	r		Read only value = 0		
21dt8	RW	0h	r	w			
31dt22	ROSL1	0h	r		Read only value = 0		

Register:	XHIF_0_P1_OF				Address:	94h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Offset value of the page #1						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	RO	0h	r		Read only value = 0		
31dt8	RW	0h	r	w			

Register:	XHIF_0_P1_CFG				Address:	98h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Configuration of the buffering mode for page #1						
Bit	Identifier	Reset	Attr.		Function / Description		
0	BUFMOD	0h	r	w	16bit access mode of page #1		
31dt1	reserved	00000000h					

Register:	XHIF_0_P2_RG				Address:	A0h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Range value of the page #2						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	ROSL2	0h	r		Read only value = 0		
21dt8	RW	0h	r	w			
31dt22	ROSL1	0h	r		Read only value = 0		

Register:	XHIF_0_P2_OF				Address:	A4h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)

Description:	Offset value of the page #2				
Bit	Identifier	Reset	Attr.		Function / Description
7dt0	RO	0h	r		Read only value = 0
31dt8	RW	0h	r	w	

Register:	XHIF_0_P2_CFG				Address:	A8h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Configuration of the buffering mode for page #2						
Bit	Identifier	Reset	Attr.		Function / Description		
0	BUFMOD	0h	r	w	16bit access mode of page #2		
31dt1	reserved	00000000h					

Register:	XHIF_0_P3_RG				Address:	B0h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:	Range value of the page #3						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	ROSL2	0h	r		Read only value = 0		
21dt8	RW	0h	r	w			
31dt22	ROSL1	0h	r		Read only value = 0		

Register:	XHIF_0_P3_OF				Address:	B4h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:	Offset value of the page #3						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	RO	0h	r		Read only value = 0		
31dt8	RW	0h	r	w			

Register:	XHIF_0_P3_CFG				Address:	B8h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Configuration of the buffering mode for page #3						
Bit	Identifier	Reset	Attr.		Function / Description		
0	BUFMOD	0h	r	w	16bit access mode of page #3		
31dt1	reserved	00000000h					

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Register:	XHIF_0_VERSION				Address:	BCh	
Bits:	31dt0	Reset value:		4000804h	Attributes:	r	
Description:	Metal-fix register for XHIF Version						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	VERSION	04000804h	r		Metal-fix register for XHIF Version		

Register:	XHIF_1_P0_RG				Address:	C0h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Range value of the page #0						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	ROSL2	0h	r		Read only value = 0		
21dt8	RW	0h	r	w			
31dt22	ROSL1	0h	r		Read only value = 0		

Register:	XHIF_1_P0_OF				Address:	C4h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Offset value of the page #0						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	RO	0h	r		Read only value = 0		
31dt8	RW	0h	r	w			

Register:	XHIF_1_P0_CFG				Address:	C8h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Configuration of the buffering mode for page #0						
Bit	Identifier	Reset	Attr.		Function / Description		
0	BUFMOD	0h	r	w			
31dt1	reserved	00000000h					

Register:	XHIF_1_P1_RG				Address:	D0h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Range value of the page #1						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	ROSL2	0h	r		Read only value = 0		
21dt8	RW	0h	r	w			

31dt22	ROSL1	0h	r		Read only value = 0

Register:	XHIF_1_P1_OF				Address:	D4h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:	Offset value of the page #1						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	RO	0h	r		Read only value = 0		
31dt8	RW	0h	r	w			

Register:	XHIF_1_P1_CFG				Address:	D8h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Configuration of the buffering mode for page #1						
Bit	Identifier	Reset	Attr.		Function / Description		
0	BUFMOD	0h	r	w	16bit access mode of page #1		
31dt1	reserved	00000000h					

Register:	XHIF_1_P2_RG				Address:	E0h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:	Range value of the page #2						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	ROSL2	0h	r		Read only value = 0		
21dt8	RW	0h	r	w			
31dt22	ROSL1	0h	r		Read only value = 0		

Register:	XHIF_1_P2_OF				Address:	E4h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	(w)
Description:	Offset value of the page #2						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	RO	0h	r		Read only value = 0		
31dt8	RW	0h	r	w			

Register:	XHIF_1_P2_CFG				Address:	E8h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Configuration of the buffering mode for page #2						

Bit	Identifier	Reset	Attr.		Function / Description
0	BUFMOD	0h	r	w	16bit access mode of page #2
31dt1	reserved	00000000h			

Regis-ter:	XHIF_1_P3_RG			Address:	F0h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	Range value of the page #3					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	ROSL2	0h	r		Read only value = 0
21dt8	RW	0h	r	w	
31dt22	ROSL1	0h	r		Read only value = 0

Regis-ter:	XHIF_1_P3_OF			Address:	F4h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	Offset value of the page #3					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	RO	0h	r		Read only value = 0
31dt8	RW	0h	r	w	

Regis-ter:	XHIF_1_P3_CFG			Address:	F8h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Configuration of the buffering mode for page #3					

Bit	Identifier	Reset	Attr.		Function / Description
0	BUFMOD	0h	r	w	16bit access mode of page #3
31dt1	reserved	00000000h			

Regis-ter:	XHIF_1_VERSION			Address:	FCh	
Bits:	31dt0	Reset value:	4000804h	Attributes:	r	
Description:	Metal-fix register for XHIF Version					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	VERSION	04000804h	r		Metal-fix register for XHIF Version

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.11 Peripheral Interface

Note

In this manual, we refer to "Consistency Control" and "Peripheral Interface" as synonyms.

The Consistency Control interface is operated as passive (i.e., as slave) at the multi-layer AHB and acts as the communication interface between applications via AHB slave 1 (e.g., external host CPU via host interface or the PN stack running internally ARM926-CPU) and internally PN communication via AHB slave 0 (PN-IP). The Consistency Control is used merely as the interface for consistent data management of the cyclic IO data.

ARM926 and external host interfaces access to Consistency Control are multiplexed (arbitrated), as only the internal ARM926 or an external host system requires access. The PN-IP has its own direct access to Consistence Control as the module always requires access to the IO data in the IO-RAM. As direct access to the IO data is possible from the PN-IP, there are no relevant lockout times for data transport. The IO data required can therefore be sent straight from or received directly by the peripheral interface without buffering at the local API (cut-through).

Consistence Control ensures the consistency of the IO data saved in the IO-RAM. Consistent storage between the PN-IP (communicative instance) and the ARM926 and host interfaces (applicative instances) is realized in the process.

The APB slave IF of the parameter module is used for application-specific configuration. Access to the entire memory area is possible over the APB for initializing the IO RAM. This requires both ready control and byte-by-byte access with byte enable at the APB.

The mechanisms required for two-way communication between the Peripheral Interface and external host CPU are implemented in the event unit. The event unit stores event bits which can be set both by internal moduls for the external host system and vice versa. These events cause the module to trigger an interrupt for the ARM926 interrupt controller (ARM-ICU, IRQ54, see chapter 2.3.5.9) or the external host system (XHIF_XIRQ, XHIF configuration, see chapter 2.3.14.10.5). The registers can be written and read with common AHB access. The event unit is configured with the parameter module and therefore the APB.

The two AHB slave interfaces (AHB slave0/1 modules) of the Consistency Control support both single access and AHB RD bursts of word size (32 bits) and with a burst length of INCR/INCR4/INCR8/INCR16 as burst access to the IO-RAM. All other types of AHB access (e.g., WR burst, burst size, ≠ word, ...) are mapped to the IO-RAM as single access.

The IO-RAM in Consistency Control has an EDC code (1-bit error, correctable; multible-bit error, detectable). If an error occurs when reading the IO-RAM, the cause of the error (CC-1B: 1-bit error corrected or CC-2B: 2-bit error detected) is saved in the SCRB.EDC_EVENT (see chapter 2.3.14.11.13) and the interrupt IRQ48 (EDC_Event) is triggered (see chapter 2.3.5.9).

The IO-RAM is initialized following a reset. The end of initialization (incl. EDC bits) is signaled in the SCRB.EDC_INIT_DONE (see chapter 2.3.14.11.13).

2.3.11.1 Interrupts for External Host

The summary interrupt CC_HOST_IRQ (XHIF_XIRQ, XHIF configuration, see chapter 2.3.14.10.5) will be generated through the following interrupt events.

Table 25: Interrupts for external host

IRQ No.	Interrupt name	Cause	Interrupt source
0	AHBERR_IRQ	AHB Error at Host-IF	Host-IF
1	XHIFERR_IRQ	XHIF Error at Host-IF	
2	SPIERR_IRQ	SPI Error at Host-IF	
3	reserved	-	-

Copyright © Siemens AG 2023. All rights reserved

IRQ No.	Interrupt name	Cause	Interrupt source
4	reserved	-	
5	reserved	-	
6	reserved	-	
7…15	reserved	-	
16	EXT_IRQ0	PN_IRQ3(0)	PN-IP
17	EXT_IRQ1	PN_IRQ3(1)	
18	EXT_IRQ2	PN_IRQ3(2)	
19	EXT_IRQ3	PN_IRQ3(3)	
20	EXT_IRQ4	PN_IRQ3(4)	
21	EXT_IRQ5	PN_IRQ3(5)	
22	EXT_IRQ6	PN_IRQ3(6)	
23	EXT_IRQ7	PN_IRQ3(7)	
24	EXT_IRQ8	PN_IRQ3(8)	
25	EXT_IRQ9	PN_IRQ3(9)	
26	EXT_IRQ10	PN_IRQ3(10)	
27	EXT_IRQ11	PN_IRQ3(11)	
28	EXT_IRQ12	PN_IRQ3(12)	
29	EXT_IRQ13	PN_IRQ3(13)	
30	EXT_IRQ14	PN_IRQ3(14)	
31	EXT_IRQ15	PN_IRQ3(15)	
63…32	reserved	-	-

2.3.11.2 Peripheral (Application) AHB interface

2.3.11.2.1 Address mapping

Start-Address	End-Address	Modul/Memory-Name
2000h	1BFFFh	IO_RAM
0h	1BFFFh	cons_control_app

Module	Register/Memory	Read	Write	Access	Address
/cons_control_app				all	
	/Command_IF_Control_APP	rh	w	all	0h
	/Command_IF_Status_APP	rh		all	4h
	/Host_IRQ_low	rh		all	8h
	/Host_IRQ_high	rh		all	Ch
	/Host_IRQmask_low	r	w	all	10h
	/Host_IRQmask_high	r	w	all	14h
	/Host_Event_low	rh	w	all	18h
	/Host_Event_high	rh	w	all	1Ch
	/Host_IRQack_low	rh	w	all	20h
	/Host_IRQack_high	rh	w	all	24h
	/Host_EOI	r	w	all	28h

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

	/PN_IRQ_low	rh		all	2Ch
	/PN_IRQ_high	rh		all	30h
	/PN_IRQmask_low	r	w	all	34h
	/PN_IRQmask_high	r	w	all	38h
	/PN_Event_low	rh	w	all	3Ch
	/PN_Event_high	rh	w	all	40h
	/PN_IRQack_low		w	8-16-32-Bit	44h
	/PN_IRQack_high		w	8-16-32-Bit	48h
	/PN_EOI	r	w	all	4Ch
	IO_RAM (DirectMode)	r	w	all	2000h – 3FFFh
	IO_RAM (FullRangeMode)	r	w	all	2000h – 9FFFh
	IO_RAM (PageMode)	r	w	all	1000h – 1BFFFh

2.3.11.2.2 Register description

Module: cons_control_app

Register:	Command_IF_Control_APP					Address:	0h	
Bits:	31dt0	Reset value:	0h			Attributes:	rh	w
Description:								
Bit	Identifier	Reset	Attr.		Function / Description			
4dt0	reserved	00h						
5	ConfRequest	0h	rh	w	only used for ConsRD / ConsWR: 0: no command-confirmation over Command_IF_Status 1: command-confirmation over Command_IF_Status			
7dt6	F_Code	0h	rh	w	consistency-command "00": no command "01": ConsRD "10": ConsWR "11": --> ConsError in Command_IF_Status			
12dt8	reserved	00h						
15dt13	User_ID	0h	rh	w	User-ID of the Software-Instance			
21dt16	CR_Number	0h	rh	w	0: don't care (no commandexecution) 1-n: valid values (commandexecution)			
31dt22	reserved	000h						

Register:	Command_IF_Status_APP					Address:	4h	
Bits:	31dt0	Reset value:	0h			Attributes:	rh	
Description:								
Bit	Identifier	Reset	Attr.		Function / Description			
4dt0	reserved	00h						
5	Con-fResponse	0h	rh		0: commandexecution is finished 1: commandexecution running			
7dt6	F_Code	0h	rh		Consistence Commando "00": ConsError (no command-execution because of: - invalid CR-Number in Command_IF_Control - invalid 3B-FSM State: - at (X)ConsRD U-Buffer - at (X)ConsWR U-Buffer			

					- invalid F_Code) "01": ConsRD "10": ConsWR "11": not possible
9dt8	reserved	0h			
10dt10	ConfValue	0h	rh		Only relevant for ConsRD: 0: no new RD-Data (appl O-Data / comm I-Data) 1: new RD-Data available (appl O-Data / comm I-Data)
12dt11	reserved	0h			
15dt13	User_ID	0h	rh		User-ID of the Software-Instance
21dt16	CR_Number	0h	rh		0: don't care (no commandexecution) 1-n: valid values (commandexecution)
31dt22	reserved	000h			

Register:	Host_IRQ_low				Address:	8h	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	IRQ_Bits	0h	rh		IRQ_Bits '0'=PN_Event_low='0' OR Host_IRQmask_low='1' '1'=PN_Event_low='01' AND Host_IRQmask_low='0'		

Register:	Host_IRQ_high				Address:	Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	IRQ_Bits	0h	rh		IRQ_Bits '0'=PN_Event_high='0' OR Host_IRQmask_high='1' '1'=PN_Event_high='01' AND Host_IRQmask_high='0'		

Register:	Host_IRQmask_low				Address:	10h	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Mask_Bits	ffffffffh	r	w	0: the event is inserted in Host_IRQ_low 1: the event is not inserted in Host_IRQ_low		

Register:	Host_IRQmask_high				Address:	14h	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Mask_Bits	ffffffffh	r	w	0: the event is inserted in Host_IRQ_high 1: the event is not inserted in Host_IRQ_high		

Register:	Host_Event_low	Address:	18h

Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:						
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	Event_Bits	0h	rh w	0: the event is inserted in PN_IRQ_low 1: the event is not inserted in PN_IRQ_low		

Register:	Host_Event_high			Address:	1Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:						
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	Event_Bits	0h	rh w	0: the event is inserted in PN_IRQ_high 1: the event is not inserted in PN_IRQ_high		

Register:	Host_IRQack_low			Address:	20h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:						
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	Ack_Bits	0h	rh w	write 0: the event-bit in Host_IRQ_low is not changed 1: the event-bit in Host_IRQ_low is cleared		

Register:	Host_IRQack_high			Address:	24h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:						
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	Ack_Bits	0h	rh w	write 0: the event-bit in Host_IRQ_high is not changed 1: the event-bit in Host_IRQ_high is cleared		

Register:	Host_EOI			Address:	28h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:						
Bit	Identifier	Reset	Attr.	Function / Description		
17dt0	wait_time	0h	r w	duration for deactivating of the interrupt-output in number of cycles		
31dt18	reserved	0000h				

Register:	PN_IRQ_low			Address:	2Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:						
Bit	Identifier	Reset	Attr.	Function / Description		
31dt0	IRQ_Bits	0h	rh	IRQ_Bits '0'=PN_Event_low='0' OR PN_IRQmask_low='1'		

					'1'=PN_Event_low='01' AND PN_IRQmask_low='0'

Register:	PN_IRQ_high				Address:	30h	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	IRQ_Bits	0h	rh		IRQ_Bits '0'=PN_Event_high='0' OR PN_IRQmask_high='1' '1'=PN_Event_high='01' AND PN_IRQmask_high='0'		

Register:	PN_IRQmask_low				Address:	34h	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Mask_Bits	ffffffffh	r	w	0: the event is inserted in PN_IRQ_low 1: the event is not inserted in PN_IRQ_low		

Register:	PN_IRQmask_high				Address:	38h	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Mask_Bits	ffffffffh	r	w	0: the event is inserted in PN_IRQ_high 1: the event is not inserted in PN_IRQ_high		

Register:	PN_Event_low				Address:	3Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Event_Bits	0h	rh	w	highactive event (edge-triggered) Bit 0: AHB-Error at Host-IF (AHBerr_IRQ ←Host-IF) Bit 1: XHIF-Error at Host-IF (XHIFerr_IRQ ←Host-IF) Bit 2: SPI-Error at Host-IF (SPIerr_IRQ ←Host-IF) Bit 3: not parametrized IO-access (CR_applAddressErr_INT) Bit 4: not parametrized IO-access (CR_comAddressErr_INT) Bit 5: illegal 3B-FSM-state (CR_State_applErr_INT) Bit 6: illegal 3B-FSM-state (CR_State_comErr_INT) Bit 7-15: receive of new O-Data (9xAR) <-- New_Data_INT(1..9) Bit 16-31: free configurable HW-Events (EXT_IRQ(0..15))		

Register:	PN_Event_high				Address:	40h	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Event_Bits	0h	rh	w	SW-Events		

Register:	PN_IRQack_low				Address:	44h	
Bits:	31dt0	Reset value:	0h		Attributes:		w
Description:	PN_IRQack_low						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Ack_Bits	0h		w	write 0: the event-bit in PN_IRQ_low is not changed 1: the event-bit in PN_IRQ_low is cleared		

Register:	PN_IRQack_high				Address:	48h	
Bits:	31dt0	Reset value:	0h		Attributes:		w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Ack_Bits	0h		w	write 0: the event-bit in PN_IRQ_high is not changed 1: the event-bit in PN_IRQ_high is cleared		

Register:	PN_EOI				Address:	4Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
17dt0	wait_time	0h	r	w	duration for deactivating of the interrupt-output in number of cycles		
31dt18	reserved	0000h					

Memory:	IO_RAM			Startaddress:	2000h	
Bits:	31dt0	Size:	8000h	Attributes:	r	w
Description:						

2.3.11.3 APB interface

2.3.11.3.1 Address mapping

Start-Address	End-Address	Modul/Memory-Name
2000h	9FFFh	IO_RAM
0h	9FFFh	cons_control_apb

Module	Register/Memory	Read	Write	Access	Address
/cons_control_apb				all	
	/CR_Address_1	r	w	all	0h
	...				
	/CR_Address_n	r	w	all	F8h
	/CR_State_1	r(h)	w	all	100h
	...				

	/CR_State_n	r(h)	w	all	1F8h
	/Address_Mode	r	w	all	600h
	/IP_Version	r		all	800h
	/IP_Development	r		all	804h
	/Access_Error	rh	w	all	808h
	/Burst_Config	r	w	all	80Ch
	IO_RAM	r	w	all	2000h-9FFFh

2.3.11.3.2 Register description

Module: cons_control_apb

Register:	CR_Address_1					Address:	0h	
Bits:	31dt0	Reset value:			1FFCh	Attributes:	r	w
Description:								
Bit	Identifier	Reset	Attr.		Function / Description			
1dt0	reserved	0h						
12dt2	CR_StartAddress	7FFh	r	w	Address_Mode = DirectMode: startaddress of the CR (4-Byte aligned, 512 Byte address range) Address_Mode = PageMode not used			
15dt13	reserved	0h						
26dt16	CR_EndAddress	0h	r	w	Address_Mode = DirectMode: EndAddress of the CR (4-Byte aligned, 512 Byte address range) Address_Mode = PageMode not used			
27	Zero_Data	0h	r	w	With a buffer-change of the applicative Interface (ConsRD: N --> U) the next reading returns: '0': the Original Data from the IO-RAM '1': Zero-Data (0x00..00)			
31dt28	New_Data_INT	0h	r	w	If the application gets new data (bufferchange D -> N): 0x0, 0xA-0xF: no New_Data_INT is emitted 0x1 - 0x9: New_Data_INT_1 - New_Data_INT_9 is emitted.			

…

Register:	CR_Address_n					Address:	F8h	
Bits:	31dt0	Reset value:			1FFCh	Attributes:	r	w
Description:								
Bit	Identifier	Reset	Attr.		Function / Description			
1dt0	reserved	0h						
12dt2	CR_StartAddress	7FFh	r	w	Address_Mode = DirectMode: startaddress of the CR (4-Byte aligned, 512 Byte address range) Address_Mode = PageMode not used			
15dt13	reserved	0h						
26dt16	CR_EndAddress	0h	r	w	Address_Mode = DirectMode:			

					EndAddress of the CR (4-Byte aligned, 512 Byte address range) Address_Mode = PageMode not used
27	Zero_Data	0h	r	w	With a buffer-change of the applicative Interface (ConsRD: N --> U) the next reading returns: '0': the Original Data from the IO-RAM '1': Zero-Data (0x00..00)
31dt28	New_Data_INT	0h	r	w	If the application gets new data (bufferchange D -> N): 0x0, 0xA-0xF: no New_Data_INT is emitted 0x1 - 0x9: New_Data_INT_1 - New_Data_INT_9 is emitted.

Register:	CR_State_1				Address:	100h	
Bits:	31dt0	Reset value:		FD2h	Attributes:	r(h)	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	Buffernumber for R1ed / Data-Buffer		
3dt2	buffer_number_r2_buffer	0h	rh	w	Buffernumber for R2ed		
5dt4	buffer_number_free_buffer	1h	rh	w	Buffernumber for F1ree- / Free-Buffer		
7dt6	buffer_number_f2_buffer	3h	rh	w	Buffernumber for F2ree-Buffer		
9dt8	buffer_number_next_buffer	3h	rh	w	Buffernumber for N1ext- / Next-Buffer		
11dt10	buffer_number_n2_buffer	3h	rh	w	Buffernumber for N2ext-Buffer		
13dt12	buffer_number_user_buffer	0h	rh	w	Buffernumber for User-Buffer		
14	Redundance_Mode	0h	r	w	only used when EXT_BUF_Mode = '0' (Normal-Mode): The Output-data (O-Data) are managed from the communicative instance in the 3-Buffer-Mode with '0': one D-Buffer '1': two R-Buffer (R1-, R2-Buffer) --> redundancy-Support		
15	EXT_BUF_Mode	0h	r	w	'0': Normal-Mode: the CR is using the "normal" 3-Buffer mode '1': Extension-Mode: the CR is using the "extended" 3-Buffer mode		
31dt16	Mapping	0h	r	w	Address_Mode = DirectMode: not used Address_Mode = PageMode mapping of the MapBits to a page in the IO-RAM (MapBit0 --> Page 0, MapBit1 --> Page1, ... MapBit15 --> Page 15) MapBit x = '0': The CR is complete or partially not in the Page x inserted MapBit x = '1': The CR is complete or partially in the Page x inserted		

...

Register:	CR_State_n				Address:	1F8h	
Bits:	31dt0	Reset value:		FD2h	Attributes:	r(h)	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	Buffernumber for R1ed / Data-Buffer		
3dt2	buffer_number_r2_buffer	0h	rh	w	Buffernumber for R2ed		
5dt4	buffer_number_free_buffer	1h	rh	w	Buffernumber for F1ree- / Free-Buffer		
7dt6	buffer_number_f2_buffer	3h	rh	w	Buffernumber for F2ree-Buffer		
9dt8	buffer_number_next_buffer	3h	rh	w	Buffernumber for N1ext- / Next-Buffer		
11dt10	buffer_number_n2_buffer	3h	rh	w	Buffernumber for N2ext-Buffer		
13dt12	buffer_number_user_buffer	0h	rh	w	Buffernumber for User-Buffer		
14	Redundance_Mode	0h	r	w	only used when EXT_BUF_Mode = '0' (Normal-Mode): The Output-data (O-Data) are managed from the communicative instance in the 3-Buffer-Mode with '0': one D-Buffer '1': two R-Buffer (R1-, R2-Buffer) --> redundancy-Support		
15	EXT_BUF_Mode	0h	r	w	'0': Normal-Mode: the CR is using the "normal" 3-Buffer mode '1': Extension-Mode: the CR is using the "extended" 3-Buffer mode		
31dt16	Mapping	0h	r	w	Address_Mode = DirectMode: not used Address_Mode = PageMode mapping of the MapBits to a page in the IO-RAM (MapBit0 --> Page 0, MapBit1 --> Page1, ... MapBit15 --> Page 15) MapBit x = '0': The CR is complete or partially not in the Page x inserted MapBit x = '1': The CR is complete or partially in the Page x inserted		

Register:	Address_Mode				Address:	600h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
0	Address_Mode	0h	r	w	ConsistencyControl (AHB-Slave0, AHB-Slave1, Address_guard) is using the addressing-mode '0': DirectMode '1': PageMode		
31dt1	reserved	00000000h					

Register:	IP_Version	Address:	800h

Bits:	31dt0	Reset value:	200h	Attributes:	r	
Description:						
Bit	Identifier	Reset	Attr.	Function / Description		
7dt0	debug_version	0h	r	incremented for Bugfixes (e.g. metalfix)		
15dt8	version	2h	r	Version of the IP: 0x01: initial Version (T-ASIC) 0x02: ERTEC200P Step 3		
31dt16	configuration	0h	r	Characteristic of the IP: 0x000: 3x1.5KByte IO-RAM, 27 CRs, 100MHz (TechASIC characteristic) 0x001: 4x8KByte IO-RAM, n CRs, 125MHz		

Register:	IP_Development			Address:	804h	
Bits:	31dt0	Reset value:	A00080Eh	Attributes:	r	
Description:						
Bit	Identifier	Reset	Attr.	Function / Description		
10dt0	r_label	eh	r	ClearCase RR-Label		
15dt11	Inkr	1h	r	Increment of the ClearCase RR-Label		
18dt16	label_structure	0h	r	patchPatch marking of metal-fixes (gray encoded)		
20dt19	type	0h	r	Type Environment of ConsistencyControl (ASIC, FPGA, user defined)		
31dt21	identification	050h	r	Identification. unique identifier for ConsistencyControl		

Register:	Access_Error			Address:	808h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:						
Bit	Identifier	Reset	Attr.	Function / Description		
16dt0	ahb_apb_address	0h	rh w	Address of the failing AHB- or APB-access to ConsistencyControl		
26dt17	reserved	000h				
27	ahb_apb_select	0h	rh w	'0': AHB-access '1': APB-access		
29dt28	ahb_apb_size	0h	rh w	"00": Byte-access (8 Bit) "01": Halfword-access (16 Bit) "10": Word-access (32 Bit) "11": reserved		
30dt30	ahb_apb_rd_wr	0h	rh w	'0': RD-access '1': WR-access		
31dt31	err_lock	0h	rh w	hardware set this bit to '1' if a failing AHB-/APB-access is recognized. No Further HW-Entries are done. The SW has to set this Bit to '0' to relock further entries. '0': no failing APB- or AHB-access is recongized '1': a failing APB- or AHB-access is recongized		

Copyright © Siemens AG 2023. All rights reserved

Register:	Burst_Config				Address:	80Ch	
Bits:	31dt0	Reset value:	3h		Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	BurstMode_comAHB	3h	r	w	A AHB RS/WR Burst on the communicative AHB-Slave 0 is "x0": not (--> single accesses) "01": for all bursts except INCR (--> single accesses) "11": for all bursts (INCR, INCR4, ..., WRAP4, ... WRAP16) supported on the SC-Bus.		
7dt2	reserved	00h					
9dt8	BurstMode_applAHB	0h	r	w	A AHB RS/WR Burst on the applicative AHB-Slave 1 is "x0": not (--> single accesses) "01": for all bursts except INCR (--> single accesses) "11": for all bursts (INCR, INCR4, ..., WRAP4, ... WRAP16) supported on the SC-Bus.		
31dt10	reserved	000000h					

Memory:	IO_RAM			Start-address:	2000h	
Bits:	31dt0	Size:	8000h	Attributes:	r	w
Description:						

2.3.12 PROFINET-IP

NOTICE

This chapter is purely informative, the PN-IP is handled completely by the PN stack. We strongly recommend **not** to modify any of these settings.

2.3.12.1 PN-IP interfaces

2.3.12.1.1 AHB interface

The PN-IP is connected to the multi-layer AHB over an AHB master and an AHB slave interface. Over the AHB master interface, the PN-IP can transfer data over the multi-layer AHB as the active bus master. At the same time, other AHB masters can access PN-IP resources over the AHB slave interface.

The AHB interfaces of the PN-IP do not support split or retry functions. All other transfer types, transfer sizes and burst operations as per the AMBA specifications are supported.

2.3.12.1.2 Interrupt management

All interrupt events generated by mechanisms of the PN-IP can be provided over CPUs connected to Interrupt Management. The ERTEC 200P supports the following CPU systems (CPU subsystems):

Copyright © Siemens AG 2023. All rights reserved

- ARM926 subsystem
- Event unit in PerIF (external host CPU)

PN stack and application SW allocation in the CPU systems is not fixed. Individual interrupt event assignment to interrupt inputs / interrupt controllers of the CPUs is therefore universally applicable. For the ARM926 and host subsystem, the specific number of all interrupt events occurring in the PN-IP is provided. This is done as follows:

- with a CPU-specific interrupt controller (PN-ICU) that generates two group interrupts each for all interrupt events for the ARM926 (PN-ICU 2) / external host (PN-ICU 3) → PN_IRQ2/3(**1:0**)
- with a CPU-specific multiplexer structure (PN-MUX) that provides 14 selectable PN interrupt events for the ARM926 (PN-ICU 2) / external host (PN-ICU 3) as single interrupts → PN_IRQ2/3(**15:2**).

Figure 29: Diagram of interrupt management in the PN-IP

The group interrupts PN_IRQ2(**1:0**) (IRQ56/57) and 14 selectable single interrupts PN_IRQ2(**15:2**) (IRQ58-71) are connected to the ARM926EJ-S interrupt controller (ARM-ICU) by the PN-IP (see chapter 2.3.5.9).

Interrupts for the host are generated by the event unit in the peripheral interface. The PN-IP forwards the relevant interrupts (group interrupts: PN_IRQ3(1:0), single interrupts: PN_IRQ3(15:2)) to the event unit, where a group interrupt signal to the external host (ASIC pin: XHIF_XIRQ) is generated.

2.3.12.1.3 PN-ICU

The group interrupt PN_IRQx(**0/1:0**) is generated as shown below:

Figure 30: Block diagram of PN-ICU for PN_IRQx(0/1:0) group interrupts

For each group interrupt (2x for ARM926 / external host: PN_IRQ2/3(1:0): PN_IRQ1(0), there is one interrupt controller. 'IRQEvent' registers are implemented for signaling and saving internal PN-IP events. The same events are connected to all interrupt controllers. One or more event bits set can each trigger a group interrupt for the external CPU subsystem → PN_IRQx(0/1:0). For all event bits, the interrupt is triggered by:

- internal PN-IP events (PN-IP events) OR
- dedicated AHB write access to IRQEvent OR
- dedicated AHB write access (SW events over IRQ_Activate).

Event bits already written/set are retained and not reset by subsequent access to the 'IRQEvent' registers. The assignment and significance of the individual event bits is identical for all PN-ICUs (1...3).

The decision whether an interrupt event (set event bit) triggers the corresponding PN_IRQx(0/1:0) group interrupt is configured with event bit masking with the 'IRQMask' registers. A set mask bit blocks the corresponding entry and prevents PN_IRQx(0/1:0) from being triggered. Over the IRQ registers, the event bits that trigger a PN_IRQx(0/1:0) group interrupt (IRQ bits) can then be read. Writing to these registers is ignored. The two group interrupts are masked differently by the SW to create interrupt groups (e.g., acyclic API and cyclic API by group).

Bits in the 'IRQEvent' registers are reset by write access to the 'IRQAck' registers. The PN-ICU may only be operated in "**Acknowledge**" mode, i.e., the event bits set are established with read access to the IRQ registers. Subsequent write access to the 'IRQAck' registers resets the register bits written in the 'IRQEvent' registers and therefore also the IRQ bits in the IRQ registers.

PN_IRQx(0/1:0) group interrupts are deactivated over the 'IRQ_EOI' registers. Write access to the registers resets the PN_IRQx(0/1:0) group interrupts. The PN_IRQx(0/1:0) group interrupt cannot be reactivated by set event bits in the 'IRQEvent' registers until the wait time (Wait_Time) written is up.

To flexibly adapt external HW events to Interrupt Management, they can be configured separately over the 'IRQExt_Event' register using *Ext_IRQx_Level* for the active level. Either level or edge triggering can also be selected with *Ext_IRQx_Edge*.

2.3.12.1.3.1 PN-MUX

Each PN_IRQx(**2:1/15:2**) single interrupt is generated as shown here:

Figure 31: Block diagram of PN-MUX for PN_IRQx(2:1/15:2) single interrupts

Event bits 63 - 60 are assigned bits(3:0) from the 'IRQ_Activate' registers to allow a SW event to be triggered with PN_IRQx(0/1:0) group interrupts and PN_IRQx(2:1/15:2) single interrupts. Write

access to the relevant register bits triggers a SW event as a pulse (8 ns pulse). Write access in the 'IRQ_Activate' registers is non-retentive. Read access always returns a value of 0x0000.

For each interrupt output, you can set which PN-IP event or SW event is implemented externally as a single interrupt over the 'IRQControl_MUXn' register with *IRQ_SelectMUX*. PN-IP event are assigned event numbers (*IRQ_SelectMUX*) in line with the assignment (bit position) of the PN-IP events in the 'IRQEvent' register.

The interrupt load of the selected single event in the connected CPU subsystems can be reduced by writing *IRQxControl_MUXn.IRQ_WaitTime*. No new single event (single interrupt) is entered in the external CPU subsystem while *IRQ_WaitTime* is running (*IRQ_WaitTime* $\neq$ *0x000*). During this time, the selected events are buffered in *IRQxControl_MUXn.IRQ_Event* and trigger a single interrupt as soon as *IRQ_WaitTime* (*IRQ_WaitTime = 0x000*) is up. At the same time, the value saved in *IRQ_Event* is cleared. The single event is extended to the configured pulse length (*IRQx_Control.IRQ_Length*) with pulse extension. The current timer value can be read back with read access. If there is another instance of read access to *IRQxControl_MUXn* before IRQ_WaitTime is up, the newly written value is used as *IRQ_WaitTime*. This allows *IRQ_WaitTime* to be extended or shortened by writing the SW in time.

As long as *IRQ_WaitTime* is not written and therefore not used (i.e., *IRQ_WaitTime* = 0x000, default value), the selected single event (single interrupt) is connected straight to the relevant CPU subsystems (combinatorial).

2.3.12.1.4 PN-PLL (incl. 3 × Application Timer Blocks)

The PN-PLL contains a clock instance (Clock_A), a time instance and 3 application timer blocks. Each application timer block controls its own application cycle and is connected to the Clock_A instance. The application clock (APPL1/2/3_CLK_INT) and 6 programmable phase signals (APPL1/2/3_ CMP_INT5...0) are available as outputs for each application timer block. These 21 outputs are connected to a 21-input multiplexer (PNPLL_OUT20...0). Any output of the 3 application timer blocks can be selected at each MUX output. The controlled clock of the clock instance (PNCLKA_OUT) can also be connected to the MUX outputs.

Figure 32 shows how these outputs (PNPLL_OUT20...0) are connected to the application instances at the ERTEC 200P top-level.

Figure 32: PNPLL with 3 application time blocks (application connection)

The connection matrix is as follows.

Table 26: PNPLL_OUT connection matrix

	GPIO	ARM-ICU	GDMA
PNPLL0	GPIO0_INT		

	GPIO	ARM-ICU	GDMA
PNPLL1	GPIO1_INT		
PNPLL2	GPIO2_INT		
PNPLL3	GPIO3_INT		
PNPLL4	GPIO4_INT		
PNPLL5	GPIO5_INT		
PNPLL6	GPIO6_INT		
PNPLL7	GPIO7_INT		
PNPLL8	GPIO8_INT		
PNPLL9		IRQ72	
PNPLL10		IRQ73	
PNPLL11		IRQ74	HWJob(0)
PNPLL12		IRQ75	HWJob(1)
PNPLL13		IRQ76	HWJob(2)
PNPLL14		IRQ77	HWJob(3)
PNPLL15			HWJob(4)
PNPLL16			HWJob(5)
PNPLL17			HWJob(6)
PNPLL18			HWJob(7)
PNPLL19			HWJob(8)
PNPLL20			HWJob(9)

The overlap of the PNPLL signals (e.g., PNPLL8..0) with different IPs is necessary to cover the different use cases. However, there is no overlapping within any one use case, i.e., multiple IPs are never triggered with the same PNPLL signals. The PNPLL signals that overlap always go to inactive IPs. Behavior is as follows:

- With GPIO, the PNPLL8..0 signals must in this case not be connected to the pins.
- With the ARM-ICU, the PNPLL14..9 signals must set the relevant interrupt masks.
- With GDMA, the PNPLL20..11 signals must not be selected for job control.

2.3.12.1.5 EDC in PN-IP RAMs

The memories in the PN-IP have an EDC code (1-bit error, correctable; 2-bit error, detectable). The PN-IP has the group outputs PN-1B (1 bit error correct) and PN-2B (2 bit error detected). If an error occurs during reading, the cause of the error (PN-1B: 1 bit error or PN-2B: 2 bit error) is saved in the SCRB register 'EDC_EVENT' (see chapter 2.3.14.11.27.1) and the interrupt EDC_Event (IRQ48) is triggered (see 2.3.5.9). 0x00…00 must be written to the 'EDC_EVENT' register to clear the error. The PN-IP RAM is initialized after this reset. The end of initialization (incl. EDC bits) is signaled in the SCRB register 'EDC_INIT_DONE' (see 2.3.14.11.13).

2.3.12.1.6 Recommended parameter assignment for ERTEC 200P

The PN-IP has no internal RAM for managing the data to be sent and received. Instead, it uses either the IO-RAM in the PerIF module or external memory connected to the EMC (see chapter 2.3.8). For high-performance data transfer from/to the PN-IP, we recommend configuring the ERTEC 200P modules listed below as detailed.

The following table shows the recommended parameter assignment for time-triggered injection with PPM and the receipt of cyclic RT frames by the CPM.

Table 27: Recommended parameter assignment for time-triggered injection

Module	Register	Comment
PerIF	**Burst_Config(7:0) = 0x03**	Communicative AHB access is implemented as burst access at the IO-RAM. Please **note** that the PerIF interrupt CR_State_comErr_INT can be triggered due to errors in the settings. Remedy: one of the two options below:

Module	Register	Comment
		- Mask interrupt in the PerIF registers 'Host_IRQmask_low' and 'PN_IRQmask_low' (this interrupt is merely a debug feature). - Make the CR one 32-bit word greater than the user data read by the PPM / written by the CPM.
	Burst_Config(15:8) = 0x00	Applicative AHB access is implemented as single access at the IO-RAM. This is necessary to ensure in-line PPM/CPM data transfer (communicative AHB access).

The following table shows the recommended parameter assignment for CPM cut-through receipt at an external memory connected to the EMC.

Table 28: Recommended parameter assignment for CPM cut-through receipt

Module	Register	Comment
PN IP	CPM_AHB_Lockout_Starttime CPM_AHB_Lockout_Stoptime CPM_AHB_Lockout_Control	Use of the registers listed to block acyclic Snd-/Rcv-API access to the AHB bus for a configurable time in the communication cycle.
SCRB	AHB_BurstBreaker(7:0) = 0x08 AHB_BurstBreaker(15:8) = 0x08	For the ARM926I and ARM926D AHB master interface, the AHB burst length is to be limited to 8 in the 'AHB_BURSTBREAKER' register (default is no limit)
GDMA	Burst_Mode = Single - INCR8	GDMA jobs whose source or destination address points to an address range of the EMC must be limited to single, INCR4 or INCR8 in Burst_Mode (i.e., do not use INCR16)
EMC	Asyn Memory Banks 0-3 Waits	Access by other masters (ARM926D, ARM926I, GDMA or HostIF) to EMC asyn memory banks 0-3 delays CPM data transfer to the backplane bus ASIC, which is also connected to one of the 4 banks. The wait states of the other 3 banks (either set in the configuration or created by an external "ready") must therefore be included in the calculation of the CPM transfer delay if they are greater than the time required for an 8-transfer burst to the SDRAM and other masters access the CPM during CPM data transfer.

2.3.12.2 Ethernet PHY (integrated)

Two single-port PHYs (Physical Layer Transceiver) that support 100BASE-TX and 100BASE-FX are integrated in the ERTEC 200P. The interfaces for 100BASE-TX and 100BASE-FX are separate for each port and can be set differently. The PHYs are fully compatible with IEEE802.3 (although 10BASE-T is not supported) and other standards (ANSI X3.263-1995 and ISO/IEC9314 etc.).
The PHY supports the following functions:

- 100Base-TX PHY (10 MBd is not supported by ERTEC 200P; the SW must take account of this)
- MII MAC interface (ERTEC 200P only uses MII)
- Support Fiber mode
- Auto MDI-X
- Fast link up time
- Single Central PLL and BIAS gen
- Maximum Roundtrip Latency of 296 ns (Transmit and Receive combined)
- Jitter-free Latency
- Fast line break detection
- 250 MHz clock can be switched off

Copyright © Siemens AG 2023. All rights reserved

The data interface to the PNIP is over MII and the management interface is over the MDIO interface (SMI interface). The clock source is 250 MHz, can be switched off, and is provided by the CRU.

2.3.12.2.1 Status of the integrated PHY

2.3.12.2.1.1 PHY status LED

The following PHY status LEDs are connected to the pinning).
Dedicated PHY LEDs:

- P1/2_XLINK_STATUS (Link / No Link)
- P1/2_XACITVITY (Receive or Transmit Activity / No Activity)

These LED outputs can be controlled directly from the PHYs. P1/2_XActivity combines the receive and transmit activity outputs of the PHYs. Alternatively, these LED outputs can also be controlled by the SW. SW control is with the 'PHY_LED_CONTROL' register in the SCRB. The status of the PHY outputs 'P1/2_XLINK_STATUS / P1/2_XACITVITY' can also be queried with the 'PHY_LED_CONTROL' register if the SW is controlling the LEDs.

Copyright © Siemens AG 2023. All rights reserved

2.3.13 CRU (Clock and Reset Unit)

2.3.13.1 Clock system

The ERTEC 200P clock system consists primarily of clock groups which are isolated with asynchronous crossings. These systems are as follows:

- ARM926EJ-S processor system
- JTAG interfaces
- AHB system, APB system, PN-IP and PerIF
- Host interface (parallel: XHIF)
- PHYs
- OSPI

Note

For clock generation and distribution and for clock monitoring, refer to the ERTEC 200P-3 data sheet.

2.3.13.2 Reset system

There are a number of options for asynchronous ERTEC 200P resets alongside an external PowerOn reset (XRESET): an additional external hardware reset by the debugger (XSRST), an internal ARM926 watchdog reset (XRES_ARM926_WD) and a number of resets with the software. The reset matrix in Table 29 shows all reset sources (rows) and their implications for each circuit component (columns).

A PowerOn reset (XRESET) is applied to ERTEC 200P with an external pin. This PowerOn reset resets the entire circuit (incl. clock system). The XSRST pin is used for a hardware reset by the debugger. The clock system is not reset and communication over the JTAF interface is possible during this reset phase. The ERTEC 200P can be monitored with an ARM926 watchdog. An ARM926 watchdog event (XRES_ARM926_WD) then resets the ERTEC 200P. The SCRB register 'ASYN_RES_CTRL_REG' (see chapter 2.3.14.11.27.1) allows you to exclude the PN-IP from a watchdog reset (EN_WD_RES_PN).

'ASYN_RES_CTRL_REG' can also be used by the software to trigger an internal asynchronous reset without the PN-IP (RES_SOFT), an internal asynchronous reset for the PN-IP (RES_SOFT_PN) only, an internal asynchronous reset for the ARM926 core system (RES_SOFT_ARM926_CORE).

The trigger event for the last reset for the ARM926EJ-S core can be read from the SCRB register 'RES_STAT_REG' (see 2.3.14.11.27.1).

Copyright © Siemens AG 2023. All rights reserved

Table 29: ERTEC 200P reset matrix

Destinations / Events	Degubber/Direction (Pin XSRST)	ARM926-Core (XRES_ARM926CORE)	ARM926-TCM Top-level SCRB-TCM926 RES_SOFT_RETURN_ADDR (XRES_SYS)	PN-IP SCRB_PHY_CONFIG/STATUS SCRB_ASYN_RES_CTRL_REG Slice3 EN_WD_RES_PN (XRES_PN-IP)	Clock System (XRES_CTSRS)	JTAG_NTRST (EAS926)	SCRB_ASYN_RES_CTRL_REG Slice5	SCRB_RES_STAT_REG	OSPI (xres_ospi_shell_sctsrs)
PowerOn Reset ERTEC 200P (Pin XRSET)	x/out	**x**	**x**	**x**	**x**	**x**	**x** (1h by PowerOn Reset)	**x** (4h by PowerOn Reset)	**x**
Debugger (Pin XSRST)	-/in	**x**	**x**	**x**	-	**x**	Set to 1	Set to 4h	**x**
JTAG Reset (Pin XTRST)	-/in	-	-	-	-	-	-	-	-
Watchdog Reset ARM926 + PN-IP (XRES_ARM926_WD + Logic)	-/in	**x** (Pulse duration)	**x** (Pulse duration)	**x** (Pulse duration)	-	-	Set to 1	Set to 1h	**x** (Pulse duration)
Watchdog Reset ARM926 without PN-IP (XRES_ARM926_WD + Logic)	-/in	**x** (Pulse duration)	**x** (Pulse duration)	-	-	-	-	Set to 1h	**x** (Pulse duration)
SW Reset ERTEC 200P without PN-IP/PHY (RES_SOFT)	-/in	**x** (Pulse duration)	**x** (Pulse duration)	-	-	-	-	Set to 2h	**x** (Pulse duration)
SW Reset PN-IP/PHY (RES_SOFT_PN)	-/in	-	-	**x** (Pulse duration)	-	-	Set to 1	-	-
Core Reset ARM926 RES_SOFT_ARM926_CORE	-/in	**x** (Pulse duration)	-	-	-	-	-	Set to 8h	-

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Key:

- Destination/module is not affected by reset event.

Note

Please refer to the ERTEC 200P-3 data sheet for the following reset options:

- Asynchronous PowerOn Reset (XRESET)
- Asynchronous Hardware Reset
- Asynchronous JTAG Reset
- Asynchronous ARM926 Watchdog Reset
- Asynchronous Software Reset for ERTEC 200P (without PN-IP)
- Asynchronous Software Reset for PN-IP
- Asynchronous Software Reset for the ARM926EJ-S Core
- Synchronous Software Reset (PN-IP, PER-IF, Host Interface)

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14 APB Peripherals

2.3.14.1 Access modes and data widths

The function blocks connected to the APB have interfaces of differing widths. The table below shows the access mechanisms supported and the data widths of these blocks.

Table 30: Data width of peripherals

Access modes				Wait states at the AHB		
Bit 31:24	**Bit 23:16**	**Bit 15:8**	**Bit 7:0**	**Read**	**Write**	**Function block**
8-bit	8-bit	8-bit	8-bit	2	0	IO filter, Boot_ROM
16-bit		16-bit				
32-bit						
8-bit	8-bit	8-bit	8-bit	Ready	0	PER_IF (except PER_IF-GPIO)
16-bit		16-bit				
32-bit						
32-bit				2	0	GPIO, PER_IF-GPIO, Timer0-5, F-counter, Watchdog, SCRB, SPI1/2, Flash controller, UART1-4, I^2C, Host interface, OSPI

Access modes that are not permitted (e.g., writing byte by byte/halfword by halfword to the timer) are not prevented by the hardware, nor are they indicated at the AHB in the form of an error response or QVZ interrupt.

Access to memory ranges that are not decoded (marked as "not used" in memory mapping) triggers IRQ52 (see chapter 2.3.5.9) and ends with a ready signal generated by the APB address decoder. Write access does not have any effect on the system. Read access returns undefined data.

Copyright © Siemens AG 2023. All rights reserved

2.3.14.2 I-Filter

Up to 80 input signals can be sent by the GPIO interface (GPIO15-0, GPIO95-32) through an I-Filter. Refer to the ERTEC 200P-3 data sheet for the corresponding I-Filter input for each GPIO. The pending input signals D_IN can be saved unfiltered and directly (mode 1: no synchronization), unfiltered and synchronized (mode 2) or filtered (mode 3) in the modules below. In all cases, the input signals are sent through the I-Filter which operates on a channel-specific basis and can be configured over the APB.

Figure 33: Block diagram of the IO filter

The APB module "I-Filter" allows you to filter out brief signal spikes and interference. Filtering is done using a synchronization unit (IN_Sync) and filter stage (RC_Filter) in the module, which are available for each input (channel). If you do not want to filter an input signal, either the synchronization stage of the I-Filter is used (mode 2) or the input signal itself is forwarded without processing by the I-Filter (mode 1). The filter stage is supplied by a central clock divider, whose configurable clock sources can also be selected on a channel-specific basis. The block diagram below shows exactly how the I-Filter operates, using the example of one channel.

Figure 34: Block diagram of the filter structure of a channel

The input signal is filtered following two-stage synchronization (this is required to avoid metastable states). The channel-specific filter is implemented as a 5-bit up/down counter that counts up when the input signal is "high" and down when the input signal is "low". The counter value after RESET is '00h' (load value).

As the input channels for the digital input modules are operated separately, each input signal requires its own counter. These counters must enable different, channel-specific filter time settings.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

The filter time is selected in the 'FILT_Delay_x' registers, which in line with the settings select different clock sources on a channel-specific basis (mode 3) or select input signal synchronization only without RC filtering (mode 2). In mode 2, the doubly synchronized input signal D_IN_SYN is connected straight to the output signal D_IN_Delay and not through the count register. In mode 1, the input signal D_IN is connected straight to the output signal D_IN_Delay, bypassing the synchronization stage and count register. Dynamic FILT_Delay_x toggling is not permitted during operation. Parameter assignment is in the initialization phase only.

The clock sources (CLOCK0 to CLOCK7) are generated for all channels by a central clock divider. The base clock for all divider stages is the APB clock PCLK (125 MHz). The block diagram below shows how the system operates.

Figure 35: Block diagram of the central clock divider

The central clock divider consists of several divider stages, some of which are cascaded. The division factors are set for each divider stage with a dedicated 'FILT_Reload_x' register. Dynamic FILT_Reload_x toggling is not permitted during runtime.

The division factors of the individual stages of cascaded dividers must be multiplied to calculate the "total division factor". The following table shows the filter times achieved with sample division factors at 125 MHz.

Table 31: Sample filter times for I-Filter

Parameter register 'FILT_Reload_x'	Sample division factor	Filter time (without 2-stage synchr.) t_{delay}	Minimum filter time (filter time – CLOCK jitter)
Reload_0	8	1.024 µs (CLOCK0)	0.960 µs
Reload_1	40	5.12 µs (CLOCK1)	4.80 µs
Reload_2	78	9.984 µs (CLOCK2)	9.360 µs
Reload_3	390	42.42 µs (CLOCK3)	39.30 µs
Reload_4	2	99.84 µs (CLOCK4)	93.60 µs
Reload_5	6	299.5 µs (CLOCK5)	280.78 µs
Reload_6	30	1.5 ms (CLOCK6)	1.406 ms
Reload_7	40	2.0 ms (CLOCK7)	1.875 ms

The filter time is calculated by multiplying the division factor (cascaded from divider 3) by 16×TPCLK, as the filter system means that the output value can only change for every 16th clock.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

The filter times are as follows:

- min. **128 ns** e.g., when: *IN_Delay_x=0 & FILT_Reload_0*=0
- max. **134 ms** e.g., when: *IN_Delay_x=7 & FILT_Reload_3*=0x3FF & *FILT_Reload_7*=0x3FF

The maximum (total) throughput time through the filter module is if configured equal to the filter time plus 16 ns for two-stage synchronization. The minimum (total) throughput time (see Table 31) in this configuration depends on the CLOCK0 to CLOCK7 clock signal jitter.

2.3.14.2.1 Operating principle of the RC filter

The most significant bit in the 5-bit counter is the filtered output signal (*D_IN_Delay*).

If the input signal (*D_IN_SYN*) switches to "High", the counter starts to count up. If the counter status changes from '15' to '16', the counter value is automatically set to '31' ('0x1Fh'). When the counter reaches '31', the output signal (*D_IN_Delay*) is also set to "High". The counter value remains at '31' until the input signal (*D_IN_SYN*) changes from "High" to "Low".

If the input signal (*D_IN_SYN*) switches to "Low", the counter starts to count down. If the counter status changes from '15' to '16', the counter value is automatically set to '0' ('0x00h'). When the counter reaches '0', the output signal (*D_IN_Delay*) is also set to "Low". The counter value remains at '0' until the input signal (*D_IN_SYN*) changes from "Low" to "High".

Brief signal changes during the counting phase cause the counter to count either up or down in line with its signal state, i.e., the counter does not start from '0' following a signal change (that would be a signal delay, not filtering).

The filter structure therefore acts as an RC circuit or digital I-controller.

