SIEMENS

ERTEC 200P

Enhanced Real-Time Ethernet Controller

Manual

Edition (05/2013)

DISCLAIMER

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

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

TECHNICAL DATA SUBJECT TO CHANGE.

Preface

Target Audience of this Manual

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

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

Structure of this Manual

- Section 1 Overview of the ERTEC 200P.
- Section 2 System features.
- Section 3 Hardware structure.
- Section 4 Hardware interfaces.
- Section 5 Software interfaces.
- Section 6 Miscellaneous

Scope of the Manual

This manual applies to the following product:

ERTEC 200P

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

Guide

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

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

Additional Support

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

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

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

http://www.siemens.com/comdec

Technical Contacts for Germany / Worldwide

Siemens AG
Automation & Drives
ComDeC
Street address:
Würzburgerstr.121
90766 Fürth
Federal Republic of Germany

Phone: +49 (0) 911/750-2080
Fax: +49 (0) 911/750-2100
E-mail: ComDeC@siemens.com

Mailing address:
P.O. Box 2355
90713 Fürth
Federal Republic of Germany

Technical Contacts for USA

PROFI Interface Center:
One Internet Plaza
PO Box 4991
Johnson City, TN 37602-4991

Fax: +1 (423)- 262- 2103
Phone: +1 (423)- 262- 2576
E-mail: pic.industry@siemens.com

Table of contents

1 OVERVIEW OF THE ERTEC 200P

The Enhanced Real-Time Ethernet Controller 200P (ERTEC 200P) is a further development of the ERTEC 200. It disposes of a PN-IP with integrated PHYs for high-performance PROFINET communication as well as of an ARM926EJ-S processor for application processing. The mechanisms of the performance upgrade "Fast Forwarding", Dynamic Frame Packing" as well as "Fragmentation"are implemented in the PN-IP. The ERTEC 200P thus allows the development of PROFINET devices with cycle times up to 31.25 µs and isochronous-specific applications. In addition to rapid real-time communication, the ERTEC 200P also allows unlimited access to TCP/IP data and service so that non-time-critical data can be transferred parallel.
Its flexible architecture allows devices to be realized that are both modular and compact. The application of the device can be executed both on the integrated ARM processor of the ERTEC 200P and on an external host CPU. Both applications are described in detail in this documentation as use cases.
The ERTEC 200P is thus suitable for use in a wide range of field device types: Rapid IO peripherals, high-precision measuring instrumentation, synchronous drives and encoders as well as all further types of high-performance and intelligent automation devices

1.1 Switch functions

The term switch functions is used on the one hand for all the functions required for forwarding telegrams by the switch and on the other hand for all the functions used to process data for the user interface.
Through the concept of isochronous real-time Ethernet data traffic via Ethernet is divided into three categories: Depending on the application, TCP/IP, RT or IRT is used for the communication. Because of the required processing of the data by the TCP/IP stack and the unforeseeable runtimes through the network the well-known TCP/IP communication is not suitable for real-time applications in control technology as required by modern control and drive concepts. Operation via RT, that runs on the same switch mechanisms as TCP/IP communication, represents an improvement. In the case of RT an increase in the deterministics of the network is achieved by prioritization of the telegrams and higher throughput is attained by processing the data for the end node..
A further increase in the deterministics is achieved by the transition to IRT traffic in which cyclic communication is executed in reserved time phases. A difference is made between address-based forwarding and topological forwarding.
A further increase in performance is provided by the performance upgrade at IRT that operates on the basis of Fast Forwarding Multicast Address (with Frame ID in Octet 1/2) and uses Pack Frames (bundling of several devices). This reduces the cut through time markedly. A further advantage is the notable reduced cycle times.

1.1.1 IRT communication

In the case of IRT, cyclic communication runs in reserved time phases. Selection between address-based forwarding (RTC2) and topological forwarding (RTC3) is possible.
In the case of address-based forwarding the control information required for a telegram transfer only has to be stored at the sender and the receiver. The path is found automatically through the MAC target address.
In the case of topological forwarding the control information required respectively is stored for each individual telegram in **all** the participating nodes, that is in addition to the sender and receiver in all the concerned forwarding nodes. Important advantages are:

- Deterministics in the network through time-driven forwarding of telegrams (prerequisite for this control is a common time base for all the participants of a contiguous real-time Ethernet network)
- Performance of the user interface that is supported by the processing of the user data in process images is supported by the hardware

The ERTEC 200P can also be used as a relative forwarder in simple network topologies.

1.1.2 Performance upgrade at IRT communication

At systems with many participants and particularly high line depths the forwarding time in each individual node becomes a determining factor for the performance. The performance upgrade uses Fast Forwarding (locally administrated MultiCast-Frames with Frame ID in Octet 1/2). This allows every device to already carry out forwarding after the first two bytes of the DA address. Resulting in reduced forwarding time.

Typical field devices in automation technology require only a few process data bytes for cyclic user data transport. PROFINET always requires a complete Ethernet frame with the associated overhead (64 bytes) to transfer these few bytes. To achieve a significant improvement in the performance compared to the current PROFINET, the available band width has to be used better with the performance upgrade. This is achieved by means of Dynamic Frame Packing (DFP). This stores the data of several devices in one frame, the so-called Pack Frame. A so-called "subframe" is assigned to each device for its data. Several "subframes" then form an Ethernet frame.

To allow smaller send cycles the performance upgrade supports TCP/IP fragmentation. Here the switch automatically fragments the TCP/IP frames when necessary.

To allow smaller send cycles the performance upgrade supports TCP/IP fragmentation. Here the switch automatically fragments the TCP/IP frames when necessary.

1.2 System functions

1.2.1 Use cases for the ERTEC 200P

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

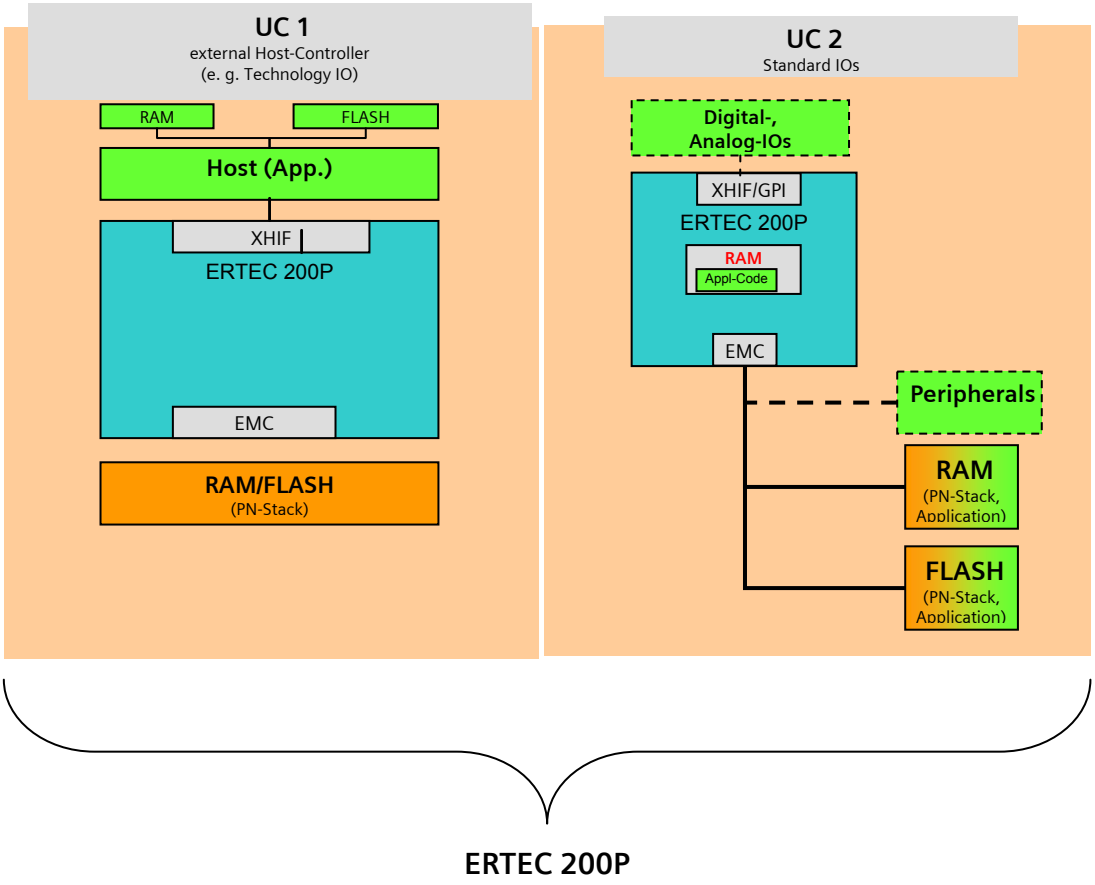


Figure 1: ERTEC 200P use cases

1.2.1.1 Use case 1 (UC1): Operation with external host

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

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

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

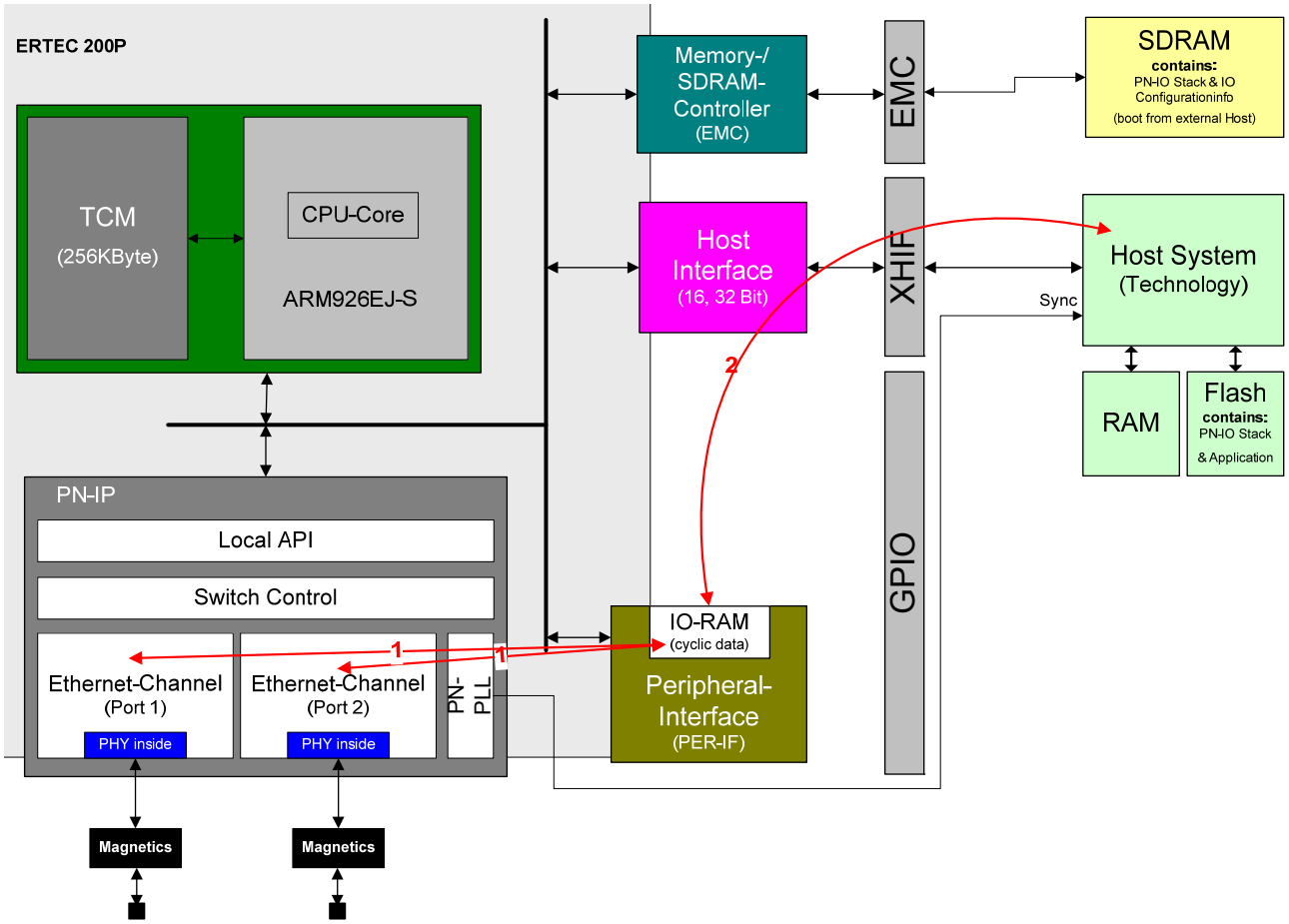


Figure 2: Application operation with external host

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

1.2.1.2 Use case 2 (UC2): Operation without external host

The ERTEC 200P is available for the realization of complex standard IO and remote IO applications with notably more application code. The ERTEC 200P has an external memory interface (flash, SDRAM, SRAM, peripherals) and a small integrated memory (256 kbytes). The PN IO stack and the application lie in the external memory. High-performance acyclic communication data or application code that is to executed rapidly are loaded in the integrated memory of the ERTEC 200P.
The digital and analog IOs are connected to the ERTEC 200P pins. Cyclic data can be controlled by the ARM926 or the GDMA controller via 2x SPI. Alternatively an external bus controller can also be connected via the EMC.

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

Operation without external host

Figure 3: Application operation without external host shows the application of the standard IO.

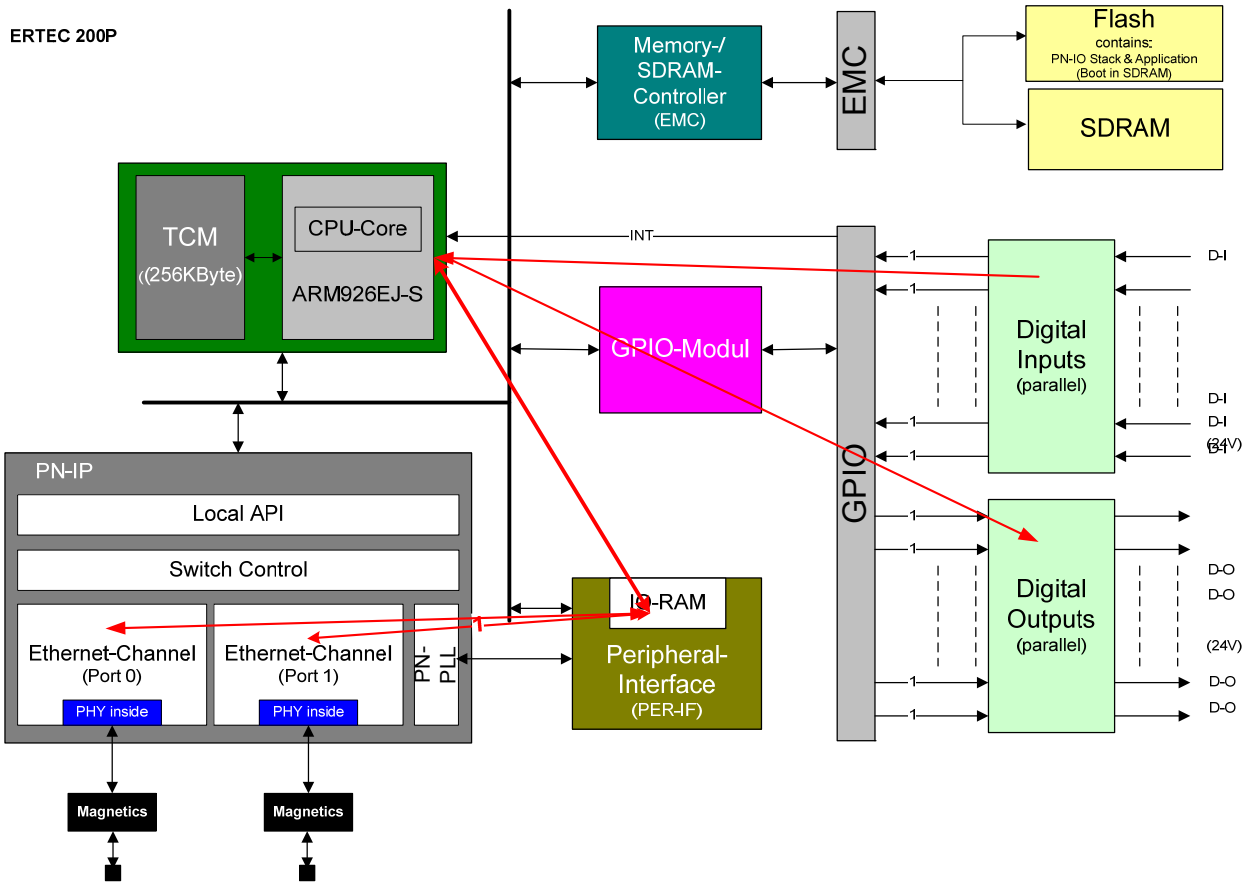


Figure 3: Application operation without external host

2 SYSTEM FEATURES

Manufacturer, technology, enclosure

- Manufacturer: Renesas Electronics
- Enclosure: FPBGA400, 17 mm * 17 mm

Operating conditions

- Ambient temperature: –40 to +85°C
- Supply voltage IOs: EMC interface: 1.8 V +5%/-10%
  Host interface: 1.8 V/3.3 V +5%/-10%
  GPIO31..0, ...: 3.3 V +5%/-10%
- Supply voltage core: 1.2 V +5%/-0.1 V
- Supply voltage PHY: 1.5 V +5%/-10%
- Power loss: Max. 1.62 W (incl. 2x int. PHY)

Processor system

- Integrated processor system ARM926EJ-S (frequency 125/250 MHz)
  - 16 kbyte Data and 16 kbyte Instruction Cache
  - 256 kbyte Instruction/Data Tightly Coupled Memory incl. Byte EDC, adjustable in 64 kbyte steps (I-TCM: 0 – 256 kbytes, D-TCM: 256 – 0 kbytes)
  - Debug capability through embedded ICE with JTAG interface, ETM cell with ETB (Embedded Trace Buffer)
  - Memory Management Unit (MMU)

Bus structure:

- Internal 32-bit structure
- Multi-layer architecture with parallel access structure multi-master to multi-slaves (125 MHz)
- 16/32-bit bus interface to external SDRAM/SRAM/flash and external peripherals

PN-IP:

- 2 Ethernet ports with integrated PHYs (100 Mbits, full-duplex)
- IRT to 31.25 µs cycle time, IRT, RT and TCP/IP data traffic

Interfaces:

- EMC (External Memory Controller) 1.8 V
  - (Mobile) SDRAM (125 MHz)
  - Asynchronous SRAM interface (4 Chip-Select areas, Ready-Control)
    Burst Flash Interface
- XHIF (External Host Interface) 1.8 V / 3.3 V
  - Interface for external host
  - 16-/32-bit data width
- GPIO: GPIO31-0 with 16 I-filters, GPIO95-32 with 64 I-filters parallel
- 1x I^2C
- 1x I^2C (in the PN-IP for POF transceivers)
- 4x UART
- 2x SPI1 (master / slave)

General functions:

- Internal cycle generation (quartz oscillator, PLL)
- Integrated boot ROM (8 kbytes)
- 6 x Timers
- Watchdog
- F-counter
- GDMA controller
- ARM926 interrupt controller

Test function:

- Boundary Scan

3 HARDWARE STRUCTURE

3.1 Block diagram ERTEC 200P

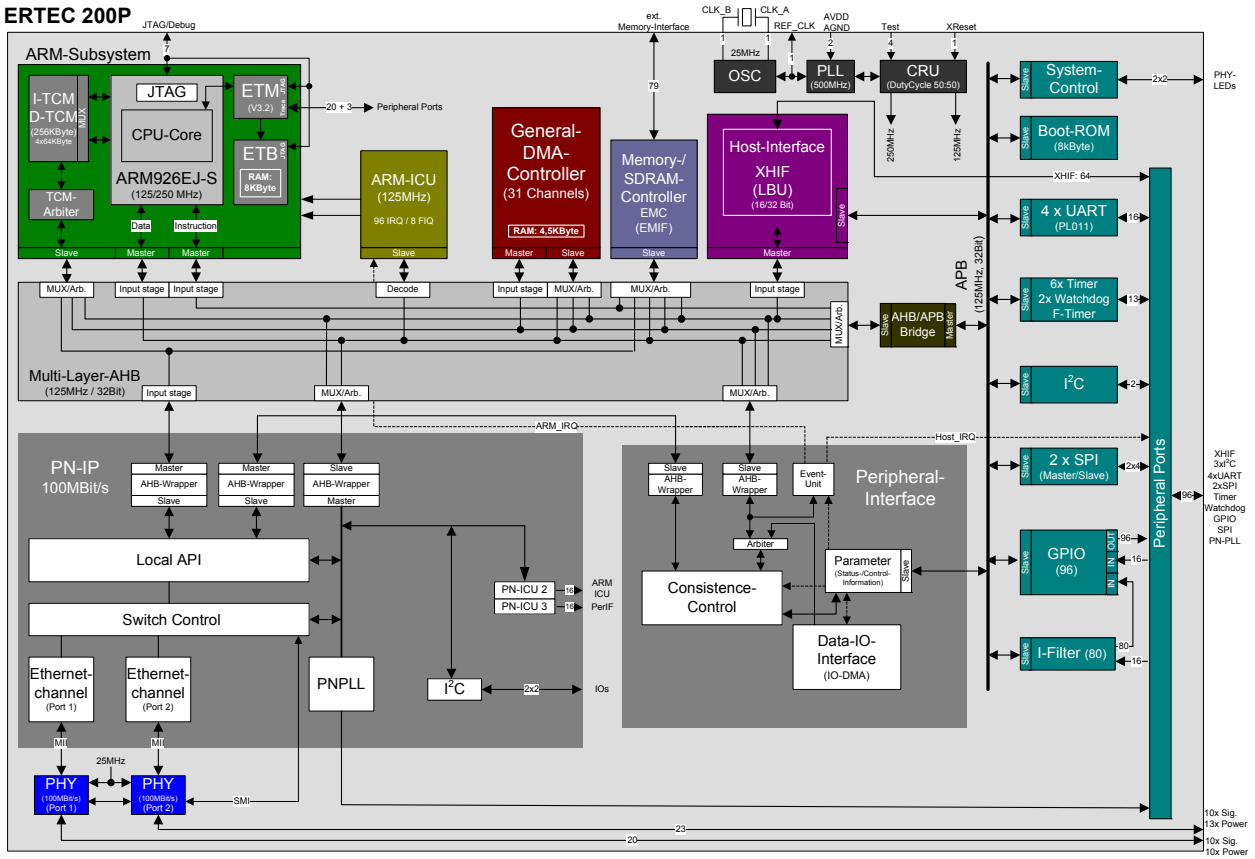


Figure 4: ERTEC 200P block diagram

3.2 Overview of the ERTEC 200P IP utilization

IP	Used by the PROFINET stack	Usable by the application	Note
TCM	no	yes	-
=> nicht im TCM ARM926	yes	yes	-
ETM	no	yes	For Debugging purposes
ETB	no	yes	For Debugging purposes
ARM-ICU	yes	yes	Details see clause 3.6.1.9
General-DMA-Controller	no	yes	-
Host-Interface	-	yes	*1)
System-Control (SCRB)	yes	yes	The System-Control Register Block contains the registers which are used by the application and PROFINET Stack
System-Control (Phy-LEDs)	yes	no	-
Boot-ROM	n/a	n/a	-
UARTs (1-4) (@APB)	no	yes	-
F-Timer (@APB)	no	yes	-
2x Watchdog (@APB)	no	yes	see also Application notes, clause 3.3.13
6x Timer (@APB)	Timer 0	Timer 1-5	Timer 0 is used by the operation system of the Evaluation Kit
I2C (@APB)	no	yes	-
2x SPIs (APB)	no	yes	-
GPIO (GPIO31-0)	yes	yes	The following GPIOs are used by the Evaluation Kit ERTEC 200P: GIO0: SyncOut GPIO9: SyncIn GPIO25-31: LEDs
GPIO (GPIO95-32)	no	yes	-
I-Filter (@APB)	no	yes	-
Peripherie-Interface	yes	yes	-
Peripherie-Interface (Host-IRQ)	no	no	1*)
PN-IP	yes	no	-
I2C (@PN-IP)	yes	no	-
*1) This interface may be usable by the application in a future version of the PROFINET stack version delivered with the Evaluation Kit ERTEC 200P. Please refer to the User Interface Description of the Evaluation Kit for supported functionality.			

Table 1: Overview of the ERTEC 200P IP utilization

3.3 Application notes

The user should pay attention to the following application notes:

3.3.1 EMC SDRAM-Interface

The EMC of the ERTEC 200P offers two setting options for the cycled signal output (address, data and control signals) to an externally connected SDRAM that can be configured using Bit (27) in the Extended_Config register (see Chapter 5.3.6):

- Extended_Config(27)= '0': SDRAM signal output with feedback clock at the pin CLK_I_SDRAM *(register value after reset)*
- Extended_Config(27)= **'1'**: SDRAM signal output with internal system cycle *(value required for correct operation!)*

⇨ The reset value of the register bit **must** be reconfigured to **Extended_Config(27) = '1'** here in the ERTEC 200P for correct SDRAM timing.

3.3.2 EMC BurstFlash-Interface

The EMC of the ERTEC 200P offers two setting options for the cycled signal output (XAV_BF) to an externally connected BurstFlash that can be configured using Bit (26) in the Extended_Config register (see Chapter 5.3.6

- Extended_Config(26)= '0': BurstFlash signal output with feedback clock at the pin CLK_I_BF *(register value after reset)*
- Extended_Config(26)= **'1'**: BurstFlash signal output with internal system cycle *(value required for correct operation!)*

⇨ The reset value of the register bit **must** be reconfigured to **Extended_Config(26) = '1'** here in the ERTEC 200P for correct BurstFlash timing.

The EMC of the ERTEC 200P offers two setting options for the selection of the operating frequency of an externally connected BurstFlash that can be configured using Bit (0) in the BF_Config register (see Chapter 5.3.6):

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

⇨ The reset value of the register bit **BF_Config(0) = '0' must** be retained here in the ERTEC 200P for correct BurstFlash timing!

3.3.3 I-/D-TCM Mapping register of the ARM926 subsystems

The ARM926 subsystem is equipped with a TCM block (256 kbytes) whose distribution between I-TCM for Code and D-TCM for Data can be set using a mapping register.

⇨ The I-/D-TCM mapping register of the **ARM926 subsystem** (SCRB register bits TCM926_MAP(2:0), see Chapter 5.3.8 and Chapter 3.4.3) may be switched over during operation at ERTEC 200P, if it is ensured that the **ARM926 is not accessing the I-/D-TCM** at the switching-over moment, meaning that neither Code nor Data access is being carried out on it.
This can be achieved by executing the code from an external RAM, such as the SDRAM, at the switching-over moment.

3.3.4 I-TCM accesses of the ARM926 with 250MHz

Accesses of the ARM926 to the I-TCM are exclusively accomplished by the ARM-Core and scale with the configured cycle rate (see clause: 3.4.3).

⇨ At the ERTEC 200P the performance of the I-TCM accesses are not scalable with the cycle rate. The performance is equal regardless if 250MHz or 125MHz are used. Reason: At a cycle rate of 250MHz the accesses to the I-TCM are accomplished with one wait state.

The access to the Cache-Memory scales with the clock rate of the ARM926 therefore the performace at 250MHz is higher than with 125MHz.

3.3.5 Accesses to the IO-RAM in the Peripherie-Interface (Per-IF)

The IO-RAM in the Per-IF can be accessed by different AHB-instances (ARM926, XHIF, GDMA) and also from the PN-IP. The Per-IF supports burst and single accesses from the AHB side. The corresponding operation mode has to be set in the Burst_Config register (see 5.3.2.2) .

⇨ For proper operation, burst accesses from the application AHB-instances ARM926, XHIF and GDMA to the IO-RAM in the Per-IF are not allowed. Therefore the default values in the Burst_Config register shall be changed from the default vaules (burst access) to single access as follows:

- Set *Burst_Config.BurstMode_comAHB(1dto0)* = **'0b01'** *(default)* → **'0b11'**
- Set *Burst_Config.BurstMode_applAHB(9dto8)* = **'0b01'** *(default)* → **'0b00'**

Note: With this PerIF settings the PerIF Interrupt *CR_State_comErr_INT* can be triggered wrongly and should be ignored. This interrupt is for debug purposes only. Per default all interrupts in the PerIF registers *Host_IRQmask_low* and *PN_IRQmask_low* are masked.
.

3.3.6 Trigger of GDMA HW-Jobs by GPIOs 0...3 is not supported

The GDMA supports jobs which can be triggered by different sources, like Timer, PNPLL and GPIOs 0...3.

⇨ The trigger of GDMA jobs by the GPIOs 0...3 is not supported.

3.3.7 Acknowledgement of the PN-ICU3 accumulative interrupts in the Event-Unit of the PerIF

The PN-ICU3 in the PN-IP provides 16 interrupts (2 accumulative interrupts and 14 individual interrupts) for the Event-Unit in the PerIF. The accumulative interrupts are provided only level triggered, therefore the interrupt service routine has to take care about the acknowledgment of these interrupts.

⇨ For the acknowledgment of the 2 accumulative PN-ICU3 interrupts the interrupt service routine has at first to acknowledge the interrupts at the PN-ICU3 and after that the interrupt service routine can acknowledge the accumulative interrupts in the Event-Unit at the PerIF.

3.3.8 AHB Burstbreaker

The values of the burst length of the ARM926I- and ARM926D-AHB-Master-Interface should be changed from the default values (burst breaker deactivated) to 8, see also clause 5.3.8.

- Set *AHB_BURSTBREAKER. NR_ADDR_ARM926_D(7dto0)* = **'0x00'** *(default)* ➔ **'0x08'**
- Set *AHB_BURSTBREAKER. NR_ADDR_ARM926_I(15dto8)* = **'0x00'** *(default)* ➔ **'0x08'**

3.3.9 XSRST function at active XRESET

The XSRST signal (JTAG interface) is combined with the CTRL-STBY function (ERTEC 200P pin: CTRL_STBY0).
If the CTRL-STBY function is activated, the related output signals will be set in tristate (high impedance).

If the XRESET signal is connected with the CTRL_STBY0 pin (on the PCB), the ERTEC 200P pin XSRST will be set in tristate if the XRESET signal is activated.
In this case the signalling of the active XRESET signal to the debugger is not possible.

⇨ The debugger has to recognize an actvie XRESET signale by different means and shall perform a reset. In the normal operation mode (XRESET signal is not active) a reset of the ERTEC 200P via the XSRST signal of the debugger is still possible, see also clause 3.8.2.3

3.3.10 EDC reporting

To improve system reliability ERTEC 200P memories are EDC (Error Detection and Correction) protected (1Bit Error is correctable, 2Bit Error are detectable).
EDC protected RAMs need an initialization, which is done by hardware at startup; status is indicated in register EDC_INIT_DONE.
This feature is enabled by default and could be disabled for ARM TCMs by setting EDC_PARITY_EN register for other RAMs it could not disabled.

If an EDC error is detected on ARM TCMs an Undefined Instruction Exception (ITCM) or Undefined Data Exception (DTCM) is generated, if other RAMs are hit IRQ48 (EDC_Event) is asserted and register EDC_EVENT must be checked for source.

An interrupt service routine for IRQ48 (EDC_Event) must be setup on ARM926 to handle this error.

Note: When disabling TCM EDC logic, FIT rate regarding SoftError-Rate exceed the specified value >1000FIT@2000m.

3.3.11 Cache-Parity-Check

I-/ D-Cache and associated Tag RAM are parity protected. The feature must be enabled **after** initialization of caches by setting I_CACHE_PAR_EN = 1 and D_CACHE_PAR_EN = 1 in register EDC_PARITY_EN. Additional an interrupt service routine for IRQ48 (EDC_Event) must be setup on ARM926 to handle this error.

3.3.12 Access Error

ERTEC 200P has an internal access monitors, which observes accesses to memory gaps on EMC-, AHB-, APB-bus and TCMs. Following interrupts on ARM926 ICU will be generated:

- IRQ53 (QVZ_EMC_ADR) when an EMC Address Error occurs
- IRQ52 (QVZ_APB_ADR) when an APB Address Error occurs
- IRQ51 (QVZ_AHB_ADR) when an AHB Address Error occurs

The error address and additional information (master of access, type …) are latched up in SCRB QVZ register for further handling.

Additional some modules have also an internal access monitor, which observes access to register gaps. If such an access happens, IRQ86 is generated and MODUL_ACCESS_ERR register hold detailed information about

Finally TCM memories have a separated monitoring. If there are accesses to TCM memory gaps following interrupt will generated:

- IRQ88 when access to gap in I-TCM ARM926 occurs
- IRQ89 when access to gap in D-TCM ARM926 occurs

Care must be taken to handle all these ISR.

3.3.13 ARM926 Watchdog

Watchdog timer should be used to prevent a hardware fault or program error.

3.3.14 PLL LOCK

At startup the software should check that PLL is locked by checking LOCK flag in PLL_STAT_REG register.

3.3.15 PLL LOSS

ERTEC 200P has an internal PLL monitor; when PLL is out of specified range IRQ49 is generated. An interrupt service routine must be setup on ARM926 to handle this error.

3.3.16 Software Reset

For faster startup after ARM926 software reset primary bootloader will jump directly to RETURN_ADDRESS specified in RES_SOFT_RETURN_ADDR.

Note: The RETURN_ADDRESS in RES_SOFT_RETURN_ADDR register has to be set, otherwise an exception (fetch code from 0x00000000) will generated.

3.3.17 XHIF IO driver strength

If XHIF interface operates at 1,8V; driver strength hat to be set to 9mA (register DRIVE47_32GPIO, DRIVE63_48GPIO, DRIVE79_64GPIO and DRIVE95_80GPIO).

3.4 ARM926 subsystem

An ARM926 subsystem is used. The ARM926 subsystem is mainly available for the application and the non-runtime-critical routines of the PN stack. Figure 5: ARM926 subsystem shows the structure of the ARM926 subsystem. It consists of a core system, the JTAG interface and a TCM_Block_926. In addition to the ARM926EJ-S processor the core system contains a Data and an Instruction Cache, a Memory Management Unit (MMU), separate interfaces to the respective AHB layers for instruction and data, a trace macro cell ETM9 (medium+). The Tightly Coupled Memories for data and instruction are located in the TCM_Block_926. AHB access to the D-TCM is effected through a 'DMA-DTCM Access Controller'. The I-TCM is only operated from the ARM926EJ-S and cannot be reached by the AHB. The ARM926EJ-S is described in detail under /7/.

Of the AHB masters the PN-IP, Host Interface and the GDMA can access the D-TCM in the ARM926 subsystem. The ARM926EJ-S has access to all the AHB slaves.

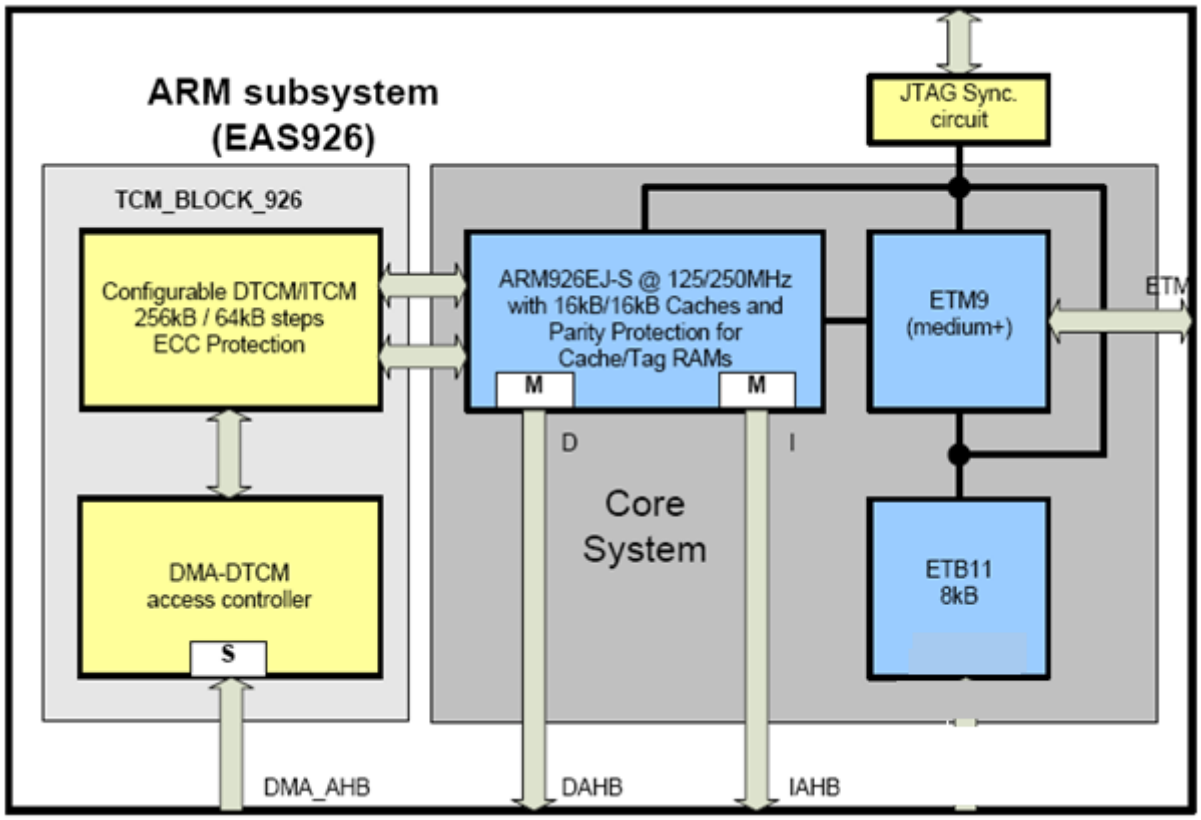


Figure 5: ARM926 subsystem

3.4.1 ARM926EJ-S processor

The ARM926EJ-S revision r0p5 contains an ARM9E-S processor core with Harvard architecture. This processor core has the improved v5TE architecture in contrast to the ARM9 processor core of the ARM9TDMI. These improvements mainly apply to the points improved "ARM/Thumb interworking" (faster changing between ARM and Thumb Code segments) and improved Multiplier Structure. In addition to the processor core the ARM9E-S has an Embedded ICE Logic RT. This logic unit is controlled through the integrated JTAG interface. Document /9/ describes the interfacing of the Embedded ICE Interface Connector to the

ERTEC 200P (reset, JTAG connector , etc.). A detailed description of the ARM9E-S is available in /6/.

The processor and the TCMs (see Chapter 3.4.3) can be operated with 125 MHz or 250 MHz. Configuration is carried out through the CONFIG(1) pin (see Chapter 4.2). The AHB interfaces always run at 125 MHz.

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

The following bit has to be set at the ARM926EJ-S core in order to recognize "Unaligned Access":

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

3.4.2 Cache structure of the ARM926EJ-S

The functional scope of the cache of the ARM926 can be described by the following properties:

- 16 kbytes Instruction Cache
- 16 kbytes Data Cache
- "write buffer" for write-back function of the Data Cache
- Caches are 4 way set associative caches with 1 kbyte segments
- A segment consists of 32 lines and each line contains 32 bytes (meaning 8 words).

The content can be locked at the cache segments. This lock functionality makes it possible to keep the command set for rapid routines permanently in the I Cache. This mechanism can only be carried out segment-granularly at the .ARM926EJ-S. Further information about caching is available in /7/.

If a time-out (QVZ) interrupt occurs an an I Cache Refill through an AHB Error (caused. for example, by a WRAP8 at the memory end), no Prefetch Abort Exception is triggered at the ARM926EJ-S. The allocation to the faulty word is no longer possible for the ARM due to I Cache. However, it is advisable to configure the MMU correspondingly so that an exception to the ARM926EJ-S is generated in this case. In this case an MMU Exception is triggered when the memory area is left.

The I- and D-Cache has a byte parity. The I- and D-Tag RAM is also parity-protected. If an error occurs while reading on the I-Cache or the corresponding I-Tag entry, the error cause (I_Cache_Parity, I_Tag_Parity) is stored in the SCRB register 'EDC Event' (see Chapter 5.3.8) and the interrupt 'EDC_Event' IRQ48 is triggered (see Chapter 5.4.1). An error while reading on the D-Cache or the corresponding D-Tag entry causes the entry (D_Cache_Parity, D_Tag_Parity) in the 'EDC Event Register' with the interrupt 'EDC_Event'. To delete the EDC Event Register has to be overwritten with '0h'.

After a reset the parity bits are undefined. The software has to initialize the caches (each cache line has to be initialized in a loop). Up to this moment the parity logic is disabled. Control is carried out through the SCRB register 'EDC_PARITY_EN' (see Chapter 5.3.8). The 'I_CACHE_PAR_EN Bit' enables the parity bits of the I-Cache and I-Tag and the 'D_CACHE_PAR_EN Bit' enables the parity bits of the D-Cache and the D-Tag. After a reset both bits are set to '0' and the parity bits are disabled. After the cache initialization the software has to set these bits to '1'.

3.4.3 ARM926 Tightly Coupled Memories (ARM926_TCM)

The I- and D-TCM are located in the TCM_Block_926. The TCM configuration encompasses 256 kbytes for Instruction and Data TCM (I/D-TCM). The memory operates with the cycle rate of the processor 125/250 MHz. The memory consists of segments of 64 kbytes that can be assigned to either the Instruction or the Data TCM. This results in a configurable memory

size of 0 – 256 kbytes for the I-TCM at 256 – 0 kbytes for the D-TCM. After the reset the TCM926 configuration is set to 256 kbytes D-TCM. The final settings is carried out during booting by the Bootloader (software) in the SCRB register 'TCM926_Map' (see Chapter 5.3.8). After this reconfiguration the bootloader still has to display the TCM memories in the coprocessor interface (CP15 c9) in the address area (see Chapter 5.1.1, 5.1.2). Note that the I-TCM and D-TCM can only be displayed in the address area in a size of 2^n steps. If, for example, a physical size of I-TCM = 192 kbytes / D-TCM = 64 kbytes was selected in 'TCM926_Map Register', the ARM926 can only assign an address area of 256 kbytes to the I-TCM and the desired address area of 64 kbytes to the D-TCM. The values 256 kbytes for I-TCM and 64 kbytes for D-TCM are therefore to be read out in the coprocessor interface of the ARM926 (CP15 c9).
The software may then not access the unassigned area / hole (I-TCM: 192 – 256 kbytes). If access is carried out in the unassigned area from ARM926, either the 'Invalid I-TCM926 Access Interrupt' (access in the hole of I-TCM) or the 'Invalid D-TCM926 Access Interrupt' (access in the hole of D-TCM) is triggered (see Chapter 5.4.1).
Unpredictable accesses in the I/D-TCM hole can not be intercepted by the ARM926, the 'Invalid I/D-TCM926 Access Interrupt' is generated. The situation can be mastered if the MMU codes out the hole and 'Invalid I/D-TCM926 Access Interrupt' is blocked. The MMU does not see any unpredictable accesses.
In the implementation an address area of 256 kbytes is assigned respectively for the I-TCM and the D-TCM. This ensures all the possible combinations of the distribution between I-TCM and D-TCM. After the insertion the I-TCM lies in the ARM926 address area from 0x0000_0000h and the D-TCM from 0x0800_0000h. Further information about the TCMs is available in /7/.
Only the ARM926 D_TCM can be reached from the AHB. If an access into the unassigned range of D-TCM is carried out from an AHB master, access is prevented and a time-out (QVZ) interrupt is triggered (see Chapter 3.10.1).
AHB access to the D-TCM is effected through the 'DMA-DTCM Access Controller'. The I-TCM is only served by the ARM926EJ-S and cannot be accessed by the AHB.

The boot process for the I-TCM is effected as follows:

In the boot ROM a primary bootloader runs that loads a secondary bootloader from the boot medium into the D-TCM in such a manner that after the switchover to the I-TCM the secondary bootloader is located behind the interrupt entry table. If an external host is responsible for booting, it transfers the secondary bootloader into the D-TCM.
After the secondary bootloader has been loaded, this 64 kbyte segment is switched over to the I-TCM. The loaded secondary bootloader is not lost during the changeover to the I-TCM. Figure 6 shows the course of the loading of a code image into the D-TCM (for example secondary bootloader). After the PowerOn Reset the TCM926 configuration is set to 256 kbytes D-TCM. Since the assignment of the SRAM blocks in the I-TCM and D-TCM memory is completely twisted, the bootloader has to fill up the code image in 64 kbyte blocks from the lower D-TCM block upwards. The address order is not twisted in the respective TCM block. After the download to the D-TCM the final TCM926 configuration is set in the TCM926_MAP register (see Chapter 5.3.8). Subsequently the bootloader has to insert the TCM memories in the coprocessor interface (CP15 c9) into the address area.
The primary bootloader then hands over to the loaded secondary bootloader in which the user then continues the application-specific booting process. One task of the secondary bootloader is to fill a code images into the I-TCM. Two options are available to this purpose:

- The secondary bootloader sets the final required TCM926 configuration in the TCM926_MAP register (see Chapter 5.3.8). Afterwards the code image can be transferred from the boot medium into the I-TCM behind the secondary bootloader.
- First the 64 kbyte segment that is already assigned to the I-TCM is filled completely. Afterward the code image is loaded further into the D-TCM in the correct order (see Figure 6). After the download to the D-TCM the final TCM926 configuration is set in the TCM926_MAP register (see Chapter 5.3.8). The secondary bootloader still has to also insert the final configuration of the TCM memories into the coprocessor interface (CP15 c9) into the address area.
  The GDMA can also be used to load the code image into the D-TCM. In the case of a booting process from the host it will load the code image into the D-TCM.

The primary bootloader has to set up a page on the D-TCM (boot RAM) in the 8 page registers of the XHIF for booting from an external host. The remaining pages can subsequently be set up specifically by the external host. It is advisable to lay further pages on the PN-IF, PER-IF, EMC-SDRAM / EMC-SRAM, GMDA and APB Peripherals.

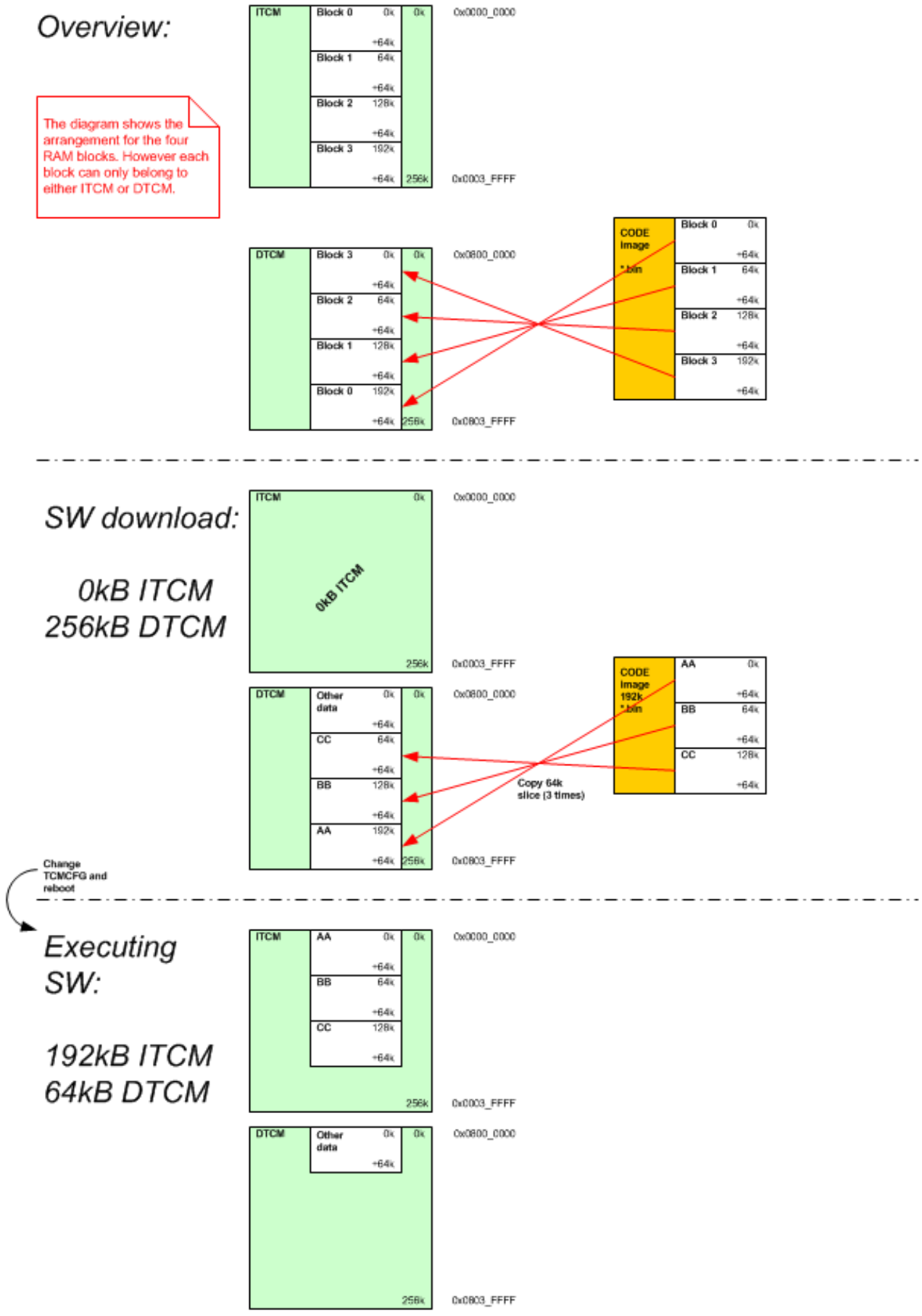


Figure 6: Booting the I-TCM ARM926EJ-S

The I- and D-TCM is provided byte-by-byte with an EDC code (5 bits per byte, 1Bit Error correctable, 2Bit Error recognizable). If an error occurs while reading, the error cause (I/D-TCM926-1B: 1Bit Error corrected or I/D-TCM926-2B: 2Bit Error recognized) is stored in the SCRB register 'EDC_EVENT' (see Chapter 5.3.8) and the interrupt 'EDC_Event' IRQ48 is triggered (see Chapter 5.4.1). To delete the EDC Event Register has to be overwritten with '0h'.
After the ARM926-TCM reset (assignment see Figure 29: Reset matrix of the) an initialization of the complete TCM is carried out automatically by the hardware. The initialization process for the complete I-/D-TCM amounts to approx. 16.5 µs. Completion of this initialization is signaled in the SCRB Register 'EDC_INIT_DONE' (see Chapter 5.3.8) (I_TCM926_INIT_DONE, D_TCM926_INIT_DONE). The EDC logic can also be disabled through the SCRB Register 'EDC_PARITY_EN' (see Chapter 5.3.8). The bit 'EDC_DISABLE_ARM926' disables the EDC logic, but the EDC logic always have to be activ. After the reset the EDC logic is enabled.
The D-TCM can be written and read by the AHB masters (PN-IP, Host Interface, GDMA). During access by one of these masters to the D-TCM the ARM is stalled (pipeline halted). As a rule this stall lasts only one processor cycle per word transfer to carry out the transfer from the 'DMA-DTCM Access Controller' (part of the ARM926 subsystem) to the D-TCM and vice versa. Arbitration is carried out in round-robin scheduling, meaning that the accesses to the D-TCM are always carried out alternatively by the ARM and the 'DMA-DTCM Access Controller'. A new arbitration is also carried out at a burst transfer after every single access.

Table 2 lists the various ARM926EJ-S access types with the possible errors I./D-TCM and the 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

Table 2: ARM926EJ-S Access types to I-/D-TCM / AHB

3.4.4 Memory Management Unit (MMU)

The MMU (see ARM926EJ-S Technical Reference Manual /7/) supports a 'demand page virtual memory' system which is required by operating systems such as Linux or eCos depending on the application. The MMU contains the access protection mechanisms for all the memory accesses. The address translation, the access protection and the region type are stored in a TLB (Translation Lookaside Buffer). Separate TLBs are available for Instruction and Data. These TLBs are automatically evaluated or updated respectively by the MMU hardware.

- Page size: 1 Mbyte, 64 kbytes, 4 kbytes and 1 kbyte
- Separate TLBs for Instruction and Data
- Access attributes can be changed without a TLB Flush

- Fast Context Switch allows a virtual address remapping in a range of 0 – 32 Mbytes
- TLB entries can be locked

The MMU RAMs do not have a parity!

3.4.5 Bus interface of the ARM926 processor

The ARM926 processor has a separate AHB interface for Opcode fetches (ARM926-I) and data transfers (ARM926-D). The interfaces operate with 125 MHz. The databus and address bus width amounts to 32 bits each. The data bus interface operates via a "write buffer" (16-step 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. Further information about the bus interface and about the "write buffer" is available in /7/.

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

For debug support an ETM9 revision r2p2 (medium+ module with integrated trace buffer (ETB11), that allows an Instruction and Data trace, is activated in the ERTEC 200P at the ARM926EJ-S system. The trace buffer has a size of 2Kx32 bits and supports both ARM926 clocks (125 / 250 MHz).
The ETM module receives all the signals necessary for the Data and Instruction trace from the processor. The ETM9 module is operated through the JTAG interface. The trace information is entered via an FIFO in the internal trace buffer.
Only the half-rate mode is supported as the clock mode. In this case the trace clock runs at half the processor frequency (62.5 MHz at 125 MHz processor cycle) and the trace data have a maximum change frequency of 125 MHz. The debugger accepts the data with both trace clock edges.
A detailed description of the ETB11 and the corresponding programming interface is provided in the document 'ETB11 Technical Reference Manual (ARM DDI0275B)' /46/.

3.4.7 Debug configuration

2 TAP controllers are circuited in the JTAG chain (see also Chapter 3.4.8):

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

This results in the following chain:

- TDI → ETB → ARM926EJ-S → TDO

The number of 'Instruction Register Bits' amounts to (see Figure 7):

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

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

Figure 7: JTAG chain

3.4.8 Debug Interface ARM926EJ-S

Only external debuggers with JTAG interface are supported.

To this purpose the pin 'CHAIN_CTRL' is defined as the VDD supplier.
The JTAG interface encompasses the standard signals (XTRST, TCK, TDI, TMS, TDO and XSRST). In addition the RTCK Clock output is made available. A JTAG cycle rate of TCK = 32 MHz is possible for debuggers that support RTCK. Otherwise the maximum cycle rate only amounts to TCK =16 MHz.

The two signals DBGREQ and DBGACK are furthermore supported via the JTAG interface for the ARM926EJ-S. When DBGREQ is active, the external debugger forces the ARM926EJ-S directly into the debug state. DBGACK active signals the debugger that the ARM926EJ-S is in the debug state. Both signals lie as an alternate function on the GPIO31-0 pins. Before these debug signals can be used, they have to be enabled in the GPIO31-0 block. If DBGREQ is not enabled in its alternate function, DBGREQ is switched inactive internally and the ARM926EJ-S is not in the debug state.

3.5 ERTEC 200P internal busses

The ERTEC 200P has 2 internal bus systems. These are a high-performance communication bus (multi-layer AHB) and an I/O bus (APB).

All 6 masters (ARM926-I, ARM926-D, PN-IP-M1, IRT PN-IP-M2, GDMA Controller, Host Interface) and the slaves (D-TCM926, Interrupt-Controller (ARM-ICU), GDMA-Register/Memory, External Memory Controller (EMC), APB Bridge, Peripheral Interface (2 connections) and PN-IP are connected directly to the multi-layer AHB that is characterized by a high data transmission rate and bus availability.

Through an AHB/APB bridge the masters can access the remaining peripherals that are connected to a low-performance I/O bus (APB). The AHB/APB bridge is the only master at the I/O bus.

3.5.1 The communication bus (multi-layer AHB)

The multi-layer AHB bus is a 32-bit-wide multi-master-compatible bus. It runs with 125 MHz and has the functionality of the ARM AHB Lite bus.
6 AHB masters can access various slaves simultaneously by switching together several AHB segments in the "multi-layer AHB" (see Chapter 3.1).
In the case of simultaneous access by several masters to one slave an arbiter decides to which master this slave is assigned.
Due to the multi-layer architecture there is no retry functionality at the AHB (AHB Lite). Similarly the split functionality is not supported (AHB Lite).

The lock functionality can be enabled for the the ARM master (ARM926-D) in the SCRB Register M_LOCK_CTRL. By default the lock functionality is disabled. The ARM926-I master cannot generate a lock.

The APB module allows the following access:

AHB-module (Slave)	allowed access	note
PN-IP	8/16/32 bit	databus: 32 bit
Peripherie-Interface	8/16/32 bit	databus: 32 bit
EMC Memory	8/16/32 bit	databus: 32 bit
EMC Register	8/16/32 bit	databus: 32 bit
GDMA (RAM + Register)	only 32 bit	databus: 32 bit
ARM-ICU	only 32 bit	databus: 32 bit
ARM926 - TCM	8/16/32 bit	databus: 32 bit

3.5.1.1 AHB arbiter

Each of the AHB arbiters uses the same arbitration process. By default round-robin is set. A fixed priority assignment of the AHB masters (see 5.3.8) can be set as an alternative arbitration algorithm by programming the bit ARB_MODE in the SCRB Register M_LOCK_CTRL (see Table 3). However, this should not be done in view of the dynamic processes at the multilayer AHB.
With round robin as the arbitration process reciprocal blocking of the AHB masters at the multi-layer AHB is prevented for a longer period (see Section3.5.1.2).

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

Table 3: Fixed priority assignment (no default)

3.5.1.2 The AHB master-slave coupling

The respective masters are not connected with all slaves within the "multi-layer AHB". The following table shows which AHB master is connected with which AHB slave.