Figure 36: Signal in signal filtering

2.3.14.2.2 Address mapping

Start-Address	End-Address	Modul/Memory-Name
0h	53h	i_filter

Module	Register/Memory	Read	Write	Access	Address
/i_filter				all	
	/FILT_IP_VERSION	r		all	0h

	/FILT_IP_DEVELOPMENT	r		all	4h
	/FILT_ACCESS_ERR	(r)(h)	(w)	all	8h
	/FILT_RELOAD_0	r	w	all	Ch
	/FILT_RELOAD_1	r	w	all	10h
	/FILT_RELOAD_2	r	w	all	14h
	/FILT_RELOAD_3	r	w	all	18h
	/FILT_RELOAD_4	r	w	all	1Ch
	/FILT_RELOAD_5	r	w	all	20h
	/FILT_RELOAD_6	r	w	all	24h
	/FILT_RELOAD_7	r	w	all	28h
	/FILT_DELAY_0	r	w	all	2Ch
	/FILT_DELAY_1	r	w	all	30h
	/FILT_DELAY_2	r	w	all	34h
	/FILT_DELAY_3	r	w	all	38h
	/FILT_DELAY_4	r	w	all	3Ch
	/FILT_DELAY_5	r	w	all	40h
	/FILT_DELAY_6	r	w	all	44h
	/FILT_DELAY_7	r	w	all	48h
	/FILT_DELAY_8	r	w	all	4Ch
	/FILT_DELAY_9	r	w	all	50h

2.3.14.2.3 Register description

Module: i_filter

Register:	FILT_IP_VERSION				Address:	0h	
Bits:	31dt0	Reset value:	10100h		Attributes:	r	
Description:	IP Version register						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	DEBUG_VERSION	0h	r		Incremented for error correction (e.g. metal fix)		
15dt8	VERSION	01h	r		IP version: 0x01: Initial version		
31dt16	CONFIGURATION	0001h	r		IP configuration: 0x0000: 8 filter times, 64 input filters 0x0001: 8 filter times, 80 input filters		

Register:	FILT_IP_DEVELOPMENT				Address:	4h	
Bits:	31dt0	Reset value:	6800808h		Attributes:	r	
Description:	Project-specific design label						
Bit	Identifier	Reset	Attr.		Function / Description		
10dt0	BASELINE	8h	r		Number of RR label		

15dt11	INKREMENT	1h	r		HDL increment of the ClearCase label
18dt16	PATCH	0h	r		For labeling metal fixes in the ASIC flow; only valid if platform = ASIC. Inkrement/Baseline is then invalid
20dt19	PLATFORM	0h	r		00 = ASIC 01 = FPGA 10 = reserved 11 = user defined
31dt21	IDENTIFIKATION	34h	r		Unique number per module

Register:	FILT_ACCESS_ERR				Address:	8h	
Bits:	31dt0	Reset value:	0h		Attributes:	(r)(h)	(w)
Description:	Access error register						
Bit	Identifier	Reset	Attr.		Function / Description		
6dt0	APB_ADDRESS	0h	rh	w	Erroneous APB address in the I-Filter		
27dt7	reserved	000000h					
29dt28	APB_SIZE	0h	rh	w	00: Byte access 01: Halfword access 10: Word access 11: reserved		
30	APB_WRITE	0h	rh	w	0: Read access 1: Write access		
31	ERR_LOCK	0h	rh	w	Set to 1 by the HW if erroneous APB access is detected. This blocks further HW entries. The SW (PNB stack) must reset the bit to 0 to enable new entries.		

Register:	FILT_RELOAD_0				Address:	Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Load value of clock divider 0 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.						
Bit	Identifier	Reset	Attr.		Function / Description		
9dt0	FILT_RELOAD_0	0h	r	w	Load value of clock divider 0 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.		
31dt10	reserved	000000h					

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Register:	FILT_RELOAD_1				Address:	10h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Load value of clock divider 1 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.						
Bit	Identifier	Reset	Attr.		Function / Description		
9dt0	FILT_RELOAD_1	0h	r	w	Load value of clock divider 1 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.		
31dt10	reserved	000000h					

Register:	FILT_RELOAD_2				Address:	14h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Load value of clock divider 2 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.						
Bit	Identifier	Reset	Attr.		Function / Description		
9dt0	FILT_RELOAD_2	0h	r	w	Load value of clock divider 2 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.		
31dt10	reserved	000000h					

Register:	FILT_RELOAD_3				Address:	18h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Load value of clock divider 3 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.						
Bit	Identifier	Reset	Attr.		Function / Description		
9dt0	FILT_RELOAD_3	0h	r	w	Load value of clock divider 3 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.		
31dt10	reserved	000000h					

Register:	FILT_RELOAD_4				Address:	1Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Load value of clock divider 4 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.						

Bit	Identifier	Reset	Attr.		Function / Description
9dt0	FILT_RE-LOAD_4	0h	r	w	Load value of clock divider 4 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.
31dt10	reserved	000000h			

Regis-ter:	FILT_RELOAD_5			Address:	20h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Load value of clock divider 5 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.					

Bit	Identifier	Reset	Attr.		Function / Description
9dt0	FILT_RE-LOAD_5	0h	r	w	Load value of clock divider 5 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.
31dt10	reserved	000000h			

Regis-ter:	FILT_RELOAD_6			Address:	24h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Load value of clock divider 6 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.					

Bit	Identifier	Reset	Attr.		Function / Description
9dt0	FILT_RE-LOAD_6	0h	r	w	Load value of clock divider 6 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.
31dt10	reserved	000000h			

Regis-ter:	FILT_RELOAD_7			Address:	28h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Load value of clock divider 7 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.					

Bit	Identifier	Reset	Attr.		Function / Description
9dt0	FILT_RE-LOAD_7	0h	r	w	Load value of clock divider 7 The division factor reduced by 1 is entered here. Load value 0x000 is converted to 0x001 by the HW.
31dt10	reserved	000000h			

Copyright © Siemens AG 2023. All rights reserved

Register:	FILT_DELAY_0					Address:	2Ch	
Bits:	31dt0		Reset value:	FFFFFFFFh		Attributes:	r	w
Description:	Selection of clock source for input signals 0 to 7							
Bit	Identifier		Reset	Attr.		Function / Description		
3dt0	IN_DELAY_0		Fh	r	w	Selection of clock source for input signal 0: 0x0: CLOCK0 0x1: CLOCK1 0x2: CLOCK2 0x3: CLOCK3 0x4: CLOCK4 0x5: CLOCK5 0x6: CLOCK6 0x7: CLOCK7 0x8 - 0xE: no filtering, only synchronization 0xF: no filtering and no synchronization, only forwarding		
7dt4	IN_DELAY_1		Fh	r	w	Selection of clock source for input signal 1: 0x0: CLOCK0 0x1: CLOCK1 0x2: CLOCK2 0x3: CLOCK3 0x4: CLOCK4 0x5: CLOCK5 0x6: CLOCK6 0x7: CLOCK7 0x8 - 0xE: no filtering, only synchronization 0xF: no filtering and no synchronization, only forwarding		
11dt8	IN_DELAY_2		Fh	r	w	Selection of clock source for input signal 2: 0x0: CLOCK0 0x1: CLOCK1 0x2: CLOCK2 0x3: CLOCK3 0x4: CLOCK4 0x5: CLOCK5 0x6: CLOCK6 0x7: CLOCK7 0x8 - 0xE: no filtering, only synchronization 0xF: no filtering and no synchronization, only forwarding		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

15dt12	IN_DELAY_3	Fh	r	w	Selection of clock source for input signal 3: 0x0: CLOCK0 0x1: CLOCK1 0x2: CLOCK2 0x3: CLOCK3 0x4: CLOCK4 0x5: CLOCK5 0x6: CLOCK6 0x7: CLOCK7 0x8 - 0xE: no filtering, only synchronization 0xF: no filtering and no synchronization, only forwarding
19dt16	IN_DELAY_4	Fh	r	w	Selection of clock source for input signal 4: 0x0: CLOCK0 0x1: CLOCK1 0x2: CLOCK2 0x3: CLOCK3 0x4: CLOCK4 0x5: CLOCK5 0x6: CLOCK6 0x7: CLOCK7 0x8 - 0xE: no filtering, only synchronization 0xF: no filtering and no synchronization, only forwarding
23dt20	IN_DELAY_5	Fh	r	w	Selection of clock source for input signal 5: 0x0: CLOCK0 0x1: CLOCK1 0x2: CLOCK2 0x3: CLOCK3 0x4: CLOCK4 0x5: CLOCK5 0x6: CLOCK6 0x7: CLOCK7 0x8 - 0xE: no filtering, only synchronization 0xF: no filtering and no synchronization, only forwarding
27dt24	IN_DELAY_6	Fh	r	w	Selection of clock source for input signal 6: 0x0: CLOCK0 0x1: CLOCK1 0x2: CLOCK2 0x3: CLOCK3 0x4: CLOCK4

Copyright © Siemens AG 2023. All rights reserved

					0x5: CLOCK5 0x6: CLOCK6 0x7: CLOCK7 0x8 - 0xE: no filtering, only synchronization 0xF: no filtering and no synchronization, only forwarding
31dt28	IN_DELAY_7	Fh	r	w	Selection of clock source for input signal 7: 0x0: CLOCK0 0x1: CLOCK1 0x2: CLOCK2 0x3: CLOCK3 0x4: CLOCK4 0x5: CLOCK5 0x6: CLOCK6 0x7: CLOCK7 0x8 - 0xE: no filtering, only synchronization 0xF: no filtering and no synchronization, only forwarding

Register:	FILT_DELAY_1				Address:	30h	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	IN_DELAY_8	Fh	r	w	see IN_DELAY_0		
7dt4	IN_DELAY_9	Fh	r	w	see IN_DELAY_0		
11dt8	IN_DELAY_10	Fh	r	w	see IN_DELAY_0		
15dt12	IN_DELAY_11	Fh	r	w	see IN_DELAY_0		
19dt16	IN_DELAY_12	Fh	r	w	see IN_DELAY_0		
23dt20	IN_DELAY_13	Fh	r	w	see IN_DELAY_0		
27dt24	IN_DELAY_14	Fh	r	w	see IN_DELAY_0		
31dt28	IN_DELAY_15	Fh	r	w	see IN_DELAY_0		

Register:	FILT_DELAY_2				Address:	34h	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	IN_DELAY_16	Fh	r	w	see IN_DELAY_0		
7dt4	IN_DELAY_17	Fh	r	w	see IN_DELAY_0		

11dt8	IN_DELAY_18	Fh	r	w	see IN_DELAY_0
15dt12	IN_DELAY_19	Fh	r	w	see IN_DELAY_0
19dt16	IN_DELAY_20	Fh	r	w	see IN_DELAY_0
23dt20	IN_DELAY_21	Fh	r	w	see IN_DELAY_0
27dt24	IN_DELAY_22	Fh	r	w	see IN_DELAY_0
31dt28	IN_DELAY_23	Fh	r	w	see IN_DELAY_0

Register:	FILT_DELAY_3				Address:	38h	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	IN_DELAY_24	Fh	r	w	see IN_DELAY_0		
7dt4	IN_DELAY_25	Fh	r	w	see IN_DELAY_0		
11dt8	IN_DELAY_26	Fh	r	w	see IN_DELAY_0		
15dt12	IN_DELAY_27	Fh	r	w	see IN_DELAY_0		
19dt16	IN_DELAY_28	Fh	r	w	see IN_DELAY_0		
23dt20	IN_DELAY_29	Fh	r	w	see IN_DELAY_0		
27dt24	IN_DELAY_30	Fh	r	w	see IN_DELAY_0		
31dt28	IN_DELAY_31	Fh	r	w	see IN_DELAY_0		

Register:	FILT_DELAY_4				Address:	3Ch	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	IN_DELAY_32	Fh	r	w	see IN_DELAY_0		
7dt4	IN_DELAY_33	Fh	r	w	see IN_DELAY_0		
11dt8	IN_DELAY_34	Fh	r	w	see IN_DELAY_0		
15dt12	IN_DELAY_35	Fh	r	w	see IN_DELAY_0		
19dt16	IN_DELAY_36	Fh	r	w	see IN_DELAY_0		
23dt20	IN_DELAY_37	Fh	r	w	see IN_DELAY_0		
27dt24	IN_DELAY_38	Fh	r	w	see IN_DELAY_0		
31dt28	IN_DELAY_39	Fh	r	w	see IN_DELAY_0		

Register:	FILT_DELAY_5				Address:	40h	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	r	w

Description:					
Bit	Identifier	Reset	Attr.		Function / Description
3dt0	IN_DELAY_40	Fh	r	w	see IN_DELAY_0
7dt4	IN_DELAY_41	Fh	r	w	see IN_DELAY_0
11dt8	IN_DELAY_42	Fh	r	w	see IN_DELAY_0
15dt12	IN_DELAY_43	Fh	r	w	see IN_DELAY_0
19dt16	IN_DELAY_44	Fh	r	w	see IN_DELAY_0
23dt20	IN_DELAY_45	Fh	r	w	see IN_DELAY_0
27dt24	IN_DELAY_46	Fh	r	w	see IN_DELAY_0
31dt28	IN_DELAY_47	Fh	r	w	see IN_DELAY_0

Register:	FILT_DELAY_6			Address:	44h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:						
Bit	Identifier	Reset	Attr.		Function / Description	
3dt0	IN_DELAY_48	Fh	r	w	see IN_DELAY_0	
7dt4	IN_DELAY_49	Fh	r	w	see IN_DELAY_0	
11dt8	IN_DELAY_50	Fh	r	w	see IN_DELAY_0	
15dt12	IN_DELAY_51	Fh	r	w	see IN_DELAY_0	
19dt16	IN_DELAY_52	Fh	r	w	see IN_DELAY_0	
23dt20	IN_DELAY_53	Fh	r	w	see IN_DELAY_0	
27dt24	IN_DELAY_54	Fh	r	w	see IN_DELAY_0	
31dt28	IN_DELAY_55	Fh	r	w	see IN_DELAY_0	

Register:	FILT_DELAY_7			Address:	48h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:						
Bit	Identifier	Reset	Attr.		Function / Description	
3dt0	IN_DELAY_56	Fh	r	w	see IN_DELAY_0	
7dt4	IN_DELAY_57	Fh	r	w	see IN_DELAY_0	
11dt8	IN_DELAY_58	Fh	r	w	see IN_DELAY_0	
15dt12	IN_DELAY_59	Fh	r	w	see IN_DELAY_0	
19dt16	IN_DELAY_60	Fh	r	w	see IN_DELAY_0	
23dt20	IN_DELAY_61	Fh	r	w	see IN_DELAY_0	
27dt24	IN_DELAY_62	Fh	r	w	see IN_DELAY_0	

Copyright © Siemens AG 2023. All rights reserved

31dt28	IN_DELAY_63	Fh	r	w	see IN_DELAY_0

Register:	FILT_DELAY_8			Address:	4Ch	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:						
Bit	Identifier	Reset	Attr.		Function / Description	
3dt0	IN_DELAY_64	Fh	r	w	see IN_DELAY_0	
7dt4	IN_DELAY_65	Fh	r	w	see IN_DELAY_0	
11dt8	IN_DELAY_66	Fh	r	w	see IN_DELAY_0	
15dt12	IN_DELAY_67	Fh	r	w	see IN_DELAY_0	
19dt16	IN_DELAY_68	Fh	r	w	see IN_DELAY_0	
23dt20	IN_DELAY_69	Fh	r	w	see IN_DELAY_0	
27dt24	IN_DELAY_70	Fh	r	w	see IN_DELAY_0	
31dt28	IN_DELAY_71	Fh	r	w	see IN_DELAY_0	

Register:	FILT_DELAY_9			Address:	50h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:						
Bit	Identifier	Reset	Attr.		Function / Description	
3dt0	IN_DELAY_72	Fh	r	w	see IN_DELAY_0	
7dt4	IN_DELAY_73	Fh	r	w	see IN_DELAY_0	
11dt8	IN_DELAY_74	Fh	r	w	see IN_DELAY_0	
15dt12	IN_DELAY_75	Fh	r	w	see IN_DELAY_0	
19dt16	IN_DELAY_76	Fh	r	w	see IN_DELAY_0	
23dt20	IN_DELAY_77	Fh	r	w	see IN_DELAY_0	
27dt24	IN_DELAY_78	Fh	r	w	see IN_DELAY_0	
31dt28	IN_DELAY_79	Fh	r	w	see IN_DELAY_0	

Copyright © Siemens AG 2023. All rights reserved

2.3.14.3 ARM926 Watchdog

2.3.14.3.1 Overview

The ARM926 watchdog primarily consists of two counters with different widths (32 and 36 bit) that count down from a parameterized initial value.

The watchdog has the following principal characteristics:

- The clock pulse supply is 125 MHz (APB clock).
- One output signal of the Counter0 watchdog (XWD_OUT0) is output at an alternate function of a GPIO pin.
- After the watchdog Counter0 expires an interrupt (WD_INT: WD_INT_ARM926) is generated (IRQ27, see chapter 2.3.5.9).
- After the watchdog Counter1 expires (XWD_OUT1) the 'XRES_ARM926_WD' is generated (refer to the ERTEC 200P-3 data sheet).

2.3.14.3.2 Block diagram

Figure 37: Block diagram ARM926 Watchdog

Counter0

Counter0 is a 32-bit wide counter output that counts to 0 starting with the value passed in the 'RELD0(_LOW/_HIGH)' register with the clock pulse from the CLK_WD pin. The watchdog is (re)started with Run/xStop_Z0=1 and, when required, stopped with Run/xStop_Z0=0.

Copyright © Siemens AG 2023. All rights reserved

The output is XWD_OUT0=0 when the watchdog is stopped. If the watchdog has been started and Counter0 ≠ 0, then the output is XWD_OUT0=1, otherwise XWD_OUT0=0.

Note that in the case that Counter0 is preloaded with 0 and is started, the output XWD_OUT0 is also 1 for one clock cycle. Probably this is not a use case, this is stated only to get a whole picture of the circuit.

If the input is Load=1, the watchdog will be triggered, i.e., the Counter0 will be loaded with the value contained in the 'RELD0(_LOW/_HIGH)' register and will continue counting from this value.

The 'WDOG0' register can be used to fetch the current value of the counter.

The Status_Counter0 output is active only when Run/xStop_Z0=1 and Counter0 has expired.

An expired Counter0, i.e., Counter0 decremented to 0, initiates an interrupt (WD_INT). For a Timing Diagram see Figure 38.

Counter1

Counter1 is a 36-bit wide counter that counts to 0 starting with the value passed in the 'RELD1(_LOW/_HIGH)' register with the clock pulse from CLK_WD pin. The 'RELD1(_LOW/_HIGH)' register contains only the high order 32 bits of the counter. The four low order bits of the counter are always loaded with the value 0x0.

The watchdog is (re)started with Run/xStop_Z1=1 and, when required, stopped with Run/xStop_Z1=0. If Counter1 is stopped or it is started but has not yet attained the value 0, then the output is XWD_OUT1=1. The output is XWD_OUT1=0, when Counter1 has expired.

If the input is Load=1, the watchdog will be triggered, i.e., the upper 32 bits of Counter1 will be loaded with the value contained in the 'RELD1(_LOW/_HIGH)' register and the watchdog will continue counting from this value to zero.

The 'WDOG1' register can be used to fetch the current value of the counter (only the upper 32 bits).

The Status_Counter1 output is active only when Run/xStop_Z1=1 and Counter1 has expired.

CTRL/Status register (X=0,1)

The register is 32-bit wide and contains data only in the lower 16 bits. The upper 16 bits contain a special signature (see also chapter 2.3.14.3.4).

- Run/xStop_ZX-Bits:
  Starts and stops the Counter0.
- Load:
  The Load signal acts simultaneously on both counters (provided they have been enabled). The Load signal from the register is synchronized for the counter side. An increasing edge at the load input of the counter loads the CounterX with the value from RELD0(_LOW/_HIGH) or RELD1(_LOW/_HIGH).
  The software does not need to reset the load bit.
- Status_counterX:
  This bit is a single status bit present separately for each counter. If the bit is set, then the associated counter has attained the value 0. If Run/xStop_ZX=0, then the associated status bit always reads logical 0.

RELD0_LOW & RELD0_HIGH

These two 32-bit registers contain user data only in the lower 16 bits. Each of the upper 16 bits is reserved for a special signature (see also chapter 2.3.14.3.4). Thus, the 32-bit value for the counter consists of the lower halves of both registers. The register values must be changed only when the counter is stopped (Run/xStop_ZX=0).

Copyright © Siemens AG 2023. All rights reserved

RELD1_LOW & RELD1_HIGH

These two 32-bit registers contain user data only in the lower 16 bits. Each of the upper 16 bits is reserved for a special signature (see also chapter 2.3.14.3.4). Thus, the 32-bit value for the counter consists of the lower halves of both registers. Only the upper 32 bits of the reload value can be specified for these counters – the lower 4 bits are always logical 0. The register values must be changed only when the counter is stopped (Run/xStop_ZX=0).

WDOG0 & WDOG1

These two registers can be used to read the current values of the two counters. Only the upper 32 bits of the current counter value can be read for Counter1. The content of the two registers will be updated after each increasing edge of the counter cycle clock (CLK_WD). Thus, without requiring the wait state, a read access to these registers always returns the contents of the two counters after the last recognized counter cycle.

2.3.14.3.3 Signal waveforms

Figure 38: XWD_OUT0/WD_INT signal sequence

After a reset, the XWD_OUT0 output is initially logical 0. The XWD_OUT0 output is inactive (=1) only after Counter0 has been started (Run/xStop_Z0=1) and provided the counter value ≠ 0.

If after the start or the last retrigger pulse Counter0 expires after time t1, then

- XWD_OUT0 output becomes active (=0),
- "Status_Counter0" status bit is set and
- INT_WD interrupt signal becomes active (the increasing edge for INT_WD initiates the interrupt).

A retrigger pulse (Load=1 & Reload-value ≠ 0) or stopping the counter (Run/xStop_Z0=0) causes the status bit and the interrupt signal to be reset.
The XWD_OUT0 output assumes the value logical 1 again only when Counter0 is subsequently restarted with a value ≠ 0.

$$t_1 = (RELD0 + 1) \times T_{REF_CLK_IN}$$

t_1:	Time until the counter expires
RELD0:	Decimal value of the reload value for Counter0
$T_{REF_CLK_IN}$:	Period duration of the system cycle clock (8 ns).
$T_{REF_CLK_IN}$ =	125 MHz:
	t_{1MIN} = 0 s, t_{1MAX} = 34.36 s, Interval = 8 ns

WDOG1: Status / counter value for Counter1
FFFF_FFFF = Reset value
Reload = Load value from RELD1 register (reload value)
Count = Countdown
Zero = Counter1 has expired

Figure 39: XWD_OUT1 signal sequence

After a reset, the XWD_OUT1 output is initially logical 1. The XWD_OUT1 output becomes logical 0 and the "Status_Counter1" status bit is set only after Counter1 has been started (Run/xStop_Z1=1) and the time t2 has expired after the start or the last retrigger pulse (Load=1).

A retrigger pulse (Load=1 & Reload-value ≠ 0) or stopping the counter (Run/xStop_Z1=0) causes the status bit and the XWD_OUT1 output to be reset again.

$$t_2 = (RELD1 \times 16 + 1) \times T_{REF_CLK_IN}$$

t_2:	Time until the counter expires
RELD1:	Decimal value of the reload value for Counter1
$T_{REF_CLK_IN}$:	Period duration of the system cycle clock (8 ns)
$T_{REF_CLK_IN}$ =	125 MHz:
	t_{2MIN} = 0 s, t_{2MAX} = 549.76 s, Interval = 128 ns

Copyright © Siemens AG 2023. All rights reserved

2.3.14.3.4 Write protection of the Watchdog register

If the Watchdog Control/Status register or one of the Watchdog Reload registers is to be written, a defined bit combination in the upper 16 bits (key bits) must be written simultaneously. The key bits are 9876h (arbitrary). The read access returns the value 0000h in the upper 16 bits.

2.3.14.3.5 Starting the Watchdog

The following sequence describes the initialization phase which at first has to be performed by software:

- Write a proper value for 'RELD0/1 (_LOW/_HIGH)'.
- Load the values by writing to the Load register.
- Activate the watchdog by writing '1' to the Run/xStop_Z0/1 bits in the 'WD_CTRL_STATUS' register.

2.3.14.3.6 Notes

- The ERTEC 200P watchdog cannot be used by architectures with an external PCI host (PCI bridge outside the ERTEC 200P), because XWD_OUT0=0 (counter0 expired) cannot be signaled to the external host over the PCI bridge.

  In this use case the watchdog function must be realized by the external host. The ERTEC 200P must increment a counter in the external PCI host memory and the host must check this counter to see whether the ERTEC 200P is still running.
- By coupling an external host over the XHIF interface directly, the watchdog in the ERTEC 200P can be used. The XWD_OUT0=0 (counter0 expired) can be signaled to the external host over a GPIO port.
- For embedded single processor systems with ERTEC 200P the watchdog function is mandatory.

2.3.14.3.7 Address mapping

Start-Address	End-Address	Modul/Memory-Name
0h	1Bh	wdog

Module	Register/Memory	Read	Write	Access	Address
/wdog				32-Bit	
	/WD_CTRL_STATUS	(r)(h)	(w)	32-Bit	0h
	/RELD0_LOW	(r)	w	32-Bit	4h
	/RELD0_HIGH	(r)	w	32-Bit	8h
	/RELD1_LOW	(r)	w	32-Bit	Ch
	/RELD1_HIGH	(r)	w	32-Bit	10h
	/WDOG0	rh		32-Bit	14h
	/WDOG1	rh		32-Bit	18h

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.3.8 Register description

Module: wdog

Register:	WD_CTRL_STATUS			Address:	0h
Bits:	31dt0	Reset value:	0h	Attributes:	(r)(h)(w)
Description:	Control/Status register. Configuration and control bits for the watchdog.				

Bit	Identifier	Reset	Attr.		Function / Description
0	Run_xStop_Z0	0h	r	w	Enable/disable watchdog counter 0: 0: Watchdog counter 0 disabled 1: Watchdog counter 0 enabled **Note**: If this bit = 0, the XWDOUT0 output of the IDC32 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	0h	r	w	Enable/disable watchdog counter 1: 0: Watchdog counter 1 disabled 1: Watchdog counter 1 enabled **Note**: If this bit = 0, the XWDOUT1 output of the IDC32 is passive (1) and the status bit of counter 1 (bit 4) is 0.
2	Load_Trigger	0h	r	w	Watchdog trigger (load watchdog counters 0 and 1 with the value of the Reload registers): 0: Do not trigger watchdog 1: Trigger watchdog Although this bit can be read back, it acts only during a write. To trigger the watchdog counters, it suffices to write a 1 to this bit- no 0/1 edge is required. The trigger signal acts on both watchdog counters.
3	Status_Counter0	0h	rh		Watchdog status counter 0 (write is ignored): 0: Watchdog counter 0 has not expired 1: Watchdog counter 0 has expired **Note**: This bit then can only be read as '1' when RUN/xStop_Z0 is active (1).
4	Status_Counter1	0h	rh		Watchdog status counter 1 (write is ignored): 0: Watchdog counter 1 has not expired 1: Watchdog counter 1 has expired **Note**: This bit then can only be read as '1' when RUN/xStop_Z1 is active (1).
15dt5	reserved	000h			
31dt16	Key_bits	0h		w	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-4 of this register will be written, otherwise the operation has no effect.

Copyright © Siemens AG 2023. All rights reserved

Register:	RELD0_LOW				Address:	4h	
Bits:	31dt0	Reset value:	FFFFh		Attributes:	(r)	w
Description:	Reload Register 0_low. Reload value for bits 15:0 of watchdog counter 0.						
Bit	Identifier	Reset	Attr.		Function / Description		
15dt0	Reload0	FFFFh	r	w	Reload value for bits 15:0 of watch-dog counter 0		
31dt16	Key_bits	0h		w	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-15 of this register will be written, otherwise the operation has no effect.		

Register:	RELD0_HIGH				Address:	8h	
Bits:	31dt0	Reset value:	FFFFh		Attributes:	(r)	w
Description:	Reload Register 0_high. Reload value for bits 31:16 of watchdog counter 0.						
Bit	Identifier	Reset	Attr.		Function / Description		
15dt0	Reload0	FFFFh	r	w	Reload value for bits 31-16 of watch-dog counter 0		
31dt16	Key_bits	0h		w	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-15 of this register will be written, otherwise the operation has no effect.		

Register:	RELD1_LOW				Address:	Ch	
Bits:	31dt0	Reset value:	FFFFh		Attributes:	(r)	w
Description:	Reload Register 1_low. Reload value for bits 19:4 of watchdog counter 1.						
Bit	Identifier	Reset	Attr.		Function / Description		
15dt0	Reload1	FFFFh	r	w	Reload value for bits 19:4 of watch-dog counter 1.		
31dt16	Key_bits	0h		w	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-15 of this register will be written, otherwise the operation has no effect.		

Register:	RELD1_HIGH				Address:	10h	
Bits:	31dt0	Reset value:	FFFFh		Attributes:	(r)	w
Description:	Reload Register 1_high. Reload value for bits 35:20 of watchdog counter 1.						
Bit	Identifier	Reset	Attr.		Function / Description		
15dt0	Reload1	FFFFh	r	w	Reload value for bits 35:20 of watch-dog counter 1		
31dt16	Key_bits	0h		w	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-15 of this register will be written, otherwise the operation has no effect.		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Register:	WDOG0				Address:	14h	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	rh	
Description:	Watchdog value 0. Value of watchdog counter 0.						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	WDOG0	FFFFFFFFh	rh		Watchdog value 0. Value of watch-dog counter 0.		

Register:	WDOG1				Address:	18h	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	rh	
Description:	Watchdog value 1. Value of watchdog counter 1.						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	WDOG1	FFFFFFFFh	rh		Watchdog value 1. Value of watch-dog counter 1.		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.4 Timer 0 – 5

2.3.14.4.1 Overview

In ERTEC 200P there are six independent timers integrated which serve for monitoring various software routines. To each of these software timers is assigned an own interrupt.

Access to the timer registers is carried out at word limits (32 bits).

Summary of the timer functions

- Bit down-counter
- Programmable bit prescaler connectable (separately for each timer)
- Loadable/reloadable
- Start, stop and continue function
- Interrupt when counter value '0' has been reached
- Read register for reading out the counter value
- Three event registers for event-driven storing of the counter value:
  - Two external HW events (selectable from the EXTERNAL_INPUTS signals of the TIMER_TOP module) for storing in two HW event registers
  - One internal SW event for storing in one SW event register
- Timers are cascadable (prerequisite: timer output of the low-order counter(s) is fed back to the TIMER_TOP module via the EXTERNAL_INPUTS signals and is selected as GATE_TRIG signal at the high-order timer)
- Clocking of the counter can be done with the input clock CLK_TIMT = APB_Takt and via the SW (internal gate/trigger signal) as well as via an input signal (external gate/trigger signal)
- Triggering of the counter (loading) can be done via the SW (internal gate/trigger signal) as well as via an input signal (external gate/trigger signal)
- Clock output for supplying further modules (bit divider for CLK_TIMT = APB_Clock).

In the zero crossing, an output pulse (TIMER_OUT) is generated which can be evaluated as interrupt. The TIMER_TOP module contains six TIMER and multiplexer submodules each for selecting the trigger signals for the individual timers, and functions expanding over all timers (e.g., synchronous releasing of all timers, address coding). The TIMER submodule contains the basic timer function (e.g., counter, prescaler, load register for counter). The distribution of the timer functions to the TIMER and TIMER_TOP modules is supposed to simplify the re-usability in different ASICs.

The TIMER and TIMER_TOP modules are described in detail in the following sections.

Copyright © Siemens AG 2023. All rights reserved

2.3.14.4.2 TIMER_TOP functionality

Block diagram

Figure 40: TIMER_TOP block diagram

The TIMER_TOP submodule contains the wiring of the individual TIMER modules, the 'GATE_TRIG_CONTROL_REG' registers, the multiplexer and timer MUX registers for the input signals INT_GATE_TRIG_TIM, EXT_GATE_TRIG_TIM, EVENT1, EVENT2, and the clock divider.

The TIMER_TOP module consists of the following functional units:

- 'GATE_TRIG_CONTROL_REG' register for triggering SW gate/trigger signals and for releasing the count pulses for the timers
- SW event trigger register for storing the current counter values in the Int_Event registers
- Six timers
- Multiplexers and the corresponding registers for selecting the sources for the inputs of the TIMER submodules

Copyright © Siemens AG 2023. All rights reserved

- Clock divider and the corresponding register for activating the clock divider.

The TIMER_TOP module has an APB interface (AMBA 2.0). The access width has to be 32 bits. Different accesses lead to a faulty writing of the register of the TIMER_TOP module.

Gate Count Register

The TIMER_TOP module has a 'GATE_TRIG_CONTROL_REG' register in which a SW gate signal can be set/reset for each TIMER module (= INT_GATE_TRIG_TIM inputs of the TIMERs). The effect depends on the selected mode.

Moreover, the count clock can be released/blocked for each TIMER module (= CLK_EN inputs of the TIMERs). By writing on this register, all TIMER modules can be started/stopped synchronously.

NOTICE

If the SW gate signal is changed while simultaneously the count clock operating mode is released/blocked, the following behavior applies to all operating modes in which the SW gate signal is relevant:

- When blocking the counting pulse:
  SW gate signal change has no effect anymore.
- When releasing the counting pulse:
  SW gate signal change is effective in the next count/load clock, depending on the selected operating mode. This also applies to operating modes in which the edge of the SW gate signal is relevant.

Software event trigger register

Writing a ’1’ into the bit “n” of the SW event trigger register triggers a positive edge at the INT_EV input of the TIMER module “n” and leads to a storing of the counter value of the TIMER module “n” in the Int_Event register of the TIMER module “n”.

The ’1’ is not stored (read=0).

By writing on this register, the current counter values of all timers can be synchronously stored in the SW event registers of the TIMER modules.

Multiplexer/TIM_MUX register

The TIMER_TOP submodule has own multiplexers and registers for their control for each timer, which serve for selecting the sources for the inputs of the TIMER modules EXT_GATE_TRIG_TIM, EVENT1, EVENT2. The EXTERNAL_INPUTS (15:0) inputs of the TIMER_TOP module are available as sources.

The registers for the multiplexers are designed in such a way that the numbers of the select signals entered there select the corresponding number of the input signal.

Each TIMER module can be assigned own sources for the inputs EXT_GATE_TRIG, EVENT1, EVENT2.

Thus, versatile measuring tasks can be configured (for examples, see chapter 2.3.5.8.3).

Clock divider/clock divider register

The TIMER_TOP module has a clock divider for providing a clock for external modules, which is symmetrical with the input clock CLK_TIMT.
The clock divider is 8 bits wide (NUM_OF_CLKDIV_BITS). The clock divider value is set by writing on the bits in the 'CLOCK_DIVIDER_REG' register. A writing of the register

Copyright © Siemens AG 2023. All rights reserved

'CLOCK_DIVIDER_REG' loads the clock divider with the clock divider value. The current value of the clock divider is non-readable.

The clock divider output CLK_OUT is ’0’ as long as the clock dividing has not been activated.

The clock dividing is activated by setting the CLK_DIV_EN bit to ’1’ and by writing the clock divider value CLOCK_DIVIDER_VALUE with a value unequal to ’0’ (in the register 'CLOCK_DIVIDER_REG').

When clock dividing is active, the following applies:

- The clock divider output CLK_OUT is ’0’ as long as the clock divider has a value smaller than CLKDIV_VALUE/2, otherwise ’1’.
- The divided clock CLK_OUT is symmetrical if the clock divider value is odd-numbered.
- The clock divider is implemented as a down-counter which counts down cyclically from the clock divider value CLKDIV_VALUE.

The clock divider divides the input clock CLK_TIMT by the factor 1 / (CLKDIV_VALUE+1).
Exception: A clock divider value of 0 supplies CLK_OUT=0.

Possible values for the clock divider value CLKDIV_VALUE: ’0’ to ’2^8-1’. At 125 MHz there are therefore output frequencies of 125 MHz / 2^8 – 125 MHz / 2 possible.

NOTICE

Writing a new clock divider value CLOCK_DIVIDER_VALUE may have the effect that CLK_OUT runs with a wrong clock period once.

Timer cascading

The TIMER modules are cascadable under certain conditions. The items stated under “Cascading of TIMER modules” have to be taken into account for that.

Copyright © Siemens AG 2023. All rights reserved

2.3.14.4.3 Timer module

Timer block diagram

Figure 41: TIMER block diagram

The submodules TIMER_TOP and TIMER operate with the operating clock CLK_TIMT or CLK. CLK_TIMT is wired to CLK. The operating clock is 125 MHz and the circuit is designed synchronous with that clock. The reset input of the submodules TIMER_TOP and TIMER is used as asynchronous reset; a synchronization of the reset input has not been implemented.

Each TIMER submodule consists of the following functional units:

- Counter/count register (counter and register for reading the current counter value)
- Prescaler/prescaler register (prescaler and register for prescaler load value)
- Load register (load value of the counter)
- Mode register (setting the operating mode)
- Event register for storing the counter values dependent on events (register Ext_Event_1, Ext_Event_2, Int_Event)

Counter/count register

Each TIMER module contains a 32-bit counter which counts down from a start value. The counter stops when the value ‘0’ has been reached or is automatically reloaded if the reload function has been activated in the mode register.

The following sources can be selected as count/load clock of the counter:

Copyright © Siemens AG 2023. All rights reserved

- Input clock CLK_TIMT (125MHz = APB clock = setting after reset)
- Input clock CLK_TIMT divided by an 8-bit prescaler
- External gate/trigger signal (input signal INT_GATE_TRIG_TIM or EXT_GATE_TRIG_TIM, synchronized with input clock CLK_TIMT (two flip flops)).

The software can read the current value of the counter in the count register.

The counter range is up to 26 days (Input Clk = 125MHz, 8bit Clock Divider, 8bit Prescaler, 32bit Timer).

Prescaler/prescaler register

In accordance with the basic setting, the counter is operated with CLK_TIMT (125MHz = APB clock). Nevertheless, each counter can be connected to an 8-bit prescaler (independent parameterization for each timer) so that the runtime of the individual TIMER modules can be increased accordingly.

Each TIMER module contains an 8-bit prescaler which counts down cyclically from a start value, and a prescaler register which contains the start value for the prescaler (= PRESCALER_VAL).

The prescaler counts/loads with the internal clock CLK_TIMT.

The prescaler is deactivated, if:

- Timer input CLK_EN=0
- Gate_effect=0 and external gate/trigger signal has passive level

The current counter value of the prescaler is non-readable. There is no signal for the prescaler indicating the counter value ’0’. The prescaler register is 8 bits wide (NUM_OF_PRESCALER_BITS). The prescaler divides the counter frequency by the factor 1 / (PRESCALER_VALUE+1).

Possible values for the prescaler register: ’0’ to ’2^8 - 1’

Load register

The load register contains the load value (start value) of the counter. Writing to the load register with a new value causes the takeover of the load value into the counter, if the following conditions are met:

- Timer input CLK_EN=’1’ during writing
- Bit DIS_RLD_WHEN_WR_LDREG in the mode register =’0’ during writing

The time of takeover of the load value into the counter also depends on the operating mode of the timer and is carried out with the count/load clock of the counter.

Mode register

The operating mode of the TIMER module is set via the following bits in the mode register:

- Init_bit
- Clk_input select
- Reload_disable
- DIS_RLD_WHEN_WR_LDREG
- Ext_gate_trig_enable
- Gate_polarity
- Gate_effect

Copyright © Siemens AG 2023. All rights reserved

- Timer_out_polarity
- Event1/2_control
- Event1/2_Inversion

The **Init_bit** is an initialization bit which resets the TIMER module, loads (restarts) the counter with the value from the load register and loads the prescaler with the value from the prescaler register. It deletes all values in the event registers (see below) and resets the edge evaluation of the event inputs and the EXT_GATE_TRIG signal. After having been set by the software, this bit is automatically deleted again, a reading of the bit by the software always results in the value ’0’. The definition is carried out in accordance with the following table:

Init_bit	Function
0	Bit Init_bit not active (no initialization)
1	Bit Init_bit active (initialization)

The counter **counts/loads** with the count/load clock. The selection is done by the mode register bit **Clk_input_select** and the value of the prescaler register in accordance with the following table:

Clk_input_select	Function
0	Count/load clock = rising edge of CLK_TIMT; The counting/loading of the counter with the CLK_TIMT edge only takes place if the prescaler value is ’0’
1	Count/load clock = external gate/trigger signal (edge evaluation with CLK_TIMT is valued as count/load enable signal)

The **Reload_disable** bit determines if the counter stops the counting process when it has reached the value ’0’ (single mode) or if the counter is automatically loaded = newly started with the reload value from the load register when the value ’0’ has been reached (reload mode). The definition is carried out in accordance with the following table:

Reload_disable	Function
0	Reload mode active
1	Single mode active

The **DIS_RLD_WHEN_WR_LDREG** bit in the mode register determines the effect of the writing of a new value on the load register:

DIS_RLD_WHEN _WR_LDREG	Function
0	Writing a load value on the load register has the effect that the load value is taken over into the counter with the next counter edge or the next gate/trigger signal (depending on the setting in the mode register).

Copyright © Siemens AG 2023. All rights reserved

DIS_RLD_WHEN _WR_LDREG	Function
1	Writing a load value on the load register has the effect that the load value is taken over into the counter under the following conditions: • The counter has reached the value '0'. • Bit Reload_disable in the mode register is '1'. If these conditions are met, the takeover is carried out with the next counter edge or the next gate/trigger signal (depending on the setting in the mode register).

By setting the DIS_RLD_WHEN_WR_LDREG bit to '1', the takeover of the load register value is only possible during the zero crossing. This function is required in order to use the timers as phase-shifted cyclic timers.

One of the two input signals **EXT_GATE_TRIG** or **INT_GATE_TRIG** can be used:

- as gate signal for releasing/blocking the counter, or
- as trigger signal for triggering the counter, or
- as clock signal for clocking the counter.

The selection of the external gate/trigger signal is done by the mode register bit **Ext_gate_trig_enable** in accordance with the following table:

Ext_gate_trig_enable	Function
0	INT_GATE_TRIG = internal gate/trigger signal
1	EXT_GATE_TRIG = external gate/trigger signal

Differentiation between INT_GATE_TRIG and EXT_GATE_TRIG (internal and external gate/trigger signal):

- INT_GATE_TRIG is triggered by the software
- EXT_GATE_TRIG is triggered by an external hardware signal.

The **Gate_polarity** bit defines the active level or the active edge of the external gate/trigger signal in accordance with the following table:

Gate_polarity	Function
0	Level of the external gate/trigger signal is “high active” or rising edge is active edge
1	Level of the external gate/trigger signal is “low active” or falling edge is active edge

Copyright © Siemens AG 2023. All rights reserved

Together with the Clk_input_select bit, the **Gate_effect** bit defines the effect of the external gate/trigger signal in accordance with the following table.

Clk_input_select=0 (counter counts with CLK_TIMT as count/load clock dependent on the preselector value) - Gate/Trigger mode switching:

Gate_effect	Function
0	Gate mode: External gate/trigger signal has the effect of a gate (=gate mode) for the count/load clock and has to be active for counting
1	Trigger mode: Each active signal edge of the external gate/trigger signal causes the triggering of the counter with the reload value if the current counter value is unequal to the reload value or has no effect if the current counter value is equal to the reload value (=Trigger mode)

Note

Note on the Trigger mode:

If the active signal edge of the external gate/trigger signal occurs in the Trigger mode during CLK_EN=1 and then CLK_EN becomes '0' without the preselector value having gone to '0', the previous active signal edge only becomes effective when CLK_EN is '1' again and the preselector value is '0', i.e., the requirement does not get lost because of CLK_EN=0.

Clk_input_select=1 (counter counts with external gate/trigger signal as count/load clock (edge evaluation)) - Count/Toggle mode switching:

Gate_effect	Function
0	Count mode: External gate/trigger signal is the count/load clock (Count mode)
1	Toggle mode: External gate/trigger signal is the count/load clock and causes the switching between: • loading of the counter with the reload value (if the current counter value is unequal to the reload value) • down-counting (if the current counter value is equal to the reload value)

Thus, delays as well as time monitorings can be implemented.

Note

The loading of the counter can simultaneously mean the starting of the counter, namely if Reload_disable = 1 and Counter_value = 0 and load value #0.

The bit combination Gate_effect = 1 and Clk_input_select = 1 can be used to generate a symmetrical output signal out of an unsymmetrical input signal (reload value=1).

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

The TIM_OUT output of the TIMER module is active when the counter has the value ’0’ (non-saving), otherwise passive. The active level of the TIM_OUT output is defined by the **Timer_out_polarity** bit in accordance with the following table

Timer_out_polarity	Function
0	TIM_OUT = high active
1	TIM_OUT = low active

The storing of the current counter values in the event registers Ext_Event_1 and Ext_Event_2 is done dependent on the bits Event1_control, Event2_control, Event1_Inversion, Event2_Inversion in the mode register and dependent on the inputs EXT_EV_1, EXT_EV_2 in accordance with the following tables.

Bit Event1_control

Event1_control	Function
00	EXT_EV_1 input does not affect register Ext_Event_1 and register Ext_Event_2
01	Rising edge of the EXT_EV_1 input leads to the storing of the counter value in the register Ext_Event_1 *)
10	Falling edge of the EXT_EV_1 input leads to the storing of the counter value in the register Ext_Event_2 *)
11	Rising edge of the EXT_EV_1 input leads to the storing of the counter value in the register Ext_Event_1, falling edge of the EXT_EV_1 input leads to the storing of the counter value in the register Ext_Event_2 *)

*) The above table is valid if the Event1_Inversion bit is set to '0'; if the bit is set to '1', the “rising edge” and “falling edge” are to be exchanged.

Bit Event2_control

Event2_control	Function
00	EXT_EV_2 input does not affect register Ext_Event_1 and register Ext_Event_2
01	Rising edge of the EXT_EV_2 input leads to the storing of the counter value in the register Ext_Event_1 *)
10	Falling edge of the EXT_EV_2 input leads to the storing of the counter value in the register xt_Event_2 *)
11	Rising edge of the EXT_EV_2 input leads to the storing of the counter value in the register Ext_Event_1, the falling edge of the EXT_EV_2 input leads to the storing of the counter value in the register Ext_Event_2 *)

*) The above table is valid, if the Event2_Inversion bit is set to 0; if the bit is set to 1, the “rising edge” and “falling edge” are to be exchanged.

All combinations of the bits Event2_control, Event1_control in which the EXT_EV_1 input as well as the EXT_EV_2 input would cause a storing of the counter value in the same register are not permitted. If these combinations are set nevertheless, the counter value is not stored.

The registers Ext_Event_1 and Ext_Event_2 are readable and writable by the software; the setting of the registers after reset is ’0000h’.

Copyright © Siemens AG 2023. All rights reserved

If the condition for storing the current counter values in the event registers Ext_Event_1 and Ext_Event_2 occurs simultaneously with a writing of one of these registers, the writing is treated preferentially.

The storing of the current counter values in the event register Int_Event is triggered by a positive edge of the INT_EV input. The register is readable by the software and writable; the setting of the registers after reset is ’0000h’.

By storing the current counter values with the edges of the inputs EXT_EV_1, EXT_EV_2 into the event registers Ext_Event_1 and Ext_Event_2 it is possible to measure time intervals and periods between the inputs EXT_EV_1, EXT_EV_2. By storing the current counter value with the rising edge of the INT_EV input to the event register Int_Event it is possible to measure time intervals and periods between write accesses to the SW event trigger register.

The **loading of the counter** can be triggered in four different ways:

- Loading of the counter by writing the Init_bit in the mode register
- Loading of the counter by writing the load register
- Automatic loading of the counter after having reached the value '0' in the reload mode
- Loading of the counter by an external load signal (EXT_GATE_TRIG signal)

The counter counts/loads only if the following count/load conditions are met:

- Input CLK_EN=1
- Count/load clock of the TIMER module active.

Exception: Loading of the counter with the value from the load register always takes place when writing the Init_bit in the mode register.

Each TIMER activity which is carried out dependent on the edge of an input signal (INT_GATE_TRIG or EXT_GATE_TRIG, EXT_EV1, 2, INT_EV) becomes effective with the third CLK_TIMT signal at the latest.

The active edges of the input signals at which the edge evaluation is carried out, must have a minimum distance, see chapter 2.3.14.4.5.

Event register

Each TIMER module has:

- Two HW event registers (Ext_Event_1 and Ext_Event_2) in which the current counter values are stored with an edge of the inputs Ext_Ev_1, Ext_Ev_2 (edge programmable), and
- One SW event register (Int_Event) for storing the current counter values with a positive edge of the input INT_EV.

2.3.14.4.4 Overview of the count modes

Gate_ polarity	Ext_ gate_ trig_ enable	Gate_ effect	Clk_ in-put_ select	Function
0	0	0	0	Gate mode: Counter counts/loads with internal clock CLK_TIMT dependent on the pre-selector value as long as INT_GATE_TRIG=1.

Gate_ polarity	Ext_ gate_ trig_ enable	Gate_ effect	Clk_ in-put_ select	Function
0	0	0	1	Count mode: Counter counts/loads with each rising edge of INT_GATE_TRIG, independently of the preselector value.
0	0	1	0	Trigger mode: Counter counts/loads with internal clock dependent on the preselector value and each rising edge of INT_GATE_TRIG triggers the counter (sets the counter to the load/reload value if the current counter value is unequal to the load/reload value or has no effect if the current counter value is equal to the reload value).
0	0	1	1	Toggle mode: Counter changes with each rising edge of INT_GATE_TRIG independently of the preselector value between • loading with load/reload value (if the current counter value is unequal to the load/reload value) and • down-counting (if the current counter value is equal to the load/reload value)
0	1	0	0	Gate mode: Counter counts/loads with internal clock CLK_TIMT dependent on the preselector value as long as EXT_GATE_TRIG=1.
0	1	0	1	Count mode: Counter counts/loads with each rising edge of EXT_GATE_TRIG, independently of the preselector value.
0	1	1	0	Trigger mode: Counter counts/loads with internal clock dependent on the preselector value and each rising edge of EXT_GATE_TRIG triggers the counter (sets the counter to the load/reload value if the current counter value is unequal to the load/reload value or has no effect if the current counter value is equal to the reload value).
0	1	1	1	Toggle mode: Counter changes with each rising edge of EXT_GATE_TRIG independently of the preselector value between • loading with load/reload value (if the current counter value is unequal to the load/reload value) and • down-counting (if the current counter value is equal to the load/reload value)
1	0	0	0	Gate mode:

Copyright © Siemens AG 2023. All rights reserved

Gate_ polarity	Ext_ gate_ trig_ enable	Gate_ effect	Clk_ in-put_ select	Function
				Counter counts/loads with internal clock CLK_TIMT dependent on the pre-selector value as long as INT_GATE_TRIG=0.
1	0	0	1	Count mode: Counter counts/loads with each falling edge of INT_GATE_TRIG, independently of the preselector value.
1	0	1	0	Trigger mode: Counter counts/loads with internal clock dependent on the preselector value and each falling edge of INT_GATE_TRIG triggers the counter (sets the counter to the load/reload value if the current counter value is unequal to the load/reload value or has no effect if the current counter value is equal to the reload value).
1	0	1	1	Toggle mode: Counter changes with each falling edge of INT_GATE_TRIG independently of the preselector value between • loading with load/reload value (if the current counter value is unequal to the load/reload value) and • down-counting (if the current counter value is equal to the load/reload value)
1	1	0	0	Gate mode: Counter counts/loads with internal clock CLK_TIMT dependent on the pre-selector value as long as EXT_GATE_TRIG=0.
1	1	0	1	Count mode: Counter counts/loads with each falling edge of EXT_GATE_TRIG independently of the preselector value.
1	1	1	0	Trigger mode: Counter counts/loads with internal clock dependent on the preselector value and each falling edge of EXT_GATE_TRIG triggers the counter (sets the counter to the load/reload value if the current counter value is unequal to the load/reload value or has no effect if the current counter value is equal to the reload value).
1	1	1	1	Toggle mode: Counter changes with each falling edge of EXT_GATE_TRIG independently of the preselector value between • loading with load/reload value (if the current counter value is unequal to the load/reload value) and • down-counting (if the current counter value is equal to the load/reload value)

Copyright © Siemens AG 2023. All rights reserved

2.3.14.4.5 Timing requirements

The following applies to the TIMER submodule.

The minimum low pulse width and the minimum high pulse width of the signals INT_GATE_TRIG, EXT_GATE_TRIG, EXT_EV1, EXT_EV2, INT_EV must be at least one CLK_TIMT clock.

Furthermore, it is required that the active edges of the input signals at which the edge evaluation is carried out must have a minimum distance according to the following table:

Signal	Minimum distance
INT_GATE_TRIG, EXT_GATE_TRIG	At least three CLK_TIMT clocks; The following applies additionally: If one of the signals INT_GATE_TRIG or EXT_GATE_TRIG is used as a gate or trigger, it has to take at least three CLK_TIMT clocks plus the number of CLK_TIMT clocks set in the preselector in order to become effective.
EXT_EV1, EXT_EV2, INT_EV	At least two CLK_TIMT clocks

2.3.14.4.6 Operating rules

Initialization sequence, modifications of the mode register

Initialization sequence:

1. Stop counter (CLK_EN=0 = setting after reset)
2. Write load register (counter is not loaded if CLK_EN=0)
3. Write mode register with Init_bit = ’1’:
   counter is loaded with load register
4. Start counter (CLK_EN=1).

The initialization sequence has to be carried out with each modification of the mode register in order to prevent any side effects when changing the operating mode.

Cascading of TIMER modules

The TIMER modules can be cascaded with each other, provided that the timer output of the lower-value counter(s) is fed back to the TIMER_TOP module via the EXTERNAL_INPUTS signals and is selected as GATE_TRIG signal at the high-order timer).

Proceed as follows:

1. Program the multiplexer in such a way that each high-order counter has the timer output of the low-order counter as GATE_TRIG signal.
2. Define the interrupt evaluation in such a way that the interrupt of the high-order counter is evaluated. The interrupt of the low-order counter(s) must not be evaluated.
3. Set the operating modes (mode register, prescaler register) of the cascaded TIMER modules the same way (not absolutely necessary, but reasonable if the cascaded TIMER modules are to count with the same clock).
4. When reading the counter value, make sure that the data is consistent, e.g., write SW event trigger register bits for all cascaded TIMER modules in order to take over the values of these TIMER modules synchronously into the Int_Event registers; then, read the Int_Event registers of these TIMER modules.

Copyright © Siemens AG 2023. All rights reserved

5. The settings of the preselectors for the cascaded TIMER modules have to be adapted to the application (same preselectors are reasonable).

Note

Note on the Toggle mode:

For the first count/load clock after having set the Toggle mode or after reset or after active Init_bit, it cannot be predicted if the count/load clock causes a loading of the counter with the reload value or a down-counting (depends on the level of the count/load clock during the setting or during the active Init_bits).

If this information is important for the software, it has to poll the level and value of the counter during the first count/load clock.

NOTICE

Be careful when storing into event registers!

The following combination of the bits Event2_control, Event1_control is **not permitted**:

Event2_control	Event1_control
01	01
01	11
10	10
10	11
11	01
11	10
11	11

2.3.14.4.7 Connections on ERTEC 200P top-level

Timer gate/trigger/event inputs

The timer module has an own multiplexer for each timer for the selection of the sources for the external timer gate/trigger inputs and the external timer event inputs. The assignment of the sources to the timer multiplexers is stated in the table below.

NOTICE

Cascading of the timers is only possible when the outputs of the lower-level timers are connected to the EXT_GATE_TRIG_MUX of the respective higher-level timer(s)!

NOTICE

It is expected that all timer inputs/outputs are synchronous with the timer clock.

Asynchronous inputs (e.g., of ASIC pins) have to be synchronized accordingly. The synchronization of the ASIC pins is done on the core level via two flip-flops each with 125 MHz.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Table 32: Assignment of the timer gate/trigger/event inputs to the timer multiplexers

Timer	Ext_Gate_Trig_MUX	Event_1_MUX	Event_2_MUX
Timer 0-5	0: TIM_TRIG0 (GPIO6/26)	0: TIM_TRIG0 (GPIO6/26)	0: TIM_TRIG0 (GPIO6/26)
	1: TIM_TRIG1 (GPIO7/27)	1: TIM_TRIG1 (GPIO7/27)	1: TIM_TRIG1 (GPIO7/27)
	2: TIM_TRIG2 (GPIO8)	2: TIM_TRIG2 (GPIO8)	2: TIM_TRIG2 (GPIO8)
	3: TIM_TRIG3 (GPIO9)	3: TIM_TRIG3 (GPIO9)	3: TIM_TRIG3 (GPIO9)
	4: TIM_TRIG4 (GPIO10)	4: TIM_TRIG4 (GPIO10)	4: TIM_TRIG4 (GPIO10)
	5: TIM_TRIG5 (GPIO11)	5: TIM_TRIG5 (GPIO11)	5: TIM_TRIG5 (GPIO11)
	6: TIM_OUT0 (Timer 0)	6: TIM_OUT0 (Timer 0)	6: TIM_OUT0 (Timer 0)
	7: TIM_OUT2 (Timer 2)	7: TIM_OUT2 (Timer 2)	7: TIM_OUT2 (Timer 2)
	8: TIM_OUT4 (Timer 4)	8: TIM_OUT4 (Timer 4)	8: TIM_OUT4 (Timer 4)
	9: CLK_OUT (clock divider output timer top)	9: CLK_OUT (clock divider output timer top)	9: CLK_OUT (clock divider output timer top)
	10: '0'	10: '0'	10: '0'
	11: '0'	11: '0'	11: '0'
	12: '0'	12: '0'	12: '0'
	13: '0'	13: '0'	13: '0'
	14: '0'	14: '0'	14: '0'
	15: '0'	15: '0'	15: '0'

With the external inputs TIM_TRIG0-5 it is possible to control the timers from external HW. With the internal inputs TIM_OUT0, TIM_OUT2, TIM_OUT4, it is possible to link timers to each other. The output of the clock divider CLK_OUT in the timer top is connected with Bit 9 of the input multiplexer.

All six timer outputs (TIM_OUT0-5) are connected to ASIC outputs via GPIOs and the Interrupt Controller (IRQ21-26) (Kap. 2.3.5.9).

The minimum distance of all signals (Bit 0 – Bit 15) must be at least 3 x 125 MHz clocks. That means e.g., clock divider value > 3 for CLK_OUT.

Note

It is advisable to share the timers between the application and the PN stack. Timer 0-2 should be used for the PN stack and Timer 3-5 for the application.

2.3.14.4.8 Address mapping

Start-Address	End-Address	Modul/Memory-Name
0h	D7h	TIMER

Module	Register/Memory	Read	Write	Access	Address
/TIMER				32-Bit	

	/TIM_0_MODE_REG	(r)	w	32-Bit	0h
	/TIM_0_PRESCALER_REG	r	w	32-Bit	4h
	/TIM_0_LOAD_REG	r	w	32-Bit	8h
	/TIM_0_COUNT_REG	r		32-Bit	Ch
	/TIM_0_INT_EV_REG	rh	w	32-Bit	10h
	/TIM_0_EXT_EV_1_REG	rh	w	32-Bit	14h
	/TIM_0_EXT_EV_2_REG	rh	w	32-Bit	18h
	/TIM_1_MODE_REG	(r)	w	32-Bit	20h
	/TIM_1_PRESCALER_REG	r	w	32-Bit	24h
	/TIM_1_LOAD_REG	r	w	32-Bit	28h
	/TIM_1_COUNT_REG	r		32-Bit	2Ch
	/TIM_1_INT_EV_REG	rh	w	32-Bit	30h
	/TIM_1_EXT_EV_1_REG	rh	w	32-Bit	34h
	/TIM_1_EXT_EV_2_REG	rh	w	32-Bit	38h
	/TIM_2_MODE_REG	(r)	w	32-Bit	40h
	/TIM_2_PRESCALER_REG	r	w	32-Bit	44h
	/TIM_2_LOAD_REG	r	w	32-Bit	48h
	/TIM_2_COUNT_REG	r		32-Bit	4Ch
	/TIM_2_INT_EV_REG	rh	w	32-Bit	50h
	/TIM_2_EXT_EV_1_REG	rh	w	32-Bit	54h
	/TIM_2_EXT_EV_2_REG	rh	w	32-Bit	58h
	/TIM_3_MODE_REG	(r)	w	32-Bit	60h
	/TIM_3_PRESCALER_REG	r	w	32-Bit	64h
	/TIM_3_LOAD_REG	r	w	32-Bit	68h
	/TIM_3_COUNT_REG	r		32-Bit	6Ch
	/TIM_3_INT_EV_REG	rh	w	32-Bit	70h
	/TIM_3_EXT_EV_1_REG	rh	w	32-Bit	74h
	/TIM_3_EXT_EV_2_REG	rh	w	32-Bit	78h
	/TIM_4_MODE_REG	(r)	w	32-Bit	80h
	/TIM_4_PRESCALER_REG	r	w	32-Bit	84h
	/TIM_4_LOAD_REG	r	w	32-Bit	88h
	/TIM_4_COUNT_REG	r		32-Bit	8Ch
	/TIM_4_INT_EV_REG	rh	w	32-Bit	90h
	/TIM_4_EXT_EV_1_REG	rh	w	32-Bit	94h
	/TIM_4_EXT_EV_2_REG	rh	w	32-Bit	98h
	/TIM_5_MODE_REG	(r)	w	32-Bit	A0h
	/TIM_5_PRESCALER_REG	r	w	32-Bit	A4h
	/TIM_5_LOAD_REG	r	w	32-Bit	A8h
	/TIM_5_COUNT_REG	r		32-Bit	ACh
	/TIM_5_INT_EV_REG	rh	w	32-Bit	B0h
	/TIM_5_EXT_EV_1_REG	rh	w	32-Bit	B4h
	/TIM_5_EXT_EV_2_REG	rh	w	32-Bit	B8h
	/GATE_TRIG_CONTROL_REG	r	w	32-Bit	C0h
	/CLOCK_DIVIDER_REG	r	w	32-Bit	C4h
	/EXT_GATE_TRIG_MUX_REG	r	w	32-Bit	C8h
	/EXT_EV_1_MUX_REG	r	w	32-Bit	CCh
	/EXT_EV_2_MUX_REG	r	w	32-Bit	D0h
	/SW_EVENT_TRIGGER_REG		w	32-Bit	D4h

Copyright © Siemens AG 2023. All rights reserved

2.3.14.4.9 Register description

Module: TIMER

Register:	TIM_0_MODE_REG				Address:	0h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	w
Description:	Timer Mode Register for Timer 0						
Bit	Identifier	Reset	Attr.		Function / Description		
0	INIT_BIT	0h		w	Initialization (0=INIT_BIT not active, 1=INIT_BIT active)		
3dt1	FW_BITS	0h	r	w	not used		
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0=CLK_TIMT, 1=ext. Gate-/Trigger-signal, see also bit GATE_EFFECT)		
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0=Reload Mode is active, 1=Single Mode is active		
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when Writimg Load-Register 0=Reload Mode is active, 1=Single Mode is active		
7	EXT_GATE_TRIG_ENABLE	0h	r	w	Selection of Gate-/Trigger signal 0=INT_GATE_TRIG, 1=EXT_GATE_TRIG is Gate-/Trigger signal		
8	GATE_POLARITY	0h	r	w	polarity of ext. Gate/Trigger signal 0=high active/rising edge, 1=low active/falling edge		
9	GATE_EFFECT	0h	r	w	Effect of ext. Gate/Trigger signal 0=Gate-/Count-Mode for CLK_INPUT_SELECT=0/1,1=Trigger-/Toggle-Mode for CLK_INPUT_SELECT=0/1		
10	TIMER_OUT_POLARITY	0h	r	w	polarity of TIM_OUT 0=high active, 1=low active		
12dt11	EVENT1_CONTROL	0h	r	w	Effect of EXT_EV1 on External Event register		
13	EVENT1_INVERSION	0h	r	w	Inversion of Event1 0=not inverted, 1=inverted		
15dt14	EVENT2_CONTROL	0h	r	w	Effect of EXT_EV2 on External Event register		
16	EVENT2_INVERSION	0h	r	w	Inversion of Event2 0=not inverted, 1=inverted		
31dt17	reserved	0000h					

Register:	TIM_0_PRESCALER_REG				Address:	4h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Timer PRESCALER Register for Timer 0						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE	0000h	r	w	Prescaler Value		
31dt8	reserved	000000h					

Register:	TIM_0_LOAD_REG	Address:	8h

Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Timer LOAD/RELOAD Register for Timer 0					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	LOAD_RELOAD_VALUE	0000h	r	w	Load/Reload value	

Register:	TIM_0_COUNT_REG			Address:	Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	Timer COUNT Register for Timer 0					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	COUNTER_VALUE	0000h	r		Counter Value	

Register:	TIM_0_INT_EV_REG			Address:	10h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Timer INT_EVENT Register for Timer 0					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	INT_EVENT_VALUE	0000h	rh	w	Internal Event Value	

Register:	TIM_0_EXT_EV_1_REG			Address:	14h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Timer EXT_EVENT_2 Register for Timer 0					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	EXT_EVENT_1_VALUE	0000h	rh	w	External Event 1 Value	

Register:	TIM_0_EXT_EV_2_REG			Address:	18h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Timer EXT_EVENT_1 Register for Timer 0					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	EXT_EVENT_2_VALUE	0000h	rh	w	External Event 2 Value	

Register:	TIM_1_MODE_REG			Address:	20h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	w
Description:	Timer Mode Register for Timer 1					
Bit	Identifier	Reset	Attr.		Function / Description	
0	INIT_BIT	0h		w	Initialization (0=INIT_BIT not active, 1=INIT_BIT active)	
3dt1	FW_BITS	0h	r	w	not used	
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0=CLK_TIMT, 1=ext. Gate-/Trigger-signal, see also bit GATE_EFFECT)	
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0=Reload Mode is active, 1=Single Mode is active	
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when Writimg Load-Register	

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

					0=Reload Mode is active, 1=Single Mode is active
7	EXT_GATE_TRIG_ENABLE	0h	r	w	Selection of Gate-/Trigger signal 0=INT_GATE_TRIG, 1=EXT_GATE_TRIG is Gate-/Trigger signal
8	GATE_POLARITY	0h	r	w	polarity of ext. Gate/Trigger signal 0=high active/rising edge, 1=low active/falling edge
9	GATE_EFFECT	0h	r	w	Effect of ext. Gate/Trigger signal 0=Gate-/Count-Mode for CLK_INPUT_SELECT=0/1,1=Trigger-/Toggle-Mode for CLK_INPUT_SELECT=0/1
10	TIMER_OUT_POLARITY	0h	r	w	polarity of TIM_OUT 0=high active, 1=low active
12dt11	EVENT1_CONTROL	0h	r	w	Effect of EXT_EV1 on External Event register
13	EVENT1_INVERSION	0h	r	w	Inversion of Event1 0=not inverted, 1=inverted
15dt14	EVENT2_CONTROL	0h	r	w	Effect of EXT_EV2 on External Event register
16	EVENT2_INVERSION	0h	r	w	Inversion of Event2 0=not inverted, 1=inverted
31dt17	reserved	0000h			

Register:	TIM_1_PRESCALER_REG				Address:	24h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Timer PRESCALER Register for Timer 1						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE	0000h	r	w	Prescaler Value		
31dt8	reserved	000000h					

Register:	TIM_1_LOAD_REG				Address:	28h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Timer LOAD/RELOAD Register for Timer 1						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	LOAD_RELOAD_VALUE	0000h	r	w	Load/Reload value		

Register:	TIM_1_COUNT_REG				Address:	2Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	r	
Description:	Timer COUNT Register for Timer 1						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	COUNTER_VALUE	0000h	r		Counter Value		

Register:	TIM_1_INT_EV_REG				Address:	30h	
Bits:	31dt0	Reset value:		0h	Attributes:	rh	w
Description:	Timer INT_EVENT Register for Timer 1						

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	INT_EVENT_VALUE	0000h	rh	w	Internal Event Value

Register:	TIM_1_EXT_EV_1_REG			Address:	34h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Timer EXT_EVENT_1 Register for Timer 1					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	EXT_EVENT_1_VALUE	0000h	rh	w	External Event 1 Value	

Register:	TIM_1_EXT_EV_2_REG			Address:	38h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Timer EXT_EVENT_2 Register for Timer 1					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	EXT_EVENT_2_VALUE	0000h	rh	w	External Event 2 Value	

Register:	TIM_2_MODE_REG			Address:	40h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	w
Description:	Timer Mode Register for Timer 2					
Bit	Identifier	Reset	Attr.		Function / Description	
0	INIT_BIT	0h		w	Initialization (0=INIT_BIT not active, 1=INIT_BIT active)	
3dt1	FW_BITS	0h	r	w	not used	
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0=CLK_TIMT, 1=ext. Gate-/Trigger-signal, see also bit GATE_EFFECT)	
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0=Reload Mode is active, 1=Single Mode is active	
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when Writimg Load-Register 0=Reload Mode is active, 1=Single Mode is active	
7	EXT_GATE_TRIG_ENABLE	0h	r	w	Selection of Gate-/Trigger signal 0=INT_GATE_TRIG, 1=EXT_GATE_TRIG is Gate-/Trigger signal	
8	GATE_POLARITY	0h	r	w	polarity of ext. Gate/Trigger signal 0=high active/rising edge, 1=low active/falling edge	
9	GATE_EFFECT	0h	r	w	Effect of ext. Gate/Trigger signal 0=Gate-/Count-Mode for CLK_INPUT_SELECT=0/1,1=Trigger-/Toggle-Mode for CLK_INPUT_SELECT=0/1	
10	TIMER_OUT_POLARITY	0h	r	w	polarity of TIM_OUT 0=high active, 1=low active	
12dt11	EVENT1_CONTROL	0h	r	w	Effect of EXT_EV1 on External Event register	
13	EVENT1_INVERSION	0h	r	w	Inversion of Event1 0=not inverted, 1=inverted	

15dt14	EVENT2_CONTROL	0h	r	w	Effect of EXT_EV2 on External Event register
16	EVENT2_INVERSION	0h	r	w	Inversion of Event2 0=not inverted, 1=inverted
31dt17	reserved	0000h			

Register:	TIM_2_PRESCALER_REG				Address:	44h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Timer PRESCALER Register for Timer 2						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE	0000h	r	w	Prescaler Value		
31dt8	reserved	000000h					

Register:	TIM_2_LOAD_REG				Address:	48h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Timer LOAD/RELOAD Register for Timer 2						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	LOAD_RELOAD_VALUE	0000h	r	w	Load/Reload value		

Register:	TIM_2_COUNT_REG				Address:	4Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	r	
Description:	Timer COUNT Register for Timer 2						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	COUNTER_VALUE	0000h	r		Counter Value		

Register:	TIM_2_INT_EV_REG				Address:	50h	
Bits:	31dt0	Reset value:		0h	Attributes:	rh	w
Description:	Timer INT_EVENT Register for Timer 2						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	INT_EVENT_VALUE	0000h	rh	w	Internal Event Value		

Register:	TIM_2_EXT_EV_1_REG				Address:	54h	
Bits:	31dt0	Reset value:		0h	Attributes:	rh	w
Description:	Timer EXT_EVENT_1 Register for Timer 2						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_1_VALUE	0000h	rh	w	External Event 1 Value		

Register:	TIM_2_EXT_EV_2_REG				Address:	58h	
Bits:	31dt0	Reset value:		0h	Attributes:	rh	w
Description:	Timer EXT_EVENT_2 Register for Timer 2						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_2_VALUE	0000h	rh	w	External Event 2 Value		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Register:	TIM_3_MODE_REG				Address:	60h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	w
Description:	Timer Mode Register for Timer 3						
Bit	Identifier	Reset	Attr.		Function / Description		
0	INIT_BIT	0h		w	Initialization (0=INIT_BIT not active, 1=INIT_BIT active)		
3dt1	FW_BITS	0h	r	w	not used		
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0=CLK_TIMT, 1=ext. Gate-/Trigger-signal, see also bit GATE_EFFECT)		
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0=Reload Mode is active, 1=Single Mode is active		
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when Writimg Load-Register 0=Reload Mode is active, 1=Single Mode is active		
7	EXT_GATE_TRIG_ENABLE	0h	r	w	Selection of Gate-/Trigger signal 0=INT_GATE_TRIG, 1=EXT_GATE_TRIG is Gate-/Trigger signal		
8	GATE_POLARITY	0h	r	w	polarity of ext. Gate/Trigger signal 0=high active/rising edge, 1=low active/falling edge		
9	GATE_EFFECT	0h	r	w	Effect of ext. Gate/Trigger signal 0=Gate-/Count-Mode for CLK_INPUT_SELECT=0/1,1=Trigger-/Toggle-Mode for CLK_INPUT_SELECT=0/1		
10	TIMER_OUT_POLARITY	0h	r	w	polarity of TIM_OUT 0=high active, 1=low active		
12dt11	EVENT1_CONTROL	0h	r	w	Effect of EXT_EV1 on External Event register		
13	EVENT1_INVERSION	0h	r	w	Inversion of Event1 0=not inverted, 1=inverted		
15dt14	EVENT2_CONTROL	0h	r	w	Effect of EXT_EV2 on External Event register		
16	EVENT2_INVERSION	0h	r	w	Inversion of Event2 0=not inverted, 1=inverted		
31dt17	reserved	0000h					

Register:	TIM_3_PRESCALER_REG				Address:	64h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Timer PRESCALER Register for Timer 3						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE	xh	r	w	Prescaler Value		
31dt8	reserved	000000h					

Register:	TIM_3_LOAD_REG				Address:	68h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Timer LOAD/RELOAD Register for Timer 3						

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	LOAD_RELOAD_VALUE	0000h	r	w	Load/Reload value

Register:	TIM_3_COUNT_REG			Address:	6Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	Timer COUNT Register for Timer 3					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	COUNTER_VALUE	0000h	r		Counter Value	

Register:	TIM_3_INT_EV_REG			Address:	70h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Timer INT_EVENT Register for Timer 3					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	INT_EVENT_VALUE	0000h	rh	w	Internal Event Value	

Register:	TIM_3_EXT_EV_1_REG			Address:	74h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Timer EXT_EVENT_1 Register for Timer 3					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	EXT_EVENT_1_VALUE	0000h	rh	w		

Register:	TIM_3_EXT_EV_2_REG			Address:	78h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Timer EXT_EVENT_2 Register for Timer 3					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	EXT_EVENT_2_VALUE	0000h	rh	w		

Register:	TIM_4_MODE_REG			Address:	80h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	w
Description:	Timer Mode Register for Timer 4					
Bit	Identifier	Reset	Attr.		Function / Description	
0	INIT_BIT	0h		w	Initialization (0=INIT_BIT not active, 1=INIT_BIT active)	
3dt1	FW_BITS	0h	r	w	not used	
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0=CLK_TIMT, 1=ext. Gate-/Trigger-signal, see also bit GATE_EFFECT)	
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0=Reload Mode is active, 1=Single Mode is active	
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when Writimg Load-Register 0=Reload Mode is active, 1=Single Mode is active	
7	EXT_GATE_TRIG_ENABLE	0h	r	w	Selection of Gate-/Trigger signal	

					0=INT_GATE_TRIG, 1=EXT_GATE_TRIG is Gate-/Trigger signal
8	GATE_POLARITY	0h	r	w	polarity of ext. Gate/Trigger signal 0=high active/rising edge, 1=low active/falling edge
9	GATE_EFFECT	0h	r	w	Effect of ext. Gate/Trigger signal 0=Gate-/Count-Mode for CLK_INPUT_SELECT=0/1,1=Trigger-/Toggle-Mode for CLK_INPUT_SELECT=0/1
10	TIMER_OUT_POLARITY	0h	r	w	polarity of TIM_OUT 0=high active, 1=low active
12dt11	EVENT1_CONTROL	0h	r	w	Effect of EXT_EV1 on External Event register
13	EVENT1_INVERSION	0h	r	w	Inversion of Event1 0=not inverted, 1=inverted
15dt14	EVENT2_CONTROL	0h	r	w	Effect of EXT_EV2 on External Event register
16	EVENT2_INVERSION	0h	r	w	Inversion of Event2 0=not inverted, 1=inverted
31dt17	reserved	0000h			

Register:	TIM_4_PRESCALER_REG				Address:	84h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Timer PRESCALER Register for Timer 4						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE	0000h	r	w	Prescaler Value		
31dt8	reserved	000000h					

Register:	TIM_4_LOAD_REG				Address:	88h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Timer LOAD/RELOAD Register for Timer 4						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	LOAD_RELOAD_VALUE	0000h	r	w	Load/Reload value		

Register:	TIM_4_COUNT_REG				Address:	8Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	r	
Description:	Timer COUNT Register for Timer 4						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	COUNTER_VALUE	0000h	r		Counter Value		

Register:	TIM_4_INT_EV_REG				Address:	90h	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	w
Description:	Timer INT_EVENT Register for Timer 4						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	INT_EVENT_VALUE	0000h	rh	w	Internal Event Value		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Register:	TIM_4_EXT_EV_1_REG				Address:	94h	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	w
Description:	Timer EXT_EVENT_1 Register for Timer 4						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_1_VALUE	0000h	rh	w	External Event 1 Value		

Register:	TIM_4_EXT_EV_2_REG				Address:	98h	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	w
Description:	Timer EXT_EVENT_2 Register for Timer 4						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_2_VALUE	0000h	rh	w	External Event 2 Value		

Register:	TIM_5_MODE_REG				Address:	A0h	
Bits:	31dt0	Reset value:	0h		Attributes:	(r)	w
Description:	Timer Mode Register for Timer 5						
Bit	Identifier	Reset	Attr.		Function / Description		
0	INIT_BIT	0h		w	Initialization (0=INIT_BIT not active, 1=INIT_BIT active)		
3dt1	FW_BITS	0h	r	w	not used		
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0=CLK_TIMT, 1=ext. Gate-/Trigger-signal, see also bit GATE_EFFECT)		
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0=Reload Mode is active, 1=Single Mode is active		
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when Writimg Load-Register 0=Reload Mode is active, 1=Single Mode is active		
7	EXT_GATE_TRIG_ENABLE	0h	r	w	Selection of Gate-/Trigger signal 0=INT_GATE_TRIG, 1=EXT_GATE_TRIG is Gate-/Trigger signal		
8	GATE_POLARITY	0h	r	w	polarity of ext. Gate/Trigger signal 0=high active/rising edge, 1=low active/falling edge		
9	GATE_EFFECT	0h	r	w	Effect of ext. Gate/Trigger signal 0=Gate-/Count-Mode for CLK_INPUT_SELECT=0/1,1=Trigger-/Toggle-Mode for CLK_INPUT_SELECT=0/1		
10	TIMER_OUT_POLARITY	0h	r	w	polarity of TIM_OUT 0=high active, 1=low active		
12dt11	EVENT1_CONTROL	0h	r	w	Effect of EXT_EV1 on External Event register		
13	EVENT1_INVERSION	0h	r	w	Inversion of Event1 0=not inverted, 1=inverted		
15dt14	EVENT2_CONTROL	0h	r	w	Effect of EXT_EV2 on External Event register		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

16	EVENT2_INVERSION	0h	r	w	Inversion of Event2 0=not inverted, 1=inverted
31dt17	reserved	0000h			

Register:	TIM_5_PRESCALER_REG			Address:	A4h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Timer PRESCALER Register for Timer 5					
Bit	Identifier	Reset	Attr.		Function / Description	
7dt0	PRESCALER_VALUE	xh	r	w	Prescaler Value	
31dt8	reserved	000000h				

Register:	TIM_5_LOAD_REG			Address:	A8h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Timer LOAD/RELOAD Register for Timer 5					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	LOAD_RELOAD_VALUE	0000h	r	w	Load/Reload value	

Register:	TIM_5_COUNT_REG			Address:	ACh	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	Timer COUNT Register for Timer 5					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	COUNTER_VALUE	0000h	r		Counter Value	

Register:	TIM_5_INT_EV_REG			Address:	B0h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Timer INT_EVENT Register for Timer 5					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	INT_EVENT_VALUE	0000h	rh	w	Internal Event Value	

Register:	TIM_5_EXT_EV_1_REG			Address:	B4h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Timer EXT_EVENT_1 Register for Timer 5					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	EXT_EVENT_1_VALUE	0000h	rh	w	External Event 1 Value	

Register:	TIM_5_EXT_EV_2_REG			Address:	B8h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Timer EXT_EVENT_2 Register for Timer 5					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	EXT_EVENT_2_VALUE	0000h	rh	w	External Event 2 Value	

Register:	GATE_TRIG_CONTROL_REG	Address:	C0h

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Timer Gate Control Register					

Bit	Identifier	Reset	Attr.		Function / Description
0	TIM_0_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 0
1	TIM_1_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 1
2	TIM_2_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 2
3	TIM_3_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 3
4	TIM_4_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 4
5	TIM_5_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 5
6	TIM_0_CLK_EN	0h	r	w	Clock Enable Timer 0
7	TIM_1_CLK_EN	0h	r	w	Clock Enable Timer 1
8	TIM_2_CLK_EN	0h	r	w	Clock Enable Timer 2
9	TIM_3_CLK_EN	0h	r	w	Clock Enable Timer 3
10	TIM_4_CLK_EN	0h	r	w	Clock Enable Timer 4
11	TIM_5_CLK_EN	0h	r	w	Clock Enable Timer 5
31dt12	reserved	00000h			

Register:	CLOCK_DIVIDER_REG			Address:	C4h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Timer Clock Devider Register					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	CLOCK_DIVIDER_VALUE	0h	r	w	Clock Divider Value
8	CLK_DIV_EN	0h	r	w	Clock Divider Enable (0=disabled)
31dt9	reserved	000000h			

Register:	EXT_GATE_TRIG_MUX_REG			Address:	C8h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Timer MUX Register External Gate					

Bit	Identifier	Reset	Attr.		Function / Description
3dt0	TIM0_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_GATE_TRIG of Timer 0
7dt4	TIM1_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_GATE_TRIG of Timer 1
11dt8	TIM2_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_GATE_TRIG of Timer 2
15dt12	TIM3_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_GATE_TRIG of Timer 3

Copyright © Siemens AG 2023. All rights reserved

19dt16	TIM4_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_GATE_TRIG of Timer 4
23dt20	TIM5_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_GATE_TRIG of Timer 5
31dt24	reserved	00h			

Register:	EXT_EV_1_MUX_REG			Address:	CCh	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Timer MUX Register Event1 Select					
Bit	Identifier	Reset	Attr.		Function / Description	
3dt0	TIM0_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_Event 1 of Timer 0	
7dt4	TIM1_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_Event 1 of Timer 1	
11dt8	TIM2_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_Event 1 of Timer 2	
15dt12	TIM3_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_Event 1 of Timer 3	
19dt16	TIM4_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_Event 1 of Timer 4	
23dt20	TIM5_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_Event 1 of Timer 5	
31dt24	reserved	00h				

Register:	EXT_EV_2_MUX_REG			Address:	D0h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Timer MUX Register Event2 Select					
Bit	Identifier	Reset	Attr.		Function / Description	
3dt0	TIM0_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_Event 2 of Timer 0	
7dt4	TIM1_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_Event 2 of Timer 1	
11dt8	TIM2_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_Event 2 of Timer 2	
15dt12	TIM3_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_Event 2 of Timer 3	
19dt16	TIM4_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_Event 2 of Timer 4	

Copyright © Siemens AG 2023. All rights reserved

23dt20	TIM5_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_IN-PUTS(15:0) for EXT_Event 2 of Timer 5
31dt24	reserved	00h			

Register:	SW_EVENT_TRIGGER_REG				Address:	D4h
Bits:	31dt0	Reset value:	0h		Attributes:	w
Description:	Timer SW Event Register					
Bit	Identifier	Reset	Attr.		Function / Description	
5dt0	SW_EVENT_TRIG5_0	0h		w	Software Event for Saving Counter value in Int_Event register of Timer 5...0 1: Software event active, read=al-ways 0	
31dt6	reserved	0000000h				

2.3.14.5 F-Counter

For F-applications, another timer which is not dependant on the system clock is required in addition to a system timer. The functionality corresponds to the F-counter function of the IBC blocks.

The F-counter is triggered with a separate BYP_CLK input.

Figure 42: F-timer block diagram

2.3.14.5.1 Function description

The asynchronous input signal F_CLK of the external independent time base is connected to a synchronization stage with the input pin BYP_CLK. The SYNC stage is implemented with 3 FF stages to prevent metastable states arising at the counter input.

In a downstream stage for edge detection, the count pulses (F-Counter-EN) are generated and connected to the 32-bit down counter. All flip-flops in the circuit operate with the APB clock (125 MHz).

The 32-bit counter is reset to 0x0000 0000h with an asynchronous block reset (XRESET, XSRST not shown in Figure 42). The next counter pulse sets the counter to 0xFFFF FFFFh. Each subsequent pulse decrements the counter value. When the counter reaches 0x0000 0000h, the next counter pulse triggers a wrap-around to 0xFFFF FFFFh. Writing a value of 0xXXXX 55AAh (X:= don't care) to the address of the F-counter reset register (word or halfword) sets the 'FCOUNTER_RES' register, which resets the 32-bit counter to 0x0000 0000h. The 'FCOUNTER_RES' register is automatically reset to 0 one clock cycle later.

With read access (word) to the F-Counter, the complete 32-bit counter is switched to the APB bus. Word access is required to ensure that the counter value is read consistently. Halfword and byte access are technically possible but will return an inconsistent counter value.

2.3.14.5.2 Application information

The maximum input frequency permitted for F_CLK is ¼ of the APB clock upon quartz failure with a 50% duty cycle.

In standard operation, the PLL outputs 500 MHz with a quartz supply of 25 MHz.

A minimum output frequency is established at the PLL following quartz failure at the ERTEC 200P. This free-running PLL frequency is 100 - 300 MHz. The min. APB_CLK frequency following quartz failure is therefore $PLLOUT_{min}/4$ (~ 25 MHz).

For safety reasons, the external independent F-CLK must not exceed ¼ of the minimum possible APB clock (following quartz failure), in order to prevent any errors in edge detection.

The current setting is for an independent F-CLK of 32768 Hz. This ensures a safe difference from the internal system clock.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.5.3 Address mapping

Start-Address	End-Address	Modul/Memory-Name
0h	7h	fcount

Mo-dule	Register/Memory	Read	Write	Access	Address
/fcount				32-Bit	
	/FCOUNTER_VAL	rh		32-Bit	0h
	/FCOUNTER_RES	rh	w	32-Bit	4h

2.3.14.5.4 Register description

Module: fcount

Register:	FCOUNTER_VAL			Address:	0h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:	Timer value of F-Counter					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	FCOUNTER_VAL	00000000h	rh		Timer value of F-Counter	

Register:	FCOUNTER_RES			Address:	4h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Reset register of F-Counter. A reset of the F-Counter in only executed if this register is written with a value of 0xXXXX 55AAh. Both word and halfword accesses are possible.					
Bit	Identifier	Reset	Attr.		Function / Description	
15dt0	FCOUNT_RES_LWORD	0h	rh	w	Lower halfword of F-Counter reset.	
31dt16	FCOUNT_RES_HWORD	0h	rh	w	Lower halfword of F-Counter reset (don’t care).	

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.6 UART1…4 (Universal Asynchronous Receiver/Transmitter)

ERTEC 200P contains 4 UARTs (UART1…4).

UART1…4 is implemented with the ARM soft macro (PL011). This is **similar** to the standard UART 16C550. A detailed description of the registers and individual functions can be found in /15/.

The PL011 soft macro differs from the standard UART 16C550 as follows:

- Receive FIFO trigger level can be set: 1/8, 1/4, 1/2, 3/4 or 7/8
- The deltas from the modem status signals are not available
- Internal register address mapping and the register bit functions differ.

The following 16C550 features are not supported by PL011:

- 1.5 stop bits (only 1 or 2 stop bits are supported)
- Independent 'Receive Clock'

The UARTs can be controlled by the GDMA (see chapter 2.3.7.2.3) and by ARM926.

As the DMA request signals available at UART1…4 are not suitable for GDMA operation, internal status signals are used from the FIFO controller and sent to the GDMA as DMA request signals. The signals for controlling the transmitter and receiver separately over one GDMA channel each (see chapter 2.3.7.2.1.2) are:

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

The following UART interrupts are connected to the ARM interrupt controller (IRQ5-12) (2.3.5.9):

- UART1-4_UARTINTR:
  - Group interrupt (modem status, receive FIFO, transmit FIFO, receive timeout, error);
  - Individual interrupts can be masked
- UART1-4_UARTEINTR:
  - Error interrupt

UART supports the following external signals:

- Transmit Data (TXD, Output)
- Receive Data (RXD, Input)
- Clear To Send (CTS, Input)
- Data Carrier Detect (DCD, Input)
- Data Set Ready (DSR, Input)
- Ring Indicator (RI, Input)
- Request to Send (RTS, Output)
- Data Terminal Ready (DTR, Output)

Copyright © Siemens AG 2023. All rights reserved

UART1-4 interfaces are available at the GPIO as alternate functions. The alternate function can be set with the SW in the GPIO registers (see chapter 2.3.14.10.5.2).

UART1 incl. modem signals is fully available on GPIO95-32. UART2 can only enable transmit, receive data, CTS and RTS. Only transmit and receive data are available from UART3-4. UART3 is designed for debugging and is located on GPIO31-0. UART2 is also located on GPIO31-0 and UART4 on GPIO95-32.

The IO-Link baud rates are supported; the residual error is < 0.01 %. A total of up to 4 IO-Link interfaces can be used. GPIOs are also required for an IO-Link channel. The function is still to be defined.

2.3.14.6.1 UART baud rates

The baud rates are based on the APB clock (125 MHz). The baud rate is calculated using the following formula (ideal baud rate):

$$\mathbf{BRI = F_{UARTCLK} / (BAUDDIV * 16) \quad or \quad BAUDDIV = (F_{UARTCLK} / (16 * BRI))}$$

where:

BAUDDIV = (BRDI),(BRDF) (e.g., when BAUDIV = 1085 → BRDI = 1, BRDF = 0.085)

BAUDDIV comprises an integer part (BRD_I) and a fractional part (BRD_F). The value (m) for setting the fractional divider is calculating using the following formula:

$$\mathbf{m = integer ((BRD_F * 64) + 0.5)}$$

This produces the baud rate divisor used in UART:

$$\mathbf{BAUDDIV_{real} = BRD_I + (m / 64)}$$

This is used to calculate the actual baud rate of the UART using the following formula:

$$\mathbf{BR = F_{UARTCLK} / (BAUDDIV_{real} * 16)}$$

The baud rate error is calculated using the following formula:

$$\mathbf{E_P = ((BR - BRI)/BRI) * 100\ [\%]}$$

where:

$F_{UARTCLK}$	= UART basic frequency = APB clock frequency (125 MHz)
BR	= baud rate
BRI	= ideal baud rate
BRD_I	= integer part of the baud rate divisor to be programmed
BRD_F	= fractional part of the baud rate divisor to be programmed
m	= fractional value to be set for the fractional divider (1/64)
E_P	= percentage baud rate deviation from the ideal baud rate

Table 33 shows the divisor values to be set for the baud rates and deviations from the standard baud rates; UART (incl. APB interface) with a clock of 125 MHz.

Copyright © Siemens AG 2023. All rights reserved

Table 33: Baud rates UART when $F_{UARTCLK}$ = 125 MHz

BRI	BRD_I (Integer)	m (Fractional)	BR	E_P %
460800	16	61	460829.49	+ 0.01
230400	33	58	230414.75	+ 0.01
187500	41	43	187476.57	- 0.01
115200	67	52	115207.37	+ 0.01
76800	101	46	76804.92	+ 0.01
57600	135	41	57597.05	- 0.01
38400	203	29	38399.51	~ 0
19200	406	58	19199.75	~ 0
14400	542	34	14400.09	~ 0
9600	813	51	9600.06	~ 0
4800	1627	39	4799.98	~ 0
2400	3255	13	2400.00	~ 0
1200	6510	27	1199.99	~ 0
110	Not supported	Not supported	Not supported	-

Copyright © Siemens AG 2023. All rights reserved

2.3.14.6.2 Address mapping

Start-Address	End-Address	Modul/Memory-Name
0h	FFFh	UART_PL011

Module	Register/Memory	Read	Write	Access	Address
/UART_PL011				16-Bit	
	/UARTDR	(r)(h)	(w)	16-Bit	0h
	/UARTRSR_UARTECR	(r)(h)		16-Bit	4h
	/UARTFR	(r)(h)		16-Bit	18h
	/USRTILPR	(r)	(w)	16-Bit	20h
	/UARTIBRD	(r)	(w)	16-Bit	24h
	/UARTFBRD	(r)	(w)	16-Bit	28h
	/UARTLCR_H	(r)	(w)	16-Bit	2Ch
	/UARTCR	(r)	(w)	16-Bit	30h
	/UARTIFLS	(r)	(w)	16-Bit	34h
	/UARTIMSC	(r)	(w)	16-Bit	38h
	/UARTRIS	(r)(h)		16-Bit	3Ch
	/UARTMIS	(r)(h)		16-Bit	40h
	/UARTICR		(w)	16-Bit	44h
	/UARTDMACR	(r)	(w)	16-Bit	48h
	/UARTTCR	(r)	(w)	16-Bit	80h
	/UARTITIP	(r)(h)	(w)	16-Bit	84h
	/UARTITOP	(r)(h)	(w)	16-Bit	88h
	/UARTTDR	(r)(h)	(w)	16-Bit	8Ch
	/UARTPeriphID0	(r)		16-Bit	FE0h
	/UARTPeriphID1	(r)		16-Bit	FE4h
	/UARTPeriphID2	(r)		16-Bit	FE8h
	/UARTPeriphID3	(r)		16-Bit	FECh
	/UARTPCellID0	(r)		16-Bit	FF0h
	/UARTPCellID1	(r)		16-Bit	FF4h
	/UARTPCellID2	(r)		16-Bit	FF8h
	/UARTPCellID3	(r)		16-Bit	FFCh

Copyright © Siemens AG 2023. All rights reserved

2.3.14.6.3 Register description

In the description of the UART IP, the read value of the unassigned register bits is in some cases specified as "unpredictable". A read value of '0' is, however, implemented.

The register assignment can be found in /15/.

Module: UART_PL011

Register:	UARTDR			Address:	0h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)(h)	(w)
Description:	UARTDR is the data register Receive (read) data character Transmit (write) data character					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	Data	0h	rh	w	Receive (read) data character. Transmit (write) data character.
8	Framing_Error	0h	rh		Framing error. When this bit is set to 1, it indicates that the received character did not have a valid stop bit (a valid stop bit is 1). In FIFO mode, this error is associated with the character at the top of the FIFO.
9	Parity_Error	0h	rh		Parity error. When this bit is set to 1, it indicates that the parity of the received data character does not match the parity selected as defined by bits 2 and 7 of the UARTLCR_H register. In FIFO mode, this error is associated with the character at the top of the FIFO.
10	Break_Error	0h	rh		Break error. This bit is set to 1 if a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time (defined as start, data, parity and stop bits). In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state), and the next valid start bit is received.
11	Overrun_Error	0h	rh		Overrun error. This bit is set to 1 if data is received and the receive FIFO is already full. This is cleared to 0 once there is an empty space in the FIFO and a new character can be written to it.

Copyright © Siemens AG 2023. All rights reserved

31dt12	reserved	00000h			

Register:	UARTRSR_UARTECR			Address:	4h
Bits:	31dt0	Reset value:	0h	Attributes:	(r)(h)
Description:	receive status register/ error clear register A write to this Register clears the framming, parity, break and overrun error. The data value is not important.				

Bit	Identifier	Reset	Attr.		Function / Description
0	Framing_Error	0h	rh		When this bit is set to 1, it indicates that the received character did not have a valid stop bit (a valid stop bit is 1).
1	Parity_Error	0h	rh		When this bit is set to 1, it indicates that the parity of the received data character does not match the parity selected in UARTLCR:H (bit 2).
2	Break_Error	0h	rh		This bit is set to 1 if a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time (defined as start, data, parity and stop bits).
3	Overrun_Error	0h	rh		This bit is set to 1 if data is received and the FIFO is already full. This bit is cleared to 0 by a write to UARTECR
31dt4	reserved	0000000h			

Register:	UARTFR			Address:	18h
Bits:	31dt0	Reset value:	90h	Attributes:	(r)(h)
Description:	flag register- after reset TXFF, RXFF and BUSY are 0, TXFE and RXFE are 1.				

Bit	Identifier	Reset	Attr.		Function / Description
0	Clear_To_Send	0h	rh		(CTS) This bit is the complement of the PrimeCell UART clear to send (nUARTCTS) modem status input. That is, the bit is 1 when the modem status input is 0.
1	Data_Set_Ready	0h	rh		(DSR) This bit is the complement of the PrimeCell UART data set ready (nUARTDSR) modem status input. That is, the bit is 1 when the modem status input is 0.
2	Data_Carrier_Detect	0h	rh		(DCD) This bit is the complement of the PrimeCell UART data carrier detect (nUARTTDCD) modem status

					input. That is, the bit is 1 when the modem status input is 0.
3	UART_Busy	0h	rh		(BUSY) If this bit is set to 1, the PrimeCell UART is busy transmitting data. This bit remain set until the complete byte, including all the stop bits, has been sent from the shift register. This bit is set as soon as the transmit FIFO becomes non-empty.
4	Receive_FIFO_Empty	1h	rh		(RXFE) The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the receive holding register is empty. If the FIFO is enabled, the RXFE bit is set when the receive FIFO is empty.
5	Transmit_FIFO_Full	0h	rh		(TXFF) The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the transmit holding register is full. If the FIFO is enabled, the TXFF bit is set when the transmit FIFO is full.
6	Receive_FIFO_Full	0h	rh		(RXFF) The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the receive holding register is full. If the FIFO is enabled, the RXFF bit is set when the receive FIFO is full.
7	Transmit_FIFO_Empty	1h	rh		(TXFE) The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the transmit holding register is empty. If the FIFO is enabled, the TXFE bit is set when the transmit FIFO is empty.
8	Ring_Indicator	0h	rh		Ring indicator. This bit is the complement of the UART ring indicator (nUARTRI) modem status input. That is, the bit is 1 when the modem status input is 0.
31dt9	reserved	000000h			

Register:	USRTILPR				Address:	20h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)
Description:	8-bit-low-power divisor value. These bits are cleared to 0 at reset.						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	USRTILPR	0h	r	w	8-bit low-power divisor value.		
31dt8	reserved	000000h					

Register:	UARTIBRD				Address:	24h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)
Description:	Integer Part of the Baud Rate Divisor Value Register						
Bit	Identifier	Reset	Attr.		Function / Description		
15dt0	UARTIBRD	0h	r	w	The integer baud rate divisor.		
31dt16	reserved	0000h					

Register:	UARTFBRD				Address:	28h	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)
Description:	Fractional Part of the Baud Rate Divisor Value Register						
Bit	Identifier	Reset	Attr.		Function / Description		
5dt0	UARTFBRD	0h	r	w	The fractional baud rate divisor.		
31dt6	reserved	0000000h					

Register:	UARTLCR_H				Address:	2Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	(r)	(w)
Description:	line control register, high byte						
Bit	Identifier	Reset	Attr.		Function / Description		
0	Send_Break	0h	r	w	If this bit is set to 1, a low level is continually output on the UARTTXD output, after completing transmission of the current character. (...) For normal use, this bit must be cleared to 0.		
1	Parity_Enable	0h	r	w	If this bit is set to 1, parity checking and generation is enabled, else parity is disabled and no parity bit added to the data frame.		
2	Even_Parity_Select	0h	r	w	If this bit is set to 1, even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits. When cleared to 0 then odd parity is performed which checks for an odd number of 1s.		
3	Two_Stop_Bits_Select	0h	r	w	If this bit is set to 1, two stop bits are transmitted at the end of the frame. The receive logic does not check for two stop bits being received.		
4	Enable_FIFOs	0h	r	w	If this bit is set to 1, transmit and receive FIFO buffers are enabled (FIFO mode). When cleared to 0 the FIFOs are disabled (character mode), that is, the FIFOs become 1-byte-deep holding registers.		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

6dt5	Word_lenght	0h	r	w	Word length. The select bits indicate the number of data bits transmitted or received in a frame as follows: 11 = 8 bits 10 = 7 bits 01 = 6 bits 00 = 5 bits.
7	Stick_Parity_Select	0h	r	w	Stick parity select. When bits 1, 2, and 7 of the UARTLCR_H register are set, the parity bit is transmitted and checked as a 0. When bits 1 and 7 are set, and bit 2 is 0, the parity bit is transmitted and checked as a 1. When this bit is cleared stick parity is disabled.
31dt8	reserved	000000h			

Register:	UARTCR			Address:	30h	
Bits:	31dt0	Reset value:	300h	Attributes:	(r)	(w)
Description:	Control register- all the bits are cleared to 0 on reset.					

Bit	Identifier	Reset	Attr.		Function / Description
0	UART_Enable	0h	r	w	If this bit is set to 1, the PrimeCell UART is enabled.
1	SIR_Enable	0h	r	w	If this bit is set to 1, the IrDA SIR Endec is enabled. This bit has no effect if the UART is not enabled by bit 0 being set to 1.
2	IrDA_SIR_Low_Power_Mode	0h	r	w	This bit selects the IrDA encoding mode. If this bit is cleared to 0, low level bits are transmitted as an active high pulse with a width of 3/16th of the bit period. If this bit is set to 1, low level bits are transmitted with a pulse width which is 3 times the period of the IrLPBaud16 input signal, regardless of the selected bit rate.Setting this bit uses less power, but may reduce transmission distances.
6dt3	reserved	0h			
7	Loop_Back_Enable	0h	r	w	If this bit is set to 1, the SIR Enable is set to 1, and the test register UARTTMR bit 1 (SIRTEST) is set to 1, the nSIROUT path is inverted, and fed through to the SIRIN path. This bit is cleared to 0 on reset, which disables the loopback mode.
8	Transmit_Enable	1h	r	w	If this bit is set to 1, the transmit section is enabled.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

9	Receive_Enable	1h	r	w	If this bit is set to 1, the receive section is enabled.
10	DTR	0h	r	w	If this bit is set to 1, the output DTR in 0.
11	RTS	0h	r	w	If this bit is set to 1, the output RTS in 0.
12	Out1	0h	r	w	If this bit is set to 1, the output OUT1 in 0.
13	Out2	0h	r	w	If this bit is set to 1, the output OUT2 in 0.
14	RTS_Enable	0h	r	w	If 1, RTS hardware flow is enabled.
15	CTS_Enable	0h	r	w	If 1, CTR hardware flow is enabled.
31dt16	reserved	0000h			

Register:	UARTIFLS			Address:	34h	
Bits:	31dt0	Reset value:	12h	Attributes:	(r)	(w)
Description:	Interrupt FIFO Level Select Register					

Bit	Identifier	Reset	Attr.		Function / Description
2dt0	Tx_Int_FIFO_Level	2h	r	w	The trigger points for the transmit interrupt: 000 = RX FIFO becomes -= 1/8 full 001 = RX FIFO becomes -= 1/4 full 010 = RX FIFO becomes -= 1/2 full 011 = RX FIFO becomes -= 3/4 full 100 = RX FIFO becomes -= 7/8 full 101:111 = reserved
5dt3	Rx_Int_FIFO_Level	2h	r	w	The trigger points for the receive interrupt: 000 = RX FIFO becomes -= 1/8 full 001 = RX FIFO becomes -= 1/4 full 010 = RX FIFO becomes -= 1/2 full 011 = RX FIFO becomes -= 3/4 full 100 = RX FIFO becomes -= 7/8 full 101:111 = reserved
31dt6	reserved	0000000h			

Register:	UARTIMSC			Address:	38h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	Interrupt Mask Set/Clear Register On a write of 1, the mask of the interrupt is set. A write of 0 clears the mask.					

Bit	Identifier	Reset	Attr.		Function / Description
0	RI_Modem_Int_Mask	0h	r	w	
1	CTS_Modem_Int_Mask	0h	r	w	
2	DCD_Modem_Int_Mask	0h	r	w	
3	DSR_Modem_Int_Mask	0h	r	w	
4	Rx_Int_Mask	0h	r	w	
5	Tx_Int_Mask	0h	r	w	
6	Receive_Timout_Mask	0h	r	w	
7	Framing_Error_Mask	0h	r	w	
8	Parity_Error_Mask	0h	r	w	
9	Break_Error_Mask	0h	r	w	
10	Overrun_Error_Mask	0h	r	w	
31dt11	reserved	000000h			

Register:	UARTRIS			Address:	3Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)(h)	
Description:	Raw Interrupt Status Register Gives the raw interrupt state (prior to masking) of the interrupt					

Bit	Identifier	Reset	Attr.		Function / Description
0	RI_Modem_Int_Status	0h	rh		
1	CTS_Modem_Int_Status	0h	rh		
2	DCD_Modem_Int_Status	0h	rh		
3	DSR_Modem_Int_Status	0h	rh		
4	Rx_Int_Status	0h	rh		
5	Tx_Int_Status	0h	rh		
6	Rx_Timout_Int_Status	0h	rh		
7	Framing_Error_Int_Status	0h	rh		
8	Parity_Error_Int_Status	0h	rh		
9	Break_Err_Int_Status	0h	rh		
10	Overrun_Err_Int_Status	0h	rh		
31dt11	reserved	000000h			

Register:	UARTMIS			Address:	40h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)(h)	
Description:	Mask Interrupt Status Register Gives the masked interrupt state (after masking) of the interrupt					

Bit	Identifier	Reset	Attr.		Function / Description
0	RIMMIS	0h	rh		...RI Intr.
1	CTSMMIS	0h	rh		...CTS Intr.
2	DCDMMIS	0h	rh		...DCD Intr.
3	DSRMMIS	0h	rh		...DSR Intr.
4	RXMIS	0h	rh		...Receive Interrupt
5	TXMIS	0h	rh		...Transmit Interrupt
6	RTMIS	0h	rh		...Receive Timeout
7	FEMIS	0h	rh		...Framing Error
8	PEMIS	0h	rh		...Parity Error
9	BEMIS	0h	rh		...Break Error
10	OEMIS	0h	rh		...Overrun Error
31dt11	reserved	000000h			

Register:	UARTICR			Address:	44h	
Bits:	31dt0	Reset value:	0h	Attributes:		(w)
Description:	Interrupt Clear Register Clears the Interrupt ...					