AHB-Master	D-TCM 926	ARM-ICU	GDMA	EMC	APB	PER-Host	PER-PN	PN-IP
ARM926-D (AHB Burst Breaker)		X	X	X	X	X		X
ARM926-I (AHB Burst Breaker)				X	X			
GDMA	X		X	X	X	X		
Host Interface	X		X	X	X	X		X
PN-IP-M1	X			X				
PN-IP-M2							X	

Table 4: AHB master slave coupling

Note:
The AHB masters may not keep a slave arbitrated for too long to ensure that long blockades do not occur during access to a slave. This could happen, for example, when the ARM926-D master has too many transactions to the EMC slave in its write buffer. Another master (for example Host Interface or PN-IP-M1) would be blocked for this time in accessing the EMC

slave and would have to wait in wait mode until the ARM926-D master releases the EMC slave again.
The ARM926-D and ARM926-I masters have a burst breaker at the AHB access that enforces an idle phase for a clock after a number of consecutive address phases that has to be configured. In this idle phase a different AHB master can the access the AHB slave. This is possible because rearbitration at an AHB slave can only be carried out when a master enables this AHB slave (idle phase). As the burst breaker is disabled per default, the application has to enable the burst breaker, see application note, claus 3.3.8.

Note:
All the AHB participants operate in Little-endian mode.

3.5.2 The I/O bus (APB)

The APB bus is a 32-bit-wide bus. The bus frequency amounts to 125 MHz. The APB bus in the ERTEC 200P has an extended functionality compared to the ARM APB bus (see Chapter 3.7). Compared to the standard APB bus it provides the option of using variable timing (Wait-states) and byte enables. The waitstate and byte enable capability is attained by inserting sideband signals. This makes is possible to also access peripherals byte-by-byte.

Only the AHB/APB bridge has master functionality on the APB. All the peripheral blocks of the APB are operated as a slave.
The AHB/APB bridge has a one-level write buffer.

The APB module allows the following access:

APB-modul	allowed access	note
SCRB	only 32 bit	databus: 32 bit
Boot-ROM	8/16/32 bit	databus: 32 bit
Host-Interface (XHIF)	only 32 bit	databus: 32 bit
UART	16/32 bit	databus: 16 bit, i.e. the upper 16 bit will be ignored
F-Timer	only 32 bit	databus: 32 bit
Timer 0-5	only 32 bit	databus: 32 bit
Watchdog	only 32 bit	databus: 32 bit
I2C	8/16/32 bit	databus: 8 bit, i.e. the upper 24 bit will be ignored
SPI	16/32 bit	databus: 16 bit, i.e. the upper 16 bit will be ignored
GPIO	only 32 bit	databus: 32 bit
I-Filter	8/16/32 bit	databus: 32 bit
Peripherie-Interface	8/16/32 bit	databus: 32 bit

3.6 Peripherals at the communication bus (multi-layer AHB)

The function blocks connected to the AHB all have 32-bit-wide interfaces.

3.6.1 ARM926 Interrupt Controller (ARM-ICU)

3.6.1.1 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 of interrupts
   - Settings for trigger modes
   - Processing of software interrupts, etc.

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

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

3.6.1.2 Interface

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

The ICU supplies a separate output signal to serve both interrupt levels of the CPU (FIQ and IRQ). The interrupt processing within the ICU is performed in separate subblocks for FIQ and IRQ. **The output signals for the CPU are level-triggered and "active high". The signals are inverted on the Toplevel before connecting to the appropriate inputs of the ARM926EJ-S.** 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 IRQ subblock supports 96 interrupt inputs; the FIQ subblock supports 8 interrupt inputs.

Each interrupt source of the IRQ subblock can be placed on each interrupt input of the FIQ subblock, where the selected interrupt sources also continue to remain active in the IRQ subblock. The "FIQ0" is the exception, because no IRQ can be placed on it. For the configu-

ration of the assignment by the software, the numbers of the selected IRQ inputs are entered in the FIQ_SEL1 – FIQ_SEL< 8 - 1> registers. Although the FIQ_SEL0 register can be read and written, it is not further processed by the hardware.

Note that both IRQs and FIQs 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 (IRQ or FIQ) 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.** IRQ and FIQ must always be set to the same triggering. Because this is not checked by the ICU, it must be ensured by the software. In addition, the software should normally mask the IRQ input in the IRQ subblock used as FIQ, because otherwise both an IRQ and an FIQ will be signaled to the CPU when the interrupt occurs.

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

The interrupt with number "0" may not be used either in the IRQ subblock or in the FIQ subblock. "IRQ0" **outside** the ICU is clamped to the inactive level; the "FIQ0" **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.

3.6.1.3 Interrupt aquisition

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 ISREG has been cleared by the EOI command and no higher-priority interrupt is present.
  - When in level-triggered mode an interrupt is removed by the source before the CPU can confirm it with an acknowledge, however after the removal of the interrupt the CPU nevertheless performs an acknowledge, the ICU returns the default vector (null vector) as interrupt vector.

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

An interrupt detected at the input will be entered in the Interrupt-Request register. If no Interrupt mask is set, the Interrupt will be forwared 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/FIQ_SWIRREG). After setting the bit, no minimum time must be observed for the renewed clearing of the bit.

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

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

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

3.6.1.4 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 FIQ_MASKREG register. The interrupts are masked at the outputs of the Interrupt-Request register (IRREG or FIQ_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 und thus all interrupts disabled.

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

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

Not only each IRQ but also each FIQ receives its own freely selectable priority. The software can set the priority using the PRIOREG0 – PRIOREG<96 - 1> or FIQPRIOREG0 – FIQPRIOREG<8 - 1> register. The value range for the priorities is:

- "0" to 96 - 1 for IRQs and
- "0" to 8 - 1 for FIQs.

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

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

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

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

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

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

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

A pending and valid interrupt leads to an active IRQ output. A pending and valid fast interrupt leads to an active FIQ output.

3.6.1.5 Interrupt post-processing

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

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

A default vector can occur:

1. When the ICU receives an acknowledge command without it previously outputting a pending interrupt.
2. When the source of a level-triggered interrupt removes it before being confirmed by the CPU.
3. When an interrupt input is masked or a clear command is performed for an interrupt input while concurrently an interrupt occurs at this input. Because the CPU commands

arrive at the interrupt controller with a time delay, the interrupt is initially forwarded to the CPU, however, removed after the command takes affect. If, however, the CPU responds with an acknowledge, the interrupt controller can only assign the default vector, because in the meantime a valid interrupt is no longer present.

4. If the time between acknowledgment and the End of Interrupt event is less than 3 ICU clock cycles, the Interrupt line to the CPU may rise again, although there is no interrupt present. In this case the CPU will read the default vector.

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

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

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

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

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

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

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

3.6.1.6 Special functions

The ICU permits the additional confirmation of an interrupt by any write access to the FIQACK/IRQACK register. This function is deactivated after reset and must be activated prior to use with the UNLOCK_RD_ONLY_ACK register bit (for each IRQ/FIQ block seperately).

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 FIVEC 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 (FIQ/IRQ) has its own version register. The ICU ID can be obtained from the register descriptions.

3.6.1.7 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.

An interrupt that was reported to the CPU using the IRQ or FIQ signal is acknowledged with a read access to the interrupt vector register (IRQACK or FIQACK). 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' (coming from the ARM926) that can act directly on the global mask bit. This procedure can be enabled and disabled using the MASK_ALL_INPUT_EN software register.

3.6.1.8 Miscellaneous

3.6.1.8.1 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 has to be high active. **Low Active level triggered interrupts are not supported by ICU**.

The CPU can use the DBG_ACK signal (coming from the ARM926) 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.

3.6.1.8.2 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.

3.6.1.8.3 Sequences

The following is a typical interrupt cycle:

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

pending interrupt has a higher priority, the ICU activates the interrupt output in the direction of the CPU.

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

5. The IRQACK register is read, which indicates the interrupt that currently has the highest priority and wants to interrupt the CPU.
6. The reading of the IRQACK 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.

3.6.1.8.4 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 FIQs with write access to acknowledge register

The software also has the option of carrying out the acknowledgement of the FIQ interrupt via a write access to the FIQACK 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 IRQs as FIQs

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

Identical priorities

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

Hardware latencies in certain back2back software accesses

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

IRCLVEC -> IRQACK

IRCLVEC -> IRREG1-5

IRCLVEC -> SWIRREG1-5

LOCKREG -> IRQACK

MASK -> IRQACK

IRQACK->IRREG1-5

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

Deletion of interrupts during the reset phase

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

3.6.1.8.5 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 should also be determined which interrupts are level/edge-triggered and stated which edge is the active one, if necessary. If a priority locking is to be carried out, this has to be configured correctly and activated before.

Furthermore, it has to be taken into account that IRQs as well as FIQs 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 (IRQ or FIQ) at the CPU after the acknowledgement.

The IRQ applied to the FIQ always has to be set to the same triggering. This is not checked by the ICU and has to be ensured by the software. Moreover, the SW should normally mask

the IRQ input in the IRQ sub-block used as FIQ since otherwise an IRQ as well as an FIQ is signaled to the CPU upon the occurrence of the interrupt.

3.6.1.9 Interrupt sources for ARM-IRQ

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

interrupt sources	No. of interrupts	Used by the PROFINET stack	Usable by the application	comments
Default interrupt	1	no	no	Interrupt which is never assigned. **Tied to '0' on toplevel**
GDMA-Controller	1	no	yes	Job finished or DMA Error
I2C (@APB)	1	no	yes	-
UART 1 - 4	8	no	yes	4x Combined – and Error interrupts
SPI 1 - 2	8	no	yes	2x Combined, Receive Overrun, Transmit FIFO empty and Receive FIFO not empty interrupts
Reserved	4	no	no	-
Timer 0 - 5	6	yes	yes	Timer 0 is used by the operation system of the Evaluation Kit
Watchdog	1	no	yes	ARM926 Watchdog
GPIO	16	no	yes	16x External interrupts
EDC_Event	1	no	yes	Combined interrupt for EDC Correction or Detection
PLL-Lock	1	no	yes	PLL Lock State
PLL-Loss	1	no	yes	PLL Loss State
AHB address error	1	no	yes	AHB access to unused memory area
APB address mismatch	1	no	yes	Invalid APB address access request
EMC address mismatch	1	no	yes	EMC-QVZ
Peripherie Interface	1	no	yes	Event Unit
PHYs	1	yes	no	Combined interrupt (PHY1 / PHY2)
PN-IP (PN-ICU2)	2	yes	no	Combined interrupt from PN-IP (PN-ICU2)
PN-IP (PN-ICU2)	14	yes	no	14 selectable interrupts in PN-IP (PN-ICU2)
PNPLL	6	yes	no	6 selectable interrupts in PNPLL (PN-IP)
SW Int	8	no	yes	Software interrupt
Modul_Access_Error	1	no	yes	Combined interrupt for address missmatches in modules
Reserved	6	no	no	-
Invalid I/D-TCM926 Access	2	no	yes	Invalid ARM926EJ-S Access to missing addresses of the I/D-TCM
Reserved	3	no	no	-
Sum of all Interrupts	96	-	-	-

Table 5: Interrupt sources for ARM-IRQ

Further informationen for all IRQs see Chapter 5.4.1.

3.6.1.10 Interrupt sources for ARM-FIQ

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

interrupt sources	number of interrupts	Used by the PROFINET stack	Usable by the application	comments
Default interrupt	1	no	no	Interrupt which is never assigned. **Tied to '0' on toplevel**
Selectable interrupt from the IRQ-Sources	7	no	yes	selectable by parameter
Sum of all Interrupts	8	-	-	-

Table 6: Interrupt sources for ARM-FIQ

3.6.1.11 Interrupts for accesses to missing addresses

Accessing to missing addresses generate a QVZ pulse. This pulse will be processed in the Interrupt Controller as an edge-triggered interrupt.

3.6.1.12 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 5.3.6) after ((MAX_EXT_WAIT + 1) x 16) AHB clock pulses creates an internal ready signal for the Memory Controller and an IRQ. The IRQ will be removed when the QVZ monitoring is disabled.

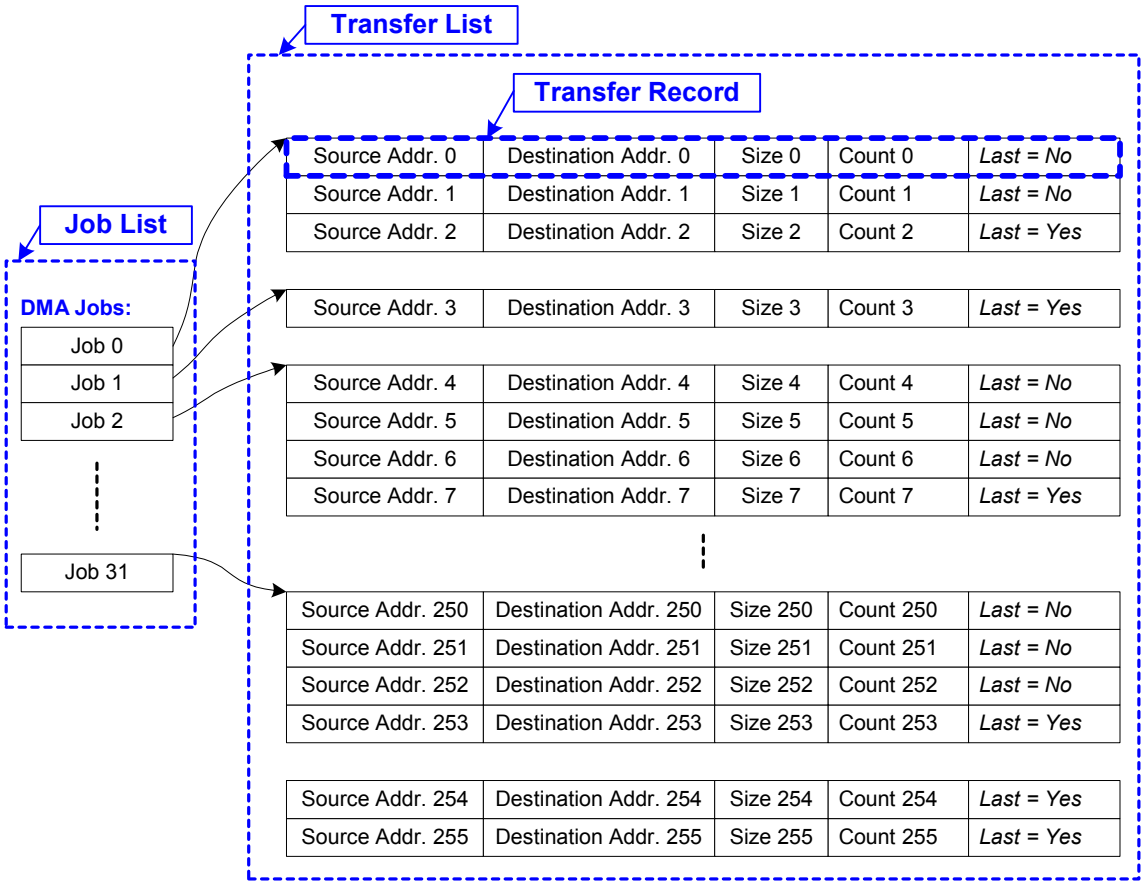
3.6.2 Host Interrupt Controller

The Host Interrupt Controller is included in the PER-IF (Host Event Unit, see Chapter 3.6.7).

3.6.3 GDMA – Generic Direct Memory Access Controller

3.6.3.1 DMA: Functional 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 8.
The transfer list is saved in the DMA RAM, which is in an internal RAM. The Job List base address is saved in a GDMA register. The number of transfers per job is programmable; the maximum number of jobs, supported by the GDMA within ERTEC 200P is 32.



Transfer_lists.vsd

Figure 8: The principle of the DMA Job organization

3.6.3.1.1 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.

3.6.3.1.2 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.

3.6.3.1.3 Details

3.6.3.1.3.1 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 PN-IP (3x Application Timer Block Modul), the Timer Unit and GPIO input signals. The input selection is programmable through the Job Control registers. The maximum number of job-starting input signals is parameterizable up to 64.

3.6.3.1.3.2 Interfaces

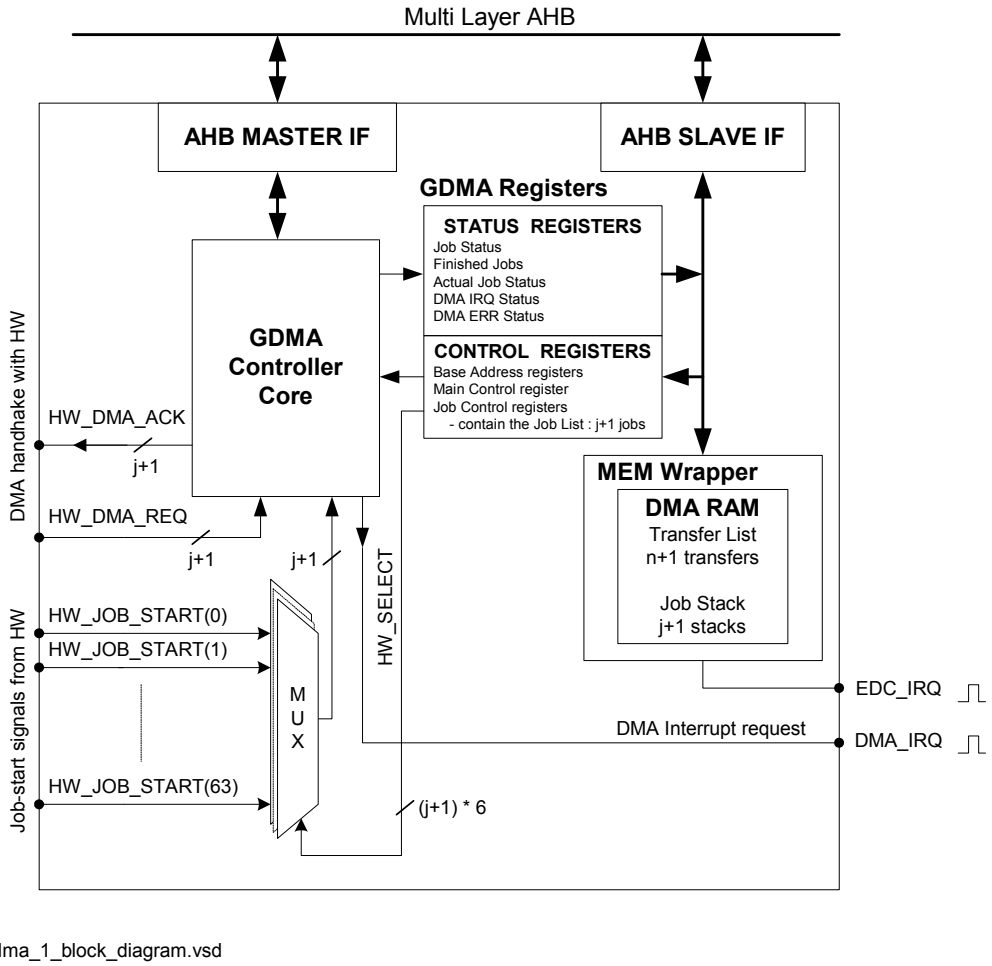
The GDMA controller contains the following interfaces:

- 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 outputs of the PN-IP (3x Application Timer Block Modul), the Timer Unit and GPIO input signals.
  Note that start of a job by HW is enabled by means of bit HW_JOB_START_EN of the Job Control register.
- DMA request by HW signals – For a job dedicated to data transfer from a peripheral, the HW DMA request signal must inform, if data is available on the peripheral. For a job dedicated to data transfer to the peripheral, the HW DMA request signal must inform, if the peripheral is ready to receive the data. According to these needs, the HW DMA request signals of the GDMA controller are connected to the corresponding outputs of the peripheral devices. The number of DMA request signals is parameterizable and equals to the maximum number of jobs.
  Note that the ready-checking of the peripheral is enabled by means of bit HW_FLOW_EN of the Job Control register. For every HW DMA request signal input an HW DMA acknowledge output exists, which is asserted at the end of a DMA transfer, if this is enabled in the DMA Transfer Record and the bit HW_FLOW_EN in the job control register is set.

- DMA Interrupt Request (DMA_IRQ) output – If a job is finished or a DMA error occurs (and if the other conditions, described later, are fulfilled), then the GDMA controller generates a DMA interrupt request.
- Error Detection and Correction (EDC) interrupt request – these signals (GDMA-1B: 1Bit Error corrected, GDMA-2B: 2Bit Error recognized) are wired to the EDC_EVENT Register in the SCRB Register Block (see Chapter 5.3.8). Addionaly the interrupt 'EDC_Event' IRQ48 is set (see Chapter 5.4.1). For the purpose of cleanup the EDC Event Register must be written with '0h'. By reading the EDC Event Register all Bits are cleared. After Reset an initialisation of the EDC-Bits is made. The conclusion of the initialisation can be read in the SCRB Register 'EDC_INIT_DONE' (see Chapter 5.3.8).
  These GDMA outputs are internally connected to the outputs of the MEM Wrapper of the internal DMA RAM. Note that this interrupt is used only when the GDMA controller is configured for an internal DMA RAM and the Memory Wrapper is applied.

3.6.3.1.3.3 Block diagram

The block diagram of the GDMA controller is shown in Figure 9.



dma_1_block_diagram.vsd

Figure 9: Block diagram of the GDMA controller

DMA jobs are programmed by means of GDMA Job Control registers and the transfer records of the DMA RAM, as illustrated in Figure 10.

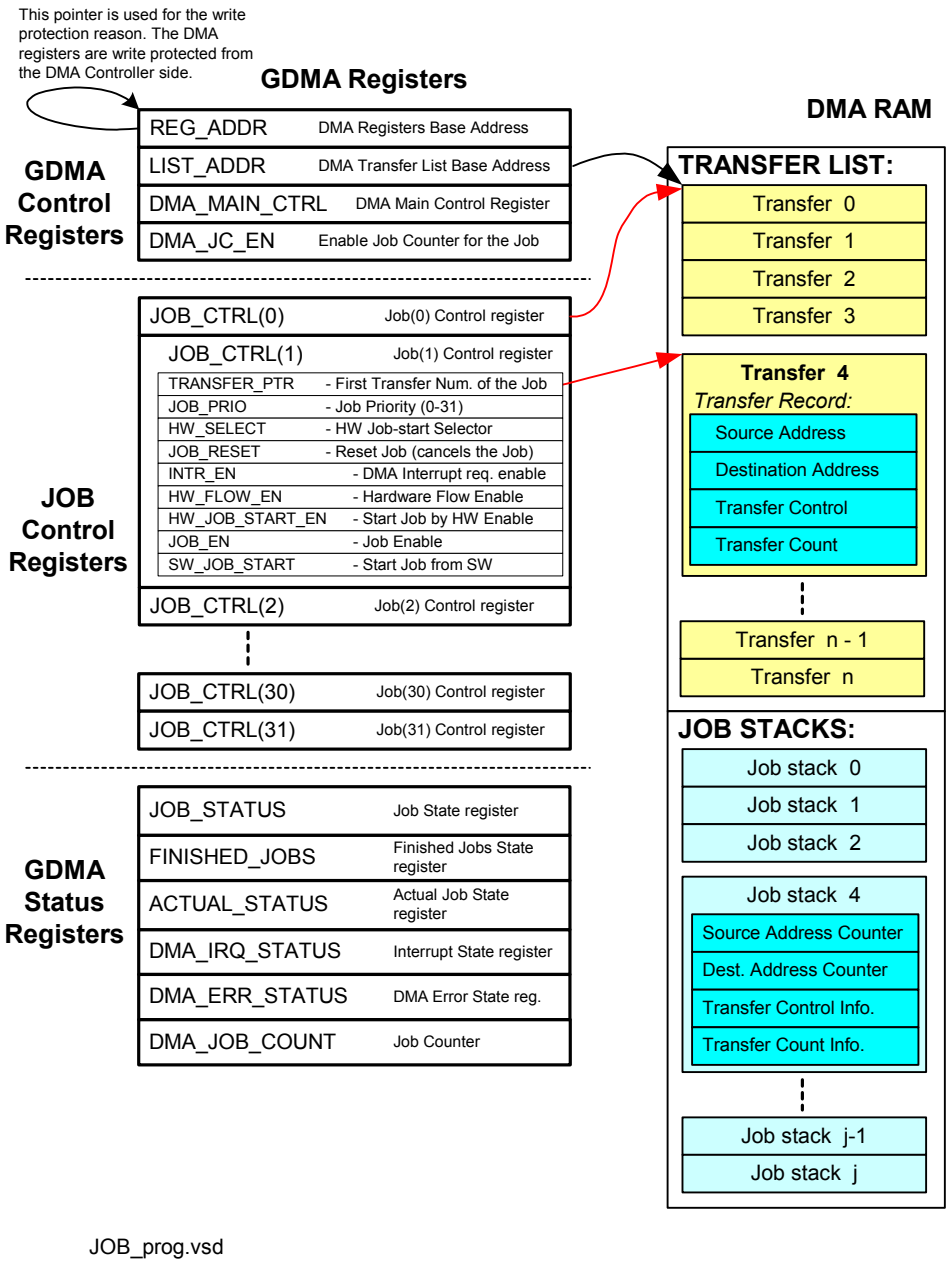


Figure 10: DMA jobs programming

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.

3.6.3.1.3.4 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 10)
- Job Priority (JOB_PRIO) defines the priority of the job. A higher value of JOB_PRIO means higher priority. If two or more jobs have the same value of JOB_PRIO, then the job priority is defined by the job number. In such a case, the smaller job number means the higher priority. When e.g. the JOB_PRIO field has

the same value for each job, the highest priority has the started job with the smallest job number.

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

3.6.3.1.3.5 Transfer list

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

- "Source Address" defines the address of source data for the DMA transfer. This can be either a memory or a peripheral address.
- "Destination Address" defines the address of the destination for the DMA transfer. This can be either a memory or a peripheral address. The addresses of the GDMA registers and DMA RAM are protected against write accesses by the DMA transfers.
  The reason for this is as follows: When the GDMA controller is programmed incorrectly, the data in the GDMA registers and DMA RAM could be corrupted and subsequently the system might fail. When a wrong destination address is programmed, then the DMA destination address error is assumed. When at the same time bit "Error Interrupt Enable" (ERR_INT_EN) in the GDMA Main Control register is "1", the relevant error bit in the GDMA Error Interrupt State register is set to "1" and the GDMA interrupt request (DMA_IRQ) is generated.
- "Transfer Control" consists of three 2-bit fields: Source Address Mode, Destination Address Mode and Burst Mode.
  - The Source Address Mode and Destination Address Mode can be either "increment" or "hold". Note that the second bit of this field serves as a reserve bit.
  - The Burst Mode holds the information about the AHB burst type which will be used, and can be Single, INCR4, INCR8 and INCR16 (see necessary selection under HW_DMA_REQ signals in Table 7).
- "Transfer Count" consists of 4 fields: 1-bit flag "Last Transfer of the Job", 1-bit flag "Enable DMA Acknowledge", 2-bit field "Element Size" and 16-bit field "Transfer Count".
  - Last Transfer of the Job - Indicates the last transfer record in a job (thus the transfers in a job are defined by pointer TRANSFER_PTR in the Job Control

register and the bit “Last Transfer of the Job”.) When no transfer in the Transfer List is marked as the last one, the transfers will be executed in an infinite loop.

- Enable DMA Acknowledge – The output HW_DMA_ACK of a job is set ‘1’, if the input HW_DMA_REQ of this job is 1, and the bit HW_FLOW_EN of the job register is 1, and the current transfer is finished, and the bit DMA_ACK_EN in the Transfer Record is 1. The output HW_DMA_ACK of a job is reset ‘0’, if the input HW_DMA_REQ of this job is 0.
- Element Size - Sets the size of elements to be copied in the transfer and can be 8, 16 or 32 bits. The number of bytes to tranfer is (element size) x (transfer count)
- Transfer count - Holds the information on how many elements have to be copied in this transfer. It can be programmed from 0 to 65535 elements, representing the range of 1 to 65536 elements.

3.6.3.1.3.6 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:
- 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.
- Software reset (SW_RESET) - The started jobs can be reset and the running or interrupted jobs cancelled when SW_RESET bit in the GDMA Main Control register is set.
- 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.
- Reset Job Counter (JC_RESET) resets the GDMA_JOB_COUNT status register. See the description of the GDMA Status registers below.

- 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.

3.6.3.1.3.7 Job control by SW or by HW

SW control of the job:

- Start of job by SW:
  The job is started by SW when bit SW_JOB_START is set in the Job Control register. Start of DMA job is disabled, when bit SW_RESET in the GDMA Main Control register is set to "1" or when bit HW_JOB_START_EN in the Job Control register is set to "1".
- Make a job running:
  When one or more jobs are started, the started job with the highest priority begins to run (becomes active). The running job can be interrupted in three ways:
  1. When a job is running and a job with higher priority is started, the running job is interrupted and the job with higher priority begins to run. After the job with the higher priority is finished, the interrupted job will continue.
  2. When bit JOB_EN in the Job Control register is reset from '1' to "0", a running job is interrupted and a started job with the next, smaller priority begins to run. After this bit is changed back to "1", the interrupted job will be reactivated. JOB_EN = 0 does not cancel a job. It makes the job only sleeping.
  3. When bit DMA_EN is reset while a DMA job is running, this job is interrupted and no new job begins to run. When subsequently bit DMA_EN is set again, the interrupted job will continue. DMA_EN has the same function as JOB_EN, but for all jobs.

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

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

HW control of the job:

- Start of job by HW:
  HW starts the job by rising edge of signal HW_JOB_START. This signal is selectable as one of „n" (max. 64) input signals using field HW_SELECT in the Job Control register and can be enabled/disabled by means of bit HW_JOB_START_EN in the GDMA Job Control register.
  Start of a DMA job is disabled also when bit SW_RESET in the GDMA Main Control register is set to "1". When HW starts a job again, before the running

job is finished and if at the same time bit “Error Interrupt Enable” (ERR_INT_EN) in the GDMA Main Control register is “1”, the relevant error bit in the GDMA Error Interrupt State register is set to “1” and the interrupt request at the DMA_IRQ output of the GDMA controller is generated.

- Make a job running:
  For the started job with the highest priority, dedicated to data transfer from a peripheral, the HW DMA request signal is inquired, whether the peripheral device is ready to send data. For the job dedicated to data transfer to a peripheral, the HW DMA request signal is inquired, whether the peripheral device is ready to receive data. If so, the job starts to run, otherwise the job is interrupted and the next job can start to run. When a job is running and the HW DMA request signal indicates that the peripheral is no longer ready to send/receive data, the job is also interrupted. When the peripheral is ready to send/receive data again, the job will continue. The check, whether the HW peripheral is ready, can be enabled/disabled by means of field HW_FLOW_EN of the Job Control register. Furthermore, the job can be interrupted or cancelled just as in the case of the SW control of the job described above.
- Job finished:
  The job is finished in the same way as in the case of the SW control of the job described above.

The job control is illustrated in Figure 11.

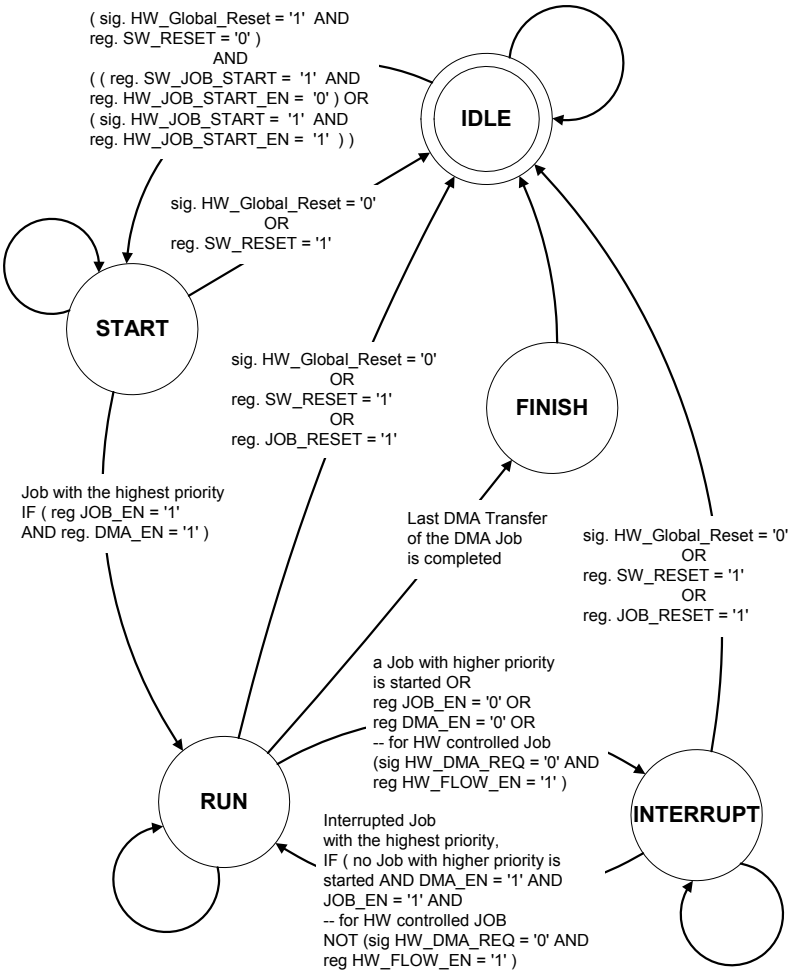


Figure 11: Finite state machine diagram of the job control algorithm

3.6.3.1.3.8 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 Control register is set to "1". Writing a "1" clears the error bit. If an error bit is set to "1", then the interrupt request (DMA_IRQ) is generated.
- Job counter register (GDMA_JOB_COUNT) counts clock cycles for a selected running job. Selection of the jobs for which the clocks shall be counted is done with 32-bit register Enable Job Counter for the Job (GDMA_JC_EN), one of the GDMA Control registers. Reset of the job counter is done by bit "Reset Job Counter" (JC_RESET) of the GDMA Main Control register (GDMA_MAIN_CTRL).

3.6.3.1.4 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.

3.6.3.1.4.1 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).

3.6.3.1.4.2 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

3.6.3.1.5 Third step

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

3.6.3.1.6 Result

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

3.6.3.1.6.1 DMA job started by SW or by HW

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

Start of the job by SW

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

Start of the job by HW

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

3.6.3.1.6.2 DMA job controlled by SW or by HW

Control of the job by SW

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

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

Control of the job by HW

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

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

3.6.3.1.6.3 Job finished

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

3.6.3.1.6.4 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.

3.6.3.1.7 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. From the GDMA controller is accessible through its AHB Master interface. Note that the datapath from the GDMA cntroller core to the GDMA RAM is: GDMA cntroller 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).
For the implementation of the GDMA for the ERTEC 200P the configuration with an internal GDMA RAM has been chosen. The GDMA RAM has size of ((n+1) x 16 Byte) + ((j+1) x 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 configuaration for ERTEC 200P is n = 255 and J = 31. **For this configuration the internal GDMA SRAM is organized in 1152 x 39 Bit, with 7 EDC bits**.
The GDMA RAM has additionaly EDC bits (7 Bit for a 32Bit word, 1Bit error correctable, 2Bit error recognizable). If an EDC error is detected, in the SCRB Register 'EDC_EVENT' (see Chapter 5.3.8) the appropriate reason is stored (GDMA-1B: 1Bit-Error corrected or GDMA-2B: 2Bit-Error recognized) and the Interrupt 'EDC_Event' IRQ48 is generated (see Chapter 5.4.1). The EDC Event Register is reseted by writing the register with '0h'.
After reset the initialisation of the EDC-Bits isn't done by hardware. This must be done by software. After this initialisation the SW must finally set 'GDMA_INIT_DONE = 1b'in the SCRB-Register 'EDC_INIT_DONE' (see Chapter 5.3.8).

3.6.3.1.8 Interrupts

The DMA controller has two interrupt request outputs. These signals are high active pulse. The length of the interrupt request pulse is at least 2 and at most 5 AHB clock cycles.
The GDMA controller incorporates no local interrupt controller of its own.
The two interrupt requests are:
DMA Interrupt Request (DMA_IRQ) - generated in the following cases:

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

The monitored types of error are:

- DMA Destination Address error - Assumed when a wrong destination address, pointing to the DMA registers or the DMA RAM, is programmed. The relevant error bit in the GDMA_ERR_IRQ_STATUS register is ERR_DST_ADDR.

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

3.6.3.1.9 HW_JOB_START Interface

Following internal HW signals are mapped to the GDMA to control DMA Jobs:

Port	Source	Active level	Description
HW_JOB_START (starts the job in HW by rising edge):			
HW_JOB_START_0	**PNPLL_OUT11 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_1	**PNPLL_OUT12 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_2	**PNPLL_OUT13 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_3	**PNPLL_OUT14 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_4	**PNPLL_OUT15 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_5	**PNPLL_OUT16 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_6	**PNPLL_OUT17 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_7	**PNPLL_OUT18 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_8	**PNPLL_OUT19 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_9	**PNPLL_OUT20 (PN-IP)**	high	coming from PNPLL in PN-IP
HW_JOB_START_10	**TIM_OUT0 (Timer Unit)**	high	coming from Timer 0 in the Timer Unit
HW_JOB_START_11	**TIM_OUT1 (Timer Unit)**	high	coming from Timer 1 in the Timer Unit
HW_JOB_START_12	**TIM_OUT2 (Timer Unit)**	high	coming from Timer2 in the Timer Unit
HW_JOB_START_13	**TIM_OUT3 (Timer Unit)**	high	coming from Timer 3 in the Timer Unit
HW_JOB_START_14	**TIM_OUT4 (Timer Unit)**	high	coming from Timer 4 in the Timer Unit
HW_JOB_START_15	**TIM_OUT5 (Timer Unit)**	high	coming from Timer 5 in the Timer Unit
HW_JOB_START_16	**reserved**		
HW_JOB_START_17	**reserved**		
HW_JOB_START_18	**reserved**		
HW_JOB_START_19	**reserved**		
HW_JOB_START_20. .63	**reserved**		
HW_DMA_REQ (controls DMA transfers to / from serial peripherals):			
HW_DMA_REQ_0	**UART1_TX-FIFO half-full_of_less**	high	UART1 Tx-FIFO not half full (GDMA to INCR8 Mode)
HW_DMA_REQ_1	**UART1_RX-FIFO not empty**	high	UART1 Rx-FIFO not empty (GDMA to Single Mode)
HW_DMA_REQ_2	**UART2_TX-FIFO half-full_of_less**	high	UART2 Tx-FIFO not half full (GDMA to INCR8 Mode)
HW_DMA_REQ_3	**UART2_RX-FIFO not empty**	high	UART2 Rx-FIFO not empty (GDMA to Single Mode)
HW_DMA_REQ_4	**UART3_TX-FIFO half-full_of_less**	high	UART3 Tx-FIFO not half full (GDMA to INCR8 Mode)
HW_DMA_REQ_5	**UART3_RX-FIFO not empty**	high	UART3 Rx-FIFO not empty (GDMA to Single Mode)
HW_DMA_REQ_6	**UART4_TX-FIFO half-full_of_less**	high	UART4 Tx-FIFO not half full (GDMA to INCR8 Mode)
HW_DMA_REQ_7	**UART4_RX-FIFO not empty**	high	UART4 Rx-FIFO not empty (GDMA to Single Mode)
HW_DMA_REQ_8	**SPI1_SSPRXDMA**	high	SPI1 Rx-FIFO not empty - DMA Request (GDMA to Single Mode)
HW_DMA_REQ_9	**SPI1_SSPTXINTR**	high	SPI1 Tx-FIFO not half full - DMA Request (**enable** SSPTXINTR Interrupt (Transmit FIFO is half full or less), GDMA to INCR4 Mode)

Port	Source	Active level	Description
HW_DMA_REQ_10	**SPI1_SSPTX_ Delayed_Request**	high	SPI1 transmit delayed – DMA. When Timer 4 with 'SPI1 Tx-FIFO not half full' expired, this bit is setted. GDMA to Single Byte Transfer. When **DMA_ACK is used**, the bit is resetted.
HW_DMA_REQ_11	**SPI2_SSPRXDMA**	high	SPI2 Rx-FIFO not empty - DMA Request (GDMA to Single Mode)
HW_DMA_REQ_12	**SPI2_SSPTXINTR**	high	SPI2 Tx-FIFO not half full - DMA Request (der SSPTXINTR Interrupt (Transmit FIFO is half full or less). GDMA to INCR4 Mode)
HW_DMA_REQ_13	**SPI2_SSPTX_ Delayed_Request**	high	SPI2 transmit delayed – DMA Request.When Timers 5 with 'SPI2 Tx-FIFO not half full' expired, this Bit is setted. GDMA to Single Byte Transfer. When **DMA_ACK is used**, this bit is reset-ted!
HW_DMA_REQ_14	**DMA_REQ_OUT**	high	SD_MMC Controller - DMA Request, **DMA_ACK is used**
……			
HW_DMA_REQ_31	**reserved**		

Table 7: HW control signals of GDMA

3.6.3.1.10 GDMA-IP Bugs

- ***JOB_Reset:***

The JOB_RESET of the last job (ERTEC 200P: JOB31) does not function reliably. The reason is that at JOB31 the JOB_RESET does not delete all the information of the last job. The information that the job was interrupted and that it has to be continued in the JOB STACK is not deleted either. A reconfiguration of the JOB therefore does not have any effect.
At ERTEC 200P JOB31 should not be used as a hardware JOB, otherwise the following error description with workaround applies. If necessary, JOB31 can be used as a software JOB if the software does not interrupt this JOB with the JOB_Reset.

Error description: JOB31 is to be aborted and restarted

1. JOB31 is disabled by
   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 where it was interrupted, meaning that deleting of JOB31 in the GDMA does not function

Workaround:

1. The JOB is disabled (JOB_EN = 0).
2. Wait until JOB no longer copies. Then check whether JOB was completed.
3. If the JOB was not completed, the first transfer of the new partial transfer list is copied from the transfer into the interruption memory (JOB stack) of the JOB.
4. The JOB is enabled (JOB_EN = 1) but no longer started.

- ***JOB Stop with GDMA_MAIN_CTRL.DMA_EN = 0 faulty:***

Boundary conditions:
- 1 job (GDMA_JOB0_CTRL)
- 1 transfer record that is executed by this job
- Triggering through software through GDMA_JOB0_CTRL.SW_JOB_START = 1

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

Workaround:
- The difference in GDMA_ACTUAL_STATUS is not a problem because the content of this register is only valid when Bit 5 = 1. At Bit 5 = 0 the content of the register may not be evaluated!
- The fact that the job counter at GDMA_MAIN_CTRL.DMA_EN = 0 continues to count is not correct. However it is only a flaw because the job counter has the purpose of measuring the runtime of a job. It would therefore be wrong to disable the job because this would falsify the measurement. It is therefore not a sensible use case.

3.6.4 External Memory Controller (EMC)

To allow masters on the AHB bus system to access external memory devices, a functionality is needed, which bridge from the internal AHB bus to external memory devices. This functionality is provided with the External Memory Controller (EMC) module:
The EMC interface contains 2 separate memory controllers – SDRAM controller (SDRAMC) and an asynchronous controller (ASYNCC) – for different devices like

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

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

- Supports AHB 2.0 lite protocol (No SPLIT, no RETRY)
- Burst transactions are accepted at the AHB input side.
  - 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 of this rule: read access to Burst Flash ROM allows 16 beat burst, read access to Page Mode ROM allows 16 beat burst).
- The EMC itself can be configured using the AHB slave interface

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

- SDRAM-Controller features:
  - 16/32 Bit databus width

- PC133 SDRAM-compatible (125 MHz synchron 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)
  - maximum 14 bits row-address MA(15:2)

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

- Asynchronous Controller features:
  - 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
  - A maximum of 64 MB 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 schematic of the EMC module is

shown in **Figure 12**.

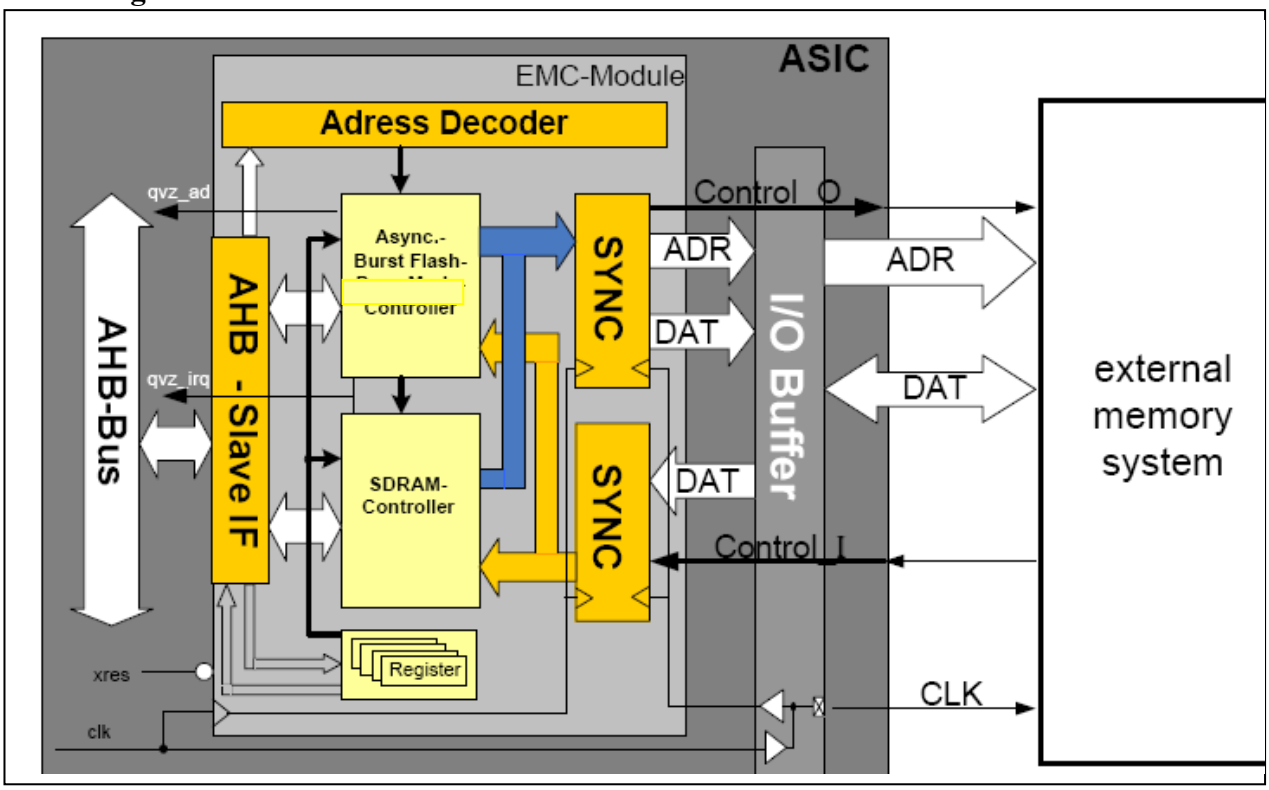


Figure 12: Block schematic EMC module

The EMC interface can only be driven by 1,8V (necessary for mobile SDRAM). The EMC IO pads are separated from the other IOs and have their own power supply (VDD1,8V and GND). This supply ring contains all 77 IO Pads of the EMC interface. So the EMC can work with 1,8V and the other IOs (e.g. GPIOs) with 3,3V.
The driver strength of the 1,8V EMC pads is 12mA after PowerOn Reset by default.

In the maximum memory configuration at the EMC interface is shown (2x SDRAM devices, 2x Burst Flash devices, 1x Peripheral / Level Shifter for 3,3V). Each memory device has its own clock. The ERTEC 200P has 3 Clocks for the SDRAM memory (CLK_O_SDRAM2/1/0) and 3 Clocks for the Burst Flash memory (CLK_O_BF2/1/0).
The CLK_O_SDRAM0 / CLK_O_BF0 are used for feedback the external memory clock to the ERTEC 200P inputs CLK_I_SDRAM / CLK_I_BF. CLK_O_SDRAM0 must always be returned back to CLK_I_SDRAM, even if only an external SRAM or an EMC / XHIF coupling to a second ETEC200P is used. This is necessary for storing the incoming read data.
CLK_O_SDRAM1 / CLK_O_BF1 are used for the respective memory device no.1 and CLK_O_SDRAM2 / CLK_O_BF2 for the respective memory device no.2.
All Clocks can be switched off, if they don't be used. In the DRIVE_EMC register (see Chapter 5.3.8) are the appropriate bits for this function. CLK_O_SDRAM0/1, CLK_O_SDRAM2, CLK_O_BF0/1 and CLK_O_BF2 could be switched separately.

After PowerOn Reset only CLK_O_SDRAM0/1 are switched on.

3.6.4.1 EMC-IP Notes

Maximum number of wait cycles:

The configured value for the maximum number of wait cycles should be configured at least 5 times bigger than the expected worst case delay time of the external wait signal. False QVZ interrupts are possibble 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.

Shift mode at an 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 maximum address range of 64 Mbytes can be addresses per ChipSelect. In the process the bit EXTENDED_CONFIG. ASYNC_ADDR_MODE in the EMC interface is set by the primary bootloader.

With this configuration it is possible to address:

- 16 Mbytes at a data bus width of 8 bits (byte)
- 32 Mbytes at a data bus width of 16 bits (half-word)
- 64 Mbytes at a data bus width of 32 bits (word)

3.6.5 Host Interface - parallel (XHIF)

A local bus unit (2x XHIF-IP) is implemented for accesses by a host system connected in parallel. The local bus unit (XHIF), structured internally with two XHIF-IPs, supports accesses by an external 16-bit and 32-bit host CPU. The ERTEC 200P can be accessed in its entire address space by an external host system through the Host Interface (through a max. of 8 settable address windows (pages) of each 256 bytes – 1 / 2 Mbytes).

The basic configurations (data width, ready polarity, read/write line) for the XHIF (together for both XHIF_0/XHIF_1-IPs) are set with the PowerOn reset via the HW_CONFIG Pins 6..3. These configurations of ARM926EJ-S can still be changed subsequently in the XHIF_CONTROL register via the APB interface. The external host should not access this register because this can result in undefined states at the Host Interface. Meaning that the basic configurations can no longer be changed subsequently by the external host.
Figure 13 shows the block diagram of the Host Interface. The right-hand section shows the structure of the XHIF with two individual XHIF modules (XHIF_0/1) (2 times 4 pages with 2 Mbytes each). Pin XHIF_SEG_2 is used to select the respective XHIF module (XHIF_0 or XHIF_1). To do so, the XHIF interface is switched to the selected XHIF module. Access to the AHB interface is effected through arbitration. Pins XHIF_SEG_0 and XHIF_SEG_1 address the respective pages of an XHIF module. The XHIF configuration (XHIF_ACC_Mode, XHIF_POL_RDY, XHIF_CPU_Width) is effected identically for both XHIF modules and is wired directly at both modules (source: Config Pins or XHIF_Control register). The XHIF interrupt (XHIFerr_IRQ) is ORed by both modules and laid outwards (to the Event unit in the PER-IF).

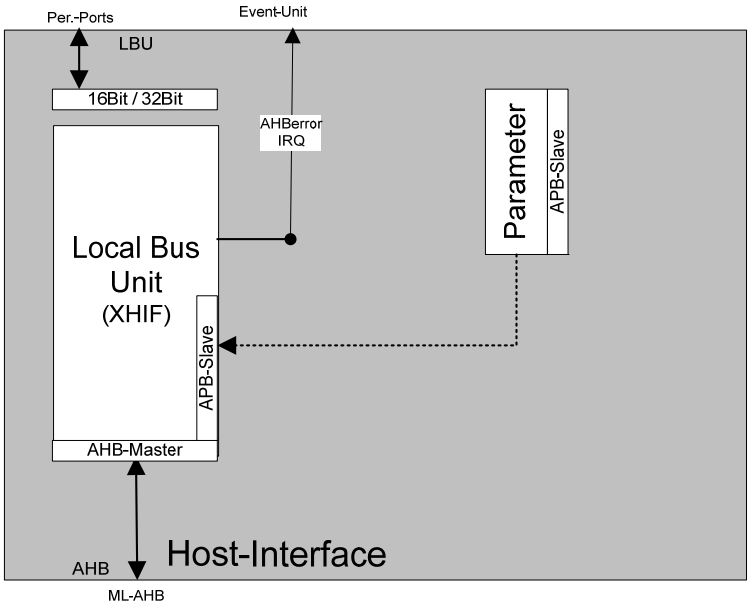


Figure 13: Host Interface block diagram

Two functions are implemented through the input pin 'XHIF_XCS_R_A20' of the ERTEC 200P. The signal at this input can thus be used as a ChipSelect for configuring the pages through the external host or as address line A20 (→ increase of the page address space to 2 Mbytes). How this input pin is used at the XHIF modules is set in the SCRB register 'XHIF_Mode' with the bit 'XHIF_Mode':

- XHIF_Mode = 0b: Page register Chipselect XHIF_XCS_R → Page = max. 1 Mbyte
- XHIF_Mode = 1b: Address line XHIF_A20 → Page = max. 2 Mbyte

Two Use Cases thus result with different maximum Page sizes.

8 pages with 1 Mbyte address space each:
This setting is available by default (XHIF_Mode = '0'). The external host is wired with its address lines as follows to the XHIF interface:

XHIF Interface (ERTEC 200P)	External host (external circuiting)
XHIF_A1	A1 at 16-bit interface fixed to '0' at 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 as a rule assigned as follows: 2x PN-IP, 1x PER-IF, Reserved, 2x EMC-SDRAM, 1x GMDA, 1x dyn. (on ARM926 D-TCM or EMC-SRAM or APB peripherals or EMC register).

8 pages with 2 Mbytes address space each:
(XHIF_Mode = '1') has to be set for this setting. The external host is wired with its address lines as follows to the XHIF interface:

XHIF Interface (ERTEC 200P)	External host (external circuiting)
XHIF_A1	A1 at 16-bit interface fixed to '0' at 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

XHIF parameter assignment is carried out as follows:

- After the ERTEC 200P has been reset, the XHIF Interface is available with 8 pages with 1 Mbyte each and parameter assignment of the Page registers via the external host (XHIF_XCS_R).
- The parameter assignment of the Page registers is effected from the point of view of the host with A20 = '0' (XHIF_XCS_R = '0').
- From the point of view of the host with A20 = '1' (XHIF_XCS_R = '1') and XHIF_XCS_M = '0' that Page is accessed through which access to the APB range is possible. In the process the register XHIF_Mode = '1" is set in the SCRB block.
- From now A20 has the function as the address line (XHIF_A20) from the point of view of the host so that 2 Mbytes address range are available per Page.

The 8 address windows (2 Mbytes each) are as a rule assigned as follows: 1x PN-IP, 1x PER-IF, Reserved, 1x EMC-SDRAM, 1x GMDA, 1x APB Peripherals, 2x dyn. (on ARM926 D-TCM or EMC-SRAM or EMC Register).

The XHIF interface can be operated with both 3.3 V and 1.8 V. The corresponding IO pads are supplied separately with VDD_XHIF (circuited with 1.8 V or 3.3 V, no other setting required) and GND. This supply ring encompasses all 64 IO pads of the Host Interface.

The drive strength of the 1.8 V XHIF pads in the ERTEC 200P is set to 6 mA after PowerOn reset. A external host processor should increase this to 9 mA for the XHIF pads (parameter assignment in the registers DRIVE47_32GPIO, DRIVE63_48GPIO, DRIVE79_64GPIO and DRIVE95_80GPIO, see Chapter 5.3.8) before it carries out the first read accesses to the XHIF interface. 6 mA are sufficient for the pin XHIF_XRDY because this pin is wired directly to the corresponding input pin of the host and only has to transload this one load.

3.6.5.1 XHIF application information

- The basic configurations (data width, ready polarity, read/write line) for the XHIF are set with the PowerOn reset via the HW_CONFIG Pins 6..3. This configuration of ARM926EJ-S can be changed subsequently in the XHIF_CONTROL register via the APB interface.
- The XHIF_CONTROL register should not be changed by the external host. According to the XHIF-IP specification the external host may not address its configuration register through the APB interface (AHB2APB bridge). Here the HW_CONFIG6..0 setting is **only** to be carried out through the PowerOn reset.
- Through the register Chipselect XHIF_XCS_R the external host can only address the XHIF-IP internal Page registers (Offset, Range, Buffermode) and the XHIF_VERSION. In the process only the lowest-value 6 address bits are considered for the register selection (addresses ≥ 0x40h are invalid).
  At a configured 16-bit data width solely half-word write / read accesses may be used and only word write /read accesses at 32-bit data width. Byte-granular accesses are not allowed!
- At an active memory Chipselect XHIF_XCS_M the write / read accesses are passed through directly to the HOSTIF AHB Master Interface - all aligned access types byte, half-word, word are allowed.
  If the 32-bit data width is configured, the XHIF_ADR(1) pin has to be terminated fixed to '0' at the ERTEC 200P (XHIF_ADR(0) is laid internally statically to '0') so that word address are always active XHIF-IP-internally.
  At a configured 16-bit data width the XHIF_ADR(1) pin is required for half-word addressing and is co-driven by the external host.
- Initialization sequence of the XHIF-IP after reset according to the IP specification:
  - Removal of PowerOn reset
  - Basic configuration through HW-CONFIG 6..3 pins, afterwards only the ARM926EJ-S can change the configuration (the external host may not access yet)
  - Page setting from ARM926EJ-S: Access through the APB on the corresponding register or Page setting from host: Through the XHIF_XCS_R Chipselect access to the corresponding register
  - Memory accesses (HOSTIF-AHB Master) through external host
  - As soon as the external host has started with the memory accesses, the ARM926EJ-S may no longer change the basic configuration (data width, ready polarity, command mode)

- Ready signaling to the external XHIF host is effected by the XHIF_XRDY signal. XHIF_XRDY is implement with a tristate driver whose Output_Enable does not become active before the access beginning (as soon as CS and RD or WR are active) and inactive again at the access end (one cycle after an active XHIF_XRDY).
  At a module design you have to ensure that XHIF_XRDY is a controlled **push-pull** output that requires an external pull resistance according to its polarity (i.e. if XHIF_XRDY is low active a pulldown has to be foreseen and vice versa). The pull resistance ensures that the XHIF_XRDY to the external host is active long enough.
  During parameter assignment of the externally connected host you have to take into account that XHIF_XRDY briefly appears as active for the host at the access beginning due to the pull resistance (it takes up to 11 ns until the pin of ERTEC 200P is driven). To ensure that the external host does not already recognize the access as acknowledged, it has to recognize the XHIF_XRDY with delay. When an ERTEC 200P functions as external host, this has to be achieved by configuring a corresponding number of "read/write strobe cycles" (see Chapter 5.3.6, parameter R__Strobe or W_Strobe in the EMC register ASYNC_BANK0-3).
- *How can the host recognize that the ERTEC 200P is ready for host accesses after the reset phase?*
  The host can recognize through the Page 0 Range and Offset registers that the ERTEC 200P is ready for host accesses. The Page 0 Range and Offset registers have the value 0x0000_0000 by default. After configuration by the ERTEC 200P the registers of the Page 0 must have the following values:
  - Page 0 Range register: 0x0010_0000
  - Page 0 Offset register: 0x0800_0000

  This means that the host has to poll this register in the CS_R area after an ERTEC 200P reset until the values initialized by the ERTEC 200P are read. The host can thus recognize that the ERTEC 200P is ready for host accesses.
- *Problem with host accesses?*
  Problems can only occur during the active reset phase of the ERTEC 200P, triggered for example by a:
  - Software reset
  - Watchdog reset

  If the host accesses the CS_M area during the reset phase, access by the ERTEC 200P is not completed with READY! Accesses by the host to CS_R are completed during the reset phase, meaning that there is no problem here!
  The following points should be observed to recognize / avoid the problem:
  - The host must recognize when the ERTEC 200P was reset via Reset (use a GPIO that signals the reset state at the host) and prevent accesses to CS_M

  - At a missing READY the XHIF master has to generate an enforcing READY (i.e. abort access) and initiate error signaling (Interrupt).