Bit	Identifier	Reset	Attr.		Function / Description
0	RIMIC	0h		w	...RI Intr.
1	CTSMIC	0h		w	...CTS Intr.
2	DCDMIC	0h		w	...DCD Intr.
3	DSRMIC	0h		w	...DSR Intr.
4	RXIC	0h		w	...Receive Interrupt
5	TXIC	0h		w	...Transmit Interrupt
6	RTIC	0h		w	...Receive Timeout
7	FEIC	0h		w	...Framing Error
8	PEIC	0h		w	...Parity Error
9	BEIC	0h		w	...Break Error
10	OEIC	0h		w	...Overrun Error
31dt11	reserved	000000h			

Register:	UARTDMACR			Address:	48h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	DMA Control Register					

Bit	Identifier	Reset	Attr.	Function / Description

0	RXDMAE	xh	r	w	If 1, DMA for the receive FIFO is enabled.
1	TXDMAE	xh	r	w	If 1, DMA for the transmit FIFO is enabled.
2	DMAONERR	xh	r	w	If 1, DMA request outputs are disabled when UART error interrupt is asserted.
31dt3	reserved	00000000h			

Register:	UARTTCR			Address:	80h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	Test control register (for integration test)					

Bit	Identifier	Reset	Attr.		Function / Description
0	ITEN	0h	r	w	Integration test enable. When this bit is 1, the UART is placed in integration test mode, otherwise it is in normal mode.
1	TESTFIFO	0h	r	w	Test FIFO enable. When this bit it 1, a write to the UARTTDR writes data into the receive FIFO, and reads from the UARTTDR reads data out of the transmit FIFO. When this bit is 0, data cannot be read directly from the transmit FIFO or written directly to the receive FIFO (normal operation).
2	SIRTEST	0h	r	w	SIR test enable. Setting this bit to 1 enables the receive data path during IrDa transmission (testing requires the SIR to be configured in full-duplex mode). This bit must be set to 1 to enable SIR system loop back testing, when the normal mode control register UARTCR bit 7, Loop Back Enable (LBE) has been set to 1. Clearing this bit to 0 disables the receive logic when the SIR is transmitting (normal operation). This bit defaults to 0 for normal operation (half-duplex peration).
31dt3	reserved	00000000h			

Register:	UARTITIP			Address:	84h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)(h)	(w)
Description:	Integration test input read/set register (for integration test)					

Bit	Identifier	Reset	Attr.		Function / Description
0	UARTRXD	0h	rh	w	value of the UARTRXD primary input.
1	SIRIN	0h	rh	w	value of the SIRIN primary input.
2	nUARTDSR	0h	rh	w	value of the nUARTDSR primary input.
3	nUARTCTS	0h	rh	w	value of the nUARTCTS primary input.
4	nUARTDCD	0h	rh	w	value of the nUARTDCD primary input.
5	nUARTRI	0h	rh	w	value of the nUARTRI primary input.
6	UARTRXDMACLR	0h	rh	w	Writes to this bit specify the value to be driven on the intra-chip input, UARTRXDMACLR, in the integration test mode. Reads return the value of UARTRXDMACLR at the output of the test multiplexor.
7	UARTTXDMACLR	0h	rh	w	Writes to this bit specify the value to be driven on the intra-chip input, UARTTXDMACLR, in the integration test mode. Reads return the value of UARTTXDMACLR at the output of the test multiplexor.
31dt8	reserved	000000h			

Register:	UARTITOP			Address:	88h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)(h)	(w)
Description:	Integration test output read/set register (for integration test)					

Bit	Identifier	Reset	Attr.		Function / Description
0	UARTTXD	0h	rh	w	Writes specify the value to be driven on the UARTTXD line in the integration test mode.
1	nSIROUT	0h	rh	w	Writes specify the value to be driven on the nSIROUT line in the integration test mode.
2	nUARTDTR	0h	rh	w	Writes specify the value to be driven on the nUARTDTR line in the integration test mode.
3	nUARTRTS	0h	rh	w	Writes specify the value to be driven on the nUARTRTS line in the integration test mode.
4	nUARTOut1	0h	rh	w	Writes specify the value to be driven on the nUARTOut1 line in the integration test mode.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

5	nUARTOut2	0h	rh	w	Writes specify the value to be driven on the nUARTOut2 line in the integration test mode.
6	UARTINTR	0h	rh	w	Writes specify the value to be driven on the UARTINTR line in the integration test mode. Reads return the value of UARTINTR at the output of the test multiplexor.
7	UARTEINTR	0h	rh	w	Writes specify the value to be driven on the UARTEINTR line in the integration test mode. Reads return the value of UARTEINTR at the output of the test multiplexor.
8	UARTRTINTR	0h	rh	w	Writes specify the value to be driven on the ARTRTINTR line in the integration test mode. Reads return the value of UARTRTINTR at the output of the test multiplexor.
9	UARTTXINTR	0h	rh	w	Writes specify the value to be driven on the UARTTXINTR line in the integration test mode. Reads return the value of UARTTXINTR at the output of the test multiplexor.
10	UARTRXINTR	0h	rh	w	Writes specify the value to be driven on the UARTRXINTR line in the integration test mode. Reads return the value of UARTRXINTR at the output of the test multiplexor.
11	UARTMSINTR	0h	rh	w	Writes specify the value to be driven on the UARTMSINTR line in the integration test mode. Reads return the value of UARTMSINTR at the output of the test multiplexor.
12	UARTRXDMABREQ	0h	rh	w	Writes specify the value to be driven on the UARTDMABREQ line in the integration test mode. Reads return the value of UARTDMABREQ at the output of the test multiplexor.
13	UARTRXDMASREQ	0h	rh	w	Writes specify the value to be driven on the ARTRXDMASREQ line in the integration test mode. Reads return the value of UARTRXDMASREQ at the output of the test multiplexor.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

14	UARTTXDMABREQ	0h	rh	w	Writes specify the value to be driven on the ARTTXDMABREQ line in the integration test mode. Reads return the value of UARTTXDMABREQ at the output of the test multiplexor.
15	UARTTXDMASREQ	0h	rh	w	Writes specify the value to be driven on the ARTTXDMASREQ line in the integration test mode. Reads return the value of UARTTXDMASREQ at the output of the test multiplexor.
31dt16	reserved	0000h			

Register:	UARTTDR			Address:	8Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)(h)	(w)
Description:	Test data register (for integration test)					
Bit	Identifier	Reset	Attr.		Function / Description	
10dt0	DATA	0h	rh	w	When the TESTFIFO signal is asserted, data is written into the receive FIFO and read out of the transmit FIFO	
31dt11	reserved	000000h				

Register:	UARTPeriphID0			Address:	FE0h	
Bits:	31dt0	Reset value:	11h	Attributes:	(r)	
Description:	Peripheral ID0 Register, hard coded					
Bit	Identifier	Reset	Attr.		Function / Description	
7dt0	PartNumber0	11h	r			
31dt8	reserved	000000h				

Register:	UARTPeriphID1			Address:	FE4h	
Bits:	31dt0	Reset value:	10h	Attributes:	(r)	
Description:	Peripheral ID1 Register, hard coded					
Bit	Identifier	Reset	Attr.		Function / Description	
3dt0	PartNumber1	0h	r			
7dt4	Designer0	1h	r			
31dt8	reserved	000000h				

Register:	UARTPeriphID2	Address:	FE8h

Bits:	31dt0	Reset value:	24h	Attributes:	(r)	
Description:	Peripheral ID2 Register, hard coded					
Bit	Identifier	Reset	Attr.		Function / Description	
3dt0	Designer1	4h	r			
7dt4	Revision	2h	r		r1p4	
31dt8	reserved	000000h				

Register:	UARTPeriphID3			Address:	FECh	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	
Description:	Peripheral ID3 Register, hard coded					
Bit	Identifier	Reset	Attr.		Function / Description	
7dt0	Configuration	0h	r			
31dt8	reserved	000000h				

Register:	UARTPCellID0			Address:	FF0h	
Bits:	31dt0	Reset value:	Dh	Attributes:	(r)	
Description:	PrimeCell ID0 Register, hard coded					
Bit	Identifier	Reset	Attr.		Function / Description	
7dt0	UARTPCellID0	Dh	r			
31dt8	reserved	000000h				

Register:	UARTPCellID1			Address:	FF4h	
Bits:	31dt0	Reset value:	F0h	Attributes:	(r)	
Description:	PrimeCell ID1 Register, hard coded					
Bit	Identifier	Reset	Attr.		Function / Description	
7dt0	UARTPCellID1	F0h	r			
31dt8	reserved	000000h				

Register:	UARTPCellID2			Address:	FF8h	
Bits:	31dt0	Reset value:	5h	Attributes:	(r)	
Description:	PrimeCell ID2 Register, hard coded					
Bit	Identifier	Reset	Attr.		Function / Description	
7dt0	UARTPCellID2	5h	r			
31dt8	reserved	000000h				

Register:	UARTPCellID3			Address:	FFCh	
Bits:	31dt0	Reset value:	B1h	Attributes:	(r)	

Description:	PrimeCell ID3 Register, hard coded				
Bit	Identifier	Reset	Attr.		Function / Description
7dt0	UARTPCellID3	B1h	r		
31dt8	reserved	000000h			

2.3.14.7 I²C (Inter-Integrated Circuit)

The ERTEC 200P contains one I²C module for general purpose applications (I²C_3, master/slave interface). This module is located at the top-level and controlled by the ARM926EJ-S. Another I²C module (master/slave interface) is located in the PN-IP. The two I²C interfaces for the LWL Tranceiver (I²C_1/2) and the general purpose interface (I²C_3) are multiplexed with other peripheral functions on the GPIO interface.

In document /29/ (see chapter 3.2) you will find the I²C Bus Specification.

To use the direct access to the I²C bus, the ARM must access appropriately the I²C interface macro registers. These registers are contained in the I²C interface address area.

The I²C interface contains 2 registers ('SW_I2C_EN' and 'SW_I2C_CTRL') which allow to control the I²C bus signals by software. This software interface is enabled by setting Bit 0 in register 'SW_I2C_EN'. If this software interface is enabled, the rest of the I²C interface cannot be used.

The 8-bit I²C modules are connected to the APB/SC bus (PN-IP). Only word addresses are used for addressing the internal registers. In the case of a write to the module the bit positions 31-8 are ignored and in the case of a read from the module these bit positions are driven with '0'.

2.3.14.7.1 Baudrate generator

In contrast to the MI2C IP specification, the **C**lock **C**ontrol **R**egister (MI2C_CCR) in the I²C interface cannot be written because the register is not implemented. The Clock Control Register for the general purpose I²C interface (I2C_3) is contained in the SCRB ('CCR_I2C'). This allows the SCL clock pulse frequency on the I²C bus to be set (see chapter 2.3.14.11.27.1).

For a baudrate of 100 kBaud on the I²C bus the module requires an internal enable signal of 1 MHz. Therefore, the input clock has to be divided. This will be done by configuring the register 'CCR_I2C' in the SCRB module. The register 'CCR_I2C' is not configured after reset.

Formula for calculating the divider value:

((PCLK MHz) / (CCR_I2C + 1)) = 1 MHz

Example:

IfPCLK = 125 MHz

Then CCR_I2C = 124d = ("01111100b")

Note

Restrictions:

Only the baudrate of 100 kBaud (normal mode) is supported. The baudrate of 400 kBaud (fast mode) cannot be supported because a divider value to generate 4 MHz is not configurable in CCR_I2C by 125 MHz (PCLK).

2.3.14.7.2 Important software rules

1. Ensure for consecutive sequences of start and stop conditions that a non-existent slave address is loaded into the data register before issuing the stop condition. After issuing the stop condition, the contents of the data register are output as slave address, followed by the stop condition. In automatic operation, the contents of the 'ERROR_SLAVE_ADDRESS' register are used as slave address. This routine can be activated using Bit 0 in the 'ERROR_SLAVE_ADDRESS' register.
2. If a deadlock occurs on the I²C bus, write accesses to Bit 7 in the register 'EX_CTRL_1' can be used to influence the SCL clock pin. Setting this bit causes SCL to become “low”. Conversely, resetting the bit switches SCL to “high”.
3. The register 'MI2C_DATA' does not have the same source or the same target when it is addressed with a write/read access. When this register is read, only the data transferred on the

Copyright © Siemens AG 2023. All rights reserved

I^2C bus is accessible. This means that the read value can differ from the written value for a write with a subsequent read.

4. To perform the handshake with the I^2C, it is expected that the ARM926 either polls the IFLG interrupt flag of the CNTR register or activates the interrupt output. Once an interrupt has been recognized, a wait time of at least 2 µs must be observed. Only then the next action on the I^2C bus can be initiated. Otherwise, errors can occur during the next address/data transfer.

5. The I^2C controller has a bug which occurs as follows:

   To initiate actions on I^2C bus, commands are written into the CNTR register of the controller in the following way:

   - Example: Send STOP condition

     a) write: bit STB is set

     b) write: bit IFLG is reset

   Randomly, the STOP condition is not sent.

   The same bug can occur when sending a repeated START condition or a STOP condition followed by a START condition. If the bug occurs the STOP condition is not sent and the following START condition is not sent, too. This malfunction is caused by setting the bit STP in one write and resetting bit IFLG in the next write. The bug can be observed by polling bit IFLG (stays 1 for ever), which should be reset by the write or by polling the Status register (keeps value before the write if bug occurs).

   To avoid this bug, write all commands in one write to the CNTR register.

6. When the I^2C macro is in slave mode software must ensure that all interrupts of the I2C macro are recorded, otherwise data could get lost or the I2C macro may not respond to its slave address.

Copyright © Siemens AG 2023. All rights reserved

2.3.14.7.3 Address mapping

Start-Address	End-Address	Modul/Memory-Name
0h	6Bh	i2c

Module	Register/Memory	Read	Write	Access	Address
/i2c				all	
	/MI2C_ADDR	r	w	all	0h
	/MI2C_DATA	rh	w	all	4h
	/MI2C_CNTR	r(h)	(w)	all	8h
	/MI2C_STAT	rh		all	Ch
	/MI2C_XADDR	r	w	all	10h
	/MI2C_SOFTWARE_RESET		w	all	1Ch
	/EX_CTRL_1	r(h)	(w)	all	20h
	/EX_ADDR_1	r	w	all	24h
	/EX_DATA_OUT_1	r	w	all	28h
	/EX_DATA_IN_1	rh		all	2Ch
	/EX_CTRL_2	r(h)	(w)	all	30h
	/EX_ADDR_2	r	w	all	34h
	/EX_DATA_OUT_2	r	w	all	38h
	/EX_DATA_IN_2	rh		all	3Ch
	/EX_CTRL_3	r(h)	(w)	all	40h
	/EX_ADDR_3	r	w	all	44h
	/EX_DATA_OUT_3	r	w	all	48h
	/EX_DATA_IN_3	rh		all	4Ch
	/EX_CTRL_4	r(h)	(w)	all	50h
	/EX_ADDR_4	r	w	all	54h
	/EX_DATA_OUT_4	r	w	all	58h
	/EX_DATA_IN_4	rh		all	5Ch
	/ERROR_SLAVE_ADDRESS	r	w	all	60h
	/SW_I2C_EN	r	w	all	64h
	/SW_I2C_CTRL	r(h)	(w)	all	68h

2.3.14.7.4 Register description

Module: i2c

Register:	MI2C_ADDR				Address:	0h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Slave Address						
Bit	Identifier	Reset	Attr.		Function / Description		
0	GCE	0h	r	w	General Call address enable		
1	SLA0	0h	r	w	Slave Address (1-0), Extended Address (9-8)		
2	SLA1	0h	r	w	Slave Address (1-0), Extended Address (9-8)		
3	SLA2	0h	r	w	Slave Address (6-2)		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

4	SLA3	0h	r	w	Slave Address (6-2)
5	SLA4	0h	r	w	Slave Address (6-2)
6	SLA5	0h	r	w	Slave Address (6-2)
7	SLA6	0h	r	w	Slave Address (6-2)
31dt8	reserved	000000h			

Register:	MI2C_DATA				Address:	4h	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	w
Description:	Transfer Data Register- Receive-/Transmit-Data						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	MI2C_DATA	0h	rh	w	Transfer Data Register- Receive-/Transmit-Data		
31dt8	reserved	000000h					

Register:	MI2C_CNTR				Address:	8h	
Bits:	31dt0	Reset value:	0h		Attributes:	r(h)	(w)
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	reserved	0h					
2	AAK	0h	r	w	Assert acknowledge		
3	IFLG	0h	rh		Interrupt flag		
4	STP	0h	rh	w	Master mode stop		
5	STA	0h	rh	w	Master mode start		
6	ENAB	0h	r	w	Bus enable		
7	IEN	0h	r	w	Interrupt enable		
31dt8	reserved	000000h					

Register:	MI2C_STAT				Address:	Ch	
Bits:	31dt0	Reset value:	F8h		Attributes:	rh	
Description:	Status register						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	MI2C_STAT	F8h	rh		Status register		
31dt8	reserved	000000h					

Register:	MI2C_XADDR				Address:	10h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Extented Slave Address						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	MI2C_XADDR	0h	r	w	Extented Slave Address		
31dt8	reserved	000000h					

Register:	MI2C_SOFTWARE_RESET	Address:	1Ch

Bits:	31dt0	Reset value:	0h	Attributes:		w
Description:	Software Reset					
Bit	Identifier	Reset	Attr.		Function / Description	
7dt0	MI2C_SOFTWARE_RESET	0h		w	Software Reset	
31dt8	reserved	000000h				

Register:	EX_CTRL_1				Address:	20h	
Bits:	31dt0	Reset value:	0h		Attributes:	r(h)	(w)
Description:	Control register for IO expander 1.						
Bit	Identifier	Reset	Attr.		Function / Description		
0	MODE	0h	r	w	0 = Service on demand, default 1 = Periodic service		
1	OUT	0h	rh	w	0 = Write service off, default 1 = Write service on Send output data as specified in MODE		
2	IN	0h	rh	w	0 = Read service off, default 1 = Read service on Read input data as specified in MODE		
3	BUSY	0h	rh		Read-only. 0 = Currently inactive, default 1= Currently active		
4	ERROR	0h	rh		Read-only. 0 = No Error, default 1= Error occurred		
6dt5	reserved	0h					
7	SCL_Toggle	0h	r	w	SCL toggle bit: '0' keeps SCL high, '1' pulls SCL to low.		
31dt8	reserved	000000h					

Register:	EX_ADDR_1				Address:	24h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Slave address for expander 1.						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	EX_ADDR_1	0h	r	w	Slave address for expander 1.		
31dt8	reserved	000000h					

Register:	EX_DATA_OUT_1				Address:	28h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Send data for expander 1.						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	EX_DATA_OUT_1	0h	r	w	Send data for expander 1.		
31dt8	reserved	000000h					

Register:	EX_DATA_IN_1				Address:	2Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	

Description:	Read data for expander 1.				
Bit	Identifier	Reset	Attr.		Function / Description
7dt0	EX_DATA_IN_1	0h	rh		Read data for expander 1.
31dt8	reserved	000000h			

Register:	EX_CTRL_2				Address:	30h	
Bits:	31dt0	Reset value:		0h	Attributes:	r(h)	(w)
Description:	Control register for IO expander 2.						
Bit	Identifier	Reset	Attr.		Function / Description		
0	MODE	0h	r	w	0 = Service on demand, default 1 = Periodic service		
1	OUT	0h	rh	w	0 = Write service off, default 1 = Write service on Send output data as specified in MODE		
2	IN	0h	rh	w	0 = Read service off, default 1 = Read service on Read input data as specified in MODE		
3	BUSY	0h	rh		Read-only. 0 = Currently inactive, default 1= Currently active		
4	ERROR	0h	rh		Read-only. 0 = No Error, default 1= Error occurred		
31dt5	reserved	0000000h					

Register:	EX_ADDR_2				Address:	34h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Slave address for expander 2.						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	EX_ADDR_2	0h	r	w	Slave address for expander 2.		
31dt8	reserved	000000h					

Register:	EX_DATA_OUT_2				Address:	38h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Send data for expander 2.						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	EX_DATA_OUT_2	0h	r	w	Send data for expander 2.		
31dt8	reserved	000000h					

Register:	EX_DATA_IN_2				Address:	3Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	rh	
Description:	Read data for expander 2.						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	EX_DATA_IN_2	0h	rh		Read data for expander 2.		

31dt8	reserved	000000h			

Register:	EX_CTRL_3				Address:	40h	
Bits:	31dt0	Reset value:		0h	Attributes:	r(h)	(w)
Description:	Control register for IO expander 3.						
Bit	Identifier	Reset	Attr.		Function / Description		
0	MODE	0h	r	w	0 = Service on demand, default 1 = Periodic service		
1	OUT	0h	rh	w	0 = Write service off, default 1 = Write service on Send output data as specified in MODE		
2	IN	0h	rh	w	0 = Read service off, default 1 = Read service on Read input data as specified in MODE		
3	BUSY	0h	rh		Read-only. 0 = Currently inactive, default 1= Currently active		
4	ERROR	0h	rh		Read-only. 0 = No Error, default 1= Error occurred		
31dt5	reserved	0000000h					

Register:	EX_ADDR_3				Address:	44h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Slave address for expander 3.						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	EX_ADDR_3	0h	r	w	Slave address for expander 3.		
31dt8	reserved	000000h					

Register:	EX_DATA_OUT_3				Address:	48h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	Send data for expander 3.						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	EX_DATA_OUT_3	0h	r	w	Send data for expander 3.		
31dt8	reserved	000000h					

Register:	EX_DATA_IN_3				Address:	4Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	rh	
Description:	Read data for expander 3.						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	EX_DATA_IN_3	0h	rh		Read data for expander 3.		
31dt8	reserved	000000h					

Register:	EX_CTRL_4	Address:	50h

Bits:	31dt0	Reset value:	0h	Attributes:	r(h)	(w)
Description:	Control register for IO expander 4.					

Bit	Identifier	Reset	Attr.		Function / Description
0	MODE	0h	r	w	0 = Service on demand, default 1 = Periodic service
1	OUT	0h	rh	w	0 = Write service off, default 1= Write service on Send output data as specified in MODE
2	IN	0h	rh	w	0 = Read service off, default 1 = Read service on Read input data as specified in MODE
3	BUSY	0h	rh		Read-only. 0 = Currently inactive, default 1= Currently active
4	ERROR	0h	rh		Read-only. 0 = No Error, default 1= Error occurred
31dt5	reserved	0000000h			

Register:	EX_ADDR_4			Address:	54h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Slave address for expander 4.					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	EX_ADDR_4	0h	r	w	Slave address for expander 4.
31dt8	reserved	000000h			

Register:	EX_DATA_OUT_4			Address:	58h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Send data for expander 4.					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	EX_DATA_OUT_4	0h	r	w	Send data for expander 4.
31dt8	reserved	000000h			

Register:	EX_DATA_IN_4			Address:	5Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:	Read data for expander 4.					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	EX_DATA_IN_4	0h	rh		Read data for expander 4.
31dt8	reserved	000000h			

Register:	ERROR_SLAVE_ADDRESS			Address:	60h	
Bits:	31dt0	Reset value:	FAh	Attributes:	r	w
Description:	Fictive target address used to automatically create a stop condition following a failed start condition. Bit 0 activates this automatic action.					

Bit	Identifier	Reset	Attr.		Function / Description
0	ERR_ON_OFF	0h	r	w	Err_on_off = '0' : Error routine off (default). Err_on_off = '1' : Error routine on.
7dt1	ERROR_SLAVE_ADDRESS	7Dh	r	w	Fictive target address
31dt8	reserved	000000h			

Register:	SW_I2C_EN			Address:	64h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Software I2C enable: 0 = disabled 1 = enabled					

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	SW_I2C_EN	0h	r	w	Software I2C enable: 0 = disabled 1 = enabled
31dt1	reserved	00000000h			

Register:	SW_I2C_CTRL			Address:	68h	
Bits:	31dt0	Reset value:	Fh	Attributes:	r(h)	(w)
Description:	Software I2C control register.					

Bit	Identifier	Reset	Attr.		Function / Description
0	SW_SCL_O	1h	r	w	Software I2C clock output: SW_SCL_O = '1' : I2C clock output is tristate (default). SW_SCL_O = '0' : I2C clock output is '0'.
1	SW_SDA_O	1h	r	w	Software I2C data output: SW_SDA_O = '1' : I2C data output is tristate (default). SW_SDA_O = '0' : I2C data output is '0'.
2	SCL_I	1h	rh		I2C clock output: SCL_I = '1' : I2C clock line = '1'. SCL_I = '0' : I2C clock line is '0'.
3	SDA_I	1h	rh		I2C data output: SDA_I = '1' : I2C data line = '1'. SDA_I = '0' : I2C data line is '0'.
31dt4	reserved	0000000h			

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.8 OSPI (Octal-SPI)

Note

Please **note** that the ERTEC 200P-3 only supports Quad-SPI mode.

After a reset, the OSPI IP is configured as Standard-SPI mode (which supports all the SPI memories). The SPI mode can be set for the external SPI memory. The following SPI modes are supported:

- Standard-SPI
- Dual-SPI
- Quad-SPI
- Octal-SPI

The following clock rates apply for the clock output of the synchronous serial interfaces in line with the parameters of the OSPI and of the used system clock.

- F_{CLKOUT} = clk_sys_o / SCKDIV

Whereby only even values in the range of 2, 4, 6, …. 65534 apply for SCKDIV. For the parameter assignment of SCKDIV = 0 the output clock is disabled. With 125 MHz system clock, cycle rates in the range of 1.907 kHz (SCKDIV = 65534) to 62.5 MHz (SCKDIV = 2) are possible.

2.3.14.8.1 Configuration of OSPI

The following table shows the IP configuration of the OSPI.

Table 34: Configuration of OSPI

Parameter	Value
System Configuration	
APB Interface Type	APB 3.0
Enable APB Error response	No
APB Data Bus Width	32
Device Configuration	
Serial master or slave configuration	Serial Master
Include Enhanced Clock Ratio Architecture?	No
Maximum Transfer Size	32
Receive FIFO buffer depth	32 (32x32 Bits=128 Byte)
Transmit FIFO buffer depth	32 (32x32 Bits=128 Byte)
Number of slave select lines	1
Include Programmable RXD Sample Logic	Yes
Maximum RXD Sample Delay	255
Peripheral ID Code	0xFFFFFFFF

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Parameter	Value
External Configuration	
Include DMA handshaking interface signals?	True
Configure interrupt pinout	Combined Interrupt
Active interrupt level	Active high
Internal Configuration	
Hard code the frame format?	No
Default frame format	Motorola SPI
Default serial clock polarity	0
Default serial clock phase	0
Toggle slave select signal between frames when in SPI mode and SCPH=0?	No
SPI Parameters	
Select SPI mode	SPI Octal Mode
Enable enhanced SPI / IO Mapping	No
Include Dual transfer rate transfers in SPI frame format?	Yes
Include data strobe signal for rxd line?	Yes
Include data mask signal for data transfers in SPI mode?	No
Clocking Parameters	
Are pclk and ssi_clk synchronous?	Yes
Generate clock enable input for ssi_clk?	Yes
pclk to ssi_clk Synchronization Depth	2
ssi_clk to pclk Synchronization Depth	2

2.3.14.8.2 WR transfer monitoring

If write data is transferred in Octal-SPI, Quad-SPI and Dual-SPI mode to an external SPI memory, the CS signal (ss_0_n) remains active (low-active) at the SPI memory until all data from the TX-FIFO of the OSPI IP was transferred. A subsequent transfer of data to the TX-FIFO re-activates the CS signal (ss_0_n). That data is interpreted as a new Octal-SPI/Quad-SPI/Dual-SPI transfer and therefore starts with a command and address sequence.

In case APB write accesses to the TX FIFO are delayed and the TX FIFO has an underflow during the transfer of write data, the result is an unintended and thus erroneous start of a new Octal-SPI/Quad-SPI/Dual-SPI transfer. In order to recognize this state, a rising edge of ss_0_n is detected and stored in *SCRB2.OCTALSPI_REG.CS_RISE_EDGE* = 1. The bitfield can be reset by SW write access only.

After a rising CS edge has been recognized, ss_0_n signal from the OSPI IP is forced to inactive ('1') towards the corresponding GPIOx signal (ospi_ss_0_n_o). This ensures that subsequent APB

write accesses do not result in a non-correctable overwriting/falsifying of the data in the external SPI memory.

The last APB write access of an Octal-SPI/Quad-SPI/Dual-SPI transfer is to be executed under the mirror addresses:

- 4002_C400 - 4002_C5FF,
- 4002_C600 - 4002_C7FF (halfword swapping),
- 4002_C800 - 4002_C9FF (word swapping).

See chapter 2.4.2.

Accesses under this APB mirror address prevent further monitoring of the CS signal for a recognized rising edge by asserting set_cs_mon_dis_o which is stored under *SCRB2.OCTALSPI_REG.CS_MON_DIS* = 1. This is necessary because the ss_0_n signal correctly becomes inactive at the planned end of the OSPI/Quad-SPI/Dual-SPI transfer, but this fact must not be stored in *SCRB.OCTALSPI_REG.CS_RISE_EDGE*.

In order to check the correct transfer of the write data to the external SPI memory, *SCRB.OCTALSPI_REG.CS_RISE_EDGE* must be read after completion of the Octal-SPI/Quad-SPI/Dual-SPI transfer. The following applies for the Octal-SPI/Quad-SPI/Dual-SPI write transfer:

- *CS_RISE_EDGE* = '0': was transferred without interruption
- *CS_RISE_EDGE* = '1': was transferred with interruption and must be repeated

In both cases CS monitoring must be re-activated by the software by setting *SCRB.OCTALSPI_REG.CS_MON_DIS* = 0.

Note

Pay attention to the fact that with active WR transfer monitoring (*SCRB.OCTALSPI_REG.CS_MON_DIS* == 0) also a transfer of read data will be monitored. This means that prior to a read transfer WR transfer monitoring should be disabled (*SCRB.OCTALSPI_REG.CS_MON_DIS* = 1).

Note

In Standard-SPI mode "idling" of the TX FIFO will also deactivate ss_0_n, but in OSPI IP the command- and address-sequence is not initiated. The external SPI memory identifies a new activation of ss_0_n as a new command and address sequence. This can be prohibited if ss_0_n is controlled via GPIO function directly by SW. The controlling via OSPI IP shall not be used. Therefore, WR transfers can be done with "idling" of the TX FIFO. When *CS_RISE_EDGE* = '1', the last written data must be inspected and the missing data in the FLASH must be programmed.

Note

To restart the OSPI after an interruption of a write transfer, disable the OSPI and re-enable it.

2.3.14.8.3 Clock shift

In order to ease constraining the timing, sclk_out is shifted in relation to txd[7:0]. The signal sclk_out from the OSPI IP is shifted by:

- In case of maximum sclk_out frequency (*BAUDR.SCKDV* = 2) or *SPI_CTRLR0.SPI_DDR_EN* = 1 of OSPI IP: 1 cycle of clk_x2_i.
- In all other cases: 2 cycles of clk_x2_i.

As a result, setup and hold times of txd[7:0] are met easily.

Figure 43: OSPI clk shift

2.3.14.8.4 OSPI_IP is used without RXDS

This mode is set in the OSPI_IP, when *SPI_CTRLR0.RXDS_EN* bit is set to '0' (default).

The SSI_CLK of the OSPI_IP is used to capture the data from the OSPI memory device (read data).

The external delays can be compensated to some degree, when using the OSPI_IP internal delay (RX_SAMPLE_DLY).

Figure 44: OSPI_IP without RXDS

2.3.14.8.5 OSPI_IP is used with RXDS

This mode is set in the OSPI IP, when *SPI_CTRLR0.RXDS_EN* bit is set to '0': data strobe (RXDS) is not used to capture read data. Additionally, the register SCRB.OCTAL-SPI_STROBE_REG.RXDS_DELAY_MODE must be set to '0'.

2.3.14.8.5.1 OSPI_IP is used with internal RXDS

This mode is set in the OSPI IP, when *SPI_CTRLR0.RXDS_EN* bit is set to '1': data strobe (RXDS) is used to capture read data. The register OCTALSPI_STROBE_REG.RXDS_DELAY_MODE is set to '1'.

As this mode uses the RXDS signal of OSPI_IP to capture read data from an OSPI memory device, which does not support the RXDS signal, the RXDS is generated internally within the OSPI_SHELL by using the OSPI_CLK_FB:

- The **OSPI_CLK_FB is delayed** by the parameter OCTALSPI_STROBE_REG.SCK_DELAY:
  - With OCTALSPI_STROBE_REG.SCK_DELAY(4:0) the delay can be controlled in multiples of Tclk_i/2 (1 ns).
  - The parameter OCTALSPI_STROBE_REG.SCK_DELAY must be set once (not before a new read command is transmitted to the external memory device).
- the **delayed OSPI_CLK_FB is used as RXDS signal**:
  - the COUNTER starts to count in multiples of T_{SCK}, when ss_0_n (low active chip select of OSPI_IP) is activated.
  - the delayed OSPI_CLK_FB is used as RXDS signal, when the COUNTER value reaches OCTALSPI_STROBE_REG.RXDS_SETUP_CYCLES.
  - the RXDS signal is '0', when the COUNTER value reaches OCTALSPI_STROBE_REG.RXDS_SETUP_CYCLES + rxds_max_value.

By using mode_i[1:0], ddr_en, ndf_i[16:0], and dfs_i[4:0] signals:

- rxds_max_value is computed as:
  $(((ndf_i << \log_2(dfs_i+1)) >> ddr_en_i) >> mode_i)$ with
  - ndf_i = number of data frames to be received
  - dfs_i[4] = number of data bits per data frame – 1
  - ddr_en_i = double data rate
  - mode_i = SPI Mode (00 = Single, 01 = Double, 10 = Quad, 11 Oct)

Note

OCTALSPI_STROBE_REG.RXDS_SETUP_CYCLES must be updated before a read command is transmitted to the external memory device.

2.3.14.8.5.2 OSPI_IP is used with external RXDS

This mode is set in the OSPI_IP, when *SPI_CTRLR0.RXDS_EN* bit is set to '1': data strobe (RXDS) is used to capture read data. The register OCTALSPI_STROBE_REG.RXDS_DELAY_MODE is set to '0'.

As this mode uses the RXDS signal of OSPI_IP to capture read data from an OSPI memory device, which does support the RXDS signal. However, to avoid capturing more data than requested, the RXDS signal is gated internally within the OSPI_SHELL if required. When rxds_max_value was received RXDS is gated. Thus, no additional RXDS signals are forwarded to the OSPI-IP. Also, RXDS is gated if ss_0_n = '1' to avoid issues in OSPI IP due to incoming RXDS pulses when ss_0_n is already '1' (round trip delay).

[4] For the DFS32 value in 'CTRLR0' register only the values (7, 15, and 31) are allowed. Thus, when the OSPI is used with internal RXDS and RXDS_DELAY_MODE = 0, the number of bits per data frame is limited to byte, halfword and word.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.8.6 Memory swapping

The OSPI provides a mirrored address range to automatically perform byte-wise halfword swapping and byte-wise word swapping capabilities respectively. APB access to the mirrored address ranges of the OSPI therefore triggers the following functions (APB address range in OSPI address mapping, see chapter 2.4.2).

Table 35: Memory swapping

APB address range	Function	Serial data transfer
0x...400 – 0x...5FF	Standard OSPI access. No swapping function	
0x...600 – 0x...7FF	Swapping function: byte-wise swapping within halfword	
0x...800 – 0x...9FF	Swapping function: byte-wise swapping within word	

The OSPI must only be configured using the original address range (0x...400 – 0x...5FF), as access to the OSPI in the other APB address ranges is subject to the functions of the mirrored address ranges. Only OSPI data transfer over the mirrored addresses is useful.

2.3.14.8.7 Software usage in the OSPI

2.3.14.8.7.1 Peripheral parameter assignment

- In **Quad-SPI mode** (*CTRLR0.SPI_FRF* = 0x2) the Quad-SPI memory checks the signal status on data line DQ2 (→ Write_Protection, when DQ2 = 0) after the programming command (inactivating the OSPI CS signal). With the inactive CS signal the signal level at this ASIC pin depends on the internal or external pull. Dedicated I/Os are used, the signal level is defined by the used pulls.
- If in **Standard-SPI mode** (*CTRLR0.SPI_FRF* = 0x0) a Dual-/Quad-/OSPI-memory is connected, additionally to DQ0, SCK and SS_N, DQ1 is used.

2.3.14.8.7.2 Reset sequence

After a reset of the ASIC that did not affect the connected flash device, it is not known in which mode the flash device is. To make sure that the device returns to Standard-SPI mode, the following three different reset sequences must be executed by the OSPI.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Figure 45: OSPI reset sequence

Only after all three sequences have been executed the device is certainly in Standard-SPI mode.

NOTICE

As described above, all eight data lines are always used regardless of the mode of the memory device that is connected on the PCB. This must be considered if the assigned GPIOs shall be used with alternate functions and external PCB circuitry.

2.3.14.8.8 OSPI programming

Before reading out or programming Quad-SPI memory blocks, three configuration steps are required, depending on the selected operating mode (Standard, Dual, Quad, Octal).

1. GPIO configuration (for SPI standard mode)
2. OSPI IP configuration (for SPI standard mode)
3. Optional reconfiguration of GPIO logic, OSPI IP if the connected memory is to be operated in dual, quad and octal mode.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

The MICRON memory device N25Q032A13E is to be used for example for the subsequent use cases. It has an address width of 24 bits, commands are byte granular. The data width can be parameterized in the range of 1…4 bytes for all SPI modes and has been selected as an example in the following use cases.

The command execution always starts with writing a command byte, followed by an optional 24-bit address and, in the case of writing / programming, with user data. In the process all data is entered right-justified with WORD accesses in the TX FIFO.

In Standard-SPI mode, the 24-bit address is always written byte-granularly to the TX-FIFO (→ 3 TX-FIFO entries). In contrast the 24-bit address is always realized right-justified by a TX-FIFO entry in Dual-SPI, Quad-SPI or OSPI mode.

2.3.14.8.8.1 Reading the Standard-SPI mode memory

Typical OSPI IP command sequence after Standard-SPI configuration:

1. READ
   - With 24-bit address (TX FIFO entry)
   - Number of user data bytes to be read (register entry)

Sequence of the TX FIFO entries (about *OCTAL_SPI.DR0*) with a READ command:

31…	…24 23…	…16 15…	…8 7… …0	
Not used	Not used	Not used	Command 0x3	1. entry
Not used	Not used	Not used	Addr bits(23…16)	2. entry
Not used	Not used	Not used	Addr bits(15…8)	3. entry
Not used	Not used	Not used	Addr bits(7…0)	4. entry

Figure 46: Sequence of TX FIFO entries with a READ command

2.3.14.8.8.2 Programming/writing the Standard-SPI mode memory

Typical OSPI IP command sequence after Standard-SPI configuration:

1. WRITE_ENABLE
2. PAGE_PROGRAM
   - With 24-bit address (TX FIFO entry)
   - Number of user data bytes to be programmed (number of the TX FIFO entries and the parameterized TX FIFO width for user data →, for example, 1 byte)

Sequence of the TX FIFO entries (about *OCTAL_SPI.DR0*) with a PAGE_PROGRAM command:

31…	…24 23…	…16 15…	…8 7… …0	
Not used	Not used	Not used	Command 0x2	1. entry
Not used	Not used	Not used	Addr bits(23…16)	2. entry
Not used	Not used	Not used	Addr bits(15…8)	3. entry
Not used	Not used	Not used	Addr bits(7…0)	4. entry
Not used	Not used	Not used	1. payload byte	5. entry
Not used	Not used	Not used	2. payload byte	6. entry
Not used	Not used	Not used	…	n. entry

Figure 47: Sequence of TX FIFO entries with a PAGE_PROGRAM command

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.8.8.3 Reading the Quad-SPI mode memory

Typical OSPI IP command sequence based on the Standard-SPI configuration.

First reconfiguration of the external memory from Standard-SPI to Quad-SPI mode:

1. WRITE_ENABLE
2. WRITE_VOLATILE_CONFIGURATION
   - With user data byte = 0xAB (TX FIFO entry)
3. WRITE_ENABLE
4. WRITE_ENHANCED_VOLATILE_CONFIGURATION
   - With user data byte = 0x6F (TX FIFO entry)
   - Reconfiguration of the GPIO parameter assignment for Quad-SPI mode
   - Reconfiguration of the OSPI IP for Quad-SPI mode
   - Reading from the external memory in Quad-SPI mode
5. FAST_READ

Sequence of the TX FIFO entries (about *OCTAL_SPI.DR0*) with a FAST_READ command:

31. ..	...24 23.. .	...16 15.. .	...8 7...	...0	
Not used	Not used	Not used	Command 0xB		1. entry
Not used	Addr bits(23 ...16)	Addr bits(15 ...8)	Addr bits(7.. .0)		2. entry

Figure 48: Sequence of TX FIFO entries with a FAST_READ command

2.3.14.8.8.4 Programming/writing the Quad-SPI mode memory

Typical OSPI IP command sequence based on the Standard-SPI configuration.

First reconfiguration of the external memory from Standard-SPI to Quad-SPI mode:

1. WRITE_ENABLE
2. WRITE_VOLATILE_CONFIGURATION
   - With user data byte = 0xAB (TX FIFO entry)
3. WRITE_ENABLE
4. WRITE_ENHANCED_VOLATILE_CONFIGURATION
   - With user data byte = 0x6F (TX FIFO entry)
   - Reconfiguration of the GPIO parameter assignment for Quad-SPI mode
   - Reconfiguration of the OSPI IP for Quad-SPI mode
   - Writing / programming of the external memory in Quad-SPI mode
5. QUAD_INPUT_FAST_PROGRAM
   - With 24-bit address (TX FIFO entry)
   - Number of the user data bytes to be programmed (number of the TX FIFO entries and the parameterized TX FIFO width for user data →, for example, 4 bytes)

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Sequence of the TX FIFO entries (about *OCTAL_SPI.DR0*) with a QUAD_INPUT_FAST_PROGRAM command:

31...24	23...16	15...8	7...0	
Not used	Not used	Not used	Command 0x32	1. entry
Not used	Addr bits(23...16)	Addr bits(15...8)	Addr bits(7...0)	2. entry
4. payload byte	3. payload byte	2. payload byte	1. payload byte	3. entry
8. payload byte	7. payload byte	6. payload byte	5. payload byte	4. entry
...	...	...	...	n. entry

Figure 49: Sequence of TX FIFO entries with a QUAD_INPUT_FAST_PROGRAM command

2.3.14.8.9 Execution In Place (XIP)

Note

Execution In Place (XIP) is not supported.

2.3.14.8.10 Address mapping

Start-Address	End-Address	Modul/Memory-Name
400h	4FFh	OCT_SPI
500h	5FCh	OCTAL_SPI – mirrored
600h	6FFh	OCT_SPI – byte-wise half_word swapping
700h	7FFh	OCT_SPI – byte-wise half_word swapping – mirrored
800h	8FFh	OCT_SPI – byte-wise word swapping
900h	9FFh	OCT_SPI – byte-wise word swapping - mirrored

Module	Register/Memory	Read	Write	Access	Address
/OCT_SPI				**all**	
	/CTRLR0	r	(w)	all	0h
	/CTRLR1	r	(w)	all	4h
	/SSIENR	r	(w)	all	8h
	/MWCR	r	(w)	all	Ch
	/SER	r	(w)	all	10h
	/BAUDR	r	(w)	all	14h
	/TXFTLR	r	(w)	all	18h
	/RXFTLR	r	(w)	all	1Ch
	/TXFLR	r		all	20h
	/RXFLR	r		all	24h
	/SR	r		all	28h
	/IMR	r	(w)	all	2Ch
	/ISR	r		all	30h
	/RISR	r		all	34h
	/TXOICR	r		all	38h
	/RXOICR	r		all	3Ch
	/RXUICR	r		all	40h
	/MSTICR	r		all	44h
	/ICR	r		all	48h

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

	/DMACR	r	(w)	all	4Ch
	/DMATDLR	r	(w)	all	50h
	/DMARDLR	r	(w)	all	54h
	/IDR	r		all	58h
	/SSI_VERSION_ID	r		all	5Ch
	/DR0	r	w	all	60h
	/DR1	r	w	all	64h
	/DR2	r	w	all	68h
	/DR3	r	w	all	6Ch
	/DR4	r	w	all	70h
	/DR5	r	w	all	74h
	/DR6	r	w	all	78h
	/DR7	r	w	all	7Ch
	/DR8	r	w	all	80h
	/DR9	r	w	all	84h
	/DR10	r	w	all	88h
	/DR11	r	w	all	8Ch
	/DR12	r	w	all	90h
	/DR13	r	w	all	94h
	/DR14	r	w	all	98h
	/DR15	r	w	all	9Ch
	/DR16	r	w	all	A0h
	/DR17	r	w	all	A4h
	/DR18	r	w	all	A8h
	/DR19	r	w	all	ACh
	/DR20	r	w	all	B0h
	/DR21	r	w	all	B4h
	/DR22	r	w	all	B8h
	/DR23	r	w	all	BCh
	/DR24	r	w	all	C0h
	/DR25	r	w	all	C4h
	/DR26	r	w	all	C8h
	/DR27	r	w	all	CCh
	/DR28	r	w	all	D0h
	/DR29	r	w	all	D4h
	/DR30	r	w	all	D8h
	/DR31	r	w	all	DCh
	/DR32	r	w	all	E0h
	/DR33	r	w	all	E4h
	/DR34	r	w	all	E8h
	/DR35	r	w	all	ECh
	/RX_SAMPLE_DLY	r	(w)	all	F0h
	/SPI_CTRLR0	r	(w)	all	F4h
	/TXD_DRIVE_EDGE	r	(w)	all	F8h
	/RSVD	r		all	FCh

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.8.11 Register description

Module: OCT_SPI

Register:	CTRLR0			Address:	0h	
Bits:	31dt0	Reset value:	70000h	Attributes:	r	(w)
Description:	This register controls the serial data transfer. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register. Reset Value: SSI_CTRLR0_RST					

Bit	Identifier	Reset	Attr.	Function / Description
3dt0	DFS	0h	r	Data Frame Size. This register field is only valid when SSI_MAX_XFER_SIZE is configured to 16. If SSI_MAX_XFER_SIZE is configured to 32, then writing to this field will not have any effect. Selects the data frame length. When the data frame size is programmed to be less than 16 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You must right-justify transmit data before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data **Note**: When SSI_SPI_MODE is either set to "Dual" or "Quad" or "Octal" mode and SPI_FRF is not set to 2'b00. - DFS value should be multiple of 2 if SPI_FRF = 01, - DFS value should be multiple of 4 if SPI_FRF = 10, - DFS value should be multiple of 8 if SPI_FRF = 11. 0x3 (FRAME_04BITS): 4-bit serial data transfer 0x4 (FRAME_05BITS): 5-bit serial data transfer 0x5 (FRAME_06BITS): 6-bit serial data transfer

					0x6 (FRAME_07BITS): 7-bit serial data transfer 0x7 (FRAME_08BITS): 8-bit serial data transfer 0x8 (FRAME_09BITS): 9-bit serial data transfer 0x9 (FRAME_10BITS): 10-bit serial data transfer 0xa (FRAME_11BITS): 11-bit serial data transfer 0xb (FRAME_12BITS): 12-bit serial data transfer 0xc (FRAME_13BITS): 13-bit serial data transfer 0xd (FRAME_14BITS): 14-bit serial data transfer 0xe (FRAME_15BITS): 15-bit serial data transfer 0xf (FRAME_16BITS): 16-bit serial data transfer
5dt4	FRF	0h	r	w	Frame Format. Selects which serial protocol transfers the data. 0x0 (MOTOROLA_SPI): Motorolla SPI Frame Format 0x1 (TEXAS_SSP): Texas Instruments SSP Frame Format 0x2 (NS_MICROWIRE): National Microwire Frame Format 0x3 (RESERVED): Reserved value
6dt6	SCPH	0h	r	w	Serial Clock Phase. Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data are captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data are captured on the second edge of the serial clock. 0x0 (SCPH_MIDDLE): Serial clock toggles in middle of first data bit

					0x1 (SCPH_START): Serial clock toggles at start of first data bit
7dt7	SCPOL	0h	r	w	Serial Clock Polarity. Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the DW_apb_ssi master is not actively transferring data on the serial bus. 0x0 (SCLK_LOW): Inactive state of serial clock is low 0x1 (SCLK_HIGH): Inactive state of serial clock is high
9dt8	TMOD	0h	r	w	Transfer Mode. Selects the mode of transfer for serial communication. This field does not affect the transfer duplicity. Only indicates whether the receive or transmit data are valid. In transmit-only mode, data received from the external device is not valid and is not stored in the receive FIFO memory; it is overwritten on the next transfer. In receive-only mode, transmitted data are not valid. After the first write to the transmit FIFO, the same word is retransmitted for the duration of the transfer. In transmit-and-receive mode, both transmit and receive data are valid. The transfer continues until the transmit FIFO is empty. Data received from the external device are stored into the receive FIFO memory, where it can be accessed by the host processor. In eeprom-read mode, receive data is not valid while control data is being transmitted. When all control data is sent to the EEPROM, receive data becomes valid and transmit data becomes invalid. All

					data in the transmit FIFO is considered control data in this mode. This transfer mode is only valid when the DW_apb_ssi is configured as master device. 00 - Transmit & Receive 01 - Transmit Only 10 - Receive Only 11 - EEPROM Read When SSI_SPI_MODE is either set to "Dual" or "Quad" or "Octal" mode and SPI_FRF is not set to 2'b00. There are only two valid combinations: 10 - Read 01 - Write 0x0 (TX_AND_RX): Transmit & receive 0x1 (TX_ONLY): Transmit only mode or Write (SPI_FRF != 2'b00) 0x2 (RX_ONLY): Receive only mode or Read (SPI_FRF != 2'b00) 0x3 (EEPROM_READ): EEPROM Read mode
10dt10	RSVD_SLV_OE	0h	r		SLV_OE Reserved field - Read-only
11dt11	SRL	0h	r	w	Shift Register Loop. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. Can be used in both serial-slave and serial-master modes. When the DW_apb_ssi is configured as a slave in loopback mode, the ss_in_n and ssi_clk signals must be provided by an external source. In this mode, the slave cannot generate these signals because there is nothing to which to loop back 0x0 (NORMAL_MODE): Normal mode operation 0x1 (TESTING_MODE): Test mode: Tx & Rx shift reg connected

15dt12	CFS	0h	r	w	Control Frame Size. Selects the length of the control word for the Microwire frame format. 0x0 (SIZE_01_BIT): 1-bit Control Word 0x1 (SIZE_02_BIT): 2-bit Control Word 0x2 (SIZE_03_BIT): 3-bit Control Word 0x3 (SIZE_04_BIT): 4-bit Control Word 0x4 (SIZE_05_BIT): 5-bit Control Word 0x5 (SIZE_06_BIT): 6-bit Control Word 0x6 (SIZE_07_BIT): 7-bit Control Word 0x7 (SIZE_08_BIT): 8-bit Control Word 0x8 (SIZE_09_BIT): 9-bit Control Word 0x9 (SIZE_10_BIT): 10-bit Control Word 0xa (SIZE_11_BIT): 11-bit Control Word 0xb (SIZE_12_BIT): 12-bit Control Word 0xc (SIZE_13_BIT): 13-bit Control Word 0xd (SIZE_14_BIT): 14-bit Control Word 0xe (SIZE_15_BIT): 15-bit Control Word 0xf (SIZE_16_BIT): 16-bit Control Word
20dt16	DFS_32	7h	r	w	Data Frame Size in 32-bit transfer size mode. Used to select the data frame size in 32-bit transfer mode. These bits are only valid when SSI_MAX_XFER_SIZE is configured to 32. When the data frame size is programmed to be less than 32 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive

					FIFO zero-padded. You are responsible for making sure that transmit data is right-justified before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. **Note**: When SSI_SPI_MODE is either set to "Dual" or "Quad" or "Octal" mode and SPI_FRF is not set to 2'b00. - DFS value should be multiple of 2 if SPI_FRF = 0x01, - DFS value should be multiple of 4 if SPI_FRF = 0x10, - DFS value should be multiple of 8 if SPI_FRF = 0x11. 0x3 (FRAME_04BITS): 4-bit serial data transfer 0x4 (FRAME_05BITS): 5-bit serial data transfer 0x5 (FRAME_06BITS): 6-bit serial data transfer 0x6 (FRAME_07BITS): 7-bit serial data transfer 0x7 (FRAME_08BITS): 8-bit serial data transfer 0x8 (FRAME_09BITS): 9-bit serial data transfer 0x9 (FRAME_10BITS): 10-bit serial data transfer 0xa (FRAME_11BITS): 11-bit serial data transfer 0xb (FRAME_12BITS): 12-bit serial data transfer 0xc (FRAME_13BITS): 13-bit serial data transfer 0xd (FRAME_14BITS): 14-bit serial data transfer 0xe (FRAME_15BITS): 15-bit serial data transfer 0xf (FRAME_16BITS): 16-bit serial data transfer 0x10 (FRAME_17BITS): 17-bit serial data transfer 0x11 (FRAME_18BITS): 18-bit serial data transfer

					0x12 (FRAME_19BITS): 19-bit serial data transfer 0x13 (FRAME_20BITS): 20-bit serial data transfer 0x14 (FRAME_21BITS): 21-bit serial data transfer 0x15 (FRAME_22BITS): 22-bit serial data transfer 0x16 (FRAME_23BITS): 23-bit serial data transfer 0x17 (FRAME_24BITS): 24-bit serial data transfer 0x18 (FRAME_25BITS): 25-bit serial data transfer 0x19 (FRAME_26BITS): 26-bit serial data transfer 0x1a (FRAME_27BITS): 27-bit serial data transfer 0x1b (FRAME_28BITS): 28-bit serial data transfer 0x1c (FRAME_29BITS): 29-bit serial data transfer 0x1d (FRAME_30BITS): 30-bit serial data transfer 0x1e (FRAME_31BITS): 31-bit serial data transfer 0x1f (FRAME_32BITS): 32-bit serial data transfer
22dt21	SPI_FRF	0h	r	w	SPI Frame Format: Selects data frame format for Transmitting/Receiving the data Bits only valid when SSI_SPI_MODE is either set to "Dual" or "Quad" or "Octal" mode. When SSI_SPI_MODE is configured for "Dual Mode", 10/11 combination is reserved. When SSI_SPI_MODE is configured for "Quad Mode", 11 combination is reserved. 0x0 (STD_SPI_FRF): Standard SPI Frame Format 0x1 (DUAL_SPI_FRF): Dual SPI Frame Format 0x2 (QUAD_SPI_FRF): Quad SPI Frame Format 0x3 (OCTAL_SPI_FRF): Octal SPI Frame Format