3.6.6 PN-IP

3.6.6.1 PN-IP Interfaces

3.6.6.1.1 AHB Interface

The PN-IP is connected with the multi-layer AHB through an AHB Master and an AHB Slave interface.

Through the AHB Master interface the PN-IP can as the active bus master transfer AHB data through the multi-layer. Simultaneously accesses by other AHB Masters to the PN-IP resources can be carried out through the AHB Slave interface.
The AHB interfaces of the PN-IP do not support Split and Retry functionalities. Otherwise all the transfer types, transfer sizes and burst operations are supported in accordance with the AMBA specification.

3.6.6.1.2 Interrupt Management

All the interrupt events that are generated from the mechanisms of the PN-IP can be made available to the connected CPUs through the Interrupt management. Here 2 CPU systems (CPU subsystems) are supported. For the ERTEC 200P these are:

- ARM926 subsystem
- Event Unit in the PER-IF (external host CPU)

Since the distribution of PN Stack and the application software to the CPU systems is not fixed, the assignment of the individual interrupt events to the respective interrupt inputs or interrupt controllers of the CPUs is universal. The set of all the interrupt events occurring in the PN-IP are made available specifically for the ARM926 and host subsystem. This is effected by:

- A CPU-specific Interrupt Controller (PN-ICU) that generates two combined interrupts of all the interrupt events for the ARM926 (PN-ICU 2) / external host (PN-ICU 3) → PN_IRQ2/3(**1:0**)
- A CPU-specific multiplexer structure (PN-MUX) that provides 14 selectable PN interrupt events each directly as individual interrupts for the ARM926 (PN-ICU 2) / external host (PN-ICU 3) → PN_IRQ2/3(**15:2**).

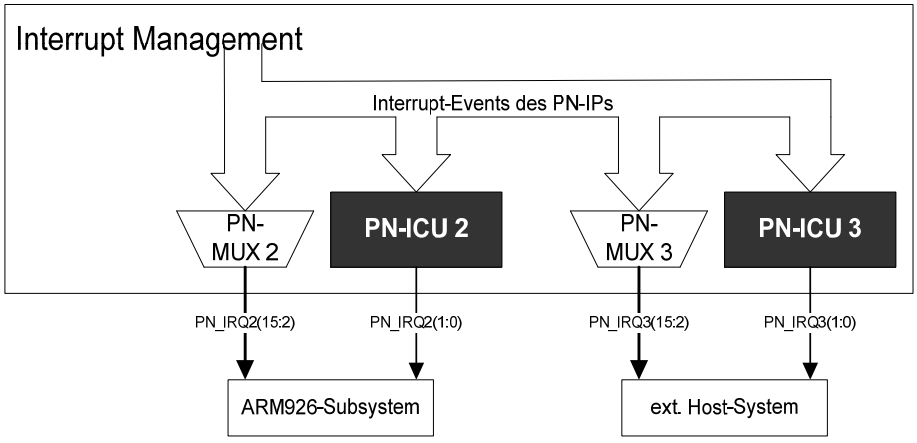


Figure 14: Basic representation of the Interrupt Management in the PN-IP

The combined interrupts PN_IRQ2(**1:0**) (IRQ56/57) and 14 selectable individual interrupts PN_IRQ2(**15:2**) (IRQ58-71) are switched by the PN-IP for the ARM926EJ-S Interrupt Controller (ARM-ICU) (Chapter 5.4.1).
The interrupt generation for the host is carried out through the Event Unit in the Peripheral Interface. The PN-IP passes the corresponding interrupts (combined interrupts: PN_IRQ3(**1:0**), individual interrupts: PN_IRQ3(**15:2**)) to the Event Unit. There the generation of a group interrupt signal to the external host (XHIF_XIRQ) is carried out.

PN-ICU:

The generation of the combined interrupt PN_IRQx(**0/1:0**) is carried out in accordance with the following scheme:

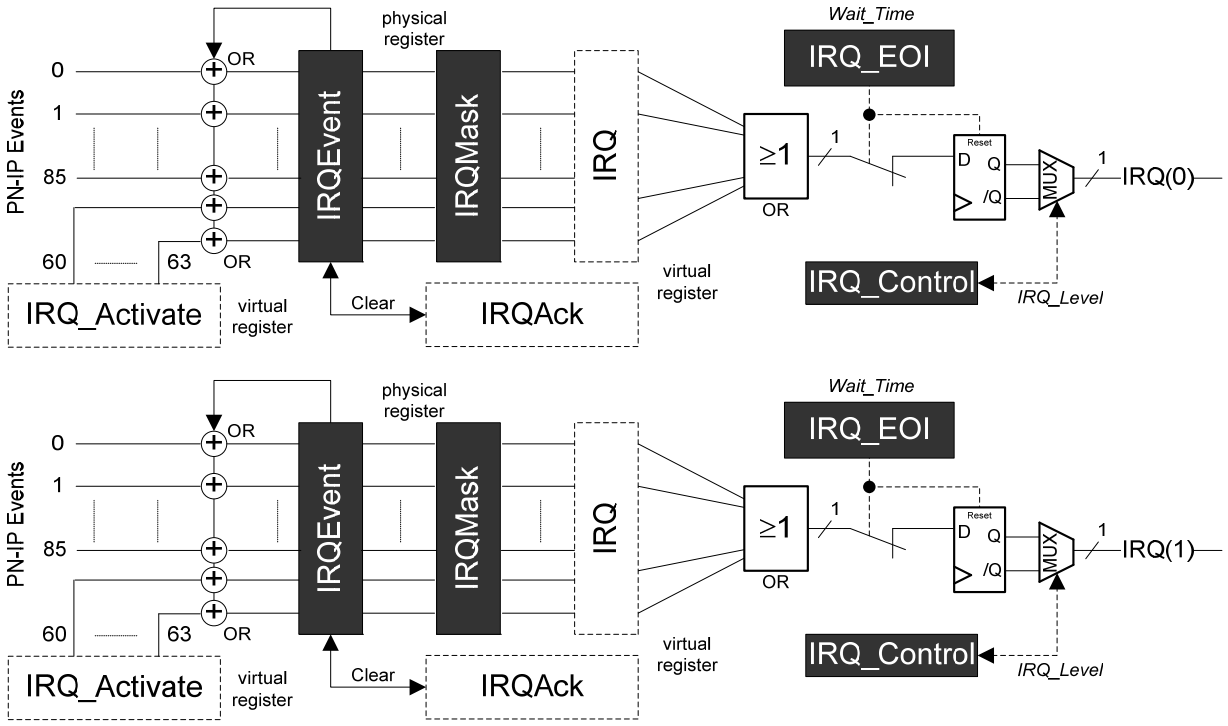


Figure 15: Block diagram of the PN-ICU for the combined interrupts PN_IRQx(0/1:0)

One Interrupt Controller each exists for each combined interrupt (2x for ARM926 / external host: PN_IRQ2/3(1:0). The IRQEvent registers are implemented to communicate and store the PN-IP internal events. The same events are interconnected on all the Interrupt Controllers of the combined interrupts. One or more set event bits can respectively trigger a combined interrupt to the external CPU subsystem → PN_IRQx(0/1:0). This is effected for all event bits through:

- Internal PN-IP events (PN-IP events) OR
- Dedicated AHB write access to IRQEvent OR
- Dedicated AHB write accesses (software events through IRQ_Activate).

Events bits already written / set are not reset during further write accesses to the IRQEvent register, they remain. The assignment and meaning of the individual event bits is identical for all the PN-ICUs (1..3).

The decision whether an interrupt event (set event bit) triggers the corresponding combined interrupt PN_IRQx(0/1:0), is configured through masking of the event bits using the IRQMask registers. A set mask bit blocks the corresponding entry so that no PN_IRQx(0/1:0) is triggered. Those event bits that trigger a combined interrupt PN_IRQx(0/1:0) can thus be read through the IRQ register (IRQ bits). Writing to this register is ignored. The two combined interrupts will receive different maskings through the software in order to form interrupt groups (for example acyclic API and cyclic API separated by groups).

Resetting of the bits in the IRQEvent registers is carried out through a write access to the RQAck registers. The PN-ICU is to be operated solely in the "**Acknowledge**" mode, meaning that the set event bits are determined through a read access on the IRQ register. A subsequent write access to the IRQAck registers results in resetting of the written register bits in the IRQEvent registers and thus to resetting of the IRQ bits in the IRQ registers.

Deactivation of the PN_IRQx(0/1:0) combined interrupts is effected through the IRQ_EOI register. The combined interrupts PN_IRQx(0/1:0) are reset with a write access to the registers. Renewed

activation of the combined interrupt PN_IRQx(0/1:0) through set event bits in the IRQEvent registers is possible at the earliest after the specified waiting time (Wait_Time) has expired.
To ensure flexible adaption of external hardware events to the Interrupt Management, these can be parameterized with regard to the active level through the IRQExt_Event register using *Ext_IRQx_Level*. *Ext_IRQx_Edge* can furthermore be used to select positive or negative edge triggering

PN-MUX:
The generation of the individual interrupt PN_IRQx(**2/1:15**) is carried out identically in accordance with the following scheme:

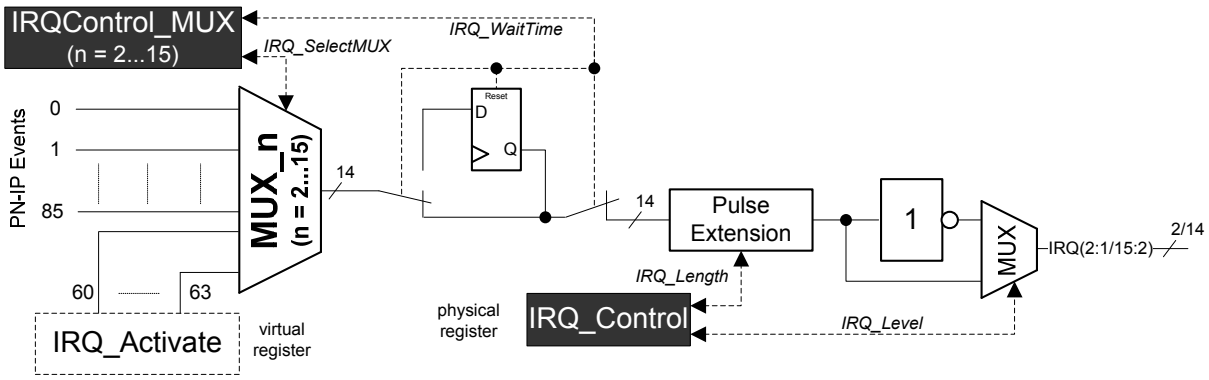


Figure 16: Block diagram of the PN-MUX for the individual interrupts PN_IRQx(2:1/15:2)

The Bits(3:0) of the IRQ_Activate registers are assigned to Event Bits 63 - 60 in order to trigger a software event through combined interrupts PN_IRQx(0/1:0) and individual interrupts PN_IRQx(2:1/15:2). In the process a software event in the form of a pulse (8-ns pulse) is triggered through a write access to the corresponding register bits. This write access is executed non-storing in the IRQ_Activate registers. A read access always supplies the value 0x0000.
Which PN-IP event or software event is transferred outwards as an individual interrupt can be set for each interrupt output through the IRQControl_MUXn registers using the *IRQ_SelectMUX*. The assignment of the PN-IP events to the respective event number (*IRQ_SelectMUX*) corresponds to the allocation (bit position) of the PN-IP events in the IRQEvent register.
The interrupt load of the selected individual events in the connected CPU Subsystems can be reduced by writing *IRQxControl_MUXn.IRQ_WaitTime*. As long as *IRQ_WaitTime* is running (*IRQ_WaitTime* ≠ 0x000), no new individual event (individual interrupt) is entered in the respective CPU subsystem. In this time the selected events are stored temporarily in *IRQxControl_MUXn.IRQ_Event* and initiate immediate triggering of an individual interrupt after *IRQ_WaitTime* has expired (*IRQ_WaitTime* = 0x000). At the same time the value stored in *IRQ_Event* is deleted. The individual event is extended to the configured pulse length (*IRQx_Control.IRQ_Length*) by means of Pulse Extension. The current timer value can be read back with a read access. If *IRQxControl_MUXn* is accessed again by means of a further write access before *IRQ_WaitTime* has expired, the newly written value is used as the *IRQ_WaitTime*. This allows the *IRQ_WaitTime* to be extended or reduced correspondingly by timely writing of the software.
As long as *IRQ_WaitTime* is not written and thus not used (meaning that *IRQ_WaitTime* = 0x000, default value), the selected individual event (individual interrupt) is transferred directly (combinatorily) to the respective CPU subsystems.

3.6.7 Peripheral Interface

The cyclic IO data are stored consistency-ensured in the Peripheral Interface.
The Periphery Interface is connected to the multi-layer AHB through 2 AHB slave interfaces. One slave interface is assigned fixed to the PN-IP (communication) and the other to the application (ARM, Host or GDMA). The Peripheral Interface is configured through the APB.
The AHB interfaces of the PER-IF do not support Split and Retry functionalities. Otherwise all the transfer types, transfer sizes and burst operations are supported in accordance with the AMBA specification.
One interrupt each of the Peripheral Interface is laid to the ARM (IRQ54) (Chapter 5.4.1) and to the external host (XHIF_XIRQ). One separate Event Unit each is available to this purpose.

The IO RAM in the PER-IF has an EDC Code (1Bit Error correctable, 2Bit Error recognizable). If an error occurs while reading to the IO RAM, the error cause (PER-IF-1B: 1Bit Error corrected or PER-IF-2B: 2Bit Error recognized) is stored in the SCRB register 'EDC Event' (see Chapter 5.3.8) and the interrupt 'EDC_Event' IRQ48 is triggered (see Chapter 5.4.1). To delete the EDC Event Register has to be overwritten with '0h'. After the reset an initialization of the EDC bits is carried out in the IO RAM. Completion of this initialization is signaled in the SCRB Register 'EDC_INIT_DONE' (see Chapter 5.3.8).

Note:
The PerIF register Burst_Config(7:0) = 0x03 has to be configured to ensure correct operation of the PN-IP module with the PerIF module, see also clause 3.3.5

3.6.8 Multiport-Ethernet-PHY

The following functions are supported by PHY:

- 100Base-TX PHY
- MII
- 100Base-FX
- Auto MDI-X
- Next Page support
- Optimized Tx and Rx latency
- Jitter free Latency
- Fast line break detection

3.7 Peripherals at the I/O bus (APB)

The function blocks connected to the APB have interfaces with different widths. The following table shows the supported access mechanisms and the data width of these blocks.

Access modes				Waitstates at the AHB		
Bit 31:24	Bit 23:16	Bit 15:8	Bit 7:0	Read	Write	Function block
8 bits	8 bits	8 bits	8 bits	2	0	IO filter, Boot_ROM
16 bits		16 bits				
32 bits						
8 bits	8 bits	8 bits	8 bits	Ready	0	PER_IF (except PER_IF-GPIO)
16 bits		16 bits				
32 bits						
32 bits				2	0	GPIO, PER_IF-GPIO, Timer0-5, F-counter Watchdog, SCRB, SPI1/2, Flash Controller, UART1-4, 1x I^2C Host Interface

Table 8 : Data width of the peripherals

Invalid types of access (byte /half-word writing to the timer) are not întercepted by hardware! There is also no signaling in the form of an Error Response or a time-out (QVZ) interrupt at the AHB.
Accesses to memory areas that are not coded out (identified as "not used" in memory mapping Chapter 5.1) trigger the IRQ52 (Chapter 5.4.1) and are completed by a Ready generated by the APB address decoder. Write accesses do not have any influence on the system.. Read accesses supply undefined data.

3.7.1 General-Purpose-I/Os

There are 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 alternatively to the XHIF)

GPIO31-0 is available as the default block and is multiplexed with the interface signals of UART2/3, SPI1/2, I^2C_1_3, Timer 0 – 5 and ARM926 Watchdog (see Chapter 4.3). After a reset this GPIO block is selected and all are switched to inputs. The alternate functions are set by the software via the GPIO register (see Chapter 5.3.14).
The GPIO15-0 inputs are passed additionally via an I-filter (see Chapter 3.7.2) and laid to the ARM Interrupt Controller (IRQ32-47) (Chapter 5.4.1). The polarity of these signals to the ARM Interrupt Controller can be set in the register GPIO_POLSEL (see Chapter 5.3.8). Depending on the polarity of the external signal a high-active interrupt level is ensured.

The GPIO3-0 inputs are passed via an I-filter (see Chapter 3.7.2) and can control hardware jobs at the GDMA (see Chapter 3.6.3.1.9). The polarity of these signals to the GDMA can be set in the register GPIO_POLSEL (see Chapter 5.3.8). A high-active level always has to be set for the GDMA. Since these input signals are not synchronized at the GDMA, the mode "Synchronize" (Mode 2) has to be parameterized selectively in the I-filter for these inputs!
GPIO31-0 have software-selectable integrated pulls (see Chapter 4.5) that have a default configuration after a reset (see corresponding Pull31_0GPIO register in the SCRB, Chapter 5.3.8). At GPIO11-0 the pulls are disabled by default because no fitting setting can be found for the application due to the alternate functions.

GPIO95-32 is completely not available when the parallel 32-bit Host Interface (XHIF) is used. If XHIF is operated in 16-bit mode, 18 GPIOs remain.

GPIO95-32 have software-selectable integrated pulls (see Chapter 4.5) that have a default configuration after a reset (see corresponding Pull95_32GPIO register in the SCRB, Chapter 5.3.8). GPIO55-54 (XHIF_IRQ, XHIF_XRDY) do not have internal pulls!

3 GPIO modules with 32 GPIOs each are set for the GPIOs. These GPIO modules include the following properties:

- The number of General Purpose Input/Outputs (GPIOs) can be configured per hardware.
- Each GPIO is programmable as an input / output (GPIO function).
- Each GPIO input (pin) can be read by the software.
- Each GPIO output can be set or reset by the software (bit-selective programming is possible).
- The GPIO function can be multiplexed with up to 3 alternative functions (Function A-C).
- 2 Reset signals have an influence on the GPIO register (XRESET_HW as an asynchronous reset, XRESET_GPIO_SM as a selective reset.)
- Most of the GPIO pins have integrated pullups / pulldowns to prevent floating.
- The GPIO pins can be laid via an integrated filter (see Chapter 3.7.2) (necessary for local IOs, external interrupts, alarms, etc.).
- Up to 16 GPIOs (GPIO15-0) can trigger interrupts to the ARM processor and are connected to the ARM Interrupt Controller to this purpose (IRQ32-47) (Chapter 5.4.1). The polarity of these signals to the ARM-ICU (ARM Interrupt Controller)can be set in the register GPIO_POLSEL (see Chapter 5.3.8). Depending on the polarity of the external signal a high-active interrupt level is ensured. The signals have to be at least 2 clocks (125 MHz) long.
- Up to 6 GPIOs can be used as gate / trigger signals for the internal timer.
- Up to 4 GPIOs (GPIO3-0) can control hardware jobs at the GDMA (see Chapter 3.6.3.1.9). The polarity of these signals to the GDMA can be set in the register GPIO_POLSEL (see Chapter 5.3.8). A high-active level always has to be set for the GDMA.

Figure 17 shows the block diagram of a GPIO module:

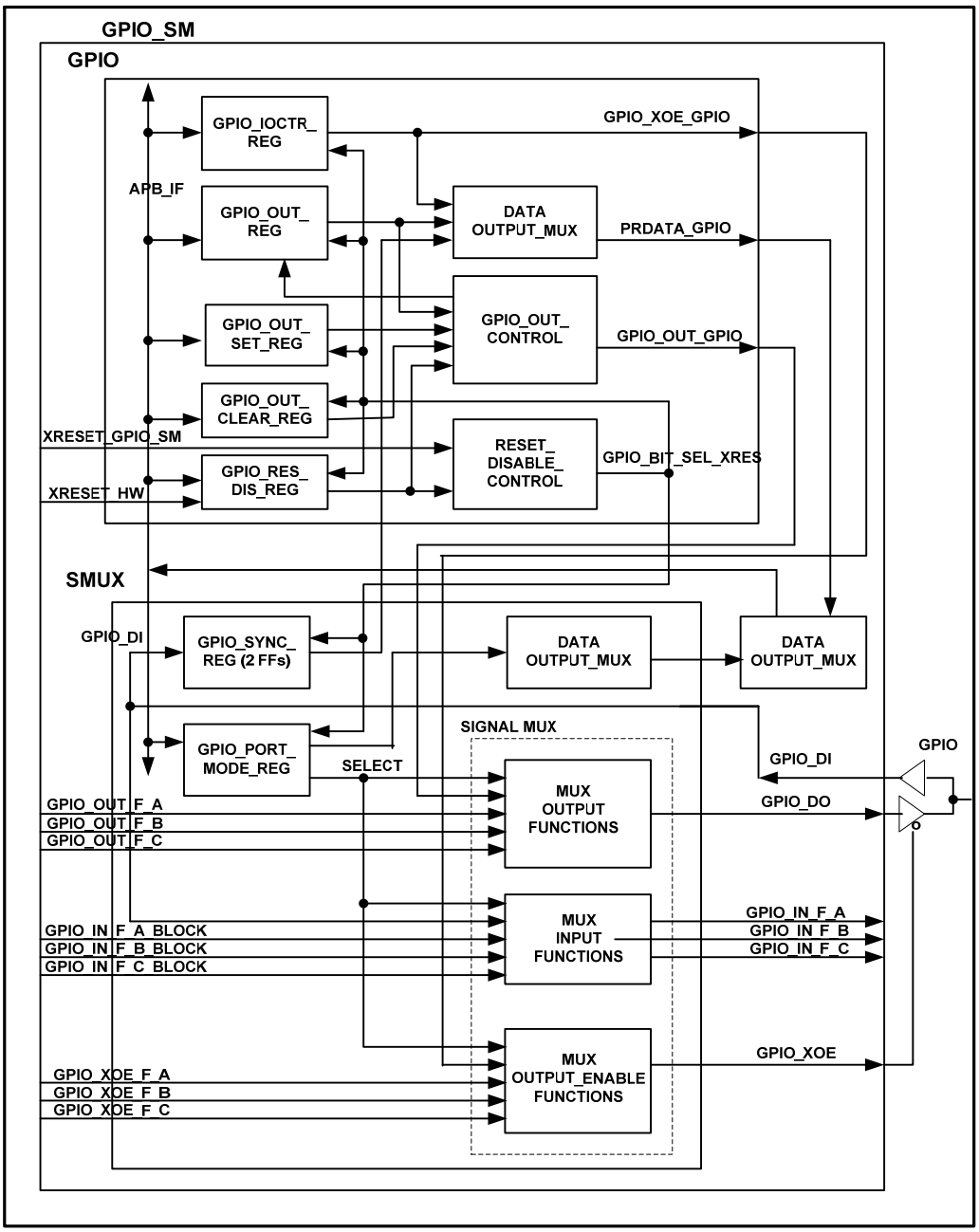


Figure 17: Block diagram GPIO module

In the ERTEC 200P the feature of an GPIO-internal "reset disable" is activated (module input: XRESET_HW circuited with reset signal. This means that in addition to the global reset (XRESET_GPIO_SM) selective resetting in the GPIO_SM is possible.

In addition to the alternate functions (A-C) integrated in the GPIO modules, the GPIO95-32 block has further alternate functions that are set via the configuration pins CONFIG(6-3) (see Chapter 4.2) or the SCRB register 'CONFIG_REG' (see Chapter 5.3.8, Host Interface). In addition to the alternate functions (A-C) integrated in the GPIO modules, the GPIO31-0 block has a further alternate function that is set via the configuration pins CONFIG(6-3) (see Chapter 4.2) or the SCRB register 'CONFIG_REG'.

3.7.2 I-Filter

Up to 80 input signals of the GPIO interface (GPIO15-0, GPIO95-32) can be passed via an I-filter. The allocation of the GPIOs to the respective I-filter input is listed in Chapter 4.3. The active input signals D_IN can be stored in the subsequent modules either unfiltered directly (Mode 1: no synchronization), unfiltered and synchronized (Mode 2) or filtered (Mode 3). In all cases the input signals are passed through the I-filter that operates channel-specifically and can be parameterized via the APB.

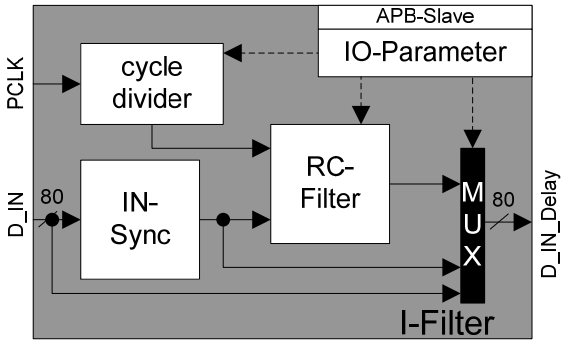


Figure 18: Block diagram of the IO Filter

The APB module "I-filter" offers the possibility of filtering short signal peaks and interference coupling. A synchronization unit (IN_Sync) and filter stage (RC_Filter) within the module are used to this purpose. These are available for every input (channel). If no filtering of the respective input signal is desired, either the synchronization stage of the I-filter (Mode 2) operates or the respective input signal is passed directly through the I-filter without influencing (Mode 1). The filter stage is supplied by a central cycle divider whose parameterizable cycle sources can also be selected channel-specifically. The exact operating principle of the I-filter is is illustrated by the following block diagram that uses a channel as the example:

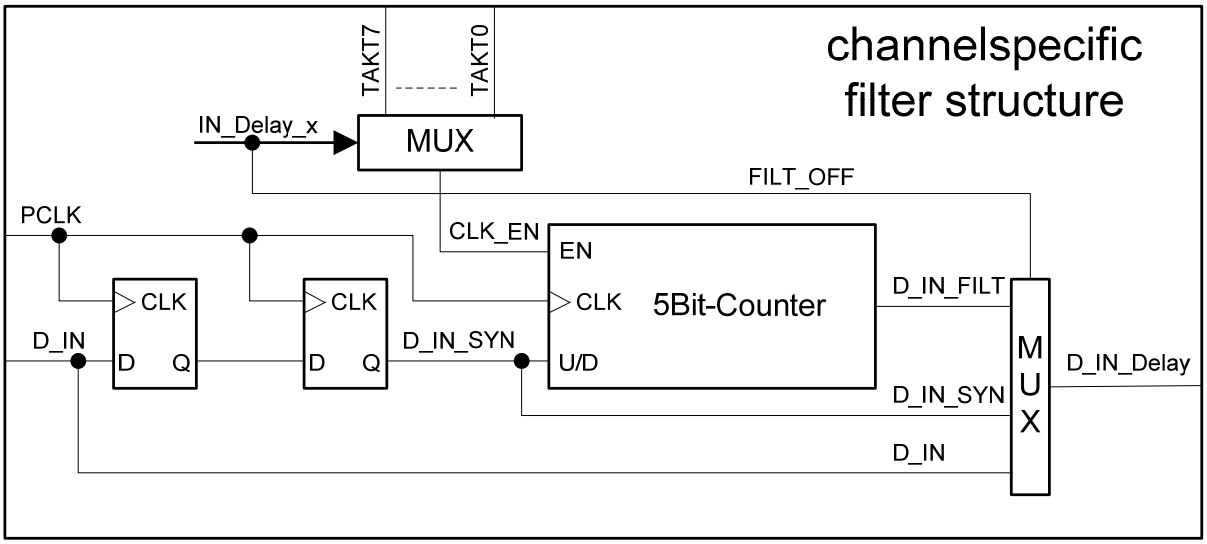


Figure 19: Block diagram of the filter structure of a channel

Filtering of the input signal is carried out after a two-stage synchronization (required to avoid metastable states). The channel-specific filter is implemented as a 5-bit up-down counter that counts up at a "High" state of the input signal and down at a "Low" state. The counter has the value '00h' (load value) after a RESET.

Since the input channels are operated separately from each other at the digital input modules, a separate counter has to be implemented for each input signal. It must also be possible to set different filter times channel-specifically with these counters. Selection of the filter time is effected in the 'FILT_Delay_x' registers that, depending on the setting, channel-specifically select different cycle sources (Mode 3) or only the synchronization of the input signal. without RC-filtering (Mode 2). If Mode 2 is the case, the two-fold synchronized input signal D_IN_SYN is passed directly to the output signal D_IN_Delay and not directed via the counter register. If Mode 1 is the case, the input signal D_IN is directed directly to the output signal D_IN_Delay. The synchronization level and the counter register are bypassed. A dynamic changeover of FILT_Delay_x during operation is not allowed. Parameterization is only carried out in the initialization phase.
The cycle sources (TAKT0 to TAKT7) themselves are generated from a central cycle divider for all the channels. The basic cycle for all the divider stages is the APB cycle PCLK (125 MHz). The operating principle is shown in the following block diagram:

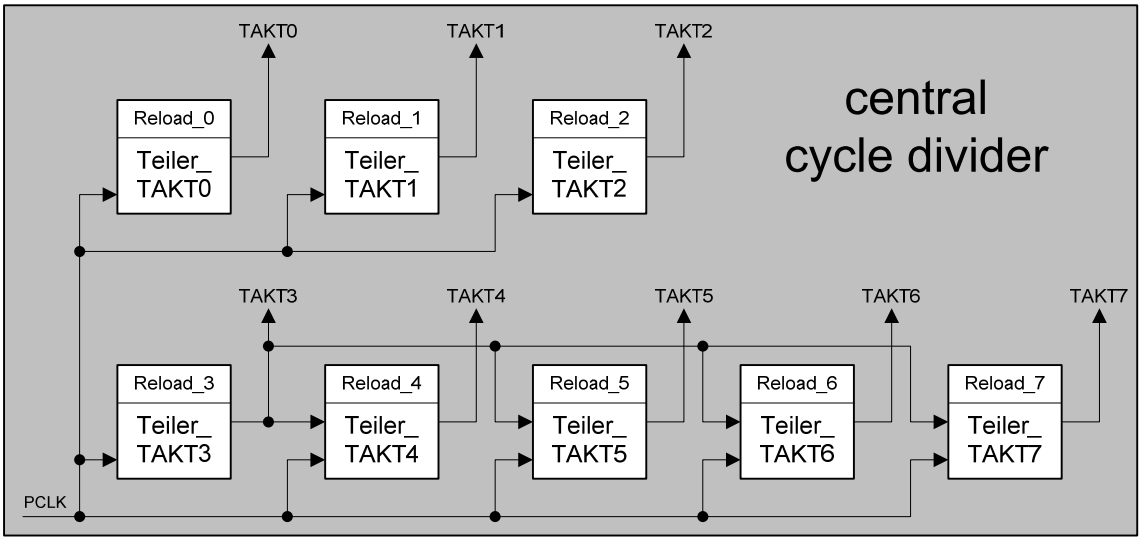


Figure 20: Block diagram of the central cycle divider

The central cycle divider consists of several divider stages that are in part also cascaded. The division factors are set for each divider stage via a separate register 'FILT_Reload_x'. A dynamic changeover of FILT_Reload_x during operation is not allowed.
To determine the "total division factor" the division factors of the individual stages have to be multiplied in the case of cascaded dividers. The following tables show the filter times that are reached with example division factors at 125 MHz:

Parameter register ***FILT_Reload_x***	**Example division factor**	**Filter time (without 2-stage sync.)** t_{delay}	**Minimum filter time (filter time - CYCLE jitter)**
Reload_0	8	1.024 µs (TAKT0)	0.960 µs
Reload_1	40	5.12 µs (TAKT1)	4.80 µs
Reload_2	78	9.984 µs (TAKT2)	9.360 µs
Reload_3	390	42.42 µs (TAKT3)	39.30 µs
Reload_4	2	99.84 µs (TAKT4)	93.60 µs
Reload_5	6	299.5 µs (TAKT5)	280.78 µs
Reload_6	30	1.5 ms (TAKT6)	1.406 ms

Reload_7	40	2.0 ms (TAKT7)	1.875 ms

Table 9 : Example filter times art the I-Filter

The respective filter time is calculated by multiplying the corresponding division factor (from divisor 3 on cascaded) with 16xTPCLK, because the output value can only change at every 16th cycle due to the filter principle. The filter times lie in the range of:

- Min. **128 ns** E.g. at: *IN_Delay_x*=0 & *FILT_Reload_0*=0
- Max. **134 ms** E.g. at: *IN_Delay_x*=7 & *FILT_Reload_3*=0x3FF & *FILT_Reload_7*=0x3FF

The maximum (total) throughput time for the filter module at the given parameterization consists of the filter time plus 16 ns for two-fold synchronization. The minimum (total) throughput time (see Table 9) at this parameterization results from the jitter of the cycle signals TAKT0 to TAKT7.

3.7.2.1 Principle of the RC filter

The highest value bit of the 5-bit counter is the filtered output signal (*D_IN_Delay*).
When the input signal (*D_IN_SYN*) changes to the state "High", the counter begins to count upward. If the counter state changes from '15' to '16', the counter is set automatically to the counter value '31' ('0x1Fh'). When the counter value '31' is reached the output signal (*D_IN_Delay*) is also set to "High". The counter value '31' remains until the input signal (*D_IN_SYN*) changes from "High" to "Low".
When the input signal (*D_IN_SYN*) changes to the state "Low", the counter begins to count downward. If the counter states changes from '16' down to '15', the counter is set automatically to the counter value '0' ('0x00h'). When the counter value '0' is reached the output signal (*D_IN_Delay*) is also set to "Low". The counter value '0' remains until the input signal (*D_IN_SYN*) changes from "Low" to "High".
In the case of short signal changes during the counting phase the counter is counted up- or downward in accordance with the signal state, meaning that the counter no longer begins at '0' at a signal change (this would be a signal delay and not filtering).
The filter structure thus forms an RC element or an I-controller in digital form.

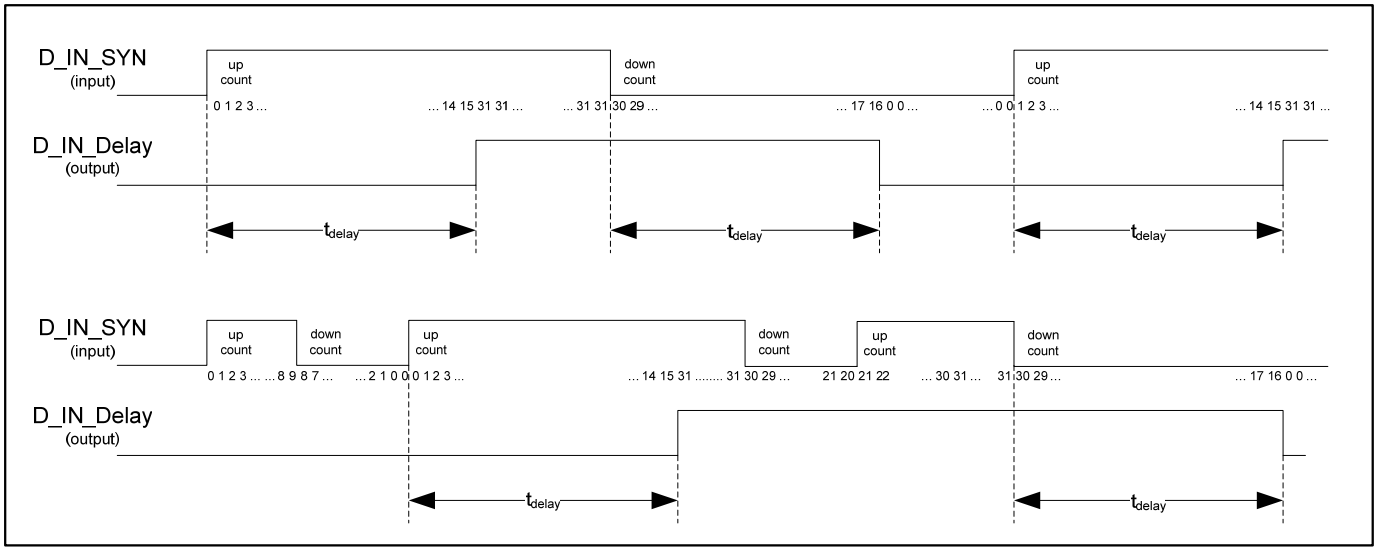


Figure 21: Signal course at signal filtering

3.7.3 Timer 0 – 5

3.7.3.1 Overview

In ERTEC 200P there are six independent timers integrated which serve for monitoring various software routines. Each of these software timers is assigned an own interrupt.
Access to the timer registers is carried out at word limits (32 bits).

Summary of the timer functions

- Bit down-counter
- Programmable bit prescaler connectable (separately for each timer)
- Loadable/reloadable
- Start, stop and continue function
- Interrupt when counter value '0' has been reached
- Read register for reading out the counter value
- Three event registers for event-driven storing of the counter value:
  - Two external HW events (selectable from the EXTERNAL_INPUTS signals of the TIMER_TOP module) for storing in two HW event registers
  - One internal SW event for storing in one SW event register
- Timers are cascadable (prerequisite: timer output of the low-order counter(s) is fed back to the TIMER_TOP module via the EXTERNAL_INPUTS signals and is selected as GATE_TRIG signal at the high-order timer)
- Clocking of the counter can be done with the input clock CLK_TIMT = APB_Takt and via the SW (internal gate/trigger signal) as well as via an input signal (external gate/trigger signal)
- Triggering of the counter (loading) can be done via the SW (internal gate/trigger signal) as well as via an input signal (external gate/trigger signal)
- Clock output for supplying further modules (bit divider for CLK_TIMT = APB_Clock).

In the zero crossing, an output pulse (TIMER_OUT) is generated which can be evaluated as interrupt. The TIMER_TOP module contains six TIMER and multiplexer submodules each for selecting the trigger signals for the individual timers, and functions expanding over all timers (e.g. synchronous releasing of all timers, address coding). The TIMER submodule contains the basic timer function (e.g. counter, prescaler, load register for counter). The distribution of the timer functions to the TIMER and TIMER_TOP modules is supposed to simplify the reusability in different ASICs.

The TIMER and TIMER_TOP modules are described in detail in the following sections.
Connections of the TIMER_TOP module to other modules or to ASIC pins are to be found in the functional description of the chip/core level.

3.7.3.2 TIMER_TOP functionality

The TIMER_TOP submodule contains the wiring of the individual TIMER modules, the Gate_Trig_Control registers, the multiplexer and timer MUX registers for the input signals INT_GATE_TRIG_TIM, EXT_GATE_TRIG_TIM, EVENT1, EVENT2, and the clock divider.
The TIMER_TOP module consists of the following functional units:

- Gate_Trig_Control register for triggering SW gate/trigger signals and for releasing the count pulses for the timers
- SW event trigger register for storing the current counter values in the Int_Event registers
- Six timers
- Multiplexers and the corresponding registers for selecting the sources for the inputs of the TIMER submodules
- Clock divider and the corresponding register for activating the clock divider.

The TIMER_TOP module has an APB interface (AMBA 2.0). The access width has to be 32 bits. Different accesses lead to a faulty writing of the register of the TIMER_TOP module.

Gate Trig Control Register:
The TIMER_TOP module has a Gate_Trig_Control register in which a SW gate signal can be set/reset for each TIMER module (= INT_GATE_TRIG_TIM inputs of the TIMERs). The effect depends on the selected mode.
Moreover, the count clock can be released/blocked for each TIMER module (= CLK_EN inputs of the TIMERs). By writing on this register, all TIMER modules can be started / stopped synchronously.
Attention: If the SW gate signal is changed while simultaneously the count clock operating mode is released/blocked, the following behavior applies to all operating modes in which the SW gate signal is relevant:

a) When blocking the counting pulse:
   SW gate signal change has no effect anymore.

b) When releasing the counting pulse:
   SW gate signal change is effective in the next count/load clock, depending on the selected operating mode. This also applies to operating modes in which the edge of the SW gate signal is relevant.

Software event trigger register:
Writing a ’1’ into the bit “n” of the SW event trigger register triggers a positive edge at the INT_EV input of the TIMER module “n” and leads to a storing of the counter value of the TIMER module “n” in the Int_Event register of the TIMER module “n”.
The ’1’ is not stored (read=0).
By writing on this register, the current counter values of all timers can be synchronously stored in the SW event registers of the TIMER modules.

Multiplexer/TIM_MUX register:
The TIMER_TOP submodule has own multiplexers and registers for their control for each timer, which serve for selecting the sources for the inputs of the TIMER modules EXT_GATE_TRIG_TIM, EVENT1, EVENT2. The EXTERNAL_INPUTS (15:0) inputs of the TIMER_TOP module are available as sources.
The registers for the multiplexers are designed in such a way that the numbers of the select signals entered there select the corresponding number of the input signal.

Each TIMER module can be assigned own sources for the inputs EXT_GATE_TRIG, EVENT1, EVENT2.
Thus, versatile measuring tasks can be configured (for examples, see Section “Sequences”).
For the assignment of the input signals of the individual multiplexers, see “Chip/core level”, Sub-section “Timer”.

Clock divider / clock divider register:
The TIMER_TOP module has a clock divider for providing a clock for external modules, which is symmetrical with the input clock CLK_TIMT. The clock divider is 8 bits wide (NUM_OF_CLKDIV_BITS). The current value of the clock divider is not 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.
- If the current value of the clock divider is ‘0’, the clock divider is loaded with the value of the bits CLOCK_DIVIDER_VALUE.

When clock dividing is enabled, the clock divider value CLOCK_DIVIDER_VALUE must not be changed. Before changing the clock divider value CLOCK_DIVIDER_VALUE the clock dividing must be disabled by setting Clock Divider Enable CLK_DIV_EN to 0.

Attention: Writing a new clock divider value CLOCK_DIVIDER_VALUE may have the effect that CLK_OUT runs with a wrong clock period once.

The clock divider is implemented as a down-counter which counts down cyclically from the clock divider value CLKDIV_VALUE.
The clock divider divides the frequency of input clock CLK_TIMT by the factor 1 / (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.

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.

3.7.3.3 Timer modul

Timer block diagram:

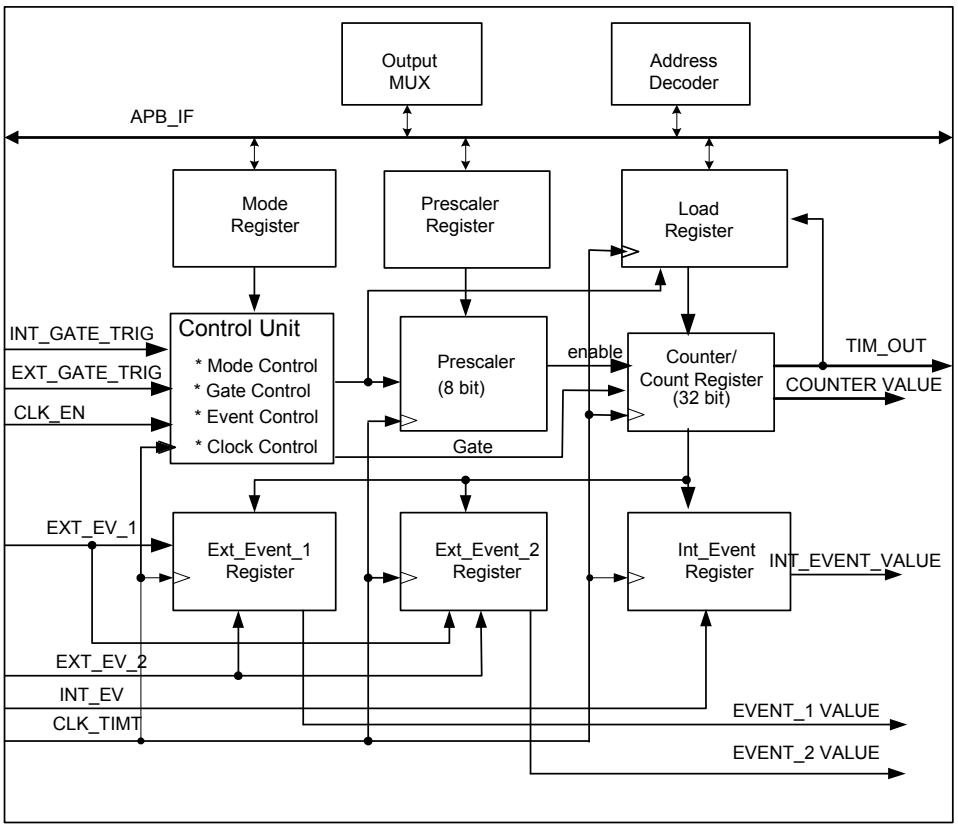


Figure 22: TIMER block diagram

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 synchronously with that clock. The reset input of the submodules TIMER_TOP and TIMER is used as asynchronous reset; a synchronization of the reset input has not been implemented.
Each TIMER submodule consists of the following functional units:

- Counter/count register (counter and register for reading the current counter value)
- Prescaler/prescaler register (prescaler and register for prescaler load value)
- Load register (load value of the counter)
- Mode register (setting the operating mode)
- Event register for storing the counter values dependent on events (register Ext_Event_1, Ext_Event_2, Int_Event)

Counter / counter register:

Each TIMER module contains a 32-bit counter which counts down from a start value. The counter stops when the value ‘0’ has been reached or is automatically reloaded if the reload function has been activated in the mode register.
The following sources can be selected as count/load clock of the counter:

- Input clock CLK_TIMT (125 MHz = 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). All inputs are expected to be synchronous to the clock CLK_TIMT.

The software can read the current value of the counter in the count register.

The counter range is up to 26 days (Input Clk = 125 MHz, 8bit Clock Divider, 8bit Prescaler, 32bit Timer).

Prescaler / prescaler register:
In accordance with the presetting, the counter is operated with CLK_TIMT (125 MHz = APB clock). Nevertheless, each counter can be connected with an 8-bit prescaler (independent parameterization for each timer) so that the runtime of the individual TIMER modules can be increased accordingly.
Each TIMER module contains an 8-bit prescaler which counts down cyclically from a start value, and a prescale register which contains the start value for the prescaler (= PRESCALER_VAL).
The prescaler counts / loads with the internal clock CLK_TIMT.
The prescaler is deactivated if

- timer input CLK_EN=0
- Gate_effect=0 and external gate/trigger signal has passive level

The current counter value of the prescaler is nonreadable. There is no signal for the prescaler indicating the counter value ’0’. The prescaler register is 8 bits wide (NUM_OF_PRESCALER_BITS). The prescaler divides the counter frequency by the factor 1 / (PRESCALER_VALUE+1).

Possible values for the prescaler register: ’0’ to ’2^8 - 1’

Load Register:
The load register contains the load value (start value) of the counter. Writing on the load register with a new value causes the takeover of the load value into the counter if the following conditions are met:

- Timer input CLK_EN = ’1’ during writing
- Bit DIS_RLD_WHEN_WR_LDREG in the mode register = ’0’ during writing

The time of takeover of the load value into the counter also depends on the operating mode of the timer and is carried out with the count/load clock of the counter.

Mode Register:
The operating mode of the TIMER module is set via the following bits in the mode register:

- Init_bit
- Clk_input select
- Reload_disable
- DIS_RLD_WHEN_WR_LDREG
- Ext_gate_trig_enable
- Gate_polarity
- Gate_effect
- Timer_out_polarity
- Event1/2_control
- Event1,2_Inversion

The **Init_bit** is an initialization bit which resets the TIMER module, loads (restarts) the counter with the value from the load register and loads the prescaler with the value from the prescaler register. It deletes all values in the event registers and resets the edge evaluation of the event inputs and the EXT_GATE_TRIG signal. After having been set by the software, this bit is automatically deleted again, a reading of the bit by the software always results in the value ’0’. The definition is carried out in accordance with the following table:

Init_bit	Function
0	Bit Init_bit not active (no initialization)
1	Bit Init_bit active (initialization)

The counter **counts/loads** with the count/load clock. The selection is done by the mode register bit Clk_input_select and the value of the prescaler register in accordance with the following table:

Clk_input_select	Function
0	Count/load clock = rising edge of CLK_TIMT; The counting/loading of the counter with the CLK_TIMT edge only takes place if the prescaler value is ’0’
1	Count/load clock = external gate/trigger signal (edge evaluation with CLK_TIMT is valued as count/load enable signal)

The **Reload_disable bit** determines if the counter stops the counting process when it has reached the value ’0’ (single mode) or if the counter is newly started with the reload value from the load register when the value ’0’ has been reachedand the next count/load clock occurs (reload mode). The definition is carried out in accordance with the following table:

Reload_disable	Function
0	Reload mode active
1	Single mode active

Note that in reload mode the cycle length(= time until counter reach zero) includes one additional count/load clock for reloading the counter.

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 registe).
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 ’0’ If these conditions are met, the takeover is carried out with the next counter edge or the next gate/trigger signal (depending on the setting in the mode register).

By setting the DIS_RLD_WHEN_WR_LDREG bit to 1, the takeover of the load register value is only possible during the zero crossing. This function is required to use the timers as phase-shifted cyclic timers.

Note: When writing load register with DIS_RLD_WHEN_WR_LDREG = 0 the resulting cycle length (= time until counter reach zero) can vary between new load value and new load value + old load value for one cycle.

One of the two input signals **EXT_GATE_TRIG** or **INT_GATE_TRIG** can be used:

- as gate signal for releasing/blocking the counter, or
- as trigger signal for triggering the counter, or
- as clock signal for clocking the counter.

The selection of the external gate/trigger signal is done by the mode register bit Ext_gate_trig_enable in accordance with the following table:

Ext_gate_trig_enable	Function
0	INT_GATE_TRIG = internal gate/trigger signal
1	EXT_GATE_TRIG = external gate/trigger signal

Differentiation between INT_GATE_TRIG and EXT_GATE_TRIG (internal and external gate/trigger signal):

- INT_GATE_TRIG is triggered by the software
- EXT_GATE_TRIG is triggered by an external hardware signal.

The **Gate_polarity bit** defines the active level or the active edge of the external gate/trigger signal in accordance with the following table:

Gate_polarity	Function
0	Level of the external gate/trigger signal is “high active” or rising edge is active edge
1	Level of the external gate/trigger signal is “low active” or falling edge is active edge

Together with the Clk_input_select bit, the **Gate_effect bit** defines the effect of the external gate/trigger signal in accordance with the following table:
Clk_input_select=0 (counter counts with CLK_TIMT as count/load clock dependemt on the preselector value): Gate/Trigger mode switching:

Gate_effect	Function
0	Gate mode: External gate/trigger signal has the effect of a gate (=gate mode) for the count/load clock and has to be active for counting
1	Trigger mode: Each active signal edge of the external gate/trigger signal causes the triggering of the counter with the reload value if the current counter value is unequal to the reload value or has no effect if the current counter value is equal to the reload value (=Trigger mode)

Note on the trigger mode:
If the active signal edge of the external gate/trigger signal occurs in the Trigger mode during CLK_EN=1 and then CLK_EN becomes 0 without the preselector value having gone to 0, the previous active signal edge only becomes effective when CLK_EN is 1 again and the preselector value is 0, i.e. the requirement does not get lost because of CLK_EN=0.
Clk_input_select=1 (counter counts with external gate/trigger signal as count/load clock (edge evaluation)): Count/Toggle mode switching:

Gate_effect	Function
0	Count mode:

	External gate/trigger signal is the count/load clock (Count mode)
1	Toggle mode: External gate/trigger signal is the count/load clock and causes the switching between • loading of the counter with the reload value (if the current counter value is un-equal to the reload value) • down-counting (if the current counter value is equal to the reload value)

Thus, delays as well as time monitorings can be implemented.
Note: The loading of the counter can simultaneously mean the starting of the counter, namely if Reload_disable=1 and Counter_value=0 and load value #0.
Note: The bit combination Gate_effect=1 and Clk_input_select=1 can be used to generate a symmetrical output signal out of an unsymmetrical input signal (reload value=1).

The TIM_OUT output of the TIMER module is active when the counter has the value ’0’ (non-saving), otherwise passive. The active level of the TIM_OUT output is defined by the **Timer_out_polarity bit** in accordance with the following table:

Timer_out_polarity	Function
0	TIM_OUT = high active
1	TIM_OUT = low active

The storing of the current counter values in the event registers **Ext_Event_1** and **Ext_Event_2** is done dependent on the bits Event1_control, Event2_control, Event1_Inversion, Event2_inversion in the mode register and dependent on the inputs EXT_EV_1, EXT_EV_2 in accordance with the following tables:

Bit Event1_control

Event1_control	Function
00	EXT_EV_1 input does not affect register Ext_Event_1 and register Ext_Event_2
01	Rising edge of the EXT_EV_1 input leads to the storing of the counter value in the register Ext_Event_1 *)
10	Falling edge of the EXT_EV_1 input leads to the storing of the counter value in the register Ext_Event_2 *)
11	Rising edge of the EXT_EV_1 input leads to the storing of the counter value in the register Ext_Event_1, falling edge of the EXT_EV_1 input leads to the storing of the counter value in the register Ext_Event_2 *)

*) The above table is valid if the Event1_Inversion bit is set to 0; if the bit is set to 1, the “rising edge” and “falling edge” are to be exchanged.

Bit Event2_control

Event2_control	Function
00	EXT_EV_2 input does not affect register Ext_Event_1 and register Ext_Event_2
01	Rising edge of the EXT_EV_2 input leads to the storing of the counter value in the register Ext_Event_1
10	Falling edge of the EXT_EV_2 input leads to the storing of the counter value in the register 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 forbidden. If these combinations are set nevertheless, the counter value is not stored.
The registers Ext_Event_1 and Ext_Event_2 are readable and writable by the software; the setting of the registers after reset is ’0000h’.
If the condition for storing the current counter values in the event registers Ext_Event_1 and Ext_Event_2 occurs simultaneously with a writing of one of these registers, the writing is treated preferentially.
The storing of the current counter values in the event register Int_Event is triggered by a positive edge of the INT_EV input. The register is readable by the software and writable; the setting of the registers after reset is ’0000h’.
By storing the current counter values with the edges of the inputs EXT_EV_1, EXT_EV_2 into the event registers Ext_Event_1 and Ext_Event_2 it is possible to measure time intervals and periods between the inputs EXT_EV_1, EXT_EV_2. By storing the current counter value with the rising edge of the INT_EV input into the event register Int_Event it is possible to measure time intervals and periods between write accesses to the SW event trigger register.

The **loading of the counter** can be triggered in four different ways:

- Loading of the counter by writing the Init_bit in the mode register
- Loading of the counter by writing the load register
- Automatic loading of the counter after having reached the value ’0’ in the reload mode
- Loading of the counter by an external load signal (EXT_GATE_TRIG signal)

The counter counts/loads only if the following count/load conditions are met:

- Input CLK_EN=1
- Count/load clock of the TIMER module active.

Exception: The loading of the counter with the value from the load register always takes place when writing the Init_bit in the mode register.
Each TIMER activity which is carried out dependent on the edge of an input signal (INT_GATE_TRIG or EXT_GATE_TRIG, EXT_EV1, 2, INT_EV) becomes effective with the third CLK_TIMT signal at the latest.
The active edges of the input signals at which the edge evaluation is carried out have to have a minimum distance, also see “Timing requirements”.

Event register:
Each TIMER module has:

- Two HW event registers (Ext_Event_1 and Ext_Event_2) in which the current counter values are stored with an edge of the inputs Ext_Ev_1, Ext_Ev_2 (edge programmable), and
- One SW event register (Int_Event) for storing the current counter values with a positive edge of the input INT_EV.

3.7.3.4 Overview of the counter count modes

Gate_ polar-ity	Ext_ gate_ trig_ en-able	Gate_ effect	Clk_ in-put_ select	Function
0	0	0	0	Gate mode: Counter counts/loads with internal clock CLK_TIMT dependent on the preselector value as long as INT_GATE_TRIG=1.
0	0	0	1	Count mode: Counter counts/loads with each rising edge of INT_GATE_TRIG, independently of the 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: Counter counts/loads with internal clock CLK_TIMT dependent on the preselector 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 preselector 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)

3.7.3.5 Timing requirements

The following applies to the TIMER submodule:

The minimum low pulse width and the minimum high pulse width of the signals INT_GATE_TRIG, EXT_GATE_TRIG, EXT_EV1, EXT_EV2, INT_EV must be at least one CLK_TIMT clock.

Furthermore it is required, that the active edges of the input signals at which the edge evaluation is carried out must have a minimum distance according to the following table:

Signal	**Minimum distance**
INT_GATE_TRIG, EXT_GATE_TRIG	At least three CLK_TIMT clocks; The following applies additionally: If one of the signals INT_GATE_TRIG or EXT_GATE_TRIG is used as a gate or trigger, it has to take at least three CLK_TIMT clocks plus the number of CLK_TIMT clocks set in the preselector in order to become effective.
EXT_EV1, EXT_EV2, INT_EV	At least two CLK_TIMT clocks

3.7.3.6 Operating rules

Initialization sequence, modifications of the mode register
Initialization sequence:

1. Stop counter (CLK_EN=0 = setting after reset)
2. Write load register (counter is not loaded if CLK_EN=0)
3. Write mode register with Init_bit = ’1’:
   counter is loaded with load register
4. Start counter (CLK_EN=1).

The initialization sequence also has to be carried out with each modification of the mode register to prevent any side effects when changing the operating mode.

Cascading of TIMER modules:
The TIMER modules can be cascaded with each other, provided that the timer output of the lower-value counter(s) is fed back to the TIMER_TOP module via the EXTERNAL_INPUTS signals and is selected as GATE_TRIG signal at the high-order timer).
Proceed as follows for that:

1. Program the multiplexer in such a way that each high-order counter has the timer output of the low-order counter as GATE_TRIG signal.
2. Define the interrupt evaluation in such a way that the interrupt of the high-order counter is evaluated. The interrupt of the low-order counter(s) must not be evaluated.
3. Set the operating modes (mode register, prescaler register) of the cascaded TIMER modules the same way (not absolutely necessary but reasonable if the cascaded TIMER modules are to count with the same clock).
4. When reading the counter value, make sure that the data is consistent, e.g. write SW event trigger register bits for all cascaded TIMER modules in order to take over the values of these TIMER modules synchronously into the Int_Event registers; then, read the Int_Event registers of these TIMER modules.
5. The setting of the preselectors for the cascaded TIMER modules has to be adapted to the application (same preselectors are reasonable).

Note on the Toggle mode
For the first count/load clock after having set the Toggle mode or after reset or after active Init_bit, it cannot be predicted if the count/load clock causes a loading of the counter with the reload value or a down-counting (depends on the level of the count/load clock during the setting or during the active Init_bits).
If this information is important for the software, it has to poll the level and value of the counter during the first count/load clock.

Be careful when storing into event registers!
The following combination of the bits Event2_control, Event1_control is forbidden:

Event2_control	Event1_control
01	01
01	11
10	10

10	11
11	01
11	10
11	11

3.7.3.7 Connections on Toplevel

Timer gate/trigger/event inputs

The timer module has an own multiplexer for each timer for the selection of the sources for the external timer gate/trigger inputs and the external timer event inputs. The assignment of the sources to the timer multiplexers is stated in the table below.
ATTENTION: Cascading of the timers is only possible when the outputs of the lower-level timers are connected to the EXT_GATE_TRIG mux of the respective higher-level timer(s)!
ATTENTION: It is expected that all timer inputs/outputs are synchronous with the timer clock. Asynchronous inputs (e.g. of ASIC pins) have to be synchronized accordingly. The synchronization of the ASIC pins is done on the core level via two flip-flops each with 125 MHz.

The final assignment of the TIM_TRIG is done after the specification of the pinning.

Timer	Ext_Gate_Trig_MUX	Event_1_MUX	Event_2_MUX
Timer 0-5	Bit0: TIM_TRIG0 (GPIO6/26) Bit1: TIM_TRIG1 (GPIO7/27) Bit2: TIM_TRIG2 (GPIO8) Bit3: TIM_TRIG3 (GPIO9) Bit4: TIM_TRIG4 (GPIO10) Bit5: TIM_TRIG5 (GPIO11) Bit6: TIM_OUT0 (Timer 0) Bit7: TIM_OUT2 (Timer 2) Bit8: TIM_OUT4 (Timer 4) Bit9: CLK_OUT (clock divider output timer top) Bit10: '0' Bit11: '0' Bit12: '0' Bit13: '0' Bit14: '0' Bit15: '0'	Bit0: TIM_TRIG0 (GPIO6/26) Bit1: TIM_TRIG1 (GPIO7/27) Bit2: TIM_TRIG2 (GPIO8) Bit3: TIM_TRIG3 (GPIO9) Bit4: TIM_TRIG4 (GPIO10) Bit5: TIM_TRIG5 (GPIO11) Bit6: TIM_OUT0 (Timer 0) Bit7: TIM_OUT2 (Timer 2) Bit8: TIM_OUT4 (Timer 4) Bit9: CLK_OUT (clock divider output timer top) Bit10: '0' Bit11: '0' Bit12: '0' Bit13: '0' Bit14: '0' Bit15: '0'	Bit0: TIM_TRIG0 (GPIO6/26) Bit1: TIM_TRIG1 (GPIO7/27) Bit2: TIM_TRIG2 (GPIO8) Bit3: TIM_TRIG3 (GPIO9) Bit4: TIM_TRIG4 (GPIO10) Bit5: TIM_TRIG5 (GPIO11) Bit6: TIM_OUT0 (Timer 0) Bit7: TIM_OUT2 (Timer 2) Bit8: TIM_OUT4 (Timer 4) Bit9: CLK_OUT (clock divider output timer top) Bit10: '0' Bit11: '0' Bit12: '0' Bit13: '0' Bit14: '0' Bit15: '0'

Table 10 : Table Assignment of the timer gate/trigger/event inputs to the timer multiplexers

With the external inputs TIM_TRIG0-5 it is possible to control the Timers from external HW. With the internal inputs TIM_OUT0, TIM_OUT2, TIM_OUT4, it is possible to link timers to each other. The output of the clock divider CLK_OUT in the timer top is connected with bit9 of the input multiplexer.
All six timer outputs (TIM_OUT0-5) are connected to ASIC outputs via GPIOs and the Interrupt Controller (IRQ21-26) (Kap. 5.4.1).

It is advisable to share the timers between the application and the PN stack. Timer 0-2 should use for the PN stack and Timer 3-5 for the application.

3.7.4 F-Counter

In the case of F-applications a further timer independent of the system cycle has to be provided in addition to the system timer. The functionality corresponds to the F-counter function of the ERTEC 200 blocks.
The F-counter is triggered through a separate input BYP_CLK. Triggering of the F-counters is not possible in the Clock Bypass mode (CONFIG(2)='1') (see Chapter 4.2).

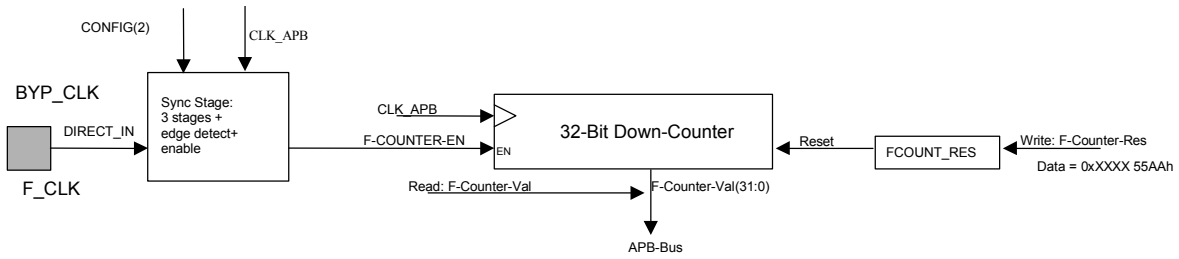


Figure 23: Block diagram F-timer

3.7.4.1 Description of function

The asynchronous input signal F_CLK of the external independent time base is applied to a synchronization stage via the input pin BYP_CLK. The SYNC stage is realized with 3 FF stages in order to reliably exclude metastable states at the counter input from arising.
In a downstream stage for edge detection the count pulses (F-counter-EN) are generated and applied to the 32-bit down counter. All the flipflops of the circuit operate with the APB cycle (125 MHz).

The 32-bit counter is reset to 0x0000 0000h with the asynchronous block reset (XRESET and XSRST, not visible in Figure 23). The next count pulse sets the counter to 0xFFFF FFFFh. Every further pulse carries out a decrement of the counter state. If the counter reach 0x0000 0000h the next count pulse set the counter to 0xFFFF FFFFh(WrapAround). The F-Counter_Res register (see Chapter 5.3.13) is set by writing the value 0xXXXX 55AAh (X:= don't care) to the address of the F-counter Reset register (word or half-word), resulting in a reset of the 32-bit counter to 0x0000 0000h. The FCOUNTER_RES register is reset automatically back to 0 one cycle later.

The complete 32-bit counter state is switched to the APB bus by a read access (word) to the F-counter. A consistency access to the counter value has to be by a word access. Half-word and byte accesses are theoretically possible, but deliver an inconsistency counter value.

3.7.4.2 Application information

The maximum permissible input frequency for F_CLK amounts to ¼ of the APB clock at a quartz failure at a 50% duty cycle.
In regular operation the PLL supplies 500 MHz at a quartz supply of 25 MHz.
In the case of a quartz failure at the ERTEC 200P a minimum output frequency is set at the PLL. This freewheel frequency of the PLL amounts to 100 - 300 MHz. At a quartz failure this results in a minimum APB_CLK frequency of $PLLOUT_{min}/4$ (~ 25 MHz).
For safety reasons the externally applied independent F-CLK should amount to a maximum of ¼ of the minimum APB clock possible (at quartz failure) in order to reliably prevent malfunctions in the edge recognition.
Currently 32768 Hz is foreseen as the independent F-CLK. This ensures a very high safety distance to the internal system clock.

3.7.5 ARM926 Watchdog

3.7.5.1 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 chapt: 5.4.1).
- After the watchdog Counter1 expires (XWD_OUT1) the 'XRES_ARM926_WD' is generated (see chapt: 3.8.2.4).

3.7.5.2 Block diagram

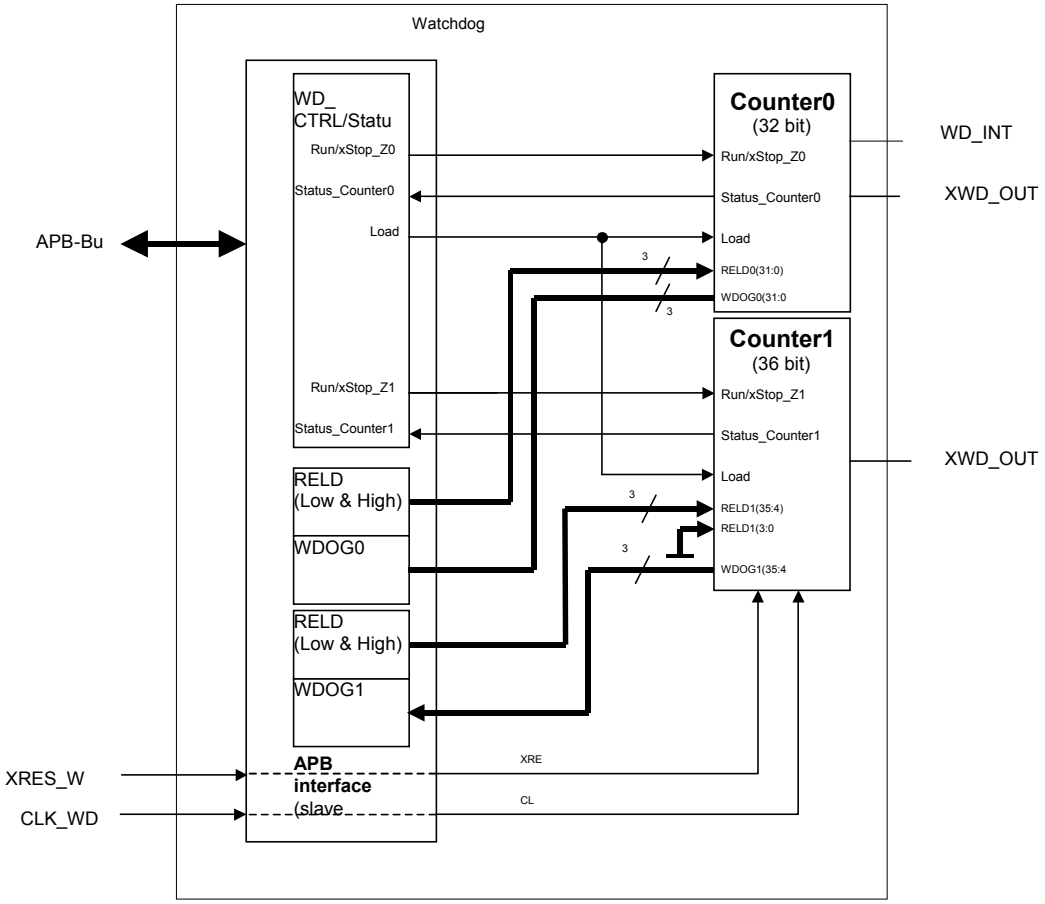


Figure 24: Block diagram ARM926 Watchdog

Counter0:
Counter0 is a 32-bit wide counter output that counts to 0 starting with the value passed in the RELD0(_LOW/_HIGH) register with the clock pulse from the CLK_WD pin. The watchdog is (re)started with Run/xStop_Z0=1 and, when required, stopped with Run/xStop_Z0=0.
The output is XWD_OUT0=0 when the watchdog is stopped. If the watchdog has been started and Counter0 ≠ 0, then the output is XWD_OUT0=1, otherwise XWD_OUT0=0.
Denote that in the case that Counter0 is preloaded with 0 and is started, the output XWD_OUT0 is also 1 for one clock cycle. Probably this is not a use case, this is stated only to get a whole picture of the circuit.

If the input is Load=1, the watchdog will be triggered, i.e. the Counter0 will be loaded with the value contained in the RELD0(_LOW/_HIGH) register and will continue counting from this value.
The WDOG0 register can be used to fetch the current value of the counter.
The Status_Counter0 output is active only when Run/xStop_Z0=1 and Counter0 has expired.
An expired Counter0, i.e. Counter0 decremented to 0, initiates an interrupt (WD_INT). For a Timing Diagram see also figure "XWD_OUT0/ WD_INT signal sequence"

Counter1:
Counter1 is a 36-bit wide counter that counts to 0 starting with the value passed in the RELD1(_LOW/_HIGH) register with the clock pulse from CLK_WD pin. The RELD1(_LOW/_HIGH) register contains only the high order 32 bits of the counter. The four low order bits of the counter are always loaded with the value 0x0.
The watchdog is (re)started with Run/xStop_Z1=1 and, when required, stopped with Run/xStop_Z1=0. If Counter1 is stopped or it is started but has not yet attained the value 0, then the output is XWD_OUT1=1. The output is XWD_OUT1=0 when Counter1 has expired.
If the input is Load=1, the watchdog will be triggered, i.e. the upper 32 bits of Counter1 will be loaded with the value contained in the RELD1(_LOW/_HIGH) register and the watchdog will continue counting from this value to zero.
The WDOG1 register can be used to fetch the current value of the counter (only the upper 32 bits).
The Status_Counter1 output is active only when Run/xStop_Z1=1 and Counter1 has expired.

CTRL/Status register (X=0,1):
The register is 32-bit wide and contains data only in the lower 16 bits. The upper 16 bits contain a special signature (see also 'write protection').

- Run/xStop_ZX-Bits: Starts and stops the Counter0.
- Load: The Load signal acts simultaneously on both counters (provided they have been enabled). The Load signal from the register is synchronized for the counter side. An increasing edge at the load input of the counter loads the CounterX with the value from RELD0(_LOW/_HIGH) or RELD1(_LOW/_HIGH).
  The software does not need to reset the load bit.
- Status_counterX: This bit is a single status bit present separately for each counter. If the bit is set, then the associated counter has attained the value 0. If Run/xStop_ZX=0, then the associated status bit always reads logical 0.

RELD0_LOW & RELD0_HIGH:
These two 32-bit registers contain user data only in the lower 16 bits. Each of the upper 16 bits is reserved for a special signature (see also 'write protection'). Thus, the 32-bit value for the counter consists of the lower halves of both registers. The register values must be changed only when the counter is stopped (Run/xStop_ZX = 0).

RELD1_LOW & RELD1_HIGH :
These two 32-bit registers contain user data only in the lower 16 bits. Each of the upper 16 bits is reserved for a special signature (see also 'write protection'). Thus, the 32-bit value for the counter consists of the lower halves of both registers. Only the upper 32 bits of the reload value can be specified for these counters – the lower 4 bits are always logical 0. The register values must be changed only when the counter is stopped (Run/xStop_ZX = 0).

WDOG0 & WDOG1:

These two registers can be used to read the current values of the two counters. Only the upper 32 bits of the current counter value can be read for Counter1. The content of the two registers will be updated after each increasing edge of the counter cycle clock (CLK_WD). Thus, without requiring the wait state, a read access to these registers always returns the contents of the two counters after the last recognized counter cycle

3.7.5.3 Signal waveforms

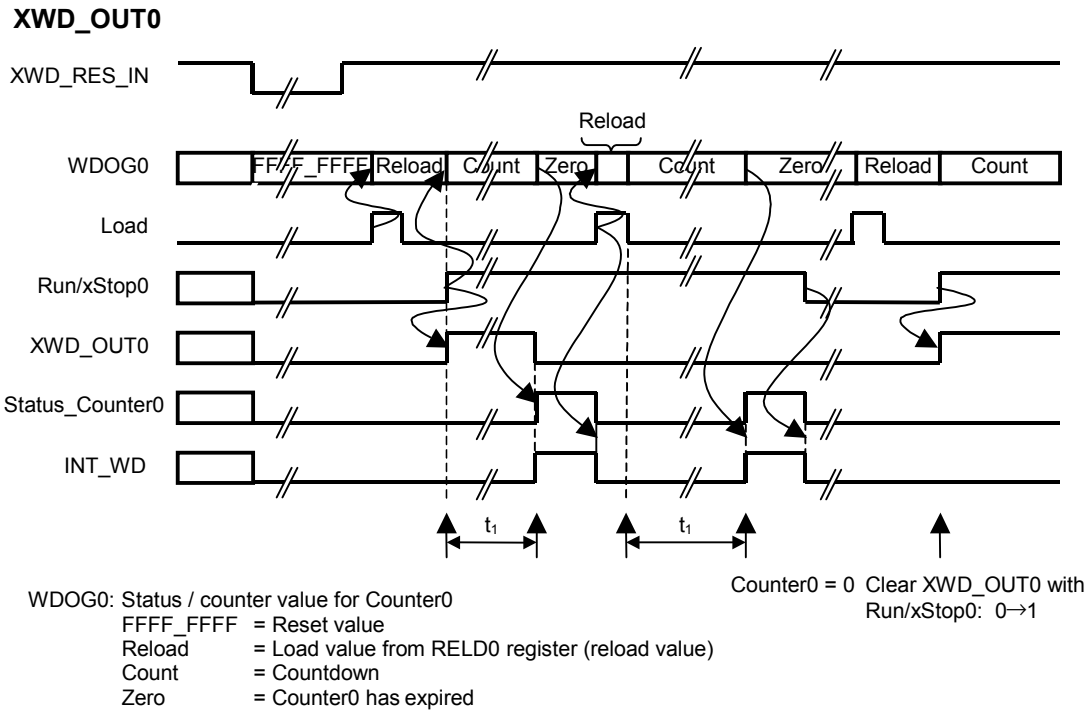


Figure 25: XWD_OUT0/ WD_INT signal sequence

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

- the XWD_OUT0 output becomes active (=0),
- the "Status_Counter0" status bit is set and
- the INT_WD interrupt signal becomes active (the increasing edge for INT_WD initiates the interrupt).

A retrigger pulse (Load=1 & Reload-value ≠ 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 sec, t_{1MAX} = 34,36 sec, Interval = 8 ns

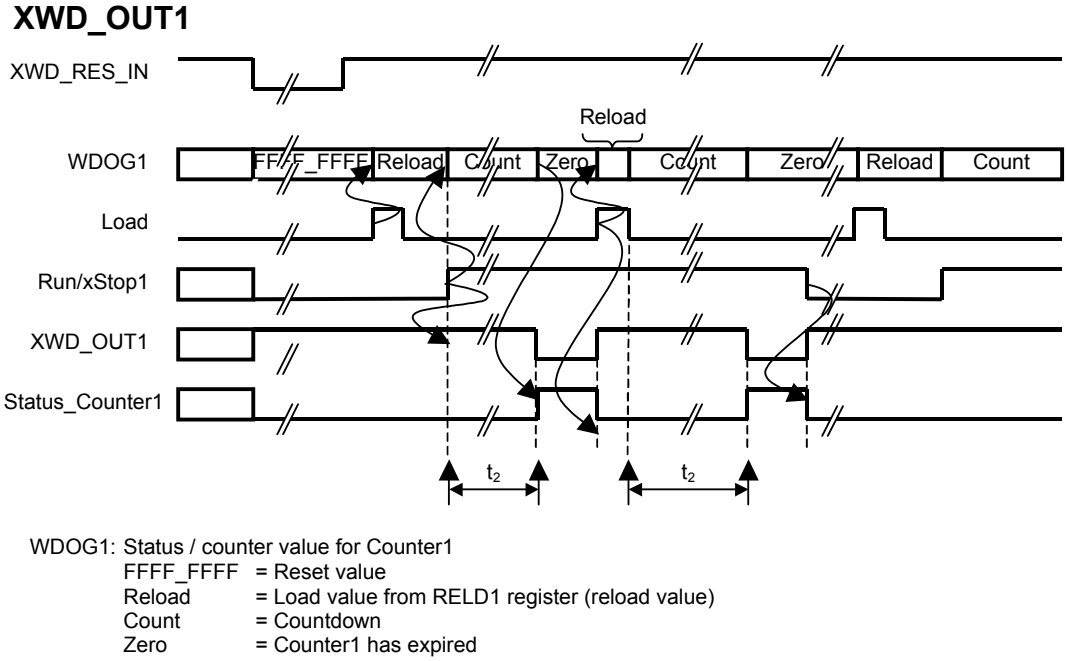


Figure 26: XWD_OUT1 signal sequence

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 sec, t_{2MAX} = 549.76 sec, Interval = 128 ns

3.7.5.4 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.

3.7.5.5 Starting the Watchdog

The following sequence describes the initialization phase which at first has to be performed by software:

- Write a proper value for RELD0/1 (_LOW/_HIGH)

- Load the values by writing to the Load register
- Activate the watchdog by writing 1 to the Run/xStop_Z0/1 registers

3.7.5.6 Notes

- 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 signalized to the external host over a GPIO port.
- For embedded single processor systems with ERTEC 200P the watchdog function is mandatory.

3.7.6 SPI1/2

The SPI1/2 interfaces are "master-slave" SPI function blocks. Interfacing to the APB bus is effected through a 16-bit interface.
The interfaces are realized with a soft macro of ARM (PL021). The base frequency for the bit-rate generation amounts to 125 MHz.

The following modes are supported by the macro:

- Motorola SPI-compatible mode
- Texas Instruments Synchronous Serial Interface
- National Semiconductor Microwire Interface

The SPI macro has the following features:

- programable Bitrate
  Master: 1922.27 Bd – 25 MBd
  Slave: 1922.27 Bd – 20,83 MBd

The SPI module can be controlled by the ARM926 or the GDMA Controller. Access by the external host via the Host Interface is also possible, but without interrupt support. The Module FIFO Interrupt is not activated or deactivated until 4 entries have been effected (depending on the transfer direction).

- SPI1/2_SSPRXDMA:
  The RX FIFO contains at least one character and is not empty. Since the SPI cannot know when the last character has been read in, the GDMA has to be operated in SINGLE-byte access mode (AHB) here. 1 to 65536 characters can thus 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) has to be enabled for operation. The TX FIFO is at least half empty and can thus hold 4 characters. The GDMA has to be operated in INCR4-byte access mode (AHB) so that a FIFO overrun does not happen. If the transmission length is not modulo 4, the remaining characters are transferred by the GDMA Controller by INCR Burst byte (indefinite length). 1 to 65536 characters can thus be transferred per DMA request, the job consists of a transfer entry.

The following SPI interrupts are switched to the ARM Interrupt Controller (IRQ13-16) (Chapter 5.4.1):

- SPI1/2_SSPINTR Combined interrupt

- SPI1/2_SSPRORINTR Overrun Error interrupt

For byte-specific operation of the SPI1/2 interfaces by the ARM926 the following internal FIFO status is switched to the ARM Interrupt Controller (ARM-ICU) from the respective SPI modules (IRQ28-31 see Chapter 5.4.1):

- SPI1/2_RNE Receive FIFO not empty (corresponds to SPI1/2_SSPRXDMA)
- SPI1/2_TFE Transmit FIFO empty

The status information is sampled by the internal SPI-IP status register SSPSR.
This ensures that the required timing for stable redundancy communication between the IM modules is ensured.

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 correspondingly or the corresponding SPI interface is set as the boot medium via the boot pins. In this case the bootloader in the internal ROM has to enable this SPI interface through corresponding GPIO Control register parameterization.

The SPI1/2 interfaces support the following external signals respectively:

- 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, these signals drive directly. They are not subject to an Output Enable control!

The following baud rates result for the cycle output of the synchronous serial interfaces depending on the parameters of the SPI1/2.

F_{CLKOUT} = 125 MHz / (CPSDVR*(1+SCR))

With the following applying for the parameters:

SPI master:	CPSDVR	:= (2..254), only in steps of 2
	SCR	:= (0..255)
SPI1 slave:	CPSDVR	:= (2..254), only in steps of 2
	SCR	:= (2..255)

3.7.7 UART1-4

The ERTEC 200P contains 4 UARTs.
The UART is realized with the soft macro of ARM (PL011). This is *similar* to the standard UART 16C550. For a detailed description of the registers and the individual functions refer to the document 'DDI0183F_uart_pl011_r1p4_trm.pdf' /15/.

The soft macro PL011 deviates 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
- The internal register address mapping and the register bit functions differ

The following 16C550 features are not supported by PL011:

- 1.5 stop bits (1 or 2 stop bits only are supported)
- Independent 'Receive Clock'

The UARTs can be controlled by the GDMA and the ARM926.

- UART1-4_RX-FIFO not empty:
  The RX FIFO contains at least one character and is not empty. Since the UART cannot know when the last character has been read in, the GDMA has to be operated in SINGLE-byte access mode (AHB) here. 1 to 65536 characters can thus be transferred per DMA request, the job consists of a transfer entry.
- UART1-4_TX-FIFO half-full or less:
  The TX FIFO is at least half empty and can thus hold 8 characters. The GDMA has to be operated in INCR8-byte access mode (AHB) so that a FIFO overrun does not happen. If the transmission length is not modulo 8, the remaining characters are transferred by the GDMA Controller by INCR Burst byte (indefinite length). 1 to 65536 characters can thus be transferred per DMA request, the job consists of a transfer entry.

The following UART interrupts are switched to the ARM Interrupt Controller (IRQ5-12) (Chapter 5.4.1):

- UART1-4_UARTINTR: Combined interrupt (Modem Status, Receive FIFO, Transmit FIFO, Receive Timeout, Error);
  individual interrupts can be masked
- UART1-4_UARTEINTR: Error Interrupt

The baud rates are derived from the APB cycle (125 MHz). The baud rate is calculated in accordance with the following equation:

$$\mathbf{BR = F_{UARTCLK} / (BRD * 16) \quad or \quad BAUDDIV = (F_{UARTCLK} / (16 * BR))}$$

With BAUDDIV = (BRD_I),(BRD_F) (e.g. at BAUDIV = 1.085 → $BRD_I = 1$, $BRD_F = 0.085$)

BAUDDIV encompasses an integer component (BRD_I) and a fractional component (BRD_F). The value (m) for setting the fractional divider is calculated in accordance with the following equation:

$$\mathbf{m = integer ((BRD_F * 64) + 0{,}5)}$$

The baud rate error is calculated in accordance with the following equation:

$$\mathbf{E_P = ((BR - BRI)/BRI) * 100\ [\%]}$$

With:
- $F_{UARTCLK}$ = UART base frequency = APB cycle frequency (125 MHz)
- BR = Baud rate
- BRI = Ideal baud rate
- BRD_I = Integer component of the Baud Rate Divisor to be programmed
- BRD_F = Fractional component of the Baud Rate Divisor to be programmed
- m = Fractional value to be set for the Fractional Divider (1/64 interval)
- E_P = Percentage deviation of the baud rate from the ideal baud rate

Table 11 lists the baud rate values to be set for the baud rates and the deviations from the standard baud rates. The UART (incl. APB interface) is supplied by a clock with 125 MHz.

BRI	BRD_I (integer)	m (fractio-nal)	BR	E_P %	Comment
460800	16	61	460829,49	+ 0,01	Debugger
230400	33	58	230414,75	+ 0,01	IO-Link
187500	41	43	187476,57	- 0,01	
115200	67	52	115207,37	+ 0,01	PC
76800	101	46	76804,92	+ 0,01	PC
57600	135	41	57597,05	- 0,01	PC
38400	203	29	38399,51	~ 0	IO-Link
19200	406	58	19199,75	~ 0	PC
14400	542	34	14400.09	~ 0	PC
9600	813	51	9600,06	~ 0	PC
4800	1627	39	4799,98	~ 0	IO-Link
2400	3255	13	2400,00	~ 0	PC
1200	6510	27	1199,99	~ 0	PC
110	Not possible!	Not possible!	Not possible!		

Table 11 : Baud rates UART at $F_{UARTCLK}$ = 125 MHz

The 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)

The UART1-4 interfaces are available as alternate functions at the GPIO. The alternate function can be set by software in the GPIO registers (see Chapter 5.3.14).
UART1 is available completely including the modem signals on the GPIO95-32. Only Transmit, Receive Data, CTS and RTS can be enabled respectively of UART2. Only Transmit and Receive Data are available of UART3-4. UART3 is planned for debugging and lies on the GPIO31-0. UART2 also lies on the GPIO31-0 and UART4 on the GPIO95-32.

The IO Link baud rates are supported with the residual error is < 0.01%. A total of up to 4 IO Link interfaces can be used. GPIOs are still required for a IO Link channel.

3.7.8 I^2C

The ERTEC 200P contains one I^2C module for general purpose applications (I^2C_3, master / slave interface). This module is located at the Toplevel and controlled by the ARM926EJ-S. Another I^2C module (master / slave interface) is located in the PN-IP and controlled by the PN-IP for the two POF Tranceiver for diagnosis. The two I^2C interfaces for the POF Tranceiver (I^2C_1/2) and the general purpose interface (I^2C_3) are multiplexed with other peripheral functions on the GPIO interface.

In the document 'I2C_Philips.pdf' /29/ (see Chapter 6.2) you will find the I^2C Bus Specification.

To use the direct access to the I^2C bus, the ARM must access appropriately the I^2C interface macro registers. These registers are contained in the I^2C interface address area. The complete functional description of the I^2C interface macro and the included registers is contained in the MI2C document.

The I^2C interface contains 2 registers (SW_I2C_EN and SW_I2C_CTRL) which allow to control the I^2C-Bus signals by Software. This software interface is enabled by setting bit 0 in register SW_I^2C_EN. If this software interface is enabled, the rest of the I^2C interface cannot be used.

The 8Bit I^2C modules are connected to the APB / SC-Bus (PN-IP). Only word addresses are used for addressing the internal registers. In the case of a write to the module the bit position 31-8 are ignored and in the case of a read from the modul these bit positions are driven with '0'.

Baudrate generator:

In contrast to the MI2C IP specification, the **C**lock **C**ontrol **R**egister (MI2C_CCR) in the I^2C interface cannot be written, because the register is not implemented. The Clock Control Register for the general purpose I^2C interface (I^2C_3) is contained in the SCRB (CCR_I2C). This allows the SCL clock pulse frequency on the I^2C bus to be set (see Chapter 5.3.8, System-Control-Register-Block (SCRB).

For a baudrate of 100 kBaud on the I^2C bus the module requires an internal enable signal of 1 MHz. Therefore the input clock has to be divided. This will be done by configuring the register CCR_I2C in the SCRB module. The register CCR_I2C is not configured after reset.

Formula for calculating the divider value:

((PCLK MHz) / (CCR_I2C + 1)) = 1 MHz

Example:

If PCLK = 125 MHz

then CCR_I2C = 124d = ("01111100b")

Only the baudrate of 100 kBaud is supported, no one more!

IO Expansion Unit:

The "IO Expansion Unit" is used to operate a maximum of 4 commonly usable 8-bit IO expanders connected to the I^2C interface of the ERTEC 200P. The corresponding registers are located in the I^2C address area.

Once configured, the unit automatically reads data from the expander into the register DATA_IN_n (when CTRL_n (IN) = 1) and writes data from the register DATA_OUT_n into the expander (when CTRL_n (OUT) = 1), where n = expander 1 to 4.

The register ADDR_n is used to store the associated slave address of the various expanders. Although EX_ADDR_n is implemented as an 8-bit register, only bits 7....1 are used as slave address. Bit 0 of the register is used by the "IO Expansion Unit" to distinguish between a read access and a write access.

In automatic operation, the registers EX_ADDR_n and EX_DATA_OUT_n must be configured before access can be started by writing to EX_CTRL_n. Depending on the setting of the register CTRL_n (MODE), this function can be used either periodically or as required.

If this register is configured for "as required" (= 0), then CTRL_n (IN) and CTRL_n (OUT) will be reset after the I^2C access. The service has to be disabled during the direct access to the I^2C bus (CTRL_n (OUT) = CTRL_n (IN) = 0).

The CTRL_n (status) bit indicates the success of the last transfer. If an error has occurred, this bit is set to ‘1’; whereas it remains ’0’ after a successful transfer. The internal status register for the I^2C module is monitored here.

The activated "IO Expansion Unit" takes control of the I^2C macro. This prevents the ARM926 from accessing the I^2C until one of the four expanders has been processed. The ARM926 must deactivate all IO Expander services in order to access the I^2C. Then the ARM926 must wait until the register CTRL_n (BUSY) indicates the end of all automatic transfers. The microprocessor now has full access to the I^2C and operates in immediate access mode. Because the default value set by the software is no longer valid, the register I2C_CNTR must be reconfigured to switch from immediate access to automatic operation.

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 I^2C 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 I2C_DATA does not have the same source or the same target when it is addressed with a write/read access. When this register is read, only the data transferred on the 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 in this way:

   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 1 write to the CNTR register.
6. When the I^2C macro is in slave mode software must ensure that all interrupts of the the I^2C macro are recorded, otherwise data could get lost or the I^2C macro may not respond to its slave address.

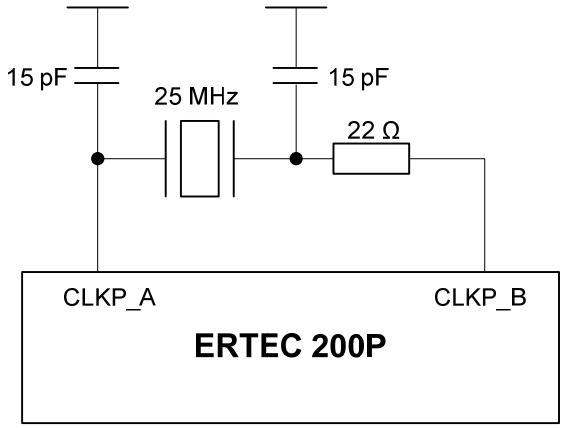
3.8 General functions

3.8.1 Cycle generation

With the exception of the JTAG cycle and the PHY cycle all the cycles are generated by the integrated PLL.

3.8.1.1 Cycle generation through PLL

The greater part of the ERTEC 200P runs with a synchronous 125 MHz or 250 MHz cycle. These cycles are generated by means of an integrated PLL. The PLL is fed with a 25 MHz cycle signal. The PLL cycle supply comes from an integrated quartz oscillator to which and external 25 MHz quartz has to be connected (CLKP_A/B).
The external quartz circuit is shown in **Figure 27** (the capacitance and resistance values listed here apply for the quartz type "TSX-3225" of Epson Toyocom):



Note:
The input capacitance of the integrated oscillator TDOSAC33N32M *in ERTEC 200 P is* ***4.2pF*** *(influence of the casing is included)*

Figure 27: Quartz circuit

3.8.1.2 Cycle supply for the PHYs and the Ethernet MACs

Both Ethernet MACs are connected directly via MII to the integrated PHYs.Figure 28 shows the cycle supply in principle for the PHYs and the Ethernet MACs.
The cycle supply of the PHYs is provided directly by the CLKP_A (quartz oscillator: 25 MHz). The PHYs generate the cycle signals RX_CLK and TX_CLK for the corresponding Ethernet-MACs. For measuring purposes the configuration pin CONFIG(0), see Chapter 4.2, can be used to enable the 25 MHz reference cycle of the integrated PHYs at the pin REF_CLK.

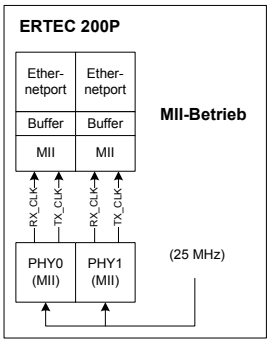


Figure 28: Internal Clock of the Ethernet interface

3.8.1.3 Cycle supply for JTAG

The debug interface of the ARM926EJ-S and the ETM macro cell of the ARM926EJ-S are operated via the JTAG interface. The cycle supply is effected via a separate JTAG-TCK. The frequency range of JTAG-TCK lies at 16/32 MHz. A maximum of 16 MHz is achieved for debuggers that do not support the RTCK Clock at the JTAG Interface. Otherwise 32 MHz are ensured.

3.8.2 Reset system of the ERTEC 200P

In addition to the external PowerOn Reset (XRESET), the possibility of an additional external Hardware Reset by the debugger (XSRST), an internal ARM926 Watchdog Reset (XRES_ARM926_WD) and various resets per software are provided for the asynchronous reset of the ERTEC 200P. The reset matrix (Figure 29) shows all the reset sources (rows) and the effects of the various circuiting parts (columns).

A PowerOn Reset (XRESET) is applied to the ERTEC 200P via an external pin. This PowerOn Reset resets the complete circuit (including cycle system). The XSRST pin is available for a hardware reset of the debugger. In the process the cycle system is not reset and communication via the JTAG interface is possible during this reset phase. The ERTEC 200P can be monitored by an ARM926 Watchdog. When an ARM926 Watchdog event (XRES_ARM926_WD) occurs, the ERTEC 200P is reset. The SCRB register 'ASYN_RES_CTRL_REG' (see Chapter 5.3.8) allows the PN-IP to be excluded from the reset when a Watchdog reset is carried out (EN_WD_RES_PN).

In addition, an internal asynchronous reset for the ARM926 Core system (RES_SOFT_ARM926_CORE) can also be triggered per software via the 'ASYN_RES_CTRL_REG'.

The triggering event of the last reset for the ARM926EJ-S Core can be read out of the SCRB register RES_STAT_REG (see Chapter 5.3.8).

Resetmatrix ERTEC 200P

Events / Sink	Debugger / Direction (Pin XSRST)	ARM926-Core (XRES_ARM926_CORE)	ARM926-TCM (XRES_SYS)	PNIP-TCM (XRES_PNIP)	PNIP (XRES_PNIP) (PNIP: res_sys_n_i und res_time_n_i)	Taktsystem (XRESET_CISRS)	JTAG_NTRST (EAS926)	Toplevel (XRES_SYS)	SCRB-TCM926 (XRES_SYS)	RES_SOFT_RETURN_ADDR (XRES_SYS)	SCRB PHY_STATUS SCRB ASYN_RES_CTRL_REG Slice3 - EN_WD_RES_PN (XRES_PNIP)	SCRB ASYN_RES_CTRL_REG Slice5 - RES_SOFT_KRISC32_CORE	SCRB RES_STAT_REG
PowerOn Reset (Pin XRESET)	x / out	x	x	x	x	x	x	x	x	x	x	x ('1') when PowerOn Reset	x ('4h') when PowerOn Reset
Debugger (Pin XSRST)	- / in	x	x	x	x	-	-	x	x	x	x	set to '1'	set to '4h'
JTAG Reset (Pin XTRST)	- / in	-	-	-	-	-	x	-	-	-	-	-	
Watchdog Reset ARM926 + PN-IP (XRES_ARM926_WD + Logic)	- / in	x (Pulse duration)	x (Pulse duration)	x (Pulse duration)	x (Pulse duration)	-	-	x (Pulse duration)	x (Pulse duration)	x (Pulse duration)	x (Pulse duration)	set to '1'	set to '1h'
Watchdog Reset ARM926 without PN-IP (XRES_ARM926_WD + Logic)	- / in	x (Pulse duration)	x (Pulse duration)	-	-	-	-	x (Pulse duration)	x (Pulse duration)	x (Pulse duration)	-	-	set to '1h'
SW Reset ERTEC200+ without PN-IP/PHY (RES_SOFT)	- / in	x (Pulse duration)	x (Pulse duration)	-	-	-	-	x (Pulse duration)	x (Pulse duration)	x (Pulse duration)	-	-	set to '2h'
SW Reset PN-IP/PHY (RES_SOFT_PN)	- / in	-	-	x (Pulse duration)	x (Pulse duration)	-	-	-	-	-	x (Pulse duration)	set to '1'	
Core-Reset ARM926-Core (RES_SOFT_ARM926_CORE)	- / in	x (Pulse duration)	-	-	-	-	-	-	-	-	-	-	set to '8h'

Legend

(grey) Reset on the modul

Figure 29: Reset matrix of the ERTEC 200P

3.8.2.1 Asynchronous PowerOn Reset

The asynchronous PowerOn Reset is connected via the pin XRESET at the ERTEC 200P. As a reaction to this reset the complete circuit (including cycle system) of the ERTEC 200P is reset and the configurations pins latched off (see Chapter 4.2). During booting of the ERTEC 200P the PowerOn Reset has to be present stably for at least 30 µs after the the voltage is present stably. Afterwards the PLL boots and after a further 1000 µs the PLL is latched. This time until the PLL latches is designated t_{LOCK}. Internally the PowerOn Reset phase is extended fixed for this period (the PLL-Lock is not evaluated) and the cycle system is not cut in until the end of the boot phase. The internal reset remains active for a further 16 clocks after the cycle system has been booted to carry out the reset internally. Communication of the debuggers via the JTAG interface is not possible during this time.

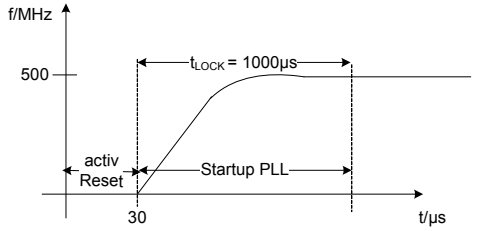


Figure 30: Booting phase of the PLL

The locked state of the PLL is monitored by the hardware. The IRQ49 interrupt reports whether the PLL has lost its input cycle (quartz break) or the PLL is not locked (PLL Monitor, monitors the input to output frequency). In addition, the two error states can also be sampled directly via the SCRB register 'PLL_STAT_REG' (see Chapter 5.3.8).
A filter integrated in the ERTEC 200P ensures that spikes <= 40 ns (best case) are suppressed on the input XRESET.
While the XRESET pin is active the bidirectional pin XSRST is switched to output and active. This allows the debugger to recognize the Power On Reset phase.
In order to carry out an analysis of the reset event after the system has been restarted, the PWRON_HW_RES bit is set in the RES_STAT_REG at a PowerOn Reset, that remains unaffected by the triggered reset function. During a restart the software can read out the RES_STAT_REG (see Chapter 5.3.8).

3.8.2.2 Asynchronous Hardware Reset

The hardware reset is triggered via the pin XSRST by the external debugger. XSRST is a bidirectional IO cell with Open Drain output. During the active XSRST phase the complete internal logic is reset, but not the cycle system. In addition, the configuration pins are not latched off. During this hardware reset phase the debugger can communicate via the JTAG Interface with the embedded ICE Logic to, for example, load a breakpoint. This means that single-stepping is possible from the reset address.
A filter integrated in the ERTEC 200P ensures that spikes <= 40 ns (best case) are suppressed on the input XSRST.
In order to carry out an analysis of the reset event after the system has been restarted, the PWRON_HW_RES bit is set in the RES_STAT_REG at a Hardware Reset, that remains unaffected by the triggered reset function. During a restart the software can read out the RES_STAT_REG (see Chapter 5.3.8).
When booting after a Hardware Reset the system uses the boot mode that was latched off internally at the PowerOn Reset.

XSRST is switched activate to the debugger when a PowerOn Reset is applied. Under no circumstances may XSRST be activated at a 'RES_SOFT_ARM926_CORE'. The debugger could not run then while the core is in reset.

3.8.2.3 Asynchronous JTAG Reset

The JTAG reset is triggered via the pin XSRST by the external debugger. In the process only the Embedded ICE logic of the ARM926EJ-S is reset. To ensure that this Embedded ICE Logic also has a defined state during operation without a debugger, this logic is also reset with the PowerOn Reset (XRESET). A logic operation is realized internally to this purpose.
A filter integrated in the ERTEC 200P ensures that spikes <= 40 ns (best case) are suppressed on XSRST. Normally a spike on XTRST is not passed on in the JTAG Controller because a sequence via TDI/TMS and TCK is required to this purpose.

Note for the module development:
For regulations on external pull circuiting of the XTRST pin see JTAG-Interface in Chapter 4.1.1.

3.8.2.4 Asynchronous ARM926 Watchdog-Reset

The ARM926 Watchdog Reset represents a hardware-end monitoring of the software on the ARM926EJ-S. Basis for monitoring is a time that can be set in the Watchdog timer. This time starts with the activation of the watchdog. If no retriggering of the timer to its initial value is carried out within this time, Watchdog Reset (XRES_ARM926_WD) is triggered (output ARM926 Watchdog: WD_XWDOUT1). If the watchdog function (WD_RES_FREI_ ARM926) (see ASYN_RES_CTRL_REG in Chapter 5.3.8) is enabled, the ERTEC 200P is reset. The actual reset signal is laid via a parameterizable pulse generation. In order to carry out an analysis of the reset event after the system has been restarted, the ARM926_WDOG_RES bit is set in the RES_STAT_REG at a Watchdog Reset, that remains unaffected by the triggered reset function in. During a restart the software can read out the RES_STAT_REG (see Chapter 5.3.8).
For applications where the course of the watchdogs should not have a negative effect on the function of the PN-IP, it is possible to remove the PN-IP from the reset per watchdog (EN_WD_RES_PN = 0 in ASYN_RES_CTRL_REG).
Note: The bit EN_WD_RES_PN is always reset (->PN-IP underlies the ARM926 Watchdog Reset) when the PN-IP is reset (see Figure 29). Should an asynchronous software reset for the PN-IP by generated by the software switch, the software has to set 'EN_WD_RES_PN = 0' again subsequently, if a reset of the PN-IP when the ARM926 watchdog expires is to be prevented.

As preprocessing for the watchdog course an interrupt is generated for the ARM926 'WD_INT_ARM926' (see Chapter 5.4.1) and the preprocessing event 'WD_XWDOUT0' is reported to the external host via a GPIO pin.
The watchdog also expires when the cycle supply fails (for example quartz break). The PLL then changes to its freewheel frequency (100 - 300 MHz).
When booting after a Watchdog Reset the system uses the boot mode that was latched off internally at the PowerOn Reset.

3.8.2.5 Asynchronous Software Reset for the ERTEC 200P

In the ERTEC 200P an asynchronous Software Reset can be triggered by setting the bits 'RES_SOFT' in the RES_CRTL_REG (in the SCRB, see Chapter 5.3.8). The PN-IP and the PHYs are not reset in the process. In order to carry out an analysis of the reset event after the system has been restarted, the SW_RES bit is set in the RES_STAT_REG at an asynchronous Software Reset, that remains unaffected by the triggered reset function. During a restart the software can read out the RES_STAT_REG (see Chapter 5.3.8).
When booting after a Software Reset the system uses the boot mode that was latched off internally at the PowerOn Reset.

.

3.8.2.6 Asynchronous Software Reset for the ARM926EJ-S Core

The ARM926EJ-S Core (without TCM_Block_926) has an own reset that can be executed asynchronously by the software via the bit 'RES_SOFT_ARM926_CORE' in the SCRB register 'ASYN_RES_ CTRL_REG' (see Chapter 5.3.8). 'RES_SOFT_ARM926_ CORE' only acts on the ARM926EJ-S Core system and not on the TCM_Block_926 (see Figure 12). The TCM_Block_926 is reset with the XRESET, XSRST-, XRES_ARM926_WD or RES_SOFT.
In order to carry out an analysis of the reset event after the system has been restarted, the SW_RES_ARM926 bit is set in the RES_STAT_REG at an asynchronous Software Reset, that remains unaffected by the triggered reset function. During a restart the software can read out the RES_STAT_REG (see Chapter 5.3.8).
The asynchronous Software Reset for the ARM926EJ-S Core system is necessary after the bootloader has set the final TCM926 configuration. Only with a reset does the ARM926EJ-S take over the TCM926 configuration (DRSIZE for the D-TCM and IRSIZE for the I-TCM -> formed from TCM926_MAP register, see Chapter 5.3.8).

3.8.2.7 Synchronous Software Reset (PN-IP, PER-IF, Host Interface)

The PN-IP, the Periphery Interface and the Host Interface can also be reset synchronously by the software in the SCRB register 'SYN_RES_CTRL_REG' (see Chapter 5.3.8). These synchronous resets act on the SYN Reset inputs of the corresponding IPs and reset only the state machines and the local registers, but not the Parameters register and the AHB Interface. The synchronous reset does not act on the reset input of a flipflop. The software has to set the corresponding bits in the 'SYN_RES_CTRL_REG' and subsequently reset them. This allows the software to determine the reset state itself (short pulse: dynamic reset, continuous '1': disable state).

3.8.3 Modules and ASIC code

The ASIC code (Version Number of the ASIC) can be read from the ID register in the SCRB.

3.9 Booting the system (boot pins)

The ERTEC 200P can be booted via several sources. The boot modes required for the various configurations are implemented. The following boot and download media are supported:

- Booting via NOR Flash: It is possible to interface different blocks to the ERTEC 200P. NOR Flashes can be interfaced in the organization widths of 8 bits, 16 bits and 32 bits (Note: 32 bit organization widths for booting is not supported). Booting is effected via EMC: Peripheral Bank 0 (XCSPER_0).
- Booting via the XHIF interface for systems at which the code is downloaded from the host processor.
- SPI

The configuration is transferred via boot pins. The state of the boot pins is latched off in the Boot register during an active PowerOn Reset XRESET (see Chapter 5.3.8). This boot combination is read out by the processor and branches to the corresponding boot routine in accordance with the coding.
After the reset has been removed the boot pins change their function and are available as EMC address bits (see Chapter 4.3).

Detailed description about the boot pins is in the documentation ERTEC200 P Datasheet.

A three-stage boot model is used:

The primary boot loader, stored in the boot ROM, sets the used hardware so that the data can be read from the boot medium and copied to the boot RAM (D-TCM). These copied data contain the

secondary boot loader that then copies the actual program to its target (I-TCM926, SDRAM or SRAM) or at a host boot branches into the loaded main program.. The user knows which hardware components his system has and can effect the settings correspondingly.
The loaded secondary boot loader can or has to carry out a remapping as required so that the vector list is reallocated to the address 0x0000.0000h. All the vectors are assigned in the boot ROM so that a "deadlock" of the system cannot occur. However, when an interrupt occurs a restart is always carried out when the vector address of the boot ROM is activated.

3.9.1 Booting via NOR Flash

When booting from an external NOR Flash it is possible to select between 8- and 16-bit ROMs. Booting is effected via the EMC Peripheral Bank 0 (XCS_PER0). The interface is set to slow timing. The user then has to reconfigure the timing to his requirements. A Burst Flash is operated asynchronously during booting, so that no clock 'CLK_O_BF_0/1' is activated (default) either. If required, the user can reparameterize the Burst Flash to the synchronous mode. However, the CLK_O_BF_0/1 in the DRIVE_EMC register also has to be activated with 'EMC_BF_CLK_BF0BF1_EBL= 1' (see Chapter 5.3.8).
Selection is effected via the boot pins that are evaluated by the boot software and that parameterizes the peripheral base correspondingly. The address to which the jump is to be carried out is set at the address 0x0000_0000h of the Flash block. The primary boot loader reads the data word at Address 0 of the Flash block and jumps into the ROM in accordance with this address (caution not a mapping address!).
The ResetOut pin for the NOR Flashes is connected with the ERTEC 200P reset. Therefore the system waits at least 70 us until access to the Flash block is carried out. This procedure is necessary if a previously activated write or delete job was aborted through a reset. The Flash block requires this time to switch to the state "Reading data".

3.9.1.1 NOR Boot Flow

After hardware reset the primary bootloader (placed in bootrom) is executed, and the EMC interface (also memory bank 0 chip select signal XCS0) has default values, which consist in slowest timing.
The primary bootloader, place stack on top of DTCM, checks BOOT[3:0] and CONFIG[6:0] registers and sets EMC register data width accordingly.

Note: Flash Reset pin may be tied to the system reset circuitry which enables the system to read the boot-up firmware

After that ARM926 (Bootrom) expects the second level boot loader at AHB address 0x3000_0000 (which is base (offset 0x0000) of flash ROM) with a “LDR PC, =ROM_RESET” command.

Here an example of the second level boot loader entry in flash ROM (exception table):

```
; AHB address 0x3000_0000/ flash offset 0x0000
entry:
        LDR     PC, Reset_Addr    @; Reset ROM
        LDR     PC, Reset_Addr    @; Undefined instruction
        LDR     PC, Reset_Addr    @; Software interrupt
        LDR     PC, Reset_Addr    @; Opcode from illegal address
        LDR     PC, Reset_Addr    @; Data from illegal address
        NOP                       @; Reserved vector
        LDR     PC, Reset_Addr    @; IRQ = Standard interrupt
        LDR     PC, Reset_Addr    @; FIQ = Fast Interrupt


Reset_Addr:
 .   word      Reset_Handler
```

```
Reset_Handler:
      .weak      copy_second_level_boot
       B copy_second_level_boot
```

At the beginning the second level boot code must be copied from flash ROM to D-/ I-TCM (compare ARM TCM sharing and D-/I-TCM enabling by CP15 register) and I-TCM must be mapped to 0x0000_0000 (compare MEM_SWAP register) then second level boot loader in I-TCM could be executed.
Till now memory controller runs with default (slowest) timing, second level boot loader now has to set correct EMC settings for used flash and SDRAM, after that remaining parts of second level boot loader and the application could be copied from flash ROM to SDRAM and application could be started.

3.9.2 Booting via Host Interface

Booting of software code via the parallel Host Interface (XHIF) has to be executed actively by the external host processor. The secondary bootloader is first transferred from the host into the boot RAM (D-TCM) of the ERTEC 200P. Subsequently the actual download of the code image (into the I-TCM926 / SDRAM / SRAM) is carried out. The secondary bootloader subsequently branches into this loaded program.
After the reset the primary bootloader sets the XHIF Interface in accordance with the Config pins with the selected XHIF_CPU_Width, XHIF_POL_RDY and XHIF_ACC_Mode . In addition at least one address window is enabled for access to the boot RAM (D-TCM).

Loading times for a 32-bit XHIF Interface and an execution time of 12 cycles per write access (125 MHz → 100 ns for 4 bytes) are estimated below.

Load times:

- Secondary bootloader into the boot RAM (4 kbytes): 103 us
- 2 Mbytes into SDRAM (32 bits): 52 ms

3.9.2.1 XHIF Boot Flow

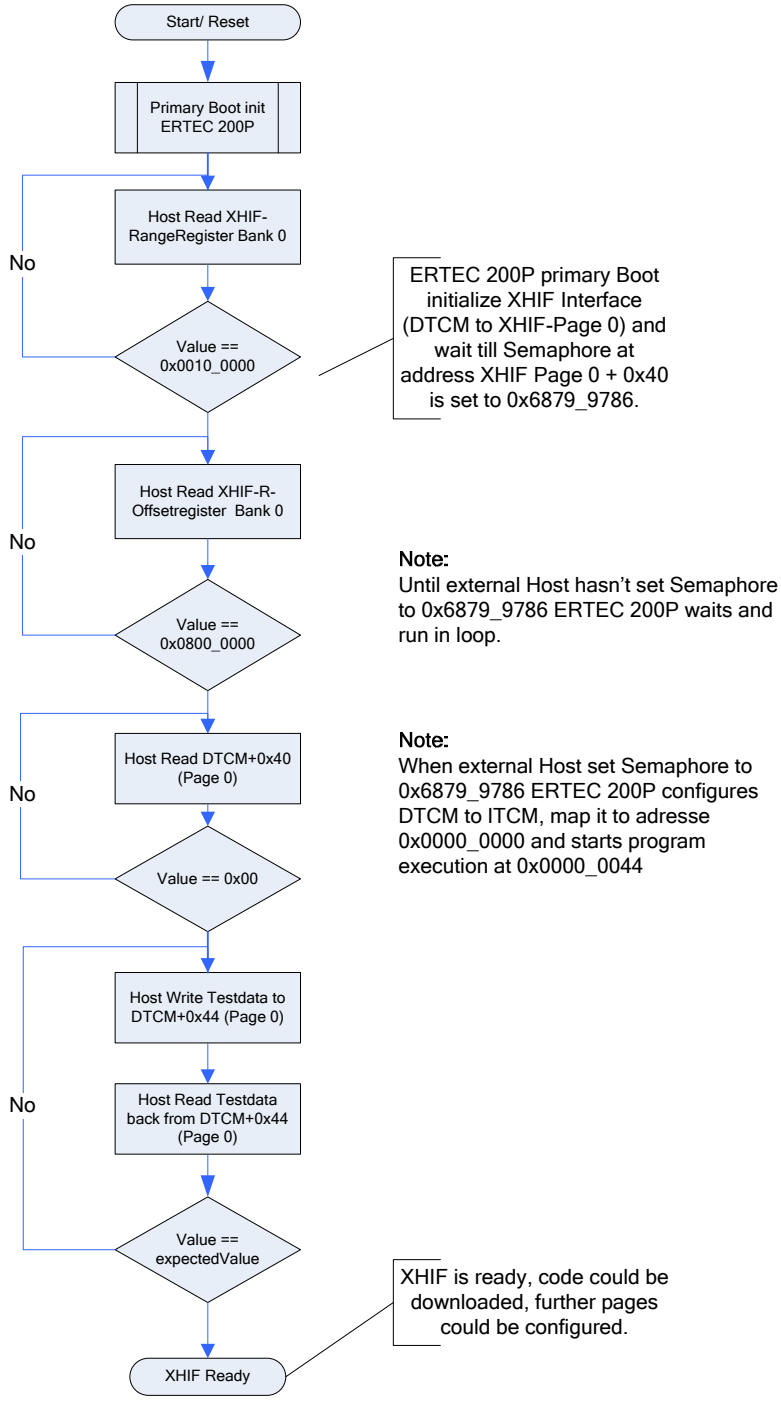
After hardware reset the primary bootloader (placed in bootrom) is executed, stack is placed on top of DTCM, BOOT and CONFIG registers are checked and XHIF interface is set accordingly (data width, XRDY polarity, …).
After that ERTEC 200P ARM926 waits for semaphore which will be set by external host.