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

23dt23	RSVD_CTRLR0_23	0h	r		CTRLR0_23 Reserved bits - Read Only
24dt24	SSTE	0h	r		Slave Select Toggle Enable. When operating in SPI mode with clock phase (SCPH) set to 0, this register controls the behavior of the slave select line (ss_*_n) between data frames. If this register field is set to 1 the ss_*_n line will toggle between consecutive data frames, with the serial clock (sclk) being held to its default value while ss_*_n is high; if this register field is set to 0 the ss_*_n will stay low and sclk will run continuously for the duration of the transfer. **Note**: This register is only valid when SSI_SCPH0_SSTOGGLE is set to 1.
31dt25	RSVD_CTRLR0	0h	r		SSTE Reserved bits - Read Only

Register:	CTRLR1			Address:	4h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	This register exists only when the DW_apb_ssi is configured as a master device. When the DW_apb_ssi is configured as a serial slave, writing to this location has no effect; reading from this location returns 0. Control register 1 controls the end of serial transfers when in receive-only mode. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
15dt0	NDF	0h	r	w	Number of Data Frames. When TMOD = 10 or TMOD = 11 , this register field sets the number of data frames to be continuously received by the DW_apb_ssi. The DW_apb_ssi continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer. When the DW_apb_ssi is

					configured as a serial slave, the transfer continues for as long as the slave is selected. Therefore, this register serves no purpose and is not present when the DW_apb_ssi is configured as a serial slave.
31dt16	RSVD_CTRLR1	0h	r		CTRLR1 Reserved bits - Read Only

Register:	SSIENR			Address:	8h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	This register enables and disables the DW_apb_ssi. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	SSI_EN	0h	r	w	SSI Enable. Enables and disables all DW_apb_ssi operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is impossible to program some of the DW_apb_ssi control registers when enabled. When disabled, the ssi_sleep output is set (after delay) to inform the system that it is safe to remove the ssi_clk, thus saving power consumption in the system. 0x0 (DISABLE): Disables Serial Transfer 0x1 (ENABLED): Enables Serial Transfer
31dt1	RSVD_SSIENR	0h	r		SSIENR Reserved bits - Read Only

Register:	MWCR			Address:	Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	This register controls the direction of the data word for the half-duplex Microwire serial protocol. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.	Function / Description

0dt0	MWMOD	0h	r	w	Microwire Transfer Mode. Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. 0x0 (NON_SEQUENTIAL): Non-Sequential Microwire Transfer 0x1 (SEQUENTIAL): Sequential Microwire Transfer
1dt1	MDD	0h	r	w	Microwire Control. Defines the direction of the data word when the Microwire serial protocol is used. When this bit is set to 0, the data word is received by the DW_apb_ssi MacroCell from the external serial device. When this bit is set to 1, the data word is transmitted from the DW_apb_ssi MacroCell to the external serial device. 0x0 (RECEIVE): SSI receives data 0x1 (TRANSMIT): SSI transmits data
2dt2	MHS	0h	r	w	Microwire Handshaking. Relevant only when the DW_apb_ssi is configured as a serial-master device. When configured as a serial slave, this bit field has no functionality. Used to enable and disable the busy/ready handshaking interface for the Microwire protocol. When enabled, the DW_apb_ssi checks for a ready status from the target slave, after the transfer of the last data/control bit, before clearing the BUSY status in the SR register. 0x0 (DISABLE): Handshaking interface is disabled 0x1 (ENABLED): Handshaking interface is enabled

31dt3	RSVD_MWCR	0h	r		MWCR Reserved bits - Read Only

Register:	SER			Address:	10h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	This register is valid only when the DW_apb_ssi is configured as a master device. When the DW_apb_ssi is configured as a serial slave, writing to this location has no effect; reading from this location returns 0. The register enables the individual slave select output lines from the DW_apb_ssi master. Up to 16 slave-select output pins are available on the DW_apb_ssi master. Register bits can be set or cleared when SSI_EN=0. If SSI_EN=1, then register bits can be set (to delay the slave select assertion while TX FIFO is getting filled) but cannot be cleared. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	SER	0h	r	w	Slave Select Enable Flag. Each bit in this register corresponds to a slave select line (ss_x_n) from the DW_apb_ssi master. When a bit in this register is set (1), the corresponding slave select line from the master is activated when a serial transfer begins. It should be noted that setting or clearing bits in this register have no effect on the corresponding slave select outputs until a transfer is started. Before beginning a transfer, you should enable the bit in this register that corresponds to the slave device with which the master wants to communicate. When not operating in broadcast mode, only one bit in this field should be set. 0x0 (NOT_SELECTED): No slave selected 0x1 (SELECTED): Slave is selected
31dt1	RSVD_SER	0h	r		SER Reserved bits - Read Only

Register:	BAUDR			Address:	14h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	This register is valid only when the DW_apb_ssi is configured as a master device. When the DW_apb_ssi is configured as a serial slave, writing to					

		this location has no effect; reading from this location returns 0. The register derives the frequency of the serial clock that regulates the data transfer. The 16-bit field in this register defines the ssi_clk divider value. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register. Reset Value: 0x0			
Bit	Identifier	Reset	Attr.		Function / Description
15dt0	SCKDV	0h	r	w	SSI Clock Divider. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out = Fssi_clk/SCKDV where SCKDV is any even value between 2 and 65534. For example: for Fssi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2 = 1.8432MHz
31dt16	RSVD_BAUDR	0h	r		BAUDR Reserved bits - Read Only

Register:	TXFTLR			Address:	18h
Bits:	31dt0	Reset value:	0h	Attributes:	r (w)
Description:		This register controls the threshold value for the transmit FIFO memory. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register. Reset Value: 0x0			
Bit	Identifier	Reset	Attr.		Function / Description
4dt0	TFT	0h	r	w	Transmit FIFO Threshold. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2-256; this register is sized to the number of address bits needed to access the

					FIFO. If you attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered. For information on the Transmit FIFO Threshold values, see the "Master SPI and SSP Serial Transfers" in the DW_apb_ssi Databook. ssi_txe_intr is asserted when TFT or less data entries are present in transmit FIFO
31dt5	RSVD_TXFTLR	0h	r		TXFTLR Reserved bits - Read Only

Register:	RXFTLR			Address:	1Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	This register controls the threshold value for the receive FIFO memory. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
4dt0	RFT	0h	r	w	Receive FIFO Threshold. Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2-256. This register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered. For information on the Receive FIFO Threshold values, see the "Master SPI and SSP Serial Transfers" in the DW_apb_ssi Databook. ssi_rxf_intr is asserted when RFT or more data entries are present in receive FIFO.
31dt5	RSVD_RXFTLR	0h	r		RXFTLR Reserved bits - Read Only

Register:	TXFLR			Address:	20h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	This register contains the number of valid data entries in the transmit FIFO memory. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
5dt0	TXTFL	0h	r		Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.
31dt6	RSVD_TXFLR	0h	r		TXFLR Reserved bits - Read Only

Register:	RXFLR			Address:	24h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	This register contains the number of valid data entries in the receive FIFO memory. This register can be ready at any time. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
5dt0	RXTFL	0h	r		Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.
31dt6	RSVD_RXFLR	0h	r		RXFLR Reserved bits - Read Only

Register:	SR			Address:	28h	
Bits:	31dt0	Reset value:	6h	Attributes:	r	
Description:	This is a read-only register used to indicate the current transfer status, FIFO status, and any transmission/reception errors that may have occurred. The status register may be read at any time. None of the bits in this register request an interrupt. Reset Value: 0x6					

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	BUSY	0h	r		SSI Busy Flag. When set, indicates that a serial transfer is in progress; when cleared indicates that the DW_apb_ssi is idle or disabled. 0x0 (INACTIVE): DW_apb_ssi is idle or disabled 0x1 (ACTIVE): DW_apb_ssi is actively transferring data

1dt1	TFNF	1h	r		Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0x0 (FULL): Transmit FIFO is full 0x1 (NOT_FULL): Transmit FIFO is not Full
2dt2	TFE	1h	r		Transmit FIFO Empty. When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0x0 (NOT_EMPTY): Transmit FIFO is not empty 0x1 (EMPTY): Transmit FIFO is empty
3dt3	RFNE	0h	r		Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. 0x0 (EMPTY): Receive FIFO is empty 0x1 (NOT_EMPTY): Receive FIFO is not empty
4dt4	RFF	0h	r		Receive FIFO Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0x0 (NOT_FULL): Receive FIFO is not full 0x1 (FULL): Receive FIFO is full
5dt5	RSVD_TXE	0h	r		TXE Reserved field - read-only
6dt6	DCOL	0h	r		Data Collision Error. Relevant only when the DW_apb_ssi is configured as a master device. This bit will be set if ss_in_n input is asserted by other master,

					when the DW_apb_ssi master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. 0x0 (NO_ERROR_CONDITION): No Error 0x1 (TX_COLLISION_ERROR): Transmit Data Collision Error
31dt7	RSVD_SR	0h	r		SR Reserved bits - Read Only

Register:	IMR			Address:	2Ch	
Bits:	31dt0	Reset value:	3Fh	Attributes:	r	(w)
Description:	This read/write reigster masks or enables all interrupts generated by the DW_apb_ssi. When the DW_apb_ssi is configured as a slave device, the MSTIM bit field is not present. This changes the reset value from 0x3F for serial-master configurations to 0x1F for serial-slave configurations. Reset Value: (SSI_IS_MASTER == 1) ? 0x3F : 0x1F					

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	TXEIM	1h	r	w	Transmit FIFO Empty Interrupt Mask 0x0 (MASKED): ssi_txe_intr interrupt is masked 0x1 (UNMASKED): ssi_txe_intr interrupt is not masked
1dt1	TXOIM	1h	r	w	Transmit FIFO Overflow Interrupt Mask 0x0 (MASKED): ssi_txo_intr interrupt is masked 0x1 (UNMASKED): ssi_txo_intr interrupt is not masked
2dt2	RXUIM	1h	r	w	Receive FIFO Underflow Interrupt Mask 0x0 (MASKED): ssi_rxu_intr interrupt is masked 0x1 (UNMASKED): ssi_rxu_intr interrupt is not masked
3dt3	RXOIM	1h	r	w	Receive FIFO Overflow Interrupt Mask 0x0 (MASKED): ssi_rxo_intr interrupt is masked

Copyright © Siemens AG 2023. All rights reserved

					0x1 (UNMASKED): ssi_rxo_intr interrupt is not masked
4dt4	RXFIM	1h	r	w	Receive FIFO Full Interrupt Mask 0x0 (MASKED): ssi_rxf_intr interrupt is masked 0x1 (UNMASKED): ssi_rxf_intr interrupt is not masked
5dt5	MSTIM	1h	r	w	Multi-Master Contention Interrupt Mask. This bit field is not present if the DW_apb_ssi is configured as a serial-slave device. 0x0 (MASKED): ssi_mst_intr interrupt is masked 0x1 (UNMASKED): ssi_mst_intr interrupt is not masked
31dt6	RSVD_IMR	0h	r		IMR Reserved bits - Read Only

Register:	ISR			Address:	30h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	This register reports the status of the DW_apb_ssi interrupts after they have been masked. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	TXEIS	0h	r		Transmit FIFO Empty Interrupt Status 0x0 (INACTIVE): ssi_txe_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_txe_intr interrupt is active after masking
1dt1	TXOIS	0h	r		Transmit FIFO Overflow Interrupt Status 0x0 (INACTIVE): ssi_txo_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_txo_intr interrupt is active after masking
2dt2	RXUIS	0h	r		Receive FIFO Underflow Interrupt Status 0x0 (INACTIVE): ssi_rxu_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_rxu_intr interrupt is active after masking
3dt3	RXOIS	0h	r		Receive FIFO Overflow Interrupt Status 0x0 (INACTIVE): ssi_rxo_intr interrupt is not active after masking

					0x1 (ACTIVE): ssi_rxo_intr interrupt is active after masking
4dt4	RXFIS	0h	r		Receive FIFO Full Interrupt Status 0x0 (INACTIVE): ssi_rxf_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_rxf_intr interrupt is full after masking
5dt5	MSTIS	0h	r		Multi-Master Contention Interrupt Status. This bit field is not present if the DW_apb_ssi is configured as a serial-slave device. 0x0 (INACTIVE): ssi_mst_intr interrupt not active after masking 0x1 (ACTIVE): ssi_mst_intr interrupt is active after masking
31dt6	RSVD_ISR	0h	r		ISR Reserved bits - Read Only

Register:	RISR			Address:	34h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	This read-only register reports the status of the DW_apb_ssi interrupts prior to masking. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	TXEIR	0h	r		Transmit FIFO Empty Raw Interrupt Status 0x0 (INACTIVE): ssi_txe_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_txe_intr interrupt is active prior masking
1dt1	TXOIR	0h	r		Transmit FIFO Overflow Raw Interrupt Status 0x0 (INACTIVE): ssi_txo_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_txo_intr interrupt is active prior masking
2dt2	RXUIR	0h	r		Receive FIFO Underflow Raw Interrupt Status 0x0 (INACTIVE): ssi_rxu_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_rxu_intr interrupt is active prior to masking
3dt3	RXOIR	0h	r		Receive FIFO Overflow Raw Interrupt Status 0x0 (INACTIVE): ssi_rxo_intr interrupt is active prior masking

					0x1 (ACTIVE): ssi_rxo_intr interrupt is not active prior to masking
4dt4	RXFIR	0h	r		Receive FIFO Full Raw Interrupt Status 0x0 (INACTIVE): ssi_rxf_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_rxf_intr interrupt is active prior to masking
5dt5	MSTIR	0h	r		Multi-Master Contention Raw Interrupt Status. This bit field is not present if the DW_apb_ssi is configured as a serial-slave device. 0x0 (INACTIVE): ssi_mst_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_mst_intr interrupt is active prior masking
31dt6	RSVD_RISR	0h	r		RISR Reserved bits - Read Only

Register:	TXOICR			Address:	38h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	Transmit FIFO Overflow Interrupt Clear Register. Reset Value: 0x0					
Bit	Identifier	Reset	Attr.		Function / Description	
0dt0	TXOICR	0h	r		Clear Transmit FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr interrupt; writing has no effect.	
31dt1	RSVD_TXOICR	0h	r		TXOICR Reserved bits - Read Only	

Register:	RXOICR			Address:	3Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	Receive FIFO Overflow Interrupt Clear Register. Reset Value: 0x0					
Bit	Identifier	Reset	Attr.		Function / Description	
0dt0	RXOICR	0h	r		Clear Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect.	

31dt1	RSVD_RXOICR	0h	r		RXOICR Reserved bits - Read Only

Register:	RXUICR			Address:	40h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	Receive FIFO Underflow Interrupt Clear Register. Reset Value: 0x0					
Bit	Identifier	Reset	Attr.		Function / Description	
0dt0	RXUICR	0h	r		Clear Receive FIFO Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect.	
31dt1	RSVD_RXUICR	0h	r		RXUICR Reserved bits - Read Only	

Register:	MSTICR			Address:	44h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	Multi-Master Interrupt Clear Register. Reset Value: 0x0					
Bit	Identifier	Reset	Attr.		Function / Description	
0dt0	MSTICR	0h	r		Clear Multi-Master Contention Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect.	
31dt1	RSVD_MSTICR	0h	r		MSTICR Reserved bits - Read Only	

Register:	ICR			Address:	48h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	Interrupt Clear Register. Reset Value: 0x0					
Bit	Identifier	Reset	Attr.		Function / Description	
0dt0	ICR	0h	r		Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the ssi_txo_intr, ssi_rxu_intr, ssi_rxo_intr, and the ssi_mst_intr interrupts. Writing to this register has no effect.	
31dt1	RSVD_ICR	0h	r		ICR Reserved bits - Read Only	

Register:	DMACR			Address:	4Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	This register is only valid when DW_apb_ssi is configured with a set of DMA Controller interface signals (SSI_HAS_DMA = 1). When DW_apb_ssi is not configured for DMA operation, this register will not exist and writing to the register's address will have no effect; reading from this register address will return zero. The register is used to enable the DMA Controller interface operation. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	RDMAE	0h	r	w	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel 0x0 (DISABLE): Receive DMA disabled 0x1 (ENABLED): Receive DMA enabled
1dt1	TDMAE	0h	r	w	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. 0x0 (DISABLE): Transmit DMA disabled 0x1 (ENABLED): Transmit DMA enabled
31dt2	RSVD_DMACR	0h	r		DMACR Reserved bits - Read Only

Register:	DMATDLR			Address:	50h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	This register is only valid when the DW_apb_ssi is configured with a set of DMA interface signals (SSI_HAS_DMA = 1). When DW_apb_ssi is not configured for DMA operation, this register will not exist and writing to its address will have no effect; reading from its address will return zero. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
4dt0	DMATDL	0h	r	w	Transmit Data Level.

					This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1. For information on the DMATDL decode values, see the "Slave SPI and SSP Serial Transfers" section in the DW_apb_ssi Databook. dma_tx_req is asserted when DMATDL or less data entries are present in the transmit FIFO
31dt5	RSVD_DMATDLR	0h	r		DMATDLR Reserved bits - Read Only

Register:	DMARDLR			Address:	54h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	This register is only valid when DW_apb_ssi is configured with a set of DMA interface signals (SSI_HAS_DMA = 1). When DW_apb_ssi is not configured for DMA operation, this register will not exist and writing to its address will have no effect; reading from its address will return zero. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
4dt0	DMARDL	0h	r	w	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMAE=1. For information on the DMARDL decode values, see the "Slave SPI and SSP Serial Transfers" section in the DW_apb_ssi Databook. dma_rx_req is asserted when DMARDL or more valid data entries are present in the receive FIFO.
31dt5	RSVD_DMARDLR	0h	r		DMARDLR Reserved bits - Read Only

Register:	IDR	Address:	58h

Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	
Description:	This register contains the peripherals identification code, which is written into the register at configuration time using coreConsultant. Reset Value: SSI_ID					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	IDCODE	ffffffffh	r		Identification code. The register contains the peripheral's identification code, which is written into the register at configuration time using CoreConsultant.

Register:	SSI_VERSION_ID			Address:	5Ch	
Bits:	31dt0	Reset value:	3430322Ah	Attributes:	r	
Description:	This read-only register stores the specific DW_apb_ssi component version. Reset Value: SSI_VERSION_ID					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	SSI_COMP_VERSION	3430322ah	r		Contains the hex representation of the Synopsys component version. Consists of ASCII value for each number in the version, followed by *. For example 32_30_31_2A represents the version 2.01*.

Register:	DR0			Address:	60h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable.					

		Reset Value: 0x0			
Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR1			Address:	64h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration	

					parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR2			Address:	68h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR3			Address:	6Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w

Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR4			Address:	70h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0.					

	Note: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR5			Address:	74h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR6			Address:	78h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the

					register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR7			Address:	7Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR8			Address:	80h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for					

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

	the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR9			Address:	84h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory					

		map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0			
Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR10			Address:	88h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					
Bit	Identifier	Reset	Attr.	Function / Description		

31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR11			Address:	8Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid.

					Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR12			Address:	90h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR13			Address:	94h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits					

	are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR14			Address:	98h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same					

		effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0			
Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR15			Address:	9Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					
Bit	Identifier	Reset	Attr.	Function / Description		

31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR16			Address:	A0h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid.

					Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR17			Address:	A4h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR18			Address:	A8h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits					

	are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR19			Address:	ACh	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same					

		effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0			
Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR20			Address:	B0h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:		The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0				
Bit	Identifier	Reset	Attr.	Function / Description	

31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR21			Address:	B4h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid.

					Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR22			Address:	B8h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR23			Address:	BCh	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits					

	are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR24			Address:	C0h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same					

		effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0			
Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR25			Address:	C4h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					
Bit	Identifier	Reset	Attr.	Function / Description		

31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR26			Address:	C8h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid.

					Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR27			Address:	CCh	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR28			Address:	D0h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits					

	are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR29			Address:	D4h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same					

		effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0			
Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR30			Address:	D8h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:		The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0				
Bit	Identifier	Reset	Attr.	Function / Description	

31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR31			Address:	DCh	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid.

					Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR32			Address:	E0h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR33			Address:	E4h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits					

	are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR34			Address:	E8h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same					

		effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0			
Bit	Identifier	Reset	Attr.		Function / Description
31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	DR35			Address:	ECh	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:		The DW_apb_ssi data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0. **Note**: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable. Reset Value: 0x0				
Bit	Identifier	Reset	Attr.	Function / Description	

31dt0	DR	0h	r	w	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer.

Register:	RX_SAMPLE_DLY			Address:	F0h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	This register is only valid when the DW_apb_ssi is configured with rxd sample delay logic (SSI_HAS_RX_SAMPLE_DELAY==1). When the DW_apb_ssi is not configured with rxd sample delay logic, this register will not exist and writing to its address location will have no effect; reading from its address will return zero. This register control the number of ssi_clk cycles that are delayed (from the default sample time) before the actual sample of the rxd input occurs. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register. Reset Value: 0x0					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	RSD	0h	r	w	Rxd Sample Delay. This register is used to delay the sample of the rxd input port. Each value represents a single ssi_clk delay on the sample of rxd. **Note**: If this register is programmed with a value that exceeds the depth of the internal shift registers (SSI_RX_DLY_SR_DEPTH) zero delay will be applied to the rxd sample.
31dt8	RSVD_RX_SAMPLE_DLY	0h	r		SAMPLE_DLY Reserved bits - Read Only

Register:	SPI_CTRLR0			Address:	F4h	
Bits:	31dt0	Reset value:	200h	Attributes:	r	(w)
Description:	This register is valid only when SSI_SPI_MODE is either set to "Dual" or "Quad" or "Octal" mode. This register is used to control the serial data transfer in SPI mode of operation. The register is only relevant when SPI_FRF is set to either 01 or 10 or 11. It is not possible to write to this register when the DW_apb_ssi is enabled (SSI_EN=1). The DW_apb_ssi is enabled and disabled by writing to the SSIENR register. Reset Value: 0x00000200					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	TRANS_TYPE	0h	r	w	Address and instruction transfer format. Selects whether DW_apb_ssi will transmit instruction/address either in Standard SPI mode or the SPI mode selected in CTRLR0.SPI_FRF field. 00 - Instruction and Address will be sent in Standard SPI Mode. 01 - Instruction will be sent in Standard SPI Mode and Address will be sent in the mode specified by CTRLR0.SPI_FRF. 10 - Both Instruction and Address will be sent in the mode specified by SPI_FRF. 11 - Reserved.
5dt2	ADDR_L	0h	r	w	Address Length. This bit defines Length of Address to be transmitted. Only after this much bits are programmed in to the FIFO the transfer can begin. For information on the ADDR_Ldecode value, see "Read Operation in Enhanced SPI Modes" section in the DW_apb_ssi Databook. 0x0 (ADDR_L_0): 0-bit Address Width 0x1 (ADDR_L_1): 4-bit Address Width 0x2 (ADDR_L_2): 8-bit Address Width 0x3 (ADDR_L_3): 12-bit Address Width

					0x4 (ADDR_L_4): 16-bit Address Width 0x5 (ADDR_L_5): 20-bit Address Width 0x6 (ADDR_L_6): 24-bit Address Width 0x7 (ADDR_L_7): 28-bit Address Width 0x8 (ADDR_L_8): 32-bit Address Width 0x9 (ADDR_L_9): 36-bit Address Width 0xa (ADDR_L_10): 40-bit Address Width 0xb (ADDR_L_11): 44-bit Address Width 0xc (ADDR_L_12): 48-bit Address Width 0xd (ADDR_L_13): 52-bit Address Width 0xe (ADDR_L_14): 56-bit Address Width 0xf (ADDR_L_15): 60-bit Address Width
7dt6	RSVD_SPI_CTRLR0_6_7	0h	r		CTRLR0_6_7 Reserved bits - Read Only
9dt8	INST_L	2h	r	w	Instruction Length Dual/Quad/Octal mode instruction length in bits. 0x0 (INST_L_0): 0-bit (No Instruction) 0x1 (INST_L_1): 4-bit Instruction 0x2 (INST_L_2): 8-bit Instruction 0x3 (INST_L_3): 16-bit Instruction
10dt10	RSVD_SPI_CTRLR0_10	0h	r		CTRLR0_10 Reserved bits - Read Only
15dt11	WAIT_CYCLES	0h	r	w	Wait cycles Number of wait cycles in Dual/Quad/Octal mode between control frames transmit and data reception. This value is specified as number of SPI clock cycles. For information on the WAIT_CYCLES decode value, see "Read Operation in Enhanced SPI Modes" section in the DW_apb_ssi Databook.
16dt16	SPI_DDR_EN	0h	r	w	SPI DDR Enable bit.

					This will enable Dual-data rate transfers in Dual/Quad/Octal frame formats of SPI.
17dt17	INST_DDR_EN	0h	r	w	Instruction DDR Enable bit. This will enable Dual-data rate transfer for Instruction phase.
18dt18	SPI_RXDS_EN	0h	r	w	Read data strobe enable bit. Once this bit is set to 1 DW_apb_ssi will use Read data strobe (RXDS) to capture read data in DDR mode.
31dt19	RSVD_SPI_CTRLR0	0h	r		SPI_CTRLR0 Reserved bits - Read Only

Register:	TXD_DRIVE_EDGE				Address:	F8h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	This Register is valid only when SSI_HAS_DDR is equal to 1. This register is used to control the driving edge of TXD register in DDR mode. It is not possible to write to this register when the DW_apb_ssi is enabled (SSI_EN=1). The DW_apb_ssi is enabled and disabled by writing to the SSIENR register. Reset Value: 0x0						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	TDE	0h	r	w	TXD Drive edge - value of which decides the driving edge of tramit data. The maximum value of this regster is = (BAUDR/2) -1.		
31dt8	RSVD_TXD_DRIVE_EDGE	0h	r		DRIVE_EDGE Reserved bits - Read Only		

Register:	RSVD				Address:	FCh	
Bits:	31dt0	Reset value:		0h	Attributes:	r	
Description:	RSVD - Reserved address location.						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	RSVD	0h	r		RSVD 31to0 Reserved address location		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.9 SPI1/2 (Serial Peripheral Interface)

The SPI1/2 interfaces are "Master-Slave" SPI function blocks. They are connected to the APB bus over a 16-bit interface.

The SPI interface of ERTEC 200P is implemented with an ARM IP, the *ARM PrimeCellTM Synchronous Serial Port Master and Slave (SSPMS) PL021*.

The basic frequency for bit rate generation is 125 MHz.

The macro supports the following modes:

- Motorola SPI-compatible mode
- Texas Instruments Synchronous Serial Interface
- National Semiconductor Microwire Interface

The SPI macro has the following features:

- Separate 16-bit wide and 8-entry deep receive and transmit FIFOs
- Programmable bit rate:
  Master: 1922.27 Bd - 62.5 MBd (theoretical), constraining is limited to 25 MBd
  Slave: max. bit rate 10.42 MBd
- Programmable data frame size of between 4 and 16 bits

The SPI module can be controlled by ARM926 or by the GDMA controller. Access from the external host over the host interface is also possible, but interrupts are not supported in this case. The module FIFO interrupt is not activated or deactivated until 4 entries have been made (in line with the transfer direction).

As the FIFO submodule for GDMA operation cannot be disabled, a module modification is required for independent GDMA operation (no ARM926 activity). For the receiver, the internal SPI module signal 'SPI1/2_SSPRXDMA, corresponding to RNE: Receive FIFO not empty' is sent to the module border to display the receive FIFO status for the GDMA. For the transmitter, the interrupt signal 'SPI1/2_SSPTXINTR, corresponding to the state: Transmit FIFO is half full or less' is used to display the transmit FIFO status for the GDMA.

- SPI1/2_SSPRXDMA:
  The RX FIFO contains at least one character and is not empty. As the SPI cannot know when the last character was read, the GDMA must operate in SINGLE-byte access mode (AHB). 1 to 65536 characters can therefore be transferred per DMA request; the job consists of a transfer entry.
- SPI1/2_SSPTXINTR:
  The SSPTXINTR interrupt (Transmit FIFO is half full or less) must be enabled for operation. The TX FIFO is at least half empty and therefore has space for 4 characters. The GDMA must operate in INCR4-byte access mode (AHB) to ensure that there is no FIFO overrun. If the transfer length is not modulo 4, the remaining characters are transferred by the GDMA controller with the INCR burst byte (undefined length). 1 to 65536 characters can therefore be transferred per DMA request; the job consists of a transfer entry.

The following SPI interrupts are connected to the ARM interrupt controller (IRQ13-16) (2.3.5.9):

- SPI1/2_SSPINTR Group interrupt
- SPI1/2_SSPRORINTR Overrun error interrupt

For byte-specific operation of the SPI1/2 interfaces by ARM926, the following internal status from the SPI modules is sent to the ARM interrupt controller (ARM-ICU) (IRQ28-31, see 2.3.5.9):

- SPI1/2_RNE Receive FIFO not empty (corresponds to SPI1/2_SSPRXDMA)
- SPI1/2_TFE Transmit FIFO empty

Copyright © Siemens AG 2023. All rights reserved

The status information is used from the internal SPI-IP status register 'SSPSR'. This ensures the timing required for steady redundant communication between IM modules.

The SPI1/2 interfaces share the external pins with GPIOs/XHIF. The SPI1/2 interfaces are only available if the GPIO control registers are programmed accordingly or the relevant SPI interface is set as the boot medium with the boot pins. In this case, the boot loader must enable the SPI interface in the internal ROM with the required GPIO control register configuration.

Each SPI1/2 interface supports the following external signals:

- SFRMOUT (Output): Serial frame output (Master)
- SCLKOUT (Output): Serial clock output (Master)
- SSPCTLOE (Output): Output enable signal for SCLKOUT and SFRMOUT
- SSPTXD (Output): Serial data output
- SSPOE (Output): Output enable signal when SSPTXD is valid
- SFRMIN (Input): Serial frame input (Slave)
- SCLKIN (Input): Serial clock input (Slave)
- SSPRXD (Input): Serial data input

If the SPI1/2 output signals are enabled at the GPIO, they drive the outputs directly and are not subject to output enable control.

The following baud rates apply for the clock output of the synchronous serial interfaces in line with the SPI1/2 parameters.

- F_{CLKOUT} = 125 MHz / (CPSDVR*(1+SCR))

The following applies for the parameters:

- SPI master: CPSDVR := (2..254), multiples of 2 only
- SCR := (0..255)
- SPI slave: CPSDVR := (4..254), multiples of 2 only
- SCR := (2..255)

This produces baud rates ranging from 62.5 MBd (master)/10.42 MBd (slave) down to 1922.27 Bd. The timing of the master for this baud rate cannot, however, be accessed from the pad cell or the board; it is limited 25 MBaud by constraining.

2.3.14.9.1 Features of the SPI interface

- Can be used as SPI master and SPI slave.
- Maximum data rate (with 125 MHz APB clock):
  - 62.5 Mbps in master mode (1/2 * APB clock frequency) → limited to 25 Mdps
  - 10.42 Mbps in slave mode (1/12 * APB clock frequency)
- Length of data frame:
  - 4 to 16 bits, adjustable
- Data bit transmission sequence:
  - MSB first (LSB first is not supported by SSPMS IP)

Copyright © Siemens AG 2023. All rights reserved

- DMA request signals
  A DMA request signal becomes active if:
  - The transmit FIFO is empty or
  - The receive FIFO contains at least one entry.
- Interrupts
  A group interrupt is activated if:
  - the transmit FIFO is empty,
  - the transmit FIFO is half full or emptier,
  - the transmit FIFO is not full,
  - the receive FIFO is not empty,
  - the receive FIFO is half full or fuller or
  - the receive FIFO overruns.

  The interrupt sources can be masked individually.

Note

Switching between master and slave mode during operation

According to the specifications, the SSPMS IP does not support dynamic toggling between master and slave mode.

2.3.14.9.1.1 SSPMS IP issues

If the SPI-IP is operated in TI mode (*SSPCR0.FRF* = "01"), the unused bits from the 16-bit data register whose used width is configured with *SSPCR0.DSS* are not defined in terms of data content and must therefore be masked out by the SW after being read.

2.3.14.9.2 SSPMS IP extension

The SSPMS IP used for the ERTEC 200P SPI interface is not the original version; it has the following additional interface signals:

- "Receive FIFO not empty" signal (SSP_SSPRXDMA)
- "Transmit FIFO empty" signal (SSP_TFE)

These signals contain the values of the RNE and TFE bits of the 'SSPSR' register.

2.3.14.9.3 SSPMS IP integration

The SSPMS IP is available either over the GPIOs (dedicated) (see Figure 50) in line with the SCRB register 'SPI_MODE' (see chapter 2.3.14.11.20) or as MISO/MOSI at the same GPIOs (see Figure 51).

Copyright © Siemens AG 2023. All rights reserved

Figure 50: SSP_SHELL

Figure 51: SPI block diagram

The following SPI1/2 signals can be alternatively assigned at the GPIOs (alternate functions) in line with SPI_MODE (see 2.3.14.11.20):

SPI1	**SPI2**	***SPI_MODE* = DIRECT**	***SPI_MODE* = MISO_MOSI**
GPIO16(A), GPIO40(C)	GPIO12(C), GPIO24(A),	SPI_1/2_SCLKOUT_a/_b	SPI1/2_CLK_a/_b
GPIO17(A), GPIO41(C)	GPIO13(C), GPIO25(A),	SPI_1/2_SFRMOUT_a/_b	SPI1/2_FRAME_N_a/_b
GPIO18(A), GPIO42(C)	GPIO14(C), GPIO26(A),	SPI_1/2_SSPTXD_a/_b	SPI1/2_MOSI_a/_b

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

SPI1	SPI2	*SPI_MODE* = DIRECT	*SPI_MODE* = MISO_MOSI
GPIO19(A), GPIO43(C)	GPIO15(C), GPIO27(A),	SPI_1/2_SSPRXD_a/_b	SPI1/2_MISO_a/_b

Note

If SPI1/2_FRAME_N_a/_b is not used, the corresponding alternate function is assigned the blocking value = '0' (see chapter 2.3.14.10.5.2).

Note

Integration notes:

- Implementation for SPI Interface
  The SSPMS IP does not have a special SPI interface. The IP interface signals SSPOE, SSPTXD, SSPRXD, SSPCTLOE, SCLKOUT, SCLKIN, SFRMOUT, SFRMIN are therefore implemented as SPI signals (SPI_MASTER_OE_O_N, SPI_SLAVE_OE_O_N, SPI_OUT_O, SPI_S_IN_I, SPI_M_IN_I, SPI_SSPCTLOE_O_N, SPI_SCLKOUT_O, SPI_SCLKIN_I, SPI_SFRMOUT_O_N, SPI_SFRMIN_I_N) as shown in the figure above when SPI_MODE = MISO_MOSI.
- Reset
  The SPI-IP cannot be reset dedicatedly. **No** SPI reset is required for switching from master to slave mode (this function has already been employed in previous projects, as the IP has already been used in a number of cases). If a reset is nevertheless required, it must be done, e.g., with SCRB.ASYN_RES_CTRL_REG.RES_SOFT of ERTEC 200P, please remember that this will lead to performance losses (ASIC re-initialization).

Note

Operation without *SPI_FRAME_N*:

If the ERTEC 200P is connected as SPI slave to a standard controller (SPI master) that does not support the SPI_FRAME_N signal, the ERTEC 200P is to be set as follows:

- Configuration of Motorola format: *SSPCR0.FRF* = '00' (see chapter 2.3.14.9.7) AND
- Configuration of the SPI clock phase: *SSPCR0.SPH* = '1' (see chapter 2.3.14.9.7) AND
- GPIO pin SPI_FRAME_N (refer to the ERTEC 200P-3 data sheet)
  - Connect with ext. pull-down OR
  - Connect with int. pull-down (*PULLxx_yyGPIO* = '11', see 2.3.14.11.15.1) OR
  - Do not select the alternate function (blocking value = '0' is active).

Please **note** the following restrictions in this operating mode:

- In the SPI status register, *SSPSR.BSY* (see 2.3.14.9.7) is not reset at the end of the transfer.
- Following a RESET, "zero data" is transferred when the first character (4-bit…16-bit, *SSPCR0.DSS*) is transferred from the SPI slave to the SPI master (return direction).

Note on communication between SPI master and SPI slave

Communication between SPI master and SPI slave is always in full-duplex mode, i.e., it is not possible for only the master or only the slave to send data. This means for example that when data is only to be transferred from the SPI master to the SPI slave, the master still clocks in the receive signal SSPRXD and enters the corresponding data words in the receive FIFO. If the receive FIFO

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

is not emptied, it will overrun after 8 data words at the latest and the receive overrun interrupt SSPRORINTR is triggered, if enabled. (The receive overrun interrupt can be enabled and disabled with the RORIE bit in the 'SSPCR1' register.)

If data have been transferred from the SPI master to the SPI slave without the receive FIFO of the master being emptied, please **note**: Before a data transfer in the opposite direction, i.e., from the slave to the master, the data from the previous transfer must be cleared from the receive FIFO of the master before the slave can read the "new" data from the receive FIFO of the master.

No new data received will be entered in the receive FIFO if it is full, but data transfer will not automatically be cancelled.

2.3.14.9.4 Send support with pause between individual characters

In one use case, the SPI transmitters are to transmit blocks of 64 bytes, for example, in a GDMA-compatible process. A set pause (e.g., 200 µs) must however be maintained between the individual bytes. To ensure this pause, the two GDMA channels required (one channel for SPI-TxD and one channel for SPI-RxD) are controlled by two timers.

A transfer record must be entered in the GDMA job list for each byte transfer (total: 64 + 1 transfer entries, see below). Timer 4 (SPI1) and timer 5 (SPI2) are also configured for cyclic, SW-triggered operation with an 1-active timer output signal. A '0' run for a given timer switches the timer output to '1' for one clock cycle in line with the TX-FIFO status (see below) and sets the corresponding 'SPI1/2_SSPTX_Delayed_ Request' FF (located in the GDMA shell). The GDMA controller then runs a single-byte transfer (transfer record), generates GDMA acknowledge and resets the 'SPI1/2_SSPTX_Delayed_Request' FF (see chapter 2.3.7.2.1.2).

Flow control for generating the timer-triggered GDMA requests can only prevent an SPI TX-FIFO overrun if the SPI TX interrupt status (TX-FIFO half full status, >= 4 bytes) is also evaluated. A TX-FIFO overrun is possible if too short a timer cycle time is configured with a short SPI TX transfer rate. The TX interrupt must therefore be enabled during configuration of the SPI-IP (-> 'SSPCR1' register(Bit 1), TIE = 1). This ensures that no characters are lost; certain timer trigger pulses are, if necessary, temporarily ignored.

To ensure equidistance when the above 64-byte transfer is repeated, the timer must be stopped after the 64th single transfer. This function is executed by the GDMA controller as the 65th transfer entry (-> Write access to 'GATE_TRIG_CONTROL_REG' register -> tim_4_clk_en or tim_5_clk_en bit reset) once the 64th single transfer is complete.

Timers 4 and 5 are also used for other functions and the timer output signals 'SPI1/2_SSPTX_Delayed_Request' can therefore also set FFs. The FFs must be specifically reset by the SW. This is done using the 'CLEAR_SPI1/2_SSPTX_Delayed_Request' bits in the 'CLEAR_DMA_SPI_REQ_REG' SCRB register (see chapter 2.3.14.11.28.1). These register bits are not retentive; they simply activate a reset pulse when '1' is written. There is no reset when a '0' is written, so the 'SPI1/2_SSPTX_Delayed_ Request' FFs can be operated separately.

Sequence for above scenario:

1. Write GDMA job list to GDMA job list RAM (e.g., 64 single transfers plus timer stop register access)
2. Stop timer in progress and then reset 'SPI1/2_SSPTX_Delayed_Request' FFs with write access to 'CLEAR_DMA_SPI_REQ_REG' (SCRB)
3. Enter pause + byte transfer time as load value in timer 4 or timer 5 LOAD register
4. Configure timer mode register for SW-triggered, cyclic operation (internal gating)
5. Start GDMA job (in HW-triggered request mode)
6. Start timer with write access to GATE_TRIG_CONTROL_REG (sets tim_4(5)_clk_en and tim_4(5)_int_gate_trig bits).

Copyright © Siemens AG 2023. All rights reserved

Block transfer is repeated as follows:

5. Start GDMA job (in HW-triggered request mode)
6. Start timer with write access to GATE_TRIG_CONTROL_REG (sets tim_4(5)_clk_en and tim_4(5)_int_gate_trig bits)

2.3.14.9.5 SPI Flash boot

An SPI can also be used to boot I-TCM and D-TCM (256 KByte in total). This boot process can involve acyclic communication data or application code for rapid execution (e.g., for the digital IO of the ECO PN). To meet fast start-up requirements, booting must be completed within ca. 100-150 msec. High-speed SPI flash is therefore required.

Figure 52 and Figure 53 show the timing for a 7-MHz SPI flash (in this case Winbond W25X10/20/40). Table 36 lists the critical timing parameters.

At a frequency of 25 MHz, booting 256 KByte takes ca. 85 ms.

Figure 52: SPI flash: serial output timing

Figure 53: SPI flash: serial input timing

Table 36: Timing parameters for SPI flash (75 MHz)

Description	Symbol	Specification	Unit
Clock High / Low Time for Fast Read	t_{CLH}, t_{CLH}	6	ns
Clock Low to Output Valid	t_{CLQV}	6	ns
Output Hold Time	t_{CLQX}	0	ns
Data In Setup Time	t_{DVCH}	2	ns
Data In Hold Time	t_{CHDX}	5	ns

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.9.6 Address mapping

Start-Address	End-Address	Modul/Memory-Name
0h	17h	SPI

Mo-dule	Register/Memory	Read	Write	Access	Address
/SPI				32-Bit	
	/SSPCR0	r	w	32-Bit	0h
	/SSPCR1	r	w	32-Bit	4h
	/SSPDR	rh	w	32-Bit	8h
	/SSPSR	r		32-Bit	Ch
	/SSPCPSR	r	w	32-Bit	10h
	/SSPIIR_SSPICR	rh		32-Bit	14h

2.3.14.9.7 Register description

Only the registers with addresses 0 – 14h are required for operating the SSPMS IP. The remaining registers (addresses 20h – B0h) are created by mirroring (address bit 5 of the APB bus is not connected to the SSPMS IP) or are only implemented for verification purposes.

Note

Read values for unassigned register bits

In the description of the SSPMS IP, the read value of the unassigned register bits is given as "unpredictable". A read value of '0' is, however, implemented.

A description of the registers can be found in the SSPMS IP documentation.

Module: SPI

Register:	SSPCR0			Address:	0h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	SSPCR0 is control register 0 and contains five bit fields that control various functions within the PrimeCell SSPMS.					

Bit	Identifier	Reset	Attr.		Function / Description
3dt0	DSS	0h	r	w	Data size select: 0000 Reserved, undefined operation 0001 Reserved, undefined operation 0010 Reserved, undefined operation 0011 4-bit data 0100 5-bit data 0101 6-bit data 0110 7-bit data 0111 8-bit data 1000 9-bit data 1001 10-bit data 1010 11-bit data 1011 12-bit data 1100 13-bit data 1101 14-bit data 1110 15-bit data 1111 16-bit data

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

5dt4	FRF	0h	r	w	Frame format: 00 Motorola SPI frame format 01 TI synchronous serial frame format 10 National Mikrowire frame format 11 Reserved, undefined operation
6	SPO	0h	r	w	SCLKOUT polarity (applicable to Motorola SPI frame format only).
7	SPH	0h	r	w	SCLKOUT phase (applicable to Motorola SPI frame format only).
15dt8	SCR	0h	r	w	Serial Clock rate. The value SCR s used to generate the transmit and receive bit range of the PrimeCell SSPMS.
31dt16	reserved	0000h			

Register:	SSPCR1			Address:	4h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	SSPCR1 is the control register 1 and contains five different bit fields, which control various functions within the PrimeCell SSPMS.					

Bit	Identifier	Reset	Attr.		Function / Description
0	RIE	0h	r	w	Receive FIFO interrupt enable: 0=Receive FIFO half-full or more condition does not generate the SSPRXINTR interrupt. 1=Receive FIFO half-full or more condition generates the SSPRXINTR interrupt.
1	TIE	0h	r	w	Transmit FIFO interrupt enable: 0=Transmit FIFO half-full or less condition does not generate the SSPTXINTR interrupt. 1=Transmit FIFO half-full or less condition generates the SSPTXINTR interrupt.
2	RORIE	0h	r	w	Receive FIFO overrun interrupt enable: 0=Overrun detection is disabled. Overrun condition does not generate the SSPRORINTR interrupt. Clearing rhis bit to zero also clears the SSPRORINTR interrupt if it is already asserted. 1=Overrun detection is enabled. Overrun condition generates the SSPRORINTR interrupt.
3	LBM	0h	r	w	Loop back mode 0=Normal serial port operation enabled 1=Output of transmit serial shifter is connected to input of receive serial shifter internally.
4	SSE	0h	r	w	Synchronous serial port enable: 0=SSP operation disabled 1= operation enabled

5	MS	0h	r	w	Master/slave mode select. This bit can be modified only when the PrimeCell SSPMS is disabled (SSE=0). 0=Device configured as master (default). 1=Device configured as slave.
6	SOD	0h	r	w	Slave mode output disable.This bis is relevant only in the slave mode (MS=1). In multiple-slave systems, it is possible for an SSPMS master to broadcast a message to all slaves in the system while ensuring that only the slave drives data onto ist serial output line. In such systems the RXD lines from multiple slaves could be tied together. To operate in such systems, the SOD may be set if the SSP slave is not supposed to drive the SSPTXD line. 0=SSP may drive the SSPTXD output in slave mode 1=SSP must not drive the SSPTXD output in slave mode.
31dt7	reserved	0000000h			

Register:	SSPDR				Address:	8h	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	w
Description:	Transmit/Receive FIFO						
Bit	Identifier	Reset	Attr.		Function / Description		
15dt0	SSPDR	0h	rh	w	Transmit/Receive FIFO		
31dt16	reserved	0000h					

Register:	SSPSR				Address:	Ch	
Bits:	31dt0	Reset value:	3h		Attributes:	r	
Description:	SSPSR is a read-only status register which contains bits that indicate the FIFO fill status and the SSP busy status.						
Bit	Identifier	Reset	Attr.		Function / Description		
0	TFE	1h	r		Transmit FIFO empty		
1	TNF	1h	r		Transmit FIFO not full		
2	RNE	0h	r		Receive FIFO not empty		
3	RFF	0h	r		Receive FIFO full		
4	BSY	0h	r		SSP busy flag		
31dt5	reserved	0000000h					

Register:	SSPCPSR				Address:	10h	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Clock prescale divisor						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	SSPCPSR	0h	r	w	Clock prescale divisor		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

31dt8	reserved	000000h			

Register:	SSPIIR_SSPICR			Address:	14h
Bits:	31dt0	Reset value:	0h	Attributes:	rh
Description:	The interrupt status is read from the SSP interrupt identification register (SSPIIR)				

Bit	Identifier	Reset	Attr.		Function / Description
0	RIS	0h	rh		Read: SSP transmit FIFO service request interrupt status
1	TIS	0h	rh		Read: SSP Receive FIFO service request interrupt status
2	RORIS	0h	rh		Read: SSP Receive FIFO overrun interrupt status
31dt3	reserved	00000000h			

2.3.14.10 GPIO (General Purpose Input/Output)

The General Purpose Input/Output (GPIO) module connects modules in the ERTEC 200P (Timer, SPI1/2, UART1/2, I²C, Octal-SPI, ...) through multiplexer structures ("Alternate Functions") to the GPIO pins. In addition, it offers the possibility to set GPIO pins to values or to read the values of GPIO pins through register access over the APB bus. The General Purpose Inputs/Outputs (referred to as GPIO below) provide the possibility to route different application-specific functions that are not required simultaneously to the ASIC pins. Through this so-called pin-sharing process it is possible to cover a multitude of functions with only one ASIC and still use an ASIC package that has a low number of pins.

There is a maximum of 96 General Purpose Input/Outputs (GPIOs). These are divided into 2 blocks:

- GPIO31-0: 32-bit (available as default GPIO port)
- GPIO95-32: 64-bit (available as an alternative to XHIF)

GPIO31-0 is available as a default block and is multiplexed with the interface signals from the PNPLL, UART2/3, SPI1/2, I2C_1_3, Timer 0 – 5 and ARM926 watchdog. Following a reset, this GPIO block is selected and connected to all inputs. The alternate functions are set by the SW using the GPIO registers (see chapter 2.3.14.11.11).

This function is set upon an XRESET PowerOn reset with the configuration pins CONFIG(6:3) or by the SW in the SCRB register 'CONFIG_REG(6:3)' (see 2.3.14.11.3).

The GPIO15-0 inputs are also connected to an I-Filter (see 2.3.14.2) and to the ARM interrupt controller (IRQ32-47, see 2.3.5.9). The polarity of these signals to the ARM interrupt controller can be set in the 'GPIO_INT_POLSEL' register (see 2.3.14.11.27.1). In line with the polarity of the external signals, this ensures a high-active interrupt level.

The GPIO7-4 inputs are also connected to PN-ICU 1 over an I-Filter (see 2.3.14.2). The polarity of these signals to the PN-ICU 1 can be set in the 'GPIO_INT_POLSEL' register (see 2.3.14.11.27.1). In line with the polarity of the external signals, this ensures a high-active interrupt level.

The GPIO3-0 inputs are connected to an I-Filter (see 2.3.14.2) and can control HW jobs with GDMA (see 2.3.7.2.1.1). The polarity of these signals to the GDMA can be set in the 'GPIO_INT_POLSEL' register (see 2.3.14.11.27.1). A high-active level to the GDMA must always be set. As these input signals are not synchronized with the GDMA, "Synchronize" mode (mode 2) must be set selectively in the I-Filter for these inputs.

GPIO31-0 have integrated pulls that can be selected by the SW. These pulls have a default assignment following a reset (see corresponding Pull31_0GPIO register in the SCRB, 2.3.14.11.27.1). In GPIO11-0, the pulls are disabled by default as the alternate functions mean that there is no suitable setting for the application.

GPIO95-32 is completely unavailable if the parallel 32-bit host interface (XHIF) is used. If XHIF is operated in 16-bit mode, there are 18 GPIOs remaining. GPIO95-32 is multiplexed with the interface signals from the local IO (64-bit parallel, 2 × serial SPI master), SPI1, UART1/4, ARM926 watchdog, I2C_1_2 and PNPLL. These alternate functions are set by the SW using the GPIO registers (see 2.3.14.10.6).

The ARM926 trace port is also multiplexed to the GPIO95-32. These functions and the corresponding host interface enable are set upon an XRESET PowerOn reset with the configuration pins CONFIG(6:3) (see 2.3.14.11.3) or by the SW in the SCRB register 'CONFIG_REG' (see 2.3.14.11.27.1).

Note

Rules for the SW

The default values of the alternate functions **must not** be changed if the XHIF interface or the ARM926 trace port is selected at the corresponding pins with the configuration pins.

Copyright © Siemens AG 2023. All rights reserved

GPIO95-32 have integrated pulls that can be selected by the SW (see 2.3.14.11.15). These pulls have a default assignment following a reset (see corresponding Pull95_32GPIO registers in the SCRB, 2.3.14.11.27.1). GPIO55-54 (XHIF_IRQ, XHIF_XRDY) have **no** internal pulls.

2.3.14.10.1 Features of the GPIO module

- APB interface (AMBA 2.0): only 32-bit access is supported (**no** check is carried out to determine whether addresses are accessed that do not have registers.)
- No. of GPIO pins: 96
- Each GPIO pin can have either:
  - one of 3 signals from internal modules connected (GPIO_PORT_MODE_L/H_x) or
  - a GPIO register value

GPIO_PORT_MODE_L/H_x		Function
0	0	GPIO function
0	1	Alternate function A
1	0	Alternate function B
1	1	Alternate function C

- With the GPIO function the signal direction (input / output) can be set individually for each GPIO pin over the register (GPIO_IOCTRL_x). The GPIO port pins can therefore be programmed bit-by-bit as an input or output and can be read through the APB.
- With the GPIO function the following actions can be carried out for the output values of the GPIO pins for each pin individually through register access:
  - set (GPIO_OUT_SET_x register),
  - reset (GPIO_OUT_CLEAR_x register) or
  - written with 0 / 1 (GPIO_OUT_x register).
- The values at the GPIO pins can be read through the GPIO_IN_x register.
- After a reset all GPIOs are switched to input.
- The selective reset for the register bits of individual GPIO pins is not supported due to the integration of the GPIO_SM in the ERTEC 200P.
- If one of the alternate functions A/B/C is selected, the inputs GPIO_IN_x are connected through directly to the selected outputs DIN_AltFuncA, DIN_AltFuncB or DIN_AltFuncC. The two alternate functions that have not been selected connect the blocking values defined for this GPIO pin (inputs BLOCKVAL_DIN_AltFuncA, BLOCKVAL_DIN_AltFuncB and BLOCKVAL_DIN_AltFuncC) to the outputs DIN_AltFuncA, DIN_AltFuncB and DIN_AltFuncC.
  - The blocking values are wired fixed to low or high level and cannot be configured by the software.
  - Unused applications can be kept in a defined idle state with blocking values. Blocking values make sense, for example, when an (unused) function triggers interrupts.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.10.2 GPIO module function description

Block diagram of a GPIO module

Figure 54: Block diagram of a GPIO module

3 GPIO modules with 32 GPIOs each are used for the GPIOs. These GPIO modules have the following properties:

- The number of general purpose input/outputs (GPIOs) is configurable with the hardware.
- Each GPIO can be programmable as input/output (GPIO function).
- Each GPIO input (pin) can be read by the SW.