The external Host has to download the software for ERTEC 200P, therefore he has to determine if ERTEC is ready to receive code by checking following sequence.

1) External Host reads on XHIF_XCS_R_A20 (see XHIF XCS_R memory map):
   Range Register Page 0 == 0x0010_0000 and Offset Register Page 0 == 0x0800_0000
   (This means primary bootloader has configured DTCM on Page 0)

2) External Host access XHIF_XCS_M (see XHIF XCS_M memory map):
   Read Page 0 Offset 0x40 == 0x0000_0000
   Write test data to Page 0 Offset 0x44
   Read test data from Page 0 Offset 0x44 and verify with written value

Compare also following flow diagram:



If verify is successful XHIF is ready and code (secondary bootloader) with maximum size of 64KB (DTCM block 3) could be downloaded to page 0.

Note: *Interrupt vectors must also be copied because DTCM block 3 will get ITCM and mapped to address 0x0000_0000.*

After download external host has to set semaphore at page 0 offset 0x40 to 0x6879_9786, this indicates ERTEC ARM 926 to configure DTCM block 3 (this is page 0) to ITCM an map it to address 0x0000_0000 (new interrupt vectors) and execute the code on offset 0x0000_0044.

Further program flow (who configures XHIF pages, downloading application to SDRAM, …) is implementation specified

3.9.3 Booting via SPI

3.9.3.1 SPI Boot Flow

After hardware reset the primary bootloader (placed in bootrom) is executed, stack is placed on top of DTCM, BOOT and CONFIG registers are checked and SPI interface is initialized as following:

- ERTEC 200P is clock master
- 8 Data Bits
- Transmit/Receive with leading MSB
- Idle Clock Line = 0
- Data a latched in with rising edge
- Data put out with falling edge
- Baud rate is 1 MBit/s
- SPI-Memory Chip-Select is implemented on Port GPIO31

NOTE: SPI flahes with read commando 0xE8 and 0x03 are supported.

After the configuration of the SPI interface the primary bootloader checks which read command is valid. Therefore it starts to read with command 0xE8 from address 0x0000 10 bytes and checks if there is a valid identifier 0x5A, if it found it proceed like described below, if not it restarts with read commando 0x03, if there is also no identifier, the watchdog is enabled and when expired a HW reset will occur (procedure starts again).
When a valid identifier is found, primary bootloader could calculate how many address bytes are required by external SPI flash (see diagram below, distance between read command and identifier defines the number of required address bytes).
After reading the identifier in the bit stream the number of words, which should be copied from SPI flash to User RAM (DTCM block 3), are expected (first high byte, then low byte). Following bytes are handled as data bytes and will be copied to User RAM.

Example:

Byte	1	2	3	4	5	6	7	8	9	10	11	...
TX	**0x03 / 0xe8**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**...**
	RD CMD											
RX	***0xff***	***0xff***	***0xff***	***0xff***	***0x5a***	***0x10***	***0x00***	***0x00***	***0x01***	***0xFF***	***0xF0***	***...***
					ID	***high byte***	***low byte***	***Data 0***	***Data 1***	***Data 2***	***Data 3***	...

Here 0x1000 words are copied from SPI flash, maximum is 256kByte (0xFFFF). User RAM is limited to 64kB.

Result in User RAM

User-RAM+0	**0x00**	**Data 0**	**Word 0**
User-RAM+1	**0x01**	**Data 1**	
User-RAM+2	**0xFF**	**Data 2**	
User-RAM+3	**0xF0**	**Data 3**	
User-RAM+4	**0x00**	**Data 4**	**Word 1**
User-RAM+5	**0x00**	**Data 5**	
User-RAM+6	**0x01**	**Data 6**	
User-RAM+7	**0x05**	**Data 7**	
User-RAM+8	**0x00**	**Data 8**	
User-Ram+0x104	**…..**	**…..**	**…..**

After coping all byte to User RAM, the code is executed at User RAM + 0x44, reloading code from SPI must be handled by secondary bootloader.

Note: *Valid Interrupt vectors must be placed because User RAM (DTCM block 3) will get ITCM and mapped to address 0x0000_0000.*

3.9.4 ERTEC 200P settings when leaving primary bootloader

When leaving primary bootloader ERTEC 200P is configured a following:

1. MMU is not enabled, default value
2. Watchdog is set to 251 ms and active (must be triggered)
3. SDRAM is not initialized
4. Stack pointer (SP) is set to 0x0800_fd00.
5. 64k I-TCM is mapped to address 0x0000_0000 and 192k D-TCM to 0x0800_0000 (expected by NOR-Boot)
6. Shift-Mode on EMC-Interface is active (*EXTENDED_CONFIG.ASYNC_ADDR_MODE* = 1)

3.9.5 Memory-Swapping

The reset vector of the ARM926 processor points to the address 0x0000_0000. Therefore the mirror range of the boot ROM lies as of address 0x0000_0000 after a reset (PowerOn, HW , SW and ARM926 Watchdog reset). In addition the boot ROM can be addressed in its original address range (see Chapter 5.1.1).

At the end of the boot process the EMC-SDRAM (max. 64 Mbytes) or the EMC Memory (max. 1x 64Mbytes, Chip Select Bank0) or I-TCM can be mapped to the address 0x0000_0000h so that the Exception Vector table for the ARM926EJ-S can be created in the address range 0x0000_0000 – 0x0000_001F. The original address areas for the boot ROM (in Segment 4), EMC-SDRAM (Segment 2) and EMC Memory (Segment 3) are influenced by memory swapping.

Swapping is effected by programming the register MEM_SWAP in the System Control Status Register Block SCRB (see Chapter 5.3.8). With the Reserved coding in the MEM_SWAP register swapping is not carried out (to be more precise: no memory area is displayed in Segment 0). If these addresses are then accessed at the AHB, a time-out (QVZ) is triggered.

3.10 Address space and time-out monitoring

In the ERTEC 200P various monitoring mechanisms for recognizing faulty addressing and time-out (ready time-out) are implemented.

3.10.1 Monitoring at the AHB end

A separate address space is assigned to each AHB master (ARM926-D, ARM926-I, PN-IP, GDMA, Host-IF). If an AHB master accesses an unused address range, the access is acknowledged with Error Response and an interrupt IRQ51 (see Chapter 5.4.1) is triggered in the ARM Interrupt Controller. In addition the address of the faulty access is stored in the SCRB register QVZ_AHB_ADR and the associated type of access (AHB Control signals: Read/Write, HBURST, HSIZE) in the SCRB register QVZ_AHB_CTRL (see Chapter 5.3.8).
The information about the master that triggered the access violation can be read in the SCRB register QVZ_AHB_M.
If an RD access at the Host-IF that was triggered by a parallel (XHIF) or serial (SPI) access was acknowledged with Error Response, an AHBerror IRQ is triggered additionally at the Event unit (Periphery Interface). This entry results in activation of the Host Interrupt XHIF_XIRQ.
The 3 diagnostics registers QVZ_AHB_ADR/CTRL/M are blocked for subsequent access violations until the register QVZ_AHB_CTRL has been read. However, only an unlocking takes place through the QVZ_AHB_CTRL register being read out - the content of the QVZ_AHB_ADR/CTRL/M register remains unchanged. Only a subsequent timeout (QVZ) results in an update of the QVZ_AHB_ADR/CTRL/M register.
If several AHB masters cause an access violation simultaneously (AHB-synchronously), only the violation of the highest-priority AHB master (prioritization in accordance with Table 2) is displayed (as described above).

3.10.2 Monitoring at the APB end

At the APB end monitoring of the APB address space takes place. In the case of false addressing in the APB address space access to the APB and AHB end is completed with an OKAY Response because the APB bus does not know signaling of the Response type.
An interrupt IRQ52 (see Chapter 5.4.1) in triggered in the ARM Interrupt Controller. In addition, the address of the erroneous access is stored in the SCRB register QVZ_APB_ADR (see Chapter 5.3.8).
The diagnostics register QVZ_APB_ADR is blocked for subsequent access violations until it has been read.

3.10.3 Monitoring in the EMC

Monitoring of the external Ready signal XRDY_PER is carried out in the EMC. If one of the 4 external memory areas that are selected via the output pins XCS_PER(3:0) is addressed, the memory controller waits for the input signal XRDY_PER (if Ready Control is activated in the corresponding configuration register ASYNC_BANK_x_CONFIG (Chapter 5.3.6)). In the register EXTENDED_CONFIG (Chapter 5.3.6) a time-out (QVZ) monitoring is activated that internally generates a Ready signal for the Memory Controller and the IRQ53 (see Chapter 5.4.1) in the ARM Interrupt Controller after a maximum of (1048575 + 1) x 16 AHB cycles. The monitoring duration is set in the register ASYNC_WAIT_CYCLE_CONFIG (Chapter 5.3.6).
In addition, the address of the erroneous access is stored in the SCRB register QVZ_EMC_ADR.
The diagnostics register QVZ_EMC_ADR is blocked for subsequent access violations until it has been read.

3.10.4 Monitoring in the modules

Monitoring for unassigned addresses is carried out in the modules 'PN-IP, PER-IF, I-Filter, Host-IF and SCRB'. If the software access these unassigned addresses, an error is generated by the module that is stored in the register 'MODUL_ACCESS_ERR' (see SCRB, Chapter 5.3.8). The error signal generated by the module respectively should have a minimum pulse duration of two system cycle periods if asynchronous (reference point: 125 MHz system cycle). A pulse duration of one system cycle period is sufficient for synchronicity. The polarity is high active.
An entry in this register 'MODUL_ACCESS_ERR' triggers the ARM926 Interrupt 'Modul_Access_Error' (see Chapter 5.4.1). At the MODUL_ACCESS_ERROR interrupt the software has to read the SCRB register 'MODUL_ACCESS_ERR' and thus identify the respective module. Subsequently the software in this module has to read the detailed data of the erroneous access into its ACCESS_ERROR register and as far as possible delete this register again immediately. Afterwards the software has to reset the MODUL_ACCESS_ERROR register in the SCRB. This is done by writing the value 0h to the register 'Modul_Access_Error'.

In case of read accesses to unassigned addresses in AHB modules an ERROR-ACK is generated additionally on the AHB (PN-IP, PER-IF). This triggers an exception at the ARM926EJ-S.

Additional access errors at the PN-IP:

- Byte write accesses to only half-word / word addresses
- Half-word write accesses to word addresses
- Writing to read-only registers

3.11 MEM Wrapper

The new technologies of ASIC design libraries (90 nm and below) also require error correction logic for the RAMs. Because of the smaller geometry of the cells, radiation particles may cause a change in the data bits (**S**oft **E**rror **R**ate, SER).
Error Detection and Correction (EDC) is found in many high-reliability and performance applications in order to improve system reliability; for example, in data storage systems or memory caches. It is more efficient in terms of performance and costs to correct an error rather than to re-transmit the data.

EDC parity encoder / decoder:
A combinatorial EDC encoder generates the parity information from user data. The EDC decoder checks if the data word bits matches to the added parity bits. In case of an error an EDC_Event interrupt is generated. The EDC decoder also corrects one bit errors.
Combinatorial means: only combinatorial logic is used inside parity encoder / decoder. This means no clock or reset signal is needed.

Important note:
The RAM itself will be not corrected in case of a Single Bit Failure. **The wrong data is corrected "on the fly" but there is no writeback to RAM in case on a Single Bit Error.**

RAM Initialization Unit (INIT_DONE):
After the reset becomes inactive, the RAM contains random data. Parity information and user data do not match. The EDC_Event interrupt (IRQ48) will be generated in case of a data read. A RAM Initialization Unit avoids this by filling up the whole memory with constant data (all zeroes) and the appropriate parity bits. After the initialization is done an INIT_DONE is generated and stored in the EDC_INIT_DONE register in the SCRB block (see Chapter 5.3.8). The Primary Bootloader waits until INIT_DONE is generated before accessing the RAM.

4 HARDWARE INTERFACES

The hardware interfaces are divided into functional pins and supply pins. The pins are described below in accordance with their function.

4.1 Assignment of the function pins

The function pins encompass the following groups:

- JTAG ports
- PHY ports (PHY interfaces + signaling)
- EMC (memory interface)
- Host Interface (parallel: XHIF) / GPIO95-32
- GPIO31-0
- Boot ports for the ERTEC 200P
- Configuration ports for the ERTEC 200P

The individual functions of the existing signal groups are described briefly below.

4.1.1 JTAG Interface

The JTAG interface is an interface with which controlling of the Boundary Scan register or debugging of the ERTEC 200P an be carried out. The JTAG reset is implemented intentionally without an internal pull resistance to allow various interfaces of the debugger. A filter integrated in the ERTEC 200P ensures that spikes <= 40 ns (best case) are suppressed on the JTAG reset XTRST. Normally a spike on XTRST is not passed on in the JTAG Controller because a sequence via TDI/TMS and TCK is required to this purpose.

The following table lists the various recommendations for the external pullup/-down circuiting of the JTAG interface signals

Note for the module development:
Pin XTRST on the module has to be circuited with a 10 KOhm pulldown to achieve the best possible interference immunity on the module (see the column "Circuit for Production" in the above table)! This results in a deactivation of the JTAG interface during operation so that interference pulses on the individual JTAG signals can no longer affect the ERTEC 200P function. The pulldown does not have any effect for the use of a debugger at the JTAG interface since the debugger pulls the signal XTRST active to '1'.

4.1.2 PHY Media Interfaceand PHY LEDs

Two 100BASE-TX/FX interfaces are available fixed. Two LEDs (Link, Activity) are provided per PHY interface for signaling the PHY states. These LEDs are default and do not share the pins with other functions.

4.1.3 EMC ((External Memory Interface)

The ERTEC 200P has a memory interface for connecting SDRAM / mobile SDRAM and standard memory blocks (Flash, SRAM).

4.1.4 Host Interface (parallel XHIF port)

An external Host Interface is required for UC1. 16-bit and 32-bit microcontrollers can be connected. Depending on the selection of the Host Interface, the Group GPIO95-32 and its Alternate Peripheral Functions is not available or only in part.

4.1.5 GPIO ports

The ERTEC 200P has a total of 96 GPIO ports on which, among others, Alternate Peripheral Functions are multiplexed. GPIO31-0 is the default and shares the pins with Alternate Peripheral Interfaces and the PHY Debug port.

GPIO95-32 shares the pins with Alternate Peripheral Interfaces, the Host Interface and PN-IP internal state machines. At UC1 GPIOs from this group are available only in part or not at all.

4.1.6 Alternate Peripheral Functions on the GPIO ports

Asynchronous Serial Interface (UART1-4)
The UART1 interface is available completely so that UART1 can also be used as a modem interface. At UART2-4 only the serial Transmit and Receive lines are available at the interface (at UART2 additionally CTS and RTS). All the UARTs can be configured as IO-Link interfaces. UART3 is foreseen as the debug interface because this interface is available at all the use cases. The further UARTs can in part only be used in UC1.

SPI1/2 interface
The SPI1/2 master interfaces are available in all use cases.

ARM926 Watchdog trigger signal
Before the ARM926 watchdog expires, a prewarning can be laid outwards. The output can be enabled in all the use cases.

Clock-Sync: PNPLL signals
The PNPLL contains a clock instance (Clock_A) and 3 application timer blocks. Each application timer block (7 outputs each) controls a separate application cycle and is connected to the Clock_A instance.
In order to couple several bus systems (PROFINET or PROFIBUS) isochronously, the Isochronous Clock PNCLKA_IN can either be fed to the ERTEC 200P or transferred by the ERTEC 200P to a neighboring system (PNCLKA_OUT). via GPIO ports. The required controlling of the corresponding driver can be effected, for example by software via the GPIO function.
The PNPLL signals can be used in all use cases, if appropriate as a subset, because further Alternate Functions can be enabled on these pins.

4.1.7 Boot modes

4 pins of the EMC Interface are available for setting the boot modes. These boot pins are latched off in the SCRB register Boot-Reg during an active PowerOn Reset XRESET (see Chapter 5.3.8). After the reset has been removed, these pins then take over their EMC function in normal operation.

Detailed description about the boot modes is in the documentation ERTEC200 P Datasheet.

4.2 Configuration pins

EMC pins that are latched off during an active PowerOn Reset XRESET in the SCRB register Config-Reg are also provided for setting global use cases or different test modes. After the reset has been removed, these pins then take over their EMC function in normal operation.
Detailed description about the configuration pins is in the documentation ERTEC200 P Datasheet.

4.3 Functional pins

Detailed description about the functions pins is in the documentation ERTEC200 P Datasheet.

4.4 Alternate Functions on the GPIO31..0 and XHIF interface

Detailed description about the alternate functions on the GPIO31..0 and XHIF interface is in the documentation ERTEC200 P Datasheet.

4.5 Setting the Signal Pads (drive strength, pull)

The Signal Pads of the EMC Interface and of the GPIO31..0- and GPIO95..32 blocks can be set with regard to their drive strength and pull properties. The corresponding registers are available in the System Control Register Block SCRB (see Chapter 5.3.8).

The following registers are used to set the **drive strength** of the outputs (see Chapter 5.3.8):

- DRIVE_EMC: Contains all the Signal Pads of the EMC Interface subdivided into signal groups (see registers), settings for 1.8V possible
- DRIVE15_0GPIO: Contains the GPIO15..0 Pads, settings for 3.3V possible
- DRIVE31_15GPIO: Contains the GPIO31..15 Pads, settings for 3.3V possible
- DRIVE47_32GPIO: Contains the GPIO47..32 Pads, settings for 1.8V/3.3V possible
- DRIVE63_48GPIO: Contains the GPIO63..48 Pads, settings for 1.8V/3.3V possible
- DRIVE79_64GPIO: Contains the GPIO79..64 Pads, settings for 1.8V/3.3V possible
- DRIVE95_80GPIO: Contains the GPIO95..80 Pads, settings for 1.8V/3.3V possible

The following combinations can be set per GPIO Pad / signal group (EMC only for 1.8V; GPIO31:0 only for 3.3V; GPIO95:32 for 1.8V / 3.3V). After the PowerOn Reset the drive strength is set by default to:

- 6 mA at the GPIO31..0 Pads with 3.3V
- 9 mA at the GPIO95..32/XHIF Pads with 3.3V
- 6 mA at the GPIO95..32/XHIF Pads with 1.8V
- 12 mA at the EMC Pads with 1.8V

If required, the external host has to increase the drive strength to 9 mA for the XHIF Interface with 1.8V before the first read access is carried out. 6 mA is OK for the XHIF_XRDY Pad because this is a pure point-to-point connection.

Coding	GPIO31..0 (3.3V)	GPIO96..32 / XHIF		EMC (1,8V)
		(3.3V)	(1,8V)	
00b	4mA	6mA	3 mA (not used)	4mA
01b	6 mA (default)	9 mA (default)	6 mA (default)	6mA
10b	8mA	18 mA (not used)	9mA	8mA
11b	12mA	24 mA (not used)	12mA	12 mA (default)

After a reset the registers have specific default values with regard to the drive strength setting. The settings can be changes at any time by the software.

The following registers are used for the **Pull setting** of the inputs (see Chapter 5.3.8):

- PULL15_0GPIO: Contains the GPIO15..0 Pads
- PULL31_15GPIO: Contains the GPIO31..15 Pads
- PULL47_32GPIO: Contains the GPIO47..32 Pads
- PULL63_48GPIO: Contains the GPIO63..48 Pads
- PULL79_64GPIO: Contains the GPIO79..64 Pads
- PULL95_80GPIO: Contains the GPIO95..80 Pads

The following combinations can be set per GPIO Pad:

Coding	Pull
00b	highZ
01b	Pull-Up
10b	highZ
11b	Pull-Down

The Pull setting is only effective when the Pad is set to input. If the Pad operates as an output, the Pull is deactivated. The Pulls have a resistance of 35 – 65 kΩ (typ. 50 kΩ).

During / After a reset the registers have specific default values with regard to the Pull setting. The Reset values are listed in the register description. At GPIO95..0 there is therefore still a dependency on the CONFIG6:3 Pins. Different Pull settings are used during / after a reset for the differing 7 modes. The settings can be changes at any time by the software.
The Pull setting does not change automatically after reconfiguration of the CONFIG6..3 bits in the CONFIG_REG through the software. This has to be changed then by the software.

4.6 Housing type

Detailed description about the housing type is in the documentation ERTEC 200P Datasheet.

4.7 Pin assignment

Detailed description about the pin assignment is in the documentation ERTEC 200P Datasheet.

5 SOFTWARE INTERFACES
5.1 Memory Mapping

5.1.1 Memory Mapping ARM926-I (Instruction AHB interface)

Start and end address	256 MB areas	Function area
0x0000_0000h ... 0x03FF_FFFFh	0	MEM_SWAP parameter assignment: 00b: -> Boot-ROM (0 - 8 KB), imaged 01b: -> EMC-SDRAM (0 - 64 MB) 10b: -> EMC-Asyn-Memory (0 - 64 MB) 11b: -> QVZ Error ARM926 I-TCM (0 – 256 KB / Step 64 KB)
0x0400_0000h ... 0x0FFF_FFFFh		Not used
0x1000_0000h ... 0x1FFF_FFFFh	1	Not used
0x2000_0000h ... 0x2FFF_FFFFh	2	EMC-SDRAM (0 - 256 MB)
0x3000_0000h ... 0x3FFF_FFFFh	3	EMC-Asyn-Memory (Area:Bank 0 - 3) (0 - 256 MB)
0x4000_0000h ... 0x4000_7FFFh	4	Not used
0x4000_8000h ... 0x4000_9FFFh		Boot ROM (8 KB)
0x4000_A000h ... 0x4FFF_FFFFh		Not used
0x5000_0000h ... 0xFFFF_FFFFh	5 -15	Not used

Start and end address	256 MB areas	Function area
0x0000_0000h ... 0x03FF_FFFFh	0	MEM_SWAP parameter assignment: 00b: -> Boot-ROM (0 - 8 KB), imaged 01b: -> EMC-SDRAM (0 - 64 MB) 10b: -> EMC-Asyn-Memory (0 - 64 MB) 11b: -> QVZ Error ARM926 I-TCM (0 – 256 KB / Step 64 KB)
0x0400_0000h ... 0x0FFF_FFFFh		Not used
0x1000_0000h ... 0x1FFF_FFFFh	1	Not used
0x2000_0000h ... 0x2FFF_FFFFh	2	EMC-SDRAM (0 - 256 MB)
0x3000_0000h ... 0x3FFF_FFFFh	3	EMC-Asyn-Memory (Area:Bank 0 - 3) (0 - 256 MB)

0x4000_0000h ... 0x4000_7FFFh	4	Not used
0x4000_8000h ... 0x4000_9FFFh		Boot ROM (8 KB)
0x4000_A000h ... 0x4FFF_FFFFh		Not used
0x5000_0000h ... 0xFFFF_FFFFh	5 -15	Not used

Table 12: Address mapping ARM926-I

5.1.2 Memory Mapping ARM926-D (Data AHB interface)

Start and end address	256 MB areas	Function area
0x0000_0000h ... 0x03FF_FFFFh	0	MEM_SWAP parameter assignment: 00b: -> Boot-ROM (0 - 8 KB), imaged 01b: -> EMC-SDRAM (0 - 64 MB) 10b: -> EMC-Asyn-Memory (0 - 64 MB) 11b: -> QVZ Error
0x0400_0000h ... 0x0401_FFFFh		PN-IP D-TCM (0 - 128 KB / Step 16 KB)
0x0402_0000h ... 0x07FF_FFFFh		Not used
0x0800_0000h ... 0x0803_FFFFh		ARM926 D-TCM (0 – 256 KB / Step 64 KB)
0x0804_0000h ... 0x0FFF_FFFFh		Not used
0x1000_0000h ... 0x10FF_FFFFh	1	ARM-ICU PN-IP PER-IF GDMA register/SRAM EMC register
0x1100_0000h ... 0x1FFF_FFFFh		Not used
0x2000_0000h ... 0x2FFF_FFFFh	2	EMC-SDRAM (0 - 256 MB)
0x3000_0000h ... 0x3FFF_FFFFh	3	EMC-Asyn-Memory (Area:Bank 0 - 3) (0 - 256 MB)
0x4000_0000h ... 0x4FFF_FFFFh	4	APB Peripherals
0x5000_0000h ... 0xFFFF_FFFFh	5 -15	Not used

Table 13: Address mapping ARM926-D

The Boot ROM can be reached from the ARM926-D master, but cannot be used from an application point of view.
After a reset no memory is assigned to the address 0. The first 64 Mbtyes of the EMC-SDRAM or of the EMC Asyn Memory (Chip Select Bank 0) can also be mapped to the address 0 via the MEM_SWAP register in the SCRB (see Chapter 5.3.8).
The ARM926 D-TCM has a size of 0 - 256 kbytes (can be set in 64 kyte steps) and can be inserted by the software in the ARM926 Coprocessor register CP15 c9 at the address 0x0800 0000h. The ARM926 then accesses in its D-TCM under this address range and not on the AHB. Note that the ARM926 D-TCM can only be displayed in the ARM address space in a size of 2^n steps. If, for example, a physical size of D-TCM = 192 kbytes was selected in the 'TCM926 Map Register', the ARM926 can only assign an address space of 256 kbytes to the D-TCM.

Memory Mapping Host Interface

Start and end address	256 MB areas	Function area
0x0000_0000h ... 0x03FF_FFFFh	0	MEM_SWAP parameter assignment: 00b: -> Boot-ROM (0 - 8 KB), imaged 01b: -> EMC-SDRAM (0 - 64 MB) 10b: -> EMC-Asyn-Memory (0 - 64 MB) 11b: -> QVZ Error
0x0400_0000h ... 0x0401_FFFFh		PN-IP D-TCM (0 - 128 KB / Step 16 KB)
0x0402_0000h ... 0x07FF_FFFFh		Not used
0x0800_0000h ... 0x0803_FFFFh		ARM926 D-TCM (0 – 256 KB / Step 64 KB)
0x0804_0000h ... 0x0FFF_FFFFh		Not used
0x1000_0000h ... 0x105F_FFFFh	1	Not used
0x1060_0000h ... 0x10BF_FFFFh		PN-IP PER-IF GDMA register/SRAM
0x10C0_0000h ... 0x10CF_FFFFh		Not used
0x10D0_0000h ... 0x10DF_FFFFh		EMC register
0x10E0_0000h ... 0x1FFF_FFFFh		Not used
0x2000_0000h ... 0x2FFF_FFFFh	2	EMC-SDRAM (0 - 256 MB)
0x3000_0000h ... 0x3FFF_FFFFh	3	EMC-Asyn-Memory (Area:Bank 0 - 3) (0 - 256 MB)
0x4000_0000h ... 0x4FFF_FFFFh	4	APB Peripherals
0x5000_0000h ... 0xFFFF_FFFFh	5 -15	Not used

Table 14: Host Interface address mapping

The primary bootloader has to set up a page on the D-TCM (boot RAM) in the page registers of the HostIF (2x XHIF) for booting from an external host. The remaining pages can subsequently be set up specifically by the external host. It is advisable to lay further pages on the PN-IP, PER-IF, D-TCM, EMC-SDRAM / EMC-SRAM, GMDA and APB Peripherals.

The following table shows the paging register set within the XHIF module. **Name**	**Address** CPU-16bit data width	**Description**	**Name**	**Address** CPU-32bit 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 read/writeable				00h
XHIF_P1_OF_H	16h					00h
XHIF_P2_OF_H	26h					00h
XHIF_P3_OF_H	36h					00h
XHIF_P0_CFG	08h	Config. of the buffering mode for each single page 1: Page is 32-Bit Page 0: Page is 16-Bit Page	XHIF_P0_CFG	08h	Config. of the buffering mode for each single page 1: Buffering mode on read enabled 0 : Buffering mode on read disabled	00h
XHIF_P1_CFG	18h		XHIF_P1_CFG	18h		00h
XHIF_P2_CFG	28h		XHIF_P2_CFG	28h		00h
XHIF_P3_CFG	38h		XHIF_P3_CFG	38h		00h
XHIF_VERSION_L XHIF_VERSION_H	3Ch 3Eh	Version of the XHIF	XHIF_ VERSION	3Ch		

Table 15: Address mapping XHIF for configuring the Page registers

Configuration of the 8 Page registers of the HostIF (2x XHIF with 4 Pages each) is effected, in addition to from the APB, also directly from the outside via the signal XHIF_XCR_R_A20 (default after reset). Selection of the two XHIF modules is controlled by the pin XHIF_SEG_2 (= 0b → XHIF0, = 1b → XHIF1).

5.1.3 Memory Mapping GDMA

Start and end address	256 MB areas	Function area
0x0000_0000h ... 0x03FF_FFFFh	0	MEM_SWAP parameter assignment: 00b: -> Boot-ROM (0 - 8 KB), imaged 01b: -> EMC-SDRAM (0 - 64 MB) 10b: -> EMC-Asyn-Memory (0 - 64 MB) 11b: -> QVZ Error
0x0400_0000h ... 0x0401_FFFFh		Not used
0x0402_0000h ... 0x07FF_FFFFh		Not used
0x0800_0000h ... 0x0803_FFFFh		ARM926 D-TCM (0 – 256 KB / Step 64 KB)
0x0804_0000h ... 0x0FFF_FFFFh		Not used
0x1000_0000h ... 0x107F_FFFFh	1	Not used
0x1080_0000h ... 0x10AF_FFFFh		PER-IF GDMA register/SRAM
0x10B0_0000h ... 0x10BF_FFFFh		Not used
0x10C0_0000h ... 0x10FF_FFFFh		EMC register
0x1100_0000h ... 0x1FFF_FFFFh		Not used
0x2000_0000h ... 0x2FFF_FFFFh	2	EMC-SDRAM (0 - 256 MB)
0x3000_0000h ... 0x3FFF_FFFFh	3	EMC-Asyn-Memory (Area:Bank 0 - 3) (0 - 256 MB)
0x4000_0000h ... 0x4FFF_FFFFh	4	APB Peripherals
0x5000_0000h ... 0xFFFF_FFFFh	5 -15	Not used

Table 16: Address mapping GDMA

The Boot ROM can be reached from the GDMA, but cannot be used from an application point of view.

5.2 Detailled Address Mapping

Seg.	Contents	Size	Range	Description
0	Boot-ROM (8 KB) or EMC-SDRAM (64 MB) or EMC Asyn Memory(64 MB) or ARM926 I-TCM (256 KB)	64 MB 256 KB	0000_0000 03FF_FFFF 0000_0000 0003_FFFF	After reset: Boot-ROM (8 KB physical; 2^{13} * imaged, Memory-Swap=00b); After Memory-Swap: EMC-SDRAM (64 MB physical; 2^{0} * imaged, Memory-Swap=01b); or EMC Asyn Memory (64 MB physical; not imaged, Memory-Swap=10b); or After activation ARM926 I-TCM (in CP15 c9 register): ARM 926 I-TCM (0 – 256 KB physical; not imaged);
	Not used	64 MB	0000_0000 03FF_FFFF	
	Not used	128 KB	0400_0000 0401_FFFF	
	Not used	64 MB – 128 KB	0402_0000 07FF_FFFF	
	ARM926 D-TCM (256 KB)	256 KB	0800_0000 0803_FFFF	ARM 926 D-TCM (0 – 256 KB physical; not imaged);
	Not used	~128 MB	0804_0000 0FFF_FFFF	
1	ARM-ICU	6 MB	1000_0000 105F_FFFF	ARM926 Interrupt Controller; 5 MB physical; 2^{0} * imaged;
	PN-IP	2 MB	1060_0000 107F_FFFF	2 MB physical; 2^{0} * imaged;
	PER_IF (consistency buffer)	2 MB	1080_0000 109F_FFFF	64 KB physical; 2^{5} * imaged
	GDMA register/SRAM	1 MB	10A0_0000 10AF_FFFF	GDMA Register and internal GDMA-Job-SRAM (size: 4608 bytes) 1 MB physical; 2^{0} * imaged; Register : 10A0_0000..10A0_00AF Job-SRAM : 10A0_00B0..10A0_12AF Not-used : 10A0_12B0..10AF_FFFF
	Reserved	1 MB	10B0_0000 10BF_FFFF	Reserved; 64 bytes physical; 2^{14} * imaged;
	Reserved	1 MB	10C0_0000 10CF_FFFF	Reserved 1 MB physical; 2^{0} * imaged;
	EMC register	1 MB	10D0_0000 10DF_FFFF	EMC register; 1 MB physical; not imaged;
	Reserved	1 MB	10E0_0000 10EF_FFFF	Reserved; 8 KB physical; 2^{7} * imaged;
	Reserved	1 MB	10F0_0000 10FF_FFFF	Reserved; 512 bytes physical; 2^{11} * imaged;
	Not used	240 MB	1100_0000 1FFF_FFFF	

2	EMC-SDRAM	256 MB	2000_0000 2FFF_FFFF	256 MB physical; 2^0 * imaged;
3	EMC Asyn Memory/Peripheral Chip Select Bank0	64 MB	3000 0000 33FF FFFF	64 MB physical; 2^0 * imaged; (if less is selected, the range is mirrored)
	EMC Asyn Memory/Peripheral Chip Select Bank1	64 MB	3400 0000 37FF FFFF	64 MB physical; 2^0 * imaged; (if less is selected, the range is mirrored)
	EMC Asyn Memory/Peripheral Chip Select Bank2	64 MB	3800 0000 3BFF FFFF	64 MB physical; 2^0 * imaged; (if less is selected, the range is mirrored)
	EMC Asyn Memory/Peripheral Chip Select Bank3	64 MB	3C00 0000 3FFF FFFF	64 MB physical; 2^0 * imaged; (if less is selected, the range is mirrored)
4 APB-Per.	PER_IF (register)	32 KB	4000_0000 4000_7FFF	32 KB physical;
	Internal Boot-ROM	8 KB	4000_8000 4000_9FFF	8 KB physical;
	UART1	4 KB	4000_A000 4000_AFFF	4 KB physical;
	UART2	4 KB	4000_B000 4000_BFFF	4 KB physical;
	UART3	4 KB	4000_C000 4000_CFFF	4 KB physical;
	UART4	4 KB	4000_D000 4000_DFFF	4 KB physical;
	I^2C_3	4 KB	4000_E000 4000_EFFF	256 bytes physical; imaged
	System Control Register Block	4 KB	4000_F000 4000_FFFF	256 bytes physical; imaged
	SPI1	256 B	4001_0000 4001_00FF	256 bytes physical;
	Reserved	256 B	4001_0100 4001_01FF	256 bytes physical;
	Reserved	256 B	4001_0200 4001_02FF	256 bytes physical;
	Reserved	256 B	4001_0300 4001_03FF	256 bytes physical;
	Reserved	256 B	4001_0400 4001_04FF	256 bytes physical;
	Reserved	2,75 KB	4001_0500 4001_0FFF	256 bytes physical; 11*imaged
	SPI2	256 B	4001_1000 4001_10FF	256 bytes physical;
	Reserved	256 B	4001_1100 4001_11FF	256 bytes physical;

	Reserved	256 B	4001_1200 4001_12FF	256 bytes physical;
	Reserved	256 B	4001_1300 4001_13FF	256 bytes physical;
	Reserved	256 B	4001_1400 4001_14FF	256 bytes physical;
	Reserved	2,75 KB	4001_1500 4001_1FFF	256 bytes physical; 11*imaged
	Timer 0-5	4 KB	4001_2000 4001_2FFF	256 bytes physical; imaged
	Watchdog	4 KB	4001_3000 4001_3FFF	32 bytes physical; imaged
	F-counter	4 KB	4001_4000 4001_4FFF	8 bytes physical; imaged
	Reserved	4 KB	4001_5000 4001_5FFF	32 bytes physical; imaged
	Reserved	4 KB	4001_6000 4001_6FFF	256 bytes physical; imaged
	Reserved	4 KB	4001_7000 4001_7FFF	128 bytes physical; imaged
	GPIO	4 KB	4001_8000 4001_8FFF	256 bytes physical; imaged
	I-Filter	4 KB	4001_9000 4001_9FFF	128 bytes physical; imaged
	Host Interface	4 KB	4001_A000 4001_AFFF	256 bytes physical; imaged
	Reserved	4 KB	4001_B000 4001_BFFF	256 bytes physical; imaged
	Reserved	4 KB	4001_C000 4001_CFFF	4 KB physical;
	Not used	12 KB	4001_D000 4FFF_FFFF	
5 - 15	Not used	2816 MB	5000_0000 FFFF_FFFF	

Table 17: Detailed address mapping

5.3 Register-description

5.3.1 General information

All the bits identified with "reserved" in the following register descriptions should have the value 0 assigned when writing in view of future extensions.
These bits may not be evaluated during reading.
If no special information is given, the registers may be changed during operation.

5.3.2 Peripheral Interface

5.3.2.1 PERIF_AHB_APP

Base address see Chapter 5.2.

Address space:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	DFFFh	perif_app	
800h	DFFFh	PERIF_IO_RAM	

Module	Register/Memory	Read	Write	Address
/perif_app				
	Com- mand_IF_Control_APP	r(h)	(w)	0h
	Command_IF_Status_APP	r(h)		4h
	Host_IRQ_low	rh		8h
	Host_IRQ_high	rh		Ch
	Host_IRQmask_low	r	w	10h
	Host_IRQmask_high	r	w	14h
	Host_Event_low	rh	w	18h
	Host_Event_high	rh	w	1Ch
	Host_IRQack_low	rh	w	20h
	Host_IRQack_high	rh	w	24h
	Host_EOI	r	w	28h
	PN_IRQ_low	rh		2Ch
	PN_IRQ_high	rh		30h
	PN_IRQmask_low	r	w	34h
	PN_IRQmask_high	r	w	38h
	PN_Event_low	rh	w	3Ch
	PN_Event_high	rh	w	40h
	PN_IRQack_low	rh	w	44h
	PN_IRQack_high	rh	w	48h
	PN_EOI	r	w	4Ch
	PERIF_IO_RAM	r	w	800h - DFFFh

Register description:

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

Module: /perif_app

Register:	Command_IF_Control_APP				Address:	0h	
Bits:	15dt0	Reset value:		0000h	Attributes:	r(h)	(w)
Description:	Command Interface Control						
Bit	Identifier	Reset	Attr.		Function / Description		
4dt0	CR_Number	00h	rh	w	CR Number		
5dt5	ConfRequest	0h	rh	w	Command Confirmation Request		
7dt6	F_Code	0h	rh	w	Consistence Commando		
9dt8	<reserved>	0000h	r		reserved		
12dt10	<reserved>	0000h	r		reserved		
15dt13	User_ID	0h	rh	w	User_ID		
31dt16	<reserved>	0000h	r		reserved		

Register:	Command_IF_Status_APP				Address:	4h	
Bits:	15dt0	Reset value:		0000h	Attributes:	r(h)	
Description:	Command Interface Status						
Bit	Identifier	Reset	Attr.		Function / Description		
4dt0	CR_Number	00h	rh		CR Number		
5dt5	ConfResponse	0h	rh		ConfResponse		
7dt6	F_Code	0h	rh		Consistence Commando		
9dt8	<reserved>	0h	r		reserved		
10dt10	ConfValue	0h	rh		Command Confirmation Value		
12dt11	<reserved>	0h	r		reserved		
15dt13	User_ID	0h	rh		User_ID		

Register:	Host_IRQ_low				Address:	8h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	rh	
Description:	Host_IRQ_low						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	IRQ_Bits	00000000h	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:		00000000h	Attributes:	rh	
Description:	Host_IRQ_high						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	IRQ_Bits	00000000h	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:	Host_IRQmask_low						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Mask_Bits	FFFFFFFFh	r	w	Mask_Bits '0' =the event is entered in PN_IRQ_low		

					'1' =the event is not entered in PN_IRQ_low

Register:	Host_IRQmask_high				Address:	14h	
Bits:	31dt0	Reset value:		FFFFFFFFh	Attributes:	r	w
Description:	Host_IRQmask_high						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Mask_Bits	FFFFFFFFh	r	w	Mask_Bits '0' =the event is entered in PN_IRQ_low '1' =the event is not entered in PN_IRQ_low		

Register:	Host_Event_low				Address:	18h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	rh	w
Description:	Host_Event_low						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Event_Bits	00000000h	rh	w	Mask_Bits '0' =the event is entered in PN_IRQ_low '1' =the event is not entered in PN_IRQ_low		

Register:	Host_Event_high				Address:	1Ch	
Bits:	31dt0	Reset value:		00000000h	Attributes:	rh	w
Description:	Host_Event_high						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Event_Bits	00000000h	rh	w	Mask_Bits '0' =the event bit is not reset in PN_Event_low '1' =the event bit is reset in PN_Event_low		

Register:	Host_IRQack_low				Address:	20h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	rh	w
Description:	Host_IRQack_low						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Ack_Bits	00000000h	rh	w	Ack_Bits write '0' =the event bit is not reset in PN_Event_low '1' =the event bit is reset in PN_Event_low		

Register:	Host_IRQack_high				Address:	24h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	rh	w
Description:	Host_IRQack_high						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Ack_Bits	00000000h	rh	w	Ack_Bits write 0' =the event bit is not reset in PN_Event_low '1' =the event bit is reset in PN_Event_low		

Register:	Host_EOI				Address:	28h	
Bits:	17dt0	Reset value:		00000h	Attributes:	r	w
Description:	Host_EOI						
Bit	Identifier	Reset	Attr.		Function / Description		
17dt0	wait_time	00000h	r	w	wait_time		

Register:	PN_IRQ_low				Address:	2Ch	
Bits:	31dt0	Reset value:		00000000h	Attributes:	rh	
Description:	PN_IRQ_low						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	IRQ_Bits	00000000h	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:		00000000h	Attributes:	rh	
Description:	PN_IRQ_high						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	IRQ_Bits	00000000h	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:	PN_IRQmask_low						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Mask_Bits	FFFFFFFFh	r	w	Mask_Bits '0' =the event bit is not reset in PN_Event_low '1' =the event bit is reset in PN_Event_low		

Register:	PN_IRQmask_high				Address:	38h	
Bits:	31dt0	Reset value:		FFFFFFFFh	Attributes:	r	w
Description:	PN_IRQmask_high						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Mask_Bits	FFFFFFFFh	r	w	Mask_Bits 0' =the event bit is not reset in PN_Event_low '1' =the event bit is reset in PN_Event_low		

Register:	PN_Event_low				Address:	3Ch	
Bits:	31dt0	Reset value:		00000000h	Attributes:	rh	w
Description:	PN_Event_low						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Event_Bits	00000000h	rh	w	Event_Bits_low		

Register:	PN_Event_high				Address:	40h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	rh	w
Description:	PN_Event_high						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Event_Bits	00000000h	rh	w	Event_Bits high		

Register:	PN_IRQack_low				Address:	44h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	rh	w
Description:	PN_IRQack_low						
Bit	Identifier	Reset	Attr.		Function / Description		

31dt0	Ack_Bits	00000000h	rh	w	Ack_Bits write 0' =the event bit is not reset in PN_Event_low '1' =the event bit is reset in PN_Event_low

Register:	PN_IRQack_high				Address:	48h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	rh	w
Description:	PN_IRQack_high						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	Ack_Bits	00000000h	rh	w	Ack_Bits write 0' =the event bit is not reset in PN_Event_low '1' =the event bit is reset in PN_Event_low		

Register:	PN_EOI				Address:	4Ch	
Bits:	17dt0	Reset value:		00000h	Attributes:	r	w
Description:	PN_EOI						
Bit	Identifier	Reset	Attr.		Function / Description		
17dt0	wait_time	00000h	r	w	wait_time		

5.3.2.2 PERIF_APB

Base address see Chapter 5.2.

Address space:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	4FFFh	perif_apb	
2000h	4FFFh	PERIF_IO_RAM	

Module	Register/Memory	Read	Write	Address
/perif_apb				
	CR_Address_1	r(h)	(w)	0h
	CR_Address_2	r(h)	(w)	4h
	CR_Address_3	r(h)	(w)	8h
	CR_Address_4	r(h)	(w)	Ch
	CR_Address_5	r(h)	(w)	10h
	CR_Address_6	r(h)	(w)	14h
	CR_Address_7	r(h)	(w)	18h
	CR_Address_8	r(h)	(w)	1Ch
	CR_Address_9	r(h)	(w)	20h
	CR_Address_10	r(h)	(w)	24h
	CR_Address_11	r(h)	(w)	28h
	CR_Address_12	r(h)	(w)	2Ch
	CR_Address_13	r(h)	(w)	30h
	CR_Address_14	r(h)	(w)	34h
	CR_Address_15	r(h)	(w)	38h
	CR_Address_16	r(h)	(w)	3Ch
	CR_Address_17	r(h)	(w)	40h

	CR_Address_18	r(h)	(w)	44h
	CR_Address_19	r(h)	(w)	48h
	CR_Address_20	r(h)	(w)	4Ch
	CR_Address_21	r(h)	(w)	50h
	CR_Address_22	r(h)	(w)	54h
	CR_Address_23	r(h)	(w)	58h
	CR_Address_24	r(h)	(w)	5Ch
	CR_Address_25	r(h)	(w)	60h
	CR_Address_26	r(h)	(w)	64h
	CR_Address_27	r(h)	(w)	68h
	CR_State_1	r(h)	(w)	100h
	CR_State_2	r(h)	(w)	104h
	CR_State_3	r(h)	(w)	108h
	CR_State_4	r(h)	(w)	10Ch
	CR_State_5	r(h)	(w)	110h
	CR_State_6	r(h)	(w)	114h
	CR_State_7	r(h)	(w)	118h
	CR_State_8	r(h)	(w)	11Ch
	CR_State_9	r(h)	(w)	120h
	CR_State_10	r(h)	(w)	124h
	CR_State_11	r(h)	(w)	128h
	CR_State_12	r(h)	(w)	12Ch
	CR_State_13	r(h)	(w)	130h
	CR_State_14	r(h)	(w)	134h
	CR_State_15	r(h)	(w)	138h
	CR_State_16	r(h)	(w)	13Ch
	CR_State_17	r(h)	(w)	140h
	CR_State_18	r(h)	(w)	144h
	CR_State_19	r(h)	(w)	148h
	CR_State_20	r(h)	(w)	14Ch
	CR_State_21	r(h)	(w)	150h
	CR_State_22	r(h)	(w)	154h
	CR_State_23	r(h)	(w)	158h
	CR_State_24	r(h)	(w)	15Ch
	CR_State_25	r(h)	(w)	160h
	CR_State_26	r(h)	(w)	164h
	CR_State_27	r(h)	(w)	168h
	reserved	-	-	200h
	reserved	-	-	204h
	reserved	-	-	208h
	reserved	-	-	20Ch
	reserved	-	-	210h
	reserved	-	-	214h
	reserved	-	-	218h
	reserved	-	-	21Ch
	reserved	-	-	220h
	reserved	-	-	224h
	reserved	-	-	228h
	reserved	-	-	22Ch
	reserved	-	-	230h
	reserved	-	-	234h

	reserved	-	-	238h
	reserved	-	-	23Ch
	reserved	-	-	240h
	reserved	-	-	244h
	reserved	-	-	248h
	reserved	-	-	24Ch
	reserved	-	-	250h
	reserved	-	-	254h
	reserved	-	-	258h
	reserved	-	-	25Ch
	reserved	-	-	260h
	reserved	-	-	264h
	reserved	-	-	268h
	reserved	-	-	300h
	reserved	-	-	524h
	reserved	-	-	528h
	reserved	-	-	52Ch
	reserved	-	-	530h
	reserved	-	-	534h
	reserved	-	-	600h
	reserved	-	-	800h
	reserved	-	-	804h
	reserved	-	-	808h
	Burst_Config	r	(w)	80Ch
	PERIF_IO_RAM	r	w	2000h - 4FFFh

Register description:

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

Module: /perif_apb

Register:	CR_Address_1				Address:	0h	
Bits:	31dt0	Reset value:		00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_1						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	<reserved>	0h	r		-		
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
15dt12	<reserved>	0h	r		-		
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
26dt26	<reserved>	0h	r		-		
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get:		

					‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_2			Address:	4h	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_2					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_3			Address:	8h	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_3					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)

31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_4			Address:	Ch	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_4					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_5			Address:	10h	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_5					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger

					0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_6				Address:	14h	
Bits:	31dt0	Reset value:		00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_6						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	<reserved>	0h	r		-		
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
15dt12	<reserved>	0h	r		-		
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
26dt26	<reserved>	0h	r		-		
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)		
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger		

Register:	CR_Address_7				Address:	18h	
Bits:	31dt0	Reset value:		00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_7						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	<reserved>	0h	r		-		
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
15dt12	<reserved>	0h	r		-		
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
26dt26	<reserved>	0h	r		-		
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)		
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger		

Register:	CR_Address_8			Address:	1Ch	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_8					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_9			Address:	20h	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_9					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_10			Address:	24h	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)

Description:	CR_Address_10				
Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_11			Address:	28h	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)

Description:	CR_Address_11				
Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_12			Address:	2Ch	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)

Description:	CR_Address_12			
Bit	Identifier	Reset	Attr.	Function / Description

1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_13				Address:	30h	
Bits:	31dt0	Reset value:		00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_13						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	<reserved>	0h	r		-		
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
15dt12	<reserved>	0h	r		-		
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
26dt26	<reserved>	0h	r		-		
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)		
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger		

Register:	CR_Address_14				Address:	34h	
Bits:	31dt0	Reset value:		00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_14						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	<reserved>	0h	r		-		
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte ad-		

					dress space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_15				Address:	38h	
Bits:	31dt0	Reset value:		00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_15						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	<reserved>	0h	r		-		
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
15dt12	<reserved>	0h	r		-		
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
26dt26	<reserved>	0h	r		-		
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)		
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger		

Register:	CR_Address_16				Address:	3Ch	
Bits:	31dt0	Reset value:		00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_16						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	<reserved>	0h	r		-		
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		

15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_17				Address:	40h	
Bits:	31dt0	Reset value:		00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_17						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	<reserved>	0h	r		-		
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
15dt12	<reserved>	0h	r		-		
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
26dt26	<reserved>	0h	r		-		
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)		
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger		

Register:	CR_Address_18				Address:	44h	
Bits:	31dt0	Reset value:		00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_18						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	<reserved>	0h	r		-		
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
15dt12	<reserved>	0h	r		-		
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte ad-		

					dress space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_19			Address:	48h	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_19					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_20			Address:	4Ch	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_20					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant

26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_21			Address:	50h	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_21					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_22			Address:	54h	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_22					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM

					‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_23			Address:	58h	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_23					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_24			Address:	5Ch	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_24					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if

					an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_25			Address:	60h	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_25					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_26			Address:	64h	
Bits:	31dt0	Reset value:	00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_26					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	<reserved>	0h	r		-
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
15dt12	<reserved>	0h	r		-
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant
26dt26	<reserved>	0h	r		-
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger

Register:	CR_Address_27				Address:	68h	
Bits:	31dt0	Reset value:		00000FFCh	Attributes:	r(h)	(w)
Description:	CR_Address_27						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	<reserved>	0h	r		-		
11dt2	CR_StartAddress	3FFh	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
15dt12	<reserved>	0h	r		-		
25dt16	CR_EndAddress	000h	r	w	Address_Mode = DirectMode: Start address of the CR (4-Byte aligned, 4KByte address space) Address_Mode = PageMode not relevant		
26dt26	<reserved>	0h	r		-		
27dt27	Zero_Data	0h	rh	w	After the buffer exchange the application will get: ‚0': the date contained in the of the IO-RAM ‚1': zero data (0x00..00)		
31dt28	New_Data_INT	0h	r	w	With the parameter New_Data_INT the application decides if an interrupt shall be issued if new output data are available: 0x0, 0xA-0xF: no New_Data_INT trigger 0x1 – 0x9: New_Data_INT trigger		

Register:	CR_State_1				Address:	100h	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_1						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states		
9dt8	buffer_number_next_buffer	3h	rh	w	internal IO buffer states		
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states		
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states		
14dt14	<reserved>	0h	r		-		
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0		
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)		

Register:	CR_State_2				Address:	104h	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_2						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		

7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_3				Address:	108h	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_3						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states		
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states		
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states		
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states		
14dt14	<reserved>	0h	r		-		
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0		
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)		

Register:	CR_State_4				Address:	10Ch	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_4						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states		
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states		
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states		
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states		
14dt14	<reserved>	0h	r		-		
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0		
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)		

Register:	CR_State_5				Address:	110h	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_5						
Bit	Identifier	Reset	Attr.		Function / Description		

1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_6				Address:	114h	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_6						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states		
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states		
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states		
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states		
14dt14	<reserved>	0h	r		-		
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0		
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)		

Register:	CR_State_7				Address:	118h	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_7						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states		
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states		
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states		
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states		
14dt14	<reserved>	0h	r		-		
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0		
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)		

Register:	CR_State_8	Address:	11Ch

Bits:	31dt0	Reset value:	00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_8					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_9			Address:	120h	
Bits:	31dt0	Reset value:	00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_9					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_10			Address:	124h	
Bits:	31dt0	Reset value:	00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_10					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0

31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_11			Address:	128h	
Bits:	31dt0	Reset value:	00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_11					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf-fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_12			Address:	12Ch	
Bits:	31dt0	Reset value:	00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_12					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf-fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_13			Address:	130h	
Bits:	31dt0	Reset value:	00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_13					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf-fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states

13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_14				Address:	134h	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_14						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states		
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states		
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states		
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states		
14dt14	<reserved>	0h	r		-		
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0		
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)		

Register:	CR_State_15				Address:	138h	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_15						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states		
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states		
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states		
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states		
14dt14	<reserved>	0h	r		-		
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0		
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)		

Register:	CR_State_16				Address:	13Ch	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_16						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		

7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_17				Address:	140h	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_17						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states		
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states		
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states		
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states		
14dt14	<reserved>	0h	r		-		
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0		
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)		

Register:	CR_State_18				Address:	144h	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_18						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states		
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states		
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states		
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states		
14dt14	<reserved>	0h	r		-		
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0		
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)		

Register:	CR_State_19				Address:	148h	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_19						
Bit	Identifier	Reset	Attr.		Function / Description		

1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_20				Address:	14Ch	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_20						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states		
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states		
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states		
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states		
14dt14	<reserved>	0h	r		-		
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0		
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)		

Register:	CR_State_21				Address:	150h	
Bits:	31dt0	Reset value:		00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_21						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states		
3dt2	<reserved>	0h	r		-		
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states		
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states		
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states		
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states		
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states		
14dt14	<reserved>	0h	r		-		
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0		
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)		

Register:	CR_State_22	Address:	154h

Bits:	31dt0	Reset value:	00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_22					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_23			Address:	158h	
Bits:	31dt0	Reset value:	00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_23					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_24			Address:	15Ch	
Bits:	31dt0	Reset value:	00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_24					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf- fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0

31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_25			Address:	160h	
Bits:	31dt0	Reset value:	00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_25					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf-fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_26			Address:	164h	
Bits:	31dt0	Reset value:	00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_26					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf-fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states
13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	CR_State_27			Address:	168h	
Bits:	31dt0	Reset value:	00000FD2h	Attributes:	r(h)	(w)
Description:	CR_State_27					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	buffer_number_data_buffer	2h	rh	w	internal IO buffer states
3dt2	<reserved>	0h	r		-
5dt4	buffer_number_free_buffer	1h	rh	w	internal IO buffer states
7dt6	buffer_number_f2_buffer	3h	rh	w	internal IO buffer states
9dt8	buf-fer_number_next_buffer	3h	rh	w	internal IO buffer states
11dt10	buffer_number_n2_buffer	3h	rh	w	internal IO buffer states

13dt12	buffer_number_user_buffer	0h	rh	w	internal IO buffer states
14dt14	<reserved>	0h	r		-
15dt15	EXT_BUF_Mode	0h	r	w	Shall always be set to 0b0
31dt16	Mapping	0000h	r	w	internal IO buffer mapping (will be set by the PROFINET stack of the Evaluation Kit ERTEC 200P)

Register:	Burst_Config			Address:	80Ch	
Bits:	31dt0	Reset value:	00000101h	Attributes:	r	(w)
Description:	Burst_Config					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	BurstMode_comAHB	1h	r	w	At the communication AHB-slave interface of the PerIF the following modes are supported: ▪ "x0": an AHB read burst access will be accomplished as a single read at the SC-bus ▪ "01": all AHB read burst accesses are supported except an INCR access will be accomplished as a single read at the SC-bus ▪ "11": all AHB read burst accesses (INCR, INCR4, ... , WRAP4, ... WRAP16) are supported at the SC-bus
7dt2	<reserved>	000000h	r		
9dt8	BurstMode_applAHB	1h	r	w	At the application AHB-slave interface of the PerIF the following modes are supported: ▪ "x0": an AHB read burst access will be accomplished as a single read at the SC-bus ▪ "01": all AHB read burst accesses are supported except an INCR access will be accomplished as a single read at the SC-bus ▪ "11": all AHB read burst accesses (INCR, INCR4, ... , WRAP4, ... WRAP16) are supported at the SC-bus
31dt10	<reserved>	000000h	r		

5.3.3 Host Interface

Base address for accesses in ERTEC 200P see Chapter 5.2.