- Each GPIO output can be set and reset by the SW (bit-selective programming is possible).
- The GPIO function is multiplexed with up to 3 alternate functions (function A-C).
- 2 reset signals affect the GPIO registers (XRESET_HW as asynchronous reset and XRESET_GPIO_SM as selective reset).
- Most GPIO pins have integrated pull-ups/pull-downs to prevent floating.
- The GPIO pins can be connected over an integrated filter (see Figure 54) (required for local IOs, external interrupts, alarms, etc.)
- Up to 16 GPIOs (GPIO15-0) can trigger interrupts at the ARM processor and are connected to the ARM interrupt controller for this purpose (IRQ32-47) (see 2.3.5.9). The polarity of these signals to the ARM-ICU (ARM interrupt controller) can be set in the 'GPIO_INT_POLSEL' register (see 2.3.14.11.27.1). In line with the polarity of the external signal, this ensures a high-active interrupt level. The signals must be at least 2 clocks (125 MHz) long.
- Up to 6 GPIOs can be used as the gate / trigger signal for the internal timers.
- Up to 4 GPIOs (GPIO3-0) can control HW jobs with GDMA (see 2.3.7.2.1.1). The polarity of these signals to the GDMA can be set in the 'GPIO_INT_POLSEL' register (see 2.3.14.11.27.1). A high-active level to the GDMA must always be set.

Using the GPIO module, either one of three signals or the value of the corresponding bit in the GPIO_OUT_x register can be connected to each GPIO pin ("*GPIO function*"). The GPIO_PORT_MODE_L/H_x registers define what is connected to the GPIO pins.

If a signal is connected to a GPIO pin, a corresponding output enable signal specifies whether the pin is the input or output. If, on the other hand, the GPIO function is used for a pin, this is specified with the GPIO_IOCTRL_x register.

With the GPIO function, the output value of a GPIO pin can be defined in three different ways:

- By writing the corresponding bit in the GPIO_OUT_x register
- By writing a '1' to the corresponding bit in the GPIO_OUT_SET_x register
  (=> The pin is set to 1.)
- By writing a '1' to the corresponding bit in the GPIO_OUT_CLEAR_x register
  (=> The pin is set to 0.)

The values can be read at the GPIO pins over the GPIO_IN_x register. If the input values of several GPIO pins are routed to the same internal input signal, the inputs are AND-operated or OR-operated, depending on the polarity before the internal input signal.

2.3.14.10.3 Integration of the GPIO module

The GPIO module connects the top-level modules to the GPIO pins. It is also connected to the APB bus. The GPIO module is operated with the APB clock.

The feature of an internal GPIO "reset disable" is activated in ERTEC 200P (module input: XRESET_HW connected to reset signal). Alongside the global reset (XRESET_GPIO_SM), a selective reset in GPIO_SM is also possible.

The connection of the internal modules to the GPIO module (GPIO_XOE_F_A/B/C, GPIO_OUT_F_A/B/C, GPIO_IN_F_A/B/C, GPIO_IN_F_A/B/C_BLOCK signals) is described in the corresponding modules and in chapter 2.3.14.10.5.

Copyright © Siemens AG 2023. All rights reserved

Figure 55: GPIO, IO circuit

2.3.14.10.4 GPIO pad multiplexing

Alongside the alternate functions (A-C) integrated in the GPIO modules, the GPIO95-32 block has a number of other alternate functions that are set with the configuration pins CONFIG(6:3) (see 2.3.14.11.3) or the SCRB register 'CONFIG_REG' (see 2.3.14.11.27.1, host interface, ARM926 trace port).

Alongside the alternate functions (A-C) integrated in the GPIO module, the GPIO31-0 block has another alternate function that is set with the configuration pins CONFIG(6:3) (see 2.3.14.11.3) or the SCRB register 'CONFIG_REG' (see 2.3.14.11.27.1).

2.3.14.10.5 GPIO assignment

2.3.14.10.5.1 IO multiplexing

For all pins designated GPIO.., the software can connect the alternate functions (A-C) downstream (see Table 37). All other pins have fixed coding set by the function groups.
TRACEPKT outputs of the trace port at GPIO53-38 (when CONFIG(6:3) = 1110b) that are not used by the debugger remain high-impedance.

2.3.14.10.5.2 GPIO Alternate Function

Table 37 lists alternate functions A-C that can be set for the GPIO pins by the SW. Some alternate function inputs are driven by two different GPIO pads (inputs with "_a/_b" versions). The GPIO alternate functions set which signal is assigned to each GPIO pad. In the case of input signals with

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

_a/_b versions, the user must however ensure that **only one of the two versions** is activated over the GPIO alternate functions.

Table 37: Overview of alternate functions (A-C)

GPIO Pin	GPIO Registers	GPIO Alternate Function A	GPIO Alternate Function B	GPIO Alternate Function C
GPIO0_INT	GPIO_*_0.0	PNPLL_OUT0_a	TIM_OUT0_a	NOT_USED
GPIO1_INT	GPIO_*_0.1	PNPLL_OUT1_a	TIM_OUT1_a	NOT_USED
GPIO2_INT	GPIO_*_0.2	PNPLL_OUT2_a	TIM_OUT2	DBGRQ
GPIO3_INT	GPIO_*_0.3	PNPLL_OUT3_a	TIM_OUT3	DBGACK
GPIO4_INT	GPIO_*_0.4	PNPLL_OUT4_a	TIM_OUT4	I2C_SDOI_1_a
GPIO5_INT	GPIO_*_0.5	PNPLL_OUT5_a	TIM_OUT5	I2C_SCLK_1_a
GPIO6_INT	GPIO_*_0.6	PNPLL_OUT6_a	TIM_TRIG0_a	I2C_SDOI_2_a
GPIO7_INT	GPIO_*_0.7	PNPLL_OUT7_a	TIM_TRIG1_a	I2C_SCLK_2_a
GPIO8_INT	GPIO_*_0.8	PNPLL_OUT8_a	TIM_TRIG2	SPI_2_SCLKIN_b
GPIO9_INT	GPIO_*_0.9	PNCLKA_IN_a	TIM_TRIG3	SPI_2_SFRMIN_b
GPIO10_INT	GPIO_*_0.10	NOT_USED	TIM_TRIG4	SPI_2_SSPOE_b
GPIO11_INT	GPIO_*_0.11	PNTIME_OUT_a	TIM_TRIG5	SPI_2_SSPCTLOE_b
GPIO12_INT	GPIO_*_0.12	PNTIME_IN	U2_CTS	SPI_2_SCLKOUT_b
GPIO13_INT	GPIO_*_0.13	WD_XWDOUT0_a	U2_RTS	SPI_2_SFRMOUT_b
GPIO14_INT	GPIO_*_0.14	I2C_SCLK_3	U2_TXD	SPI_2_SSPTXD_b
GPIO15_INT	GPIO_*_0.15	I2C_SDOI_3	U2_RXD	SPI_2_SSPRXD_b
GPIO16	GPIO_*_0.16	SPI_1_SCLKOUT_a	NOT_USED	NOT_USED
GPIO17	GPIO_*_0.17	SPI_1_SFRMOUT_a	NOT_USED	NOT_USED
GPIO18	GPIO_*_0.18	SPI_1_SSPTXD_a	NOT_USED	NOT_USED
GPIO19	GPIO_*_0.19	SPI_1_SSPRXD_a	NOT_USED	NOT_USED
GPIO20	GPIO_*_0.20	SPI_1_SSPOE_a	NOT_USED	NOT_USED
GPIO21	GPIO_*_0.21	SPI_1_SSPCTLOE_a	NOT_USED	NOT_USED
GPIO22	GPIO_*_0.22	SPI_1_SFRMIN_a	NOT_USED	NOT_USED
GPIO23	GPIO_*_0.23	SPI_1_SCLKIN_a	PNCLKA_IN_b	NOT_USED

Copyright © Siemens AG 2023. All rights reserved

GPIO Pin	GPIO Registers	GPIO Alternate Function A	GPIO Alternate Function B	GPIO Alternate Function C
GPIO24	GPIO_*_0.24	SPI_2_SCLKOUT_a	TIM_OUT0_b	NOT_USED
GPIO25	GPIO_*_0.25	SPI_2_SFRMOUT_a	TIM_OUT1_b	NOT_USED
GPIO26	GPIO_*_0.26	SPI_2_SSPTXD_a	TIM_TRIG0_b	NOT_USED
GPIO27	GPIO_*_0.27	SPI_2_SSPRXD_a	TIM_TRIG1_b	NOT_USED
GPIO28	GPIO_*_0.28	SPI_2_SCLKIN_a	U3_TXD	PNTIME_OUT_b
GPIO29	GPIO_*_0.29	SPI_2_SFRMIN_a	U3_RXD	PNPLL_OUT6_b
GPIO30	GPIO_*_0.30	SPI_2_SSPOE_a	NOT_USED	PNPLL_OUT7_b
GPIO31	GPIO_*_0.31	SPI_2_SSPCTLOE_a	NOT_USED	PNPLL_OUT8_b

GPIO Pin	GPIO Registers	GPIO Alternate Function A	GPIO Alternate Function B	GPIO Alternate Function C
XHIF_A1	GPIO_*_1.0	NOT_USED	NOT_USED	NOT_USED
XHIF_A2	GPIO_*_1.1	NOT_USED	NOT_USED	NOT_USED
XHIF_A3	GPIO_*_1.2	NOT_USED	NOT_USED	NOT_USED
XHIF_A4	GPIO_*_1.3	NOT_USED	NOT_USED	NOT_USED
XHIF_A5	GPIO_*_1.4	NOT_USED	NOT_USED	NOT_USED
XHIF_A6	GPIO_*_1.5	NOT_USED	NOT_USED	NOT_USED
XHIF_A7	GPIO_*_1.6	NOT_USED	NOT_USED	NOT_USED
XHIF_A8	GPIO_*_1.7	NOT_USED	NOT_USED	NOT_USED
XHIF_A9	GPIO_*_1.8	NOT_USED	NOT_USED	SPI_1_SCLKOUT_b
XHIF_A10	GPIO_*_1.9	NOT_USED	NOT_USED	SPI_1_SFRMOUT_b
XHIF_A11	GPIO_*_1.10	NOT_USED	NOT_USED	SPI_1_SSPTXD_b
XHIF_A12	GPIO_*_1.11	NOT_USED	NOT_USED	SPI_1_SSPRXD_b
XHIF_A13	GPIO_*_1.12	NOT_USED	NOT_USED	SPI_1_SSPOE_b
XHIF_A14	GPIO_*_1.13	NOT_USED	NOT_USED	SPI_1_SSPCTLOE_b
XHIF_A15	GPIO_*_1.14	NOT_USED	NOT_USED	SPI_1_SCLKIN_b
XHIF_A16	GPIO_*_1.15	NOT_USED	NOT_USED	SPI_1_SFRMIN_b

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

GPIO Pin	GPIO Registers	GPIO Alternate Function A	GPIO Alternate Function B	GPIO Alternate Function C
XHIF_A17	GPIO_*_1.16	NOT_USED	NOT_USED	U1_CTS
XHIF_A18	GPIO_*_1.17	NOT_USED	NOT_USED	U1_DCD
XHIF_A19	GPIO_*_1.18	NOT_USED	NOT_USED	U1_DSR
XHIF_SEG_2	GPIO_*_1.19	NOT_USED	NOT_USED	U1_RI
XHIF_SEG_0	GPIO_*_1.20	NOT_USED	NOT_USED	U1_RTS
XHIF_SEG_1	GPIO_*_1.21	NOT_USED	NOT_USED	U1_DTR
XHIF_XRDY	GPIO_*_1.22	NOT_USED	NOT_USED	U4_TXD
XHIF_XRQ	GPIO_*_1.23	NOT_USED	NOT_USED	U4_RXD
XHIF_XWR	GPIO_*_1.24	NOT_USED	NOT_USED	PNPLL_OUT0_b
XHIF_XRD	GPIO_*_1.25	NOT_USED	NOT_USED	PNPLL_OUT1_b
XHIF_XCS_R_A20	GPIO_*_1.26	NOT_USED	NOT_USED	PNPLL_OUT2_b
XHIF_XCS_M	GPIO_*_1.27	NOT_USED	NOT_USED	PNPLL_OUT3_b
XHIF_XBE0	GPIO_*_1.28	NOT_USED	NOT_USED	PNPLL_OUT4_b
XHIF_XBE1	GPIO_*_1.29	NOT_USED	NOT_USED	PNPLL_OUT5_b
XHIF_XBE2	GPIO_*_1.30	NOT_USED	NOT_USED	U1_TXD
XHIF_XBE3	GPIO_*_1.31	NOT_USED	NOT_USED	U1_RXD
XHIF_D0	GPIO_*_2.0	NOT_USED	NOT_USED	NOT_USED
XHIF_D1	GPIO_*_2.1	NOT_USED	NOT_USED	NOT_USED
XHIF_D2	GPIO_*_2.2	NOT_USED	NOT_USED	NOT_USED
XHIF_D3	GPIO_*_2.3	NOT_USED	NOT_USED	NOT_USED
XHIF_D4	GPIO_*_2.4	NOT_USED	NOT_USED	NOT_USED
XHIF_D5	GPIO_*_2.5	NOT_USED	NOT_USED	NOT_USED
XHIF_D6	GPIO_*_2.6	NOT_USED	NOT_USED	NOT_USED
XHIF_D7	GPIO_*_2.7	NOT_USED	NOT_USED	XSSI_DATA
XHIF_D8	GPIO_*_2.8	NOT_USED	NOT_USED	NOT_USED
XHIF_D9	GPIO_*_2.9	NOT_USED	NOT_USED	NOT_USED

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

GPIO Pin	GPIO Registers	GPIO Alternate Function A	GPIO Alternate Function B	GPIO Alternate Function C
XHIF_D10	GPIO_*_2.10	NOT_USED	NOT_USED	NOT_USED
XHIF_D11	GPIO_*_2.11	NOT_USED	NOT_USED	WD_XWDOUT0_b
XHIF_D12	GPIO_*_2.12	NOT_USED	NOT_USED	NOT_USED
XHIF_D13	GPIO_*_2.13	NOT_USED	NOT_USED	NOT_USED
XHIF_D14	GPIO_*_2.14	NOT_USED	NOT_USED	NOT_USED
XHIF_D15	GPIO_*_2.15	NOT_USED	NOT_USED	NOT_USED
XHIF_D16	GPIO_*_2.16	NOT_USED	NOT_USED	I2C_SDOI_1_b
XHIF_D17	GPIO_*_2.17	NOT_USED	NOT_USED	I2C_SCLK_1_b
XHIF_D18	GPIO_*_2.18	NOT_USED	NOT_USED	I2C_SDOI_2_b
XHIF_D19	GPIO_*_2.19	NOT_USED	NOT_USED	I2C_SCLK_2_b
XHIF_D20	GPIO_*_2.20	NOT_USED	NOT_USED	NOT_USED
XHIF_D21	GPIO_*_2.21	NOT_USED	NOT_USED	NOT_USED
XHIF_D22	GPIO_*_2.22	NOT_USED	NOT_USED	NOT_USED
XHIF_D23	GPIO_*_2.23	NOT_USED	NOT_USED	NOT_USED
XHIF_D24	GPIO_*_2.24	NOT_USED	NOT_USED	NOT_USED
XHIF_D25	GPIO_*_2.25	NOT_USED	NOT_USED	NOT_USED
XHIF_D26	GPIO_*_2.26	NOT_USED	NOT_USED	NOT_USED
XHIF_D27	GPIO_*_2.27	NOT_USED	NOT_USED	NOT_USED
XHIF_D28	GPIO_*_2.28	NOT_USED	NOT_USED	NOT_USED
XHIF_D29	GPIO_*_2.29	NOT_USED	NOT_USED	NOT_USED
XHIF_D30	GPIO_*_2.30	NOT_USED	NOT_USED	NOT_USED
XHIF_D31	GPIO_*_2.31	NOT_USED	NOT_USED	NOT_USED

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.10.6 Address mapping

Start-Address	End-Address	Modul/Memory-Name
0h	5Fh	gpio

Module	Register/Memory	Read	Write	Access	Address
/gpio				32-Bit	
	/GPIO_IOCTRL_0	r	w	32-Bit	0h
	/GPIO_OUT_0	rh	w	32-Bit	4h
	/GPIO_OUT_SET_0		wt	32-Bit	8h
	/GPIO_OUT_CLEAR_0		wt	32-Bit	Ch
	/GPIO_RES_DIS_0	r	w	32-Bit	10h
	/GPIO_IN_0	rh		32-Bit	14h
	/GPIO_PORT_MODE_0_L	r	w	32-Bit	18h
	/GPIO_PORT_MODE_0_H	r	w	32-Bit	1Ch
	/GPIO_IOCTRL_1	r	w	32-Bit	20h
	/GPIO_OUT_1	rh	w	32-Bit	24h
	/GPIO_OUT_SET_1		wt	32-Bit	28h
	/GPIO_OUT_CLEAR_1		wt	32-Bit	2Ch
	/GPIO_RES_DIS_1	r	w	32-Bit	30h
	/GPIO_IN_1	rh		32-Bit	34h
	/GPIO_PORT_MODE_1_L	r	w	32-Bit	38h
	/GPIO_PORT_MODE_1_H	r	w	32-Bit	3Ch
	/GPIO_IOCTRL_2	r	w	32-Bit	40h
	/GPIO_OUT_2	rh	w	32-Bit	44h
	/GPIO_OUT_SET_2		wt	32-Bit	48h
	/GPIO_OUT_CLEAR_2		wt	32-Bit	4Ch
	/GPIO_RES_DIS_2	r	w	32-Bit	50h
	/GPIO_IN_2	rh		32-Bit	54h
	/GPIO_PORT_MODE_2_L	r	w	32-Bit	58h
	/GPIO_PORT_MODE_2_H	r	w	32-Bit	5Ch

Copyright © Siemens AG 2023. All rights reserved

2.3.14.10.7 Register description

Module: gpio

Register:	GPIO_IOCTRL_0				Address:	0h	
Bits:	31dt0	Reset value:	FFFFFFFFh		Attributes:	r	w
Description:	Configuration register for General Purpose IOs (31:0) 0: GPIOx = output, 1: GPIOx = input						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	GPIO_IOCTRL_0	FFFFFFFFh	r	w	Configuration register for General Purpose IOs (31:0) 0: GPIOx = output, 1: GPIOx = input		

Register:	GPIO_OUT_0				Address:	4h	
Bits:	31dt0	Reset value:	0h		Attributes:	rh	w
Description:	Output register for General Purpose IOs (31:0) 0: GPIO outputx = 0, 1: GPIO outputx = 1						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	GPIO_OUT_0	0h	rh	w	Output register for General Purpose IOs (31:0) 0: GPIO outputx = 0, 1: GPIO outputx = 1		

Register:	GPIO_OUT_SET_0				Address:	8h	
Bits:	31dt0	Reset value:	0h		Attributes:		wt
Description:	Bit-selective setting of the output register for General Purpose IOs (31:0) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 1 Read always returns 0						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	GPIO_OUT_SET_0	0h		wt	Bit-selective setting of the output register for General Purpose IOs (31:0) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 1 Read always returns 0		

Register:	GPIO_OUT_CLEAR_0				Address:	Ch	
Bits:	31dt0	Reset value:	0h		Attributes:		wt
Description:	Bit-selective reset of the output register for General Purpose IOs (31:0) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 0 Read always returns 0						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	GPIO_OUT_CLEAR_0	0h		wt	Bit-selective reset of the output register for General Purpose IOs (31:0) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 0		

Copyright © Siemens AG 2023. All rights reserved

					Read always returns 0

Register:	GPIO_RES_DIS_0			Address:	10h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Bit-selective reset disabling of XRESET_GPIO_SM signal for registers of General Purpose IOs (31:0) For writing: 0: XRESET_GPIO_SM resets the corresponding register bit of all registers except of GPIO_RES_DIS 1: XRESET_GPIO_SM has no effect on the corresponding register bit					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	GPIO_RES_DIS_0	0h	r	w	Bit-selective reset disabling of XRESET_GPIO_SM signal for registers of General Purpose IOs (31:0) For writing: 0: XRESET_GPIO_SM resets the corresponding register bit of all registers except of GPIO_RES_DIS 1: XRESET_GPIO_SM has no effect on the corresponding register bit

Register:	GPIO_IN_0			Address:	14h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:	Input register for General Purpose IOs (31:0)					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	GPIO_IN_0	00000000h	rh		Input register for General Purpose IOs (31:0)

Register:	GPIO_PORT_MODE_0_L			Address:	18h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Configuration register for GPIO port (15:0) Function assignment: 00=function0 01=function1 10=function2 11=function3					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	GPIO_0_MODE_0_L	0h	r	w	Port GPIO(0)
3dt2	GPIO_1_MODE_0_L	0h	r	w	Port GPIO(1)
5dt4	GPIO_2_MODE_0_L	0h	r	w	Port GPIO(2)
7dt6	GPIO_3_MODE_0_L	0h	r	w	Port GPIO(3)
9dt8	GPIO_4_MODE_0_L	0h	r	w	Port GPIO(4)
11dt10	GPIO_5_MODE_0_L	0h	r	w	Port GPIO(5)
13dt12	GPIO_6_MODE_0_L	0h	r	w	Port GPIO(6)
15dt14	GPIO_7_MODE_0_L	0h	r	w	Port GPIO(7)
17dt16	GPIO_8_MODE_0_L	0h	r	w	Port GPIO(8)
19dt18	GPIO_9_MODE_0_L	0h	r	w	Port GPIO(9)
21dt20	GPIO_10_MODE_0_L	0h	r	w	Port GPIO(10)

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

23dt22	GPIO_11_MODE_0_L	0h	r	w	Port GPIO(11)
25dt24	GPIO_12_MODE_0_L	0h	r	w	Port GPIO(12)
27dt26	GPIO_13_MODE_0_L	0h	r	w	Port GPIO(13)
29dt28	GPIO_14_MODE_0_L	0h	r	w	Port GPIO(14)
31dt30	GPIO_15_MODE_0_L	0h	r	w	Port GPIO(15)

Register:	GPIO_PORT_MODE_0_H			Address:	1Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Configuration register for GPIO port (31:16) Function assignment: 00=function0 01=function1 10=function2 11=function3					
Bit	Identifier	Reset	Attr.		Function / Description	
1dt0	GPIO_16_MODE_0_H	0h	r	w	Port GPIO(16)	
3dt2	GPIO_17_MODE_0_H	0h	r	w	Port GPIO(17)	
5dt4	GPIO_18_MODE_0_H	0h	r	w	Port GPIO(18)	
7dt6	GPIO_19_MODE_0_H	0h	r	w	Port GPIO(19)	
9dt8	GPIO_20_MODE_0_H	0h	r	w	Port GPIO(20)	
11dt10	GPIO_21_MODE_0_H	0h	r	w	Port GPIO(21)	
13dt12	GPIO_22_MODE_0_H	0h	r	w	Port GPIO(22)	
15dt14	GPIO_23_MODE_0_H	0h	r	w	Port GPIO(23)	
17dt16	GPIO_24_MODE_0_H	0h	r	w	Port GPIO(24)	
19dt18	GPIO_25_MODE_0_H	0h	r	w	Port GPIO(25)	
21dt20	GPIO_26_MODE_0_H	0h	r	w	Port GPIO(26)	
23dt22	GPIO_27_MODE_0_H	0h	r	w	Port GPIO(27)	
25dt24	GPIO_28_MODE_0_H	0h	r	w	Port GPIO(28)	
27dt26	GPIO_29_MODE_0_H	0h	r	w	Port GPIO(29)	
29dt28	GPIO_30_MODE_0_H	0h	r	w	Port GPIO(30)	
31dt30	GPIO_31_MODE_0_H	0h	r	w	Port GPIO(31)	

Register:	GPIO_IOCTRL_1			Address:	20h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:	Configuration register for General Purpose IOs (63:32) 0: GPIOx = output, 1: GPIOx = input					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	GPIO_IOCTRL_1	FFFFFFFFh	r	w	Configuration register for General Purpose IOs (63:32) 0: GPIOx = output, 1: GPIOx = input	

Register:	GPIO_OUT_1			Address:	24h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Output register for General Purpose IOs (63:32) 0: GPIO outputx = 0, 1: GPIO outputx = 1					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	GPIO_OUT_1	0h	rh	w	Output register for General Purpose IOs (63:32) 0: GPIO outputx = 0, 1: GPIO outputx = 1

Register:	GPIO_OUT_SET_1			Address:	28h	
Bits:	31dt0	Reset value:	0h	Attributes:		wt
Description:	Bit-selective setting of the output register for General Purpose IOs (63:32) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 1 Read always returns 0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	GPIO_OUT_SET_1	0h		wt	Bit-selective setting of the output register for General Purpose IOs (63:32) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 1 Read always returns 0

Register:	GPIO_OUT_CLEAR_1			Address:	2Ch	
Bits:	31dt0	Reset value:	0h	Attributes:		wt
Description:	Bit-selective reset of the output register for General Purpose IOs (63:32) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 0 Read always returns 0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	GPIO_OUT_CLEAR_1	0h		wt	Bit-selective reset of the output register for General Purpose IOs (63:32) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 0 Read always returns 0

Register:	GPIO_RES_DIS_1			Address:	30h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Bit-selective reset disabling of XRESET_GPIO_SM signal for registers of General Purpose IOs (31:0) For writing: 0: XRESET_GPIO_SM resets the corresponding register bit of all registers except of GPIO_RES_DIS 1: XRESET_GPIO_SM has no effect on the corresponding register bit					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	GPIO_RES_DIS_1	0h	r	w	Bit-selective reset disabling of XRESET_GPIO_SM signal for registers of General Purpose IOs (31:0) For writing:

					0: XRESET_GPIO_SM resets the corresponding register bit of all registers except of GPIO_RES_DIS 1: XRESET_GPIO_SM has no effect on the corresponding register bit

Register:	GPIO_IN_1			Address:	34h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:	Input register for General Purpose IOs (63:32) 0: GPIO inputx = 0, 1: GPIO inputx = 1					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	GPIO_IN_1	00000000h	rh		Input register for General Purpose IOs (63:32) 0: GPIO inputx = 0, 1: GPIO inputx = 1	

Register:	GPIO_PORT_MODE_1_L			Address:	38h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Configuration register for GPIO port (47:32) Function assignment: 00=function0 01=function1 10=function2 11=function3					
Bit	Identifier	Reset	Attr.		Function / Description	
1dt0	GPIO_32_MODE_1_L	0h	r	w	Port GPIO(32)	
3dt2	GPIO_33_MODE_1_L	0h	r	w	Port GPIO(33)	
5dt4	GPIO_34_MODE_1_L	0h	r	w	Port GPIO(34)	
7dt6	GPIO_35_MODE_1_L	0h	r	w	Port GPIO(35)	
9dt8	GPIO_36_MODE_1_L	0h	r	w	Port GPIO(36)	
11dt10	GPIO_37_MODE_1_L	0h	r	w	Port GPIO(37)	
13dt12	GPIO_38_MODE_1_L	0h	r	w	Port GPIO(38)	
15dt14	GPIO_39_MODE_1_L	0h	r	w	Port GPIO(39)	
17dt16	GPIO_40_MODE_1_L	0h	r	w	Port GPIO(40)	
19dt18	GPIO_41_MODE_1_L	0h	r	w	Port GPIO(41)	
21dt20	GPIO_42_MODE_1_L	0h	r	w	Port GPIO(42)	
23dt22	GPIO_43_MODE_1_L	0h	r	w	Port GPIO(43)	
25dt24	GPIO_44_MODE_1_L	0h	r	w	Port GPIO(44)	
27dt26	GPIO_45_MODE_1_L	0h	r	w	Port GPIO(45)	
29dt28	GPIO_46_MODE_1_L	0h	r	w	Port GPIO(46)	
31dt30	GPIO_47_MODE_1_L	0h	r	w	Port GPIO(47)	

Register:	GPIO_PORT_MODE_1_H			Address:	3Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Configuration register for GPIO port (63:48) Function assignment: 00=function0 01=function1					

		10=function2 11=function3			
Bit	Identifier	Reset	Attr.		Function / Description
1dt0	GPIO_48_MODE_1_H	0h	r	w	Port GPIO(48)
3dt2	GPIO_49_MODE_1_H	0h	r	w	Port GPIO(49)
5dt4	GPIO_50_MODE_1_H	0h	r	w	Port GPIO(50)
7dt6	GPIO_51_MODE_1_H	0h	r	w	Port GPIO(51)
9dt8	GPIO_52_MODE_1_H	0h	r	w	Port GPIO(52)
11dt10	GPIO_53_MODE_1_H	0h	r	w	Port GPIO(53)
13dt12	GPIO_54_MODE_1_H	0h	r	w	Port GPIO(54)
15dt14	GPIO_55_MODE_1_H	0h	r	w	Port GPIO(55)
17dt16	GPIO_56_MODE_1_H	0h	r	w	Port GPIO(56)
19dt18	GPIO_57_MODE_1_H	0h	r	w	Port GPIO(57)
21dt20	GPIO_58_MODE_1_H	0h	r	w	Port GPIO(58)
23dt22	GPIO_59_MODE_1_H	0h	r	w	Port GPIO(59)
25dt24	GPIO_60_MODE_1_H	0h	r	w	Port GPIO(60)
27dt26	GPIO_61_MODE_1_H	0h	r	w	Port GPIO(61)
29dt28	GPIO_62_MODE_1_H	0h	r	w	Port GPIO(62)
31dt30	GPIO_63_MODE_1_H	0h	r	w	Port GPIO(63)

Register:	GPIO_IOCTRL_2			Address:	40h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:	Configuration register for General Purpose IOs (95:64) 0: GPIOx = output, 1: GPIOx = input					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	GPIO_IOCTRL_2	FFFFFFFFh	r	w	Configuration register for General Purpose IOs (95:64) 0: GPIOx = output, 1: GPIOx = input	

Register:	GPIO_OUT_2			Address:	44h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Output register for General Purpose IOs (95:64) 0: GPIO outputx = 0, 1: GPIO outputx = 1					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	GPIO_OUT_2	0h	rh	w	Output register for General Purpose IOs (95:64) 0: GPIO outputx = 0, 1: GPIO outputx = 1	

Register:	GPIO_OUT_SET_2			Address:	48h	
Bits:	31dt0	Reset value:	0h	Attributes:		wt
Description:	Bit-selective setting of the output register for General Purpose IOs (95:64) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 1 Read always returns 0					

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	GPIO_OUT_SET_2	0h		wt	Bit-selective setting of the output register for General Purpose IOs (95:64) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 1 Read always returns 0

Register:	GPIO_OUT_CLEAR_2			Address:	4Ch	
Bits:	31dt0	Reset value:	0h	Attributes:		wt
Description:	Bit-selective reset of the output register for General Purpose IOs (95:64) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 0 Read always returns 0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	GPIO_OUT_CLEAR_2	0h		wt	Bit-selective reset of the output register for General Purpose IOs (95:64) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 0 Read always returns 0

Register:	GPIO_RES_DIS_2			Address:	50h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Bit-selective reset disabling of XRESET_GPIO_SM signal for registers of General Purpose IOs (31:0) For writing: 0: XRESET_GPIO_SM resets the corresponding register bit of all registers except of GPIO_RES_DIS 1: XRESET_GPIO_SM has no effect on the corresponding register bit					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	GPIO_RES_DIS_2	0h	r	w	Bit-selective reset disabling of XRESET_GPIO_SM signal for registers of General Purpose IOs (31:0) For writing: 0: XRESET_GPIO_SM resets the corresponding register bit of all registers except of GPIO_RES_DIS 1: XRESET_GPIO_SM has no effect on the corresponding register bit

Register:	GPIO_IN_2			Address:	54h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:	Input register for General Purpose IOs (95:64)					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	GPIO_IN_2	00000000h	rh		Input register for General Purpose IOs (95:64)

Register:	GPIO_PORT_MODE_2_L	Address:	58h

Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Configuration register for GPIO port (79:64) Function assignment: 00=function0 01=function1 10=function2 11=function3					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	GPIO_64_MODE_2_L	0h	r	w	Port GPIO(64)
3dt2	GPIO_65_MODE_2_L	0h	r	w	Port GPIO(65)
5dt4	GPIO_66_MODE_2_L	0h	r	w	Port GPIO(66)
7dt6	GPIO_67_MODE_2_L	0h	r	w	Port GPIO(67)
9dt8	GPIO_68_MODE_2_L	0h	r	w	Port GPIO(68)
11dt10	GPIO_69_MODE_2_L	0h	r	w	Port GPIO(69)
13dt12	GPIO_70_MODE_2_L	0h	r	w	Port GPIO(70)
15dt14	GPIO_71_MODE_2_L	0h	r	w	Port GPIO(71)
17dt16	GPIO_72_MODE_2_L	0h	r	w	Port GPIO(72)
19dt18	GPIO_73_MODE_2_L	0h	r	w	Port GPIO(73)
21dt20	GPIO_74_MODE_2_L	0h	r	w	Port GPIO(74)
23dt22	GPIO_75_MODE_2_L	0h	r	w	Port GPIO(75)
25dt24	GPIO_76_MODE_2_L	0h	r	w	Port GPIO(76)
27dt26	GPIO_77_MODE_2_L	0h	r	w	Port GPIO(77)
29dt28	GPIO_78_MODE_2_L	0h	r	w	Port GPIO(78)
31dt30	GPIO_79_MODE_2_L	0h	r	w	Port GPIO(79)

Register:	GPIO_PORT_MODE_2_H			Address:	5Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	w
Description:	Configuration register for GPIO port (95:80) Function assignment: 00=function0 01=function1 10=function2 11=function3					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	GPIO_80_MODE_2_H	0h	r	w	Port GPIO(80)
3dt2	GPIO_81_MODE_2_H	0h	r	w	Port GPIO(81)
5dt4	GPIO_82_MODE_2_H	0h	r	w	Port GPIO(82)
7dt6	GPIO_83_MODE_2_H	0h	r	w	Port GPIO(83)
9dt8	GPIO_84_MODE_2_H	0h	r	w	Port GPIO(84)
11dt10	GPIO_85_MODE_2_H	0h	r	w	Port GPIO(85)
13dt12	GPIO_86_MODE_2_H	0h	r	w	Port GPIO(86)
15dt14	GPIO_87_MODE_2_H	0h	r	w	Port GPIO(87)
17dt16	GPIO_88_MODE_2_H	0h	r	w	Port GPIO(88)
19dt18	GPIO_89_MODE_2_H	0h	r	w	Port GPIO(89)
21dt20	GPIO_90_MODE_2_H	0h	r	w	Port GPIO(90)

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

23dt22	GPIO_91_MODE_2_H	0h	r	w	Port GPIO(91)
25dt24	GPIO_92_MODE_2_H	0h	r	w	Port GPIO(92)
27dt26	GPIO_93_MODE_2_H	0h	r	w	Port GPIO(93)
29dt28	GPIO_94_MODE_2_H	0h	r	w	Port GPIO(94)
31dt30	GPIO_95_MODE_2_H	0h	r	w	Port GPIO(95)

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.11 SCRB (System Control Register Block)

This block contains central registers for controlling the ERTEC 200P.

2.3.14.11.1 Hardware Identifier register

Contains the identification number of the ERTEC 200P, see register description, chapter 2.3.14.11.28.1.

2.3.14.11.2 Boot register

The required boot mode is saved in the BOOT_REG boot register. Some of the pins of the EMC interface are assigned to the boot register bits 0...4 (refer to the ERTEC 200P-3 data sheet).

2.3.14.11.3 Config register

Global use cases and different test modes can be set with EMC pins that are latched in the CONFIG_REG register during an XRESET active PowerOn reset. These pins resume their EMC function in normal mode once the reset is cleared.

CONFIG(0):	Enable REF_CLK output (25 MHz) or disable (tristate).
CONFIG(1):	always = '1'.
CONFIG(2):	always = '0'.
CONFIG(6:3):	XHIF interface setting: (off, 16/32-bit, ARM926 trace port) XHIF_XRDY setting: (low-active, high-active) XHIF_XWR setting: (Wr or Read/Write Control) GPIO31-0 setting: GPIO.
EMC_HOSTIF_MUX:	always = '0'.

For additional information on configuration adjustment, refer to the ERTEC 200P-3 data sheet.

2.3.14.11.4 Reset registers

2.3.14.11.4.1 Reset Control register

With the control bits in the ASYN_RES_CTRL_REG and SYN_RES_CTRL_REG registers:

1. An enable for triggering a WD reset for ARM926 and PN-IP can be configured
2. Triggering a **synchronous** module reset for PerIF, HostIF, PN-IP, RS controller
3. Triggering an **asynchronous** block or module reset in response to the following events can be activated/deactivated:
   - Expiry of the watchdog
   - SW activation of a reset (RES_SOFT, RES_SOFT_...) → One-shot bit

---

Note

All reset sources are reset with a power-on reset (XRESET) only. This prevents a combinational loop in the design. The register can only be written word by word. 8-bit and 16-bit accesses result in access errors.

2.3.14.11.4.2 Reset Status register

Active reset events that have occurred since the last PowerOn reset are saved in RES_STAT_REG.

Copyright © Siemens AG 2023. All rights reserved

The vector address for the secondary boot loader following an ARM926EJ-S SW reset is saved in RES_SOFT_RETURN_ADDR.

2.3.14.11.5 ACCESS_ERROR and QVZ register

Access errors by the individual modules are combined here (OR logic operation) and forwarded to the interrupt controller (ARM-ICU) as AHB_Address_Error (IRQ51), APB_Address_Error (IRQ52), EMC_Address_Error (IRQ53) or IP_Address_Error (IRQ86):

1. Master access error interrupt (AHB_Address_Error): QVZ error at **ML-AHB** caused by master access by ARM_I, ARM_D, PN, GDMA, HostIF and RS controller.
2. Slave access error interrupt (APB_Address_Error): QVZ error at **APB** caused by master access by ARM_I, ARM_D, PN, GDMA, HostIF and RS controller.
3. EMC access error interrupt (EMC_Address_Error): QVZ error at **EMC** caused by master access by ARM_I, ARM_D, PN, GDMA, HostIF and RS controller.
4. Slave access error interrupt (IP_Address_Error): Slave module **access error** at the APB or AHB bus (PN-IP, PerIF, I-Filter, HostIF, SCRB and RS controller).

The QVZ_AHB_M contains all the modules (AHB masters) that have caused an access error. Following an activated interrupt, the CPU can read the QVZ_AHB_M register and identify the trigger module on the basis of the bit position in the register. The software can then read the QVZ registers of the AHB masters in the SCRB ('QVZ_AHB_ADR'; 'QVZ_APB_ADR'; 'QVZ_AHB_CTRL') and identify the address and access type that led to the error.

As a general rule, an access error sets the corresponding bit in 'QVZ_AHB_M', freezes the corresponding address and control information and generates an interrupt.

- Until the bit of an AHB master in 'QVZ_AHB_M' is deleted by the SW, another AHB access error (from any AHB master) will **not** be saved. This allows unique assignment of the address and control information in the 'QVZ_AHB_ADR', 'QVZ_APB_ADR' and 'QVZ_AHB_CTRL' registers to the corresponding AHB master, even when multiple access errors from different AHB masters occur at the same time.
- Until the bit of an AHB/APB slave MODUL_ACCESS_ERR is cleared by the SW, another access error for the same APB slave will **not** be saved. An access error for another APB slave can however be locked at the same time, as the address and control information for the erroneous access is buffered in the corresponding APB module or in the SCRB register ('ACCESS_ERROR_SCRB') and locked.

2.3.14.11.6 PLL Status register

You can read the status of the PLL LOCK bit and the oscillator loss information by reading the PLL status register.

2.3.14.11.7 Memory swapping

The reset vector of the ARM926 processor points to address 0x0000_0000. That is why the mirrored area of the boot ROM starts at address 0x0000_0000 (MEM_SWAP = "x00") after reset (HW, SW or ARM926 watchdog reset). The boot ROM can also be addressed in its original address range.

At the end of the boot operation (primary boot or secondary boot), the EMC interface maps:

- asynchronous memory **without** ECC protection (MEM_SWAP = "010") or
- SDRAM **without** ECC protection (MEM_SWAP = "001") or
- asynchronous memory **with** ECC protection (MEM_SWAP = "110") or
- SDRAM **with** ECC protection (MEM_SWAP = "101")

Copyright © Siemens AG 2023. All rights reserved

to address 0x0000_0000h to allow the exception vector table for ARM926EJ-S to be created in address range 0x0000_0000 – 0x0000_001F. The original address ranges for boot ROM (in segment 4), EMC SDRAM (segment 2) and EMC memory (segment 3) are not affected by memory swapping.

Swapping is done by programming the 'MEM_SWAP' register. There is no swapping in MEM_SWAP with reserved coding (MEM_SWAP = “x11”) (or to be precise, no memory space is shown in segment 0). If these addresses are then accessed at the AHB, a QVZ (see chapter “Monitoring at the AHB Side”) is triggered.

Note

MEM_SWAP must not be changed during active accesses to the EMC.

2.3.14.11.8 ARM Control register

The following control information can be configured in this register:

1. ARB: Setting for the arbitration procedure for ML-AHB (see chapter 2.3.14.11.8.1)
2. ARM926 AHB-LOCK

2.3.14.11.8.1 AHB Arbiters

Each of the ML-AHB AHB arbiters uses the same arbitration procedure. The default setting is round robin. A fixed priority assignment for the AHB masters could be used as an alternative arbitration algorithm by programming the ARB bit in the SCRB register. Switching to "fixed priority" fixes the priority set at ML-AHB when the arbitration type was configured. This option should not, however, be implemented in the light of the dynamic processes at the multi-layer AHB.

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

2.3.14.11.9 PHY register

The configuration of both integrated PHYs can be set in the **PHY control register** PHY_CONFIG. This configuration is possible alongside the PHY configuration over SMI (PN-IP). With PHY_LED_CONTROL, you can control whether the PHY-LED link status and activity are to be controlled by the integrated PHY or SW. The LED status of the SW is also saved.

When writing a '1' to bitfields P1_PHY_ENB or P2_PHY_ENB, the change effects the output signals pxphyen_o(2:1) after a delay of 25 cycles of CLK. Writing a '0' is not delayed. (This is implemented to ease constraining.)

The readiness of both integrated PHYs can be read from the **PHY status register** PHY_STATUS.

2.3.14.11.10 AHB Burst Breaker register

This register limits the maximum number of addresses of an AHM master in an ARM926 burst. A maximum limit of 256 addresses can be set. After the set number of addresses, burst access by the ARM926 is interrupted by an IDLE sequence. This function is required if an AHB master is blocking the bus for too long with burst access and preventing another AHB master from accessing the slave in time (for details, see chapter 2.3.4.3.3).

2.3.14.11.11 GPIO Control register

With LOCAL_OUT_READ_L and LOCAL_OUT_READ_H, you can configure whether the value of the GPIO block or of the data IO value is returned for GPIOs 32…95 (see chapter 2.3.14.10.5.2).

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

To adapt the interrupt-compatible GPIOs 0...15 to an external circuit that supplies both low-active and high-active signals, they can be connected to the following modules over 'GPIO_INT_POLSEL' either unchanged (not inverted) or inverted:

- ARM-ICU (GPIO0…15 → IRQ32…47, see 2.3.5.9)
- PN-ICU1 (GPIO4…7, see 2.3.12.1.2)
- GDMA (GPIO0…3 → HW jobs 16…19, see 2.3.7.2.1.1)

POLSEL_GPIOx must be configured to ensure that these modules always receive a high-active signal. The value in the 'GPIO_IN_0' register is not affected by POLSEL_GPIOx.

2.3.14.11.12 I²C Clock Divider

The bit rate of the I²C bus is configured with the 'CCR_I2C' register. I²C transmission rates of max. 100 kb/s require a 1 MHz enable for I²C-IP (MI2C: Inventra). This enable is implemented in *CCR_I2C.CDIV_VAL* configuration. The value to be configured is calculated using the following formula:
CLK [MHz] / (ClockDivider + 1) = 1 MHz → with a 125 MHz system clock → *CDIV_VAL* = 0x7C (Default)

Note

A max. transmission rate of 100 kb/s is supported.

2.3.14.11.13 EDC register

The EDC logic for I/D-TCM and I/D-Cache in ARM926 can be enabled and disabled in the EDC register 'EDC_PARITY_EN'. The parity logic for the I and D cache for ARM926 can also be activated.

One-bit and multi-bit errors for all EDC-protected RAM are saved in 'EDC_EVENT' and the corresponding level-triggered interrupt event EDC_Event (IRQ48) triggered in ARM-ICU (see chapter 2.3.5.9). The register bits (i.e., the active interrupt level) can only be reset with write access by the FW (writing a '0').

Note

The register bits can also be set by the FW for debugging with write access (writing a '1').

If a one-bit or multi-bit error is saved at the same time as the FW deletes the error with write access, the one-bit or multi-bit error will not be deleted.

The end of RAM initialization (INIT_DONE) for all EDC-protected RAM is signaled in EDC_INIT_DONE.

Memories are integrated in the modules of the ERTEC 200P-3. They store code and management structures that are configured once and that remain unchanged for long periods of time. A soft error in these memory structures can therefore not be corrected by a cyclic SW update of these memories during operation. The memories are:

- ARM926: I-TCM, D-TCM, I-Cache, D-Cache
- PN-IP: FDB-RAM, PackData-RAM0/1, APICtrl-RAM, SYNC-RAM

A cyclic refresh of the memory contents in the HW can be activated for these memories by means of *EDC_CTRL_REG.EN_EDC_RFRSH_…* . In this process the contents of the RAM are "refreshed" cyclically by reading a memory cell and, if applicable, writing back this memory cell (refresh sequence). Writing back only takes place if a one-bit EDC error is recognized when the memory cell is read. In addition, this one-bit EDC error is entered RAM-specifically into the *EDC_STAT_REG* and the global correction counter *EDC_STAT_REG.EDC_CORR_COUNTER* is

Copyright © Siemens AG 2023. All rights reserved

incremented (an overflow of the correction counter does not take place - the SW can reset *EDC_STAT_REG* explicitly by writing a 0x00). In this case, **no** ...*TCM..._1B* is entered in the 'EDC_EVENT'. Multi-bit errors due to cyclic refresh are saved in 'EDC_EVENT' and the corresponding level-triggered interrupt event EDC_Event (IRQ48) triggered in ARM-ICU (see chapter “Interrupt sources for ARM-IRQ”). The register bits (i.e., the active interrupt level) can only be reset with write access by the FW (writing a '0').

The interval between two consecutive refresh sequences can be set by means of *EDC_CTRL_REG.EDC_REFRESH_CYCLE* in cycles of 0 ms, 4 ms, 8 ms, 16 ms, 32 ms, 64 ms, 128 ms, 256 ms.

NOTICE

With *EDC_CTRL.EDC_REFRESH_CYCLE* = 0x0 the refresh cycle is carried out permanently. This can be used, for example, during operation to stress the ASIC specifically.

This setting is prohibited for normal operation!

The refresh sequence (RD-WR) is only executed on the memory cell if a memory access of the module/SW is not pending at the same time. The memory access of the module/SW always takes higher priority and is not extended/slowed down by the refresh sequence. This also applies within a refresh sequence, meaning that after reading the refresh sequence it can be interrupted by a RD/WR access of the module/SW. The read access of the refresh is repeated afterwards.

If a one-bit error at address “x” is recognized during a memory access of the module/SW, the refresh sequence is continued directly at address “x” (without execution of *EDC_REFRESH_CYCLE*) and therefore corrects the erroneous RAM contents, if *EDC_CTRL_REG.EN_EDC_CORR_*... is set **additionally**. The next refresh cycle is then carried out after the expiry of *EDC_REFRESH_CYCLE* on the subsequent address x+4.

Each complete execution of the cyclic refresh across the entire memory, meaning **no** one-bit EDC error due to a memory access of the module/SW was recognized and corrected by the refresh logic, results in an automatic resetting of the respective *EDC_CTRL_REG.EN_EDC_RFRSH_*... at *EDC_CTRL_REG.OS_EDC_RFRSH_*... = '1' (a recognized and corrected one-bit EDC error in the course of a cyclic refresh is not relevant for automatic resetting). Afterwards the cyclic refresh is deactivated (*EDC_CTRL_REG.EN_EDC_RFRSH_...RAM* = '0').

If **solely** *EDC_CTRL_REG.EN_EDC_CORR_*... is set, only the faulty RAM content that was recognized as a one-bit error by a memory access of the module/SW at address “x” is corrected directly at address “x” promptly through a correction access. In addition, this one-bit EDC error is entered RAM-specifically into the *EDC_STAT_REG*, and the global correction counter *EDC_STAT_REG.EDC_CORR_COUNTER* is incremented.

NOTICE

The cyclic refresh (*EDC_CTRL_REG.EN_EDC_RFRSH_...*) must only be activated **after** the partitioning of ARM926 subsystem TCM into I-TCM and D-TCM. Otherwise, the behavior of refresh is undefined.

2.3.14.11.14 ARM926 mapping

The partitioning of ARM926 subsystem TCM into I-TCM and D-TCM can be configured in steps of 64 KB in 'TCM926_MAP' (see SCRB register 'TCM926_MAP' in chapter 2.3.14.11.28.1).

2.3.14.11.15 PAD Control register

The IO buffers at the XHIF/GPIO interface support both 3.3 V and 1.8 V IO. The IO buffers of the EMC interface are designed for an IO voltage of 1.8 V. The **driver power** of these special IO buffers depends on the IO voltage and can be configured in the SCRB registers 'DRIVE_EMC' and 'DRIVExx_yyGPIO'.

Copyright © Siemens AG 2023. All rights reserved

The **pull properties** of the GPIO31-0 and GPIO95-32 block signal pads can also be set in the SCRB register 'PULLxx_yyGPIO'.

2.3.14.11.15.1 Pull setting (PULL)

The following registers are used to set the pull for the inputs (see 2.3.14.11.27.1):

- PULL15_0GPIO: Covers the GPIO15-0 pads
- PULL31_16GPIO: Covers the GPIO31-16 pads
- PULL47_32GPIO: Covers the GPIO47-32 pads
- PULL63_48GPIO: Covers the GPIO63-48 pads
- PULL79_64GPIO: Coves the GPIO79-64 pads
- PULL95_80GPIO: Covers the GPIO95-80 pads

The following combinations can be set for each GPIO pad:

Coding	Pull
00b	highZ
01b	Pull-Up
10b	highZ
11b	Pull-Down

The pull setting is only effective if the pad is connected to the input. If the pad is operating as an output, the pull is deactivated. The pulls have a resistance of 35 - 65 kΩ (typ. 50 kΩ).

During/following a reset, the pull settings in the registers are preassigned. The reset values can be found in the register description. With GPIO95-0, there is also a dependency on the CONFIG(6:3) pins. Different pull settings are used during/after a reset for the 7 different modes. The settings can be changed by the SW at any time.

Note

The pull setting does not change automatically following reconfiguration of the CONFIG(6:3) bits in CONFIG_REG by the SW. It must be changed by the SW.

2.3.14.11.15.2 Driver Power (DRIVE)

The following registers are used to set the driver power of the outputs.

- DRIVE_EMC:
  Contains all EMC interface signal pads, divided into signal groups G1 – G11 (see register); settings for 1.8 V possible
- DRIVE15_0GPIO:
  Contains the GPIO15-0 pads; settings only possible for 3.3 V
- DRIVE31_16GPIO:
  Contains the GPIO31-16 pads; settings only possible for 3.3 V
- DRIVE47_32GPIO:
  Contains the GPIO47-32 pads; settings for 1.8 V/3.3 V possible
- DRIVE63_48GPIO:
  Contains the GPIO63-48 pads; settings for 1.8 V/3.3 V possible

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

- DRIVE79_64GPIO:
  Contains the GPIO79-64 pads; settings for 1.8 V/3.3 V possible
- DRIVE95_80GPIO:
  Contains the GPIO95-80 pads; settings for 1.8 V/3.3 V possible

The following combinations can be set for each GPIO pad/signal group (EMC for 1.8 V/3.3 V; GPIO31-0 for 3.3 V only; GPIO95-32 for 1.8 V/3.3 V). Following a PowerOn reset, the driver power is by default set to:

- 8 mA for GPIO31-0 pads at 3.3 V
- 8 mA for GPIO95-32/XHIF pads at 3.3 V
- 4 mA for GPIO95-32/XHIF pads at 1.8 V
- 6 mA for EMC pads at 1.8 V

For the 1.8 V XHIF interface, the external host may need to increase the driver power to 9 mA before it executes the first instance of read access. 4 mA is OK for the XHIF_XRDY pad, as the connection is a simple point-to-point one. The alternate functions (trace port, ...) are operated at 3.3 V.

Table 38: Driver power after PowerOn reset

Coding	GPIO31-0 (3.3 V)	GPIO95-32 / XHIF		EMC
		(3.3 V)	(1.8 V)	(1.8 V)
00b	2 mA	2 mA	1 mA	1 mA
01b	4 mA	4 mA	2 mA	2 mA
10b	8 mA (default)	8 mA (default)	4 mA (default)	4 mA
11b	12 mA	12 mA	6 mA	6 mA (default)

Following a reset, the driver power settings in the registers are preassigned. The settings can be changed by the SW at any time.

2.3.14.11.15.3 Reset behavior of EMC pins without Pull Circuit

Some of the ERTEC 200P EMC output pins have no internal pull resistors:

- CLK_O_SDRAM2

  CLK_O_SDRAM0-1 are by default connected in line with DRIVE_EMC(22) (see chapter 2.3.14.11.27.1) and are therefore not high-impedance during a reset.
- CLK_O_BF0 – 2

Take this into account when connecting external components, as these pins become high-impedance during a reset and produce floating input signals at the other components. Blocks whose inputs are connected to these ERTEC 200P EMC output pins without internal pull resistors must, if necessary, be protected from a floating input voltage with external pulls.

2.3.14.11.16 Clear DMA request

See chapter 2.3.14.9.4.