Address space:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	FCh	HOSTIF	<interface>

Module	Register/Memory	Read	Write	Address	XHIF_XCS_R Address
/HOSTIF					
	HOST_CONTROL	r	w	0h	
	IP_VERSION	r		40h	
	IP_DEVELOPMENT	r		44h	
	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

Registerbeschreibung:

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

Module: /HOSTIF

Register:	HOST_CONTROL	Address:	0h

Bits:	0dt0	Reset value:	0h	Attributes:	r	w
Description:	Selection of the serial or parallel path					

Bit	Identifier	Reset	Attr.		Function / Description
0	CONNECT_MODE	0h	r	w	0 =parallel connection (XHIF)

Register:	IP_VERSION			Address:	40h	
Bits:	31dt0	Reset value:	100h	Attributes:	r	
Description:	Metal-fix register for IP Version					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	DEBUG_VERSION	00h	r		IP Debug Version
15dt8	VERSION	01h	r		IP Version
31dt16	IP_CONFIGURATION	0000h	r		IP Configuration

Register:	IP_DEVELOPMENT			Address:	44h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	Metal-fix register for IP Development					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	LABEL_STRUCTURE	00000000h	r		Design Label

30	APB_RD_WR	0h	r		'0': RD access '1': WR access
31	ERROR_LOCK	0h	r	w	Is set by the HW to '1' when an erroneous APB access is recognized. This blocks further HW entries. The SW has to reset the bit to '0' to enable new entries.

Register:	XHIF_CONTROL			Address:	70h	
Bits:	3dt0	Reset value:	0h	Attributes:	rh	w
Description:	XHIF Interface Settings After a reset and when writing the CONFIG_REG register of the SCRB these are adopted from there					

Bit	Identifier	Reset	Attr.		Function / Description
0	XHIF_ACC_MODE	0h	rh	w	XHIF Handshake protocol: 0 =Intel Mode, 1 =Motorola Mode
1	XHIF_POL_RDY	0h	rh	w	0': XHIF_XRDY is low_active ‘'1': XHIF_XRDY is high_active
3dt2	XHIF_CPU_WIDTH	0h	rh	w	Data bus width: 00 =8b *, 01 =16b, 10 =32b, 11 =32b * 00 =8b is, however, not supported by the XHIF interface!

Register:	XHIF_0_P0_RG			Address:	80h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	Range value of the page #0					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	XHIF_0_P0_RG_ROSL2	00h	r		Read only value = 0
21dt8	XHIF_0_P0_RG_RW	0000h	r	w	Read/Write part of the Range register

31dt22	XHIF_0_P0_RG_ROSL1	000h	r		Read only value = 0

Register:	XHIF_0_P0_OF				Address:	84h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Offset value of the page #0						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_0_P0_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_0_P0_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:	XHIF_0_P0_CFG				Address:	88h	
Bits:	0dt0	Reset value:		0h	Attributes:	r	w
Description:	Configuration of the buffering mode for page #0						
Bit	Identifier	Reset	Attr.		Function / Description		
0	XHIF_0_P0_CFG_BUFMOD	0h	r	w	16bit access mode of page #0		

Register:	XHIF_0_P1_RG				Address:	90h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Range value of the page #1						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_0_P1_RG_ROSL2	00h	r		Read only value = 0		
21dt8	XHIF_0_P1_RG_RW	0000h	r	w	Read/Write part of the Range register		
31dt22	XHIF_0_P1_RG_ROSL1	000h	r		Read only value = 0		

Register:	XHIF_0_P1_OF				Address:	94h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Offset value of the page #1						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_0_P1_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_0_P1_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:	XHIF_0_P1_CFG				Address:	98h	
Bits:	0dt0	Reset value:		0h	Attributes:	r	w
Description:	Configuration of the buffering mode for page #1						
Bit	Identifier	Reset	Attr.		Function / Description		
0	XHIF_0_P1_CFG_BUFMOD	0h	r	w	16bit access mode of page #1		

Register:	XHIF_0_P2_RG				Address:	A0h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Range value of the page #2						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_0_P2_RG_ROSL2	00h	r		Read only value = 0		
21dt8	XHIF_0_P2_RG_RW	0000h	r	w	Read/Write part of the Range register		
31dt22	XHIF_0_P2_RG_ROSL1	000h	r		Read only value = 0		

Register:	XHIF_0_P2_OF				Address:	A4h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Offset value of the page #2						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_0_P2_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_0_P2_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:	XHIF_0_P2_CFG				Address:	A8h	
Bits:	0dt0	Reset value:		0h	Attributes:	r	w
Description:	Configuration of the buffering mode for page #2						
Bit	Identifier	Reset	Attr.		Function / Description		
0	XHIF_0_P2_CFG_BUFMOD	0h	r	w	16bit access mode of page #2		

Register:	XHIF_0_P3_RG				Address:	B0h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Range value of the page #3						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_0_P3_RG_ROSL2	00h	r		Read only value = 0		
21dt8	XHIF_0_P3_RG_RW	0000h	r	w	Read/Write part of the Range register		
31dt22	XHIF_0_P3_RG_ROSL1	000h	r		Read only value = 0		

Register:	XHIF_0_P3_OF				Address:	B4h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Offset value of the page #3						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_0_P3_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_0_P3_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:	XHIF_0_P3_CFG				Address:	B8h	
Bits:	0dt0	Reset value:		0h	Attributes:	r	w
Description:	Configuration of the buffering mode for page #3						
Bit	Identifier	Reset	Attr.		Function / Description		
0	XHIF_0_P3_CFG_BUFMOD	0h	r	w	16bit access mode of page #3		

Register:	XHIF_0_VERSION				Address:	BCh	
Bits:	31dt0	Reset value:		04000804h	Attributes:	r	
Description:	Metal-fix register for XHIF Version						

Register:	XHIF_1_P0_RG				Address:	C0h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Range value of the page #0						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_1_P0_RG_ROSL2	00h	r		Read only value = 0		
21dt8	XHIF_1_P0_RG_RW	0000h	r	w	Read/Write part of the Range register		

31dt22	XHIF_1_P0_RG_ROSL1	000h	r		Read only value = 0

Register:	XHIF_1_P0_OF				Address:	C4h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Offset value of the page #0						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_1_P0_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_1_P0_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:	XHIF_1_P0_CFG				Address:	C8h	
Bits:	0dt0	Reset value:		0h	Attributes:	r	w
Description:	Configuration of the buffering mode for page #0						
Bit	Identifier	Reset	Attr.		Function / Description		
0	XHIF_1_P0_CFG_BUFMOD	0h	r	w	16bit access mode of page #0		

Register:	XHIF_1_P1_RG				Address:	D0h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Range value of the page #1						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_1_P1_RG_ROSL2	00h	r		Read only value = 0		
21dt8	XHIF_1_P1_RG_RW	0000h	r	w	Read/Write part of the Range register		
31dt22	XHIF_1_P1_RG_ROSL1	000h	r		Read only value = 0		

Register:	XHIF_1_P1_OF				Address:	D4h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Offset value of the page #1						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_1_P1_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_1_P1_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:	XHIF_1_P1_CFG				Address:	D8h	
Bits:	0dt0	Reset value:		0h	Attributes:	r	w
Description:	Configuration of the buffering mode for page #1						
Bit	Identifier	Reset	Attr.		Function / Description		
0	XHIF_1_P1_CFG_BUFMOD	0h	r	w	16bit access mode of page #1		

Register:	XHIF_1_P2_RG				Address:	E0h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Range value of the page #2						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_1_P2_RG_ROSL2	00h	r		Read only value = 0		
21dt8	XHIF_1_P2_RG_RW	0000h	r	w	Read/Write part of the Range register		
31dt22	XHIF_1_P2_RG_ROSL1	000h	r		Read only value = 0		

Register:	XHIF_1_P2_OF				Address:	E4h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Offset value of the page #2						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_1_P2_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_1_P2_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:	XHIF_1_P2_CFG				Address:	E8h	
Bits:	0dt0	Reset value:		0h	Attributes:	r	w
Description:	Configuration of the buffering mode for page #2						
Bit	Identifier	Reset	Attr.		Function / Description		
0	XHIF_1_P2_CFG_BUFMOD	0h	r	w	16bit access mode of page #2		

Register:	XHIF_1_P3_RG				Address:	F0h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Range value of the page #3						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_1_P3_RG_ROSL2	00h	r		Read only value = 0		
21dt8	XHIF_1_P3_RG_RW	0000h	r	w	Read/Write part of the Range register		
31dt22	XHIF_1_P3_RG_ROSL1	000h	r		Read only value = 0		

Register:	XHIF_1_P3_OF				Address:	F4h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Offset value of the page #3						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	XHIF_1_P3_OF_RO	00h	r		Read only value = 0		
31dt8	XHIF_1_P3_OF_RW	000000h	r	w	Read/Write value of the offset register		

Register:	XHIF_1_P3_CFG				Address:	F8h	
Bits:	0dt0	Reset value:		0h	Attributes:	r	w
Description:	Configuration of the buffering mode for page #3						
Bit	Identifier	Reset	Attr.		Function / Description		
0	XHIF_1_P3_CFG_BUFMOD	0h	r	w	16bit access mode of page #3		

Register:	XHIF_1_VERSION				Address:	FCh	
Bits:	31dt0	Reset value:		04000804h	Attributes:	r	
Description:	Metal-fix register for XHIF Version						

5.3.4 Interrupt Controller Unit ARM-ICU

Base address see Chapter 5.2.

Address space:

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

	PRIOREG62	r	w	60F8h
	PRIOREG63	r	w	60FCh
	PRIOREG64	r	w	6100h
	PRIOREG65	r	w	6104h
	PRIOREG66	r	w	6108h
	PRIOREG67	r	w	610Ch
	PRIOREG68	r	w	6110h
	PRIOREG69	r	w	6114h
	PRIOREG70	r	w	6118h
	PRIOREG71	r	w	611Ch
	PRIOREG72	r	w	6120h
	PRIOREG73	r	w	6124h
	PRIOREG74	r	w	6128h
	PRIOREG75	r	w	612Ch
	PRIOREG76	r	w	6130h
	PRIOREG77	r	w	6134h
	PRIOREG78	r	w	6138h
	PRIOREG79	r	w	613Ch
	PRIOREG80	r	w	6140h
	PRIOREG81	r	w	6144h
	PRIOREG82	r	w	6148h
	PRIOREG83	r	w	614Ch
	PRIOREG84	r	w	6150h
	PRIOREG85	r	w	6154h
	PRIOREG86	r	w	6158h
	PRIOREG87	r	w	615Ch
	PRIOREG88	r	w	6160h
	PRIOREG89	r	w	6164h
	PRIOREG90	r	w	6168h
	PRIOREG91	r	w	616Ch
	PRIOREG92	r	w	6170h
	PRIOREG93	r	w	6174h
	PRIOREG94	r	w	6178h
	PRIOREG95	r	w	617Ch
/icu_ertec_addr_dec_top/icu8_inst				
	ID_REGISTER	r		8000h
	IRVEC	rh		8004h
	ACK	rht		8008h
	IRCLVEC		wt	800Ch
	MASKALL	r	w	8010h
	EOI		t	8014h
	UNLOCK_RD_ONLY_ACK	r	w	8018h
	MASK_ALL_INPUT_EN	r	w	801Ch
	LOCKREG	r	w	8020h
	MASKREG0	r	w	9000h
	IRR0	rh		9100h
	ISR0	rh		9200h
	TRIGREG0	r	w	9300h
	EDGEREG0	r	w	9400h
	SWIRREG0	rh	w	9500h

	PRIOREG0	r	w	A000h
	PRIOREG1	r	w	A004h
	PRIOREG2	r	w	A008h
	PRIOREG3	r	w	A00Ch
	PRIOREG4	r	w	A010h
	PRIOREG5	r	w	A014h
	PRIOREG6	r	w	A018h
	PRIOREG7	r	w	A01Ch
/icu_ertec_addr_dec_top/icu_general_registers_inst				
	FIQ_SEL_0	r	w	48000h
	FIQ_SEL_1	r	w	48004h
	FIQ_SEL_2	r	w	48008h
	FIQ_SEL_3	r	w	4800Ch
	FIQ_SEL_4	r	w	48010h
	FIQ_SEL_5	r	w	48014h
	FIQ_SEL_6	r	w	48018h
	FIQ_SEL_7	r	w	4801Ch

Register description:

A '0' is read from Software for each not specified Bit in the registers.
The addressing is organized: WORD - wise

Module: /icu_ertec_addr_dec_top
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 6:ICU core IP					

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

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

Register:	IRCLVEC	Address:	400Ch

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

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

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

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

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

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

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

Register:	MASKREG1			Address:	5004h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w

Description:	Interrupt Mask Register Enable/disable the interrupt inputs 32 – 63 inputs of the interrupt controller 0' = Interrupt input enabled '1' = Interrupt input disabled

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

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

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

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

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

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

Register:	ISR2	Address:	5208h

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Register:	PRIOREG4			Address:	6010h	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w

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

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

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

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

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

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

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

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

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

Register:	PRIOREG13	Address:	6034h

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

	Binary code of the priority input

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

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

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

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

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

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

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

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

Register:	PRIOREG39			Address:	609Ch	
Bits:	6dt0	Reset value:	5Fh	Attributes:	r	w

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

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

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

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

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

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

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

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

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

Register:	PRIOREG48	Address:	60C0h

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

	Binary code of the priority input

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Register:	PRIOREG83	Address:	614Ch

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

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

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

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

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

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

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

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

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

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

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

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

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

Module: /icu_ertec_addr_dec_top/icu8_inst

Register:	ID_REGISTER			Address:	8000h	
Bits:	15dt0	Reset value:	0006h	Attributes:	r	
Description:	Version number of the Interrupt Controller Unit Version number 6:ICU core IP					

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

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

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

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

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

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

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

Register:	LOCKREG				Address:	8020h	
Bits:	31dt0		Reset value:	00000000h	Attributes:	r	w
Description:	Priority Lock Register Specification of a priority to lock interrupt requests with lower or equal priority						
Bit	Identifier	Reset	Attr.		Function / Description		
2dt0	LOCK_PRIO	0h	r	w	Binary code of the locking priority		
31	LOCKREG_ENABLE	0h	r	w	0 = lock inactive / 1 = lock active		

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

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

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

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

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

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

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

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

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

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

Register:	PRIOREG4			Address:	A010h	

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

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

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

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

Module: /icu_ertec_addr_dec_top/icu_general_registers_inst

Register:	FIQ_SEL_0			Address:	48000h	
Bits:	6dt0	Reset value:	00h	Attributes:	r	w
Description:	Important: This FIQ_SEL0 register can be written and read by Software - nevertheless this register does not affect the Hardware at all					

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

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

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

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

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

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

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

5.3.5 General DMA Controller - GDMA

Base address see Chapter 5.2.

Address space:

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

Module	Register/Memory	Read	Write	Address	Revision
/gdma					
	GDMA_REG_ADDR	r	(w)	0h	SOC2.0
	GDMA_LIST_ADDR	r	(w)	4h	SOC2.0
	GDMA_MAIN_CTRL	r	(w)	8h	SOC2.0
	GDMA_JC_EN	r	w	Ch	SOC2.0
	GDMA_JOB0_CTRL	(r)	(w)(t)	10h	SOC2.0
	GDMA_JOB1_CTRL	(r)	(w)(t)	14h	SOC2.0
	GDMA_JOB2_CTRL	(r)	(w)(t)	18h	SOC2.0
	GDMA_JOB3_CTRL	(r)	(w)(t)	1Ch	SOC2.0
	GDMA_JOB4_CTRL	(r)	(w)(t)	20h	SOC2.0
	GDMA_JOB5_CTRL	(r)	(w)(t)	24h	SOC2.0
	GDMA_JOB6_CTRL	(r)	(w)(t)	28h	SOC2.0
	GDMA_JOB7_CTRL	(r)	(w)(t)	2Ch	SOC2.0
	GDMA_JOB8_CTRL	(r)	(w)(t)	30h	SOC2.0
	GDMA_JOB9_CTRL	(r)	(w)(t)	34h	SOC2.0
	GDMA_JOB10_CTRL	(r)	(w)(t)	38h	SOC2.0
	GDMA_JOB11_CTRL	(r)	(w)(t)	3Ch	SOC2.0
	GDMA_JOB12_CTRL	(r)	(w)(t)	40h	SOC2.0
	GDMA_JOB13_CTRL	(r)	(w)(t)	44h	SOC2.0
	GDMA_JOB14_CTRL	(r)	(w)(t)	48h	SOC2.0
	GDMA_JOB15_CTRL	(r)	(w)(t)	4Ch	SOC2.0
	GDMA_JOB16_CTRL	(r)	(w)(t)	50h	SOC2.0
	GDMA_JOB17_CTRL	(r)	(w)(t)	54h	SOC2.0
	GDMA_JOB18_CTRL	(r)	(w)(t)	58h	SOC2.0
	GDMA_JOB19_CTRL	(r)	(w)(t)	5Ch	SOC2.0
	GDMA_JOB20_CTRL	(r)	(w)(t)	60h	SOC2.0
	GDMA_JOB21_CTRL	(r)	(w)(t)	64h	SOC2.0
	GDMA_JOB22_CTRL	(r)	(w)(t)	68h	SOC2.0
	GDMA_JOB23_CTRL	(r)	(w)(t)	6Ch	SOC2.0
	GDMA_JOB24_CTRL	(r)	(w)(t)	70h	SOC2.0
	GDMA_JOB25_CTRL	(r)	(w)(t)	74h	SOC2.0
	GDMA_JOB26_CTRL	(r)	(w)(t)	78h	SOC2.0
	GDMA_JOB27_CTRL	(r)	(w)(t)	7Ch	SOC2.0
	GDMA_JOB28_CTRL	(r)	(w)(t)	80h	SOC2.0
	GDMA_JOB29_CTRL	(r)	(w)(t)	84h	SOC2.0
	GDMA_JOB30_CTRL	(r)	(w)(t)	88h	SOC2.0
	GDMA_JOB31_CTRL	(r)	(w)(t)	8Ch	SOC2.0
	GDMA_JOB_STATUS	rh		90h	SOC2.0
	GDMA_FINISHED_JOBS	rh	w	94h	SOC2.0

	GDMA_ACTUAL_STATUS	r(h)		98h	SOC2.0
	GDMA_IRQ_STATUS	rh	w	9Ch	SOC2.0
	GDMA_ERR_IRQ_STATUS	r(h)	(w)	A0h	SOC2.0
	GDMA_JOB_COUNT	rh		A4h	SOC2.0
	REVISION_CODE	r		A8h	SOC2.0
	LIST_RAM	R	W	B0h - 10AFh	
	JOB_STACK_RAM	R	W	10B0h - 12AFh	

Register description:

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

Module: /gdma

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

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

Register:	GDMA_MAIN_CTRL				Address:	8h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	(w)
Description:	Main Control register						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	DMA_EN	0h	r	w	Global Enable		
1dt1	SW_RESET	0h	r	w	Software Reset '0' = don't care, '1' = Reset		
14dt2	<notused>	00h	r		all bits = '0'		
15dt15	JC_RESET	0h	r	w	Reset Job Counter '0' = don't care, '1' = Reset		
16dt16	ERR_INT_EN	0h	r	w	Error Interrupt Enable		
23dt17	<notused>	00h	r		all bits = '0'		
31dt24	LIST_SIZE	00h	r	w	Total Number of Transfers in Transfer List (0-255) 0 means 1, 255-> 256		

Register:	GDMA_JC_EN			Address:	Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w

Description:	Enable Job Counter for Job

Register:	GDMA_JOB0_CTRL			Address:	10h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 0					

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

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

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

Register:	GDMA_JOB2_CTRL			Address:	18h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 2					

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

Register:	GDMA_JOB3_CTRL			Address:	1Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 3					

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

Register:	GDMA_JOB4_CTRL			Address:	20h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 4					

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable

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

Register:	GDMA_JOB5_CTRL				Address:	24h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 5						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	<notused>	0h	r		all bits = '0'		
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB6_CTRL				Address:	28h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 6						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		

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

Register:	GDMA_JOB7_CTRL				Address:	2Ch	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 7						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	<notused>	0h	r		all bits = '0'		
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB8_CTRL				Address:	30h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 8						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		

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

Register:	GDMA_JOB9_CTRL			Address:	34h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 9					

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

Register:	GDMA_JOB10_CTRL			Address:	38h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 10					

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

Register:	GDMA_JOB11_CTRL	Address:	3Ch

Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 11					

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

Register:	GDMA_JOB12_CTRL			Address:	40h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 12					

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

Register:	GDMA_JOB13_CTRL			Address:	44h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 13					

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start

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

Register:	GDMA_JOB14_CTRL				Address:	48h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 14						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		
13dt8	HW_SELECT	00h	r	w	HW Job Start Selector (selects one from 64 inputs)		
15dt14	<notused>	0h	r		all bits = '0'		
20dt16	JOB_PRIO	00h	r	w	Job Priority (0-31)		
23dt21	<notused>	0h	r		all bits = '0'		
31dt24	TRANSFER_PTR	00h	r	w	First Transfer Number of Job in Transfer List (0-255)		

Register:	GDMA_JOB15_CTRL				Address:	4Ch	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 15						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		

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

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

Register:	GDMA_JOB17_CTRL				Address:	54h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 17						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		
5dt5	JOB_RESET	0h	r	w	Reset Job (cancel running job) 0 = don't care, 1 = Cancel		
7dt6	<notused>	0h	r		all bits = '0'		

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

Register:	GDMA_JOB18_CTRL			Address:	58h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 18					

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

Register:	GDMA_JOB19_CTRL			Address:	5Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 19					

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

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

Register:	GDMA_JOB20_CTRL			Address:	60h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 20					

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

Register:	GDMA_JOB21_CTRL			Address:	64h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 21					

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

Register:	GDMA_JOB22_CTRL			Address:	68h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 22					

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

Register:	GDMA_JOB23_CTRL			Address:	6Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 23					

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

Register:	GDMA_JOB24_CTRL			Address:	70h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 24					

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable

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

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

Register:	GDMA_JOB26_CTRL				Address:	78h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 26						
Bit	Identifier	Reset	Attr.		Function / Description		
0dt0	SW_JOB_START	0h		t	Start Job from SW 0 = don't care, 1 = Start		
1dt1	JOB_EN	0h	r	w	Job Enable 0 = disable, 1 = enable		
2dt2	HW_JOB_START_EN	0h	r	w	Start Job by Hardware 0 = disable, 1 = enable		
3dt3	HW_FLOW_EN	0h	r	w	Hardware Triggered Flow Enable 0 = disabled, 1 = enabled		
4dt4	INTR_EN	0h	r	w	Interrupt Request Generation Enable 0 = disabled, 1 = enabled		

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

Register:	GDMA_JOB27_CTRL			Address:	7Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 27					

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

Register:	GDMA_JOB28_CTRL			Address:	80h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 28					

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

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

Register:	GDMA_JOB29_CTRL			Address:	84h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 29					

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

Register:	GDMA_JOB30_CTRL			Address:	88h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)(t)
Description:	Job Control Registers 30					

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

Register:	GDMA_JOB_STATUS	Address:	90h

Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Job (30:0) State 0 = not started, 1 = started					

Register:	GDMA_FINISHED_JOBS			Address:	94h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	w
Description:	Finished Job(30:0) State (write 1 to clear) (INTR_EN independent) 0 = not finished, 1 = finished					

Register:	GDMA_ACTUAL_STATUS			Address:	98h	
Bits:	31dt0	Reset value:	0000001Fh	Attributes:	r(h)	
Description:	Actual Job State					

Bit	Identifier	Reset	Attr.		Function / Description
4dt0	ACT_JOB	1Fh	rh		Actual Job Number – the number of the running Job
5dt5	ACT_JOB_VAL	0h	rh		Actual Job Number Valid 0 = no job running, 1 = Job Number valid Note: this flag is needed for ACT_JOB = “00000”
31dt6	<notused>	0000000h	r		all bits = '0'

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

Register:	GDMA_ERR_IRQ_STATUS			Address:	A0h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r(h)	(w)
Description:	Error interrupt State register					

Bit	Identifier	Reset	Attr.		Function / Description
0dt0	ERR_DST_ADDR	0h	rh	w	DMA Destination Address Error (write 1 to clear) 0 = no Error, 1 = Error occurred
5dt1	ERR_DST_ADDR_JOB_NR	00h	rh		DMA Destination Address Error JOB number
6dt6	ERR_AHB	0h	rh	w	AHB Master Interface Error (write 1 to clear) 0 = no Error, 1 = AHB Error Response occurred
11dt7	ERR_AHB_JOB_NR	00h	rh		AHB Master Interface Error JOB number
12dt12	ERR_JOB_START	0h	rh	w	Hardware start of Job occurred befor Job is finished (write 1 to clear) 0 = no Error, 1 = Error
17dt13	ERR_JOB_START_JOB_NR	00h	rh		Hardware start of Job occurred 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	00h	rh		ERR_AM_HOLD JOB number
31dt25	<notused>	00h	r		all bits = '0'

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

Register:	REVISION_CODE			Address:	A8h	
Bits:	15dt0	Reset value:	0101h	Attributes:	r	
Description:	Revison Code register					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	MINOR_REVISION	01h	r		minor revision = 0x1
15dt8	MAJOR_REVISION	01h	r		major revision = 0x1

DMA RAM:

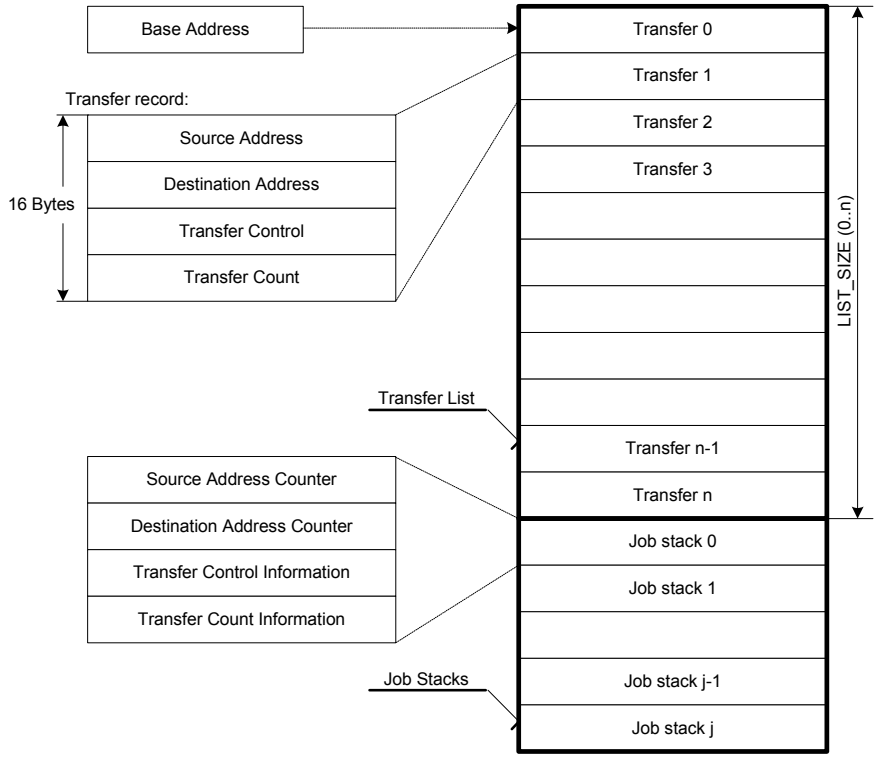


Figure 31: The DMA RAM address space.

DMA Transfer Record – Source Address SRC_ADDR

Bit no.	Name	Description
31:0	SRC	Source Address

DMA Transfer Record – Destination Address DEST_ADDR

Bit no.	Name	Description
31:0	DEST	Destination Address

DMA Transfer Record – Transfer Control

Bit no.	Name	Description
31:24		reserved
23:22	SRC_AMODE	Source Address Mode 00 = increment address 10 = hold address x1 = reserved for future implementation = hold address
21:20	DEST_AMODE	Destination Address Mode (reserved = hold address) 00 = increment address 10 = hold address x1 = reserved for future implementation = hold address
19:18	BURST_MODE	00 = Single 01 = INCR4 10 = INCR8 11 = INCR16
17:16		reserved

DMA Transfer Record – Transfer Count

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

DMA Stack – Source Address Stack

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

DMA Stack – Destination Address Stack

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

DMA Stack – Transfer Control Stack

Bit no.	Name	Description
31:24		reserved
23:22	SRC_AMODE	Source Address Mode 00 = increment address 10 = hold address x1 = reserved for future implementation = hold address
21:20	DEST_AMODE	Destination Address Mode 00 = increment address 10 = hold address x1 = reserved for future implementation = hold address
19:18	BURST_MODE	00 = Single 01 = INCR4 10 = INCR8 11 = INCR16
15:8		reserved
7:0	CTRANSFER	Current Transfer List Number to continue from after interruption

DMA Stack – Transfer Count Stack

Bit no.	Name	Description
31	LAST_TR	Last Transfer of the Job 0 = not last, 1=last

30	EN_DMA_ACK	1 = set output HW_DMA_ACK if transfer is finished
29:24		reserved
23:22	ESIZE	Element Size 00 = 8 bit 01 = 16 bit 10 = 32 bit 11 = reserved
21:16		reserved
15:0	CTR_COUNT	Current Transfer Count 0x0 = 1 element, 0xFFFF = 63536 elements

5.3.6 EMC-Register

Base address see Chapter 5.2.

Address space:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	34h	emc_reg	AHB

Module	Register/Memory	Read	Write	Address
/emc_reg				
	REVISION_CODE	r		0h
	ASYNC_WAIT_CYCLE_CONFIG	r	(w)	4h
	SDRAM_CONFIG	r	(w)(t)(p)	8h
	SDRAM_REFRESH	r(h)	(w)	Ch
	ASYNC_BANK0	r	w	10h
	ASYNC_BANK1	r	w	14h
	ASYNC_BANK2	r	w	18h
	ASYNC_BANK3	r	w	1Ch
	EXTENDED_CONFIG	r	(w)	20h
	AT_ADDR	rh		24h
	LPEMR	r	(w)	28h
	BF_CONFIG	r	(w)	2Ch
	RECOV_CONFIG	r	(w)	34h

Register description:

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

Module: /emc_reg

Register:	REVISION_CODE				Address:	0h	
Bits:	31dt0	Reset value:		04200803h	Attributes:	r	
Description:	Revision Code						
Bit	Identifier	Reset	Attr.		Function / Description		
10dt0	LABEL	003h	r		Release Label		
15dt11	INCREMENT	01h	r		Increment		
18dt16	PATCH	0h	r		Patch Version		
20dt19	PLATFORM	0h	r		Platform code		
31dt21	ID	021h	r		Identification code		

Register:	ASYNC_WAIT_CYCLE_CONFIG				Address:	4h	
Bits:	31dt0	Reset value:		40000080h	Attributes:	r	(w)
Description:	Async. Wait Cycle Configuration register						
Bit	Identifier	Reset	Attr.		Function / Description		
19dt0	MAX_EXT_WAIT	00080h	r	w	Maximum number of wait cycles. If an access is delayed by WAIT input for more than		

					[(MAX_EXT_WAIT+1) x 16] clocks, the access is completed.
23dt20	reserved_3	0h	r		reserved
24	BE_CTRL1	0h	r	w	Byte Enable Control Async. Bank 1: 0 = all Byte Enables low during read 1 = Byte Enable reflect AHB HSIZE during read
25	BE_CTRL2	0h	r	w	Byte Enable Control Async. Bank 2: 0 = all Byte Enables low during read 1 = Byte Enable reflect AHB HSIZE during read
26	BE_CTRL3	0h	r	w	Byte Enable Control Async. Bank 3: 0 = all Byte Enables low during read 1 = Byte Enable reflect AHB HSIZE during read
27	BE_CTRL4	0h	r	w	Byte Enable Control Async. Bank 4: 0 = all Byte Enables low during read 1 = Byte Enable reflect AHB HSIZE during read
29dt28	reserved_2	0h	r		reserved
30	WAIT_POLARITY	1h	r	w	WAIT input polarity 0 = low active 1 = high active
31	reserved_1	0h	r		reserved

Register:	SDRAM_CONFIG			Address:	8h	
Bits:	31dt0	Reset value:	00002020h	Attributes:	r	(w)(t)(p)
Description:	SDRAM Configuration register A write on this register starts a Load Mode Register sequence. This register can only be written after the SDRAM Initialization sequence is done (Bit 29 of register SDRAM_REFRESH).					

Bit	Identifier	Reset	Attr.		Function / Description
2dt0	PAGE_SIZE	0h	r	w	Page Size 000 = 8 Column address lines 001 = 9 Column address lines 010 = 10 Column address lines 011 = 11 Column address lines 100-111 = reserved
3	reserved_1	0h	r	w	reserved
6dt4	SDRAM_BANKS	2h	r	w	Number of banks inside SDRAM 000 = 1 Bank 001 = 2 Bank 010 = 4 Bank 011-111 = reserved
7	reserved_2	0h	r	w	reserved
10dt8	ROW_SIZE	0h	r	wpt	Row Size 000 = 8 Row address lines 001 = 9 Row address lines 010 = 10 Row address lines 011 = 11 Row address lines 100 = 12 Row address lines 101 = 13 Row address lines 110 = 14 Row address lines 111 = 15 Row address lines Write protected as long as bit29 in SDRAM_REFRESH = 0. A write starts a Mode Register Set to external SDRAM.
12dt11	reserved_3	0h	r		reserved
13	CL	1h	r	wpt	CAS Latency 0 = 2 clocks 1 = 3 clocks

					Write protected as long as bit29 in SDRAM_REFRESH = 0. A write starts a Mode Register Set to external SDRAM.
31dt14	reserved_4	00000h	r		reserved

Register:	SDRAM_REFRESH				Address:	Ch	
Bits:	31dt0	Reset value:		00000190h	Attributes:	r(h)	(w)
Description:	SDRAM Refresh Control register						
Bit	Identifier	Reset	Attr.		Function / Description		
12dt0	REFRESH_RATE	0190h	r	w	Refresh Rate Number of clocks between 2 SDRAM Refresh cycles = REFRESH_RATE + 1		
28dt13	reserved_1	0000h	r		reserved		
29	INIT_DONE	0h	rh		SDRAM Initialization done 0 = default 1 = SDRAM initialization 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 ex-pired.		
31	reserved_2	0h	r		reserved		

Register:	ASYNC_BANK0				Address:	10h	
Bits:	31dt0	Reset value:		3FFFFFF2h	Attributes:	r	w
Description:	Async. Bank 0 Configuration register						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	SIZE	2h	r	w	Bank Size 00 = 8 bit device 01 = 16 bit device 10 = 32 bit device 11 = reserved		
3dt2	MODE	0h	r	w	Mode 00 = SRAM 01 = reserved 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 = reserved 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 = reserved 10 = reserved 11 = Burstflash ROM		
6dt4	R_HOLD	7h	r	w	Read Hold Cycles Hold Phase of read access lasts (R_HOLD + 1) clocks.		
12dt7	R_STROBE	3Fh	r	w	Read Strobe Duration Cycles Strobe Phase of read access (Read Enable = 0) lasts (R_STROBE +		

					1) clocks. Value 0 is not supported.
16dt13	R_SU	Fh	r	w	Read Setup Cycles Setup Phase of read access lasts R_SU clocks.
19dt17	W_HOLD	7h	r	w	Write Hold Cycles Hold Phase of write access lasts (W_HOLD + 1) clocks.
25dt20	W_STROBE	3Fh	r	w	Write Strobe Duration Cycles Strobe Phase of write access (Write Enable = 0) lasts (W_STROBE + 1) clocks.
29dt26	W_SU	Fh	r	w	Write Setup Cycles Setup Phase of write access lasts W_SU clocks.
30	EW	0h	r	w	Extend Wait mode 0 = WAIT input = don't care 1 = Wait input extends access (valid in SRAM mode only)
31	WSM	0h	r	w	Wait input synchronisation mode 0 = 2 Flipflops 1 = 1 Flipflop

Register:	ASYNC_BANK3				Address:	1Ch	
Bits:	31dt0	Reset value:		3FFFFFF2h	Attributes:	r	w
Description:	Async. Bank 3 Configuration register						
Bit	Identifier	Reset	Attr.		Function / Description		
1dt0	SIZE	2h	r	w	Bank Size 00 = 8 bit device 01 = 16 bit device 10 = 32 bit device 11 = reserved		
3dt2	MODE	0h	r	w	Mode 00 = SRAM 01 = reserved 10 = reserved 11 = Burstflash ROM		
6dt4	R_HOLD	7h	r	w	Read Hold Cycles Hold Phase of read access lasts (R_HOLD + 1) clocks.		
12dt7	R_STROBE	3Fh	r	w	Read Strobe Duration Cycles Strobe Phase of read access (Read Enable = 0) lasts (R_STROBE + 1) clocks. Value 0 is not supported.		
16dt13	R_SU	Fh	r	w	Read Setup Cycles Setup Phase of read access lasts R_SU clocks.		
19dt17	W_HOLD	7h	r	w	Write Hold Cycles Hold Phase of write access lasts (W_HOLD + 1) clocks.		
25dt20	W_STROBE	3Fh	r	w	Write Strobe Duration Cycles Strobe Phase of write access (Write Enable = 0) lasts (W_STROBE + 1) clocks.		
29dt26	W_SU	Fh	r	w	Write Setup Cycles Setup Phase of write access lasts W_SU clocks.		
30	EW	0h	r	w	Extend Wait mode 0 = WAIT input = don't care 1 = Wait input extends access (valid in SRAM mode only)		
31	WSM	0h	r	w	Wait input synchronisation mode 0 = 2 Flipflops 1 = 1 Flipflop		

Register:	EXTENDED_CONFIG			Address:	20h	
Bits:	31dt0	Reset value:	01F70000h	Attributes:	r	(w)
Description:	Extended Configuration register					

Bit	Identifier	Reset	Attr.		Function / Description
6dt0	reserved_1	00h	r	w	reserved
7	AT_EN	0h	r	w	Async. Timeout Enable 0 = disabled 1 = enabled
8	SDRAM_DW	0h	r	w	SDRAM interface data width 0 = 32 bit 1 = 16 bit
9	MOBILE_SDRAM	0h	r	w	Mobile SDRAM Mode 0 = disabled (1 SDRAM mode register) 1 = enabled (2 SDRAM mode register)
10	ASYNC_ADDR_MODE	0h	r	w	Async. Address Output Mode 0 = no address shift (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_2	0h	r	w	reserved
15dt14	TRCD	0h	r	w	TRCD (RAS to CAS delay) 00 = 2 clocks 01 = 3 clocks 10 = 4 clocks 11 = reserved
18dt16	SDRAM_BL	7h	r	w	SDRAM Burst Length This value is written into SDRAM Mode Register. 000 = 1 (32 bit SDRAM interface only) 001 = 2 (16 bit SDRAM interface only) 010 = 4 (not supported) 011 = 8 100-110 = reserved 111 = continuous/full page (recommended)
19	reserved_3	0h	r		reserved
23dt20	TRFC	Fh	r	w	AUTO REFRESH period (cycle) time Minimum time between to AUTO REFRESH commands in clocks 0x0 = reserved 0x1 = 4 clocks : 0xF = 18 clocks
24	ADB	1h	r	w	active data bus 0 = disabled 1 = enabled
25	reserved_4	0h	r	w	reserved
26	BFODM	0h	r	w	BurstflashROM Output Delay Mode 0 = XBF_AVD is delayed with register clocked by input Burst flashROM clock 1 = XBF_AVD is delayed with delay buffer . Value required for operation of ERTEC 200P = 1
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 Value required for operation of ERTEC 200P = 1
28	TM4	0h	r	w	Test Mode bit 4 0 = normal operation 1 = async write data outputs toggle 1 clock after address outputs
29	TM2	0h	r	w	Test Mode bit 2 0 = normal operation

					1 = all SDRAM reads/writes are of type miss (32 bit SDRAM only)
30	TM1	0h	r	w	Test Mode bit 1 0 = normal operation 1 = SDRAM initialization delay after reset is cleared
31	reserved_5	0h	r	w	reserved

Register:	AT_ADDR			Address:	24h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Async. Timeout Address register Contains after asyc. timeout address of the access, which was finished with timeout.					

Register:	LPEMR				Address:	28h	
Bits:	31dt0		Reset value:	00000000h	Attributes:	r	(w)
Description:	Low Power Extended Mode Register This value is written into Extended Mode Register of Mobile SDRAM						
Bit	Identifier		Reset	Attr.	Function / Description		
12dt0	LPEMR		0000h	r w	This value is written into Extended Mode Register of Mobile SDRAM. See Mobile SDRAM datasheet.		
31dt13	reserved_1		00000h	r	reserved		

Register:	BF_CONFIG			Address:	2Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	(w)
Description:	Burstflash ROM Configuration register					
Bit	Identifier	Reset	Attr.	Function / Description		
0	BF_CLK_F	0h	r w	Clock Frequency 0 = Burstflash clock = HCLK/2 Value required for operation of ERTEC 200P = 0 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_1	0h	r w	reserved		
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_2	0h	r w	reserved		
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
23	reserved_3	0h	r	w	reserved
31dt24	reserved_4	00h	r		reserved

Register:	RECOV_CONFIG				Address:	34h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	(w)
Description:	Recovery Phase Configuration register						
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	RECOV0	0h	r	w	Async. Bank 0 Recovery Phase 0000 = 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_1	0000h	r		reserved		

5.3.7 I²C register

Base addresses see Chapter 5.2.
Access to the I²C register is effected through "32-bit" addresses.

Adress space:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	68h	i2c	APB

Module	Register/Memory	Read	Write	Address
/i2c				
	MI2C_ADDR	r	w	0h
	MI2C_DATA	rh	w	4h
	MI2C_CNTR	r(h)	(w)	8h
	MI2C_STAT	rh		Ch
	MI2C_XADDR	r	w	10h
	MI2C_SOFTWARE_RESET		w	1Ch
	EX_CTRL_1	r(h)	(w)	20h
	EX_ADDR_1	r	w	24h
	EX_DATA_OUT_1	r	w	28h
	EX_DATA_IN_1	rh		2Ch
	EX_CTRL_2	r(h)	(w)	30h
	EX_ADDR_2	r	w	34h
	EX_DATA_OUT_2	r	w	38h
	EX_DATA_IN_2	rh		3Ch
	EX_CTRL_3	r(h)	(w)	40h
	EX_ADDR_3	r	w	44h
	EX_DATA_OUT_3	r	w	48h
	EX_DATA_IN_3	rh		4Ch
	EX_CTRL_4	r(h)	(w)	50h
	EX_ADDR_4	r	w	54h
	EX_DATA_OUT_4	r	w	58h
	EX_DATA_IN_4	rh		5Ch
	ERROR_SLAVE_ADDRESS	r	w	60h
	SW_I2C_EN	r	w	64h
	SW_I2C_CTRL	r(h)	(w)	68h

Register description:

Module: /i2c

Register:	MI2C_ADDR				Address:	0h	
Bits:	7dt0	Reset value:	00h		Attributes:	r	w
Description:	Slave Adress						
Bit	Identifier	Reset	Attr.		Function / Description		
0	GCE	0h	r	w	General Call adress enable		
1	SLA0	0h	r	w	Slave Adress (1-0), Extended Adress (9-8)		

2	SLA1	0h	r	w	Slave Adress (1-0), Extended Adress (9-8)
3	SLA2	0h	r	w	Slave Adress (6-2)
4	SLA3	0h	r	w	Slave Adress (6-2)
5	SLA4	0h	r	w	Slave Adress (6-2)
6	SLA5	0h	r	w	Slave Adress (6-2)
7	SLA6	0h	r	w	Slave Adress (6-2)

Register:	MI2C_DATA			Address:	4h	
Bits:	7dt0	Reset value:	00h	Attributes:	rh	w
Description:	Transfer Data Register; Receive-/Transmit-Data					

Register:	MI2C_CNTR				Address:	8h	
Bits:	7dt2	Reset value:		00h	Attributes:	r(h)	(w)
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
2	AAK	0h	r	w	Assert acknowledge		
3	IFLG	0h	rh		Interrupt flag		
4	STP	0h	rh	w	Master mode stop		
5	STA	0h	rh	w	Master mode start		
6	ENAB	0h	r	w	Bus enable		
7	IEN	0h	r	w	Interrupt enable		

Register:	MI2C_STAT			Address:	Ch	
Bits:	7dt0	Reset value:	F8h	Attributes:	rh	
Description:	Status register with content the following 31 status codes: 00h: Bus error 08h: START condition transmitted 10h: Repeated START condition transmitted 18h: Address + write bit transmitted, ACK received 20h: Address + write bit transmitted, ACK not received 28h: Data byte transmitted in master mode, ACK received 30h: Data byte transmitted in master mode, ACK not received 38h: Arbitration lost in address or data byte 40h: Address + read bit transmitted, ACK received 48h: Address + read bit transmitted, ACK not received 50h: Data byte received in master mode, ACK transmitted 58h: Data byte received in master mode, not ACK transmitted 60h: Slave address + write bit received, ACK transmitted 68h: Arbitration lost in address as master, slave address + write bit received, ACK transmitted 70h: General Call address received, ACK transmitted 78h: Arbitration lost in address as master, General Call address received, ACK transmitted 80h: Data byte received after slave address received, ACK transmitted 88h: Data byte received after slave address received, not ACK transmitted 90h: Data byte received after General Call received, ACK transmitted 98h: Data byte received after General Call received, not ACK transmitted A0h: STOP or repeated START condition received in slave mode A8h: Slave address + read bit received, ACK transmitted B0h: Arbitration lost in address as master, slave address + read bit received, ACK transmitted B8h: Data byte transmitted in slave mode, ACK received C0h: Data byte transmitted in slave mode, ACK not received C8h: Last byte transmitted in slave mode, ACK received D0h: Second Address byte + write bit transmitted, ACK received D8h: Second Address byte + write bit transmitted, ACK not received					

	E0h: Second Address byte + read bit transmitted, ACK received E8h: Second Address byte + read bit transmitted, ACK not received F8h: No relevant status information, IFLG=0

Register:	MI2C_XADDR			Address:	10h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:	Extented Slave Adress					

0	SLAX0	0h	r	w	Extended Slave Address
1	SLAX1	0h	r	w	Extended Slave Address
2	SLAX2	0h	r	w	Extended Slave Address
3	SLAX3	0h	r	w	Extended Slave Address
4	SLAX4	0h	r	w	Extended Slave Address
5	SLAX5	0h	r	w	Extended Slave Address
6	SLAX6	0h	r	w	Extended Slave Address
7	SLAX7	0h	r	w	Extended Slave Address

Register:	MI2C_SOFTWARE_RESET			Address:	1Ch	
Bits:	7dt0	Reset value:	00h	Attributes:		w
Description:	Software Reset					

Register:	EX_CTRL_1			Address:	20h	
Bits:	7dt0	Reset value:	0h	Attributes:	r(h)	(w)
Description:	Control register for IO expander 1.					

Bit	Identifier	Reset	Attr.		Function / Description
0	MODE	0h	r	w	0 = Service on demand, default 1 = Periodic service
1	OUT	0h	rh	w	0 = Write service off, default 1 = Write service on Send output data as specified in MODE
2	IN	0h	rh	w	0 = Read service off, default 1 = Read service on Read input data as specified in MODE
3	BUSY	0h	rh		Read-only. 0 = Currently inactive, default 1 = Currently active
4	ERROR	0h	rh		Read-only. 0 = No Error, default 1 = Error occured
7	SCL_Toggle	0h	r	w	SCL toggle bit: '0' keeps SCL high, '1' pulls SCL to low.

Register:	EX_ADDR_1			Address:	24h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:	Slave address for expander 1.					

Register:	EX_DATA_OUT_1			Address:	28h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:	Send data for expander 1.					

Register:	EX_DATA_IN_1			Address:	2Ch	
Bits:	7dt0	Reset value:	00h	Attributes:	rh	
Description:	Read data for expander 1.					

Register:	EX_CTRL_2				Address:	30h	
Bits:	4dt0	Reset value:	0h		Attributes:	r(h)	(w)
Description:	Control register for IO expander 2.						

Bit	Identifier	Reset	Attr.		Function / Description
0	MODE	0h	r	w	0 = Service on demand, default 1 = Periodic service
1	OUT	0h	rh	w	0 = Write service off, default 1 = Write service on Send output data as specified in MODE
2	IN	0h	rh	w	0 = Read service off, default 1 = Read service on Read input data as specified in MODE
3	BUSY	0h	rh		Read-only. 0 = Currently inactive, default 1 = Currently active
4	ERROR	0h	rh		Read-only. 0 = No Error, default 1 = Error occured

Register:	EX_ADDR_2				Address:	34h	
Bits:	7dt0	Reset value:	00h		Attributes:	r	w
Description:	Slave address for expander 2.						

Register:	EX_DATA_OUT_2				Address:	38h	
Bits:	7dt0	Reset value:	00h		Attributes:	r	w
Description:	Send data for expander 2.						

Register:	EX_DATA_IN_2				Address:	3Ch	
Bits:	7dt0	Reset value:	00h		Attributes:	rh	
Description:	Read data for expander 2.						

Register:	EX_CTRL_3				Address:	40h	
Bits:	4dt0	Reset value:	0h		Attributes:	r(h)	(w)
Description:	Control register for IO expander 3.						

Bit	Identifier	Reset	Attr.		Function / Description
0	MODE	0h	r	w	0 = Service on demand, default 1 = Periodic service
1	OUT	0h	rh	w	0 = Write service off, default 1 = Write service on Send output data as specified in MODE
2	IN	0h	rh	w	0 = Read service off, default 1 = Read service on Read input data as specified in MODE
3	BUSY	0h	rh		Read-only. 0 = Currently inactive, default 1 = Currently active
4	ERROR	0h	rh		Read-only. 0 = No Error, default 1 = Error occured

Register:	EX_ADDR_3				Address:	44h	
Bits:	7dt0	Reset value:	00h		Attributes:	r	w

Description:	Slave address for expander 3.

Register:	EX_DATA_OUT_3			Address:	48h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:	Send data for expander 3.					

Register:	EX_DATA_IN_3			Address:	4Ch	
Bits:	7dt0	Reset value:	00h	Attributes:	rh	
Description:	Read data for expander 3.					

Register:	EX_CTRL_4			Address:	50h	
Bits:	4dt0	Reset value:	0h	Attributes:	r(h)	(w)
Description:	Control register for IO expander 4.					

Bit	Identifier	Reset	Attr.		Function / Description
0	MODE	0h	r	w	0 = Service on demand, default 1 = Periodic service
1	OUT	0h	rh	w	0 = Write service off, default 1 = Write service on Send output data as specified in MODE
2	IN	0h	rh	w	0 = Read service off, default 1 = Read service on Read input data as specified in MODE
3	BUSY	0h	rh		Read-only. 0 = Currently inactive, default 1 = Currently active
4	ERROR	0h	rh		Read-only. 0 = No Error, default 1 = Error occured

Register:	EX_ADDR_4			Address:	54h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:	Slave address for expander 4.					

Register:	EX_DATA_OUT_4			Address:	58h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:	Send data for expander 4.					

Register:	EX_DATA_IN_4			Address:	5Ch	
Bits:	7dt0	Reset value:	00h	Attributes:	rh	
Description:	Read data for expander 4.					

Register:	ERROR_SLAVE_ADDRESS			Address:	60h	
Bits:	7dt0	Reset value:	FAh	Attributes:	r	w
Description:	Fictive target address used to automatically create a stop condition following a failed start condition. Bit 0 activates this automatic action.					

Bit	Identifier	Reset	Attr.		Function / Description
0	ERR_ON_OFF	0h	r	w	Err_on_off = '0' : Error routine off (default). Err_on_off = '1' : Error routine on.
7dt1	ERROR_SLAVE_ADDRESS	7Dh	r	w	Fictive target address

Register:	SW_I2C_EN			Address:	64h	
Bits:	0dt0	Reset value:	0h	Attributes:	r	w
Description:	Software I2C enable: 0 = disabled 1 = enabled					

Register:	SW_I2C_CTRL			Address:	68h	
Bits:	3dt0	Reset value:	Fh	Attributes:	r(h)	(w)
Description:	Software I2C control register.					

Bit	Identifier	Reset	Attr.		Function / Description
0	SW_SCL_O	1h	r	w	Software I2C clock output: SW_SCL_O = '1' : I2C clock output is 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 clock output: SDA_I = '1' : I2C data line = '1'. SDA_I = '0' : I2C clock line is '0'.

5.3.8 System-Control-Register-Block (SCRB)

Basisadresse: siehe Kap. 5.2.

Adressraum:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	B8h	SCRB	APB

Module	Register/Memory	Read	Write	Address
/SCRB				
	ID_REG	r		0h
	BOOT_REG	rh		4h
	CONFIG_REG	rh	(w)	8h
	ASYN_RES_CTRL_REG	rh	w	Ch
	SYN_RES_CTRL_REG	r	w	10h
	RES_STAT_REG	rh		14h
	PLL_STAT_REG	rh		18h
	QVZ_AHB_ADR	rh		1Ch
	QVZ_AHB_CTRL	rh		20h
	QVZ_AHB_M	rh		24h
	QVZ_APB_ADR	rh		28h
	QVZ_EMC_ADR	rh		2Ch
	MEM_SWAP	r	w	30h
	ERTEC200PLUS_TAG	r		38h
	AHB_BURSTBREAKER	r	w	44h
	CCR_I2C	r	w	50h
	EDC_EVENT	rh	w	54h
	EDC_INIT_DONE	rh		58h
	TCM926_MAP	r	w	5Ch
	GPIO_INT_POLSEL	r	w	60h
	EDC_PARITY_EN	r	w	64h
	MODUL_ACCESS_ERR	rh	w	68h
	RES_SOFT_RETURN_ADDR	r	w	6Ch
	DRIVE_EMC	r	w	78h
	DRIVE15_0GPIO	r	w	7Ch
	DRIVE31_16GPIO	r	w	80h
	DRIVE47_32GPIO	r	w	84h
	DRIVE63_48GPIO	r	w	88h
	DRIVE79_64GPIO	r	w	8Ch
	DRIVE95_80GPIO	r	w	90h
	PULL15_0GPIO	rh	w	94h
	PULL31_16GPIO	rh	w	98h
	PULL47_32GPIO	rh	w	9Ch
	PULL63_48GPIO	rh	w	A0h
	PULL79_64GPIO	rh	w	A4h
	PULL95_80GPIO	rh	w	A8h
	XHIF_MODE	r	w	B8h

Register description:

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

Module: /SCRB

Register:	ID_REG			Address:	0h	
Bits:	31dt0	Reset value:	40280100h	Attributes:	r	
Description:	Identification ERTEC200 P					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	MET_FIX	00h	r		Metall-Fix: 00h
15dt8	HW_R	01h	r		HW-Release: 01h
31dt16	COMP	4028h	r		ERTEC 200P identifier 4028h

Register:	BOOT_REG			Address:	4h	
Bits:	3dt0	Reset value:	0h	Attributes:	rh	
Description:	Boot mode pins BOOT (3:0) can be read					

Bit	Identifier	Reset	Attr.		Function / Description
0	BOOT_0	0h	rh		The value present at the Boot(0) pin is latched in when the PowerOn Reset is deactivated
1	BOOT_1	0h	rh		The value present at the Boot(1) pin is latched in when the PowerOn Reset is deactivated
2	BOOT_2	0h	rh		The value present at the Boot(2) pin is latched in when the PowerOn Reset is deactivated
3	BOOT_3	0h	rh		The value present at the Boot(3) pin is latched in when the PowerOn Reset is deactivated

Register:	CONFIG_REG			Address:	8h	
Bits:	6dt0	Reset value:	0h	Attributes:	rh	(w)
Description:	ERTEC 200P Config Pins CONFIG(6:0) readable, Config-REG(6-3) writable					

Bit	Identifier	Reset	Attr.		Function / Description
0	CONF_0	0h	rh		The value present at the Config(0) pin is latched in when the PowerOn Reset is deactivated. (see Chapter 4.2)
1	CONF_1	0h	rh		The value present at the Config(1) pin is latched in when the PowerOn Reset is deactivated. (see Chapter 4.2)
2	CONF_2	0h	rh		The value present at the Config(2) pin is latched in when the PowerOn Reset is deactivated. (see Chapter 4.2)
6dt3	CONF_6_3	0h	rh	w	The value present at the Config(6-3) pin is latched in when the PowerOn Reset is deactivated and after deactivation of the internally extended system reset imported into the register. Subsequently the value can be changed via write access (see Chapter 4.2)

Register:	ASYN_RES_CTRL_REG			Address:	Ch	
Bits:	31dt0	Reset value:	28h	Attributes:	rh	w
Description:	Control register for resetting the ERTEC 200P Resetting is effected for EN_WD_RES_PN via **XRES_PNIP**, for RES_SOFT_PN-IP_CORE via **XRES_PWONRES**.					

	The other Bits are resetted via **XRES_SYS**.				
Bit	Identifier	Reset	Attr.		Function / Description
0	WD_RES_FREI_ARM926	0h	rh	w	1: Enable Watchdog Reset for the ARM926
1	RES_SOFT	0h	rh	w	1:Asynchronous Software Reset (non-storing, PULSE_DUR forms reset length) resets all except PN-IP
2	RES_SOFT_PN	0h	rh	w	1: Asynchronous Software Reset (non-storing, PULSE_DUR forms reset length) resets only the PN-IP
3	EN_WD_RES_PN	1h	rh	w	0: PN-IP are not reset at the ARM926 Watchdog Reset 1: PN-IP are reset at the ARM926 Watchdog Reset
4	<reserved>	0h	rh		
5	<reserved>	1h	rh		
6	RES_SOFT_ARM926_CORE	0h	rh	w	1: Asynchronous Software Reset (non-storing, PULSE_DUR forms reset length) resets only the ARM926EJ-S Core system
7	<reserved>	0h	rh		
31dt16	PULSE_DUR	0000h	rh	w	Pulse duration of the SW Resets RES_SOFT, RES_SOFT_PN, ARM926 Watchdog and PN-IP Watchdog Reset TRES_PULSE = (8 x n + 8) x TCLK TCLK: APB cycle period (1/125 MHz = 8 ns) n: Value of PULSE_DUR (0 .. 65535) The integrated PHYs require a reset duration of > 100 us. Therefore n > 1562 has to be set

Register:	SYN_RES_CTRL_REG				Address:	10h	
Bits:	2dt0	Reset value:		0h	Attributes:	r	w
Description:	Control register for synchronous resetting of the ERTEC 200P						
Bit	Identifier	Reset	Attr.		Function / Description		
0	SYN_RES_PER_IF	0h	r	w	0: No synchronous reset for PER-IF 1: Synchronous reset for PER-IF (storing)		
1	SYN_RES_HOST	0h	r	w	0: No synchronous reset for Host Interface 1: Synchronous reset for Host Interface (storing)		
2	SYN_RES_PN_IP	0h	r	w	0: No synchronous reset for PN-IP 1: Synchronous reset for PN-IP (storing)		

Register:	RES_STAT_REG				Address:	14h	
Bits:	3dt0	Reset value:		4h	Attributes:	rh	
Description:	Status register for resetting the ERTEC 200P. Only the bit of the last reset event that occurred is set always. The two other bits are reset.						
Bit	Identifier	Reset	Attr.		Function / Description		
0	ARM926_WDOG_RES	0h	rh		1: Last reset was reset via the ARM926 Watchdog		
1	SW_RES	0h	rh		1: Last reset was reset via the Software Reset		
2	PWRON_HW_RES	1h	rh		1: Last reset was reset via a PowerOn or Hardware Reset		
3	SW_RES_ARM926	0h	rh		1: Last reset was reset via the Software Reset ARM926 Core		

Register:	PLL_STAT_REG			Address:	18h	
Bits:	1dt0	Reset value:	1h	Attributes:	rh	
Description:	Status register for PLL of the ERTEC 200P					

Bit	Identifier	Reset	Attr.		Function / Description
0	LOCK	1h	rh		Lock: Latching in at the operating frequency; status of the PLL: 0: PLL is unlocked 1: PLL is locked This bit represents the current lock state of the PLL. Only readable
1	LOSS	0h	rh		Loss: Monitoring status PLL Input Clock 1: PLL Input Clock not recognized 0: PLL Input Clock exists This bit shows the current monitoring status of the PLL Input Clock. Only readable

Register:	QVZ_AHB_ADR			Address:	1Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Address of a false addressing at the Multi Layer AHB					

Register:	QVZ_AHB_CTRL			Address:	20h	
Bits:	6dt0	Reset value:	0h	Attributes:	rh	
Description:	Control signals of a false addressing at the Multi Layer AHB					

Bit	Identifier	Reset	Attr.		Function / Description
0	R_W	0h	rh		Read/Write 0: read 1: write
3dt1	HSIZE	0h	rh		HSIZE
6dt4	HBURST	0h	rh		HBURST

Register:	QVZ_AHB_M			Address:	24h	
Bits:	5dt0	Reset value:	0h	Attributes:	rh	
Description:	Master identifier of a false 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	rh		
3	PN	0h	rh		PN-IP
4	GDMA	0h	rh		GDMA
5	HOSTIF	0h	rh		Host Interface

Register:	QVZ_APB_ADR			Address:	28h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Address of a false addressing at the APB					

Register:	QVZ_EMC_ADR			Address:	2Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	

Description:	Address that resulted in time-out at the EMC

Register:	MEM_SWAP			Address:	30h	
Bits:	1dt0	Reset value:	0h	Attributes:	r	w
Description:	Memory Swapping in Segment 0 on the AHB (ROM, EMC-SDRAM, EMC-Standard-Memory, I/D-TCM)					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	SWAP	0h	r	w	Selection of the memory in Segment 0 on the AHB: 00: Boot ROM as of Adr 0h 01: EMC-SDRAM as of Adr 0h (only the first 64 MB) 10: EMC-Standard-Memory as of Adr 0h (only the first 64 MB) 11: reserved (no memory area laid to Adr 0h, at access time-out (QVZ) generated)

Register:	ERTEC200PLUS_TAG			Address:	38h	
Bits:	31dt0	Reset value:	800h	Attributes:	r	
Description:	Tag Number of the current switching state					

Bit	Identifier	Reset	Attr.		Function / Description
10dt0	R_LABEL	000h	r		R-label
15dt11	INKREMENT_NR	01h	r		Increment No. (is not used, see ID_REG Slice HW_R)
18dt16	PATCH_NR	0h	r		Patch No. (is not used, see ID_REG Slice MET_FIX)
20dt19	PLATFORM	0h	r		Platform:00 = ASIC, 01 = FPGA, 10 = reserved, 11 = user-defined
31dt21	IDENTIFICATION	000h	r		Identification (is not used, see ID_REG Slice COMP)

Register:	AHB_BURSTBREAKER			Address:	44h	
Bits:	23dt0	Reset value:	0h	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	00h	r	w	0: bypass 1-255: max. number of addresses at ARM926-D AHB port
15dt8	NR_ADDR_ARM926_I	00h	r	w	0: bypass 1-255: max. number of addresses at ARM926-I AHB port
23dt16	<reserved>	00h	r		

Register:	CCR_I2C			Address:	50h	
Bits:	7dt0	Reset value:	7Ch	Attributes:	r	w
Description:	Clock Control register for the I2C_3 interface divider value for determining the bit rate.					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	CDIV_VAL	7Ch	r	w	Divider value for determining the bit rate fBR = fCLK/(CCR_I2C+1); fCLK = fAPB =125 MHz in the I2C fBR: Bit rate clock (I2C) fCLK: I2C interface system cycle clock

					for fBR = 1MHz and fCLK =125MHz: CCR_I2C =124(dec.)

Register:	EDC_EVENT			Address:	54h	
Bits:	17dt0	Reset value:	0h	Attributes:	rh	w
Description:	EDC Event Register - to delete the register has to be overwritten with '0h'.					

Bit	Identifier	Reset	Attr.		Function / Description
0	I_TCM926_1B	0h	rh	w	A 1Bit Error occurred in the I-TCM of the ARM926 and has been corrected
1	I_TCM926_2B	0h	rh	w	A 2Bit Error occurred in the I-TCM of the ARM926
2	D_TCM926_1B	0h	rh	w	A 1Bit Error occurred in the D-TCM of the ARM926 and has been corrected
3	D_TCM926_2B	0h	rh	w	A 2Bit Error occurred in the D-TCM of the ARM926
4	GDMA_1B	0h	rh	w	A 1Bit Error occurred in the GDMA Memory and has been corrected
5	GDMA_2B	0h	rh	w	A 2Bit Error occurred in the GDMA Memory
6	PN_1B	0h	rh	w	A 1Bit Error occurred in one of the PN-IP memories and was corrected
7	PN_2B	0h	rh	w	A 2Bit Error occurred in one of the PN-IP memories
8	PERIF_1B	0h	rh	w	A 1Bit Error occurred in the PerlF Memory and has been corrected
9	PERIF_2B	0h	rh	w	A 2Bit Error occurred in the PerlF Memory
10	I_CACHE_PAR	0h	rh	w	A Parity Error occurred in the I-Cache during reading
11	D_CACHE_PAR	0h	rh	w	A Parity Error occurred in the D-Cache during reading
12	I_TAG_PAR	0h	rh	w	A Parity Error occurred in the I-Tag during reading
13	D_TAG_PAR	0h	rh	w	A Parity Error occurred in the D-Tag during reading
14	<reserved>	0h	rh		
15	<reserved>	0h	rh		
16	<reserved>	0h	rh		
17	<reserved>	0h	rh		

Register:	EDC_INIT_DONE			Address:	58h	
Bits:	6dt0	Reset value:	0h	Attributes:	rh	
Description:	EDC Init Done Register - the status can be read					

Bit	Identifier	Reset	Attr.		Function / Description
0	I_TCM926_INIT_DONE	0h	rh		1: The initialization of the EDC bits in the I-TCM of the ARM926 is completed
1	D_TCM926_INIT_DONE	0h	rh		1: The initialization of the EDC bits in the D-TCM of the ARM926 is completed
2	GDMA_INIT_DONE	0h	rh		1: The initialization of the EDC bits in the GDMA is completed
3	PN_IP_INIT_DONE	0h	rh		1: The initialization of the EDC bits in the PN-IP is completed
4	PERIF_INIT_DONE	0h	rh		1: The initialization of the EDC bits in the PER-IF is completed
5	<reserved>	0h	rh		

6	<reserved>	0h	rh		

Register:	TCM926_MAP			Address:	5Ch	
Bits:	3dt0	Reset value:	0h	Attributes:	r	w
Description:	TCM926 Map Register Distribution 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 of I-TCM 0: No booting of I-TCM. Booting in accordance with the parameterization of MEM_SWAP. 1: I-TCM is activated after ARM926Core Reset

Register:	GPIO_INT_POLSEL			Address:	60h	
Bits:	15dt0	Reset value:	0h	Attributes:	r	w
Description:	Interrupt-Polarität für GPIO-Interrupts (15:0)					

Bit	Identifier	Reset	Attr.		Function / Description
0	INT_POLSEL_GPIO0	0h	r	w	0: GPIO0 is not inverted to ARM-ICU (IRQ32) and to GDMA 1: GPIO0 is inverted to ARM-ICU (IRQ32) and to GDMA
1	INT_POLSEL_GPIO1	0h	r	w	0: GPIO1 is not inverted to ARM-ICU (IRQ33) and to GDMA 1: GPIO1 is inverted to ARM-ICU (IRQ33) and to GDMA
2	INT_POLSEL_GPIO2	0h	r	w	0: GPIO2 is not inverted to ARM-ICU (IRQ34) and to GDMA 1: GPIO2 is inverted to ARM-ICU (IRQ34) and to GDMA
3	INT_POLSEL_GPIO3	0h	r	w	0: GPIO3 is not inverted to ARM-ICU (IRQ35) and to GDMA 1: GPIO3 is inverted to ARM-ICU (IRQ35) and to GDMA
4	INT_POLSEL_GPIO4	0h	r	w	0: GPIO4 is not inverted to ARM-ICU (IRQ36) and PN-ICU 1 1: GPIO4 is inverted to ARM-ICU (IRQ36) and PN-ICU 1
5	INT_POLSEL_GPIO5	0h	r	w	0: GPIO5 is not inverted to ARM-ICU (IRQ37) and PN-ICU 1 1: GPIO5 is inverted to ARM-ICU (IRQ37) and PN-ICU 1
6	INT_POLSEL_GPIO6	0h	r	w	0: GPIO6 is not inverted to ARM-ICU (IRQ38) and PN-ICU 1 1: GPIO6 is inverted to ARM-ICU (IRQ38) and PN-ICU 1
7	INT_POLSEL_GPIO7	0h	r	w	0: GPIO7 is not inverted to ARM-ICU (IRQ39) and PN-ICU 1 1: GPIO7 is inverted to ARM-ICU (IRQ39) and PN-ICU 1
8	INT_POLSEL_GPIO8	0h	r	w	0: GPIO8 is not inverted to ARM-ICU (IRQ40) 1: GPIO8 is inverted to ARM-ICU (IRQ40)
9	INT_POLSEL_GPIO9	0h	r	w	0: GPIO9 is not inverted to ARM-ICU (IRQ41)

					1: GPIO9 is inverted to ARM-ICU (IRQ41)
10	INT_POLSEL_GPIO10	0h	r	w	0: GPIO10 is not inverted to ARM-ICU (IRQ42) 1: GPIO10 is inverted to ARM-ICU (IRQ42)
11	INT_POLSEL_GPIO11	0h	r	w	0: GPIO11 is not inverted to ARM-ICU (IRQ43) 1: GPIO11 is inverted to ARM-ICU (IRQ43)
12	INT_POLSEL_GPIO12	0h	r	w	0: GPIO12 is not inverted to ARM-ICU (IRQ44) 1: GPIO12 is inverted to ARM-ICU (IRQ44)
13	INT_POLSEL_GPIO13	0h	r	w	0: GPIO13 is not inverted to ARM-ICU (IRQ45) 1: GPIO13 is inverted to ARM-ICU (IRQ45)
14	INT_POLSEL_GPIO14	0h	r	w	0: GPIO14 is not inverted to ARM-ICU (IRQ46) 1: GPIO14 is inverted to ARM-ICU (IRQ46)
15	INT_POLSEL_GPIO15	0h	r	w	0: GPIO15 is not inverted to ARM-ICU (IRQ47) 1: GPIO15 is inverted to ARM-ICU (IRQ47)

Register:	EDC_PARITY_EN				Address:	64h	
Bits:	3dt0	Reset value:	0h		Attributes:	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: The EDC logic in ARM926 I- / D-TCM is enabled (default) 1: The EDC logic in ARM926 I- / D-TCM is disabled		
3	<reserved>	0h	r				

Register:	MODUL_ACCESS_ERR				Address:	68h	
Bits:	5dt0	Reset value:	0h		Attributes:	rh	w
Description:	Module Access Error Register (5:0) - to delete the register has to be overwritten with '0h'						
Bit	Identifier	Reset	Attr.		Function / Description		
0	PN_IP_ACCESS_ERR	0h	rh	w	0: No Access Error occurred in the PN-IP 1: An Access Error occurred in the PN-IP		
1	PER_IF_ACCESS_ERR	0h	rh	w	0: No Access Error occurred in the PER-IF 1: An Access Error occurred in the PER-IF		
2	I_FILTER_ACCESS_ERR	0h	rh	w	0: No Access Error occurred in the I-Filter 1: An Access Error occurred in the I-Filter		
3	HOST_IF_ACCESS_ERR	0h	rh	w	0: No Access Error occurred in the HOST-IF 1: An Access Error occurred in the HOST-IF		
4	SCRB_ACCESS_ERR	0h	rh	w	0: No Access Error occurred in the SCRB 1: An Access Error occurred in the SCRB		
5	<reserved>	0h	rh				

Register:	RES_SOFT_RETURN_ADDR				Address:	6Ch	
Bits:	31dt0	Reset value:	0h		Attributes:	r	w
Description:	Entry address for the secondary bootloader after a SW Reset ARM926EJ-S						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	RETURN_ADDRESS	00000000	r	w	Entry address for the secondary bootloader after		

		h			a RES_SOFT_ARM926_CORE. This SW reset is effected after the parameterization of the TCM 926 configuration in the TCM926_MAP register.

Register:	DRIVE_EMC			Address:	78h	
Bits:	25dt0	Reset value:	7FFFFFh	Attributes:	r	w
Description:	SCRB Drive Current of Dedicated Signals (1.8V) Every GPIO bit is set in accordance with the following coding on the drive current -----1.8V-------- 00 - 4 mA 01 - 6 mA 10 - 8 mA 11 - 12 mA and Enable/Disable of the cycles for Burst Flash and SDRAM					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	DR_EMC_C	3h	r	w	Signal list: DTXR, XOE_DDRIVER
3dt2	DR_EMC_AL	3h	r	w	Signal list: A14 - A0
5dt4	DR_EMC_AH	3h	r	w	Signal list: A23 - A15
7dt6	DR_EMC_DL	3h	r	w	Signal list: D15 - D0, XBE0_DQM0, XBE1_XBE1
9dt8	DR_EMC_DH	3h	r	w	Signal list: D31 - D16, XBE2_DQM2, XBE3_XBE3
11dt10	DR_EMC_RW	3h	r	w	Signal list: XWR, XRD
13dt12	DR_EMC_PER	3h	r	w	Signal list: XCS_PER0 - 3
15dt14	DR_EMC_CLK_SDRAM	3h	r	w	Signal list: CLK_O_SDRAM0, CLK_O_SDRAM1, CLK_O_SDRAM2
17dt16	DR_EMC_SDRAM	3h	r	w	Signal list: XCS_SDRAM, XRAS_SDRAM, XCAS_SDRAM, XWE_SDRAM
19dt18	DR_EMC_CLK_BF	3h	r	w	Signal list: CLK_O_BF0 - 2
21dt20	DR_EMC_XAV_BF	3h	r	w	Signal list: XAV_BF
22	EMC_SDRAM_CLK_SD0SD1_EBL	1h	r	w	enable(=1) / disable(=0) CLK_O_SDRAM0 and CLK_O_SDRAM1 (an SDRAM block)
23	EMC_SDRAM_CLK_SD2_EBL	0h	r	w	enable(=1) / disable(=0) CLK_O_SDRAM2 (additionally 2nd SDRAM block)
24	EMC_BF_CLK_BF0BF1_EBL	0h	r	w	enable(=1) / disable(=0) CLK_O_BF0 and CLK_O_BF1 (one Flash block)
25	EMC_BF_CLK_BF2_EBL	0h	r	w	enable(=1) / disable(=0) CLK_O_BF2 (additionally 2nd Flash block)

Register:	DRIVE15_0GPIO			Address:	7Ch	
Bits:	31dt0	Reset value:	55555555h	Attributes:	r	w
Description:	SCRB Drive Current of GPIO Signals Every GPIO bit is set in accordance with the following coding on the drive current 00 - 4 mA 01 - 6 mA 10 - 8 mA 11 - 12mA					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	DR_GPIO0	1h	r	w	drive current GPIO0
3dt2	DR_GPIO1	1h	r	w	drive current GPIO1
5dt4	DR_GPIO2	1h	r	w	drive current GPIO2
7dt6	DR_GPIO3	1h	r	w	drive current GPIO3

9dt8	DR_GPIO4	1h	r	w	drive current GPIO4
11dt10	DR_GPIO5	1h	r	w	drive current GPIO5
13dt12	DR_GPIO6	1h	r	w	drive current GPIO6
15dt14	DR_GPIO7	1h	r	w	drive current GPIO7
17dt16	DR_GPIO8	1h	r	w	drive current GPIO8
19dt18	DR_GPIO9	1h	r	w	drive current GPIO9
21dt20	DR_GPIO10	1h	r	w	drive current GPIO10
23dt22	DR_GPIO11	1h	r	w	drive current GPIO11
25dt24	DR_GPIO12	1h	r	w	drive current GPIO12
27dt26	DR_GPIO13	1h	r	w	drive current GPIO13
29dt28	DR_GPIO14	1h	r	w	drive current GPIO14
31dt30	DR_GPIO15	1h	r	w	drive current GPIO15

Register:	DRIVE31_16GPIO			Address:	80h	
Bits:	31dt0	Reset value:	55555555h	Attributes:	r	w
Description:	SCRB Drive Current of GPIO Signals Every GPIO bit is set in accordance with the following coding on the drive current 00 - 4 mA 01 - 6 mA 10 - 8 mA 11 - 12mA					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	DR_GPIO16	1h	r	w	drive current GPIO16
3dt2	DR_GPIO17	1h	r	w	drive current GPIO17
5dt4	DR_GPIO18	1h	r	w	drive current GPIO18
7dt6	DR_GPIO19	1h	r	w	drive current GPIO19
9dt8	DR_GPIO20	1h	r	w	drive current GPIO20
11dt10	DR_GPIO21	1h	r	w	drive current GPIO21
13dt12	DR_GPIO22	1h	r	w	drive current GPIO22
15dt14	DR_GPIO23	1h	r	w	drive current GPIO23
17dt16	DR_GPIO24	1h	r	w	drive current GPIO24
19dt18	DR_GPIO25	1h	r	w	drive current GPIO25
21dt20	DR_GPIO26	1h	r	w	drive current GPIO26
23dt22	DR_GPIO27	1h	r	w	drive current GPIO27
25dt24	DR_GPIO28	1h	r	w	drive current GPIO28
27dt26	DR_GPIO29	1h	r	w	drive current GPIO29
29dt28	DR_GPIO30	1h	r	w	drive current GPIO30
31dt30	DR_GPIO31	1h	r	w	drive current GPIO31

Register:	DRIVE47_32GPIO			Address:	84h	
Bits:	31dt0	Reset value:	55555555h	Attributes:	r	w
Description:	SCRB Drive Current of GPIO Signals (3.3V / 1.8V) Every GPIO bit is set in accordance with the following coding on the drive current -----3.3V--------- -----1.8V-------- 00 - 6 mA 00 - 3 mA (n.a.) 01 - 9 mA 01 - 6 mA 10 - 18 mA (n.a.) 10 - 9 mA **recommended** 11 - 24mA (n.a.) 11 - 12 mA					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	DR_GPIO32	1h	r	w	drive current GPIO32
3dt2	DR_GPIO33	1h	r	w	drive current GPIO33
5dt4	DR_GPIO34	1h	r	w	drive current GPIO34
7dt6	DR_GPIO35	1h	r	w	drive current GPIO35
9dt8	DR_GPIO36	1h	r	w	drive current GPIO36
11dt10	DR_GPIO37	1h	r	w	drive current GPIO37
13dt12	DR_GPIO38	1h	r	w	drive current GPIO38
15dt14	DR_GPIO39	1h	r	w	drive current GPIO39
17dt16	DR_GPIO40	1h	r	w	drive current GPIO40
19dt18	DR_GPIO41	1h	r	w	drive current GPIO41
21dt20	DR_GPIO42	1h	r	w	drive current GPIO42
23dt22	DR_GPIO43	1h	r	w	drive current GPIO43
25dt24	DR_GPIO44	1h	r	w	drive current GPIO44
27dt26	DR_GPIO45	1h	r	w	drive current GPIO45
29dt28	DR_GPIO46	1h	r	w	drive current GPIO46
31dt30	DR_GPIO47	1h	r	w	drive current GPIO47

Register:	DRIVE63_48GPIO			Address:	88h	
Bits:	31dt0	Reset value:	55555555h	Attributes:	r	w
Description:	SCRB Drive Current of GPIO Signals (3.3V / 1.8V) Every GPIO bit is set in accordance with the following coding on the drive current -----3.3V--------- -----1.8V-------- 00 - 6 mA 00 - 3 mA (n.a.) 01 - 9 mA 01 - 6 mA 10 - 18 mA (n.a.) 10 - 9 mA **recommended** 11 - 24mA (n.a.) 11 - 12 mA					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	DR_GPIO48	1h	r	w	drive current GPIO48
3dt2	DR_GPIO49	1h	r	w	drive current GPIO49
5dt4	DR_GPIO50	1h	r	w	drive current GPIO50
7dt6	DR_GPIO51	1h	r	w	drive current GPIO51
9dt8	DR_GPIO52	1h	r	w	drive current GPIO52
11dt10	DR_GPIO53	1h	r	w	drive current GPIO53
13dt12	DR_GPIO54	1h	r	w	drive current GPIO54
15dt14	DR_GPIO55	1h	r	w	drive current GPIO55
17dt16	DR_GPIO56	1h	r	w	drive current GPIO56
19dt18	DR_GPIO57	1h	r	w	drive current GPIO57
21dt20	DR_GPIO58	1h	r	w	drive current GPIO58
23dt22	DR_GPIO59	1h	r	w	drive current GPIO59
25dt24	DR_GPIO60	1h	r	w	drive current GPIO60
27dt26	DR_GPIO61	1h	r	w	drive current GPIO61
29dt28	DR_GPIO62	1h	r	w	drive current GPIO62
31dt30	DR_GPIO63	1h	r	w	drive current GPIO63

Register:	DRIVE79_64GPIO			Address:	8Ch	
Bits:	31dt0	Reset value:	55555555h	Attributes:	r	w

Description:	SCRB Drive Current of GPIO Signals (3.3V / 1.8V) Every GPIO bit is set in accordance with the following coding on the drive current -----3.3V--------- -----1.8V-------- 00 - 6 mA 00 - 3 mA (n.a.) 01 - 9 mA 01 - 6 mA 10 - 18 mA (n.a.) 10 - 9 mA **recommended** 11 - 24mA (n.a.) 11 - 12 mA

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	DR_GPIO64	1h	r	w	drive current GPIO64
3dt2	DR_GPIO65	1h	r	w	drive current GPIO65
5dt4	DR_GPIO66	1h	r	w	drive current GPIO66
7dt6	DR_GPIO67	1h	r	w	drive current GPIO67
9dt8	DR_GPIO68	1h	r	w	drive current GPIO68
11dt10	DR_GPIO69	1h	r	w	drive current GPIO69
13dt12	DR_GPIO70	1h	r	w	drive current GPIO70
15dt14	DR_GPIO71	1h	r	w	drive current GPIO71
17dt16	DR_GPIO72	1h	r	w	drive current GPIO72
19dt18	DR_GPIO73	1h	r	w	drive current GPIO73
21dt20	DR_GPIO74	1h	r	w	drive current GPIO74
23dt22	DR_GPIO75	1h	r	w	drive current GPIO75
25dt24	DR_GPIO76	1h	r	w	drive current GPIO76
27dt26	DR_GPIO77	1h	r	w	drive current GPIO77
29dt28	DR_GPIO78	1h	r	w	drive current GPIO78
31dt30	DR_GPIO79	1h	r	w	drive current GPIO79

Register:	DRIVE95_80GPIO			Address:	90h	
Bits:	31dt0	Reset value:	55555555h	Attributes:	r	w

Description:	SCRB Drive Current of GPIO Signals (3.3V / 1.8V) Every GPIO bit is set in accordance with the following coding on the drive current -----3.3V--------- -----1.8V-------- 00 - 6 mA 00 - 3 mA (n.a.) 01 - 9 mA 01 - 6 mA 10 - 18 mA (n.a.) 10 - 9 mA **recommended** 11 - 24mA (n.a.) 11 - 12 mA

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	DR_GPIO80	1h	r	w	drive current GPIO80
3dt2	DR_GPIO81	1h	r	w	drive current GPIO81
5dt4	DR_GPIO82	1h	r	w	drive current GPIO82
7dt6	DR_GPIO83	1h	r	w	drive current GPIO83
9dt8	DR_GPIO84	1h	r	w	drive current GPIO84
11dt10	DR_GPIO85	1h	r	w	drive current GPIO85
13dt12	DR_GPIO86	1h	r	w	drive current GPIO86
15dt14	DR_GPIO87	1h	r	w	drive current GPIO87
17dt16	DR_GPIO88	1h	r	w	drive current GPIO88
19dt18	DR_GPIO89	1h	r	w	drive current GPIO89
21dt20	DR_GPIO90	1h	r	w	drive current GPIO90
23dt22	DR_GPIO91	1h	r	w	drive current GPIO91
25dt24	DR_GPIO92	1h	r	w	drive current GPIO92

27dt26	DR_GPIO93	1h	r	w	drive current GPIO93
29dt28	DR_GPIO94	1h	r	w	drive current GPIO94
31dt30	DR_GPIO95	1h	r	w	drive current GPIO95

Register:	PULL15_0GPIO			Address:	94h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	SCRB Pull GPIO Signals Every GPIO bit is set in accordance with the following coding on the pull resistances 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistance is deactivated automatically Reset value is not required, because register is loaded in accordance with the function set via 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
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	Attributes:	rh	w
Description:	SCRB Pull GPIO Signals Every GPIO bit is set in accordance with the following coding on the pull resistances 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistance is deactivated automatically Reset value is not required, because register is loaded in accordance with the function set via 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
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	Attributes:	rh	w
Description:	SCRB Pull GPIO Signals Every GPIO bit is set in accordance with the following coding on the pull resistances 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistance is deactivated automatically Reset value is not required, because register is loaded in accordance with the function set via 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	Attributes:	rh	w
Description:	SCRB Pull GPIO Signals Every GPIO bit is set in accordance with the following coding on the pull resistances					

	00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistance is deactivated automatically Reset value is not required, because register is loaded in accordance with the function set via 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	Attributes:	rh	w
Description:	SCRB Pull GPIO Signals Every GPIO bit is set in accordance with the following coding on the pull resistances 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistance is deactivated automatically Reset value is not required, because register is loaded in accordance with the function set via 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 Every GPIO bit is set in accordance with the following coding on the pull resistances 00 - highZ 01 - Pull-up 10 - highZ 11 - Pull-down If the PAD is switched to output, the pull resistance is deactivated automatically Reset value is not required, because register is loaded in accordance with the function set via 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:	XHIF_MODE			Address:	B8h	
Bits:	0dt0	Reset value:	0h	Attributes:	r	w
Description:	Umschalten XHIF-Pin zwischen XHIF_A20 und XHIF_XCS_R					

Bit	Identifier	Reset	Attr.		Function / Description
0	XHIF_MODE	0h	r	w	The input pin XHIF_XCS_R_A20 is used as 0: Page Register Chipselect (XHIF_XCS_R) 1: Address line (XHIF_A20). (The non-selected XHIF input is switched inactive, i.e. XHIF_A20 ='0', XHIF_XCS_R ='1')

5.3.9 UART1-4 register

Base addresses see Chapter 5.2.

Address space:

Start_Address	End_Address	Modul/Memory_Name	Interface	Fill_Mode
0h	FFCh	UART_PL011	APB	<fillmode>

Module	Register/Memory	Read	Write	Address	Revision
/UART_PL011					
	UARTDR	r(h)	(w)	0h	
	UARTRSR_UARTECR	r(h)		4h	
	UARTFR	r(h)		18h	
	USRTILPR	r	(w)	20h	
	UARTIBRD	r	(w)	24h	
	UARTFBRD	r	(w)	28h	
	UARTLCR_H	r	(w)	2Ch	
	UARTCR	r	(w)	30h	
	UARTIFLS	r	(w)	34h	
	UARTIMSC	r	(w)	38h	
	UARTRIS	r(h)		3Ch	
	UARTMIS	r(h)		40h	
	UARTICR	(r)	(w)	44h	
	UARTDMACR	r	(w)	48h	
	UARTPeriphID0	r		FE0h	
	UARTPeriphID1	r		FE4h	
	UARTPeriphID2	r		FE8h	
	UARTPeriphID3	r		FECh	
	UARTPCellID0	r		FF0h	
	UARTPCellID1	r		FF4h	
	UARTPCellID2	r		FF8h	
	UARTPCellID3	r		FFCh	

Register description:

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

Module: /UART_PL011

Register:	UARTDR				Address:	0h	
Bits:	31dt0	Reset value:	0h		Attributes:	r(h)	(w)
Description:	UARTDR is the data register Receive (read) data character Transmit (write) data character						

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	Data	00h	rh	w	Receive (read) data character. Transmit (write) data character.
8	Framing_Error	0h	rh		Framing error. When this bit is set to 1, it indicates that the

					received character did not have a valid stop bit (a valid stop bit is 1). In FIFO mode, this error is associated with the character at the top of the FIFO.
9	Parity_Error	0h	rh		Parity error. When this bit is set to 1, it indicates that the parity of the received data character does not match the parity selected as defined by bits 2 and 7 of the UARTLCR_H register. In FIFO mode, this error is associated with the character at the top of the FIFO.
10	Break_Error	0h	rh		Break error. This bit is set to 1 if a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time (defined as start, data, parity and stop bits). In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state), and the next valid start bit is received.
11	Overrun_Error	0h	rh		Overrun error. This bit is set to 1 if data is received and the receive FIFO is already full. This is cleared to 0 once there is an empty space in the FIFO and a new character can be written to it.
31dt12	<unused>	00000h	r		

Register:	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	<unused>	0000000h	r		Reserved, unpredictable when read.

Register:	UARTFR			Address:	18h	
Bits:	31dt0	Reset value:	90h	Attributes:	r(h)	
Description:	flag register; after reset TXFF, RXFF and BUSY are 0, TXFE and RXFE are 1.					

Bit	Identifier	Reset	Attr.		Function / Description
0	Clear_To_Send	0h	rh		(CTS) This bit is the complement of the PrimeCell UART clear to send (nUARTCTS) modem status input. That is, the bit is 1 when the modem status input is 0.
1	Data_Set_Ready	0h	rh		(DSR) This bit is the complement of the PrimeCell UART data set ready (nUARTDSR) modem status input. That is, the bit is 1 when the modem status input is 0.
2	Data_Carrier_Detect	0h	rh		(DCD) This bit is the complement of the PrimeCell UART data carrier detect (nUARTTDCD) modem status input. That

					is, the bit is 1 when the modem status input is 0.
3	UART_Busy	0h	rh		(BUSY) If this bit is set to 1, the PrimeCell UART is busy transmitting data. This bit remain set until the complete byte, including all the stop bits, has been sent from the shift register. This bit is set as soon as the transmit FIFO becomes non-empty.
4	Receive_FIFO_Empty	1h	rh		(RXFE) The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the receive holding register is empty. If the FIFO is enabled, the RXFE bit is set when the receive FIFO is empty.
5	Transmit_FIFO_Full	0h	rh		(TXFF) The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the transmit holding register is full. If the FIFO is enabled, the TXFF bit is set when the transmit FIFO is full.
6	Receive_FIFO_Full	0h	rh		(RXFF) The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the receive holding register is full. If the FIFO is enabled, the RXFF bit is set when the receive FIFO is full.
7	Transmit_FIFO_Empty	1h	rh		(TXFE) The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the transmit holding register is empty. If the FIFO is enabled, the TXFE bit is set when the transmit FIFO is empty.
8	Ring_Indicator	0h	rh		Ring indicator. This bit is the complement of the UART ring indicator (nUARTRI) modem status input. That is, the bit is 1 when the modem status input is 0.
31dt9	<unused>	000000h	r		Reserved, read as zero.

Register:	USRTILPR				Address:	20h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	8-bit-low-power divisor value. These bits are cleared to 0 at reset.						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	USRTILPR	00h	r	w	8-bit low-power divisor value.		
31dt8	<unused>	000000h	r		Reserved, read as zero.		

Register:	UARTIBRD				Address:	24h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Integer Part of the Baud Rate Divisor Value Register						
Bit	Identifier	Reset	Attr.		Function / Description		
15dt0	UARTIBRD	0000h	r	w	The integer baud rate divisor.		
31dt16	<unused>	0000h	r		Reserved, read as zero.		

Register:	UARTFBRD				Address:	28h	
Bits:	31dt0	Reset value:		0h	Attributes:	r	(w)
Description:	Fractional Part of the Baud Rate Divisor Value Register						
Bit	Identifier	Reset	Attr.		Function / Description		
5dt0	UARTFBRD	00h	r	w	The fractional baud rate divisor.		
31dt6	<unused>	0000000h	r		Reserved, do not modify, read as zero.		

Register:	UARTLCR_H			Address:	2Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	line control register, high byte					

Bit	Identifier	Reset	Attr.		Function / Description
0	Send_Break	0h	r	w	If this bit is set to 1, a low level is 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.
6dt5	Word_lenght	0h	r	w	Word length. The select bits indicate the number of data bits transmitted or received in a frame as follows: 11 = 8 bits 10 = 7 bits 01 = 6 bits 00 = 5 bits.
7	Stick_Parity_Select	0h	r	w	Stick parity select. When bits 1, 2, and 7 of the UARTLCR_H register are set, the parity bit is transmitted and checked as a 0. When bits 1 and 7 are set, and bit 2 is 0, the parity bit is transmitted and checked as a 1. When this bit is cleared stick parity is disabled.
31dt8	<unused>	000000h	r		Reserved, do not modify, read as zero.

Register:	UARTCR			Address:	30h	
Bits:	31dt0	Reset value:	300h	Attributes:	r	(w)
Description:	Control register; all the bits are cleared to 0 on reset.					

Bit	Identifier	Reset	Attr.		Function / Description
0	UART_Enable	0h	r	w	If this bit is set to 1, the PrimeCell UART is enabled.
1	SIR_Enable	0h	r	w	If this bit is set to 1, the IrDA SIR 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 an width of 3/16th of the bit period. If this bit is set to 1, low level bits are transmitted with a pulse width which is 3 times the period of the IrLPBaud16 input signal, regardless of the selected bit rate.Setting this bit uses less power, but may reduce transmission distances.
6dt3	<unused>	0h	r	w	Reserved, do not modify, read as zero.
7	Loop_Back_Enable	0h	r	w	If this bit is set to 1, the SIR Enable is set to 1, and the test register UARTTMR bit 1 (SIRTEST) is set to 1, the nSIROUT path is inverted, and fed through to the SIRIN path. This bit is cleared to 0 on reset, which disables the loopback mode.

8	Transmit_Enable	1h	r	w	If this bit is set to 1, the transmit section is enabled.
9	Receive_Enable	1h	r	w	If this bit is set to 1, the transmit section is enabled.
10	DTR	0h	r	w	If this bit is set to 1, the output DTR in 0.
11	RTS	0h	r	w	If this bit is set to 1, the output RTS in 0.
12	Out1	0h	r	w	If this bit is set to 1, the output OUT1 in 0.
13	Out2	0h	r	w	If this bit is set to 1, the output OUT2 in 0.
14	RTS_Enable	0h	r	w	If 1, RTS hardware flow is enabled.
15	CTS_Enable	0h	r	w	If 1, CTR hardware flow is enabled.
31dt16	<unused>	0h	r	w	Reserved, do not modify, read as zero.

Register:	UARTIFLS			Address:	34h	
Bits:	31dt0	Reset value:	12h	Attributes:	r	(w)
Description:	Interrupt FIFO Level Select Register					

Bit	Identifier	Reset	Attr.		Function / Description
2dt0	Tx_Int_FIFO_Level	2h	r	w	The trigger points for the transmit interrupt: 000 = RX FIFO becomes < = 1/8 full 001 = RX FIFO becomes < = 1/4 full 010 = RX FIFO becomes < = 1/2 full 011 = RX FIFO becomes < = 3/4 full 100 = RX FIFO becomes < = 7/8 full 101:111 = reserved
5dt3	Rx_Int_FIFO_Level	2h	r	w	The trigger points for the receive interrupt: 000 = RX FIFO becomes > = 1/8 full 001 = RX FIFO becomes > = 1/4 full 010 = RX FIFO becomes > = 1/2 full 011 = RX FIFO becomes > = 3/4 full 100 = RX FIFO becomes > = 7/8 full 101:111 = reserved
31dt6	<unused>	0000000h	r		Reserved, do not modify, read as zero.

Register:	UARTIMSC			Address:	38h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	Interrupt Mask Set/Clear Register On a write of 1, the mask of the interrupt is set. A write of 0 clears the mask.					

Bit	Identifier	Reset	Attr.		Function / Description
0	RI_Modem_Int_Mask	0h	r	w	UARTRI modem interrupt mask. On a read the current mask for the RIMIM interrupt is returned. On a write of 1, the mask of the RIMIM interrupt is set. A write of 0 clears the mask.
1	CTS_Modem_Int_Mask	0h	r	w	UARTCTS modem interrupt mask. On a read the current mask for the CTSMIM interrupt is returned. On a write of 1, the mask of the CTSMIM interrupt is set. A write of 0 clears the mask.
2	DCD_Modem_Int_Mask	0h	r	w	UARTDCD modem interrupt mask. On a read the current mask for the DCDMIM interrupt is returned. On a write of 1, the mask of the DCDMIM interrupt is set. A write of 0 clears the mask.
3	DSR_Modem_Int_Mask	0h	r	w	UARTDSR modem interrupt mask. On a read the current mask for the DSRMIM interrupt is returned. On a write of 1, the mask of the DSRMIM interrupt is set. A write of 0 clears the mask
4	Rx_Int_Mask	0h	r	w	Receive interrupt mask. On a read the current mask for the RXIM interrupt is returned. On a write of 1, the mask of the RXIM interrupt is set. A

					write of 0 clears the mask.
5	Tx_Int_Mask	0h	r	w	Transmit interrupt mask. On a read the current mask for the TXIM interrupt is returned. On a write of 1, the mask of the TXIM interrupt is set. A write of 0 clears the mask
6	Receive_Timout_Mask	0h	r	w	Receive timeout interrupt mask. On a read the current mask for the RTIM interrupt is returned. On a write of 1, the mask of the RTIM interrupt is set. A write of 0 clears the mask
7	Framing_Error_Mask	0h	r	w	Framing error interrupt mask. On a read the current mask for the FEIM interrupt is returned. On a write of 1, the mask of the FEIM interrupt is set. A write of 0 clears the mask.
8	Parity_Error_Mask	0h	r	w	Parity error interrupt mask. On a read the current mask for the PEIM interrupt is returned. On a write of 1, the mask of the PEIM interrupt is set. A write of 0 clears the mask.
9	Break_Error_Mask	0h	r	w	Break error interrupt mask. On a read the current mask for the BEIM interrupt is returned. On a write of 1, the mask of the BEIM interrupt is set. A write of 0 clears the mask
10	Overrun_Error_Mask	0h	r	w	Overrun error interrupt mask. On a read, the current mask for the OEIM interrupt is returned. On a write of 1, the mask of the OEIM interrupt is set. A write of 0 clears the mask.
31dt11	<unused>	000000h	r		Reserved, do not modify, read as zero.

Register:	UARTRIS			Address:	3Ch	
Bits:	31dt0	Reset value:	0h	Attributes:	r(h)	
Description:	Raw Interrupt Status Register Gives the raw interrupt state (prior to masking) of the interrupt					

Bit	Identifier	Reset	Attr.		Function / Description
0	RI_Modem_Int_Status	0h	rh		UARTRI modem interrupt status. Gives the raw interrupt state (prior to masking) of the UARTRIINTR interrupt
1	CTS_Modem_Int_Status	0h	rh		UARTCTS modem interrupt status. Gives the raw interrupt state (prior to masking) of the UARTCTSINTR interrupt
2	DCD_Modem_Int_Status	0h	rh		UARTDCD modem interrupt status. Gives the raw interrupt state (prior to masking) of the UARTDCDINTR interrupt
3	DSR_Modem_Int_Status	0h	rh		UARTDSR modem interrupt status. Gives the raw interrupt state (prior to masking) of the UARTDSRINTR interrupt
4	Rx_Int_Status	0h	rh		Receive interrupt status. Gives the raw interrupt state (prior to masking) of the UARTRXINTR interrupt
5	Tx_Int_Status	0h	rh		Transmit interrupt status. Gives the raw interrupt state (prior to masking) of the UARTTXINTR interrupt
6	Rx_Timout_Int_Status	0h	rh		Receive timeout interrupt status. Gives the raw interrupt state (prior to masking) of the UARTRTINTR interrupta
7	Framing_Error_Int_Status	0h	rh		Framing error interrupt status. Gives the raw interrupt state (prior to masking) of the UARTFEINTR interrupt
8	Parity_Error_Int_Status	0h	rh		Parity error interrupt status. Gives the raw interrupt state (prior to masking) of the UARTPEINTR interrupt
9	Break_Err_Int_Status	0h	rh		Break error interrupt status. Gives the raw interrupt state (prior to masking) of the UARTBEINTR interrupt
10	Overrun_Err_Int_Status	0h	rh		Overrun error interrupt status. Gives the raw interrupt state (prior to masking) of the UARTOEINTR interrupt
31dt11	<unused>	000000h	r		Reserved, do not modify, read as zero.

Register:	UARTMIS	Address:	40h

Bits:	31dt0	Reset value:	0h	Attributes:	r(h)	
Description:	Mask Interrupt Status Register Gives the masked interrupt state (after masking) of the interrupt					

Bit	Identifier	Reset	Attr.		Function / Description
0	RIMMIS	0h	rh		...RI Intr.
1	CTSMMIS	0h	rh		...CTS Intr.
2	DCDMMIS	0h	rh		...DCD Intr.
3	DSRMMIS	0h	rh		...DSR Intr.
4	RXMIS	0h	rh		...Receive Interrupt
5	TXMIS	0h	rh		...Transmit Interrupt
6	RTMIS	0h	rh		...Receive Timeout
7	FEMIS	0h	rh		...Framing Error
8	PEMIS	0h	rh		...Parity Error
9	BEMIS	0h	rh		...Break Error
10	OEMIS	0h	rh		...Overrun Error
31dt11	<unused>	000000h	r		Reserved, do not modify, read as zero.

Register:	UARTICR			Address:	44h	
Bits:	31dt0	Reset value:	0h	Attributes:	(r)	(w)
Description:	Interrupt Clear Register Clears the Interrupt ...					

Bit	Identifier	Reset	Attr.		Function / Description
0	RIMIC	0h		w	...RI Intr.
1	CTSMIC	0h		w	...CTS Intr.
2	DCDMIC	0h		w	...DCD Intr.
3	DSRMIC	0h		w	...DSR Intr.
4	RXIC	0h		w	...Receive Interrupt
5	TXIC	0h		w	...Transmit Interrupt
6	RTIC	0h		w	...Receive Timeout
7	FEIC	0h		w	...Framing Error
8	PEIC	0h		w	...Parity Error
9	BEIC	0h		w	...Break Error
10	OEIC	0h		w	...Overrun Error
31dt11	<unused>	000000h	r		Reserved, do not modify, read as zero.

Register:	UARTDMACR			Address:	48h	
Bits:	31dt0	Reset value:	0h	Attributes:	r	(w)
Description:	DMA Control Register					

Bit	Identifier	Reset	Attr.		Function / Description
0	RXDMAE	0h	r	w	If 1, DMA for the receive FIFO is enabled.
1	TXDMAE	0h	r	w	If 1, DMA for the transmit FIFO is enabled.
2	DMAONERR	0h	r	w	If 1, DMA request outputs are disabled when UART error interrupt is asserted.
31dt3	<unused>	00000000h	r		Reserved, do not modify, read as zero.

Register:	UARTPeriphID0	Address:	FE0h

Bits:	31dt0	Reset value:	11h	Attributes:	r	
Description:	Peripheral ID0 Register, hard coded					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	UARTPeriphID0	11h	r		These bits read back as 0x11
31dt8	<unused>	000000h	r		Reserved, read undefined must read as zeros

Register:	UARTPeriphID1			Address:	FE4h	
Bits:	31dt0	Reset value:	10h	Attributes:	r	
Description:	Peripheral ID1 Register, hard coded					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	UARTPeriphID1	10h	r		These bits read back as 0x10
31dt8	<unused>	000000h	r		Reserved, read undefined must read as zeros

Register:	UARTPeriphID2			Address:	FE8h	
Bits:	31dt0	Reset value:	24h	Attributes:	r	
Description:	Peripheral ID2 Register, hard coded					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	UARTPeriphID2	34h	r		These bits read back as 0x34
31dt8	<unused>	000000h	r		Reserved, read undefined must read as zeros

Register:	UARTPeriphID3			Address:	FECh	
Bits:	31dt0	Reset value:	0h	Attributes:	r	
Description:	Peripheral ID3 Register, hard coded					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	UARTPeriphID3	00h	r		These bits read back as 0x00
31dt8	<unused>	000000h	r		Reserved, read undefined must read as zeros

Register:	UARTPCellID0			Address:	FF0h	
Bits:	31dt0	Reset value:	Dh	Attributes:	r	
Description:	PrimeCell ID0 Register, hard coded					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	UARTPCellID0	0Dh	r		These bits read back as 0x0D
31dt8	<unused>	000000h	r		Reserved, read undefined must read as zeros

Register:	UARTPCellID1			Address:	FF4h	
Bits:	31dt0	Reset value:	F0h	Attributes:	r	
Description:	PrimeCell ID1 Register, hard coded					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	UARTPCellID1	F0h	r		These bits read back as 0xF0
31dt8	<unused>	000000h	r		Reserved, read undefined must read as zeros

Register:	UARTPCellID2			Address:	FF8h	
Bits:	31dt0	Reset value:	5h	Attributes:	r	
Description:	PrimeCell ID2 Register, hard coded					

Bit	Identifier	Reset	Attr.	Function / Description

7dt0	UARTPCellID2	05h	r		These bits read back as 0x05
31dt8	<unused>	000000h	r		Reserved, read undefined must read as zeros

Register:	UARTPCellID3			Address:	FFCh	
Bits:	31dt0	Reset value:	B1h	Attributes:	r	
Description:	PrimeCell ID3 Register, hard coded					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	UARTPCellID3	B1h	r		These bits read back as 0xB1
31dt8	<unused>	000000h	r		Reserved, read undefined must read as zeros

5.3.10 SPI11-2 register

Base addresses see Chapter 5.2.

Address space:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	14h	SPI	APB

Module	Regis-ter/Memory	Read	Write	Address
/SPI				
	SSPCR0	r	w	0h
	SSPCR1	r	w	4h
	SSPDR	rh	w	8h
	SSPSR	r		Ch
	SSPCPSR	r	w	10h
	SSPIIR_SSPICR	rh		14h

Register description:

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

Module: /SPI

Register:	SSPCR0			Address:	0h	
Bits:	15dt0	Reset value:	0h	Attributes:	r	w
Description:	SSPCR0 is control register 0 and contains five bit fields that control various functions within the PrimeCell SSPMS.					

Bit	Identifier	Reset	Attr.		Function / Description
3dt0	DSS	0h	r	w	Data size select: 0000 Reserved, undefined operation 0001 Reserved, undefined operation 0010 Reserved, undefined operation 0011 4-bit data 0100 5-bit data 0101 6-bit data 0110 7-bit data 0111 8-bit data 1000 9-bit data 1001 10-bit data 1010 11-bit data 1011 12-bit data 1100 13-bit data 1101 14-bit data 1110 15-bit data 1111 16-bit data
5dt4	FRF	0h	r	w	Frame format: 00 Motorola SPI frame format 01 TI synchronous serial frame format 10 National Mikrowire frame format 11 Reserved, undefined operation
6	SPO	0h	r	w	SCLKOUT polarity (applicable to Motorola SPI frame format only).

7	SPH	0h	r	w	SCLKOUT phase (applicable to Motorola SPI frame format only).
15dt8	SCR	00h	r	w	Serial Clock rate. The value SCR s used to generate the transmit and receive bit range of the PrimeCell SSPMS.

Register:	SSPCR1				Address:	4h	
Bits:	6dt0	Reset value:		0h	Attributes:	r	w
Description:	SSPCR1 is the control register 1 and contains five different bit fields, which control various functions within the PrimeCell SSPMS.						
Bit	Identifier	Reset	Attr.		Function / Description		
0	RIE	0h	r	w	Receive FIFO interrupt enable: 0 =Receive FIFO half-full or more condition does not generate the SSPRXINTR interrupt. 1 =Receive FIFO half-full or more condition generates the SSPRXINTR interrupt.		
1	TIE	0h	r	w	Transmit FIFO interrupt enable: 0 =Transmit FIFO half-full or less condition does not generate the SSPTXINTR interrupt. 1 =Transmit FIFO half-full or less condition generates the SSPTXINTR interrupt.		
2	RORIE	0h	r	w	Receive FIFO overrun interrupt enable: 0 =Overrun detection is disabled. Overrun condition does not generate the SSPRORINTR interrupt. Clearing 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.		

Register:	SSPDR				Address:	8h	
Bits:	15dt0	Reset value:		0000h	Attributes:	rh	w
Description:	Transmit/Receive FIFO						

Register:	SSPSR				Address:	Ch	
Bits:	4dt0	Reset value:		3h	Attributes:	r	
Description:	SSPSR is a read-only status register wich contains bits that indicate the FIFO fill status and the SSP busy status.						
Bit	Identifier	Reset	Attr.		Function / Description		

0	TFE	1h	r		Transmit FIFO empty
1	TNF	1h	r		Transmit FIFO not full
2	RNE	0h	r		Receive FIFO not empty
3	RFF	0h	r		Receive FIFO full
4	BSY	0h	r		SSP busy flag

Register:	SSPCPSR			Address:	10h	
Bits:	7dt0	Reset value:	00h	Attributes:	r	w
Description:	Clock prescale divisor					

Register:	SSPIIR_SSPICR			Address:	14h	
Bits:	2dt0	Reset value:	0h	Attributes:	rh	
Description:	The interrupt status is read from the SSP interrupt identification register (SSPIIR)					

Bit	Identifier	Reset	Attr.		Function / Description
0	RIS	0h	rh		Read: SSP transmit FIFO service request interrupt status
1	TIS	0h	rh		Read: SSP Receive FIFO service request interrupt status
2	RORIS	0h	rh		Read: SSP Receive FIFO overrun interrupt status

5.3.11 Timer 0-5 Register

Base address see Chapter 5.2.