2.3.14.11.17 SPI parity error

The group interrupt INT_SPI_ParityERR for the ARM-ICU (IRQ17) is generated in SPI_PARITY_ERROR.

2.3.14.11.18 XHIF mode

Eight different address ranges (pages) of the ERTEC 200P can be addressed individually over the parallel host interface in HostIF (XHIF, see 2.3.10.2). The size of the max. address ranges (pages) can be configured with XHIF_MODE. Following a reset, a default max. address range per page is 1 MByte; the ASIC pin *XHIF_XCS_R_A20* is connected to XHIF_XCS_R, so the pages can be reconfigured at the XHIF interface by the ext. host. When XHIF_MODE = '1', the ASIC pin *XHIF_XCS_R_A20* is connected to XHIF_A20, expanding the address range for access to the ERTEC 200P to 2 MByte per page.

2.3.14.11.19 Ext. Driver Enable

With EXT_DRIVER_EN, you can configure the address range of the asynchronous EMC interface (CS0…CS3) for the chip select-specific control of an external driver at the EMC interface using the DTXR and XOE_DRIVER signals (for details, see 2.3.8.4.4).

It is also possible to configure the CS0 address range (0x3000_0000 – 0x33FF_FFFF) at the EMC pin *XRDY_BF* by latching driver control information (*EXT_DRIVER_DISABLE_CS0*) when the reset (XRESET) is cleared. The integrated pull-up means that control for an external driver is deactivated by default and needs to be activated with a pull-down for the module.

2.3.14.11.20 SPI mode

SPI1 or SPI2 operation as MISO/MOSI or SPI-IP wiring straight to the GPIOs can be set on an SPI-specific basis with 'SPI_MODE' (for details, see 2.3.14.9.3)

2.3.14.11.21 SD signal handling

When the ERTEC 200P is operated with fiber-optics, the external fiber-optic transceiver signals with signal detect (SD) whether a fiber-optic signal has been received. This SD signal is to be filtered on a port-specific basis and output as a differentiated signal for the ERTEC 200P integrated PHY. To avoid complex logic for preprocessing the SD signal in a given module, the ERTEC 200P can implement port-specific filtering and differentiation of the SD signal itself with SD_CONTROL.

A general enable for filtering and differentiation is implemented in SD1/2_ENABLE for this purpose. If it is not, the ERTEC die port-specific SD+ and SD- are switched to high impedance to continue to allow the direct connection of an external differentiated SD signal over the ERTEC 200P BGA balls.

If SD1/2_ENABLE = '1', filtering and differentiation are over the ERTEC die. With SD1/2_MUX, you can select the I-Filter output over which the SD signal of a given port is received. A logical '0' can also be forced in place of the SD input signal for SW control of the SD signal when SD1/2_MUX = 96…112. A logical '1' is used when SD1/2_MUX = 113…127.

For the I-Filter selected by SD1/2_MUX with SD1/2_SELECT you can select the input from which the SD signal is received (GPIOx, FX_SD_OUT(1), or FX_SD_OUT(2). The settings made for SD1 have higher priority than the settings made for SD2.

```
For (i = 0; I < 96; i++) {
    If (SD1_MUX == i) {
        SD1_SELECT == 0 ? GPIO(n) : FX_SD_OUT(1);
    } else if (SD2_MUX == i) {
        SD2_SELECT == 0 ? GPIO(n) : FX_SD_OUT(2);
    }
}
```

Note

When SD_CONTROL.SD1/2_MUX = 16…31, there is differentiation but generally **no** filtering (see chapter 2.3.14.2).

Figure 56: SD signal handling

For external wiring over GPIO, refer to the ERTEC 200P-3 data sheet.

2.3.14.11.22 Extension mode

After power-on reset the enabled features depend on SCRB.EXTENSION_REG.X_MODE; it must be configured to '1', all other values are reserved.

2.3.14.11.23 Setup register

The following control information is saved in this register:

1. PERIF_CC_MUX: Reserved, must be '1'.
2. OCTALSPI_EN: Enabling of the clock systems of Octal-SPI.
3. DEV_HWK: Device hardware identification from the dedicated ASIC pins (refer to the ERTEC 200P-3 data sheet).

2.3.14.11.24 Enabling of Octal-SPI

After a reset (reset causes, see chapter 3.3.16.4) the clock systems of Octal-SPI are not enabled Through parameter assignment of *OCTALSPI_EN* the clock system is activated and while operating can only be reset by using a Reset. *OCTALSPI_EN* can therefore only be written once with '1'. Writing with '0' and read accesses are possible several times.

2.3.14.11.25 Reset register extensions

The module I²C can be reset dedicatedly by means of a module-specific synchronous reset (RESET_REG.SW_RES_...). The reset level acts asynchronously.

In addition, the register informs you whether the dedicated reset which was activated through the SCRB.RESET_REG.SW_RES_... was executed in the module I²C.

The SW can therefore determine that the modules are in Reset by polling RESET_REG.SW_RES_..._DONE. Writing a '0' to RESET_REG.SW_RES_... takes effect only if RESET_REG_..._DONE is '1'. When the dedicated reset in SW_RES_... is deleted, the associated SW_RES_..._DONE is also reset.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

2.3.14.11.26 GDMA control

To allow a running job of the GDMA to be interrupted and restarted via SW access, the GDMA jobs 31…24 can be controlled through GDMA_CTRL.JOBx_SUSPEND (see chapter 2.3.7.2.1.2).

For this purpose, the HW FlowControl has to be activated in GDMA for the respective job: *GDMA_JOBx_CTRL.HW_FLOW_EN* = '1'

The respective JOBx (x = 31…24) is then interrupted by the SW, if the following applies:

- The associated bit is set in the SCRB: *JOBx_SUSPEND* = '1'

The respective JOBx (x = 31…24) is then restarted by the SW (i.e., processed further), if the following applies:

- The associated bit is reset in the SCRB: *JOBx_SUSPEND* = '0'

2.3.14.11.27 Address mapping

2.3.14.11.27.1 SCRB1 address mapping

Start-Address	End-Address	Modul/Memory-Name
0h	FFh	PN_SUB_SCRB

Module	Register/Memory	Read	Write	Access	Address
/PN_SUB_SCRB				**32-Bit**	
	/ID_REG	r		32-Bit	0h
	/BOOT_REG	rh		32-Bit	4h
	/CONFIG_REG	rh	(w)	32-Bit	8h
	/ASYN_RES_CTRL_REG	rh	w	32-Bit	Ch
	/SYN_RES_CTRL_REG	r	w	32-Bit	10h
	/RES_STAT_REG	rh	(w)	32-Bit	14h
	/PLL_STAT_REG	rh		32-Bit	18h
	/QVZ_AHB_ADR	rh		32-Bit	1Ch
	/QVZ_AHB_CTRL	rh	w	32-Bit	20h
	/QVZ_AHB_M	rh		32-Bit	24h
	/QVZ_APB_ADR	rh	w	32-Bit	28h
	/QVZ_EMC_ADR	rh	w	32-Bit	2Ch
	/MEM_SWAP	r	w	32-Bit	30h
	/M_LOCK_CTRL	(r)	(w)	32-Bit	34h
	/ERTEC200PLUS_TAG	r		32-Bit	38h
	/PHY_CONFIG	rh	w	32-Bit	3Ch
	/PHY_STATUS	rh		32-Bit	40h
	/AHB_BURSTBREAKER	r	w	32-Bit	44h
	/LOCAL_OUT_READ_L	r	w	32-Bit	48h
	/LOCAL_OUT_READ_H	r	w	32-Bit	4Ch
	/CCR_I2C	r	w	32-Bit	50h
	/EDC_EVENT	rh	w	32-Bit	54h
	/EDC_INIT_DONE	rh		32-Bit	58h
	/TCM926_MAP	r	w	32-Bit	5Ch
	/GPIO_INT_POLSEL	r	w	32-Bit	60h
	/EDC_PARITY_EN	r	w	32-Bit	64h
	/MODUL_ACCESS_ERR	rh	w	32-Bit	68h

Copyright © Siemens AG 2023. All rights reserved

	/RES_SOFT_RETURN_ADDR	r	w	32-Bit	6Ch
	/PHY_LED_CONTROL	(r)(h)	(w)	32-Bit	70h
	/ACCESS_ERROR_SCRB	(r)(h)	(w)	32-Bit	74h
	/DRIVE_EMC	r	w	32-Bit	78h
	/DRIVE15_0GPIO	r	w	32-Bit	7Ch
	/DRIVE31_16GPIO	r	w	32-Bit	80h
	/DRIVE47_32GPIO	r	w	32-Bit	84h
	/DRIVE63_48GPIO	r	w	32-Bit	88h
	/DRIVE79_64GPIO	r	w	32-Bit	8Ch
	/DRIVE95_80GPIO	r	w	32-Bit	90h
	/PULL15_0GPIO	rh	w	32-Bit	94h
	/PULL31_16GPIO	rh	w	32-Bit	98h
	/PULL47_32GPIO	rh	w	32-Bit	9Ch
	/PULL63_48GPIO	rh	w	32-Bit	A0h
	/PULL79_64GPIO	rh	w	32-Bit	A4h
	/PULL95_80GPIO	rh	w	32-Bit	A8h
	/CLEAR_DMA_SPI_REQ_REG	rh	w	32-Bit	ACh
	/SPI_PARITY_ERROR	rh	w	32-Bit	B0h
	reserved	-	-	32-Bit	B4h
	/XHIF_MODE	r	w	32-Bit	B8h
	/EXT_DRIVER_EN	r(h)	w	32-Bit	BCh
	/SPI_MODE	r	w	32-Bit	C0h
	/SD_CONTROL	r(h)	w	32-Bit	C4h
	/EXTENSION_REG	rh	(w)(p)	32-Bit	FCh

2.3.14.11.27.2 SCRB2 address mapping

Start-Address	End-Address	Modul/Memory-Name
0h	3Fh	PN_SUB_SCRB2

Module	Register/Memory	Read	Write	Access	Address
/PN_SUB_SCRB2				**32-Bit**	
	/SETUP_REG	r(h)	(w)	32-Bit	0h
	/RESET_REG	r(h)	(w)	32-Bit	4h
	reserved	-	-	32-Bit	8h
	/GDMA_CTRL	r	w	32-Bit	Ch
	reserved	-	-	32-Bit	10h
	/OCTALSPI_REG	rh	w	32-Bit	14h
	/OCTALSPI_STROBE_REG	r	w	32-Bit	18h
	reserved	r	w	32-Bit	1Ch
	reserved	r	w	32-Bit	20h
	reserved	r	w	32-Bit	24h
	reserved	-	-	32-Bit	28h

Copyright © Siemens AG 2023. All rights reserved

	reserved	r	w	32-Bit	2Ch
	reserved	r	w	32-Bit	30h
	reserved	r	w	32-Bit	34h
	/EDC_CTRL_REG	r(h)	w(p)	32-Bit	38h
	/EDC_STAT_REG	rh	w	32-Bit	3Ch

2.3.14.11.28 Register description

2.3.14.11.28.1 SCRB1 register description

Module: PN_SUB_SCRB

Register:	ID_REG			Address:	0h	
Bits:	31dt0	Reset value:	40280301h	Attribu-tes:	r	
Description:	Identification ERTEC200P					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	MET_FIX	1h	r		
15dt8	HW_R	3h	r		HW-Release: 03h
31dt16	COMP	4028h	r		ERTEC200P ID: 4028h

Register:	BOOT_REG			Address:	4h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	rh	
Description:	Boot-Mode-Pins readable					

Bit	Identifier	Reset	Attr.		Function / Description
0	BOOT_0	0h	rh		The value pending at the Boot(0) pin is latched when PowerOn reset is deactivated
1	BOOT_1	0h	rh		The value pending at the Boot(1) pin is latched when PowerOn reset is deactivated
2	BOOT_2	0h	rh		The value pending at the Boot(1) pin is latched when PowerOn reset is deactivated
3	BOOT_3	0h	rh		The value pending at the Boot(3) pin is latched when PowerOn reset is deactivated

4	BOOT_4	0h	rh		The value pending at the Boot(4) pin is latched when PowerOn reset is deactivated
31dt5	reserved	0000000h			

Register:	CONFIG_REG			Address:	8h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	rh	(w)
Description:	ERTEC200P ConfigPins CONFIG(6:0) readable, config REG(6:3) writable					

Bit	Identifier	Reset	Attr.		Function / Description
0	CONF_0	0h	rh		The value pending at the Con-fig(0) pin is latched when PowerOn reset is deactivated
1	CONF_1	0h	rh		The value pending at the Con-fig(1) pin is latched when PowerOn reset is deactivated
2	CONF_2	0h	rh		The value pending at the Con-fig(2) pin is latched when PowerOn reset is deactivated
6dt3	CONF_6_3	0h	rh	w	The value pending at the Con-fig(6:3) pins is latched when PowerOn reset is deactivated and added to the register following de-activation of the internally extended system re-set; the value can then be modified with write access 00-0: XHIF = on, 16-bit mode, GPIO94-79, 61-60 on (all inputs), XHIF_XWR has read/write control 00-1: XHIF = on, 16-bit mode, GPIO94-79, 61-60 on (all inputs), XHIF_XRD /

					XHIF_XWR separate 000-: XHIF = on, 16-bit mode, GPIO94-79, 61-60 on (all inputs), XHIF_XRDY is highactive 001-: XHIF = on, 16-bit mode, GPIO94-79, 61-60 on (all inputs), XHIF_XRDY is lowactive 01-0: XHIF = on, 32-bit mode, GPIO95-32 off, XHIF_XWR has read/write control 01-1: XHIF = on, 32-bit mode, GPIO95-32 off, XHIF_XRD / XHIF_XWR separate 010-: XHIF = on, 32-bit mode, GPIO95-32 off, XHIF_XRDY is high-active 011-: XHIF = on, 32-bit mode, GPIO95-32 off, XHIF_XRDY is low-active 1000: XHIF = SPI, GPIO89-32 on (all inputs) 1001: XHIF = off, GPIO95-32 on (all inputs) 1100: reserved 1101: reserved
7	EMC_HOSTIF_MUX	0h	rh		EMC pins are used as 0: memory interface (EMC) (default) 1: reserved Only one-time settable by the EMC_HOSTIF_MUX strapping pin.
31dt8	reserved	000000h			

Register:	ASYN_RES_CTRL_REG			Address:	Ch	
Bits:	31dt0	Reset value:	28h	Attributes:	rh	w
Description:	Control register for ERTEC200P reset Reset for slice EN_WD_RES_PN is with PNIP reset. The remaining slices are reset with a system reset.					
Bit	Identifier	Reset	Attr.	Function / Description		

0	WD_RES_FREI_ARM926	0h	rh	w	1: Watchdog reset enable for ARM926
1	RES_SOFT	0h	rh	w	1: Asynchronous software reset (not retentive, PULSE_DUR is reset length), resets everything except PN-IP
2	RES_SOFT_PN	0h	rh	w	1: Asynchronous software reset (not retentive, PULSE_DUR is reset length), resets PN-IP only
3	EN_WD_RES_PN	1h	rh	w	0: PN-IP is not reset by an ARM926 watchdog reset 1: PN-IP is reset by an ARM926 watchdog reset
4	reserved	0h	rh	w	
5	reserved	1h	rh	w	
6	RES_SOFT_ARM926_CORE	0h	rh	w	1: Asynchronous software reset (not retentive, PULSE_DUR is reset length), resets ARM926EJ-S core system only
7	reserved	0h	rh	w	
15dt8	reserved	00h			
31dt16	PULSE_DUR	0h	rh	w	Pulse duration for SW resets RES_SOFT, RES_SOFT_PN, ARM926 watchdog reset TRES_PULSE = (8 x n + 8) x TCLK TCLK: APB clock period (1/125 MHz = 8 ns) n: Value of PULSE_DUR (0 .. 65535) The integrated PHYs require a re-set duration of - 100 us. n - 1562 must therefore be set

Register:	SYN_RES_CTRL_REG	Address:	10h

Bits:	31dt0	Reset value:	0h	Attribu-tes:	r	w
Description:	Control register for synchronous ERTEC200P reset					

Bit	Identifier	Reset	Attr.		Function / Description
0	SYN_RES_PER_IF	0h	r	w	0: No synchronous reset with PER-IF 1: Synchronous reset with PER-IF (retentive)
1	SYN_RES_HOST	0h	r	w	0: No synchronous reset with host interface 1: Synchronous reset with host in-terface (retentive)
2	SYN_RES_PN_IP	0h	r	w	0: No synchronous reset with PN-IP 1: Synchronous reset with PN-IP (retentive)
3	SYN_RES_RS_CONTROLLER	0h	r	w	0: No synchronous reset with RS controller 1: Synchronous reset with RS controller (retentive)
31dt4	reserved	0000000h			

Register:	RES_STAT_REG			Address:	14h	
Bits:	31dt0	Reset value:	80000004h	Attribu-tes:	rh	(w)
Description:	Status register for ERTEC200P reset Only the bit for the most recent reset event is ever set. The two other bits are reset.					

Bit	Identifier	Reset	Attr.		Function / Description
0	ARM926_WDOG_RES	0h	rh		1: The last reset was with the ARM926 watchdog
1	SW_RES	0h	rh		1: The last reset was a software reset
2	PWRON_HW_RES	1h	rh		1: The last reset was a power on or hardware reset
3	SW_RES_ARM926	0h	rh		1: The last reset was an ARM926 core software reset

30dt4	reserved	0000000h			
31	reserved	1h	rh	w	

Register:	PLL_STAT_REG			Address:	18h	
Bits:	31dt0	Reset value:	1h	Attribu-tes:	rh	
Description:	Status register for ERTEC200P PLL					
Bit	Identifier	Reset	Attr.		Function / Description	
0	LOCK	1h	rh		Lock: Lock at operating frequency status of PLL: 0: PLL is unlocked 1: PLL is locked This bit represents the current lock status of the PLL. Read-only	
1	LOSS	0h	rh		Loss: PLL input clock monitoring status 1: PLL input clock not recognized 0: PLL input clock This bit shows the current moni-toring status of the PLL input clock. Read-only	
31dt2	reserved	00000000h				

Register:	QVZ_AHB_ADR			Address:	1Ch	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	rh	
Description:	Address of incorrect addressing at the multi-layer AHB					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	QVZ_AHB_ADR	00000000h	rh		Address of incorrect addressing at the multilayer AHB	

Register:	QVZ_AHB_CTRL			Address:	20h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	rh	w
Description:	Control signals of incorrect addressing at the multi-layer AHB. The first occurance of an error is stored in QVZ_AHB_ADR, QVZ_AHB_CTRL, and QVZ_AHB_M. A new entry is only pos-sible if the Lock-Bit in QVZ_AHB_CTRL was reset by by a write access.					

Bit	Identifier	Reset	Attr.		Function / Description
0	R_W	0h	rh	w	Read/Write 0: read 1: write **Note**: Writing this registers only resets the Lock-Bit and has no effect on the field value.
3dt1	HSIZE	0h	rh	w	HSIZE **Note**: Writing this registers only resets the Lock-Bit and has no effect on the field value.
6dt4	HBURST	0h	rh	w	HBURST **Note**: Writing this registers only resets the Lock-Bit and has no effect on the field value.
31dt7	reserved	0000000h			

Register:	QVZ_AHB_M			Address:	24h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:	Master detection for incorrect addressing at the multi-layer AHB					

Bit	Identifier	Reset	Attr.		Function / Description
0	ARM_I	0h	rh		ARM926-I
1	ARM_D	0h	rh		ARM926-D
2	reserved	0h			
3	PN	0h	rh		PROFINET_IP
4	GDMA	0h	rh		GDMA
5	HOSTIF	0h	rh		Host Interface
6	RSC1	0h	rh		Reed-Solomon Controller
7	RSC2	0h	rh		Reed-Solomon Controller
8	reserved	0h			
9	reserved	0h			
31dt8	reserved	000000h			

Register:	QVZ_APB_ADR			Address:	28h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Address of incorrect addressing at the APB. The first occurence of an error is stored in the register.					

		A new entry is only possible if the Lock-Bit in QVZ_APB_ADR was reset by by a write access.			
Bit	Identifier	Reset	Attr.		Function / Description
31dt0	QVZ_APB_ADR	00000000h	rh	w	Address of incorrect addressing at the APB **Note**: Writing this registers only resets the Lock-Bit and has no effect on the field value.

Register:	QVZ_EMC_ADR			Address:	2Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Address that resulted in QVZ at EMC The first occurence of an error is stored in the register. A new entry is only possible if the Lock-Bit was reset by by a write access.					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	QVZ_EMC_ADR	00000000h	rh	w	Address that resulted in QVZ at EMC **Note**: Writing this registers only resets the Lock-Bit and has no effect on the field value.	

Register:	MEM_SWAP			Address:	30h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Memory swapping in segment 0 on the AHB (ROM, EMC-SDRAM, EMC standard memory, I/D-TCM)					
Bit	Identifier	Reset	Attr.		Function / Description	
2dt0	SWAP	0h	rh	w	Selection of memory in segment 0 on the AHB: x00: Boot ROM from addr 0h 001: EMC-SDRAM from addr 0h (only the first 64 MByte) without ECC 010: EMC standard memory from addr 0h (only the first 64 MByte) without ECC 101: EMC-SDRAM from addr 0h (only the first 64 MByte) with ECC 110: EMC standard memory from addr 0h (only the first 64 MByte) with ECC	

					111: reserved (no memory range is to addr 0h; QVZ is generated upon access)
31dt3	reserved	00000000h			

Register:	M_LOCK_CTRL			Address:	34h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	(r)	(w)
Description:	AHB master lock enable. Master-specific enable of AHB lock function.					

Bit	Identifier	Reset	Attr.		Function / Description
0	LE_ARM926_I	0h	r	w	Lock enable, AHB master ARM926-I: (Imaster does not support locked trans-fers) 0: Lock disabled 1: Lock enabled
1	LE_ARM926_D	0h	r	w	Lock enable, AHB master ARM926-D: 0: Lock disabled 1: Lock enabled
2	reserved	0h	r	w	
5dt3	reserved	0h			
6	ARB	0h	r	w	Select arbitration algorithm for AHB arbiter (ARB_MODE): 0: Round robin 1: Fixed priority assignment This bit should not be changed (default: round robin)!
31dt7	reserved	0000000h			

Register:	ERTEC200PLUS_TAG			Address:	38h	
Bits:	31dt0	Reset value:	1843h	Attribu-tes:	r	
Description:	Tag number of current revision					

Bit	Identifier	Reset	Attr.		Function / Description
10dt0	R_LABEL	43h	r		R-Label
15dt11	INKREMENT_NR	3h	r		Increment no. (not used, see ID_REG slice HW_R)
18dt16	PATCH_NR	0h	r		Increment no. (not used, see ID_REG slice HW_R)

20dt19	PLATFORM	0h	r		Platform:00 = ASIC, 01 = FPGA, 10 = reserved, 11 = userdefined
31dt21	IDENTIFICATION	0h	r		Identification (not used, see ID_REG Slice COMP)

Register:	PHY_CONFIG			Address:	3Ch	
Bits:	31dt0	Reset value:	8080h	Attribu-tes:	rh	w
Description:	Configuration of PHY1 and PHY2 Important: PHY_CONFIG register is only reset with an asyn-chronous PN-IP reset.					

Bit	Identifier	Reset	Attr.		Function / Description
0	P1_PHY_ENB	0h	rh	w	0: PHY1 disabled (power down mode) - Comment: If the PHY is 'Disabled' and then 'Enabled' again, the SW must comply with a disable time of 100 s. 1: PHY1 enabled - Comment: P1/2_PHYENABLE = 1 triggers a reset extension of 5.2 msec in the PHY. During this time, the PLL and all analog and digital components are started up. Readiness is signaled in the PHY_Status register with P1/2_PWRUPRST = 1. Comment: When CON-FIG(6:3)="1101", the bit is not writable and is fixed at the default value.
1	P1_FX_MODE	0h	rh	w	1: Enables 100BASE-FX interface (only useful when P1_PHY_Mode = 0010 or 0011 0: The 100BASE-FX interface is disabled
5dt2	P1_2_PHY_MODE	0h	rh	w	Setting applies to Phy P1 only. 0000: reserved

					0001: reserved 0010: reserved 0011: 100BASE-TX/FX FD, Auto-Neg disabled 0100: reserved 0101: reserved 0110: PHY starts in power down mode 0111: Auto-Neg enabled, AutoMDIX enabled, all possible 1000: reserved 1001: reserved 1010: reserved 1011: reserved 1100: reserved 1101: reserved 1110: reserved 1111: Loopback mode. The Phy starts in loopback mode, with the 0.14 bit set. In this mode the Phy must be configured through the SMI interface and manually enabled by clearing bit 0.14. Until then it runs in loopback mode and is isolated from the line. Apart from Bit 0.14 the Phy is configured as with Phymode = 0111.
6	P1_AUTOMDIXEN	0h	rh	w	1: Enable AutoMDIX state machine 0: Disable AutoMDIX state machine
7	P1_PHY_CLK_EN	1h	rh	w	Clock signal of P1_PHY is 0: disabled 1: enabled
8	P2_PHY_ENB	0h	rh	w	0: PHY2 disabled (Powerdown Mode) see comment on Bit 0

					1: PHY2 enabled see comment on Bit 0
9	P2_FX_MODE	0h	rh	w	1: Enables 100BASE-FX interface (only useful when P2_PHY_Mode=0010 or 0011 0: The 100BASE-FX interface is disabled
13dt10	P2_PHY_MODE	0h	rh	w	Setting applies only to Phy P2. 0000: reserved 0001: reserved 0010: reserved 0011: 100BASE-TX/FX FD, Auto-Neg disabled 0100: reserved 0101: reserved 0110: PHY starts in power down mode 0111: Auto-Neg enabled, Au-toMDIX enabled, all possible 1000: reserved 1001: reserved 1010: reserved 1011: reserved 1100: reserved 1101: reserved 1110: reserved 1111: Loopback mode. The Phy starts in loopback mode, with the 0.14 bit set. In this mode the Phy must be configured through the SMI interface and manually ena-bled by clearing bit 0.14. Until then it runs in loopback mode and is isolated from the line. Apart from Bit 0.14 the Phy is configured as with Phymode = 0111.

14	P2_AUTOMDIXEN	0h	rh	w	1: Enable AutoMDIX state machine 0: Disable AutoMDIX state machine
15	P2_PHY_CLK_EN	1h	rh	w	Clock signal of P2_PHY is 0: disabled 1: enabled
31dt16	reserved	0000h			

Register:	PHY_STATUS			Address:	40h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	
Description:	Status of PHY1 and PHY2 Important: PHY_STATUS register is only reset with an asynchronous PN-IP reset					
Bit	Identifier	Reset	Attr.		Function / Description	
0	P1_PWRUPRST	0h	rh		0: PHY1 in power down mode or internal reset still active 1: PHY1 is ready	
7dt1	reserved	00h				
8	P2_PWRUPRST	0h	rh		0: PHY2 in power down mode or internal reset still active 1: PHY2 is ready	
31dt9	reserved	000000h				

Register:	AHB_BURSTBREAKER			Address:	44h	
Bits:	31dt0	Reset value:	80808h	Attributes:	r	w
Description:	Maximal number of beats in an ARM9 burst. Longer bursts are split by inserting an IDLE transfer.					
Bit	Identifier	Reset	Attr.		Function / Description	
7dt0	NR_ADDR_ARM926_D	8h	r	w	0: bypass 1-255: max. number of addresses at ARM926-D AHB port	
15dt8	NR_ADDR_ARM926_I	8h	r	w	0: bypass 1-255: max. number of addresses at ARM926-I AHB port	
23dt16	reserved	8h	r	w		

31dt24	reserved	00h			

Register:	LOCAL_OUT_READ_L			Address:	48h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	r	w
Description:	Selection of GPIO_In path					

Bit	Identifier	Reset	Attr.		Function / Description
0	GPIO32_IN_MUX	0h	r	w	0: Reads GPIO32 pad 1: Reads LOC_IO0 Out
1	GPIO33_IN_MUX	0h	r	w	0: Reads GPIO33 pad 1: Reads LOC_IO1 Out
2	GPIO34_IN_MUX	0h	r	w	0: Reads GPIO34 pad 1: Reads LOC_IO2 Out
3	GPIO35_IN_MUX	0h	r	w	0: Reads GPIO35 pad 1: Reads LOC_IO3 Out
4	GPIO36_IN_MUX	0h	r	w	0: Reads GPIO36 pad 1: Reads LOC_IO4 Out
5	GPIO37_IN_MUX	0h	r	w	0: Reads GPIO37 pad 1: Reads LOC_IO5 Out
6	GPIO38_IN_MUX	0h	r	w	0: Reads GPIO38 pad 1: Reads LOC_IO6 Out
7	GPIO39_IN_MUX	0h	r	w	0: Reads GPIO39 pad 1: Reads LOC_IO7 Out
8	GPIO40_IN_MUX	0h	r	w	0: Reads GPIO40 pad 1: Reads LOC_IO8 Out
9	GPIO41_IN_MUX	0h	r	w	0: Reads GPIO41 pad 1: Reads LOC_IO9 Out
10	GPIO42_IN_MUX	0h	r	w	0: Reads GPIO42 pad 1: Reads LOC_IO10 Out
11	GPIO43_IN_MUX	0h	r	w	0: Reads GPIO43 pad 1: Reads LOC_IO11 Out
12	GPIO44_IN_MUX	0h	r	w	0: Reads GPIO44 pad 1: Reads LOC_IO12 Out
13	GPIO45_IN_MUX	0h	r	w	0: Reads GPIO45 pad 1: Reads LOC_IO13 Out
14	GPIO46_IN_MUX	0h	r	w	0: Reads GPIO46 pad 1: Reads LOC_IO14 Out
15	GPIO47_IN_MUX	0h	r	w	0: Reads GPIO47 pad

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

					1: Reads LOC_IO15 Out
16	GPIO48_IN_MUX	0h	r	w	0: Reads GPIO48 pad 1: Reads LOC_IO16 Out
17	GPIO49_IN_MUX	0h	r	w	0: Reads GPIO49 pad 1: Reads LOC_IO17 Out
18	GPIO50_IN_MUX	0h	r	w	0: Reads GPIO50 pad 1: Reads LOC_IO18 Out
19	GPIO51_IN_MUX	0h	r	w	0: Reads GPIO51 pad 1: Reads LOC_IO19 Out
20	GPIO52_IN_MUX	0h	r	w	0: Reads GPIO52 pad 1: Reads LOC_IO20 Out
21	GPIO53_IN_MUX	0h	r	w	0: Reads GPIO53 pad 1: Reads LOC_IO21 Out
22	GPIO54_IN_MUX	0h	r	w	0: Reads GPIO54 pad 1: Reads LOC_IO22 Out
23	GPIO55_IN_MUX	0h	r	w	0: Reads GPIO55 pad 1: Reads LOC_IO23 Out
24	GPIO56_IN_MUX	0h	r	w	0: Reads GPIO56 pad 1: Reads LOC_IO24 Out
25	GPIO57_IN_MUX	0h	r	w	0: Reads GPIO57 pad 1: Reads LOC_IO25 Out
26	GPIO58_IN_MUX	0h	r	w	0: Reads GPIO58 pad 1: Reads LOC_IO26 Out
27	GPIO59_IN_MUX	0h	r	w	0: Reads GPIO59 pad 1: Reads LOC_IO27 Out
28	GPIO60_IN_MUX	0h	r	w	0: Reads GPIO60 pad 1: Reads LOC_IO28 Out
29	GPIO61_IN_MUX	0h	r	w	0: Reads GPIO61 pad 1: Reads LOC_IO29 Out
30	GPIO62_IN_MUX	0h	r	w	0: Reads GPIO62 pad 1: Reads LOC_IO30 Out
31	GPIO63_IN_MUX	0h	r	w	0: Reads GPIO63 pad 1: Reads LOC_IO31 Out

Register:	LOCAL_OUT_READ_H			Address:	4Ch	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	r	w
Description:	Selection of GPIO_In path					

Bit	Identifier	Reset	Attr.		Function / Description
0	GPIO64_IN_MUX	0h	r	w	0: Reads GPIO64 pad 1: Reads LOC_IO32 Out
1	GPIO65_IN_MUX	0h	r	w	0: Reads GPIO65 pad 1: Reads LOC_IO33 Out
2	GPIO66_IN_MUX	0h	r	w	0: Reads GPIO66 pad 1: Reads LOC_IO34 Out
3	GPIO67_IN_MUX	0h	r	w	0: Reads GPIO67 pad 1: Reads LOC_IO35 Out
4	GPIO68_IN_MUX	0h	r	w	0: Reads GPIO68 pad 1: Reads LOC_IO36 Out
5	GPIO69_IN_MUX	0h	r	w	0: Reads GPIO69 pad 1: Reads LOC_IO71 Out
6	GPIO70_IN_MUX	0h	r	w	0: Reads GPIO70 pad 1: Reads LOC_IO38 Out
7	GPIO71_IN_MUX	0h	r	w	0: Reads GPIO71 pad 1: Reads LOC_IO39 Out
8	GPIO72_IN_MUX	0h	r	w	0: Reads GPIO72 pad 1: Reads LOC_IO40 Out
9	GPIO73_IN_MUX	0h	r	w	0: Reads GPIO73 pad 1: Reads LOC_IO41 Out
10	GPIO74_IN_MUX	0h	r	w	0: Reads GPIO74 pad 1: Reads LOC_IO42 Out
11	GPIO75_IN_MUX	0h	r	w	0: Reads GPIO75 pad 1: Reads LOC_IO43 Out
12	GPIO76_IN_MUX	0h	r	w	0: Reads GPIO76 pad 1: Reads LOC_IO44 Out
13	GPIO77_IN_MUX	0h	r	w	0: Reads GPIO77 pad 1: Reads LOC_IO45 Out
14	GPIO78_IN_MUX	0h	r	w	0: Reads GPIO78 pad 1: Reads LOC_IO46 Out
15	GPIO79_IN_MUX	0h	r	w	0: Reads GPIO79 pad 1: Reads LOC_IO47 Out
16	GPIO80_IN_MUX	0h	r	w	0: Reads GPIO80 pad 1: Reads LOC_IO48 Out
17	GPIO81_IN_MUX	0h	r	w	0: Reads GPIO81 pad 1: Reads LOC_IO49 Out
18	GPIO82_IN_MUX	0h	r	w	0: Reads GPIO82 pad

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

					1: Reads LOC_IO50 Out
19	GPIO83_IN_MUX	0h	r	w	0: Reads GPIO83 pad 1: Reads LOC_IO51 Out
20	GPIO84_IN_MUX	0h	r	w	0: Reads GPIO84 pad 1: Reads LOC_IO52 Out
21	GPIO85_IN_MUX	0h	r	w	0: Reads GPIO85 pad 1: Reads LOC_IO53 Out
22	GPIO86_IN_MUX	0h	r	w	0: Reads GPIO86 pad 1: Reads LOC_IO54 Out
23	GPIO87_IN_MUX	0h	r	w	0: Reads GPIO87 pad 1: Reads LOC_IO55 Out
24	GPIO88_IN_MUX	0h	r	w	0: Reads GPIO88 pad 1: Reads LOC_IO56 Out
25	GPIO89_IN_MUX	0h	r	w	0: Reads GPIO89 pad 1: Reads LOC_IO57 Out
26	GPIO90_IN_MUX	0h	r	w	0: Reads GPIO90 pad 1: Reads LOC_IO58 Out
27	GPIO91_IN_MUX	0h	r	w	0: Reads GPIO91 pad 1: Reads LOC_IO59 Out
28	GPIO92_IN_MUX	0h	r	w	0: Reads GPIO92 pad 1: Reads LOC_IO60 Out
29	GPIO93_IN_MUX	0h	r	w	0: Reads GPIO93 pad 1: Reads LOC_IO61 Out
30	GPIO94_IN_MUX	0h	r	w	0: Reads GPIO94 pad 1: Reads LOC_IO62 Out
31	GPIO95_IN_MUX	0h	r	w	0: Reads GPIO95 pad 1: Reads LOC_IO63 Out

Register:	CCR_I2C			Address:	50h	
Bits:	31dt0	Reset value:	7Ch	Attribu-tes:	r	w
Description:	Clock Control register for the I2C_3 interface divider value for determining the bit rate.					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	CDIV_VAL	7Ch	r	w	Divider value for determining the bit rate fBR = fCLK/(CCR_I2C+1)- fCLK = fAPB=125 MHz in the I2C fBR: Bit rate clock (I2C)

					fCLK: I2C interface system cycle clock for fBR = 1MHz and fCLK=125MHz: CCR_I2C=124(dec.)
31dt8	reserved	000000h			

Register:	EDC_EVENT			Address:	54h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	rh	w
Description:	EDC event register - '0h' bits must be written with '0' to to clear. Can be written with '1' for debug purposes.					

Bit	Identifier	Reset	Attr.		Function / Description
0	I_TCM926_1B	0h	rh	w	A 1-bit error has occurred in the I-TCM of ARM926 and been corrected
1	I_TCM926_2B	0h	rh	w	A 2-bit error has occurred in the I-TCM of ARM926
2	D_TCM926_1B	0h	rh	w	A 1-bit error has occurred in the D-TCM of ARM926 and been corrected
3	D_TCM926_2B	0h	rh	w	A 2-bit error has occurred in the D-TCM of ARM926
4	GDMA_1B	0h	rh	w	A 1-bit error has occurred in the GDMA memory and been corrected
5	GDMA_2B	0h	rh	w	A 2-bit error has occurred in the GDMA memory
6	PN_1B	0h	rh	w	A 1-bit error has occurred in one of the PN-IP memories and been corrected
7	PN_2B	0h	rh	w	A 2-bit error has occurred in one of the PN-IP memories
8	PERIF_1B	0h	rh	w	A 1-bit error has occurred in the PerIF memory and been corrected
9	PERIF_2B	0h	rh	w	A 2-bit error has occurred in the PerIF memory

10	I_CACHE_PAR	0h	rh	w	A parity error has occurred in the I-cache during reading
11	D_CACHE_PAR	0h	rh	w	A parity error has occurred in the D-cache during reading
12	I_TAG_PAR	0h	rh	w	A parity error has occurred in the I-Tag RAM during reading
13	D_TAG_PAR	0h	rh	w	A parity error has occurred in the D-Tag RAM during reading
14	reserved	0h	rh	w	
15	reserved	0h	rh	w	
16	reserved	0h	rh	w	
17	reserved	0h	rh	w	
18	reserved	0h	rh	w	
19	reserved	0h	rh	w	
20	SPI_PFU_1B	0h	rh	w	A 1-bit error has occurred in the SPI-PFU memory and been corrected
21	SPI_PFU_2B	0h	rh	w	A 2-bit error has occurred in the SPI-PFU memory
22	reserved	0h	rh	w	
23	reserved	0h	rh	w	
24	I_TCM926_2B_RFSH	0h	rh	w	A 2-bit error has occurred in the I-TCM of ARM926 memory during refresh if I_TMC926_2B = '0'
25	D_TCM926_2B_RFSH	0h	rh	w	A 2-bit error has occurred in the D-TCM of ARM926 memory during refresh if D_TMC926_2B = '0'
26	PN_2B_RFSH	0h	rh	w	A 2-bit error has occurred in the PN-IP memory during refresh if PN_2B = '0'
27	reserved	0h	rh	w	
28	reserved	0h	rh	w	
29	reserved	0h	rh	w	
30	reserved	0h			
31	EMC	0h	rh	w	A 2-bit error has occurred in the memory connected to EMC during refresh.

Register:	EDC_INIT_DONE			Address:	58h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	rh	
Description:	EDC Init Done register - the status can be read					

Bit	Identifier	Reset	Attr.		Function / Description
0	I_TCM926_INIT_DONE	0h	rh		1: Initialization of the EDC bits in the I-TCM of ARM926 is complete
1	D_TCM926_INIT_DONE	0h	rh		1: Initialization of the EDC bits in the D-TCM of ARM926 is complete
2	GDMA_INIT_DONE	0h	rh		1: Initialization of the EDC bits in the GDMA is complete
3	PN_IP_INIT_DONE	0h	rh		1: Initialization of the EDC bits in the PN_IP is complete
4	PERIF_INIT_DONE	0h	rh		1: Initialization of the EDC bits in PER-IF is complete
5	reserved	0h	rh		
6	reserved	0h	rh		
7	reserved	0h	rh		
31dt8	reserved	000000h			

Register:	TCM926_MAP			Address:	5Ch	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	r	w
Description:	TCM926 Map Register Division into I / D-TCM and INITRAM					

Bit	Identifier	Reset	Attr.		Function / Description
2dt0	PARTITION_TCM926	0h	r	w	I-TCM / D-TCM ARM926 000b: 0 KB / 256 KB 001b: 64 KB / 192 KB 010b: 128 KB / 128 KB 011b: 192 KB / 64 KB 100b: 256 KB / 0 KB
3	INITRAM	0h	r	w	Booting I-TCM 0: I-TCM not booted. Booting in accordance with MEM_SWAP configuration.

					1: I-TCM is activated after ARM926 core reset
31dt4	reserved	0000000h			

Register:	GPIO_INT_POLSEL			Address:	60h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	r	w
Description:	Interrupt polarity for GPIO interrupts (15:0)					

Bit	Identifier	Reset	Attr.		Function / Description
0	INT_POLSEL_GPIO0	0h	r	w	0: GPIO0 is not inverted from the A-ICU (IRQ32) and GDMA 1: GPIO0 is inverted from the A-ICU (IRQ32) and GDMA
1	INT_POLSEL_GPIO1	0h	r	w	0: GPIO1 is not inverted from the A-ICU (IRQ33) and GDMA 1: GPIO1 is inverted from the A-ICU (IRQ33) and GDMA
2	INT_POLSEL_GPIO2	0h	r	w	0: GPIO2 is not inverted from the A-ICU (IRQ34) and GDMA 1: GPIO2 is inverted from the A-ICU (IRQ34) and GDMA
3	INT_POLSEL_GPIO3	0h	r	w	0: GPIO3 is not inverted from the A-ICU (IRQ35) and GDMA 1: GPIO3 is inverted from the A-ICU (IRQ35) and GDMA
4	INT_POLSEL_GPIO4	0h	r	w	0: GPIO4 is not inverted from the A-ICU (IRQ36) and PN-ICU 1 1: GPIO4 is inverted from the A-ICU (IRQ36) and PN-ICU 1
5	INT_POLSEL_GPIO5	0h	r	w	0: GPIO5 is not inverted from the A-ICU (IRQ37) and PN-ICU 1 1: GPIO5 is inverted from the A-ICU (IRQ37) and PN-ICU 1
6	INT_POLSEL_GPIO6	0h	r	w	0: GPIO6 is not inverted from the A-ICU (IRQ38) and PN-ICU 1 1: GPIO6 is inverted from the A-ICU (IRQ38) and PN-ICU 1
7	INT_POLSEL_GPIO7	0h	r	w	0: GPIO7 is not inverted from the A-ICU (IRQ39) and PN-ICU 1 1: GPIO7 is inverted from the A-ICU (IRQ39) and PN-ICU 1
8	INT_POLSEL_GPIO8	0h	r	w	0: GPIO8 is not inverted from the A-ICU (IRQ40) 1: GPIO8 is inverted from the A-ICU (IRQ40)

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

9	INT_POLSEL_GPIO9	0h	r	w	0: GPIO9 is not inverted from the A-ICU (IRQ41) 1: GPIO9 is inverted from the A-ICU (IRQ41)
10	INT_POLSEL_GPIO10	0h	r	w	0: GPIO10 is not inverted from the A-ICU (IRQ42) 1: GPIO10 is inverted from the A-ICU (IRQ42)
11	INT_POLSEL_GPIO11	0h	r	w	0: GPIO11 is not inverted from the A-ICU (IRQ43) 1: GPIO11 is inverted from the A-ICU (IRQ43)
12	INT_POLSEL_GPIO12	0h	r	w	0: GPIO12 is not inverted from the A-ICU (IRQ44) 1: GPIO12 is inverted from the A-ICU (IRQ44)
13	INT_POLSEL_GPIO13	0h	r	w	0: GPIO13 is not inverted from the A-ICU (IRQ45) 1: GPIO13 is inverted from the A-ICU (IRQ45)
14	INT_POLSEL_GPIO14	0h	r	w	0: GPIO14 is not inverted from the A-ICU (IRQ46) 1: GPIO14 is inverted from the A-ICU (IRQ46)
15	INT_POLSEL_GPIO15	0h	r	w	0: GPIO15 is not inverted from the A-ICU (IRQ47) 1: GPIO15 is inverted from the A-ICU (IRQ47)
31dt16	reserved	0000h			

Register:	EDC_PARITY_EN			Address:	64h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	r	w
Description:	EDC- and Parity Enable (2:0)					

Bit	Identifier	Reset	Attr.		Function / Description
0	I_CACHE_PAR_EN	0h	r	w	0: Parity logic for ARM926 I-Cache / I-Tag is disabled (default) 1: Parity logic for ARM926 I-Cache / I-Tag is enabled
1	D_CACHE_PAR_EN	0h	r	w	0: Parity logic for ARM926 D-Cache / D-Tag is disabled (default) 1: Parity logic for ARM926 D-Cache / D-Tag is enabled

2	EDC_DISABLE_ARM926	0h	r	w	0: Die EDC logic in ARM926 I- / D-TCM is enabled (default) 1: Die EDC logic in ARM926 I- / D-TCM is disabled
3	reserved	0h	r	w	
31dt4	reserved	0000000h			

Register:	MODUL_ACCESS_ERR			Address:	68h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	rh	w
Description:	Module Access Error register (5:0) - '0'h must be written to the register to clear					
Bit	Identifier	Reset	Attr.		Function / Description	
0	PN_IP_ACCESS_ERR	0h	rh	w	0: No access error has occurred in the PN-IP 1: An access error has occurred in the PN-IP	
1	PER_IF_ACCESS_ERR	0h	rh	w	0: No access error has occurred in the PERIF 1: An access error has occurred in the PERIF	
2	I_FILTER_ACCESS_ERR	0h	rh	w	0: No access error has occurred in the I-Filter 1: An access error has occurred in the I-Filter	
3	HOST_IF_ACCESS_ERR	0h	rh	w	0: No access error has occurred in the HOST IF 1: An access error has occurred in the HOST IF	
4	SCRB_ACCESS_ERR	0h	rh	w	0: No access error has occurred in the SCRB 1: An access error has occurred in the SCRB	
5	reserved	0h	rh	w		
6	RSC1_ACCESS_ERR	0h	rh	w	0: No access error has occurred in the RS controller 1 1: An access error has occurred in the RS controller 1	
7	RSC2_ACCESS_ERR	0h	rh	w	0: No access error has occurred in the RS controller 2 1: An access error has occurred in the RS controller 2	
8	reserved	0h	rh	w		

9	SPI_PFU_ACCESS_ERR	0h	rh	w	0: No access error has occurred in the SPI_PFU 1: An access error has occurred in the SPI_PFU
10	reserved	0h	rh	w	
31dt11	reserved	000000h			

Register:	RES_SOFT_RETURN_ADDR			Address:	6Ch	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	r	w
Description:	Vector address for the secondary boot loader following an ARM926EJS SW reset					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	RETURN_ADDRESS	0h	r	w	Vector address for the secondary boot loader following RES_SOFT_ARM926_CORE. This SW reset is executed in the TCM926_MAP register after TCM 926 configuration.	