Address space:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	D4h	timer_top	APB

Module	Register/Memory	Read	Write	Address
/timer_top				
	TIM_0_MODE_REG	(r)	(w)	0h
	TIM_0_PRESCALER_REG	(r)	(w)	4h
	TIM_0_LOAD_REG	r	w	8h
	TIM_0_COUNT_REG	r		Ch
	TIM_0_INT_EV_REG	r	w	10h
	TIM_0_EXT_EV_1_REG	r	w	14h
	TIM_0_EXT_EV_2_REG	r	w	18h
	TIM_1_MODE_REG	(r)	(w)	20h
	TIM_1_PRESCALER_REG	(r)	(w)	24h
	TIM_1_LOAD_REG	r	w	28h
	TIM_1_COUNT_REG	r		2Ch
	TIM_1_INT_EV_REG	r	w	30h
	TIM_1_EXT_EV_1_REG	r	w	34h
	TIM_1_EXT_EV_2_REG	r	w	38h
	TIM_2_MODE_REG	(r)	(w)	40h
	TIM_2_PRESCALER_REG	(r)	(w)	44h
	TIM_2_LOAD_REG	r	w	48h
	TIM_2_COUNT_REG	r		4Ch
	TIM_2_INT_EV_REG	r	w	50h
	TIM_2_EXT_EV_1_REG	r	w	54h
	TIM_2_EXT_EV_2_REG	r	w	58h
	TIM_3_MODE_REG	(r)	(w)	60h
	TIM_3_PRESCALER_REG	(r)	(w)	64h
	TIM_3_LOAD_REG	r	w	68h
	TIM_3_COUNT_REG	r		6Ch
	TIM_3_INT_EV_REG	r	w	70h
	TIM_3_EXT_EV_1_REG	r	w	74h
	TIM_3_EXT_EV_2_REG	r	w	78h
	TIM_4_MODE_REG	(r)	(w)	80h
	TIM_4_PRESCALER_REG	(r)	(w)	84h
	TIM_4_LOAD_REG	r	w	88h
	TIM_4_COUNT_REG	r		8Ch
	TIM_4_INT_EV_REG	r	w	90h
	TIM_4_EXT_EV_1_REG	r	w	94h
	TIM_4_EXT_EV_2_REG	r	w	98h
	TIM_5_MODE_REG	(r)	(w)	A0h
	TIM_5_PRESCALER_REG	(r)	(w)	A4h
	TIM_5_LOAD_REG	r	w	A8h
	TIM_5_COUNT_REG	r		ACh
	TIM_5_INT_EV_REG	r	w	B0h

	TIM_5_EXT_EV_1_REG	r	w	B4h
	TIM_5_EXT_EV_2_REG	r	w	B8h
	GATE_TRIG_CONTROL_REG	(r)	(w)	C0h
	CLOCK_DIVIDER_REG	(r)	(w)	C4h
	EXT_GATE_TRIG_MUX_REG	(r)	(w)	C8h
	EXT_EV_1_MUX_REG	(r)	(w)	CCh
	EXT_EV_2_MUX_REG	(r)	(w)	D0h
	SW_EVENT_TRIGGER_REG		(w)	D4h

Register allocation:

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

Module: /timer_top

Register:	TIM_0_MODE_REG				Address:	0h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)
Description:	Timer Mode Register for Timer 0						

Bit	Identifier	Reset	Attr.		Function / Description
0	INIT_BIT	0h		w	Initialization (0 =INIT_BIT not active, 1 =INIT_BIT active)
3dt1	<reserved>	0h			not used
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0 =CLK_TIMT, 1 =ext. Gate-/Triggersignal, see also bit GATE_EFFECT)
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0 =Reload Mode is active, 1 =Single Mode is active
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when 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>	0h			not used

Register:	TIM_0_PRESCALER_REG				Address:	4h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)
Description:	Timer PRESCALER Register for Timer 0						

Bit	Identifier	Reset	Attr.	Function / Description

7dt0	PRESCALER_VALUE	00h	r	w	Prescaler Value
31dt8	<reserved>	000000h			not used

Register:	TIM_0_LOAD_REG			Address:	8h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Timer LOAD/RELOAD Register for Timer 0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	LOAD_RELOAD_VALUE	00000000h	r	w	Load/Reload value

Register:	TIM_0_COUNT_REG			Address:	Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	
Description:	Timer COUNT Register for Timer 0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	COUNTER_VALUE	00000000h	r		Counter Value

Register:	TIM_0_INT_EV_REG			Address:	10h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Timer INT_EVENT Register for Timer 0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	INT_EVENT_VALUE	00000000h	r	w	Internal Event Value

Register:	TIM_0_EXT_EV_1_REG			Address:	14h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Timer EXT_EVENT_2 Register for Timer 0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	EXT_EVENT_1_VALUE	00000000h	r	w	External Event 1 Value

Register:	TIM_0_EXT_EV_2_REG			Address:	18h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Timer EXT_EVENT_1 Register for Timer 0					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	EXT_EVENT_2_VALUE	00000000h	r	w	External Event 2 Value

Register:	TIM_1_MODE_REG			Address:	20h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)
Description:	Timer Mode Register for Timer 1					

Bit	Identifier	Reset	Attr.		Function / Description
0	INIT_BIT	0h		w	Initialization (0 =INIT_BIT not active, 1 =INIT_BIT active)
3dt1	<reserved>	0h			not used
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0 =CLK_TIMT, 1 =ext. Gate-/Triggersignal, see also bit GATE_EFFECT)
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0 =Reload Mode is active, 1 =Single Mode is active

6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when 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>	0h			not used

Register:	TIM_1_PRESCALER_REG				Address:	24h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)
Description:	Timer PRESCALER Register for Timer 1						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE	00h	r	w	Prescaler Value		
31dt8	<reserved>	000000h			not used		

Register:	TIM_1_LOAD_REG				Address:	28h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	w
Description:	Timer LOAD/RELOAD Register for Timer 1						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	LOAD_RELOAD_VALUE	00000000h	r	w	Load/Reload value		

Register:	TIM_1_COUNT_REG				Address:	2Ch	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	
Description:	Timer COUNT Register for Timer 1						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	COUNTER_VALUE	00000000h	r		Counter Value		

Register:	TIM_1_INT_EV_REG				Address:	30h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	w
Description:	Timer INT_EVENT Register for Timer 1						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	INT_EVENT_VALUE	00000000h	r	w	Internal Event Value		

Register:	TIM_1_EXT_EV_1_REG				Address:	34h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	w
Description:	Timer EXT_EVENT_1 Register for Timer 1						

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	EXT_EVENT_1_VALUE	00000000h	r	w	External Event 1 Value

Register:	TIM_1_EXT_EV_2_REG			Address:	38h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Timer EXT_EVENT_2 Register for Timer 1					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	EXT_EVENT_2_VALUE	00000000h	r	w	External Event 2 Value

Register:	TIM_2_MODE_REG			Address:	40h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)
Description:	Timer Mode Register for Timer 2					

Bit	Identifier	Reset	Attr.		Function / Description
0	INIT_BIT	0h		w	Initialization (0 =INIT_BIT not active, 1 =INIT_BIT active)
3dt1	<reserved>	0h			not used
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0 =CLK_TIMT, 1 =ext. Gate-/Triggersignal, see also bit GATE_EFFECT)
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0 =Reload Mode is active, 1 =Single Mode is active
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when 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>	0h			not used

Register:	TIM_2_PRESCALER_REG			Address:	44h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)
Description:	Timer PRESCALER Register for Timer 2					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	PRESCALER_VALUE	00h	r	w	Prescaler Value
31dt8	<reserved>	000000h			not used

Register:	TIM_2_LOAD_REG			Address:	48h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w

Description:	Timer LOAD/RELOAD Register for Timer 2				
Bit	Identifier	Reset	Attr.		Function / Description
31dt0	LOAD_RELOAD_VALUE	00000000h	r	w	Load/Reload value

Register:	TIM_2_COUNT_REG			Address:	4Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	
Description:	Timer COUNT Register for Timer 2					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	COUNTER_VALUE	00000000h	r		Counter Value	

Register:	TIM_2_INT_EV_REG			Address:	50h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Timer INT_EVENT Register for Timer 2					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	INT_EVENT_VALUE	00000000h	r	w	Internal Event Value	

Register:	TIM_2_EXT_EV_1_REG			Address:	54h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Timer EXT_EVENT_1 Register for Timer 2					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	EXT_EVENT_1_VALUE	00000000h	r	w	External Event 1 Value	

Register:	TIM_2_EXT_EV_2_REG			Address:	58h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Timer EXT_EVENT_2 Register for Timer 2					
Bit	Identifier	Reset	Attr.		Function / Description	
31dt0	EXT_EVENT_2_VALUE	00000000h	r	w	External Event 2 Value	

Register:	TIM_3_MODE_REG			Address:	60h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)
Description:	Timer Mode Register for Timer 3					
Bit	Identifier	Reset	Attr.		Function / Description	
0	INIT_BIT	0h		w	Initialization (0 =INIT_BIT not active, 1 =INIT_BIT active)	
3dt1	<reserved>	0h			not used	
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0 =CLK_TIMT, 1 =ext. Gate-/Triggersignal, see also bit GATE_EFFECT)	
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0 =Reload Mode is active, 1 =Single Mode is active	
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when 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>	0h			not used

Register:	TIM_3_PRESCALER_REG				Address:	64h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	(r)	(w)
Description:	Timer PRESCALER Register for Timer 3						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE	00h	r	w	Prescaler Value		
31dt8	<reserved>	000000h			not used		

Register:	TIM_3_LOAD_REG				Address:	68h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	r	w
Description:	Timer LOAD/RELOAD Register for Timer 3						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	LOAD_RELOAD_VALUE	00000000h	r	w	Load/Reload value		

Register:	TIM_3_COUNT_REG				Address:	6Ch	
Bits:	31dt0	Reset value:	00000000h		Attributes:	r	
Description:	Timer COUNT Register for Timer 3						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	COUNTER_VALUE	00000000h	r		Counter Value		

Register:	TIM_3_INT_EV_REG				Address:	70h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	r	w
Description:	Timer INT_EVENT Register for Timer 3						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	INT_EVENT_VALUE	00000000h	r	w	Internal Event Value		

Register:	TIM_3_EXT_EV_1_REG				Address:	74h	
Bits:	31dt0	Reset value:	00000000h		Attributes:	r	w
Description:	Timer EXT_EVENT_1 Register for Timer 3						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_1_VALUE	00000000h	r	w	External Event 1 Value		

Register:	TIM_3_EXT_EV_2_REG	Address:	78h

Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Timer EXT_EVENT_2 Register for Timer 3					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	EXT_EVENT_2_VALUE	00000000h	r	w	External Event 2 Value

Register:	TIM_4_MODE_REG			Address:	80h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)
Description:	Timer Mode Register for Timer 4					

Bit	Identifier	Reset	Attr.		Function / Description
0	INIT_BIT	0h		w	Initialization (0 =INIT_BIT not active, 1 =INIT_BIT active)
3dt1	<reserved>	0h			not used
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0 =CLK_TIMT, 1 =ext. Gate-/Triggersignal, see also bit GATE_EFFECT)
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0 =Reload Mode is active, 1 =Single Mode is active
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when 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>	0h			not used

Register:	TIM_4_PRESCALER_REG			Address:	84h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	(r)	(w)
Description:	Timer PRESCALER Register for Timer 4					

Bit	Identifier	Reset	Attr.		Function / Description
7dt0	PRESCALER_VALUE	00h	r	w	Prescaler Value
31dt8	<reserved>	000000h			not used

Register:	TIM_4_LOAD_REG			Address:	88h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Timer LOAD/RELOAD Register for Timer 4					

Bit	Identifier	Reset	Attr.		Function / Description
31dt0	LOAD_RELOAD_VALUE	00000000h	r	w	Load/Reload value

Register:	TIM_4_COUNT_REG				Address:	8Ch	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	
Description:	Timer COUNT Register for Timer 4						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	COUNTER_VALUE	00000000h	r		Counter Value		

Register:	TIM_4_INT_EV_REG				Address:	90h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	w
Description:	Timer INT_EVENT Register for Timer 4						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	INT_EVENT_VALUE	00000000h	r	w	Internal Event Value		

Register:	TIM_4_EXT_EV_1_REG				Address:	94h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	w
Description:	Timer EXT_EVENT_1 Register for Timer 4						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_1_VALUE	00000000h	r	w	External Event 1 Value		

Register:	TIM_4_EXT_EV_2_REG				Address:	98h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	w
Description:	Timer EXT_EVENT_2 Register for Timer 4						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_2_VALUE	00000000h	r	w	External Event 2 Value		

Register:	TIM_5_MODE_REG				Address:	A0h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)
Description:	Timer Mode Register for Timer 5						
Bit	Identifier	Reset	Attr.		Function / Description		
0	INIT_BIT	0h		w	Initialization (0 =INIT_BIT not active, 1 =INIT_BIT active)		
3dt1	<reserved>	0h			not used		
4	CLK_INPUT_SELECT	0h	r	w	Selection of Count-/Load-Clock 0 =CLK_TIMT, 1 =ext. Gate-/Triggersignal, see also bit GATE_EFFECT)		
5	RELOAD_DISABLE	0h	r	w	Reload Mode disable 0 =Reload Mode is active, 1 =Single Mode is active		
6	DIS_RLD_WHEN_WR_LDREG	0h	r	w	Disable Reload when 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>	0h			not used

Register:	TIM_5_PRESCALER_REG				Address:	A4h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)
Description:	Timer PRESCALER Register for Timer 5						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	PRESCALER_VALUE	00h	r	w	Prescaler Value		
31dt8	<reserved>	000000h			not used		

Register:	TIM_5_LOAD_REG				Address:	A8h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	w
Description:	Timer LOAD/RELOAD Register for Timer 5						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	LOAD_RELOAD_VALUE	00000000h	r	w	Load/Reload value		

Register:	TIM_5_COUNT_REG				Address:	ACh	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	
Description:	Timer COUNT Register for Timer 5						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	COUNTER_VALUE	00000000h	r		Counter Value		

Register:	TIM_5_INT_EV_REG				Address:	B0h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	w
Description:	Timer INT_EVENT Register for Timer 5						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	INT_EVENT_VALUE	00000000h	r	w	Internal Event Value		

Register:	TIM_5_EXT_EV_1_REG				Address:	B4h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	w
Description:	Timer EXT_EVENT_1 Register for Timer 5						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_1_VALUE	00000000h	r	w	External Event 1 Value		

Register:	TIM_5_EXT_EV_2_REG				Address:	B8h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	r	w
Description:	Timer EXT_EVENT_2 Register for Timer 5						
Bit	Identifier	Reset	Attr.		Function / Description		
31dt0	EXT_EVENT_2_VALUE	00000000h	r	w	External Event 2 Value		

Register:	GATE_TRIG_CONTROL_REG				Address:	C0h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)
Description:	Timer Gate Control Register						
Bit	Identifier	Reset	Attr.		Function / Description		
0	TIM_0_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 0		
1	TIM_1_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 1		
2	TIM_2_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 2		
3	TIM_3_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 3		
4	TIM_4_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 4		
5	TIM_5_INT_GATE_TRIG	0h	r	w	Software Gate-/Trigger signal Timer 5		
6	TIM_0_CLK_EN	0h	r	w	Clock Enable Timer 0		
7	TIM_1_CLK_EN	0h	r	w	Clock Enable Timer 1		
8	TIM_2_CLK_EN	0h	r	w	Clock Enable Timer 2		
9	TIM_3_CLK_EN	0h	r	w	Clock Enable Timer 3		
10	TIM_4_CLK_EN	0h	r	w	Clock Enable Timer 4		
11	TIM_5_CLK_EN	0h	r	w	Clock Enable Timer 5		
31dt12	<reserved>	00000h			not used		

Register:	CLOCK_DIVIDER_REG				Address:	C4h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)
Description:	Timer Clock Devider Register						
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	CLOCK_DIVIDER_VALUE	00h	r	w	Clock Divider Value		
8	CLK_DIV_EN	0h	r	w	Clock Divider Enable (0 =disabled		
31dt9	<reserved>	000000h			not used		

Register:	EXT_GATE_TRIG_MUX_REG				Address:	C8h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)
Description:	Timer MUX Register External Gate						
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	TIM0_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_GATE_TRIG of Timer 0		
7dt4	TIM1_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_GATE_TRIG of Timer 1		
11dt8	TIM2_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_GATE_TRIG of Timer 2		
15dt12	TIM3_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_GATE_TRIG of Timer 3		
19dt16	TIM4_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_GATE_TRIG of Timer 4		
23dt20	TIM5_EXT_G_T_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_GATE_TRIG of Timer 5		
31dt24	<reserved>	00h			not used		

Register:	EXT_EV_1_MUX_REG				Address:	CCh	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)
Description:	Timer MUX Register Event1 Select						

Bit	Identifier	Reset	Attr.		Function / Description
3dt0	TIM0_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 1 of Timer 0
7dt4	TIM1_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 1 of Timer 1
11dt8	TIM2_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 1 of Timer 2
15dt12	TIM3_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 1 of Timer 3
19dt16	TIM4_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 1 of Timer 4
23dt20	TIM5_EVENT1_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 1 of Timer 5
31dt24	<reserved>	00h			not used

Register:	EXT_EV_2_MUX_REG				Address:	D0h	
Bits:	31dt0	Reset value:		00000000h	Attributes:	(r)	(w)
Description:	Timer MUX Register Event2 Select						

Bit	Identifier	Reset	Attr.		Function / Description
3dt0	TIM0_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 2 of Timer 0
7dt4	TIM1_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 2 of Timer 1
11dt8	TIM2_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 2 of Timer 2
15dt12	TIM3_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 2 of Timer 3
19dt16	TIM4_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 2 of Timer 4
23dt20	TIM5_EVENT2_SEL_3_0	0h	r	w	Selection of EXTERNAL_INPUTS(15:0) for EXT_Event 2 of Timer 5
31dt24	<reserved>	00h			not used

Register:	SW_EVENT_TRIGGER_REG				Address:	D4h	
Bits:	31dt0	Reset value:		00000000h	Attributes:		(w)
Description:	Timer SW Event Register						

Bit	Identifier	Reset	Attr.		Function / Description
5dt0	SW_EVENT_TRIG5_0	00h		w	Software Event for Saving Counter value in Int_Event register of Timer 5...0 1: Software event active, read =always 0
31dt6	<reserved>	0000000h			not used

5.3.12 ARM926 Watchdog-Register

Base address see Chapter 5.2.

Address space:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	18h	wdog	

Module	Register/Memory	Read	Write	Address
wdog	WD_CTRL_STATUS	(r)(h)	(w)(k)	0h
	RELD0_LOW	(r)	w(k)	4h
	RELD0_HIGH	(r)	w(k)	8h
	RELD1_LOW	(r)	w(k)	Ch
	RELD1_HIGH	(r)	w(k)	10h
	WDOG0	rh		14h
	WDOG1	rh		18h

Register allocation:

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

Module: /wdog

Register:	WD_CTRL_STATUS				Address:	0h	
Bits:	31dt0	Reset-Value:	00000000h		Attribute:	(r)(h)	(w)(k)
Beschreibung:	Control/Status register. Configuration and control bits for the watchdog.						

Bit	Bezeichner	Reset	Attr.		Function / Description
31dt16	Key_bits	0000h		wk	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-4 of this register will be written, otherwise the operation has no effect.
4	Status_Counter1	0h	rh		Watchdog status counter 1 (write is ignored): 0: Watchdog counter 1 has not expired 1: Watchdog counter 1 has expired Note: This bit then can only be read as '1' when RUN/xStop_Z1 is active (1).
3	Status_Counter0	0h	rh		Watchdog status counter 0 (write is ignored): 0: Watchdog counter 0 has not expired 1: Watchdog counter 0 has expired Note: This bit then can only be read as '1' when RUN/xStop_Z0 is active (1).
2	Load_Trigger	0h	r	w	Watchdog trigger (load watchdog counters 0 and 1 with the value of the Reload registers): 0: Do not trigger watchdog 1: Trigger watchdog Although this bit can be read back, it acts only during a write. To trigger the watchdog counters, it suffices to write a 1 to this bit; no 0/1 edge is required. The trigger signal acts on both watchdog counters.
1	Run_xStop_Z1	0h	r	w	Enable/disable watchdog counter 1: 0: Watchdog counter 1 disabled 1: Watchdog counter 1 enabled Note: If this bit = 0, the XWDOUT1 output of the ERTEC 200P is passive (1) and the status bit of counter 1 (bit 4)

					is 0.
0	Run_xStop_Z0	0h	r	w	Enable/disable watchdog counter 0: 0: Watchdog counter 0 disabled 1: Watchdog counter 0 enabled Note: If this bit = 0, the XWDOUT0 output of the ERTEC 200P is active (0), the interrupt of the watchdog (WDINT) is 0 and the status bit of counter 0 (bit 3) is 0.

Register:	RELD0_LOW				Address:	4h	
Bits:	31dt0	Reset-Value:		0000FFFFh	Attribute:	(r)	w(k)
Beschreibung:	Reload Register 0_low. Reload value for bits 15:0 of watchdog counter 0.						
Bit	Bezeichner	Reset	Attr.		Function / Description		
31dt16	Key_bits	0000h		wk	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-15 of this register will be written, otherwise the operation has no effect.		
15dt0	Reload0	FFFFh	r	w	Reload value for bits 15:0 of watchdog counter 0		

Register:	RELD0_HIGH				Address:	8h	
Bits:	31dt0	Reset-Value:		0000FFFFh	Attribute:	(r)	w(k)
Beschreibung:	Reload Register 0_high. Reload value for bits 31:16 of watchdog counter 0.						
Bit	Bezeichner	Reset	Attr.		Function / Description		
31dt16	Key_bits	0000h		wk	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-15 of this register will be written, otherwise the operation has no effect.		
15dt0	Reload0	FFFFh	r	w	Reload value for bits 31-16 of watchdog counter 0		

Register:	RELD1_LOW				Address:	Ch	
Bits:	31dt0	Reset-Value:		0000FFFFh	Attribute:	(r)	w(k)
Beschreibung:	Reload Register 1_low. Reload value for bits 19:4 of watchdog counter 1.						
Bit	Bezeichner	Reset	Attr.		Function / Description		
31dt16	Key_bits	0000h		wk	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-15 of this register will be written, otherwise the operation has no effect.		
15dt0	Reload1	FFFFh	r	w	Reload value for bits 19:4 of watchdog counter 1.		

Register:	RELD1_HIGH				Address:	10h	
Bits:	31dt0	Reset-Value:		0000FFFFh	Attribute:	(r)	w(k)
Beschreibung:	Reload Register 1_high. Reload value for bits 35:20 of watchdog counter 1.						
Bit	Bezeichner	Reset	Attr.		Function / Description		
31dt16	Key_bits	0000h		wk	Key bits for writing this register (read = 0). If bits 31-16 = 9876h, bits 0-15 of this register will be written, otherwise the operation has no effect.		
15dt0	Reload1	FFFFh	r	w	Reload value for bits 35:20 of watchdog counter 1		

Register:	WDOG0			Address:	14h	
Bits:	31dt0	Reset-Value:	FFFFFFFFh	Attribute:	rh	
Beschreibung:	Watchdog value 0. Value of watchdog counter 0.					

Register:	WDOG1			Address:	18h	
Bits:	31dt0	Reset-Value:	FFFFFFFFh	Attribute:	rh	

Beschreibung:	Watchdog value 1. Value of watchdog counter 1.

5.3.13 F-Timer-Register

Base address see Chapter 5.2.

Address space:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	4h	FCOUNT	<interface>

Module	Register/Memory	Read	Write	Address
/FCOUNT				
	FCOUNTER_VAL	r(h)		0h
	FCOUNTER_RES	rh	w	4h

Register allocation:

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

Module: /FCOUNT

Register:	FCOUNTER_VAL			Address:	0h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r(h)	
Description:	Timerwert des F-Timers					

Register:	FCOUNTER_RES			Address:	4h	
Bits:	31dt0	Reset value:	0h	Attributes:	rh	w
Description:	Reset register for F-counters. Resetting of the F-counter is only carried out if a data 0xXXXX 55AAh is entered in this register. Resets are thus possible through word and double-word accesses.					

Bit	Identifier	Reset	Attr.		Function / Description
15dt0	FCOUNT_RES_LWORD	0000h	r(h)	w	Lower half-word of the F-counter reset
31dt16	FCOUNT_RES_HWORD	0000h	r(h)	w	Upper half-word of the F-counter reset (don't care)

5.3.14 GPIO register

Base address see Chapter 5.2.

Address space:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	5Ch	gpio	APB

Module	Register/Memory	Read	Write	Address
/gpio				
	GPIO_IOCTRL_0	r	w	0h
	GPIO_OUT_0	rh	w	4h
	GPIO_OUT_SET_0	rh	wt	8h
	GPIO_OUT_CLEAR_0	rh	wt	Ch
	GPIO_RES_DIS_0	r	w	10h
	GPIO_IN_0	rh		14h
	GPIO_PORT_MODE_0_L	r	w	18h
	GPIO_PORT_MODE_0_H	r	w	1Ch
	GPIO_IOCTRL_1	r	w	20h
	GPIO_OUT_1	rh	w	24h
	GPIO_OUT_SET_1	rh	wt	28h
	GPIO_OUT_CLEAR_1	rh	wt	2Ch
	GPIO_RES_DIS_1	r	w	30h
	GPIO_IN_1	rh		34h
	GPIO_PORT_MODE_1_L	r	w	38h
	GPIO_PORT_MODE_1_H	r	w	3Ch
	GPIO_IOCTRL_2	r	w	40h
	GPIO_OUT_2	rh	w	44h
	GPIO_OUT_SET_2	rh	wt	48h
	GPIO_OUT_CLEAR_2	rh	wt	4Ch
	GPIO_RES_DIS_2	r	w	50h
	GPIO_IN_2	rh		54h
	GPIO_PORT_MODE_2_L	r	w	58h
	GPIO_PORT_MODE_2_H	r	w	5Ch

Register allocatiom:

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

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					

Register:	GPIO_OUT_0			Address:	4h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	w
Description:	Output register for General Purpose IOs (31:0) 0: GPIO outputx = 0, 1: GPIO outputx = 1					

Register:	GPIO_OUT_SET_0			Address:	8h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	wt
Description:	Bit-selective setting of the output register for General Purpose IOs (31:0) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 1 Read always returns 0					

Register:	GPIO_OUT_CLEAR_0			Address:	Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	wt
Description:	Bit-selective reset of the output register for General Purpose IOs (31:0) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 0 Read always returns 0					

Register:	GPIO_RES_DIS_0			Address:	10h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Bit-selective reset disabling of XRESET_GPIO_SM signal for registers of General Purpose IOs (31:0) For writing: 0: XRESET_GPIO_SM resets the corresponding register bit of all registers except of GPIO_RES_DIS 1: XRESET_GPIO_SM has no effect on the corresponding register bit					

Register:	GPIO_IN_0			Address:	14h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Input register for General Purpose IOs (31:0)					

Register:	GPIO_PORT_MODE_0_L			Address:	18h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Configuration register for GPIO port (15:0) Function assignment: 00 =GPIO function 01 =alternate function A 10 =alternate function B 11 =alternate function C					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	GPIO_0_MODE_0_L	0h	r	w	Port GPIO(0)
3dt2	GPIO_1_MODE_0_L	0h	r	w	Port GPIO(1)
5dt4	GPIO_2_MODE_0_L	0h	r	w	Port GPIO(2)
7dt6	GPIO_3_MODE_0_L	0h	r	w	Port GPIO(3)
9dt8	GPIO_4_MODE_0_L	0h	r	w	Port GPIO(6)
11dt10	GPIO_5_MODE_0_L	0h	r	w	Port GPIO(5)
13dt12	GPIO_6_MODE_0_L	0h	r	w	Port GPIO(6)
15dt14	GPIO_7_MODE_0_L	0h	r	w	Port GPIO(7)
17dt16	GPIO_8_MODE_0_L	0h	r	w	Port GPIO(8)
19dt18	GPIO_9_MODE_0_L	0h	r	w	Port GPIO(9)
21dt20	GPIO_10_MODE_0_L	0h	r	w	Port GPIO(10)
23dt22	GPIO_11_MODE_0_L	0h	r	w	Port GPIO(11)
25dt24	GPIO_12_MODE_0_L	0h	r	w	Port GPIO(12)

27dt26	GPIO_13_MODE_0_L	0h	r	w	Port GPIO(13)
29dt28	GPIO_14_MODE_0_L	0h	r	w	Port GPIO(14)
31dt30	GPIO_15_MODE_0_L	0h	r	w	Port GPIO(15)

Register:	GPIO_PORT_MODE_0_H			Address:	1Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Configuration register for GPIO port (31:16) Function assignment: 00 =GPIO function 01 =alternate function A 10 =alternate function B 11 =alternate function C					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	GPIO_16_MODE_0_H	0h	r	w	Port GPIO(16)
3dt2	GPIO_17_MODE_0_H	0h	r	w	Port GPIO(17)
5dt4	GPIO_18_MODE_0_H	0h	r	w	Port GPIO(18)
7dt6	GPIO_19_MODE_0_H	0h	r	w	Port GPIO(19)
9dt8	GPIO_20_MODE_0_H	0h	r	w	Port GPIO(20)
11dt10	GPIO_21_MODE_0_H	0h	r	w	Port GPIO(21)
13dt12	GPIO_22_MODE_0_H	0h	r	w	Port GPIO(22)
15dt14	GPIO_23_MODE_0_H	0h	r	w	Port GPIO(23)
17dt16	GPIO_24_MODE_0_H	0h	r	w	Port GPIO(24)
19dt18	GPIO_25_MODE_0_H	0h	r	w	Port GPIO(25)
21dt20	GPIO_26_MODE_0_H	0h	r	w	Port GPIO(26)
23dt22	GPIO_27_MODE_0_H	0h	r	w	Port GPIO(27)
25dt24	GPIO_28_MODE_0_H	0h	r	w	Port GPIO(28)
27dt26	GPIO_29_MODE_0_H	0h	r	w	Port GPIO(29)
29dt28	GPIO_30_MODE_0_H	0h	r	w	Port GPIO(30)
31dt30	GPIO_31_MODE_0_H	0h	r	w	Port GPIO(31)

Register:	GPIO_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					

Register:	GPIO_OUT_1			Address:	24h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	w
Description:	Output register for General Purpose IOs (63:32) 0: GPIO outputx = 0, 1: GPIO outputx = 1					

Register:	GPIO_OUT_SET_1			Address:	28h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	wt
Description:	Bit-selective setting of the output register for General Purpose IOs (63:32) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 1 Read always returns 0					

Register:	GPIO_OUT_CLEAR_1			Address:	2Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	wt
Description:	Bit-selective reset of the output register for General Purpose IOs (63:32) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 0 Read always returns 0					

Register:	GPIO_RES_DIS_1			Address:	30h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Bit-selective reset disabling of XRESET_GPIO_SM signal for registers of General Purpose IOs (31:0) For writing: 0: XRESET_GPIO_SM resets the corresponding register bit of all registers except of GPIO_RES_DIS 1: XRESET_GPIO_SM has no effect on the corresponding register bit					

Register:	GPIO_IN_1			Address:	34h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Input register for General Purpose IOs (63:32) 0: GPIO inputx = 0, 1: GPIO inputx = 1					

Register:	GPIO_PORT_MODE_1_L			Address:	38h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Configuration register for GPIO port (47:32) Function assignment: 00 =GPIO function 01 =alternate function A 10 =alternate function B 11 =alternate function C					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	GPIO_32_MODE_1_L	0h	r	w	Port GPIO(32)
3dt2	GPIO_33_MODE_1_L	0h	r	w	Port GPIO(33)
5dt4	GPIO_34_MODE_1_L	0h	r	w	Port GPIO(34)
7dt6	GPIO_35_MODE_1_L	0h	r	w	Port GPIO(35)
9dt8	GPIO_36_MODE_1_L	0h	r	w	Port GPIO(36)
11dt10	GPIO_37_MODE_1_L	0h	r	w	Port GPIO(37)
13dt12	GPIO_38_MODE_1_L	0h	r	w	Port GPIO(38)
15dt14	GPIO_39_MODE_1_L	0h	r	w	Port GPIO(39)
17dt16	GPIO_40_MODE_1_L	0h	r	w	Port GPIO(40)
19dt18	GPIO_41_MODE_1_L	0h	r	w	Port GPIO(41)
21dt20	GPIO_42_MODE_1_L	0h	r	w	Port GPIO(42)
23dt22	GPIO_43_MODE_1_L	0h	r	w	Port GPIO(43)
25dt24	GPIO_44_MODE_1_L	0h	r	w	Port GPIO(44)

27dt26	GPIO_45_MODE_1_L	0h	r	w	Port GPIO(45)
29dt28	GPIO_46_MODE_1_L	0h	r	w	Port GPIO(46)
31dt30	GPIO_47_MODE_1_L	0h	r	w	Port GPIO(47)

Register:	GPIO_PORT_MODE_1_H			Address:	3Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Configuration register for GPIO port (63:48) Function assignment: 00 =GPIO function 01 =alternate function A 10 =alternate function B 11 =alternate function C					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	GPIO_48_MODE_1_H	0h	r	w	Port GPIO(48)
3dt2	GPIO_49_MODE_1_H	0h	r	w	Port GPIO(49)
5dt4	GPIO_50_MODE_1_H	0h	r	w	Port GPIO(50)
7dt6	GPIO_51_MODE_1_H	0h	r	w	Port GPIO(51)
9dt8	GPIO_52_MODE_1_H	0h	r	w	Port GPIO(52)
11dt10	GPIO_53_MODE_1_H	0h	r	w	Port GPIO(53)
13dt12	GPIO_54_MODE_1_H	0h	r	w	Port GPIO(54)
15dt14	GPIO_55_MODE_1_H	0h	r	w	Port GPIO(55)
17dt16	GPIO_56_MODE_1_H	0h	r	w	Port GPIO(56)
19dt18	GPIO_57_MODE_1_H	0h	r	w	Port GPIO(57)
21dt20	GPIO_58_MODE_1_H	0h	r	w	Port GPIO(58)
23dt22	GPIO_59_MODE_1_H	0h	r	w	Port GPIO(59)
25dt24	GPIO_60_MODE_1_H	0h	r	w	Port GPIO(60)
27dt26	GPIO_61_MODE_1_H	0h	r	w	Port GPIO(61)
29dt28	GPIO_62_MODE_1_H	0h	r	w	Port GPIO(62)
31dt30	GPIO_63_MODE_1_H	0h	r	w	Port GPIO(63)

Register:	GPIO_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					

Register:	GPIO_OUT_2			Address:	44h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	w
Description:	Output register for General Purpose IOs (95:64) 0: GPIO outputx = 0, 1: GPIO outputx = 1					

Register:	GPIO_OUT_SET_2			Address:	48h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	wt
Description:	Bit-selective setting of the output register for General Purpose IOs (95:64) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 1 Read always returns 0					

Register:	GPIO_OUT_CLEAR_2			Address:	4Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	wt
Description:	Bit-selective reset of the output register for General Purpose IOs (95:64) For writing: 0: GPIO outputx remains unchanged 1: GPIO outputx = 0 Read always returns 0					

Register:	GPIO_RES_DIS_2			Address:	50h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Bit-selective reset disabling of XRESET_GPIO_SM signal for registers of General Purpose IOs (31:0) For writing: 0: XRESET_GPIO_SM resets the corresponding register bit of all registers except of GPIO_RES_DIS 1: XRESET_GPIO_SM has no effect on the corresponding register bit					

Register:	GPIO_IN_2			Address:	54h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	rh	
Description:	Input register for General Purpose IOs (95:64)					

Register:	GPIO_PORT_MODE_2_L			Address:	58h	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Configuration register for GPIO port (79:64) Function assignment: 00 =GPIO function 01 =alternate function A 10 =alternate function B 11 =alternate function C					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	GPIO_64_MODE_2_L	0h	r	w	Port GPIO(64)
3dt2	GPIO_65_MODE_2_L	0h	r	w	Port GPIO(65)
5dt4	GPIO_66_MODE_2_L	0h	r	w	Port GPIO(66)
7dt6	GPIO_67_MODE_2_L	0h	r	w	Port GPIO(67)
9dt8	GPIO_68_MODE_2_L	0h	r	w	Port GPIO(68)
11dt10	GPIO_69_MODE_2_L	0h	r	w	Port GPIO(69)
13dt12	GPIO_70_MODE_2_L	0h	r	w	Port GPIO(70)
15dt14	GPIO_71_MODE_2_L	0h	r	w	Port GPIO(71)
17dt16	GPIO_72_MODE_2_L	0h	r	w	Port GPIO(72)
19dt18	GPIO_73_MODE_2_L	0h	r	w	Port GPIO(73)
21dt20	GPIO_74_MODE_2_L	0h	r	w	Port GPIO(74)
23dt22	GPIO_75_MODE_2_L	0h	r	w	Port GPIO(75)
25dt24	GPIO_76_MODE_2_L	0h	r	w	Port GPIO(76)

27dt26	GPIO_77_MODE_2_L	0h	r	w	Port GPIO(77)
29dt28	GPIO_78_MODE_2_L	0h	r	w	Port GPIO(78)
31dt30	GPIO_79_MODE_2_L	0h	r	w	Port GPIO(79)

Register:	GPIO_PORT_MODE_2_H			Address:	5Ch	
Bits:	31dt0	Reset value:	00000000h	Attributes:	r	w
Description:	Configuration register for GPIO port (95:80) Function assignment: 00 =GPIO function 01 =alternate function A 10 =alternate function B 11 =alternate function C					

Bit	Identifier	Reset	Attr.		Function / Description
1dt0	GPIO_80_MODE_2_H	0h	r	w	Port GPIO(80)
3dt2	GPIO_81_MODE_2_H	0h	r	w	Port GPIO(81)
5dt4	GPIO_82_MODE_2_H	0h	r	w	Port GPIO(82)
7dt6	GPIO_83_MODE_2_H	0h	r	w	Port GPIO(83)
9dt8	GPIO_84_MODE_2_H	0h	r	w	Port GPIO(84)
11dt10	GPIO_85_MODE_2_H	0h	r	w	Port GPIO(85)
13dt12	GPIO_86_MODE_2_H	0h	r	w	Port GPIO(86)
15dt14	GPIO_87_MODE_2_H	0h	r	w	Port GPIO(87)
17dt16	GPIO_88_MODE_2_H	0h	r	w	Port GPIO(88)
19dt18	GPIO_89_MODE_2_H	0h	r	w	Port GPIO(89)
21dt20	GPIO_90_MODE_2_H	0h	r	w	Port GPIO(90)
23dt22	GPIO_91_MODE_2_H	0h	r	w	Port GPIO(91)
25dt24	GPIO_92_MODE_2_H	0h	r	w	Port GPIO(92)
27dt26	GPIO_93_MODE_2_H	0h	r	w	Port GPIO(93)
29dt28	GPIO_94_MODE_2_H	0h	r	w	Port GPIO(94)
31dt30	GPIO_95_MODE_2_H	0h	r	w	Port GPIO(95)

5.3.15 I-Filter-Register

Base address see Chapter 5.2.

Address space:

Start_Address	End_Address	Modul/Memory_Name	Interface
0h	50h	i_filter	APB

Module	Register/Memory	Read	Write	Address
/i_filter				
	FILT_IP_VERSION	r		0h
	FILT_IP_DEVELOPMENT	r		4h
	FILT_ACCESS_ERR	(r)(h)	(w)	8h
	FILT_RELOAD_0	r	w	Ch
	FILT_RELOAD_1	r	w	10h
	FILT_RELOAD_2	r	w	14h
	FILT_RELOAD_3	r	w	18h
	FILT_RELOAD_4	r	w	1Ch
	FILT_RELOAD_5	r	w	20h
	FILT_RELOAD_6	r	w	24h
	FILT_RELOAD_7	r	w	28h
	FILT_DELAY_0	r	w	2Ch
	FILT_DELAY_1	r	w	30h
	FILT_DELAY_2	r	w	34h
	FILT_DELAY_3	r	w	38h
	FILT_DELAY_4	r	w	3Ch
	FILT_DELAY_5	r	w	40h
	FILT_DELAY_6	r	w	44h
	FILT_DELAY_7	r	w	48h
	FILT_DELAY_8	r	w	4Ch
	FILT_DELAY_9	r	w	50h

Register allocation:

Module: /i_filter

Register:	FILT_IP_VERSION				Address:	0h	
Bits:	31dt0	Reset value:		10100h	Attributes:	r	
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
7dt0	DEBUG_VERSION	00h	r		Is incremented for error correction (e.g. Metal Fix)		
15dt8	VERSION	01h	r		Version of the IP: 0x01: Initial state		
31dt16	CONFIGURATION	0001h	r		Configuration of the IP: 0x0000: 8 filter times, 64 input filters 0x0001: 8 filter times, 80 input filters		

Register:	FILT_IP_DEVELOPMENT			Address:	4h	
Bits:	31dt0	Reset value:	4E00807h	Attributes:	r	

Description:		Project-specific imaging of the design label			
Bit	Identifier	Reset	Attr.		Function / Description
10dt0	BASELINE	007h	r		Number of the RR Label
15dt11	INKREMENT	01h	r		HDL increments of the ClearCase label
18dt16	PATCH	0h	r		For identifying Metal fixes in the ASIC Flow. Only valid if platform = ASIC. Through it Increment/Baseline become invalid
20dt19	PLATFORM	0h	r		00 = ASIC 01 = FPGA 10 = reserved 11 = user defined
31dt21	IDENTIFIKATION	027h	r		Unique number per module

Register:	FILT_ACCESS_ERR				Address:	8h	
Bits:	31dt0	Reset value:		none	Attributes:	(r)(h)	(w)
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
6dt0	APB_ADDRESS	00h	rh	w	Erroneous APB address in I-Filter		
27dt7	<reserved>						
29dt28	APB_SIZE	0h	rh	w	00: Byte access 01: Half-word access 10: Word access 11: Reserved		
30	APB_WRITE	0h	rh	w	0: Read access 1: Write access		
31	ERR_LOCK	0h	rh	w	Is set by the HW to 1 when an erroneous APB access is recognized. This blocks further HW entries. The SW (PNB-Stack) has to reset the bit to 0 to enable new entries.		

Register:	FILT_RELOAD_0			Address:	Ch	
Bits:	9dt0	Reset value:	000h	Attributes:	r	w
Description:	Load value of the cycle divider 0 Here the division factor reduced by 1 has to be entered. The load value 0x000 is converted by the HW to 0x001.					

Register:	FILT_RELOAD_1			Address:	10h	
Bits:	9dt0	Reset value:	000h	Attributes:	r	w
Description:	Load value of the cycle divider 1 Here the division factor reduced by 1 has to be entered. The load value 0x000 is converted by the HW to 0x001.					

Register:	FILT_RELOAD_2			Address:	14h	
Bits:	9dt0	Reset value:	000h	Attributes:	r	w
Description:	Load value of the cycle divider 2 Here the division factor reduced by 1 has to be entered. The load value 0x000 is converted by the HW to 0x001.					

Register:	FILT_RELOAD_3			Address:	18h	
Bits:	9dt0	Reset value:	000h	Attributes:	r	w
Description:	Load value of the cycle divider 3 Here the division factor reduced by 1 has to be entered. The load value 0x000 is converted by the HW to 0x001.					

Register:	FILT_RELOAD_4			Address:	1Ch	
Bits:	9dt0	Reset value:	000h	Attributes:	r	w
Description:	Load value of the cycle divider 4 Here the division factor reduced by 1 has to be entered. The load value 0x000 is converted by the HW to 0x001.					

Register:	FILT_RELOAD_5			Address:	20h	
Bits:	9dt0	Reset value:	000h	Attributes:	r	w
Description:	Load value of the cycle divider 5 Here the division factor reduced by 1 has to be entered. The load value 0x000 is converted by the HW to 0x001.					

Register:	FILT_RELOAD_6			Address:	24h	
Bits:	9dt0	Reset value:	000h	Attributes:	r	w
Description:	Load value of the cycle divider 6 Here the division factor reduced by 1 has to be entered. The load value 0x000 is converted by the HW to 0x001.					

Register:	FILT_RELOAD_7			Address:	28h	
Bits:	9dt0	Reset value:	000h	Attributes:	r	w
Description:	Load value of the cycle divider 7 Here the division factor reduced by 1 has to be entered. The load value 0x000 is converted by the HW to 0x001.					

Register:	FILT_DELAY_0			Address:	2Ch	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:						

Bit	Identifier	Reset	Attr.		Function / Description
3dt0	IN_DELAY_0	Fh	r	w	Selection of the cycle source for input signal 0: 0x0: TAKT0 0x1: TAKT1 0x2: TAKT2 0x3: TAKT3 0x4: TAKT4 0x5: TAKT5 0x6: TAKT6 0x7: TAKT7 0x8 - 0xE: No filtering, only synchronization 0xF: No filtering, no synchronization, only passing on
7dt4	IN_DELAY_1	Fh	r	w	See IN_DELAY_0
11dt8	IN_DELAY_2	Fh	r	w	See IN_DELAY_0
15dt12	IN_DELAY_3	Fh	r	w	See IN_DELAY_0
19dt16	IN_DELAY_4	Fh	r	w	See IN_DELAY_0
23dt20	IN_DELAY_5	Fh	r	w	See IN_DELAY_0
27dt24	IN_DELAY_6	Fh	r	w	See IN_DELAY_0
31dt28	IN_DELAY_7	Fh	r	w	See IN_DELAY_0

Register:	FILT_DELAY_1			Address:	30h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:						

Bit	Identifier	Reset	Attr.	Function / Description

3dt0	IN_DELAY_8	Fh	r	w	See IN_DELAY_0
7dt4	IN_DELAY_9	Fh	r	w	See IN_DELAY_0
11dt8	IN_DELAY_10	Fh	r	w	See IN_DELAY_0
15dt12	IN_DELAY_11	Fh	r	w	See IN_DELAY_0
19dt16	IN_DELAY_12	Fh	r	w	See IN_DELAY_0
23dt20	IN_DELAY_13	Fh	r	w	See IN_DELAY_0
27dt24	IN_DELAY_14	Fh	r	w	See IN_DELAY_0
31dt28	IN_DELAY_15	Fh	r	w	See IN_DELAY_0

Register:	FILT_DELAY_2				Address:	34h	
Bits:	31dt0	Reset value:		FFFFFFFFh	Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	IN_DELAY_16	Fh	r	w	See IN_DELAY_0		
7dt4	IN_DELAY_17	Fh	r	w	See IN_DELAY_0		
11dt8	IN_DELAY_18	Fh	r	w	See IN_DELAY_0		
15dt12	IN_DELAY_19	Fh	r	w	See IN_DELAY_0		
19dt16	IN_DELAY_20	Fh	r	w	See IN_DELAY_0		
23dt20	IN_DELAY_21	Fh	r	w	See IN_DELAY_0		
27dt24	IN_DELAY_22	Fh	r	w	See IN_DELAY_0		
31dt28	IN_DELAY_23	Fh	r	w	See IN_DELAY_0		

Register:	FILT_DELAY_3				Address:	38h	
Bits:	31dt0	Reset value:		FFFFFFFFh	Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	IN_DELAY_24	Fh	r	w	See IN_DELAY_0		
7dt4	IN_DELAY_25	Fh	r	w	See IN_DELAY_0		
11dt8	IN_DELAY_26	Fh	r	w	See IN_DELAY_0		
15dt12	IN_DELAY_27	Fh	r	w	See IN_DELAY_0		
19dt16	IN_DELAY_28	Fh	r	w	See IN_DELAY_0		
23dt20	IN_DELAY_29	Fh	r	w	See IN_DELAY_0		
27dt24	IN_DELAY_30	Fh	r	w	See IN_DELAY_0		
31dt28	IN_DELAY_31	Fh	r	w	See IN_DELAY_0		

Register:	FILT_DELAY_4				Address:	3Ch	
Bits:	31dt0	Reset value:		FFFFFFFFh	Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	IN_DELAY_32	Fh	r	w	See IN_DELAY_0		
7dt4	IN_DELAY_33	Fh	r	w	See IN_DELAY_0		
11dt8	IN_DELAY_34	Fh	r	w	See IN_DELAY_0		
15dt12	IN_DELAY_35	Fh	r	w	See IN_DELAY_0		
19dt16	IN_DELAY_36	Fh	r	w	See IN_DELAY_0		
23dt20	IN_DELAY_37	Fh	r	w	See IN_DELAY_0		

27dt24	IN_DELAY_38	Fh	r	w	See IN_DELAY_0
31dt28	IN_DELAY_39	Fh	r	w	See IN_DELAY_0

Register:	FILT_DELAY_5				Address:	40h	
Bits:	31dt0	Reset value:		FFFFFFFFh	Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	IN_DELAY_40	Fh	r	w	See IN_DELAY_0		
7dt4	IN_DELAY_41	Fh	r	w	See IN_DELAY_0		
11dt8	IN_DELAY_42	Fh	r	w	See IN_DELAY_0		
15dt12	IN_DELAY_43	Fh	r	w	See IN_DELAY_0		
19dt16	IN_DELAY_44	Fh	r	w	See IN_DELAY_0		
23dt20	IN_DELAY_45	Fh	r	w	See IN_DELAY_0		
27dt24	IN_DELAY_46	Fh	r	w	See IN_DELAY_0		
31dt28	IN_DELAY_47	Fh	r	w	See IN_DELAY_0		

Register:	FILT_DELAY_6				Address:	44h	
Bits:	31dt0	Reset value:		FFFFFFFFh	Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	IN_DELAY_48	Fh	r	w	See IN_DELAY_0		
7dt4	IN_DELAY_49	Fh	r	w	See IN_DELAY_0		
11dt8	IN_DELAY_50	Fh	r	w	See IN_DELAY_0		
15dt12	IN_DELAY_51	Fh	r	w	See IN_DELAY_0		
19dt16	IN_DELAY_52	Fh	r	w	See IN_DELAY_0		
23dt20	IN_DELAY_53	Fh	r	w	See IN_DELAY_0		
27dt24	IN_DELAY_54	Fh	r	w	See IN_DELAY_0		
31dt28	IN_DELAY_55	Fh	r	w	See IN_DELAY_0		

Register:	FILT_DELAY_7				Address:	48h	
Bits:	31dt0	Reset value:		FFFFFFFFh	Attributes:	r	w
Description:							
Bit	Identifier	Reset	Attr.		Function / Description		
3dt0	IN_DELAY_56	Fh	r	w	See IN_DELAY_0		
7dt4	IN_DELAY_57	Fh	r	w	See IN_DELAY_0		
11dt8	IN_DELAY_58	Fh	r	w	See IN_DELAY_0		
15dt12	IN_DELAY_59	Fh	r	w	See IN_DELAY_0		
19dt16	IN_DELAY_60	Fh	r	w	See IN_DELAY_0		
23dt20	IN_DELAY_61	Fh	r	w	See IN_DELAY_0		
27dt24	IN_DELAY_62	Fh	r	w	See IN_DELAY_0		
31dt28	IN_DELAY_63	Fh	r	w	See IN_DELAY_0		

Register:	FILT_DELAY_8			Address:	4Ch	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:						

Bit	Identifier	Reset	Attr.		Function / Description
3dt0	IN_DELAY_64	Fh	r	w	See IN_DELAY_0
7dt4	IN_DELAY_65	Fh	r	w	See IN_DELAY_0
11dt8	IN_DELAY_66	Fh	r	w	See IN_DELAY_0
15dt12	IN_DELAY_67	Fh	r	w	See IN_DELAY_0
19dt16	IN_DELAY_68	Fh	r	w	See IN_DELAY_0
23dt20	IN_DELAY_69	Fh	r	w	See IN_DELAY_0
27dt24	IN_DELAY_70	Fh	r	w	See IN_DELAY_0
31dt28	IN_DELAY_71	Fh	r	w	See IN_DELAY_0

Register:	FILT_DELAY_9			Address:	50h	
Bits:	31dt0	Reset value:	FFFFFFFFh	Attributes:	r	w
Description:						

Bit	Identifier	Reset	Attr.		Function / Description
3dt0	IN_DELAY_72	Fh	r	w	See IN_DELAY_0
7dt4	IN_DELAY_73	Fh	r	w	See IN_DELAY_0
11dt8	IN_DELAY_74	Fh	r	w	See IN_DELAY_0
15dt12	IN_DELAY_75	Fh	r	w	See IN_DELAY_0
19dt16	IN_DELAY_76	Fh	r	w	See IN_DELAY_0
23dt20	IN_DELAY_77	Fh	r	w	See IN_DELAY_0
27dt24	IN_DELAY_78	Fh	r	w	See IN_DELAY_0
31dt28	IN_DELAY_79	Fh	r	w	See IN_DELAY_0

5.4 Detailed ARM Interrupt description

5.4.1 Interrupts at the ARM Interrupt Controller (IRQ)

Typ F/S Edge-triggered interrupt on rising edge
Typ F/F Edge-triggered interrupt on falling edge
Typ P Level-triggered interrupt
Typ *) Type depending on the selected operating mode of the module
Typ **) Type depending on the external connection (ERTEC 200P pins). Can be F/F or F/S or P, depending on the external connection. Parameterization has to be made depending on the requirements of the application.

Interrupt IRQ<n>	Interrupt Name	Interrupt Source	Type Inverter	Description
0	DEFAULT_VEC	-	-	Default Vector, **must be tied to '0' on toplevel**
1	GDMA_IRQ	GDMA	F/S	Combined Interrupt
2	INT_I2C	I2C	P	Combined Interrupt
3	Invalid Access	PN-IP	F/S	PN-IP hat in ein I-TCM Adresslücke gegriffen
4	Invalid Access	PN-IP	F/S	PN-IP hat in ein D-TCM Adresslücke gegriffen
5	UART1_UARTINTR	UART1	P	Combined Interrupt
6	UART1_UARTEINTR	UART1	P	Error Interrupt
7	UART2_UARTINTR	UART2	P	Combined Interrupt
8	UART2_UARTEINTR	UART2	P	Error Interrupt
9	UART3_UARTINTR	UART3	P	Combined Interrupt
10	UART3_UARTEINTR	UART3	P	Error Interrupt
11	UART4_UARTINTR	UART4	P	Combined Interrupt
12	UART4_UARTEINTR	UART4	P	Error Interrupt
13	SPI1_SSPINTR	SPI1	P	Combined Interrupt
14	SPI1_SSPRORINTR	SPI1	P	Overrun Error Interrupt
15	SPI2_SSPINTR	SPI2	P	Combined Interrupt
16	SPI2_SSPRORINTR	SPI2	P	Overrun Error Interrupt
17	reserved	Reserved	-	-
18	Reserved	Reserved	-	Reserved
19	Reserved	Reserved	-	Reserved
20	Reserved	Reserved	-	Reserved
21	TIM_OUT0	Timer 0	*)	Timer 0 Interrupt
22	TIM_OUT1	Timer 1	*)	Timer 1 Interrupt
23	TIM_OUT2	Timer 2	*)	Timer 2 Interrupt
24	TIM_OUT3	Timer 3	*)	Timer 3 Interrupt
25	TIM_OUT4	Timer 4	*)	Timer 4 Interrupt
26	TIM_OUT5	Timer 5	*)	Timer 5 Interrupt
27	WD_INT_ARM926	Watchdog ARM926	F/S	ARM926 Watchdog Interrupt
28	SPI1_TFE	SPI1	P	Transmit FIFO empty
29	SPI1_RNE	SPI1	P	Receive FIFO not empty (entspricht SPI1_SSPRXDMA)
30	SPI2_TFE	SPI2	P	Transmit FIFO empty

31	SPI2_RNE	SPI2	P	Receive FIFO not empty (entspricht SPI2_SSPRXDMA)
32	GPIO0	External Interrupt	**)	GPIO
33	GPIO1	External Interrupt	**)	GPIO
34	GPIO2	External Interrupt	**)	GPIO
35	GPIO3	External Interrupt	**)	GPIO
36	GPIO4	External Interrupt	**)	GPIO
37	GPIO5	External Interrupt	**)	GPIO
38	GPIO6	External Interrupt	**)	GPIO
39	GPIO7	External Interrupt	**)	GPIO
40	GPIO8	External Interrupt	**)	GPIO
41	GPIO9	External Interrupt	**)	GPIO
42	GPIO10	External Interrupt	**)	GPIO
43	GPIO11	External Interrupt	**)	GPIO
44	GPIO12	External Interrupt	**)	GPIO
45	GPIO13	External Interrupt	**)	GPIO
46	GPIO14	External Interrupt	**)	GPIO
47	GPIO15	External Interrupt	**)	GPIO
48	EDC_Event	EDC_EVENT Register	P	Combined Interrupt (EDC-Event Register)
49	PLL-Lock	Clock Unit	F/S	PLL Lock State
50	PLL-Loss	Clock Unit	F/S	PLL Loss State
51	AHB Address Error	Multilayer AHB	F/S	Access to non-existing addresss at the AHB
52	APB Address Error	AHB/APB Bridge	F/S	Access to non-existing addresss at the APB
53	EMC Address Error	EMC	P	Access to non-existing addresss in the EMC address area
54	PER_IF_ARM_IRQ	Per_IF	P	Event Interrupt
55	P1/2_INTERP	PHY0/1	F/S	Interrupt von PHY1/2
56	PN_IRQ2(**0**)	PN-IP (PN-ICU2)	F/S	PN Combined Interrupt
57	PN_IRQ2(**1**)	PN-IP (PN-ICU2)	F/S	PN Combined Interrupt
58	PN_IRQ2(**2**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
59	PN_IRQ2(**3**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
60	PN_IRQ2(**4**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
61	PN_IRQ2(**5**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
62	PN_IRQ2(**6**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
63	PN_IRQ2(**7**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
64	PN_IRQ2(**8**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
65	PN_IRQ2(**9**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
66	PN_IRQ2(**10**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt

67	PN_IRQ2(**11**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
68	PN_IRQ2(**12**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
69	PN_IRQ2(**13**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
70	PN_IRQ2(**14**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
71	PN_IRQ2(**15**)	PN-IP (PN-ICU2)	F/S	selectable special PN Interrupt
72	PNPLL_OUT9	PNPLL	F/S	selectable PNPLL Output
73	PNPLL_OUT10	PNPLL	F/S	selectable PNPLL Output
74	PNPLL_OUT11	PNPLL	F/S	selectable PNPLL Output
75	PNPLL_OUT12	PNPLL	F/S	selectable PNPLL Output
76	PNPLL_OUT13	PNPLL	F/S	selectable PNPLL Output
77	PNPLL_OUT14	PNPLL	F/S	selectable PNPLL Output
78	SW_INT_0	Software Int's	-	Software interrupt (fixed in hardware)
79	SW_INT_1	Software Int's	-	Software interrupt (fixed in hardware)
80	SW_INT_2	Software Int's	-	Software interrupt (fixed in hardware)
81	SW_INT_3	Software Int's	-	Software interrupt (fixed in hardware)
82	SW_INT_4	Software Int's	-	Software interrupt (fixed in hardware)
83	SW_INT_5	Software Int's	-	Software interrupt (fixed in hardware)
84	SW_INT_6	Software Int's	-	Software interrupt (fixed in hardware)
85	SW_INT_7	Software Int's	-	Software interrupt (fixed in hardware)
86	Modul_Access_Error	Modul_Access_ Error Register	F/S	Combined Interrupt for address missmatches in modules
87	Reserved		-	
88	Invalid I-TCM926 Access	ARM926 Sub-system	F/S	ARM926 hat in ein I-TCM Adresslücke gegriffen
89	Invalid D-TCM926 Access	ARM926 Sub-system	F/S	ARM926 hat in ein D-TCM Adresslücke gegriffen
90	Reserved		-	
91	Reserved		-	
92	Reserved		-	
93	Reserved		-	
94	Reserved		-	
95	Reserved		-	

Table 18: ARM IRQ-Interrupts

The IRQ Interrupt output of the ARM-ICU (ICUIRQ_O) is switched inverted to the Interrupt output (nIRQ) of the ARM926EJ-S.

5.4.2 Fast-Interrupts at the ARM Interrupt Controller (FIQ)

The Interrupt sources listed in Table 19 are used as high-priority interrupts. The interrupts for FIQ serve for example for debugging.

Interrupt FIQ<n>	Interrupt Name	Interrupt Source	Type Inverter	Description
0	DEFAULT_VEC	-	-	Default Vector **must be tied to '0' on top level**
1	Optional	**Optionally from IRQs**	F/S	Freely programmable IRQ source
2	Optional	**Optionally from IRQs**	F/S	Freely programmable IRQ source
3	Optional	**Optionally from IRQs**	F/S	Freely programmable IRQ source
4	Optional	**Optionally from IRQs**	F/S	Freely programmable IRQ source
5	Optional	**Optionally from IRQs**	F/S	Freely programmable IRQ source
6	Optional	**Optionally from IRQs**	F/S	Freely programmable IRQ source
7	Optional	**Optionally from IRQs**	F/S	Freely programmable IRQ source

Table 19: FIQ-Interrupts

The FIQ Interrupt output of the ARM-ICU (ICUFIQ_O) is switched inverted to the Interrupt output (nFIQ) of the ARM926EJ-S.

6 MISCELLANEOUS

6.1 Directory of abbreviations / terms

AHB	AMBA **A**dvanced **H**ighperformance **B**us
AMBA	**A**dvanced **M**icrocontroller **B**us **A**rchitecture
APB	AMBA **A**dvanced **P**eripheral **B**us
AR	**A**pplication **R**elationship (z.B. Controller <-> Device)
CR	**C**ommunication **R**elationship (z.B. Input Data / Output Data)
D_TCM	**D**ata **T**ightly **C**oupled **M**emory
DFT	**D**esign **F**or **T**est
DFP	**D**ynamic **F**rame **P**acking
GDMA	**G**enerell **D**irect **M**emory **A**ccess
EMC	**E**xternal **M**emory **C**ontroller (before **EMIF**)
ETM	**E**mbedded **T**race **M**acrocell
FIQ	**F**ast **I**nterrupt **Req**uest
GPIO	**G**eneral **P**urpose **I**nput/**O**utput
HW	**H**ard**w**are
I_TCM	**I**nstruction **T**ightly **C**oupled **M**emory
ICU	**I**nterrupt **C**ontrol **U**nit
IRQ	**I**nterrupt **Req**uest
IRT	**I**sochrones **Real**time
ISR	**I**nterrupt **S**ervice **R**egister
IP	Intellectual Property
JTAG	**J**oint **T**est **A**ction **G**roup
MC	**M**otion **C**ontrol
NMI	**N**on **M**askable **I**nterrupt
PCB	**P**rinted **C**ircuit **B**oard
PHY	**Ph**ysical Layer
PLL	**P**hase **L**ocked **L**oop
RT	**Real**time
SCRB	**S**ystem **C**ontrol **R**egister **B**lock
SDRAM	**S**ynchronous **D**ynamic **RAM**
SPI	**S**tandard Serial **P**eripheral **I**nterface
SW	**S**oft**w**are
t.b.d.	**T**o **b**e **d**efined
TCM	**T**ightly **C**oupled **M**emory
TIA	**T**otally **I**ntegrated **A**utomation
UART	**U**niversal **A**synchronous **R**eceiver/Transmitter
UC	**U**se **C**ase
XHIF	**H**ost **I**nter**f**ace **(**before **LBU**)

6.2 Literature list

/1/ IEEE 802.1D – 2004 (MAC-Bridges)
/2/ IEEE 802.1Q – 2003 (VLANs)
/3/ IEEE1149.1 (Boundary Scan)
/4/ ETM9 Technical Reference Manual (Rev. r2p2) (ARM DDI 0157G)
/5/ ETM Specification (ARM IHI 0014 Q)
/6/ ARM9E-S Technical Reference Manual (Rev. 1)(ARM DDI 0165B)
/7/ ARM926EJ-S Technical Reference Manual (Version: r0p5)(ARM DDI 0198E)
/9/ Using Embedded ICE; Appl. Note 31; Issue C
/15/ DDI0183G_uart_pl011_r1p4_trm.pdf, UART (PL011) Technical Reference Manual
/16/ IEC 61158-5-10 V2.3 (PNO) und IEC 61158-6-10 V2.3 (PNO)
/29/ I 2C-Bus Specification, V2.1 von 01.2000
/46/ ETB11 Technical Reference Manual (ARM DDI0275D)