Register:	PHY_LED_CONTROL			Address:	70h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	(r)(h)	(w)
Description:	PHY-LED control					
Bit	Identifier	Reset	Attr.		Function / Description	
0	EN_P1_XLINK_STATUS	0h	r	w	0: LED: P1_XLINK_STATUS is controlled by the PHY 1: LED: P1_XLINK_STATUS is controlled by the SW	
1	EN_P1_XACTIVITY	0h	r	w	0: LED: P1_XLINK_XACTIVITY is controlled by the PHY 1: LED: P1_XLINK_XACTIVITY is controlled by the SW	
2	EN_P2_XLINK_STATUS	0h	r	w	0: LED: P2_XLINK_STATUS is controlled by the PHY 1: LED: P2_XLINK_STATUS is controlled by the SW	
3	EN_P2_XACTIVITY	0h	r	w	0: LED: P2_XLINK_XACTIVITY is controlled by the PHY 1: LED: P2_XLINK_XACTIVITY is controlled by the SW	

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

7dt4	reserved	0h			
8	P1_XLINK_STATUS_SW	0h	r	w	0: LED: P1_XLINK_STATUS is '0', set by the SW 1: LED: P1_XLINK_STATUS is '1', set by the SW
9	P1_XACTIVITY_SW	0h	r	w	0: LED: P1_XLINK_XACTIVITY is '0', set by the SW 1: LED: P1_XLINK_XACTIVITY is '1', set by the SW
10	P2_XLINK_STATUS_SW	0h	r	w	0: LED: P2_XLINK_STATUS is '0', set by the SW 1: LED: P2_XLINK_STATUS is '1', set by the SW
11	P2_XACTIVITY_SW	0h	r	w	0: LED: P2_XLINK_XACTIVITY is '0', set by the SW 1: LED: P2_XLINK_XACTIVITY is '1', set by the SW
15dt12	reserved	0h			
16	P1_XLINK_STATUS_PHY	0h	rh		P1_XLINK_STATUS controlled by the PHY
17	P1_XACTIVITY_PHY	0h	rh		P1_XACTIVITY controlled by the PHY
18	P2_XLINK_STATUS_PHY	0h	rh		P2_XLINK_STATUS controlled by the PHY
19	P2_XACTIVITY_PHY	0h	rh		P2_XACTIVITY controlled by the PHY
31dt20	reserved	000h			

Register:	ACCESS_ERROR_SCRB				Address:	74h	
Bits:	31dt0	Reset value:		20000000h	Attribu-tes:	(r)(h)	(w)
Description:	SCRB Access Error						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	APB_ADDRESS	0h	rh	w	Erroneous APB address in SCRB		
27dt8	reserved	00000h					
29dt28	APB_SIZE	2h	r		Clamped at 10 (word access) as only word access is permitted		
30	APB_WRITE	0h	rh	w	0: Read access 1: Write access		
31	ERR_LOCK	0h	rh	w	Set to '1' by the HW if erroneous SCRB access is detected. Further HW entries are then blocked. The SW must reset the bit to '0' to enable new entries		

Register:	DRIVE_EMC			Address:	78h	
Bits:	31dt0	Reset value:	C3FFFFh	Attribu-tes:	r	w
Description:	SCRB Drive Current of Dedicated Signals (1.8 V) Each GPIO-Bit is set to the drive current based on Coding of the fields.					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	DR_EMC_C	3h	r	w	Signal list: DTXR, XOE_DDRIVER 00 - 1 mA 01 - 2 mA 10 - 4 mA 11 - 6 mA
3dt2	DR_EMC_AL	3h	r	w	Signal list: A14 - A0 00 - 2 mA 01 - 4 mA 10 - 6 mA 11 - 8 mA
5dt4	DR_EMC_AH	3h	r	w	Signal list: A23 - A15 00 - 1 mA 01 - 2 mA 10 - 4 mA 11 - 6 mA
7dt6	DR_EMC_DL	3h	r	w	Signal list: D15 - D0, XBE0_DQM0, XBE1_DQM1, XRDY_PER 00 - 2 mA 01 - 4 mA 10 - 6 mA 11 - 8 mA
9dt8	DR_EMC_DH	3h	r	w	Signal list: D31 - D16, XBE2_DQM2, XBE3_DQM3 00 - 2 mA 01 - 4 mA 10 - 6 mA 11 - 8 mA
11dt10	DR_EMC_RW	3h	r	w	Signal list: XWR, XRD 00 - 2 mA 01 - 4 mA 10 - 6 mA

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

					11 - 8 mA
13dt12	DR_EMC_PER	3h	r	w	Signal list: XCS_PER0 - 3 00 - 1 mA 01 - 2 mA 10 - 4 mA 11 - 6 mA
15dt14	DR_EMC_CLK_SDRAM	3h	r	w	Signal list:CLK_O_SDRAM0, CLK_O_SDRAM1, CLK_O_SDRAM2 00 - 2 mA 01 - 4 mA 10 - 6 mA 11 - 8 mA
17dt16	DR_EMC_SDRAM	3h	r	w	Signal list: XCS_SDRAM, XRAS_SDRAM, XCAS_SDRAM, XWE_SDRAM 00 - 2 mA 01 - 4 mA 10 - 6 mA 11 - 8 mA
21dt18	reserved	0h			
22	EMC_SDRAM_CLK_SD0SD1_EBL	1h	r	w	enable(=1) / disable(=0) CLK_O_SDRAM0 and CLK_O_SDRAM1 (one SDRAM block)
23	EMC_SDRAM_CLK_SD2_EBL	1h	r	w	enable(=1) / disable(=0) CLK_O_SDRAM2 (with 2nd SDRAM block)
31dt24	reserved	00h			

Register:	DRIVE15_0GPIO			Address:	7Ch	
Bits:	31dt0	Reset value:	AAAAAAAAh	Attribu-tes:	r	w
Description:	SCRB Drive Current of GPIO Signals Each GPIO bit is set to the drive current on the basis of the following coding 00 - 2 mA 01 - 4 mA 10 - 8 mA 11 - 12 mA					
Bit	Identifier	Reset	Attr.	Function / Description		

1dt0	DR_GPIO0	2h	r	w	drive current GPIO0
3dt2	DR_GPIO1	2h	r	w	drive current GPIO1
5dt4	DR_GPIO2	2h	r	w	drive current GPIO2
7dt6	DR_GPIO3	2h	r	w	drive current GPIO3
9dt8	DR_GPIO4	2h	r	w	drive current GPIO4
11dt10	DR_GPIO5	2h	r	w	drive current GPIO5
13dt12	DR_GPIO6	2h	r	w	drive current GPIO6
15dt14	DR_GPIO7	2h	r	w	drive current GPIO7
17dt16	DR_GPIO8	2h	r	w	drive current GPIO8
19dt18	DR_GPIO9	2h	r	w	drive current GPIO9
21dt20	DR_GPIO10	2h	r	w	drive current GPIO10
23dt22	DR_GPIO11	2h	r	w	drive current GPIO11
25dt24	DR_GPIO12	2h	r	w	drive current GPIO12
27dt26	DR_GPIO13	2h	r	w	drive current GPIO13
29dt28	DR_GPIO14	2h	r	w	drive current GPIO14
31dt30	DR_GPIO15	2h	r	w	drive current GPIO15

Register:	DRIVE31_16GPIO			Address:	80h	
Bits:	31dt0	Reset value:	AAAAAAAAh	Attribu-tes:	r	w
Description:	SCRB Drive Current of GPIO Signals Each GPIO bit is set to the drive current on the basis of the following coding 00 - 2 mA 01 - 4 mA 10 - 8 mA 11 - 12 mA					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	DR_GPIO16	2h	r	w	drive current GPIO16
3dt2	DR_GPIO17	2h	r	w	drive current GPIO17
5dt4	DR_GPIO18	2h	r	w	drive current GPIO18
7dt6	DR_GPIO19	2h	r	w	drive current GPIO19
9dt8	DR_GPIO20	2h	r	w	drive current GPIO20
11dt10	DR_GPIO21	2h	r	w	drive current GPIO21
13dt12	DR_GPIO22	2h	r	w	drive current GPIO22
15dt14	DR_GPIO23	2h	r	w	drive current GPIO23

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

17dt16	DR_GPIO24	2h	r	w	drive current GPIO24
19dt18	DR_GPIO25	2h	r	w	drive current GPIO25
21dt20	DR_GPIO26	2h	r	w	drive current GPIO26
23dt22	DR_GPIO27	2h	r	w	drive current GPIO27
25dt24	DR_GPIO28	2h	r	w	drive current GPIO28
27dt26	DR_GPIO29	2h	r	w	drive current GPIO29
29dt28	DR_GPIO30	2h	r	w	drive current GPIO30
31dt30	DR_GPIO31	2h	r	w	drive current GPIO31

Register:	DRIVE47_32GPIO				Address:	84h
Bits:	31dt0	Reset value:	AAAAAAAAh		Attribu-tes:	r w
Description:	SCRB Drive Current of GPIO Signals (3.3V / 1.8V) Each GPIO bit is set to the drive current on the basis of the following coding ---1.8V--- | ---3.3V--- 00 - 1 mA | 00 - 2 mA 01 - 2 mA | 01 - 4 mA 10 - 4 mA | 10 - 8 mA 11 - 6 mA | 11 - 12 mA					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	DR_GPIO32	2h	r	w	drive current GPIO32
3dt2	DR_GPIO33	2h	r	w	drive current GPIO33
5dt4	DR_GPIO34	2h	r	w	drive current GPIO34
7dt6	DR_GPIO35	2h	r	w	drive current GPIO35
9dt8	DR_GPIO36	2h	r	w	drive current GPIO36
11dt10	DR_GPIO37	2h	r	w	drive current GPIO37
13dt12	DR_GPIO38	2h	r	w	drive current GPIO38
15dt14	DR_GPIO39	2h	r	w	drive current GPIO39
17dt16	DR_GPIO40	2h	r	w	drive current GPIO40
19dt18	DR_GPIO41	2h	r	w	drive current GPIO41
21dt20	DR_GPIO42	2h	r	w	drive current GPIO42
23dt22	DR_GPIO43	2h	r	w	drive current GPIO43
25dt24	DR_GPIO44	2h	r	w	drive current GPIO44
27dt26	DR_GPIO45	2h	r	w	drive current GPIO45
29dt28	DR_GPIO46	2h	r	w	drive current GPIO46
31dt30	DR_GPIO47	2h	r	w	drive current GPIO47

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Register:	DRIVE63_48GPIO			Address:	88h	
Bits:	31dt0	Reset value:	AAAAAAAAh	Attribu-tes:	r	w
Description:	SCRB Drive Current of GPIO Signals (3.3V / 1.8V) Each GPIO bit is set to the drive current on the basis of the following coding ---1.8V--- | ---3.3V--- 00 - 1 mA | 00 - 2 mA 01 - 2 mA | 01 - 4 mA 10 - 4 mA | 10 - 8 mA 11 - 6 mA | 11 - 12 mA					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	DR_GPIO48	2h	r	w	drive current GPIO48
3dt2	DR_GPIO49	2h	r	w	drive current GPIO49
5dt4	DR_GPIO50	2h	r	w	drive current GPIO50
7dt6	DR_GPIO51	2h	r	w	drive current GPIO51
9dt8	DR_GPIO52	2h	r	w	drive current GPIO52
11dt10	DR_GPIO53	2h	r	w	drive current GPIO53
13dt12	DR_GPIO54	2h	r	w	drive current GPIO54
15dt14	DR_GPIO55	2h	r	w	drive current GPIO55
17dt16	DR_GPIO56	2h	r	w	drive current GPIO56
19dt18	DR_GPIO57	2h	r	w	drive current GPIO57
21dt20	DR_GPIO58	2h	r	w	drive current GPIO58
23dt22	DR_GPIO59	2h	r	w	drive current GPIO59
25dt24	DR_GPIO60	2h	r	w	drive current GPIO60
27dt26	DR_GPIO61	2h	r	w	drive current GPIO61
29dt28	DR_GPIO62	2h	r	w	drive current GPIO62
31dt30	DR_GPIO63	2h	r	w	drive current GPIO63

Register:	DRIVE79_64GPIO			Address:	8Ch	
Bits:	31dt0	Reset value:	AAAAAAAAh	Attribu-tes:	r	w
Description:	SCRB Drive Current of GPIO Signals (3.3V / 1.8V) Each GPIO bit is set to the drive current on the basis of the following coding ---1.8V--- | ---3.3V--- 00 - 1 mA | 00 - 2 mA 01 - 2 mA | 01 - 4 mA 10 - 4 mA | 10 - 8 mA					

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

		11 - 6 mA | 11 - 12 mA			
Bit	Identifier	Reset	Attr.		Function / Description
1dt0	DR_GPIO64	2h	r	w	drive current GPIO64
3dt2	DR_GPIO65	2h	r	w	drive current GPIO65
5dt4	DR_GPIO66	2h	r	w	drive current GPIO66
7dt6	DR_GPIO67	2h	r	w	drive current GPIO67
9dt8	DR_GPIO68	2h	r	w	drive current GPIO68
11dt10	DR_GPIO69	2h	r	w	drive current GPIO69
13dt12	DR_GPIO70	2h	r	w	drive current GPIO70
15dt14	DR_GPIO71	2h	r	w	drive current GPIO71
17dt16	DR_GPIO72	2h	r	w	drive current GPIO72
19dt18	DR_GPIO73	2h	r	w	drive current GPIO73
21dt20	DR_GPIO74	2h	r	w	drive current GPIO74
23dt22	DR_GPIO75	2h	r	w	drive current GPIO75
25dt24	DR_GPIO76	2h	r	w	drive current GPIO76
27dt26	DR_GPIO77	2h	r	w	drive current GPIO77
29dt28	DR_GPIO78	2h	r	w	drive current GPIO78
31dt30	DR_GPIO79	2h	r	w	drive current GPIO79

Register:	DRIVE95_80GPIO			Address:	90h	
Bits:	31dt0	Reset value:	AAAAAAAAh	Attribu-tes:	r	w
Description:	SCRB Drive Current of GPIO Signals (3.3V / 1.8V) Each GPIO bit is set to the drive current on the basis of the following coding ---1.8V--- | ---3.3V--- 00 - 1 mA | 00 - 2 mA 01 - 2 mA | 01 - 4 mA 10 - 4 mA | 10 - 8 mA 11 - 6 mA | 11 - 12 mA					
Bit	Identifier	Reset	Attr.		Function / Description	
1dt0	DR_GPIO80	2h	r	w	drive current GPIO80	
3dt2	DR_GPIO81	2h	r	w	drive current GPIO81	
5dt4	DR_GPIO82	2h	r	w	drive current GPIO82	
7dt6	DR_GPIO83	2h	r	w	drive current GPIO83	
9dt8	DR_GPIO84	2h	r	w	drive current GPIO84	
11dt10	DR_GPIO85	2h	r	w	drive current GPIO85	

13dt12	DR_GPIO86	2h	r	w	drive current GPIO86
15dt14	DR_GPIO87	2h	r	w	drive current GPIO87
17dt16	DR_GPIO88	2h	r	w	drive current GPIO88
19dt18	DR_GPIO89	2h	r	w	drive current GPIO89
21dt20	DR_GPIO90	2h	r	w	drive current GPIO90
23dt22	DR_GPIO91	2h	r	w	drive current GPIO91
25dt24	DR_GPIO92	2h	r	w	drive current GPIO92
27dt26	DR_GPIO93	2h	r	w	drive current GPIO93
29dt28	DR_GPIO94	2h	r	w	drive current GPIO94
31dt30	DR_GPIO95	2h	r	w	ERTEC-Mode: DR_GPIO95 drive current GPIO95

Register:	PULL15_0GPIO			Address:	94h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	rh	w
Description:	SCRB Pull GPIO Signals Each GPIO bit is set to the pull resistors on the basis of the following coding 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistors are automatically deactivated Reset value does not apply as the register is loaded in accordance with the function set with CONFIG.					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	PR_GPIO0	0h	rh	w	pull control GPIO0
3dt2	PR_GPIO1	0h	rh	w	pull control GPIO1
5dt4	PR_GPIO2	0h	rh	w	pull control GPIO2
7dt6	PR_GPIO3	0h	rh	w	pull control GPIO3
9dt8	PR_GPIO4	0h	rh	w	pull control GPIO4
11dt10	PR_GPIO5	0h	rh	w	pull control GPIO5
13dt12	PR_GPIO6	0h	rh	w	pull control GPIO6
15dt14	PR_GPIO7	0h	rh	w	pull control GPIO7
17dt16	PR_GPIO8	0h	rh	w	pull control GPIO8

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

19dt18	PR_GPIO9	0h	rh	w	pull control GPIO9
21dt20	PR_GPIO10	0h	rh	w	pull control GPIO10
23dt22	PR_GPIO11	0h	rh	w	pull control GPIO11
25dt24	PR_GPIO12	0h	rh	w	pull control GPIO12
27dt26	PR_GPIO13	0h	rh	w	pull control GPIO13
29dt28	PR_GPIO14	0h	rh	w	pull control GPIO14
31dt30	PR_GPIO15	0h	rh	w	pull control GPIO15

Register:	PULL31_16GPIO			Address:	98h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	rh	w
Description:	SCRB Pull GPIO Signals Each GPIO bit is set to the pull resistors on the basis of the following coding 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistors are auto-matically deactivated Reset value does not apply as the register is loaded in accord-ance with the function set with CONFIG.					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	PR_GPIO16	0h	rh	w	pull control GPIO16
3dt2	PR_GPIO17	0h	rh	w	pull control GPIO17
5dt4	PR_GPIO18	0h	rh	w	pull control GPIO18
7dt6	PR_GPIO19	0h	rh	w	pull control GPIO19
9dt8	PR_GPIO20	0h	rh	w	pull control GPIO20
11dt10	PR_GPIO21	0h	rh	w	pull control GPIO21
13dt12	PR_GPIO22	0h	rh	w	pull control GPIO22
15dt14	PR_GPIO23	0h	rh	w	pull control GPIO23
17dt16	PR_GPIO24	0h	rh	w	pull control GPIO24
19dt18	PR_GPIO25	0h	rh	w	pull control GPIO25
21dt20	PR_GPIO26	0h	rh	w	pull control GPIO26
23dt22	PR_GPIO27	0h	rh	w	pull control GPIO27
25dt24	PR_GPIO28	0h	rh	w	pull control GPIO28
27dt26	PR_GPIO29	0h	rh	w	pull control GPIO29

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

29dt28	PR_GPIO30	0h	rh	w	pull control GPIO30
31dt30	PR_GPIO31	0h	rh	w	pull control GPIO31

Register:	PULL47_32GPIO			Address:	9Ch	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	rh	w
Description:	SCRB Pull GPIO Signals Each GPIO bit is set to the pull resistors on the basis of the following coding 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistors are automatically deactivated Reset value does not apply as the register is loaded in accordance with the function set with CONFIG.					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	PR_GPIO32	0h	rh	w	pull control GPIO32
3dt2	PR_GPIO33	0h	rh	w	pull control GPIO33
5dt4	PR_GPIO34	0h	rh	w	pull control GPIO34
7dt6	PR_GPIO35	0h	rh	w	pull control GPIO35
9dt8	PR_GPIO36	0h	rh	w	pull control GPIO36
11dt10	PR_GPIO37	0h	rh	w	pull control GPIO37
13dt12	PR_GPIO38	0h	rh	w	pull control GPIO38
15dt14	PR_GPIO39	0h	rh	w	pull control GPIO39
17dt16	PR_GPIO40	0h	rh	w	pull control GPIO40
19dt18	PR_GPIO41	0h	rh	w	pull control GPIO41
21dt20	PR_GPIO42	0h	rh	w	pull control GPIO42
23dt22	PR_GPIO43	0h	rh	w	pull control GPIO43
25dt24	PR_GPIO44	0h	rh	w	pull control GPIO44
27dt26	PR_GPIO45	0h	rh	w	pull control GPIO45
29dt28	PR_GPIO46	0h	rh	w	pull control GPIO46
31dt30	PR_GPIO47	0h	rh	w	pull control GPIO47

Register:	PULL63_48GPIO	Address:	A0h

Bits:	31dt0	Reset value:	0h	Attribu-tes:	rh	w
Description:	SCRB Pull GPIO Signals Each GPIO bit is set to the pull resistors on the basis of the following coding 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistors are automatically deactivated Reset value does not apply as the register is loaded in accordance with the function set with CONFIG.					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	PR_GPIO48	0h	rh	w	pull control GPIO48
3dt2	PR_GPIO49	0h	rh	w	pull control GPIO49
5dt4	PR_GPIO50	0h	rh	w	pull control GPIO50
7dt6	PR_GPIO51	0h	rh	w	pull control GPIO51
9dt8	PR_GPIO52	0h	rh	w	pull control GPIO52
11dt10	PR_GPIO53	0h	rh	w	pull control GPIO53
13dt12	PR_GPIO54	0h	rh	w	pull control GPIO54
15dt14	PR_GPIO55	0h	rh	w	pull control GPIO55
17dt16	PR_GPIO56	0h	rh	w	pull control GPIO56
19dt18	PR_GPIO57	0h	rh	w	pull control GPIO57
21dt20	PR_GPIO58	0h	rh	w	pull control GPIO58
23dt22	PR_GPIO59	0h	rh	w	pull control GPIO59
25dt24	PR_GPIO60	0h	rh	w	pull control GPIO60
27dt26	PR_GPIO61	0h	rh	w	pull control GPIO61
29dt28	PR_GPIO62	0h	rh	w	pull control GPIO62
31dt30	PR_GPIO63	0h	rh	w	pull control GPIO63

Register:	PULL79_64GPIO			Address:	A4h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	rh	w
Description:	SCRB Pull GPIO Signals Each GPIO bit is set to the pull resistors on the basis of the following coding 00 - highZ					

	01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistors are automatically deactivated Reset value does not apply as the register is loaded in accordance with the function set with CONFIG.

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	PR_GPIO64	0h	rh	w	pull control GPIO64
3dt2	PR_GPIO65	0h	rh	w	pull control GPIO65
5dt4	PR_GPIO66	0h	rh	w	pull control GPIO66
7dt6	PR_GPIO67	0h	rh	w	pull control GPIO67
9dt8	PR_GPIO68	0h	rh	w	pull control GPIO68
11dt10	PR_GPIO69	0h	rh	w	pull control GPIO69
13dt12	PR_GPIO70	0h	rh	w	pull control GPIO70
15dt14	PR_GPIO71	0h	rh	w	pull control GPIO71
17dt16	PR_GPIO72	0h	rh	w	pull control GPIO72
19dt18	PR_GPIO73	0h	rh	w	pull control GPIO73
21dt20	PR_GPIO74	0h	rh	w	pull control GPIO74
23dt22	PR_GPIO75	0h	rh	w	pull control GPIO75
25dt24	PR_GPIO76	0h	rh	w	pull control GPIO76
27dt26	PR_GPIO77	0h	rh	w	pull control GPIO77
29dt28	PR_GPIO78	0h	rh	w	pull control GPIO78
31dt30	PR_GPIO79	0h	rh	w	pull control GPIO79

Register:	PULL95_80GPIO			Address:	A8h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	SCRB Pull GPIO Signals Each GPIO bit is set to the pull resistors on the basis of the following coding 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistors are automatically					

		deactivated Reset value does not apply as the register is loaded in accordance with the function set with CONFIG.			
Bit	Identifier	Reset	Attr.		Function / Description
1dt0	PR_GPIO80	0h	rh	w	pull control GPIO80
3dt2	PR_GPIO81	0h	rh	w	pull control GPIO81
5dt4	PR_GPIO82	0h	rh	w	pull control GPIO82
7dt6	PR_GPIO83	0h	rh	w	pull control GPIO83
9dt8	PR_GPIO84	0h	rh	w	pull control GPIO84
11dt10	PR_GPIO85	0h	rh	w	pull control GPIO85
13dt12	PR_GPIO86	0h	rh	w	pull control GPIO86
15dt14	PR_GPIO87	0h	rh	w	pull control GPIO87
17dt16	PR_GPIO88	0h	rh	w	pull control GPIO88
19dt18	PR_GPIO89	0h	rh	w	pull control GPIO89
21dt20	PR_GPIO90	0h	rh	w	pull control GPIO90
23dt22	PR_GPIO91	0h	rh	w	pull control GPIO91
25dt24	PR_GPIO92	0h	rh	w	pull control GPIO92
27dt26	PR_GPIO93	0h	rh	w	pull control GPIO93
29dt28	PR_GPIO94	0h	rh	w	pull control GPIO94
31dt30	PR_GPIO95	0h	rh	w	pull control GPIO95

Register:	CLEAR_DMA_SPI_REQ_REG			Address:	ACh	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Reset SPI1/2_SSPTX_Delayed_ Request (SPI1/2 pause mode)					
Bit	Identifier	Reset	Attr.		Function / Description	
0	CLEAR_SPI1_SSPTX_DELAYED_REQUEST	0h	rh	w	1: SPI1_SSPTX_Delayed_ Request reset in GDMA shell, non-retentive	
1	CLEAR_SPI2_SSPTX_DELAYED_REQUEST	0h	rh	w	1: SPI2_SSPTX_Delayed_ Request reset in GDMA shell, non-retentive	
31dt2	reserved	00000000h				

Register:	SPI_PARITY_ERROR			Address:	B0h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w

Description:	SPI1/2 parity error (there is only a parity check when SPI data are read in the address range with APB address bits(10dt9) = 01)				
Bit	Identifier	Reset	Attr.		Function / Description
0	SPI1_PARITY_ERR	0h	rh	w	1: Parity-Error SPI1 (Even Parity)
1	SPI2_PARITY_ERR	0h	rh	w	1: Parity-Error SPI2 (Even Parity)
31dt2	reserved	00000000h			

Register:	XHIF_MODE			Address:	B8h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	r	w
Description:	Switch XHIF-Pin between XHIF_A20 und XHIF_XCS_R					
Bit	Identifier	Reset	Attr.		Function / Description	
0	XHIF_MODE	0h	r	w	The input pin XHIF_XCS_R_A20 is used as 0: Page register chip select (XHIF_XCS_R) 1: Address line (XHIF_A20) (The XHIF input not used is switched to inactive, i.e. XHIF_A20='0', XHIF_XCS_R='1')	
31dt1	reserved	00000000h				

Register:	EXT_DRIVER_EN			Address:	BCh	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	r(h)	w
Description:						
Bit	Identifier	Reset	Attr.		Function / Description	
0	CS0_ENABLE	0h	rh	w	1: CS0 activates XOE_DRIVER 0: CS0 has no effect on XOE_DRIVER When the reset is cleared, the inverted value of XRDY_BF is latched.	
1	CS1_ENABLE	0h	r	w	1: CS1 activates XOE_DRIVER 0: CS1 has no effect on XOE_DRIVER.	
2	CS2_ENABLE	0h	r	w	1: CS2 activates XOE_DRIVER 0: CS2 has no effect on XOE_DRIVER	
3	CS3_ENABLE	0h	r	w	1: CS3 activates XOE_DRIVER	

					0: CS3 has no effect on XOE_DRIVER
31dt4	reserved	0000000h			

Register:	SPI_MODE			Address:	C0h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	r	w
Description:	Toggling SPI outputs between direct mode and MISO_MOSI mode					
Bit	Identifier	Reset	Attr.		Function / Description	
0	SPI1_MODE	0h	r	w	0: direct-Mode for SPI1 1: MISO_MOSI-Mode for SPI1	
1	SPI2_MODE	0h	r	w	0: direct-Mode for SPI2 1: MISO_MOSI-Mode for SPI2	
31dt2	reserved	00000000h				

Register:	SD_CONTROL			Address:	C4h	
Bits:	31dt0	Reset value:	0h	Attribu-tes:	r(h)	w
Description:	Differentiating and filtering the SignalDetect signal (SD signal) of an external fiber-optic transceiver for the integrated PHY over the ERTEC die					
Bit	Identifier	Reset	Attr.		Function / Description	
0	SD1_ENABLE	0h	r	w	Differentiation/filtering of the SD1 signal for port 1 is 0: direct, over the balls for the integrated PHY 1: over ERTEC die for the integrated PHY	
7dt1	reserved	00h				
14dt8	SD1_MUX	0h	r	w	only relevant when SD1_ENABLE = '1'0...95: GPIO pin 0...95 over which the SD1 signal of the fiber-optic transceiver is received. 96...112, no GPIO selection, input signal of '0' is assumed 113...127,	

					no GPIO selection, input signal of '1' is assumed
15	SD1_SELECT	0h	rh	w	the source for I-Filter input n is: 0: GPIOx set in SD1_MUX 1: FX_SD_OUT of PN-PHY 1
16	SD2_ENABLE	0h	r	w	Differentiation/filtering of the SD2 signal for port 2 is 0: direct, over the balls for the integrated PHY 1: over ERTEC die for the integrated PHY
23dt17	reserved	00h			
30dt24	SD2_MUX	0h	r	w	only relevant when SD2_ENABLE = '1' 0...95: GPIO pin 0...95 over which the SD2 signal of the fiber-optic transceiver is received. 96...112, no GPIO selection, input signal of '0' is assumed 113...127, no GPIO selection, input signal of '1' is assumed
31	SD2_SELECT	0h	rh	w	The source for I-Filter input n is: 0: GPIOx set in SD2_MUX 1: FX_SD_OUT of PN-PHY 2

Register:	EXTENSION_REG			Address:	FCh	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	(w)(p)
Description:	Extension Mode: Enable of additional ERTEC200P Step 3 features and address spaces					

Bit	Identifier	Reset	Attr.		Function / Description
0	X_MODE	0h	rh	wp	0: reserved 1: ERTEC200P Step 3 **Note**: Only one time writable with '1' after reset, all successive write accesses will be ignored
7dt1	reserved	00h			
8	reserved	0h	rh		
15dt9	reserved	00h			

31dt16	KEYWORD	0h	rh	w	Writing the KEYWORD 0x9876 to Bits 31 downto 16 enables the write access to the bits (15:0); the read value is always 0000

2.3.14.11.28.2 SCRB2 register description

Module: PN_SUB_SCRB2

Register:	SETUP_REG				Address:	0h	
Bits:	31dt0	Reset value:	0h		Attributes:	r(h)	(w)
Description:	Setup Register						
Bit	Identifier	Reset	Attr.		Function / Description		
0	reserved	0h					
1	reserved	0h	r	w			
2	PERIF_CC_MUX	0h	rh	w	IO data consistency is supported with 0: reserved 1: must be configured to ‘1’ **Note**: Only one-time writeable with '1' after reset, all successive write accesses will be ignored		
7dt3	reserved	00h					
8	OCTALSPI_EN	0h	rh	w	1: OctalSPI control clock enable **Note**: Only one-time writeable with '1' after reset, all successive write accesses will be ignored		
15dt9	reserved	00h					
16	DEV_HWK_0	0h	rh		The value pending at the DEV_HWK(0) pin is latched when PowerOn reset is deactivated		
17	DEV_HWK_1	0h	rh		The value pending at the DEV_HWK(1) pin is latched when PowerOn reset is deactivated		
18	DEV_HWK_2	0h	rh		The value pending at the DEV_HWK(2) pin is latched when PowerOn reset is deactivated		
19	DEV_HWK_3	0h	rh		The value pending at the DEV_HWK(3) pin is latched when PowerOn reset is deactivated		
20	DEV_HWK_4	0h	rh		The value pending at the DEV_HWK(4) pin is latched when PowerOn reset is deactivated		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

21	DEV_HWK_5	0h	rh		The value pending at the DEV_HWK(5) pin is latched when PowerOn reset is deactivated
22	DEV_HWK_6	0h	rh		The value pending at the DEV_HWK(6) pin is latched when PowerOn reset is deactivated
23	DEV_HWK_7	0h	rh		The value pending at the DEV_HWK(7) pin is latched when PowerOn reset is deactivated
31dt24	reserved	00h			

Register:	RESET_REG				Address:	4h	
Bits:	31dt0	Reset value:		0h	Attributes:	r(h)	(w)
Description:	Reset Register						
Bit	Identifier	Reset	Attr.		Function / Description		
0	SW_RES_I2C	0h	r	w	1: asynchronous reset for I2C is active 0: asynchronous reset for I2C is not active and SW_RES_I2C_DONE is cleared		
1	reserved	0h	r	w			
7dt2	reserved	00h					
8	SW_RES_I2C_DONE	0h	rh		0: asynchronous reset for I2C is not active or busy 1: asynchronous reset for I2C is done		
9	reserved	0h	rh				
31dt10	reserved	000000h					

Register:	GDMA_CTRL				Address:	Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	GDMA Control GDMA JOB 31...24 Suspend via GDMA FlowControl						
Bit	Identifier	Reset	Attr.		Function / Description		
0	JOB_SUSPEND24	0h	r	w	JOB x is 0: running 1: suspended bit n -> JOB(24+n)		
1	JOB_SUSPEND25	0h	r	w	JOB x is 0: running 1: suspended bit n -> JOB(24+n)		

	...				
7	JOB_SUSPEND31	0h	r	w	JOB x is 0: running 1: suspended bit n -> JOB(24+n)
31dt8	reserved	000000h			

Register:	OCTALSPI_REG				Address:	14h	
Bits:	31dt0	Reset value:		1h	Attributes:	rh	w
Description:	OctalSPI Control OctalSPI control and status register						
Bit	Identifier	Reset	Attr.		Function / Description		
0	CS_MON_DIS	1h	rh	w	Montoring of rising CS signal (SS_0_N) is 0: enabled 1: disabled		
1	CS_RISE_EDGE	0h	rh	w	Rising CS edge (SS_0_N) is 0: not detected 1: detected (when enabled) Writing 1 clears the bit.		
31dt2	reserved	00000000h					

Register:	OCTALSPI_STROBE_REG				Address:	18h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	w
Description:	OctalSPI Strobe Control Generation of internal RXD Strobe signal						
Bit	Identifier	Reset	Attr.		Function / Description		
15dt0	reserved	0000h					
23dt16	RXDS_SETUP_CYCLES	0h	r	w	Number of clock cycles (SCK) which corresponds to the time needed for the RXDS setup.		
25dt24	reserved	0h					
30dt26	SCK_DELAY	0h	r	w	Only when RXDS_DELAY_MODE = 1: Delay of SCK with granularity of 1ns (500MHz on both edges) and range between: 0h: 7ns (min. delay) 1h: 8ns 2h: 9ns ...		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

					1Eh: 37ns 1Fh: 38ns (max. delay) -> SCK_DELAYx1ns + 7ns
31	RXDS_DELAY_MODE	0h	r	w	RXDS Delay is 0: not active 1: active (using internal RXD strobe)

Register:	EDC_CTRL_REG				Address:	38h	
Bits:	31dt0	Reset value:		700000h	Attributes:	r(h)	w(p)
Description:	EDC Refresh Control Register						
Bit	Identifier	Reset	Attr.		Function / Description		
0	EN_EDC_RFRSH_I_TCM926	0h	rh	wp	0: Disable refresh in ARM926 I-TCM 1: Enable refresh in ARM926 I-TCM		
1	EN_EDC_RFRSH_D_TCM926	0h	rh	wp	0: Disable refresh in ARM926 D-TCM 1: Enable refresh in ARM926 D-TCM		
2	reserved	0h	rh	wp			
3	reserved	0h	rh	wp			
4	EN_EDC_RFRSH_PNIP	0h	rh	wp	0: Disable refresh in PN-IP 1: Enable refresh in PN-IP		
5	reserved	0h	rh	wp			
6	OS_EDC_RFRSH_I_TCM926	0h	r	wp	EDC refresh I-TCM926 1: oneShot 0: permanent		
7	OS_EDC_RFRSH_D_TCM926	0h	r	wp	EDC refresh D-TCM926 1: oneShot 0: permanent		
8	OS_EDC_RFRSH_I_TCM966	0h	r	wp	EDC refresh I-TCM966 1: oneShot 0: permanent		
9	OS_EDC_RFRSH_D_TCM966	0h	r	wp	EDC refresh D-TCM966 1: oneShot 0: permanent		
10	OS_EDC_RFRSH_PNIP	0h	r	wp	EDC refresh PN_IP 1: oneShot 0: permanent		
11	reserved	0h	r	wp			
12	EN_EDC_CORR_I_TCM926	0h	r	wp	0: Disable correction in ARM926 I-TCM		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

					1: Enable correction in ARM926 I-TCM
13	EN_EDC_CORR_D_TCM926	0h	r	wp	0: Disable correction in ARM926 D-TCM 1: Enable correction in ARM926 D-TCM
14	reserved	0h	r	wp	
15	reserved	0h	r	wp	
16	EN_EDC_CORR_PNIP	0h	r	wp	0: Disable correction in PN-IP D-TCM 1: Enable correction in PN-IP
17	reserved	0h	r	wp	
19dt18	reserved	0h			
22dt20	EDC_REFRESH_CYCLE	7h	r	wp	Min. Duration between 2 refresh sequences is 000: no duration (continues refresh) 001: 4ms 010: 8ms 011: 16ms 100: 32ms 101: 64ms 110: 128ms 111: 256ms
23	reserved	0h			
31dt24	KEYWORD	0h	rh	w	Writing the KEYWORD 0x98 to bits (31:24) enables the write access to the bits (17:0); the read value is always 0000h

Register:	EDC_STAT_REG				Address:	3Ch	
Bits:	31dt0	Reset value:		0h	Attributes:	rh	w
Description:	EDC Refresh Status Register						
Bit	Identifier	Reset	Attr.		Function / Description		
0	EDC_CORR_I_TCM926	0h	rh	w	0: no EDC correction in ARM926 I-TCM occurred 1: EDC correction in ARM926 I-TCM occurred Writing 1 clears the bit.		
1	EDC_CORR_D_TCM926	0h	rh	w	0: no EDC correction in ARM926 D-TCM occurred 1: EDC correction in ARM926 D-TCM occurred Writing 1 clears the bit.		
2	reserved	0h	rh	w			
3	reserved	0h	rh	w			

4	EDC_CORR_PNIP	0h	rh	w	0: no EDC correction in PN-IP occurred 1: EDC correction in PN-IP occurred Writing 1 clears the bit.
5	reserved	0h	rh	w	
6	reserved	0h	rh	w	
7	reserved	0h			
15dt8	EDC_CORR_COUNTER	0h	rh	w	Counts the number of all EDC corrections <> FFh: increment + 1 = FFh: no increment Writing a value <> 0 clears the counter.
31dt16	reserved	0000h			

Copyright © Siemens AG 2023. All rights reserved

2.4 Memory mapping

2.4.1 Host IF memory mapping

For booting from an external host, the primary boot loader must create a page for the D-TCM (boot RAM) in the page registers of the HostIF (2x XHIF). The remaining pages can then be set up specifically by the external host. It makes sense to set up additional pages for PN-IP, PerIF, EMC-SDRAM/EMC-SRAM, GDMA and APB peripherals.

The following table shows the paging register set within the XHIF module.

Table 39: Paging register set within the XHIF module

Name	Address CPU 16-bit data width	Description	Name	Address CPU 32-bit data width	Description	Default
XHIF_P0_RG_L	00h	Lower 16 bit of the range configuration 15:8 write- and readable 7:0 only readable	XHIF_P0_RG	00h	32 bit of the range configuration 31:22 only readable 21:8 write- and readable 7:0 only readable	00h
XHIF_P1_RG_L	10h		XHIF_P1_RG	10h		00h
XHIF_P2_RG_L	20h		XHIF_P2_RG	20h		00h
XHIF_P3_RG_L	30h		XHIF_P3_RG	30h		00h
XHIF_P0_RG_H	02h	Upper 16 bit of the range configuration 15:6 only readable 5:0 write- and readable				00h
XHIF_P1_RG_H	12h					00h
XHIF_P2_RG_H	22h					00h
XHIF_P3_RG_H	32h					00h
XHIF_P0_OF_L	04h	Lower 16 bit of the offset configuration 15:8 write- and readable 7:0 only readable	XHIF_P0_OF	04h	32 bit of the offset configuration 31:8 write- and readable 7:0 only readable	00h
XHIF_P1_OF_L	14h		XHIF_P1_OF	14h		00h
XHIF_P2_OF_L	24h		XHIF_P2_OF	24h		00h
XHIF_P3_OF_L	34h		XHIF_P3_OF	34h		00h
XHIF_P0_OF_H	06h	Upper 16 bit of the offset configuration 15:0 write- and readable				00h
XHIF_P1_OF_H	16h					00h
XHIF_P2_OF_H	26h					00h
XHIF_P3_OF_H	36h					00h
XHIF_P0_CFG	08h	Config. of the buffering mode for each single page 1: Page is 32-bit page 0: Page is 16-bit page	XHIF_P0_CFG	08h	Config. of the buffering mode for each single page 1: Buffering mode on read enabled 0 : Buffering mode on read disabled	00h
XHIF_P1_CFG	18h		XHIF_P1_CFG	18h		00h
XHIF_P2_CFG	28h		XHIF_P2_CFG	28h		00h
XHIF_P3_CFG	38h		XHIF_P3_CFG	38h		00h

Name	Address CPU 16-bit data width	Description	Name	Address CPU 32-bit data width	Description	Default
XHIF_VER-SION_L	3Ch	Version of the XHIF	XHIF_ VERSION	3Ch		IA&DT IP ver-sion compli-ant
XHIF_VER-SION_H	3Eh					

The 8-page register of HostIF (2x XHIF with 4 pages each) can be configured either over the APB or externally with the XHIF_XCR_R_A20 signal (default after reset). The selection of the two XHIF modules is controlled by the pin XHIF_SEG_2 (= 0b → XHIF0, = 1b → XHIF1).

2.4.2 Detailed address mapping

Address (hex)			Size				Comment		Slaves	Masters							
Seg	start	end	value	unit	physical	mirrored				ARM926-I	ARM926-D	PN-IP NRT	PN-IP RT/IRT	HOST IF	GDMA	RSC 1	RSC 2
0	0000_0000	03FF_FFFF	64	MB	8 KB	2^{13}	MEM_SWAP parameter assignment: x00b: -> Boot ROM (0 - 8 KB) 001b: -> EMC SDRAM (0 - 64 MB) 101b: -> EMC SDRAM (0 - 64 MB) with EDC 010b: -> EMC asyn memory (0 - 64 MB) 110b: -> EMC asyn memory (0 - 64 MB) with EDC x11b: -> QVZ Error		Boot ROM[1]	x	x						
					64 MB	---			EMC SDRAM[2]	x	x	x		x	x		
					64 MB	---			EMC_SDRAM with EDC[2]								
					64 MB	---			EMC asyn memory[2]								
					64 MB	---			EMC asyn memory with EDC[2]								
					64 MB	---			QVZ Error								
					256 KB	---	After ARM926 I-TCM activation (in the CP15 c9 register): ARM 926 I-TCM (0 – 256 KB physical. / Step 64 KB)		ARM926 I-TCM[3,5]	x							
					> 63 MB	---											
	0400_0000	0403_FFFF	256	KB	256 KB	---			_not used_								
	0404_0000	07FF_FFFF	63.75	MB	< 64 MB	----			_not used_								
	0800_0000	0803_FFFF	256	KB	256 KB	----	ARM 926 D-TCM (0 – 256 KB physical. / Step 64 KB)		ARM926 D-TCM[4]		x	x		x	x		
	0804_0000	0FFF_FFFF	127.75	MB	<128 MB	----			_not used_								
1	1000_0000	105F_FFFF	6	MB	6 MB	---			ARM-ICU		x						
	1060_0000	107F_FFFF	2	MB	2 MB	---			PN-IP		x			x			
	1080_0000	109F_FFFF	2	MB	128 KB	2^{4}	Appl. access: 128 KB physical		PerIF		x		x	x	x		
	10A0_0000	10AF_FFFF	1	MB	1 MB	---	GDMA register and internal GDMA Job SRAM (Size: 4608 B) Register: 10A0_0000..10A0_00AF JOB SRAM: 10A0_00B0..10A0_12AF Not used: 10A0_12B0..10AF_FFFF		GDMA		x			x	x		
	10B0_0000	10BF_FFFF	1	MB	1 MB	---			_not used_								

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Address (hex)			Size				Comment		Slaves	Masters							
Seg	start	end	value	unit	physical	mirrored				ARM926-I	ARM926-D	PN-IP NRT	PN-IP RT/IRT	HOST IF	GDMA	RSC 1	RSC 2
	10C0_0000	10CF_FFFF	1	MB	1 MB	---			_not used_								
	10D0_0000	10DF_FFFF	1	MB	1 MB	---			EMC register[7]		x			x	x		
	10E0_0000	10EF_FFFF	1	MB	1 MB	---			ETB11 Memory		x				x		
	10F0_0000	10FF_FFFF	1	MB	512 B	2^{11}			ETB11 register		x				x		
	1100_0000	110F_FFFF	1	MB	512 B	2^{11}			RS Controller 1		x			x	x		
	1110_0000	111F_FFFF	1	MB	512 B	2^{11}			RS Controller 2		x			x	x		
	1120_0000	112F_FFFF	1	MB	1 MB	---			_not used_								
	1130_0000	113F_FFFF	1	MB	1 MB	---			_not used_								
	1140_0000	17FF_FFFF	108	MB	108 MB	---			_not used_								
	1800_0000	1FFF_FFFF	128	MB	128 MB				OctalSPI	x	x			x	x	x	x
2	2000_0000	2FFF_FFFF	256	MB	256 MB	---			EMC SDRAM[7]	x	x	x		x	x	x	x
3	3000_0000	33FF_FFFF	64	MB	64 MB	---	Chip Select Bank 0		EMC asyn memory[7]	x	x	x		x	x	x	x
	3400_0000	37FF_FFFF	64	MB	64 MB	---	Chip Select Bank 1										
	3800_0000	3BFF_FFFF	64	MB	64 MB	---	Chip Select Bank 2										
	3C00_0000	3FFF_FFFF	64	MB	64 MB	---	Chip Select Bank 3										
4	4000_0000	4000_7FFF	32	KB	32 KB	---		APB_Peripherals	PerIF (Consistency Control)		x			x	x		
	4000_8000	4000_9FFF	8	KB	8 KB	---	Reduced Boot ROM (8 KB of BootROM)		BOOT ROM	x	x						
	4000_A000	4000_AFFF	4	KB	4 KB	---			UART1		x			x	x		
	4000_B000	4000_BFFF	4	KB	4 KB	---			UART2		x			x	x		
	4000_C000	4000_CFFF	4	KB	4 KB	---			UART3		x			x	x		
	4000_D000	4000_DFFF	4	KB	4 KB	---			UART4		x			x	x		
	4000_E000	4000_EFFF	4	KB	4 KB	---			I^2C		x			x	x		
	4000_F000	4000_F0FF	256	B	256 B	---			SCRB		x			x	x		
	4000_F100	4000_FFFF	3.75	KB	64 B	60			SCRB2		x			x	x		
	4001_0000	4001_00FF	256	B	256 B	---	No 8/16-bit reversal of endianess No parity generation/check		SPI1		x			x	x		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Address (hex)			Size				Comment		Slaves	Masters							
Seg	start	end	value	unit	physical	mirrored				ARM926-I	ARM926-D	PN-IP NRT	PN-IP RT/IRT	HOST IF	GDMA	RSC 1	RSC 2
	4001_0100	4001_01FF	256	B	256 B	---	8-bit reversal of endianess No parity generation/check										
	4001_0200	4001_02FF	256	B	256 B	---	No 8-bit reversal of endianess Parity generation/check										
	4001_0300	4001_03FF	256	B	256 B	---	8-bit reversal of endianess Parity generation/check										
	4001_0400	4001_04FF	256	B	256 B	---	16-bit reversal of endianess No parity generation/check										
	4001_0500	4001_0FFF	2.75	KB	<3 KB	---	reserved corresponds to 0x4001_0000 - 0x4001_00FF										
	4001_1000	4001_10FF	256	B	256 B	---	No 8/16-bit reversal of endianess No parity generation/check										
	4001_1100	4001_11FF	256	B	256 B	---	8-bit reversal of endianess No parity generation/check										
	4001_1200	4001_12FF	256	B	256 B	---	No 8-bit reversal of endianess Parity generation/check										
	4001_1300	4001_13FF	256	B	256 B	---	8-bit reversal of endianess Parity generation/check		SPI2		x			x	x		
	4001_1400	4001_14FF	256	B	256 B	---	16-bit reversal of endianess No parity generation/check										
	4001_1500	4001_1FFF	2.75	KB	<3 KB	---	reserved corresponds to 0x4001_0000 - 0x4001_00FF										
	4001_2000	4001_2FFF	4	KB	256 B	2^4			Timer 0 – 5		x			x	x		
	4001_3000	4001_3FFF	4	KB	32 B	2^7			Watchdog		x			x	x		
	4001_4000	4001_4FFF	4	KB	8 B	2^9			F-Counter		x			x	x		
	4001_5000	4001_5FFF	4	KB	4 KB	---			_not used_								
	4001_6000	4001_6FFF	4	KB	4 KB	---			_not used_								
	4001_7000	4001_7FFF	4	KB	4 KB	---			_not used_								
	4001_8000	4001_8FFF	4	KB	256 B	2^4			GPIO		x			x	x		

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Address (hex)			Size				Comment		Slaves	Masters							
Seg	start	end	value	unit	physical	mirrored				ARM926-I	ARM926-D	PN-IP NRT	PN-IP RT/IRT	HOST IF	GDMA	RSC 1	RSC 2
	4001_9000	4001_9FFF	4	KB	128 B	2^5			I-Filter		x			x	x		
	4001_A000	4001_AFFF	4	KB	256 B	2^4			HostIF		x			x	x		
	4001_B000	4001_BFFF	4	KB	4 KB	---			_not used_								
	4001_C000	4001_FFFF	16	KB	16 KB	---			_not used_								
	4002_0000	4002_5FFF	24	KB	24 KB	---			_not_used_								
	4002_6000	4002_6FFF	4	KB	4 KB	---			_not_used_								
	4002_7000	4002_7FFF	4	KB	4 KB	---			_not_used_								
	4002_8000	4002_AFFF	12	KB	12 KB	---			_not_used_								
	4002_B000	4002_BFFF	4	KB	4 KB	---			_not_used_								
	4002_C000	4002_CFFF	4	KB	1 KB	---	_not_used_		_not_used_		x			x	x	x	x
					256 B	2^1	4002_C400 - 4002_C5FF: OSPI IP original region without Memory Swapping		OctalSPI via dedicated AHB/APB Bridge								
					256 B	2^1	4002_C600 - 4002_C7FF: OSPI IP Mirror region with Halfword Memory Swapping										
					256 B	2^1	4002_C800 - 4002_C9FF: OSPI IP Mirror region with Word Memory Swapping										
					1.5 KB	---	_not_used_										
	4002_D000	4002_D3FF	1	KB	1 KB	---			_not_used_								
	4004_D000	400F_FFFF	716	KB	716 KB	---			_not_used_								
	4010_0000	4010_FFFF	64	KB	64 KB	---	Complete Boot ROM whole 64 KB		Boot ROM[6]	x	x						
	4011_0000	4FFF_FFFF	255	MB	255 MB	---			_not_used_								
5	5000_0000	5FFF_FFFF	256	MB	256 MB	---			_not_used_								
6	6000_0000	67FF_FFFF	128	MB	128 MB	---			EMC SDRAM[7] with EDC	x	x	x		x	x	x	x
	6800_0000	69FF_FFFF	32	MB	32 MB	---			EMC asyn memory[7] with EDC	x	x	x		x	x	x	x

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

Address (hex)			Size				Comment		Slaves	Masters							
Seg	start	end	value	unit	physical	mirrored				ARM926-I	ARM926-D	PN-IP NRT	PN-IP RT/IRT	HOST IF	GDMA	RSC 1	RSC 2
	6A00_0000	6BFF_FFFF	32	MB	32 MB	---			EMC asyn memory[7] with EDC	x	x	x		x	x	x	x
	6C00_0000	6DFF_FFFF	32	MB	32 MB	---			EMC asyn memory[7] with EDC	x	x	x		x	x	x	x
	6E00_0000	6FFF_FFFF	32	MB	32 MB	---			EMC asyn memory[7] with EDC	x	x	x		x	x	x	x
7	7000_0000	7FFF_FFFF	256	MB	256 MB	---			_not_used_								
8-C	8000_0000	CFFF_FFFF	1280	MB	1280 MB	---			_not_used_								
D-F	D000_0000	FFFF_FFFF	768	MB	768 MB	---			_not_used_								

Key:

1 After a reset, the boot ROM is at address 0. The first 64 MB of EMC-SDRAM or EMC asyn memory (chip select Bank0) can also be mapped to address 0 with the 'MEM_SWAP' register in the SCRB.

2 After a reset, no memory is assigned to address 0. The first 64 MB of EMC-SDRAM or EMC asyn memory (chip select Bank0) can also be mapped to address 0 with the 'MEM_SWAP' register in the SCRB.

3 ARM926 I-TCM can only be accessed internally from the ARM926 and not over the AHB. After activation, this address range is no longer shown at the AHB. This range can also be shown at other addresses with alignment in segment 0, however not in the ARM926 D-TCM area.

4 The ARM926 D-TCM is 0 - 256 KB (adjustable in 64 KB increments) and can be shown at address 0x0800 0000h by the SW in the ARM926 coprocessor register 'CP15 c9'. ARM926 then accesses its D-TCM and not the AHB in this address range. Please **note** that ARM926 D-TCM can only be displayed in the ARM address area in increments of 2^n. If, for example, a physical size of D-TCM = 192 KB has been selected in 'TCM926_MAP' register, ARM926 can only assign D-TCM an address range of 256 KB.

5 The ARM926 I-TCM is 0 - 256 KB (adjustable in 64 KB increments) and can be displayed by the SW in the ARM926 coprocessor register 'CP15 c9' at address 0 or at another address in segment 0 (except in the D-TCM926). The I-TCM is disabled after reset. After display, ARM926 accesses its I-TCM and not the AHB in this address range (ARM926 cannot access its I-TCM over the AHB). Please note that ARM926 I-TCM can only be displayed in the ARM address range in increments of 2^n. If, for example, a physical size of I-TCM = 192 KB has been selected in 'TCM926_MAP' register, ARM926 can only assign I-TCM an address range of 256 KB.

6 Accesses from masters that cannot reach the Boot ROM result in an AHB access error, whereas in all other regions of the APB an APB access error is generated.

7 EMC is only accessible when EMC_HOSTIF_MUX is set to '0'.

Copyright © Siemens AG 2023. All rights reserved
Technical data subject to change

3 MISCELLANEOUS

3.1 Abbreviations

Abbreviation	Description
AHB	AMBA **A**dvanced **H**igh-performance **B**us
AMBA	**A**dvanced **M**icrocontroller **B**us **A**rchitecture
APB	AMBA **A**dvanced **P**eripheral **B**us
AR	**A**pplication **R**elationship (e.g., controller <-> device)
AXI	**A**dvanced e**X**tensible **I**nterface
BS	**B**oundary **S**can
CAS	**C**olumn **A**ddress **S**ignal
CPM	**C**onsumer **P**rotocol **M**achine for cyclic transmitted data
CR	**C**ommunication **R**elationship (e.g., input data / output data)
CRU	**C**lock and **R**eset **U**nit
CS	**C**hip **S**elect
D-TCM	**D**ata **T**ightly **C**oupled **M**emory
DA	**D**evice **A**ccess
DDR	**D**ouble **D**ata **R**ate
DFT	**D**esign **F**or **T**est
DFP	**D**ynamic **F**rame **P**acking
GDMA	**G**eneric **D**irect **M**emory **A**ccess
ECC	**E**rror **C**orrection **C**ode
EDC	**E**rror **D**etection and **C**orrection
EMC	**E**xternal **M**emory **C**ontroller
EOI	**E**nd **O**f **I**nterrupt
ETB	**E**mbedded **T**race **B**uffer
ETM	**E**mbedded **T**race **M**acrocell
FD	**F**ull-**D**uplex
FF	**F**lip-**F**lop
FIFO	**F**irst **I**n **F**irst **O**ut
FIQ	**F**ast **I**nterrupt re**Q**uest
FIT	**F**ailure **I**n **T**ime

Copyright © Siemens AG 2023. All rights reserved

Abbreviation	Description
GMII	**G**igabit **M**edia **I**ndependent **I**nterface
GPIO	**G**eneral **P**urpose **I**nput/**O**utput
HD	**H**alf-**D**uplex
HW	**H**ard**w**are
I^2C	**I**nter-**I**ntegrated **C**ircuit
I-Filter	**I**ngress **Filter**
I-TCM	**I**nstruction **T**ightly **C**oupled **M**emory
ICE	**I**n-**C**ircuit-**E**mulator
ICU	**I**nterrupt **C**ontrol **U**nit
INTA	**INT**errupt Subblock **A** of ARM-ICU
INTB	**INT**errupt Subblock **B** of ARM-ICU
IRQ	**I**nterrupt **R**e**q**uest
IRT	**I**sochronous **R**eal-**T**ime
IRT High Performance	**I**sochronous **R**eal-**T**ime with fast-forwarding, pack/unpack, and NRT fragmentation
ISR	**I**nterrupt **S**ervice **R**egister
IP	**I**ntellectual **P**roperty
JTAG	**J**oint **T**est **A**ction **G**roup
LSB	**L**east **S**ignificant **B**it
MAC	**M**edium **A**ccess **C**ontrol
MC	**M**otion **C**ontrol
MDI-X	**M**edium-**D**ependent **I**nterface crossover
MII	**M**edia-**I**ndependent **I**nterface
MISO	**M**aster **I**n **S**lave **O**ut
MMD	**M**emory **M**ap **D**ecoder
MMU	**M**emory **M**anagement **U**nit
MOSI	**M**aster **O**ut **S**lave **I**n
MSB	**M**ost **S**ignificant **B**it
NMI	**N**on **M**askable **I**nterrupt
NRT	**N**on **R**eal-**T**ime
OSPI	**O**ctal-**SPI**

Abbreviation	Description
PBL	**P**rimary **B**oot **L**oader
PCB	**P**rinted **C**ircuit **B**oard
PCI	**P**eripheral **C**ommunication **I**nterface
PerIF	**Per**ipheral **I**nter**F**ace
PHY	**Phy**sical Layer
PLL	**P**hase **L**ocked **L**oop
PNPLL	**P**ROFI**N**ET **PLL**
PPM	**P**rovider **P**rotocol **M**achine for cyclic transmitted data
PTCP	**P**recision **T**ransparent **C**lock **P**rotocol
QVZ	Acknowledgement delay (German: **Q**uittungs**v**er**z**ug)
RAS	**R**ow **A**ddress **S**ignal
RSC	**R**eed-**S**olomon **C**ontroller
RT	**R**eal-**T**ime
RTC	**R**eal-**T**ime **C**lock
SBL	**S**econdary **B**oot **L**oader
SCRB	**S**ystem **C**ontrol **R**egister **B**lock
SDR	**S**ingle **D**ata **R**ate
SDRAM	**S**ynchronous **D**ynamic **R**andom-**A**ccess **M**emory
SER	**S**oft **E**rror **R**ate
SMI	**S**erial **M**anagement **I**nterface
SPI	**S**erial **P**eripheral **I**nterface
SRAM	**S**tatic **R**andom-**A**ccess **M**emory
SW	**S**oft**w**are
TCM	**T**ightly **C**oupled **M**emory
TLB	**T**ranslation **L**ookaside **B**uffer
UART	**U**niversal **A**synchronous **R**eceiver/**T**ransmitter
WD	**W**atch**D**og

3.2 References

Reference	Title
/4/	ETM9 Technical Reference Manual (Rev. 2a) (ARM DDI 0157E)
/5/	ETM Specification (ARM IHI 0014 H)
/6/	ARM9E-S Technical Reference Manual (ARM DDI 0198D)
/7/	ARM926EJ-S Technical Reference Manual (ARM DDI 0201A)
/15/	DDI0183F_uart_pl011_r1p4_trm.pdf, UART (PL011) Technical Reference Manual
/29/	I²C-Bus Specification, V2.1 of January 2000
/46/	ETB11 Technical Reference Manual (ARM DDI0275B) of 2002/2003

Copyright © Siemens AG 2023. All rights reserved